ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

Teória a aplikácie párových ansámblových modelov

Dizertačná práca

28360020233002

Študijný program: Aplikovaná informatika

Študijný odbor: Informatika

Pracovisko: Katedra matematických metód a operačnej analýzy

Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

Školiteľ: doc. Mgr. Ondrej Šuch, PhD.

Školiteľ špecialista: Ing. Peter Tarábek, PhD.

Žilina, 2023

Ing. René Fabricius

Čestne prehlasujem, že som prácu vypracoval samostatne s využitím dostupnej literatúry a vlastných vedomostí. Všetky zdroje použité v dizertačnej práci som uviedol v súlade s predpismi.

Súhlasím so zverejnením práce a jej výsledkov.

V Žiline, dňa Meno Priezvisko

Abstrakt

FABRICIUS René, Ing.: Teória a aplikácie párových ansámblových modelov. [Dizertačná práca] Žilinská univerzita v Žiline. Fakulta riadenia a informatiky. Katedra matematických metód a operačnej analýzy. - Školiteľ: doc. Mgr. Ondrej Such, PhD. - Žilina: FRI ŽU, 2023, 166 strán.

Klasifikačné ansámblové modely produkujú klasifikačné predikcie pomocou kombinácie viacerých klasifikačných algoritmov, z ktorých sú zložené. Ansámblové modely sú z dôvodu svojej robustnosti a presnosti súčasťou oblasti strojového učenia už od jej počiatkov. Ich hlavnou výhodou je schopnosť kombinovať rôznorodé predikcie svojich členov a získať tak predikciu často lepšiu, ako by vyprodukoval ktorýkoľvek člen samostatne. V úvode práce poskytujeme prehľad existujúcich ansámblových metód, ich vlastností a stavebných blokov, špeciálnu pozornosť venujeme párovým ansámblovým modelom. Jadrom práce je tvorba novej parametrickej ansámblovej metódy využívajúcej binarizáciu. Metóda priraďuje váhy jednotlivým dvojiciam tried každého kombinovaného klasifikátora. Navrhovanú metódu nazývame Vážený lineárny ansámbel (WLE). Práca zahŕňa návrh metódy, tvorbu metodológie jej trénovania a použitia, ladenie hyperparametrov metódy a otestovanie metódy na niekoľkých datasetoch. Na datasetoch CIFAR-100 a ImageNet porovnávame navrhovanú metódu s populárnym priemerovacím ansámblom. Výsledky týchto experimentov ukazujú, že navrhovaná metóda vo väčšine prípadov produkuje kvalitnejšie predikcie. V práci skúmame tiež možnosť využitia WLE metódy na detekciu neznámych vzoriek a porovnávame ju so zaužívanými prístupmi na riešenie tejto úlohy. Experimenty na úlohe rozlišovania medzi datasetmi CIFAR-10 a CIFAR-100 ukazujú, že aplikácia metódy maximálnej predikovanej pravdepodobnosti na výstupy WLE poskytuje lepšiu detekciu neznámych vzoriek, ako aplikácie tej istej metódy na výstupy priemerovacieho ansámblu.

Kľúčové slová: klasifikačná úloha, lineárne klasifikačné metódy, neurónové siete, ansámblové modely

Abstract

FABRICIUS René, Ing.: Theory and Applications of Pairwise Ensemble Models. [Dissertation thesis] University of Žilina in Žilina. Faculty of Management Science and Informatics. Department of Mathematical Methods and Operations Research. - Thesis supervisor: doc. Mgr. Ondrej Šuch, PhD. - Žilina: FRI ŽU, 2023, 166 pages.

Ensemble models for classification form their predictions by combining the outputs of their constituent classifiers. Due to their accuracy and robustness, they are part of the machine learning field since its early days. The main advantage of ensemble models is their ability to combine diverse predictions of their members and to form a prediction that is often better than a prediction of any constituent member. In this work, we study the structure of a general ensemble model, its basic building blocks, and their properties. We also provide an overview of popular existing ensemble models. We focus our attention on the category of pairwise ensemble models. The core part of the thesis is the formulation of a new parametric ensembling method utilizing binarization. This method assigns weights to each pair of classes for each combined classifier. We call this new method a Weighted linear ensemble (WLE). The thesis includes the construction of the WLE method, the creation of the methodology for training and use of the method, hyperparameters tuning, and testing on several datasets. We compare the proposed method with an averaging baseline ensemble on datasets CIFAR-100 and ImageNet. The results of these experiments show the superior performance of the WLE method in the majority of cases. We also examine the potential of the WLE method for out-of-distribution (OOD) detection functionality by comparing it with existing OOD detection methods. Experiments on the problem of distinguishing datasets CIFAR-10 and CIFAR-100 show, that applying the method of maximum softmax probability on the outputs of WLE provides better OOD detection than applying the same method on the outputs of the baseline averaging ensemble.

Keywords: classification task, linear classifiers, neural networks, ensemble models

Obsah

Zoznam skratiek

BC	Bayesovská kovariancia (Bayes covariance)
CNN	Konvolučné neurónové siete (Convolutional neural networks)
DNN	Hlboké neurónové siete (Deep neural networks)
LDA	Lineárna diskriminačná analýza (Linear discriminant analysis)
NN	Neurónové siete (Neural networks)
SGD	Stochastický pokles gradientu (Stochastic gradient descent)
SVM	Metóda podporných vektorov (Support vector machines)
WLE	Vážený lineárny ansámbel (Weighted linear ensemble)
MSP	Maximálna pravdepodobnostná predikcia (Maximum softmax probability)
OOD	(vzorky) Mimo rozdelenia (Out of distribution)
IND	(vzorky) V rozdelení (In distribution)

Zoznam obrázkov

Úvod

Keď stojíme pred komplexnou a náročnou úlohou je bežnou praxou obrátiť sa na tím expertov s rozličnými relevantnými oblasťami expertízy. V súlade s porekadlom: "Viac hláv, viac rozumu," má takýto tím pri vhodnom spôsobe spolupráce väčšiu šancu dospieť k dobrému riešeniu, ako jednotliví jeho členovia samostatne.

Na základe podobnej filozofie sa v oblasti strojového učenia vytvárajú "tímy" predikčných modelov spolupracujúcich na riešení zadaného problému. Takéto "tímy" sa označujú pojmom ansámblové modely. Použitie ansámblového prístupu je umožnené existenciou veľkého množstva rôznorodých predikčných modelov, ktoré majú rozličné silné a slabé stránky.

V poslednom čase vo viacerých oblastiach strojového učenia prevláda použitie hlbokých umelých neurónových sietí (DNN). Rýchly vývoj v oblasti DNN má za následok dostupnosť veľkého množstva rozličných architektúr sietí s rôznymi vlastnosťami. Trénovanie takýchto sietí je stochastický proces, ktorý je možné do veľkej miery modifikovať a prispôsobiť nastavením rozličných parametrov, čo má za následok rozličné výsledné modely aj pri použití jednej architektúry. Tieto skutočnosti robia z DNN vhodných kandidátov na vytváranie ansámblových modelov.

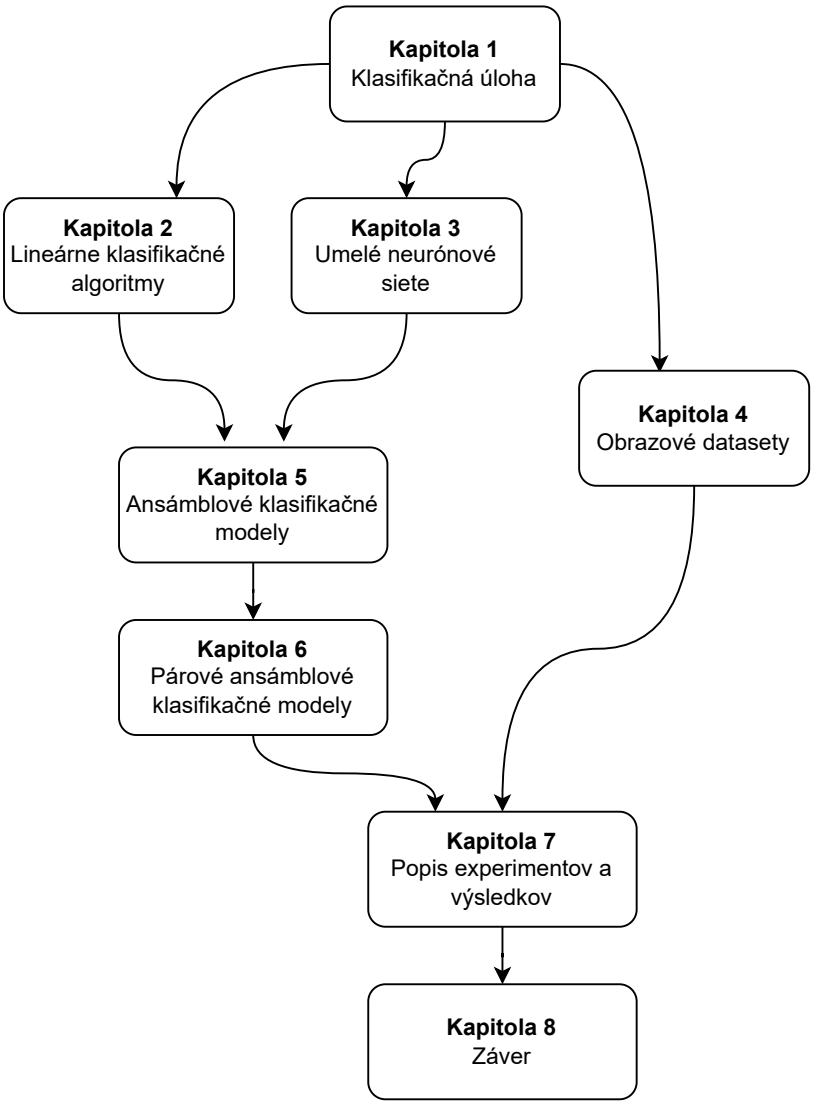
Pre získanie výsledného rozhodnutia ansámblového modelu je potrebné skombinovať rozhodnutia jednotlivých jeho členov. Existuje veľké množstvo menej, či viac sofistikovaných kombinačných metód, ktoré to umožňujú. Niektoré kombinačné metódy sú jednoduché algebraické pravidlá, iné zahŕňajú komplexné algoritmy, ktoré vyžadujú vlastné trénovanie. Súčasťou niektorých komplexnejších kombinačných metód sú jednoduchšie metódy štatistického učenia, v niektorých prípadoch ale aj neurónové siete. Výber vhodnej kombinačnej metódy záleží, ako od druhu riešenej úlohy, tak aj od typu kombinovaných modelov.

V úvode práce poskytujeme prehľad existujúcich ansámblových modelov, ich možné

využitie a základné stavebné bloky. Zameriavame sa na klasifikačné úlohy, špecificky na klasifikáciu obrazu. Naša pozornosť sa sústredí na druh ansámblov využívajúci binarizáciu, nazývaný tiež párové ansámble. Párové ansámble majú svoj pôvod pri kombinovaní inherentne dvojtriednych klasifikátorov za účelom viactriednej klasifikácie. Historicky sa párové ansámble objavili pri zovšeobecnení SVM na viactriednu klasifikáciu.

Jadrom práce je návrh novej párovej ansámblovej metódy. Na rozdiel od prvotných párových ansámblových metód, naša metóda kombinuje viactriedne klasifikátory a jej výstupom je tiež viactriedna klasifikácia. Súčasťou metódy je rozloženie vstupných klasifikátorov na dvojtriedne klasifikátory a ich následné kombinovanie.

Práca je rozčlenená do ôsmych kapitol. Prvá kapitola definuje klasifikačnú úlohu a základné pojmy s ňou spojené. Druhá kapitola sa venuje lineárnym klasifikačným metódam, ktoré súvisia s párovými ansámblami, alebo majú využitie v kombinačných metódach. Tretia kapitola vysvetľuje fungovanie umelých neurónových sietí a bližšie sa venuje konvolučným neurónovým sieťam a obrazovým transformerom, ktoré sa využívajú na spracovanie obrazu. Štvrtá kapitola predstavuje použité datasety, uvádza ich parametre a problémy. Piata kapitola sa zaoberá princípmi ansámblových modelov a vysvetľuje fungovanie niekoľkých populárnych ansámblových metód. Šiesta kapitola sa sústredí na párové ansámblové modely. Prvá sekcia popisuje niekoľko existujúcich párových zväzovacích metód. Vo zvyšných sekciách navrhujeme spôsob použitia párových zväzovacích metód na kombinovanie viactriednych klasifikátorov a vytvárame tak novú ansámblovú metódu. Siedma kapitola popisuje experimenty a ich výsledky, ktoré vedú k vybudovaniu metodiky použitia navrhnutej ansámblovej metódy a odladeniu jej hyperparametrov. Odladenú metódu porovnávame s existujúcou štandardne využívanou ansámblovou metódou. Venujeme sa tu tiež otestovaniu funkcionality detekcie neznámych vzoriek. Ôsma kapitola obsahuje zhrnutie dosiahnutých výsledkov a navrhuje možné smerovanie ďalšieho výskumu.



Obrázok 1: Diagram logickej nadväznosti kapitol.

Kapitola 1

Klasifikačná úloha

Klasifikačná úloha je jednou zo základných úloh strojového učenia. Funkcie, ktoré aproximujú riešenie klasifikačnej úlohy sa nazývajú klasifikátory. Vstupom do klasifikátora je usporiadaný súbor číselných príznakov $\boldsymbol{x}$ klasifikovaného objektu a očakávaným výstupom je trieda y klasifikovaného objektu. Trieda y patrí do konečnej množiny klasifikovaných tried. Vo všeobecnejšom prípade môže byť výstupom klasifikátora tiež rozdelenie pravdepodobnosti $\boldsymbol{p}$ nad množinou klasifikovaných tried. Klasifikáciu je možné použiť na rôzne druhy objektov, ako sú napríklad obraz, zvuk, alebo text. Stačí, aby sme dokázali z objektu vyextrahovať usporiadaný súbor príznakov $\boldsymbol{x}$.

Príkladom niekoľkých klasických klasifikátorov sú lineárna diskriminačná analýza (LDA) [1], rozhodovacie stromy [2], metóda podporných vektorov (SVM) [3] a rozličné druhy neurónových sietí. Napriek tomu, že klasifikáciu je možné vykonávať na jednorozmernom vektore príznakov $\boldsymbol{x}$, viaceré klasifikátory s výhodou využívajú štruktúru konkrétneho druhu dát. Na spracovanie dát, ktoré sú prirodzene usporiadané v postupnosti, ako zvuk, text alebo sieťová prevádzka sú špeciálne prispôsobené rekurentné neurónové siete [4]. Iný druh neurónových sietí, konvolučné neurónové siete (CNN) [5], je prispôsobený na spracovanie dát s priestorovou štruktúrou. Klasickým príkladom takýchto dát sú obrázky, alebo trojrozmerné rádiologické dáta. Pre spracovanie dát prirodzene reprezentovaných ako graf sú prispôsobené grafové neurónové siete [6].

Všetky spomínané metódy patria do kategórie algoritmov učenia s učiteľom. Pri svojom trénovaní teda potrebujú označené dáta. Trénovacia množina pozostáva z usporiadaných dvojíc vektora príznakov $\boldsymbol{x}_i$ a označenia triedy y_i: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$. Trénovanie každého klasifikátora realizuje špeciálny trénovací algoritmus. Tento al-

goritmus dostane k dispozícii trénovaciu množinu S a vyprodukuje klasifikačný model. Konkrétna inštancia klasifikačného modelu je určená hodnotami jej parametrov $\boldsymbol{\theta}$. Klasifikačný model môže byť interpretovaný ako hypotéza h, ktorá aproximuje neznámu funkciu f: $y = f(\boldsymbol{x})$, kde f je funkcia, ktorá priradí každej prípustnej vzorke $\boldsymbol{x}$ správnu triedu y. Trénovanie modelu môže potom byť interpretované ako prehľadávanie množiny hypotéz H, ktoré je trénovaný klasifikačný model schopný reprezentovať.

Kapitola 2

Lineárne klasifikačné algoritmy

Medzi prvé klasifikačné metódy využívajúce štatistické učenie patria logistická regresia a lineárna diskriminačná analýza. Obe tieto metódy rozdeľujú priestor prediktorov pomocou lineárnych nadrovín. Logistická regresia je založená na využití logistickej funkcie, ktorú ako jeden z prvých použil Verhulst pri študovaní vývoja populácie v prostredí s obmedzenými zdrojmi [7].

Lineárna diskriminačná analýza (LDA) je založená na práci R. A. Fishera [1], v ktorej sa venoval rozlišovaniu troch druhov kosatcov na základe meraní šírky a dĺžky okvetných a kališných lupienkov.

Príkladom klasifikačnej metódy, ktorá dokáže modelovať aj nelineárne hranice medzi triedami, je metóda podporných vektorov (SVM) [3]. Táto metóda bola vyvinutá v 90. rokoch minulého storočia rozšírením skoršej lineárnej metódy klasifikátora maximálneho rozpätia. SVM funguje na princípe transformácie príznakových vektorov do viacrozmerného priestoru a ich následnej lineárnej separácii.

Vyššie spomínané metódy nie sú špeciálne prispôsobené úlohe klasifikácie obrazu, ktorá je témou práce, využívame ich však pri budovaní ansámblov z iných špecializovaných metód.

2.1 Logistická regresia

Logistická regresia [8, Chapter 3][9, Chapter 4] vo svojej základnej podobe vykonáva dvojtriednu klasifikáciu. Pri klasifikovaní modeluje pravdepodobnosť, že objekt patrí do konkrétnej triedy. Triedy môžeme označiť ako 0 a 1 a pre zjednodušenie môžeme

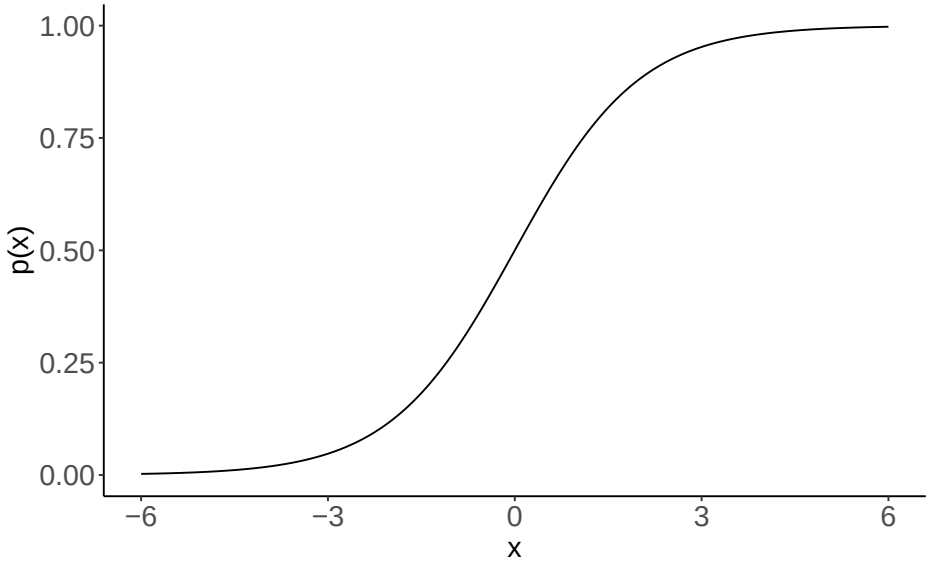
pravdepodobnosť príslušnosti objektu do triedy 1 označiť ako $p(\boldsymbol{x}) = P(Y = 1|\boldsymbol{x})$, kde Y je premenná označujúca triedu. Táto pravdepodobnosť je modelovaná pomocou logistickej funkcie so všeobecným predpisom

$$f(x) = \frac{\exp(x)}{1 + \exp(x)}. \tag{2.1}$$

Graf tejto funkcie je zobrazený na obrázku 2.1. V logistickej regresii ako parameter do tejto funkcie vstupuje lineárna kombinácia prediktorov $\boldsymbol{x}$, výsledná funkcia má potom podobu:

$$p(\boldsymbol{x}) = \frac{\exp(\beta_0 + \boldsymbol{\beta_1}^T \boldsymbol{x})}{1 + \exp(\beta_0 + \boldsymbol{\beta_1}^T \boldsymbol{x})}, \tag{2.2}$$

kde $\beta_0, \boldsymbol{\beta_1}$ sú parametre modelu. Ako je z predpisu a grafu logistickej funkcie zrejmé,



Obrázok 2.1: Graf logistickej funkcie.

funkcia je definovaná na množine všetkých reálnych vektorov a nadobúda hodnoty medzi 0 a 1. Hodnoty parametrov $\beta_0, \boldsymbol{\beta_1}$ sa určujú pri procese trénovania. Ak vyjadríme logaritmus pomeru pravdepodobností príslušnosti do prvej a druhej triedy, tiež nazývaný logit:

$$\ln\left(\frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})}\right) = \beta_0 + \boldsymbol{\beta_1}^T \boldsymbol{x}, \tag{2.3}$$

vidíme, že je lineárny vzhľadom k vektoru prediktorov $\boldsymbol{x}$.

Trénovanie modelu logistickej regresie spočíva v maximalizácii funkcie nazývanej maximálna vierohodnosť (ang. maximum likelihood)

$$l(\beta_0, \boldsymbol{\beta_1}) = \prod_{i:y_i=1} p(\boldsymbol{x_i}) \prod_{j:y_j=0} (1 - p(\boldsymbol{x_j})). \tag{2.4}$$

Pri praktickej realizácii sa pre lepšie numerické vlastnosti maximalizuje logaritmus maximálnej vierohodnosti. V prípade dobrej separovateľnosti jednotlivých tried, alebo pri malej veľkosti trénovacej množiny môže byť výpočet parametrov modelu logistickej regresie nestabilný. V týchto prípadoch stojí za zváženie použitie regularizácie, alebo inej klasifikačnej metódy, ako napríklad LDA [8, Chapter 3][9, Chapter 4].

2.2 Lineárna diskriminačná analýza

LDA na rozdiel od Logistickej regresie najprv modeluje rozdelenie pravdepodobnosti príznakového vektora $\boldsymbol{x}$ pre jednotlivé triedy a následne s použitím Bayesovej vety vypočíta odhady pre aposteriórne pravdepodobnosti $P(Y = k \mid \boldsymbol{x})$. Hustotu rozdelenia pravdepodobnosti príznakového vektora pre triedu k označíme ako $f_k(\boldsymbol{x}) = P(\boldsymbol{x} \mid Y = k)$. Pre aplikovanie Bayesovej vety potrebujeme ešte poznať apriórne pravdepodobnosti pre jednotlivé triedy, teda pravdepodobnosti, že náhodne vybrané pozorovanie bude patriť do triedy k. Apriórnu pravdepodobnosť pre triedu k označíme ako $\pi_k = P(Y = k)$. Podobne ako v predchádzajúcej časti označíme $p_k(\boldsymbol{x}) = P(Y = k \mid \boldsymbol{x})$. S týmito označeniami môžeme Bayesovu vetu zapísať ako

$$p_k(\boldsymbol{x}) = \frac{\pi_k f_k(\boldsymbol{x})}{\sum_{l=1}^{K} \pi_l f_l(\boldsymbol{x})}, \tag{2.5}$$

kde K je počet tried. Aby sme mohli odhad $p_k(\boldsymbol{x})$ vyčísliť, potrebujeme určiť rozdelenie $f_k(\boldsymbol{x})$. Pri metóde LDA sa $f_k(\boldsymbol{x})$ modeluje viacrozmerným normálnym rozdelením, ktorého hustota je daná rovnicou

$$f_k(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k))}{\sqrt{(2\pi)^p |\boldsymbol{\Sigma}_k|}}, \tag{2.6}$$

kde p je počet prvkov príznakového vektora, $\boldsymbol{\mu}_k$ je stredná hodnota príznakových vektorov pre triedu k a $\boldsymbol{\Sigma}_k$ je kovariančná matica pre triedu k. LDA ďalej predpokladá, že kovariančné matice sú pre všetky triedy rovnaké

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}, \ldots, \boldsymbol{\Sigma}_K = \boldsymbol{\Sigma},. \tag{2.7}$$

Za týchto predpokladov je možné logaritmovaním $p_k(\boldsymbol{x})$ a odčítaním členov, ktoré sú pre každú triedu rovnaké, vyjadriť pre každú triedu lineárnu diskriminačnú funkciu

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \log(\pi_k). \tag{2.8}$$

Priradenie pozorovania s príznakovým vektorom $\boldsymbol{x}$ do triedy s najvyššou hodnotou $p_k(\boldsymbol{x})$ je potom ekvivalentné jeho priradeniu do triedy s najvyššou hodnotou $\delta_k(\boldsymbol{x})$. Ako je z rovnice 2.8 zrejmé, lineárna diskriminačná funkcia je vzhľadom k $\boldsymbol{x}$ skutočne lineárna.

Trénovanie LDA modelu pozostáva z odhadnutia parametrov π_k, $\boldsymbol{\mu}_k$ a $\boldsymbol{\Sigma}$. Tieto parametre sa odhadujú z trénovacích dát $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ podľa rovníc

$$\begin{aligned} \hat{\pi}_k &= N_k/N \\ \hat{\boldsymbol{\mu}}_k &= \sum_{i:y_i=k} \boldsymbol{x}_i/N_k \\ \hat{\boldsymbol{\Sigma}} &= \sum_{k=1}^{K} \sum_{i:y_i=k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^T/(N-K), \end{aligned} \tag{2.9}$$

kde K je počet tried, N je počet všetkých trénovacích pozorovaní a N_k je počet trénovacích pozorovaní pre triedu k.

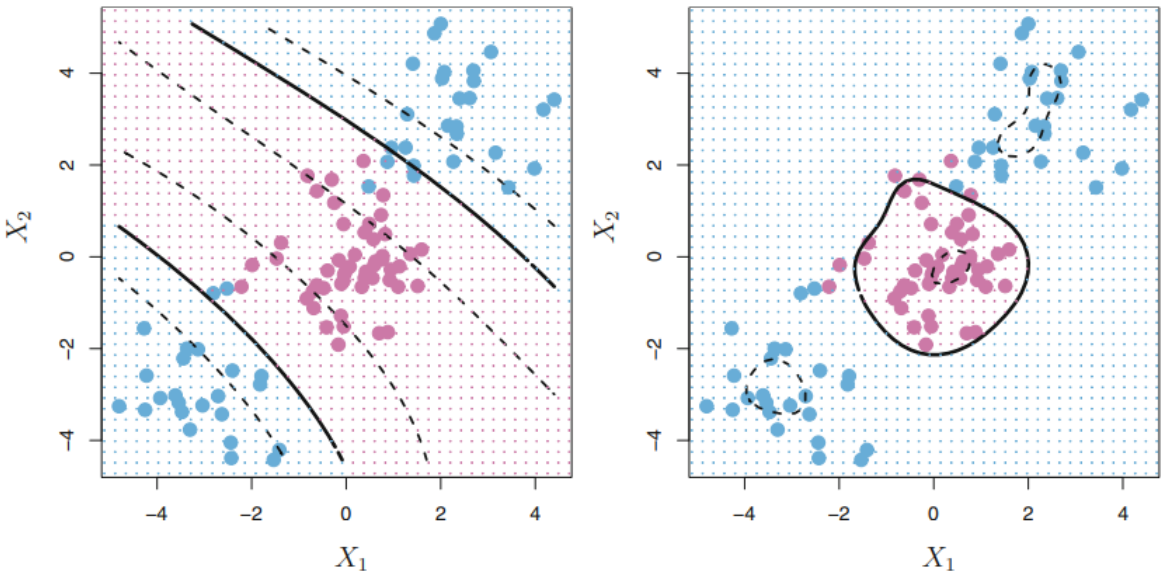
2.3 Metóda podporných vektorov

Metóda podporných vektorov (SVM) [3] je dvojtriedna klasifikačná metóda, ktorá vznikla ako rozšírenie klasifikátora maximálneho rozpätia. Klasifikátor maximálneho rozpätia delí priestor príznakových vektorov na dve časti pomocou nadroviny. Tento klasifikátor hľadá takú nadrovinu, aby pozorovania z prvej triedy ležali na jednej strane a pozorovania z druhej triedy na druhej strane. Naviac pre túto nadrovinu musí platiť, že spomedzi všetkých prípustných nadrovín má okolo seba najširší pás, v ktorom neležia žiadne trénovacie pozorovania.

Pre prípad keď trénovacie dáta nie sú lineárne separovateľné vzniklo rozšírenie v podobe klasifikátora podporných vektorov. Tento klasifikátor dovoľuje trénovacím dátam, aby ležali vnútri pásu okolo nadroviny, alebo aj na nesprávnej strane nadroviny, snaží sa ale tieto prípady minimalizovať.

Metóda podporných vektorov ďalej rozširuje klasifikátor podporných vektorov aplikovaním kernelového triku. Kernel v tomto kontexte predstavuje funkciu dvoch vektorov, ktorá je zovšeobecnením skalárneho súčinu vektorov, a ktorá musí mať určité vlastnosti. Tento trik umožňuje dosiahnuť rovnaký výsledok, ako by dosiahla transformácia príznakových vektorov do viacrozmerného priestoru a ich oddelenie v tomto priestore pomocou nadroviny. V pôvodnom priestore príznakových vektorov takto

vznikne nelineárna oddeľujúca hranica [9, Chapter 9]. Príklad použitia dvoch takýchto kernelov môžeme vidieť na obrázku 2.2.



Obrázok 2.2: Príklad použitia polynomiálneho kernelu stupňa tri (vľavo) a radiálneho kernelu (vpravo) na oddelenie lineárne neseparovateľných dát. Farebnými bodkami sú znázornené trénovacie pozorovania, pričom farbou je označená ich trieda. Plnou čiarou je znázornená nájdená oddeľujúca nadrovina a čiarkovanými čiarami okraje pásu okolo oddeľujúcej nadroviny. (Prevzaté z [9, Chapter 9])

SVM je populárna klasifikačná metóda, jej obmedzenie na dve triedy ale neumožňovalo jej využitie na viactriednu klasifikáciu. Boli preto vyvinuté metódy, ktoré kombináciou viacerých dvojtriednych klasifikátorov umožňujú vykonávať viactriednu klasifikáciu. Jedným z možných postupov je vytvoriť párový klasifikátor pre každú dvojicu tried. Takéto párové klasifikátory môžu byť potom kombinované hlasovaním - výstupná trieda klasifikácie je tá trieda, ktorá vyhrá najväčší počet párových porovnaní. Ak sú ale párové klasifikátory schopné produkovať ako svoj výstup pravdepodobnosť príslušnosti do jednotlivých tried, ako je tomu aj u SVM s rozšírením podľa [10], je možné použiť aj sofistikovanejšie párové zväzovacie metódy. Jednu z takýchto metód navrhli autori v práci [11], kde ju aj otestovali na klasifikátore k najbližších susedov, LDA a SVM. Ďalšie dve párové kombinačné metódy boli navrhnuté v práci [12]. Tieto metódy získali značnú popularitu, aj vďaka ich implementácii v knižnici LIBSVM [13].

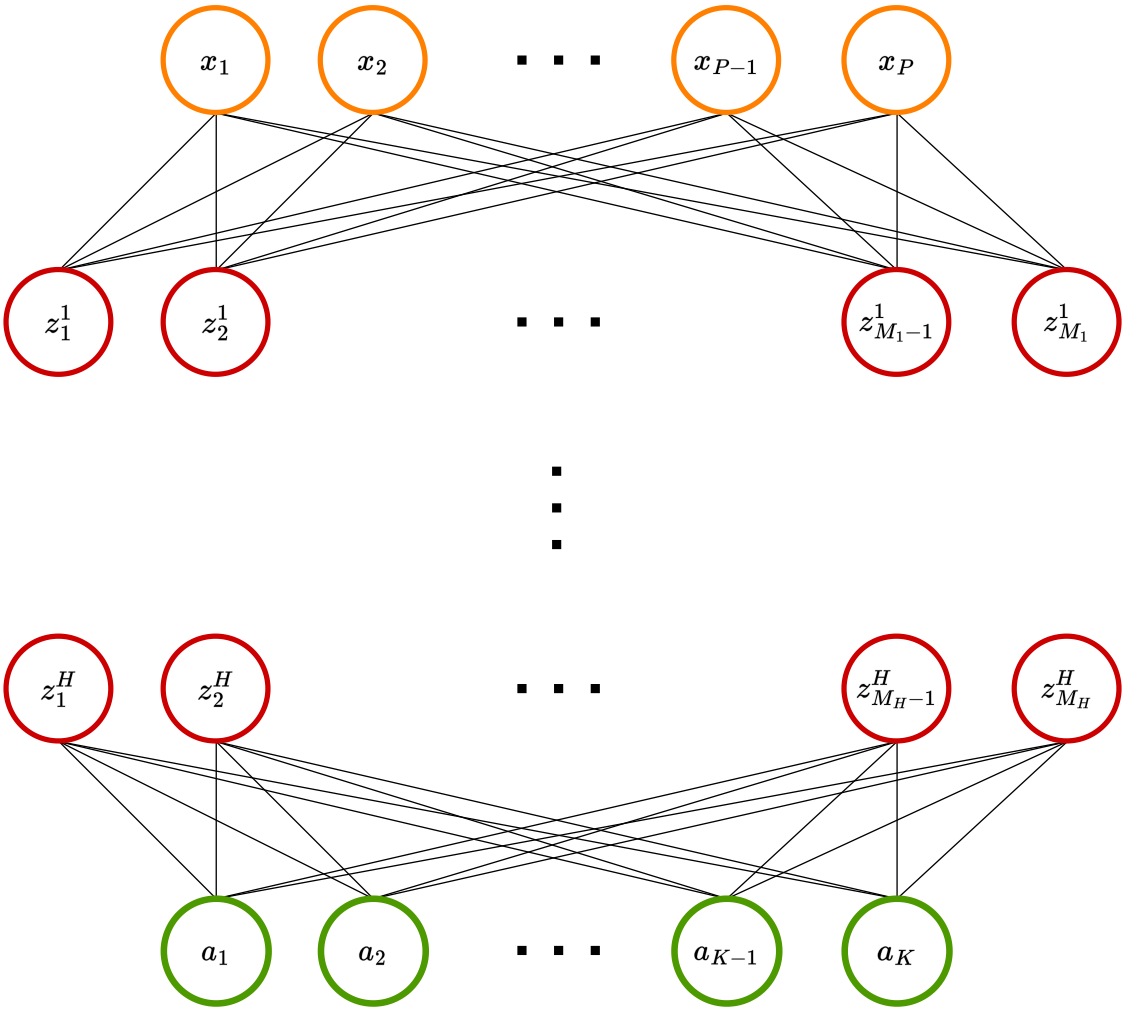
Kapitola 3

Umelé neurónové siete

Umelé neurónové siete sú štatistické modely strojového učenia inšpirované biologickými neurónovými sieťami. Ich stavebné prvky sa preto označujú rovnako ako v biológii - neuróny. Matematické základy pre umelé neuróny boli vyvinuté v práci [14] a neskôr boli aplikované ako modely s jedným [15] a neskôr s viacerými [16] neurónmi. Objavenie trénovacej metódy spätná propagácia gradientov (ang. gradient backpropagation) [17] umožnilo rozvoj sietí s viacerými skrytými vrstvami. Takéto siete sa nazývajú hlboké neurónové siete. Neskôr bolo dokázané, že siete so skrytými vrstvami a nelineárnymi aktivačnými funkciami sú schopné aproximovať ľubovoľnú spojitú funkciu [18, 19]. Táto vlastnosť z nich robí užitočné nástroje štatistického učenia, čo sa odráža na ich aktuálnej širokej popularite v regresných, ako aj klasifikačných úlohách. Hlboké neurónové siete boli prispôsobené použitiu na spracovanie rozličných typov dát. V tejto práci sú pre nás zaujímavé najmä konvolučné neurónové siete [5] a obrazové transformery [20], ktoré sú prispôsobené na spracovanie obrazu.

3.1 Hlboké umelé neurónové siete

Štruktúru hlbokej doprednej neurónovej siete s P vstupmi, K výstupmi, H skrytými vrstvami a M_h neurónmi v jednotlivých skrytých vrstvách $h = 1, 2, \ldots, H$ je možné reprezentovať pomocou diagramu zobrazeného na obrázku 3.1. Ako je z diagramu vidieť, sieť sa skladá z niekoľkých vrstiev. Prvá vrstva, označená ako x_p pre $p = 1, 2, \ldots, P$, predstavuje vstupy do siete, za ňou nasleduje H skrytých vrstiev, s neurónmi označenými ako z_m^h pre $h = 1, 2, \ldots, H$ a $m = 1, 2, \ldots, M_h$. Výstupy zo siete



Obrázok 3.1: Schéma neurónovej siete s viacerými skrytými vrstvami. Krúžky s označeniami x_i predstavujú vstupy, krúžky s označeniami z_m^h predstavujú neuróny skrytých vrstiev a krúžky s označeniami a_k predstavujú výstupy.

sú zobrazené v poslednej vrstve, označenej ako a_k pre $k = 1, 2, \ldots, K$. Siete s vysokým počtom skrytých vrstiev sa nazývajú hlboké neurónové siete (DNN). Výstup každého neurónu v skrytej vrstve z_m^h je vytvorený aplikovaním nelineárnej aktivačnej funkcie σ na lineárnu kombináciu výstupov neurónov z predchádzajúcej vrstvy. V prípade prvej skrytej vrstvy sa σ aplikuje na lineárnu kombináciu prvkov vstupného vektora

$$\begin{aligned} z_m^1 &= \sigma\Big(\sum_{p=1}^{P} w_{pm}^1 x_p + b_m^1\Big), && \text{pre } m = 1, 2, \ldots, M_1 \\ z_m^h &= \sigma\Big(\sum_{i=1}^{M_{h-1}} w_{im}^h z_i^{h-1} + b_m^h\Big), && \text{pre } h = 2, 3, \ldots, H, m = 1, 2, \ldots, M_h, \end{aligned} \tag{3.1}$$

kde b_m^h je bias neurónu číslo m vo vrstve h, a w_{im}^h je váha spojenia medzi i-tym neurónom vrstvy $h-1$ a m-tým neurónom vrstvy h [8, Chapter 11].

Ako aktivačná funkcia v skrytých vrstvách sa zvykol využívať sigmoid

$$\sigma(v) = 1/(1 + e^{-v}) \tag{3.2}$$

alebo tangens hyperbolický. Pri hlbších sieťach ale tieto funkcie spôsobovali problém miznúcich gradientov (ang. vanishing gradients problem) [21]. Jedným z možných riešení tohto problému je použitie aktivačnej funkcie ReLU

$$\sigma(v) = \begin{cases} v, & \text{pre } v > 0 \\ 0, & \text{inak} \end{cases}, \tag{3.3}$$

alebo niektorého jej variantu [22]. Aplikácia nelineárnej aktivačnej funkcie σ je dôvodom nelinearity celého modelu. V prípade použitia lineárnej σ by bol model lineárny a dal by sa teda nahradiť jednoduchou lineárnou formou.

Vstupom do siete je číselný vektor príznakov $\boldsymbol{x} = (x_1, x_2, \ldots, x_P)$. Podoba výstupného vektora $\boldsymbol{a} = (a_1, a_2, \ldots, a_K)$ závisí od typu riešenej úlohy. Pri regresnej úlohe sa zvykne používať $K = 1$, teda jeden výstup. Pri klasifikačnej úlohe počet výstupov zodpovedá počtu tried riešenej úlohy. Výstupy sa počítajú aplikovaním výstupnej aktivačnej funkcie g_k na lineárne kombinácie výstupov poslednej skrytej vrstvy. Pri klasifikačnej úlohe sa spravidla využíva výstupná aktivačná funkcia softmax

$$g_k(\boldsymbol{b}) = \frac{\exp(b_k)}{\sum_{i=1}^{K} \exp(b_i)}. \tag{3.4}$$

Potom jednotlivé výstupy spočítame ako

$$\begin{aligned} l_k &= \sum_{i=1}^{M_H} w_{ik}^{H+1} z_i^H + b_k^{H+1}, \qquad \text{pre } k = 1, 2, \ldots, K \\ f_k(\boldsymbol{x}) &= a_k = g_k(\boldsymbol{l}), \qquad\qquad \text{pre } k = 1, 2, \ldots, K \end{aligned} \tag{3.5}$$

kde $\boldsymbol{l} = (l_1, l_2, \ldots, l_K)$, konštanty w_{ik}^{H+1}, b_k^{H+1} plnia rovnakú úlohu ako pri výstupoch neurónov skrytých vrstiev a vektorová funkcia $\boldsymbol{f} = (f_1, f_2, \ldots, f_K)$ reprezentuje činnosť celej DNN. Lineárne kombinácie výstupov poslednej skrytej vrstvy $\boldsymbol{l}$, na ktoré sa aplikuje softmax sa nazývajú logity. Pri použití softmaxu môžeme takto spočítané výstupy interpretovať ako pravdepodobnosti príslušnosti do jednotlivých tried.

Parametre b_m^h a w_{ij}^h sa nazývajú váhy siete. Množina všetkých týchto váh sa ozna-

čuje ako $\boldsymbol{\theta}$ a zahŕňa:

$$\begin{gathered}\{b_m^h; \text{pre } h = 1, 2, \ldots, H, m = 1, 2, \ldots, M_h\},\\ \{b_k^{H+1}; \text{pre } k = 1, 2, \ldots, K\},\\ \{w_{ij}^h; \text{pre } h = 2, 3, \ldots, H, i = 1, 2, \ldots, M_{h-1}, j = 1, 2, \ldots, M_h\},\\ \{w_{ij}^1; \text{pre } i = 1, 2, \ldots, P, j = 1, 2, \ldots, M_1\},\\ \{w_{ij}^{H+1}; \text{pre } i = 1, 2, \ldots, M_H, j = 1, 2, \ldots, K\}.\end{gathered} \tag{3.6}$$

Štruktúra siete, teda počet vstupov, výstupov, počet skrytých vrstiev a počty neurónov v nich, ako aj použité aktivačné funkcie sa súhrnne označujú ako architektúra siete.

3.2 Trénovanie neurónových sietí

Pred použitím neurónovej siete je nutné ju natrénovať. Trénovanie pozostáva z nastavovania váh siete tak, aby výstupy zo siete dobre zodpovedali trénovacím dátam. Nevyhnutným krokom pred začiatkom trénovania je nastavenie hyperparametrov trénovania. Hyperparametre určujú, ako bude trénovanie prebiehať a ich hodnoty je potrebné prispôsobiť architektúre siete, ako aj riešenej úlohe. Proces trénovania a popis základných hyperparametrov je popísaný v nasledujúcich odsekoch.

Pri trénovaní klasifikačnej neurónovej siete sa používa takzvané "one-hot" kódovanie závislých premenných y_i. Ak pracujeme s K triedami, potom ako označenie triedy y_i používame K prvkový vektor $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iK})$, ktorý obsahuje samé nuly a jednotku na tom indexe, ktorý zodpovedá správnej triede pozorovania i.

Na určenie toho, ako dobre výstupy zo siete korešpondujú s trénovacími dátami sa používa chybová funkcia $R(\boldsymbol{\theta})$. V prípade klasifikácie v nasledujúcich rovniciach uvažujeme one-hot kódovanie závislých premenných $\boldsymbol{y}_i$ v trénovacej množine. Pre regresiu a zriedkavejšie aj pre klasifikáciu sa ako chybová funkcia využíva suma štvorcových chýb

$$R(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(\boldsymbol{x}_i))^2. \tag{3.7}$$

Minimalizácia tejto chybovej funkcie je ekvivalentná minimalizácii Brierovho skóre [23], keďže sa tieto dve funkcie líšia len konštantným faktorom $1/N$. Populárnejšou alternatívou pre klasifikáciu je ale vo väčšine prípadov krížová entropia

$$R(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(f_k(\boldsymbol{x}_i)). \tag{3.8}$$

V procese trénovania sa snažíme chybovú funkciu minimalizovať. Väčšinou však cieľom nie je nájsť globálne minimum, pretože to môže viesť k zohľadňovaniu detailov a zákonitostí, ktoré platia iba v použitej trénovacej množine a nie sú platné vo všeobecnosti. Takýto stav sa nazýva pretrénovaním.

Technicky sa na minimalizáciu chybovej funkcie používa metóda nazývaná spätná propagácia gradientov. Ide o metódu pozostávajúcu z dvoch krokov: dopredný prechod a spätný prechod. V doprednom prechode sú váhy siete konštantné a počítajú sa predikované hodnoty $f_k(\boldsymbol{x}_i)$ podľa (3.5). V spätnom prechode sa z nich spočítajú chyby a gradienty a tie sa spätne šíria a upravujú váhy siete. Miera týchto úprav je daná hyperparametrom **rýchlosť učenia** (ang. learning rate). Teoreticky by sa mali chyby spočítať pre všetky trénovacie dáta a až potom by sa malo prejsť k výpočtu gradientov a ich spätnému šíreniu. Takýto prístup je ale veľmi výpočtovo náročný a v praxi sa zvyknú trénovacie dáta spracovávať po dávkach. Preto sa tento prístup nazýva stochastický gradientový zostup (SGD). Veľkosť týchto dávok je daná hyperparametrom **veľkosť dávky** (ang. batch size). Jeden prechod cez všetky trénovacie dáta sa nazýva epocha. Počet epoch pri trénovaní je riadený ďalším hyperparametrom.

Aby proces spätnej propagácie gradientov fungoval, je pred začiatkom trénovania potrebné nastaviť váhy siete na nenulové hodnoty. Keďže je chybová funkcia $R(\boldsymbol{\theta})$ nekonvexná a má viacero lokálnych miním, má toto počiatočné nastavenie váh výrazný vplyv na ich výsledné hodnoty po trénovaní.

Dôležitým problémom, s ktorým sa pri trénovaní neurónových sietí stretávame je pretrénovanie. Bolo vyvinutých viacero postupov, ktoré umožňujú tento problém riešiť.

Jedným z nich je **včasné zastavenie učenia** (ang. early stopping). Pre určenie vhodného času zastavenia je užitočné mať k dispozícii validačnú množinu dát. Na validačnej množine je možné periodicky kontrolovať presnosť modelu a keď začne táto presnosť klesať, čo naznačuje stratu schopnosti modelu zovšeobecňovať, trénovanie sa zastaví.

Ďalším možným prístupom je zoslabovanie alebo **regularizácia váh** (ang. weight decay). Tento prístup sa snaží o udržanie váh siete na hodnotách blízkych nule. Prakticky sa realizuje pridaním pokutovej funkcie $J(\boldsymbol{\theta})$ k chybovej funkcii. Vo výsledku sa potom minimalizuje $R(\boldsymbol{\theta}) + \lambda J(\boldsymbol{\theta})$, kde λ je hyperparameter určujúci aká váha je

prikladaná pokutovej funkcii. Často používané podoby pokutovej funkcie sú L1 norma

$$J(\boldsymbol{\theta}) = \sum_{m,h} |b_m^h| + \sum_{i,j,h} |w_{ij}^h|, \tag{3.9}$$

alebo L2 norma

$$J(\boldsymbol{\theta}) = \sum_{m,h} \left(b_m^h\right)^2 + \sum_{i,j,h} \left(w_{ij}^h\right)^2. \tag{3.10}$$

Ďalší, odlišný prístup realizuje metóda **dropout**. **Dropout** pri trénovaní neurónovej siete vynuluje výstupy niektorých náhodne vybraných neurónov. Náhodný výber neurónov sa deje s pravdepodobnosťou zadanou hyperparametrom. Tento hyperparameter môže byť nastavený pre každú vrstvu na inú hodnotu.Vynulovaním výstupov niektorých neurónov sa v praxi dosiahne to, že v danom kroku sa na výpočte predikcie, ako aj na trénovaní podieľajú iba niektoré neuróny. Reálne tak vždy trénujeme len určitú náhodne vybranú podsieť celej siete.

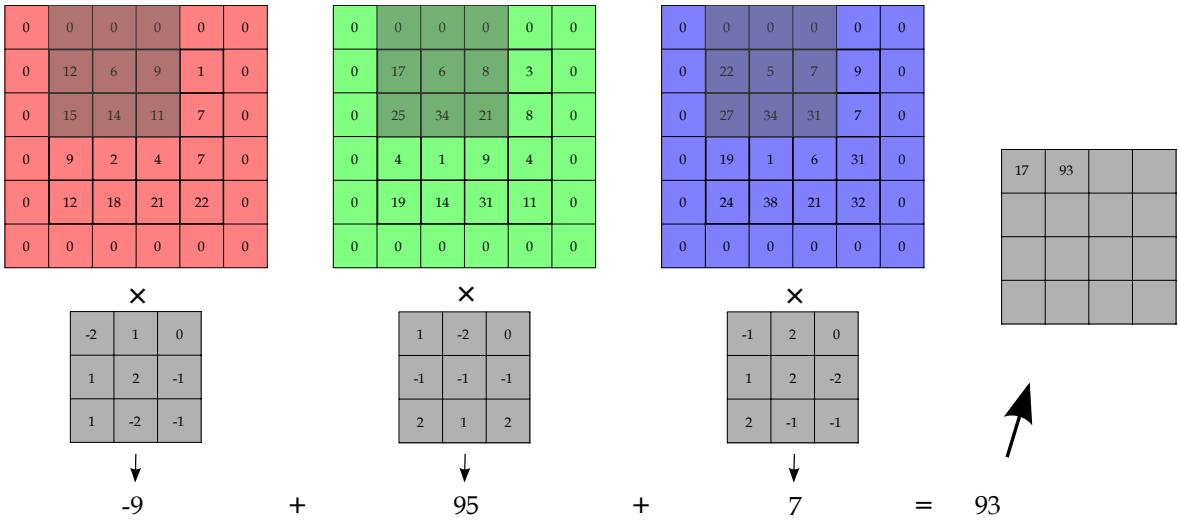
3.3 Konvolučné neurónové siete

Neurónové siete, v podobe v akej boli predstavené, sú užitočným nástrojom strojového učenia [24]. Svoj vstup však prijímajú vo forme jednorozmerného vektora, čo im neumožňuje v plnej miere využiť dvojrozmerný charakter obrazových dát. Pre lepšie prispôsobenie dvoj a viac rozmerným dátam boli vyvinuté konvolučné vrstvy a siete, ktoré ich využívajú sa nazývajú konvolučné neurónové siete [5].

Konvolučné neurónové siete (CNN) zvyčajne pozostávajú z niekoľkých konvolučných vrstiev nasledovaných plne prepojenými vrstvami. CNN na spracovanie obrazu využívajú 2D konvolúciu. 2D konvolúcia je proces pri ktorom sa vstupné dáta spracujú aplikovaním kernelu. V tomto kontexte je kernel mriežka s pevne určenými rozmermi, ktorá sa skladá z trénovaných váh. Veľkosť kernelu je zvyčajne $3 \times 3 \times$ počet kanálov.

Aplikácia jedného kernelu na celý dvojrozmerný vstup šetrí počet váh modelu a tiež umožňuje zohľadniť vzťahy medzi susediacimi pixelmi. Pri aplikácii sa kernel posúva určeným krokom (ang. stride) po celom vstupnom obraze. Hodnota v obraze sa vynásobí zodpovedajúcou hodnotou v kerneli a výsledky tohto násobenia sa sčítajú cez celý kernel. Následne sa sčítajú hodnoty cez všetky kanály, čím vznikne jedna číselná hodnota pre každé umiestnenie kernelu. Z týchto hodnôt je tvorený výstup konvolučnej vrstvy. Pokiaľ chceme, aby mal výstup rovnaké rozmery ako mal vstup, je

potrebné okolo vstupných dát pridať orámovanie šírky jedna (pre kernel 3×3 a krok jedna), ktoré môže byť vyplnené nulami (ang. zero padding). Aplikáciu kernelu na obrazové dáta môžeme vidieť na obrázku 3.2. V každej vrstve je možné použiť viacero kernelov, čím vznikne výstup s viacerými kanálmi.

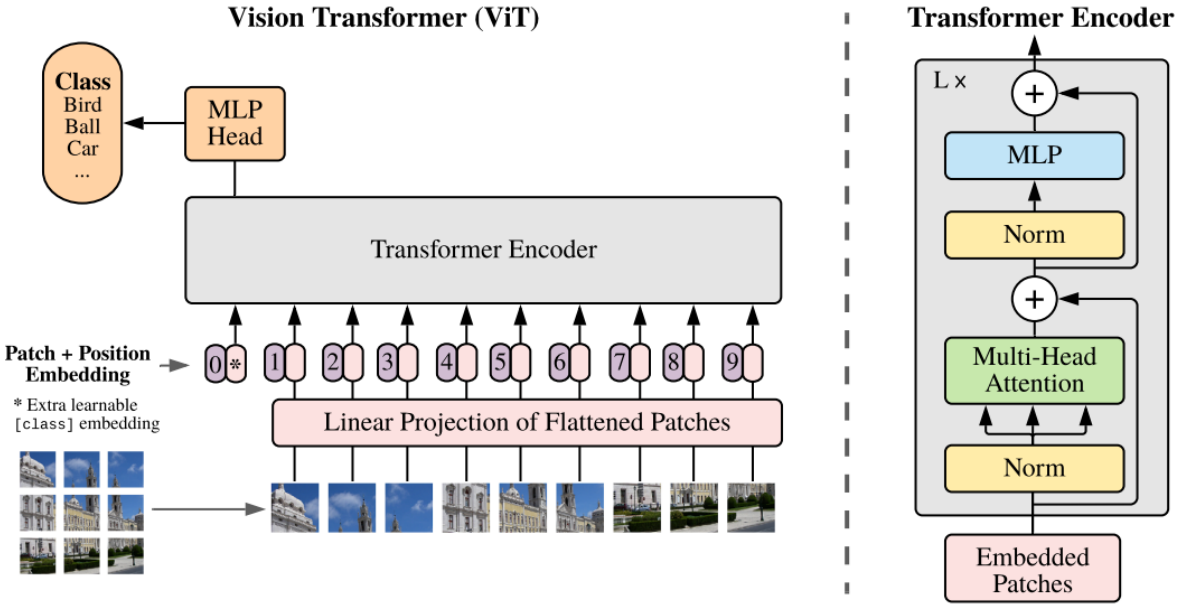


Obrázok 3.2: Princíp aplikovania konvolučného kernelu na vstupný farebný obrázok. Kanály vstupného obrázka zodpovedajú základným farbám: červenej, zelenej a modrej. Kernel má veľkosť 3×3 a na vstup je aplikovaný zero padding.

3.4 Obrazové transformery

Transformery sú neurónové siete, ktorých architektúra bola pôvodne navrhnutá na spracovanie sekvenčných dát, akými je hovorená reč, alebo text. Vo svojej štandardnej podobe transformery spracovávajú vstupný reťazec a produkujú výstupný reťazec. Základnými črtami ich architektúry je enkóder-dekóder schéma a využitie mechanizmu nazývaného **pozornosť** (ang. attention). Pozornosť umožňuje transformerom pri enkódovaní každého prvku vstupnej sekvencie priradiť rôzne váhy informáciám od ostatných prvkov vstupnej sekvencie. Podobne pri dekódovaní umožňuje pozornosť priraďovať rôzne váhy informáciám z jednotlivých úrovní procesu enkódovania a tiež informáciám od už vytvorených prvkov výstupnej sekvencie. Takýto prístup umožňuje zohľadňovať závislosti medzi prvkami vstupnej sekvencie, ktoré sú od seba vzdialené [25]. Transformery dosiahli významné úspechy v oblasti spracovania jazyka a upútali tým pozornosť ostatných oblastí strojového učenia.

Obrazové dáta nemajú prirodzene jednorozmernú štruktúru a preto na ich spracovanie nebolo možné použiť transformery v ich základnej podobe. Autori v [20] navrhli modifikáciu, ktorá rozdelí obrázok na niekoľko menších štvorcových častí, tieto zakóduje do jednorozmerných vektorov a následne ich spracuje ako sekvenčné dáta. Do zakódovaných vektorov sa pridáva trénovaná informácia, ktorá je schopná naučiť sa vzájomné polohy spracovávaných vektorov. Schéma práce obrazového transformeru je znázornená na obrázku 3.3. Podobne ako v oblasti spracovania reči, aj v oblasti spraco-



Obrázok 3.3: Schéma spracovania obrazového vstupu pomocou obrazového transformeru. Na rozdiel od klasického transformeru, obrazový transformer využíva iba enkóder časť architektúry. Obrázok prevzatý z [20].

vania obrazu dosiahli transformery významné úspechy a v mnohých testoch prekonali konvolučné neurónové siete [26]. Existuje tiež niekoľko projektov s verejne dostupným zdrojovým kódom, ktoré dávajú k dispozícii predtrénované modely veľkých obrazových transformerov[1,2]. Tieto modely sú natrénované na obrovskom množstve dát (300M-400M obrázkov) s veľkým počtom tried. Takéto modely je možné dotrénovať a použiť na úlohy spracovania obrazu na zvolenom datasete.

[1]https://github.com/google-research/vision_transformer
[2]https://github.com/openai/CLIP

3.5 Kalibrácia neurónových sietí

Žiaducou vlastnosťou pravdepodobnostného klasifikátora je, aby predikované pravdepodobnosti zodpovedali skutočným pravdepodobnostiam. Ak teda klasifikátor predikuje výslednú triedu s pravdepodobnosťou 70%, mala by takáto predikcia byť správna v 70% prípadov. Moderné neurónové siete ale často túto vlastnosť nemajú [27]. Tento problém sa nazýva kalibrácia spoľahlivosti a na jeho riešenie bolo vyvinuté množstvo rôznych prístupov [27, 28, 29].

Jednoduchým a populárnym spôsobom je teplotné škálovanie (ang. temperature scaling). Ide o metódu, ktorá využíva jeden parameter - teplotu t s pomocou ktorej upravuje výsledné pravdepodobnosti. Teplotné škálovanie vstupuje do predikčného procesu pred aplikovaním softmax funkcie podelením jej vstupu teplotou t. Táto úprava nemení poradie preferencie pre predikované triedy, nemá teda vplyv na presnosť klasifikátora. Teplotu t je potrebné natrénovať na validačných dátach. Tréning sa realizuje po dokončení trénovania neurónovej siete. Pri trénovaní sa najprv vypočítajú logity $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_V$ poslednej plne prepojenej vrstvy pre validačné dáta $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_V, \boldsymbol{y}_V)\}$. Následne sa rieši optimalizačná úloha

$$\underset{t}{\operatorname{argmin}} \left(-\sum_{i=1}^{V} \sum_{k=1}^{K} y_{ik} \log(g_k(\boldsymbol{l}_i/t)) \right). \tag{3.11}$$

Aby bolo možné problém kalibrácie efektívne identifikovať a riešiť, je potrebné určiť spôsob výpočtu odhadu kalibračnej chyby. Tu tiež existuje niekoľko prístupov, z ktorých väčšina je založená na použití histogramu s predikovanou spoľahlivosťou na vodorovnej osi a skutočnou presnosťou na zvislej. V práci používame metódu navrhnutú v [28], ktorá automaticky určí počet intervalov v histograme a jednotlivé intervaly volí tak, aby obsahovali približne rovnaký počet prvkov.

Kapitola 4

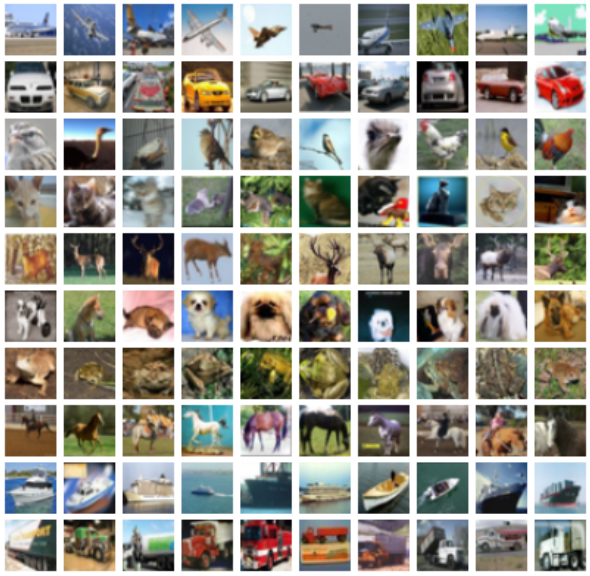
Obrazové datasety

Neoddeliteľnou súčasťou implementačného procesu metód strojového učenia s učiteľom je trénovanie. Klasifikačné metódy predstavené v tejto práci všetky spadajú do kategórie metód učenia s učiteľom. Pre úspešný rozvoj takýchto klasifikačných metód sú potrebné kvalitné datasety, na ktorých je možné testovať nové inovácie a porovnávať ich s najmodernejšími (ang. state-of-the-art) riešeniami. V nasledujúcej časti preto predstavíme niekoľko populárnych datasetov. V práci sa venujeme klasifikácii obrazu, preto sa zameriame na obrazové datasety.

4.1 Obrazové datasety

CIFAR-10 a CIFAR-100

CIFAR-10 a CIFAR-100 obsahujú farebné obrázky veľkosti 32x32 pixelov. Datasety boli vytvorené pomocou manuálneho anotovania a s použitím automatickej detekcie duplikátov [30] z veľkého datasetu malých obrázkov Tiny Images [31] s hrubými, nespoľahlivými označeniami tried. Oba datasety pozostávajú z 50000 trénovacích obrázkov a 10000 testovacích obrázkov. Obrázky v datasete CIFAR-10 sú rozdelené do 10 tried, pričom každá trieda má rovnaké zastúpenie ako v trénovacej, tak aj v testovacej množine. CIFAR-100 je rozdelený do 20 nad-tried, pričom každá z nich obsahuje 5 pod-tried. Dokopy teda obsahuje 100 tried, ktoré, rovnako ako v datasete CIFAR-10, majú rovnomerné zastúpenie. Množiny tried CIFAR-10 a CIFAR-100 sú disjunktné.



Obrázok 4.1: Ukážka niekoľkých náhodne vybraných obrázkov z datasetu CIFAR-10 [30].

ImageNet

Dataset ImageNet je vytváraný podľa štruktúry WordNet [32]. WordNet organizuje slová do grafovej štruktúry podľa ich významu. Vrcholy grafu predstavujú skupiny synoným. Triedy datasetu ImageNet kopírujú tieto skupiny synoným. ImageNet obsahuje farebné obrázky rôznych veľkostí. Dataset je vytváraný pomocou automatického vyhľadávania obrázkov na internete a ich následného manuálneho filtrovania [33]. Na manuálne filtrovanie je použitá platforma Amazon Mechanical Turk[1]. V čase písania tejto práce bolo spracovaných viac ako 14 miliónov obrázkov patriacich do viac ako 20 tisíc množín synoným. Aktuálny stav datasetu je možné sledovať na oficiálnom webe ImageNet[2]. V kapitole 7 tento dataset označujeme ako ImageNet21k.

Každoročne v rozmedzí rokov 2010 až 2017 bola organizovaná súťaž ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [34] pre ktorú bola vybraná podmnožina tried z aktuálneho datasetu ImageNet. V súťaži sa okrem klasifiácie riešila tiež úloha lokalizácie jedného objektu a úloha detekcie a lokalizácie viacerých objektov v obrázku. Trénovacie a validačné datasety z týchto súťaží sú voľne dostupné. Dataset pre klasifikáciu obsahuje približne 1.2 milióna trénovacích obrázkov v 1000 rôznych triedach, validačná množina obsahuje 50 tisíc obrázkov. Tieto datasety sa na rozdiel od celého

[1]https://www.mturk.com/

[2]https://www.image-net.org/

datasetu ImageNet nemenia a sú preto často používané na testovanie a porovnávanie nových algoritmov počítačového videnia. V práci používame dataset ILSVRC2012 a označujeme ho ako ImageNet1k.

Obrázok 4.2: Príklad obrázkov z datasetu ImageNet [34]. Triedy obrázkov zľava doprava: morský had, polievková miska, sup.

4.2 Problémy veľkých datasetov

Ako môžeme vidieť z popisu obrazových datasetov vyššie, proces ich tvorby je spravidla poloautomatický, alebo manuálny. Pri tvorbe takýchto datasetov môže dochádzať k chybám ako v automatickej časti, tak aj v časti manuálnej anotácie. Chyby vznikajú ako v trénovacej [35], tak aj v testovacej [36, 37] časti datasetov. Výskum v oblasti hlbokých neurónových sietí ukázal vysokú odolnosť voči šumu pri trénovaní [38, 39, 40]. Šum v testovacej časti datasetov predstavuje preto väčší problém, keďže podľa presnosti na testovacej množine sa zvyčajne vyberá najlepší model na použitie v praxi. Problém chybných anotácií v testovacej časti viacerých datasetov skúmali autori v [36]. Potencionálne chyby vyhľadávali s pomocou modelu strojového učenia [35] a následne ich manuálne overili pomocou crowdsourcingu. Zistené podiely chybných označení v testovacích množinách skúmaných obrazových datasetov sú zobrazené v tabuľke 4.1.

Autori v [36] tiež skúmali, ako si rôzne modely strojového učenia počínajú na chybne označených testovacích obrázkoch pred a po opravení označení. Zistili, že na chybne označených obrázkoch dosahujú najvyššiu presnosť komplexnejšie modely, zatiaľ čo pri vyhodnotení presnosti s opravenými označeniami ich ďaleko prekonajú jednoduchšie modely.

K podobným zisteniam dospeli aj autori v článku [37], kde sa zamerali špecificky

Dataset	Veľkosť obrázkov (v pixeloch)	Počet tried	Podiel chybných označení v testovacej množine
CIFAR-10	32×32	10	0.54%
CIFAR-100	32×32	100	5.85%
ILSVRC2012	priemerne 469×387	1000	5.83%

Tabuľka 4.1: Základné charakteristiky a podiely chybných označení obrázkov v testovacích množinách známych datasetov.

na ImageNet. V tomto článku autori tiež použili podobný postup zmenšenia množiny obrázkov, ktoré je potrebné manuálne skontrolovať, pomocou modelov strojového učenia. V tejto práci autori umožnili každému obrázku priradiť aj viacero označení, keďže na jednom obrázku sa často vyskytuje viacero objektov. Podiel testovacích obrázkov ImageNetu u ktorých zistili nesprávne označenia bol v tejto práci až 9.98%. V práci sa tiež venovali skúmaniu závislosti presnosti modelov na pôvodných označeniach a presnosti na opravených označeniach. Medzi týmito dvoma veličinami zistili lineárnu závislosť. Pre komplexnejšie modely, ktoré dosahovali vyššiu presnosť mala ale táto závislosť nižší koeficient. Zlepšenie presnosti na pôvodných označeniach u jednoduchších modelov teda viedlo k vyššiemu relatívnemu zlepšeniu presnosti na opravených označeniach ako u komplexnejších modelov. Autori z toho usudzujú, že komplexnejšie modely sa naučili modelovať aj dôsledky procesov tvorby datasetu, ktoré do neho zaniesli chybné označenia [37].

Kapitola 5

Ansámblové klasifikačné modely

Pri prijímaní rozhodnutí o dôležitých verejných alebo osobných záležitostiach je bežné konzultovať názor viacerých strán. Finálne rozhodnutie je potom vytvorené so zohľadnením všetkých získaných názorov. Príklady takéhoto postupu môžeme vidieť v našom každodennom živote, nech už ide o vládu demokratického štátu, tím lekárov, alebo porotu rozhodcov na športovom podujatí, zohľadňovanie viacerých názorov znižuje pravdepodobnosť prijatia zlého rozhodnutia.

Podobný postup sa aplikuje aj v strojovom učení, kde sa označuje pojmom ansámblové metódy. Ansámblové metódy je možné využiť na riešenie ako regresných, tak aj klasifikačných úloh. Pri aplikácii ansámblovej klasifikačnej metódy sa kombinuje niekoľko individuálnych klasifikátorov. Tieto klasifikátory môžu patriť do rovnakého druhu, vtedy hovoríme o homogénnych ansámbloch, alebo do rozličných druhov, vtedy ide o heterogénne ansámble. Ansámblové modely rôznych druhov našli uplatnenie vo viacerých praktických aplikáciách. Niekoľko príkladov je: vyhodnocovanie kvality senzorických dát [41], rozpoznávanie osôb v interiéri áut [42] a rozličné medicínske aplikácie ako predikcia biologickej aktivity farmaceutických molekúl [43], určovanie regiónov dôležitých pre klasifikáciu v spektre magnetickej rezonancie [44] alebo diagnóza rakoviny prsníka [45].

Aby mohla ansámblová metóda vyprodukovať klasifikátor poskytujúci lepšiu predikciu ako individuálne klasifikátory, musia byť chyby, ktoré tieto klasifikátory robia, rozličné [46]. Metódam, ktoré zabezpečujú rozličnosť (diverzitu) kombinovaných klasifikátorov, sa venujeme v sekcii 5.2.

Výsledná predikcia, ktorú ansámblový model vyprodukuje je kombináciou predikcií

jednotlivých členov ansámblu. Postupy, ktoré tieto kombinácie vytvárajú sa nazývajú kombinačné metódy. Štúdiu kombinačných metód sa venujeme v sekcii 5.3.

Pojmom ansámblová metóda tu budeme označovať postup vytvárania ansámblového klasifikačného modelu - klasifikátora. Ansámblová metóda teda špecifikuje ako získame individuálne klasifikátory, ako zabezpečíme ich diverzitu a ako ich skombinujeme. Samotný ansámblový klasifikačný model pozostáva z natrénovaných klasifikátorov a produkuje predikcie kombináciou predikcií týchto klasifikátorov.

5.1 Výhody ansámblov

Ansámblové modely sú často schopné dosahovať presnejšie a robustnejšie predikcie ako individuálne klasifikátory. Okrem toho, množstvo ansámblových modelov umožňuje dobrú modularitu, teda možnosť identifikovať a nahradiť tú časť modelu, ktorá sa podieľa na tvorbe určitého druhu chýb. V tejto sekcii sa venujeme niekoľkým teoretickým príčinám, ktoré zdôvodňujú dobré výsledky ansámblov. V týchto zdôvodneniach je využitá interpretácia činnosti trénovacieho algoritmu ako prehľadávania priestoru hypotéz H, ktoré je trénovaný model schopný reprezentovať, ako bola uvedená v prvej kapitole.

5.1.1 Štatistická výhoda

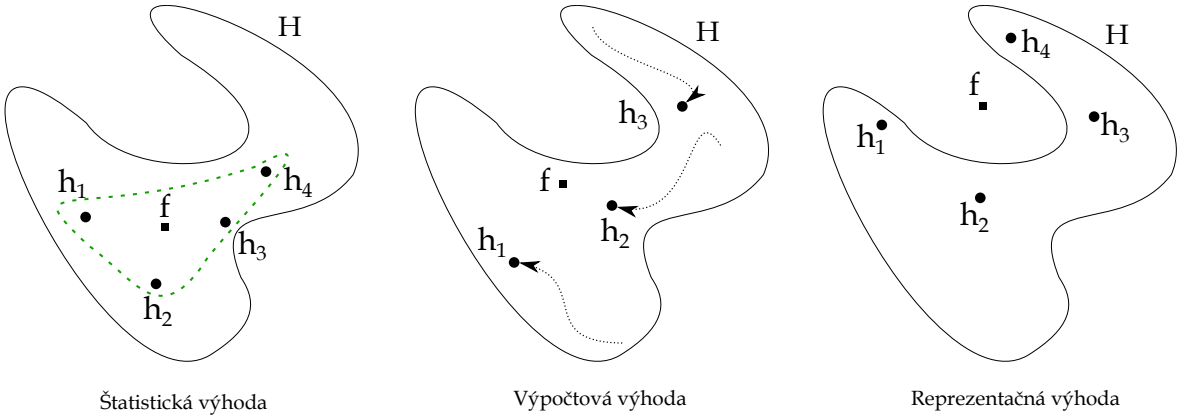
Štatistická výhoda sa prejaví v prípade, že trénovacia množina je príliš malá v porovnaní s množinou hypotéz, ktorú trénovací algoritmus prehľadáva. V takomto prípade sa môže stať, že trénovací algoritmus nájde viacero rôznych hypotéz - klasifikačných modelov, ktoré na trénovacích dátach dosahujú rovnakú presnosť. S použitím dostupných dát si teda trénovací algoritmus spomedzi týchto hypotéz nevie vybrať. Ansámblová metóda dokáže takéto hypotézy skombinovať, čím sa zníži pravdepodobnosť výberu obzvlášť zlého klasifikátora [47]. Vizualizácia tohto prípadu je zobrazená na obrázku 5.1 vľavo. Čiarkovanou čiarou je zobrazená podmnožina množiny H, v ktorej sa nachádzajú hypotézy s rovnakou presnosťou na trénovacích dátach. Na nových dátach môžu mať tieto hypotézy rôzne presnosti.

5.1.2 Výpočtová výhoda

Trénovacie algoritmy často optimalizujú kritérium, ktoré má viacero lokálnych extrémov. Optimálne riešenie takejto úlohy môže byť NP-ťažké, ako je tomu aj u neurónových sietí [48]. Optimálne trénovať klasifikátory teda často nie je možné a výsledkom trénovania je hypotéza v niektorom z lokálnych extrémov. Keď začneme trénovanie viac krát s rôznymi inicializáciami parametrov modelu, často skončíme v rôznych lokálnych extrémoch. Kombináciou takto získaných hypotéz sa môžeme lepšie priblížiť neznámej funkcii klasifikácie f [47]. Tento prípad je vizualizovaný na obrázku 5.1 v strede. Čiarkované čiary tu znázorňujú trajektórie trénovacieho algoritmu v priestore hypotéz.

5.1.3 Reprezentačná výhoda

Najmä v prípade jednoduchších klasifikátorov sa stáva, že nie sú schopné reprezentovať skutočnú neznámu funkciu f. Tento problém ale neabsentuje ani u komplexnejších klasifikátorov, akými sú neurónové siete, kde je priestor hypotéz, ktoré dokáže trénovací algoritmus prehľadať, limitovaný trénovacou množinou. **Reprezentačná výhoda** umožňuje kombináciou takýchto klasifikátorov rozšíriť priestor hypotéz, ktoré je ansámblový model schopný reprezentovať [47]. Znázornenie tohto prípadu je na obrázku 5.1 vpravo.



Obrázok 5.1: Vizualizácia výhod ansámblových modelov v porovnaní s individuálnymi klasifikátormi. Množina hypotéz H, ktoré dokáže klasifikačný model reprezentovať je znázornená plnou čiarou. Individuálne klasifikátory sú zobrazené ako krúžky s označeniami $h_1 - h_4$ a skutočná neznáma odhadovaná funkcia ako štvorec s označením f.

5.2 Metódy na vytváranie diverzity

Pre vytvorenie dobrého ansámblového modelu je potrebné aby jednotlivé klasifikátory, z ktorých sa skladá, poskytovali lepšie predikcie, ako náhodné hádanie a aby robili rozdielne chyby [46]. Bolo vyvinutých veľa rozličných metód, ktoré sa usilujú zabezpečiť diverzitu trénovaných klasifikátorov. Autori v [49] navrhli rozdelenie týchto metód do troch hlavných kategórií podľa toho, akým spôsobom pracujú:

- modifikácia štartovacieho bodu prehľadávania v priestore hypotéz,
- modifikácia množiny dostupných hypotéz,
- modifikácia spôsobu prehľadávania množiny hypotéz.

V nasledujúcich podsekciách podrobnejšie preberieme jednotlivé kategórie.

5.2.1 Modifikácie štartovacieho bodu prehľadávania

Pri neurónových sieťach môžeme štartovací bod prehľadávania meniť pomocou inicializácie váh siete. Štartovanie trénovacieho algoritmu z rozličných bodov v priestore hypotéz zvyšuje pravdepodobnosť, že výsledné modely budú odlišné. Pre menšie siete sa tento prístup ukázal ako málo efektívny [50]. Avšak, nové experimenty ukazujú, že

pre moderné hlboké siete je tento prístup dostatočný na tvorbu kvalitných ansámblových modelov [51, 52]. Podrobne sa tejto metóde v kontexte hlbokých neurónových sietí venujú autori v práci [53].

5.2.2 Modifikácie množiny dostupných hypotéz

Množina dostupných hypotéz môže byť modifikovaná prostredníctvom úprav v architektúre trénovaného modelu, alebo pomocou úprav trénovacej množiny.

Úpravy architektúry trénovaného modelu

Autori v práci [50] zistili, že pre malé siete sú úpravy architektúry, alebo použitie rozličných architektúr o niečo efektívnejšie vo vytváraní diverzity ako náhodné inicializácie váh sietí, stále je to však pre tieto siete málo efektívny prístup. Pre väčšie siete je prístup použiteľný [54, 55].

Výber správnej množiny architektúr je náročný proces a preto boli vyvinuté automatické nástroje, ktoré tento krok zabezpečujú. Nástroj Addemup [54] využíva na vytváranie množiny vhodných neurónových sietí genetický algoritmus. Ďalší nástroj, CNNE [55], inkrementálnym postupom zostavuje a trénuje neurónové siete pre ansámbel. CNNE určuje počet skrytých vrstiev, počet neurónov v nich a tiež počet epoch trénovania pre jednotlivé členy ansámblu. Funkcionalitu zostavovania ansámblov z viacerých modelov rozličných architektúr majú tiež niektoré nástroje automatického strojového učenia (ang. AutoML). Medzi takéto nástroje patria napríklad H2O.ai [1] alebo Microsoft Azure [2].

Do tejto kategórie metód zabezpečenia diverzity medzi členmi ansámblového modelu môžeme zaradiť tiež heterogénne ansámble. Autori v prácach [56, 57] testovali kombinácie neurónových sietí a rozhodovacích stromov. V práci [58] autori do ansámblu k neurónovým sieťam a rozhodovacím stromom pridali aj SVM. Tento ansámbel úspešne testovali na veľkom množstve rôznych datasetov.

[1] https://docs.h2o.ai/

[2] https://azure.microsoft.com/en-us/

Úpravy trénovacej množiny

Populárnym postupom pre upravovanie trénovacej množiny je jej rozdelenie na niekoľko disjunktných, alebo prekrývajúcich sa častí a trénovanie každého klasifikátora na inej časti. Trénovacia množina spravidla pozostáva z N trénovacích vzoriek - usporiadaných dvojíc $(\boldsymbol{x}_i, y_i)$, pričom vektor prediktorov $\boldsymbol{x}_i$ obsahuje P príznakov. Rozdelenie trénovacej množiny je možné robiť dvoma spôsobmi a to na:

- podmnožiny trénovacích vzoriek,
- podmnožiny príznakov.

Rozdelenie na rôzne podmnožiny trénovacích vzoriek využívajú viaceré populárne ansámblové metódy ako bagging [59], boosting [60] alebo AdaBoost [61], ktoré budú bližšie spomenuté v ďalšej sekcii.

Aplikácie ale tiež našiel druhý spôsob, rozdelenie trénovacej množiny na podmnožiny príznakov. Podmnožiny príznakov môžu byť vytvorené skúmaním korelácie medzi jednotlivými príznakmi a klasifikovanými triedami [62, 63]. Pri použití takéhoto postupu sú jednotlivé členy ansámblu špecializované na rôzne triedy alebo množiny tried. Ďalší úspešný prístup vytvára podmnožiny príznakov na základe vzájomnej informácie [64]. Autori najprv vytvoria skupiny príznakov s vysokou vzájomnou informáciou a potom každému klasifikátoru dajú k dispozícii z každej takejto skupiny aspoň jeden príznak.

5.2.3 Modifikácie spôsobu prehľadávania množiny hypotéz

Keď je množina dostupných hypotéz pevne určená použitou architektúrou modelu a trénovacou množinou, stále môžeme ovplyvniť trajektóriu, ktorou pri trénovaní trénovací algoritmus prechádza [49].

Pri neurónových sieťach môžeme trajektóriu prehľadávania meniť modifikovaním hyperparametra rýchlosť učenia (ang. learning rate) v priebehu trénovania siete. Rýchlosť učenia určuje veľkosť krokov v priestore váh siete, ktoré sa počas trénovania siete vykonávajú. Vyššia rýchlosť učenia umožňuje robiť väčšie kroky a rýchlejšie preskúmať väčšiu oblasť v priestore váh siete. Nižšia rýchlosť učenia umožňuje lepšie využiť lokálne vlastnosti priestoru váh siete a skonvergovať do lokálneho extrému. V priebehu trénovania neurónovej siete sa rýchlosť učenia zvyčajne postupne znižuje.

Nedávny výskum [65, 66] sa zaoberal možnosťou získať niekoľko inštancií neurónovej siete počas jedného vykonania trénovacieho algoritmu. Autorom sa podarilo získať dostatočne diverzifikované inštancie pomocou periodického znižovania a zvyšovania rýchlosti učenia. V priebehu zvýšenej rýchlosti učenia sa prehľadávanie presúvalo medzi rôznymi oblasťami priestoru váh siete a v priebehu znižovania rýchlosti učenia model konvergoval k lokálnemu extrému. Pri dosiahnutí najnižšej rýchlosti učenia boli váhy siete uložené a bol tak získaný nový člen ansámblu. Tento prístup má dôležitú výhodu vo forme nízkeho času potrebného na natrénovanie všetkých členov ansámblového modelu. Natrénovanie celého ansámblu je s použitím tejto metódy dosiahnuté za podobný čas, ako štandardne trvá natrénovanie jednej neurónovej siete.

5.3 Kombinačné metódy

Kombinačné metódy kombinujú výstupy natrénovaných členov ansámblu a produkujú tak finálnu predikciu ansámblového modelu. Kombinačné metódy môžeme rozdeliť do dvoch skupín na:

- metódy bez trénovania,
- metódy s trénovaním.

Metódy bez trénovania sú jednoduchšie, neumožňujú ale zohľadniť vlastnosti jednotlivých členov ansámblu. Učiace sa metódy sú zvyčajne úspešnejšie keď majú jednotlivé členy ansámblu rozdielnu presnosť, alebo keď spracúvajú rozličné množiny príznakov [67]. V ďalších sekciách sa venujeme niekoľkým populárnym klasifikačným kombinačným metódam.

5.3.1 Metódy bez trénovania

V tejto podsekcii preberieme niekoľko základných kombinačných metód bez trénovania. Napriek ich jednoduchosti sa tieto metódy často využívajú a môžu byť tiež použité ako referencia pri testovaní zložitejších metód.

Väčšinové hlasovanie

Pri väčšinovom hlasovaní priradí ansámbel pozorovanie do tej triedy, pre ktorú hlasuje väčšina členov ansámblu. Matematicky môže byť klasifikácia pozorovania $\boldsymbol{x}$ pri väčšinovom hlasovaní vyjadrená ako

$$Class(\boldsymbol{x}) = \underset{k \in K}{\operatorname{argmax}} \sum_{c \in C} d_k^c(\boldsymbol{x}), \tag{5.1}$$

kde K je množina klasifikovaných tried, C je množina členov ansámblu a $d_k^c(\boldsymbol{x})$ je rozhodnutie klasifikátora c o triede k. $d_k^c(\boldsymbol{x})$ nadobúda hodnotu 1, ak klasifikátor c zaradil pozorovanie $\boldsymbol{x}$ do triedy k, inak nadobúda hodnotu 0. Pokiaľ sú výstupom klasifikátora pravdepodobnosti príslušnosti do jednotlivých tried, klasifikátor hlasuje za najpravdepodobnejšiu triedu [67].

Súčet rozdelení pravdepodobnosti

Súčet rozdelení pravdepodobnosti je možné použiť v prípade, že členy ansámblu produkujú ako svoj výstup rozdelenia pravdepodobnosti [67]. Pokiaľ pri neurónovej sieti c použijeme ako výstupnú aktivačnú funkciu softmax (3.4), môžeme jej výstupy interpretovať ako pravdepodobnosti príslušnosti do jednotlivých tried: $f_k^c(\boldsymbol{x}) = \hat{P}_c(Y = k|\boldsymbol{x})$. Potom výslednú klasifikáciu určíme ako

$$Class(\boldsymbol{x}) = \underset{k \in K}{\operatorname{argmax}} \sum_{c \in C} f_k^c(\boldsymbol{x}). \tag{5.2}$$

Algebraické pravidlá

Pri výstupoch členov ansámblu vo forme rozdelení pravdepodobnosti môžeme použiť rôzne algebraické kombinačné metódy [68]. Príkladom takýchto metód sú **metóda maxima**:

$$Class(\boldsymbol{x}) = \underset{k \in K}{\operatorname{argmax}}\{\max_{c \in C}\{f_k^c(\boldsymbol{x})\}\}, \tag{5.3}$$

metóda minima:

$$Class(\boldsymbol{x}) = \underset{k \in K}{\operatorname{argmax}}\{\min_{c \in C}\{f_k^c(\boldsymbol{x})\}\}, \tag{5.4}$$

alebo **mediánová metóda**:

$$Class(\boldsymbol{x}) = \underset{k \in K}{\operatorname{argmax}}\{\operatorname{med}_{c \in C}\{f_k^c(\boldsymbol{x})\}\}. \tag{5.5}$$

Všetky spomínané metódy môžu byť rozšírené priradením váh jednotlivým klasifikátorom. Tieto váhy môžu byť určené rozličnými spôsobmi. Jednoduchým spôsobom je priradenie váh proporcionálnych presnostiam jednotlivých členov ansámblu. Iným prístupom je použitie váh nepriamo úmerných entropii výstupného vektora klasifikátora [67]. Pokiaľ chceme umožniť lepšie prispôsobenie váh vlastnostiam kombinovaných klasifikátorov, môžeme použiť unikátnu váhu pre každú triedu každého klasifikátora. Na určenie hodnôt takéhoto typu váh môže byť použitá lineárna regresia [69]. Pri použití niektorých typov váh môžu byť tieto metódy považované za kombinačné metódy s trénovaním.

5.3.2 Metódy s trénovaním

Metódy s trénovaním sú zväčša komplikovanejšie ako metódy bez trénovania, poskytujú ale lepšie možnosti prispôsobenia sa vlastnostiam kombinovaných klasifikátorov. Tieto metódy často zahŕňajú použitie viacerých úrovní modelov štatistického učenia. V nasledujúcich podsekciách sa venujeme niekoľkým základným učiacim sa kombinačným metódam.

Stohovanie

Stohovanie (ang. stacking) používa dvojúrovňovú klasifikáciu. Prvá úroveň pozostáva z klasických členov ansámblu a druhá úroveň je meta-klasifikátor. Meta-klasifikátor je natrénovaný na meta-datasete, ktorý je vytvorený z výstupov členov ansámblu na prvej úrovni. Označenia prvkov v tomto datasete ostávajú rovnaké ako v pôvodnom datasete, ide teda o označenia tried. Pri klasifikovaní nového pozorovania je toto pozorovanie najprv spracované klasifikátormi na prvej úrovni a následne sú ich výstupy spracované meta-klasifikátorom. Výstup meta-klasifikátora je finálnym výstupom ansámblu. Pri trénovaní sa odporúča rozdeliť trénovaciu množinu na dve časti. Na prvej časti sa natrénujú klasifikátory na prvej úrovni a z druhej časti sa vytvorí meta-dataset pre trénovanie meta-klasifikátora [67].

Autori v práci [70] skúmali efektivitu stohovania na viacerých ansámblových modeloch a viacerých datasetoch a zistili, že je zhruba ekvivalentná výberu najlepšieho členu ansámblu pomocou krížovej validácie, ale nie lepšia.

Triedenie

Triedenie (ang. grading) je ďalšia metóda, ktorá sa dá chápať ako dvojúrovňová. V prípade tejto kombinačnej metódy sa tiež používajú meta-klasifikátory. V prípade triedenia je ku každému klasifikátoru priradený osobitný meta-klasifikátor. Meta-datasety na trénovanie týchto meta-klasifikátorov sú zostavené z prediktorov pôvodného datasetu. Závislá premenná je binárna a vyjadruje, či príslušný klasifikátor správne klasifikoval dané pozorovanie. Meta-klasifikátory teda pre každé pozorovanie predikujú, či im priradený klasifikátor správne určí triedu daného pozorovania. Do výslednej kombinácie potom vstupujú výstupy len tých klasifikátorov, ktoré im príslušiaci meta-klasifikátor predikuje ako správne [71].

Bránovanie

Bránovanie (ang. gating) je podobná metóda ako triedenie s tým rozdielom, že využíva len jeden meta-klasifikátor, nazývaný bránovací (ang. gating) klasifikátor. Do bránovacieho klasifikátora vstupujú rovnaké dáta ako do klasických členov ansámblu, výstupom sú ale pravdepodobnosti toho, že jednotlivé členy ansámblu vyprodukujú pre dané vstupy správnu predikciu. Na základe týchto pravdepodobností je náhodne vybraný jeden z členov ansámblu, ktorého predikcia predstavuje výslednú predikciu ansámblu. Bránovací klasifikátor sa trénuje súčasne s ostatnými členmi ansámblu, pričom sa vždy upravujú len váhy toho členu ansámblu, ktorý bol vybraný a váhy bránovacieho klasifikátora [72].

5.4 Populárne ansámblové metódy

Existuje niekoľko zaužívaných ansámblových klasifikačných metód, ktoré špecifikujú typ použitých klasifikátorov, spôsob ich trénovania, metódu zabezpečenia diverzity a tiež kombinačnú metódu. Rôzne metódy umožňujú vykonávať rôznu mieru zmien v týchto základných stavebných blokoch. Tieto metódy môžeme rozdeliť do dvoch kategórií: závislé a nezávislé. Závislé metódy využívajú pri trénovaní členov ansámblu informácie, ktoré im poskytujú výstupy už natrénovaných členov. Nezávislé metódy trénujú všetky členy nezávisle. Členy závislých metód teda musia byť natrénované sériovo, pričom členy nezávislých metód môžu byť trénované paralelne.

5.4.1 Nezávislé metódy

Bagging

Bagging, skratka pre bootstrap aggregating, je použiteľný ako pre regresiu, tak aj pre klasifikáciu. Táto metóda dobre funguje pre nestabilné učiace sa algoritmy [59]. Nestabilné učiace sa algoritmy sú také, pri ktorých malá zmena v trénovacej množine môže spôsobiť veľkú zmenu vo výslednom klasifikačnom modeli. Výskum zaoberajúci sa nestabilitou učiacich sa algoritmov [73] ukázal, že klasifikačné a regresné stromy a neurónové siete sú nestabilné učiace sa algoritmy na rozdiel od klasifikátora k-najbližších susedov, ktorý je stabilný.

Zabezpečenie diverzity klasifikátorov pri bagging-u funguje pomocou použitia rôznych trénovacích množín. Jednotlivé členy ansámblu sú trénované na rôznych podmnožinách trénovacej množiny. Tieto podmnožiny sú vytvorené náhodným výberom s opakovaním spomedzi prvkov pôvodnej trénovacej množiny. Takýto postup tvorenia podmnožín sa nazýva bootstrapping. Autori v [59] odporúčajú použiť podmnožiny rovnakej veľkosti ako je veľkosť pôvodnej trénovacej množiny. Takýto prístup má za následok, že niektoré prvky trénovacej množiny sa v niektorých podmnožinách vyskytujú viac krát, pričom iné v nich absentujú.

Odporúčanou kombinačnou metódou pre použitie bagging-u na klasifikáciu je väčšinové hlasovanie [59].

5.4.2 Závislé metódy

Boosting

Metóda **boosting** bola vyvinutá za účelom transformácie slabých učiacich sa algoritmov na silný učiaci sa algoritmus pri úlohe dvojtriednej klasifikácie. Slabý učiaci sa algoritmus je taký algoritmus, ktorý dokáže s vysokou pravdepodobnosťou vyprodukovať klasifikátor, ktorý je aspoň o trochu lepší ako náhodné hádanie. Silný učiaci sa algoritmus dokáže s vysokou pravdepodobnosťou vyprodukovať klasifikátor, ktorý má ľubovoľne malú chybu. Pomocou boosting-u bola za určitých podmienok dokázaná ekvivalencia medzi problémami, pre ktoré existuje slabý učiaci sa algoritmus a tými, pre ktoré existuje silný učiaci sa algoritmus [60].

Boosting môže byť použitý na kombinovanie ľubovoľných slabých učiacich sa algo-

ritmov. Nech T je trénovacia množina a L je zvolený slabý učiaci sa algoritmus. Pri vykonávaní algoritmu L sú trénovacie dáta náhodne vyberané podľa určitého rozdelenia pravdepodobnosti. Priebeh boosting-u pozostáva z nasledujúcich krokov:

1. Vykonanie zvoleného učiaceho sa algoritmu L s výberom trénovacích dát podľa rovnomerného rozdelenia D_1 nad množinou T - vznikne hypotéza h_1.

2. Určenie rozdelenia D_2 nad T tak, aby platilo, že prvok z tohto rozdelenia má približne rovnakú šancu byť hypotézou h_1 klasifikovaný správne ako nesprávne.

3. Vykonanie algoritmu L s výberom dát z D_2 - vznikne hypotéza h_2.

4. Určenie rozdelenia D_3 nad T tak, aby platilo, že hypotézy h_1 a h_2 klasifikujú prvky z tohto rozdelenia odlišne.

5. Vykonanie algoritmu L s výberom dát z D_3 - vznikne hypotéza h_3.

Výstup ansámblu je tvorený väčšinovým hlasovaním medzi hypotézami h_1, h_2 a h_3. Chyba takéhoto ansámblového modelu je ohraničená zhora výrazom

$$3\alpha^2 - 2\alpha^3 \tag{5.6}$$

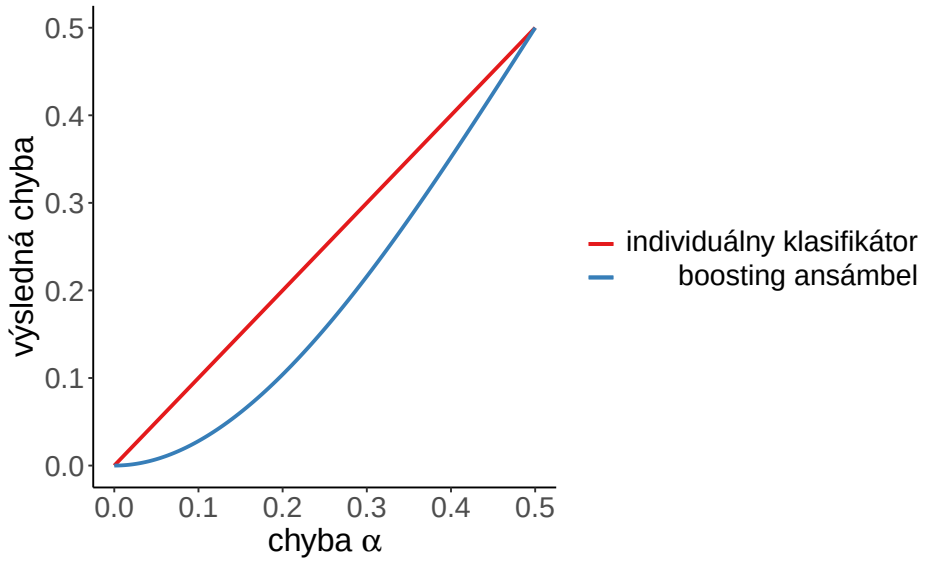
kde α je horná hranica chyby slabého učiaceho sa algoritmu L [60]. Porovnanie grafu tohto výrazu s grafom identity je zobrazené na obrázku 5.2.

Silný učiaci sa algoritmus je možné získať pomocou rekurzívneho aplikovania uvedeného postupu. Technické detaily a dôkazy je možné nájsť v [60].

AdaBoost

AdaBoost [61] je ďalším prístupom k transformácii slabých učiacich sa algoritmov na silný učiaci sa algoritmus. Je založený na podobných princípoch ako boosting. Pre AdaBoost boli vyvinuté aj rozšírenia, ktoré umožňujú riešiť viactriednu klasifikáciu a regresiu. Dôležitou výhodou AdaBoostu oproti boostingu je, že horná hranica chyby AdaBoost ansámblu je ohraničená výrazom, ktorý je závislý od presnosti všetkých členov ansámblu. Pri boostingu je chyba závislá len od presnosti najhoršieho členu (5.6).

Pred začiatkom vykonávania algoritmu je každému prvku trénovacej množiny priradená rovnaká váha. Algoritmus pracuje v cykle a v každej iterácii je natrénovaná



Obrázok 5.2: Porovnanie horného ohraničenia chyby boosting ansámblu s chybou α individuálneho klasifikátora.

nová hypotéza slabého učiaceho sa algoritmu. Trénovacie dáta pre trénovanie hypotézy sú náhodne vyberané z trénovacej množiny s pravdepodobnosťami proporcionálnymi váham priradeným jednotlivým prvkom. Po natrénovaní je každá hypotéza otestovaná na celej trénovacej množine. Na základe tohto testovania sú upravené váhy prvkov trénovacej množiny. Pokiaľ nová hypotéza prvok klasifikuje správne, je jeho váha znížená, v opačnom prípade je zvýšená. V dôsledku týchto úprav majú nesprávne klasifikované prvky trénovacej množiny väčšiu pravdepodobnosť byť vybraté pre trénovanie ďalšej hypotézy. Tento postup má za následok, že neskoršie hypotézy sú trénované na dátach, ktoré sa pre predchádzajúce hypotézy ukázali ako problematické. Výsledná klasifikácia ansámblu je vytvorená ako vážený súčet rozdelení pravdepodobností s váhami proporcionálnymi presnostiam členov ansámblu [61].

5.5 Párové ansámblové modely

Párové ansámblové modely sú viactriedne klasifikačné modely, ktoré sú zostavené z pravdepodobnostných dvojtriednych klasifikátorov. Na rozdiel od štandardného postupu, kde viactriedne klasifikačné ansámble sú tvorené z viactriednych klasifikátorov [68], pri párových ansámbloch je výsledná viactriedna klasifikácia tvorená kombinovaním výstupov dvojtriednych klasifikátorov. Každý dvojtriedny klasifikátor rozlišuje

medzi dvojicou tried viactriednej úlohy a ich výstupy sú kombinované pomocou špeciálnych párových zväzovacích metód [11, 12, 74, 75, 76, 77, 78]. Viactriednou klasifikáciou s využitím výstupov dvojtriednych klasifikátorov sa zaoberá aj metóda nazývaná error-correcting output codes [79], pri tejto metóde ale dvojtriedne klasifikátory rozlišujú medzi podmnožinami tried.

Prvé aplikácie párových ansámblových modelov môžeme nájsť pri rozširovaní metódy podporných vektorov (SVM) [3] na viactriednu klasifikáciu. Prvým predpokladom bolo získanie pravdepodobnostných predikcií z SVM. To bolo dosiahnuté modelovaním pravdepodobnosti logistickou funkciou na základe vzdialenosti od oddeľujúcej nadroviny [10]. Následne boli natrénované SVM pre každú dvojicu tried a ich výstupy boli skombinované pomocou párovej zväzovacej metódy [12]. Popísaný SVM ansámbel je súčasťou populárnej knižnice LIBSVM [13].

5.5.1 Párové zväzovacie metódy

Párové zväzovacie (ang. pairwise coupling) metódy sú špeciálnym prípadom kombinačných metód bez trénovania. Existuje veľké množstvo rozličných párových zväzovacích metód [11, 12, 74, 75, 76, 77, 78]. Bližšie sa budeme venovať dvom metódam navrhnutým autormi Wu-Lin-Weng v práci [12], Bayesovsky kovariantnej metóde [78] a metóde od autorov Šuch, Benuš a Tinajová [77]. Metódy od autorov Wu-Lin-Weng boli v čase vytvárania knižnice LIBSVM [13] experimentálne vyhodnotené ako najvhodnejšie na kombinovanie SVM klasifikátorov. Metóda z práce [78] spĺňa unikátnu teoretickú vlastnosť Bayesovskej kovariancie.

Tieto metódy predpokladajú, že pre každé pozorovanie $\boldsymbol{x}$ a označenie triedy y máme k dispozícii výstupy dvojtriednych klasifikátorov r_{ij}, ktoré aproximujú pravdepodobnosti $\mu_{ij} = P(Y = i \mid Y = j \text{ alebo } Y = i, \boldsymbol{x})$. Cieľom párových kombinačných metód je s pomocou všetkých r_{ij} odhadnúť $\boldsymbol{p} = (p_1, p_2, \ldots, p_K)^T$ kde $p_i = P(Y = i \mid \boldsymbol{x})$.

5.5.2 Metódy od autorov Wu-Lin-Weng

Autori v práci [12] navrhujú dve nové párové zväzovacie metódy. Prvá z nich je inšpirovaná metódou [11] a druhá je odvodená matematickou úpravou z prvej.

Prvá metóda

Prvá metóda je vo výsledkoch experimentov označovaná ako **m1**. Táto metóda navrhuje pre nájdenie $\boldsymbol{p}$ riešiť systém rovníc

$$\begin{gathered} p_i = \sum_{j:j\neq i} \left(\frac{p_i + p_j}{K-1}\right) r_{ij}, \quad \text{pre } i = 1, 2, \ldots, K \\ \text{za podmienok } \sum_{i=1}^{K} p_i = 1, \quad p_i \geq 0, \text{ pre } i = 1, 2, \ldots, K. \end{gathered} \tag{5.7}$$

Ako je v práci [12] ukázané, riešenie tohto systému je ekvivalentné riešeniu systému s vynechaním podmienok nezápornosti p_i. Takýto systém sa dá maticovo zapísať ako:

$$\boldsymbol{Qp} = \boldsymbol{p}, \quad \sum_{i=1}^{K} p_i = 1, \quad \text{kde } Q_{ij} = \begin{cases} r_{ij}/(K-1) & \text{pre } i \neq j, \\ \sum_{s:s\neq i} r_{is}/(K-1) & \text{pre } i = j. \end{cases} \tag{5.8}$$

Keďže $\sum_{i=1}^{K} Q_{ij} = 1$ pre $j = 1, 2, \ldots, K$, každá z rovníc v $\boldsymbol{Qp} = \boldsymbol{p}$ je lineárnou kombináciou zvyšných rovníc. Môžeme teda jednu z týchto rovníc vynechať a riešiť sústavu K lineárnych rovníc o K neznámych.

S použitím predpokladu $r_{ij} + r_{ji} = 1$ je možné model (5.7) interpretovať ako konvexnú optimalizačnú úlohu:

$$\begin{gathered} \underset{\boldsymbol{p}}{\operatorname{argmin}} \quad \sum_{i=1}^{K} \left(\sum_{j:j\neq i} r_{ji} p_i - \sum_{j:j\neq i} r_{ij} p_j \right)^2 \\ \text{za podmienok} \sum_{i=1}^{K} p_i = 1, \quad p_i \geq 0, \text{ pre } i = 1, 2, \ldots, K. \end{gathered} \tag{5.9}$$

Z tejto interpretácie prvej metódy je odvodená druhá metóda.

Druhá metóda

Druhá metóda je vo výsledkoch experimentov označovaná ako **m2**. Táto metóda je založená na riešení optimalizačnej úlohy podobnej ako (5.9) s mierne upravenou účelovou funkciou. Model upravenej úlohy je nasledovný:

$$\begin{gathered} \underset{\boldsymbol{p}}{\operatorname{argmin}} \quad \sum_{i=1}^{K} \sum_{j:j\neq i} (r_{ji} p_i - r_{ij} p_j)^2 \\ \text{za podmienok} \sum_{i=1}^{K} p_i = 1, \quad p_i \geq 0, \text{ pre } i = 1, 2, \ldots, K. \end{gathered} \tag{5.10}$$

V tomto modeli je opäť možné vynechať podmienky nezápornosti p_i a účelovú funkciu modelu (5.10) je možné maticovo zapísať ako:

$$\underset{\boldsymbol{p}}{\operatorname{argmin}} \frac{1}{2}\boldsymbol{p}^T\boldsymbol{Q}\boldsymbol{p}, \quad \text{kde } Q_{ij} = \begin{cases} \sum_{s:s\neq i} r_{si}^2 & \text{pre } i = j, \\ -r_{ji}r_{ij} & \text{pre } i \neq j. \end{cases} \tag{5.11}$$

Riešenie tohto modelu môžeme získať riešením sústavy lineárnych rovníc:

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{e} \\ \boldsymbol{e}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix}, \tag{5.12}$$

kde $\boldsymbol{e}$ je jednotkový vektor rozmeru $k \times 1$, $\boldsymbol{0}$ je nulový vektor rozmeru $k \times 1$ a b je Lagrangeov multiplikátor podmienky rovnosti z modelu (5.11). Alternatívnym spôsobom riešenia modelu (5.11) je iteratívny algoritmus odvodený v [12].

5.5.3 Bayesovsky kovariantná metóda

Bayesovsky kovariantná (BC) metóda je vo výsledkoch experimentov označovaná ako **bc**. Túto metódu navrhli autori v práci [78]. Pre vysvetlenie základnej vlastnosti tejto metódy - Bayesovskej kovariantnosti potrebujeme zadefinovať niekoľko vzťahov. Výstupy párových klasifikátorov r_{ij} môžeme s pomocou Bayesovej vety zapísať ako:

$$r_{ij} \approx \frac{P(\boldsymbol{x} \mid Y = i)P(Y = i)}{P(\boldsymbol{x}, Y = i \text{ alebo } Y = j)} = \frac{P(\boldsymbol{x} \mid Y = i)P(Y = i)}{P(\boldsymbol{x} \mid Y = i)P(Y = i) + P(\boldsymbol{x} \mid Y = j)P(Y = j)}. \tag{5.13}$$

BC metóda požaduje, aby platilo $r_{ij} + r_{ji} = 1$ a tiež $r_{ij} > 0$ pre všetky $i \neq j$. Matice $\boldsymbol{R} = (r_{ij})$, ktoré tieto vlastnosti spĺňajú, autori označujú ako prípustné. Hľadané výsledné pravdepodobnosti môžeme zapísať ako:

$$P(Y = i \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid Y = i)P(Y = i)}{\sum_{j=1}^{K} P(\boldsymbol{x} \mid Y = j)P(Y = j)}. \tag{5.14}$$

Ak označíme $l_i = P(\boldsymbol{x} \mid Y = i)P(Y = i)$ z (5.13) získame systém Bradley-Terry-ho rovníc:

$$r_{ij} = \frac{l_i}{l_i + l_j}, \tag{5.15}$$

ktorého riešením môžeme získať hľadané rozdelenie pravdepodobnosti:

$$p_i = P(Y = i \mid \boldsymbol{x}) = \frac{l_i}{\sum_j l_j}. \tag{5.16}$$

Vlastnosť Bayesovskej kovariancie sa týka zmien hodnôt apriórnych pravdepodobností $\pi_i = P(Y = i)$. Od týchto apriórnych pravdepodobností priamo závisí systém Bradley-Terry-ho rovníc (5.15). Pokiaľ sa apriórne pravdepodobnosti zmenia na $\pi'_i = q_i\pi_i$, podľa (5.15) by sa párové klasifikácie r_{ij} mali zmeniť na r'_{ij}, ktoré spĺňajú:

$$\frac{1}{r'_{ij}} - 1 = \frac{q_j}{q_i}\left(\frac{1}{r_{ij}} - 1\right). \tag{5.17}$$

Ak označíme BC metódu ako funkciu B, matice párových klasifikácií, ktoré do nej vstupujú ako $\boldsymbol{R} = (r_{ij})$, $\boldsymbol{R'} = (r'_{ij})$ a výstupy z tejto metódy ako $\boldsymbol{p} = B(\boldsymbol{R})$, $\boldsymbol{p'} = B(\boldsymbol{R'})$, potom z (5.14) by malo platiť:

$$(p'_1, p'_2, \dots) \propto (q_1p_1, q_2p_2, \dots). \tag{5.18}$$

Autori v [78] považujú kombinačnú metódu B za Bayesovsky kovariantnú, pokiaľ spĺňa túto vlastnosť pre ľubovoľnú prípustnú maticu $\boldsymbol{R}$ a ľubovoľný kladný váhový vektor $\boldsymbol{q} = (q_1, q_2, \dots, q_K)$. Pre BC metódu teda platí, že nezáleží na tom, či úpravy apriórnych pravdepodobností vykonáme pred, alebo po aplikácii tejto metódy.

V práci [78] autori navrhli metódu, ktorá spĺňa uvedené vlastnosti. Aplikácia metódy je výpočtovo nenáročná a pozostáva z použitia elementárnych matematických funkcií a maticových operácií. Pre použitie BC metódy potrebujeme zadefinovať funkciu $h = (h_1, h_2)$, ktorá je bijektívnym zobrazením medzi množinou $\{1, 2, \dots, K(K-1)/2\}$ a množinou usporiadaných dvojíc $\{(i, j) \mid 1 \leq i < j \leq K\}$. Potom na základe tejto funkcie zadefinujeme maticu $\boldsymbol{M} = (m_{ij})$ s rozmermi $K(K-1)/2 \times (K-1)$ ako:

$$m_{ij} = \begin{cases} -1 & \text{ak } h_1(i) = j, \\ 1 & \text{ak } h_2(i) = j, \\ 0 & \text{inak.} \end{cases} \tag{5.19}$$

Ďalej potrebujeme vektor

$$\boldsymbol{s} = (s_{h(1)}, s_{h(2)}, \dots, s_{h(K(K-1)/2)})^T, \quad \text{kde } s_{ij} = \log\left(\frac{1}{r_{ij}} - 1\right). \tag{5.20}$$

S pomocou $\boldsymbol{M}$ a $\boldsymbol{s}$ spočítame vektor $\boldsymbol{u}$ ako:

$$\boldsymbol{u} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T\boldsymbol{s}. \tag{5.21}$$

Tento vektor pozostáva z prvkov $\boldsymbol{u} = (u_1, u_2, \dots, u_{K-1})^T$ a výstup BC metódy je potom daný ako:

$$p_i = \frac{\exp(u_{i-1})}{\sum_{i=1}^{K}\exp(u_{i-1})}, \quad \text{kde } u_0 = 0. \tag{5.22}$$

Zo špeciálnej podoby súčinu $\boldsymbol{M}^T\boldsymbol{M}$ plynie, že jeho inverzia má podobu:

$$(\boldsymbol{M}^T\boldsymbol{M})^{-1} = \frac{1}{K}(\boldsymbol{I} + \boldsymbol{e}\boldsymbol{e}^T), \tag{5.23}$$

kde $\boldsymbol{I}$ je matica veľkosti $(K-1)\times(K-1)$ s jednotkami na diagonále a $\boldsymbol{e}$ je jednotkový vektor rozmeru $(K-1)\times 1$ [78].

5.5.4 Metóda od autorov Šuch-Benuš-Tinajová

Metóda od autorov Šuch, Benuš a Tinajová (SBT) je vo výsledkoch experimentov označovaná ako **sbt**. Túto metódu navrhli autori v práci [77]. Metóda vyžaduje, aby pre párové pravdepodobnosti platilo $0 < r_{ij} < 1$. Metóda je založená na úvahe, že ak pozorovanie patrí do triedy $m \in \{1, 2, \ldots, K\}$, potom sa relevantné informácie k jeho klasifikácii nachádzajú len v párových klasifikáciách r_{mj} (a $r_{jm} = 1 - r_{mj}$) pre $j = 1, 2, \ldots, K, j \neq m$. Pre týchto $K-1$ hodnôt je možné vyriešiť systém Bradley-Terry-ho rovníc

$$r_{ij} = \frac{p_i}{p_i + p_j} \tag{5.24}$$

exaktne nasledovným spôsobom. Z 5.24 máme

$$\sum_{j\neq m}\frac{1}{r_{mj}} = \sum_{j\neq m}\frac{p_m + p_j}{p_m} = (K-1) + \frac{1-p_m}{p_m}, \tag{5.25}$$

z čoho potom môžeme odvodiť odhad $p_m^{(m)}$ pravdepodobnosti p_m ako

$$p_m^{(m)} = \left(\sum_{j\neq m}\frac{1}{r_{mj}} - (K-2)\right)^{-1}, \tag{5.26}$$

kde horný index (m) zdôrazňuje, že tento odhad bol spočítaný len s použitím párových pravdepodobností r_{mj} zahŕňajúcich triedu m. Odhady pravdepodobností pre zvyšné triedy spočítame ako

$$p_j^{(m)} = p_m^{(m)}\left(\frac{1}{r_{mj}} - 1\right). \tag{5.27}$$

Tento výpočet môžeme zopakovať pre všetky $m = 1, 2, \ldots, K$. Vzniknuté odhady pravdepodobností budú vo všeobecnom prípade konfliktné, teda $p_i^{(m)} \neq p_i^{(n)}$ pre $m \neq n$. Autori párovej zväzovacej metódy sa pri riešení tejto nekonzistencie inšpirujú zákonom o úplnej pravdepodobnosti $P(A) = \sum_i P(A|B_i)P(B_i)$, kde javy B_i tvoria úplný systém javov. Odhad hľadaného rozdelenia pravdepodobnosti $\boldsymbol{p}$ potom získame podľa

$$p_j = \sum_{m=1}^{K} p_j^{(m)} p_m \quad \text{pre } j = 1, 2, \ldots, K, \tag{5.28}$$

kde p_j je j-ty prvok vektora $\boldsymbol{p}$. Zavedením podmienky

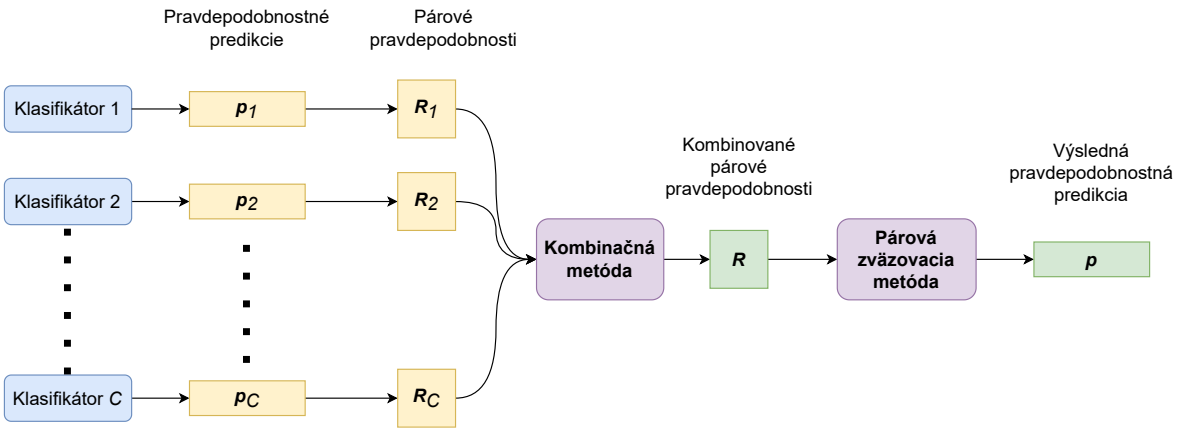
$$\sum_{i=1}^{K} p_i = 1 \tag{5.29}$$

potom získame unikátne riešenie.

Kapitola 6

Vážený lineárny ansámbel

V tejto kapitole sa venujeme návrhu novej ansámblovej metódy pomenovanej Vážený lineárny ansámbel (WLE). WLE metóda rieši úlohu viactriednej klasifikácie kombinovaním niekoľkých viactriednych klasifikátorov. Diagram činnosti metódy je zobrazený na obrázku 6.1. Metóda využíva postup **binarizácie**, pri ktorom rozdelí riešený problém na niekoľko dvojtriednych podproblémov. V kontexte týchto dvojtriednych podproblémov skombinuje predikcie členov ansámblu pomocou **kombinačnej metódy**. Získané dvojtriedne predikcie metóda skombinuje použitím **párovej zväzovacej metódy** a získa tak výslednú viactriednu pravdepodobnostnú klasifikáciu.



Obrázok 6.1: Diagram činnosti WLE metódy.

Postup získania dvojtriednych predikcií zo vstupných viactriednych predikcií je založený na predpoklade platnosti axiómu irelevantných alternatív. Ak vektor $\boldsymbol{p} = (p_1, p_2, \ldots, p_K)$ je výstup viactriedneho pravdepodobnostného klasifikátora a p_i predstavuje pravdepodobnosť príslušnosti klasifikovaného objektu do triedy i, potom výstup

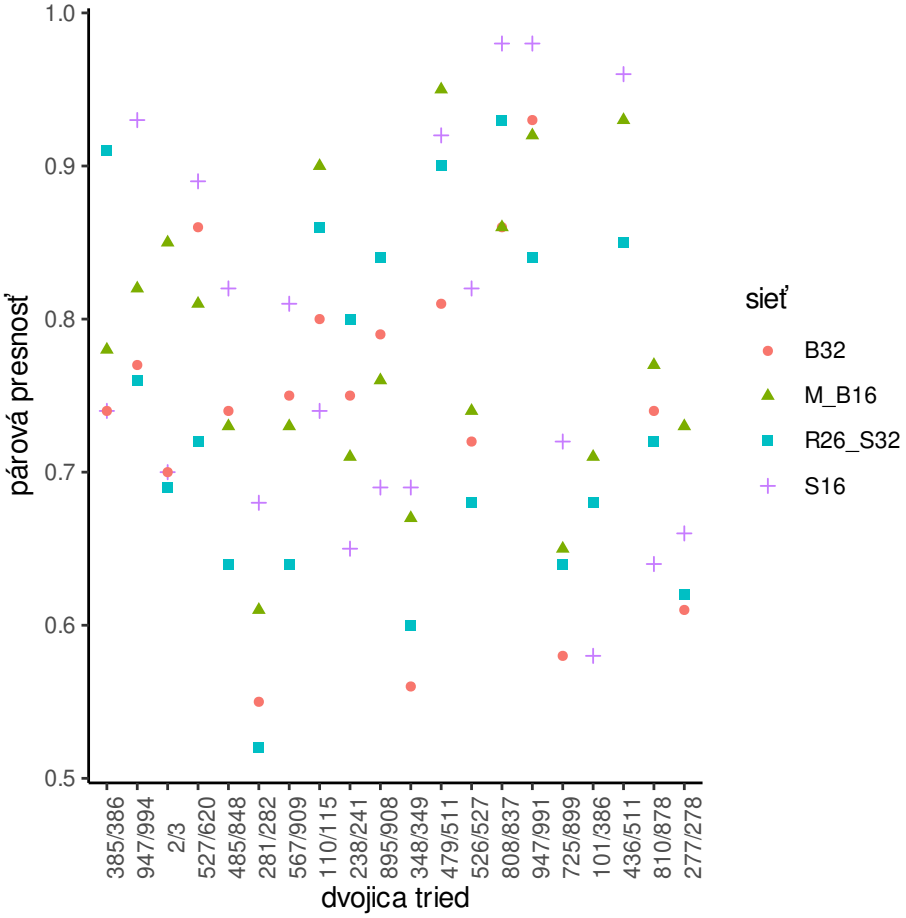
dvojtriedneho klasifikátora rozlišujúceho medzi triedami i a j môžeme vyjadriť ako

$$\left(\frac{p_i}{p_i + p_j}, \frac{p_j}{p_i + p_j}\right). \tag{6.1}$$

Pre binarizáciu predikcií viactriednych klasifikátorov je potrebné, aby pre takéto viactriedne klasifikátory platil axióm o nezávislosti irelevantných alternatív (IIA). Formulácia tohto axiómu hovorí o tom, že ak alternatívy x, y z množiny T majú pravdepodobnosti výberu $P_T(x), P_T(y)$ a obe alternatívy x, y patria tiež do množiny S, ktorá je podmnožinou T, potom pre pravdepodobnosti výberu alternatív x, y z množiny S, označené ako $P_S(x), P_S(y)$, platí $\frac{P_T(x)}{P_T(y)} = \frac{P_S(x)}{P_S(y)}$ [80].

Motivácia pre kombinovanie viactriednych klasifikátorov s využitím transformácie na dvojtriedne klasifikátory spočíva v tom, že rôzne klasifikátory môžu mať rozličné schopnosti rozlišovať medzi jednotlivými dvojicami tried. Takúto situáciu vidíme na obrázku 6.2. Z obrázku môžeme vidieť veľké rozdiely medzi párovými presnosťami jednotlivých sietí. Tiež môžeme pozorovať, že poradie sietí podľa párových presností sa pre rôzne dvojice tried líši.

Párové predikcie, ktoré získame pomocou binarizácie je potrebné zlúčiť do jednej matice párových predikcií, ktorá slúži ako vstup do párových zväzovacích metód. Za týmto účelom sme navrhli niekoľko kombinačných metód, ktoré predstavíme v nasledujúcich sekciách.



Obrázok 6.2: Párové presnosti štyroch neurónových sietí na datasete ImageNet1k. Presnosti sú zobrazené pre 20 párov tried s najvyššími rozptylmi v párových presnostiach skúmaných sietí. Označenia tried reprezentujú poradie (indexované od nuly) zodpovedajúceho priečinku medzi abecedne zoradenými priečinkami tried Imagenet1k datasetu. Informácie o použitých sieťach sú v sekcii 7.2.

6.1 Kombinačné metódy

Ako vstup do kombinačných metód používame výstupy poslednej vrstvy neurónových sietí pred aplikovaním funkcie softmax, nazývané logity. V kapitole 3 označujeme tieto hodnoty ako $\boldsymbol{l}$. Pri klasifikácii má posledná vrstva štandardne rovnaký počet výstupov, ako je počet tried. Logity kombinovaných klasifikátorov $1, 2, \ldots, C$ označujeme ako $\boldsymbol{l}^1, \boldsymbol{l}^2, \ldots, \boldsymbol{l}^C$, kde $\boldsymbol{l}^c = (l_1^c, l_2^c, \ldots, l_K^c)$ pre jednotlivé triedy $1, 2, \ldots, K$. Logity vyjadrujú mieru podpory siete pre danú triedu, pričom nespĺňajú vlastnosť nezápornosti, ani nemajú pevne stanovený súčet. Výstupom kombinačných metód je matica párových pravdepodobností $\boldsymbol{R}$, ktorá je vstupom pre párové zväzovacie metódy. Istým spôso-

bom sa tento krok nášho ansámblového modelu dá považovať za **stacking** ansámbel a kombinačné metódy za metaklasifikátor.

Kombinačné metódy, ktoré sme navrhli možno rozdeliť do dvoch hlavných kategórií:

- bezparametrické,
- parametrické.

Bezparametrické metódy nevyžadujú trénovanie a teda ani žiadne dáta navyše. Parametrické metódy používajú na kombinovanie klasifikátorov parametre, pre ktoré je potrebné nájsť vhodné hodnoty trénovaním. Metodológiou trénovania kombinačných metód sa zaoberáme v sekcii 7.3. V nasledujúcich podsekciách opíšeme jednotlivé metódy, ktoré sme implementovali a s ktorými sme vykonávali experimenty.

6.1.1 Bezparametrické metódy

Navrhli a implementovali sme dve bezparametrické metódy využívajúce aritmetický priemer. Prvá z nich je **prob_average** - *pravdepodobnostný priemer*. Táto metóda vytvára výslednú maticu $\boldsymbol{R} = (r_{ij})$ ako priemer matíc párových pravdepodobností jednotlivých kombinovaných klasifikátorov. Matematicky môžeme túto metódu vyjadriť ako

$$r_{ij} = \frac{1}{C} \sum_{c=1}^{C} \text{expit}(l_i^c - l_j^c), \tag{6.2}$$

kde funkcia $\text{expit}(x) = \frac{1}{1+e^{-x}}$. Matematicky ekvivalentný je postup

$$\begin{aligned} \boldsymbol{p}^c &= softmax(\boldsymbol{l}^c), \quad \text{pre } c = 1, 2, \ldots, C \\ r_{ij} &= \frac{1}{C} \sum_{c=1}^{C} \frac{p_i^c}{p_i^c + p_j^c}. \end{aligned}$$

Druhou implementovanou bezparametrickou metódou je kombinačná metóda **average** - *priemer*. Jej činnosť môžeme matematicky zapísať ako

$$r_{ij} = \text{expit}(\frac{1}{C} \sum_{c=1}^{C} l_i^c - l_j^c), \tag{6.3}$$

kde r_{ij} je prvok výstupnej matice $\boldsymbol{R}$. Na rozdiel od metódy *pravdepodobnostný priemer* aplikuje metóda *priemer* funkciu *expit* na aritmetický priemer rozdielov logitov pre dvojice tried.

6.1.2 Parametrické metódy

Parametrické metódy využívajú pri kombinovaní klasifikátorov trénovateľné parametre. Kombinácia je lineárna vzhľadom na parameter funkcie expit. Matematicky môžeme výpočet vyjadriť ako

$$r_{ij} = \text{expit}(\sum_{c=1}^{C} w_{ij}^c (l_i^c - l_j^c) + b_{ij}) \tag{6.4}$$

kde r_{ij} je prvok výstupnej matice $\boldsymbol{R}$, w_{ij}^c je koeficient prislúchajúci k dvojici tried i, j a klasifikátoru c a b_{ij} je bias prislúchajúci k dvojici tried i, j. Pre výpočet r_{ji} sa používajú rovnaké parametre w_{ij}^c a b_{ij} ako pre výpočet r_{ij}. Nech počet všetkých tried je K a počet kombinovaných klasifikátorov C, potom počet parametrov takéhoto kombinačného modelu je $K(K-1)(C+1)/2$. Pri niektorých kombinačných metódach používame pre všetky dvojice tried rovnaké parametre, počet parametrov modelu je potom $C+1$. Jednotlivé kombinačné metódy sa líšia v tom, akým spôsobom nastavujú hodnoty kombinačných koeficientov.

Teplotné škálovanie

Navrhli sme dve metódy, ktoré používajú rovnaké parametre pre všetky dvojice tried. Obe z nich vypočítavajú parametre pomocou kalibračnej metódy teplotné škálovanie (ang. temperature scaling) a nastavujú ich ako $w^c = \frac{1}{t^c}$, kde t^c je kalibračná teplota pre klasifikátor c. Obe metódy používajú nulový bias.

Prvá z nich, **cal_prob_average** - *kalibrovaný pravdepodobnostný priemer* aplikuje funkciu *expit* pred sčítaním, nespĺňa teda všeobecnú rovnicu 6.4. Kombinácia s použitím tejto metódy je realizovaná ako

$$r_{ij} = \frac{1}{C} \sum_{c=1}^{C} \text{expit}(\frac{l_i^c - l_j^c}{t^c}), \tag{6.5}$$

kde r_{ij} je prvok výstupnej matice $\boldsymbol{R}$. Táto metóda je matematicky ekvivalentná kalibrácii kombinovaných klasifikátorov, výpočtu matice párových pravdepodobností pre každý z nich a následnému spriemerovaniu týchto matíc.

Druhá metóda využívajúca kalibračné koeficienty, **cal_average** - *kalibrovaný priemer*, pracuje podľa rovnice 6.4.

Ostatné kombinačné metódy využívajú samostatné koeficienty pre jednotlivé dvojice tried, líšia sa ale v tom, akým spôsobom určujú hodnoty pre tieto koeficienty.

Lineárna diskriminačná analýza

Kombinačná metóda **lda** využíva na určenie hodnôt koeficientov klasifikačný algoritmus **lineárna diskriminačná analýza** (lda). Metóda **lda** rieši pre každú dvojicu tried $\{i,j\}$ dvojtriednu klasifikačnú úlohu. Prediktory v tejto úlohe sú rozdiely logitov pre jednotlivé kombinované klasifikátory $l_i^c - l_j^c$ pre $c = 1, 2, \ldots, C$. Výsledné parametre sú určené parametrami natrénovaného modelu lda. Lineárna diskriminačná analýza využíva predpoklad normálneho rozdelenia prediktorov. Tento predpoklad ale v našej aplikácii nie je vždy splnený.

Logistická regresia

Podobne ako kombinačná metóda **lda** funguje aj skupina kombinačných metód založených na klasifikačnom algoritme **logistická regresia**. Jednotlivé kombinačné metódy založené na tomto algoritme sa líšia v tom, či je pri trénovaní modelu logistickej regresie použitý aj bias a aká je miera regularizácie.

Základná kombinačná metóda **logreg** trénuje aj bias a používa l_2 regularizáciu. Variant **logreg__no__interc** trénuje model bez biasu, koeficienty b_{ij} teda ostávajú nulové, a taktiež používa l_2 regularizáciu. Ďalšie dva varianty sú analogické predchádzajúcim dvom s tým rozdielom, že pre každú dvojicu tried otestujú niekoľko rôznych hodnôt regularizačného koeficientu C' pre l_2 regularizáciu a vyberú tú, ktorá dáva na validačných dátach najlepšiu presnosť. Tieto kombinačné metódy označujeme ako **logreg__sweep__C** a **logreg__no__interc__sweep__C**.

V počiatočných fázach experimentov sme používali logistickú regresiu z knižnice *www.scikit-learn.org*. Táto knižnica ale neumožňuje využitie GPU a s ním spojenej paralelizácie výpočtov. Po dosiahnutí dobrých výsledkov s použitím **logreg** metód sme preto pristúpili k vlastnej implementácii. Pri implementácii sme využili skutočnosť, že stratová funkcia metódy maximálnej vierohodnosti (ang. negative log likelihood) je konvexná [8] a konvexnosť si zachováva aj pri pridaní l_2 regularizácie. Regularizovanú stratovú funkciu logistickej regresie môžeme vyjadriť ako

$$l(\boldsymbol{w}, b) = \frac{\boldsymbol{w}^T \boldsymbol{w}}{C} - \frac{C'}{N} \sum_{n=1}^{N} \{y_n \log(p(\boldsymbol{x}_n, \boldsymbol{w}, b)) + (1 - y_n) \log(1 - p(\boldsymbol{x}_n, \boldsymbol{w}, b))\}, \quad (6.6)$$

kde $\boldsymbol{w}$ sú váhy trénovaného modelu, b je bias trénovaného modelu, N je počet trénovacích dát, C je počet kombinovaných klasifikátorov, y_n sú označenia tried trénovacích

dát (0, alebo 1), $\boldsymbol{x}_n$ sú príznaky trénovacích dát a C' je regularizačný koeficient. Funkcia $p(\boldsymbol{x}_n, \boldsymbol{w}, b)$ produkuje pravdepodobnostný výstup z modelu logistickej regresie s využitím funkcie expit

$$p(\boldsymbol{x}_n, \boldsymbol{w}, b) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x}_n + b)}}. \tag{6.7}$$

Sčítaním konvexných stratových funkcií pre všetky modely trénované na rôznych dvojiciach tried získame konvexnú výslednú funkciu. Táto výsledná funkcia má podobu

$$\begin{aligned} &\sum_{i=1}^{K} \sum_{j=i+1}^{K} l(\boldsymbol{w}_{ij}, b_{ij}) = \\ &\sum_{i=1}^{K} \sum_{j=i+1}^{K} \frac{{\boldsymbol{w}_{ij}}^T \boldsymbol{w}_{ij}}{C} - \\ &\qquad \frac{C'}{N} \sum_{n=1}^{N} \{ y_n^{ij} \log(p(\boldsymbol{x}_n^{ij}, \boldsymbol{w}_{ij}, b_{ij})) + \\ &\qquad (1 - y_n^{ij}) \log(1 - p(\boldsymbol{x}_n^{ij}, \boldsymbol{w}_{ij}, b_{ij})) \}, \end{aligned} \tag{6.8}$$

kde K je celkový počet tried, y_n^{ij} a $\boldsymbol{x}_n^{ij}$ sú označenia tried a príznaky trénovacích dát pre model rozlišujúci medzi triedami i a j a $\boldsymbol{w}_{ij}, b_{ij}$ sú parametre tohto modelu. Ako trénovacie príznaky $\boldsymbol{x}_n^{ij}$ využívame rozdiely logitov vzoriek, ktoré patria do jednej z tried i, j, teda $\boldsymbol{x}_n^{ij} = (l^1_{i\,n} - l^1_{j\,n}, l^2_{i\,n} - l^2_{j\,n}, \ldots, l^C_{i\;n} - l^C_{j\;n})^T$, kde index n prechádza cez vzorky, ktoré patria do triedy i alebo j a $l^c_{i\,n}$ je logit pre vzorku n vo výstupe kombinovaného klasifikátora c pre triedu i. Vektor váh $\boldsymbol{w}_{ij} = (w^1_{ij}, w^2_{ij}, \ldots, w^C_{ij})^T$ obsahuje váhy zadefinované v (6.4). Z rovnice (6.8) môžeme vidieť, že sčítanec prislúchajúci dvojici tried i, j je závislý len na parametroch $\boldsymbol{w}_{i,j}, b_{ij}$. Konvexná funkcia (6.8) teda nadobudne svoje minimum keď nadobudne svoje minimum každý zo sčítancov prislúchajúcich jednotlivým dvojiciam tried. Táto skutočnosť nám umožňuje s pomocou gradientovej optimalizačnej metódy trénovať niekoľko modelov logistickej regresie súčasne.

Gradientové metódy

Ďalšia skupina kombinačných metód využíva takzvané end-to-end trénovanie. Hodnoty kombinačných koeficientov sú v tomto prípade určované na základe výslednej pravdepodobnostnej klasifikácie vystupujúcej z párovej zväzovacej metódy. Trénovanie je realizované pomocou metódy stochastického gradientového zostupu (ang. stochastic gradient descent) (SGD) a s použitím stratovej funkcie metódy maximálnej vierohodnosti (ang. negative log likelihood) (NLL). Tieto kombinačné metódy sú označované

ako **grad_m1**, **grad_m2**, **grad_bc** a **grad_sbt**, kde časť názvu za podčiarkovníkom zodpovedá označeniu párovej zväzovacej metódy použitej pri trénovaní. Pri predikcii môžu byť koeficienty získané pomocou týchto metód použité aj s inou párovou zväzovacou metódou ako bola použitá pri trénovaní, rovnako ako pri ostatných kombinačných metódach.

Random

Najmä za účelom overenia efektívnosti kombinovania jednotlivých párových klasifikátorov pre každú dvojicu tried sme navrhli kombinačnú metódu **random**. Táto kombinačná metóda pri inicializácii pre každú dvojicu tried náhodne vyberie jeden z kombinovaných klasifikátorov. Vybraný klasifikátor dostane váhu 1, ostatné klasifikátory dostanú váhu 0. Bias je v tejto metóde nastavený na 0.

Neurónová sieť

Otestovali sme tiež kombinačnú metódu využívajúcu jednoduchú plne prepojenú neurónovú sieť. Vstupom do tejto siete sú logity kombinovaných klasifikátorov a výstupom je matica párových pravdepodobností, ktorá vstupuje do párovej zväzovacej metódy. Ani po otestovaní viacerých architektúr siete a konfigurácií hyperparametrov sme neboli schopní s touto kombinačnou metódou dosiahnuť výsledky použiteľnej kvality, preto ju v nasledujúcich častiach neuvádzame.

Modifikácia kombinačných metód

Pre lepšie zohľadnenie toho, akú váhu prisudzujú kombinované klasifikátory určitej dvojici tried sme implementovali rozšírenie zanášajúce neistotu do matice párových pravdepodobností. Rozšírenie vykonáva úpravu matice párových pravdepodobností, ktorá vystupuje z kombinačnej metódy predtým ako vstúpi do párovej zväzovacej metódy. Činnosť modifikácie je možné vyjadriť pomocou vzorca

$$r'_{ij} = \alpha_{ij} r_{ij} + (1 - \alpha_{ij}) \times 0.5, \tag{6.9}$$

kde r_{ij} je prvok matice párových pravdepodobností $\boldsymbol{R}$, ktorá vystupuje z kombinačnej metódy, r'_{ij} je prvok upravenej matice párových pravdepodobností $\boldsymbol{R}'$, ktorá vstupuje do párovej zväzovacej metódy a α_{ij} je koeficient, ktorý vyjadruje mieru príslušnosti

klasifikovaného objektu do jednej z tried i,j. Koeficient α_{ij} spočítavame podľa vzorca

$$\alpha_{ij} = \frac{1}{C}\sum_{c=1}^{C}(p_i^c + p_j^c), \tag{6.10}$$

kde p_i^c je kalibrovaný pravdepodobnostný výstup z kombinovaného klasifikátora c pre triedu i. V experimentálnych výstupoch označujeme použitie tejto modifikácie pridaním **.uncert** k názvu kombinačnej metódy. Kombinačná metóda **logreg** s použitím tejto modifikácie bude teda označená ako **logreg.uncert**.

6.2 Detekcia neznámych vzoriek

Kľúčovým problémom väčšiny moderných klasifikátorov je, že sú natrénované na obmedzenú množinu tried a pri klasifikovaní vzorky, ktorá nespadá do tejto množiny je ich výstup nevyhnutne nesprávny. Tento problém sa snaží riešiť funkcionalita detekcie neznámych vzoriek (ang. out of distribution (OOD) detection), ktorá umožňuje klasifikátoru vyhodnotiť vzorku ako neznámu a nezaradiť ju do žiadnej z tried na ktoré bol natrénovaný. OOD detekciu je možné implementovať rôznymi spôsobmi, niektoré vyžadujú úpravy modelu, alebo procesu trénovania, iné je možné aplikovať na akýkoľvek pravdepodobnostný klasifikátor. OOD detekciu je možné realizovať ak klasifiátor dokáže kvantifikovať istotu, prípadne neistotu, s vykonanou predikciou. Na základe výstupov na vhodnej validačnej množine a so zohľadnením požiadaviek riešeného problému je potom možné zvoliť hraničnú hodnotu istoty (resp. neistoty). Predikcie s istotou (resp. neistotou) na jednej strane tejto hraničnej hodnoty sú považované za platné a predikcie na druhej strane sú považované za detekovanú neznámu vzorku.

Jeden zo základných prístupov, ktorý vykazuje dobrú úspešnosť v rozličných aplikáciách [81], modeluje istotu (ang. confidence) modelu pomocou maximálnej hodnoty spomedzi pravdepodobností na jeho výstupe. V našich experimentoch tento prístup označujeme ako MSP (ang. maximum softmax probability).

Použitie párových zväzovacích metód otvára možnosti využitiu nových prístupov ku kvantifikovaniu istoty resp. neistoty klasifikátora s vykonanou predikciou. Pri kombinovaní množiny všetkých párových predikcií máme k dispozícii veľké množstvo redundantných informácií. Výslednú viactriednu klasifikáciu je možné získať z každého stĺpca matice kombinovaných párových pravdepodobností. V praktických prípadoch ale môže byť každá z týchto predikcií odlišná. Autori v práci [78] operujú s pojmom

Bradley-Terryho varieta. Ide o varietu, na ktorej je systém Bradley-Terryho rovníc (5.15) riešiteľný. Matica párových pravdepodobností leží v Bradley-Terryho variete v prípade, že sú kombinované párové predikcie úplne konzistentné a z každého stĺpca by sme získali rovnakú výslednú predikciu [82]. Pri klasifikovaní neznámeho objektu môžeme predpokladať menej konzistentné párové predikcie a maticu párových predikcí viac vzdialenú od Bradley-Terryho variety. Existuje viacero spôsobov, ktorými je možné túto vzdialenosť vyjadriť. Pri metódach autorov Wu, Lin a Weng [12] sú prirodzenými kandidátmi na jej vyjadrenie účelové funkcie pri interpretácii ako optimalizačná úloha. Pre metódu **m1** je to funkcia (5.9) a pre metódu **m2** funkcia (5.10). Súčasťou párovej zväzovacej metódy **bc** je kolmý priemet vektoru **s** získaného zo vstupných párových pravdepodobností na vektor **u**, ktorý leží v Bradley-Terryho variete. Pri metóde **bc** preto neistotu vyjadrujeme ako euklidovskú vzdialenosť vektorov **s** a **u**.

6.3 Predikcia pri úlohách s veľkým počtom tried

Výpočtová a pamäťová náročnosť kombinačných metód ako aj párových zväzovacích metód, z ktorých sa WLE skladá rastie priamo úmerne s druhou mocninou počtu tried riešeného problému. Pri trénovaní kombinačných metód, ktoré pre každú dvojicu tried hľadajú hodnoty kombinačných parametrov sa nedokážeme vyhnúť spracovaniu všetkých dvojíc tried. Pri predikcii ale môžeme výpočtovú aj pamäťovú náročnosť ansámblovej predikcie znížiť s využitím výstupov kombinovaných klasifikátorov predtým ako ich poskytneme ansámblovému modelu. Navrhli sme preto úpravu založenú na úvahe, že je nepravdepodobné, aby sa trieda, ktorá má u všetkých kombinovaných klasifikátorov nízku podporu dostala v ansámblovej predikcii medzi triedy s vysokou podporou. Úprava teda funguje tak, že vezme z každého z kombinovaných klasifikátorov stanovený počet tried s najvyššou podporou, vytvorí zjednotenie týchto tried a ansámblový model skombinuje predikcie len pre triedy v zjednotení. Zvyšné triedy majú vo výstupe ansámblového modelu nulovú pravdepodobnosť. Počet tried, ktoré sú vybrané pre každý z kombinovaných klasifikátorov je riadený hyperparametrom *topl*.

6.4 Teoretická analýza pre homoskedastické dáta

V tejto sekcii uvádzame analýzu teoretického prípadu úlohy s normálnym homoskedastickým rozdelením vstupných dát. Členy analyzovaného ansámblu sú trénované na komplementárnych dátach a nesú komplementárne informácie. Za týchto predpokladov ukážeme, že ak budeme kombinovať optimálne klasifikátory, tak aj výstup PWE ansámblu bude optimálny. Táto analýza bola vykonaná v [83].

Uvažujme dáta rozdelené do troch tried O_1, O_2, O_3 s troma číselnými príznakmi $\mathbf{x} = (x_1, x_2, x_3)$. Príznaky pre každú triedu sú rozdelené podľa viacrozmerného normálneho rozdelenia so spoločnou kovariančnou maticou $\mathbf{\Sigma}$ podľa

$$p(\mathbf{x}|O_i) \sim N(\mathbf{m_i}, \mathbf{\Sigma}). \tag{6.11}$$

Ako členy ansámblu uvažujeme tri klasifikátory, z ktorých každý počas tréningu mal k dispozícii len jeden z príznakov x_1, x_2, x_3. Predpokladáme, že každý z nich poskytuje, vzhľadom k dátach, ktoré má k dispozícii, optimálnu klasifikáciu. Optimálna viactriedna klasifikácia s použitím príznaku x_k je daná Bayesovou vetou ako

$$p\left(O_i|x_k\right) = \frac{p\left(x_k|O_i\right)p(O_i)}{\sum_j p(x_k|O_j)p(O_j)}. \tag{6.12}$$

Ako vstup do kombinačnej metódy PWE používame rozdiely logitov, pre dvojice tried, získané z jednotlivých viactriednych kombinovaných klasifikátorov. Rozdiel logitov pre triedy O_i a O_j získaný z klasifikátora, ktorý mal k dispozícii príznak x_k je daný ako

$$\log\frac{p(O_i|x_k)}{p(O_j|x_k)} = \log\frac{p(x_k|O_i)}{p(x_k|O_j)} + \log\frac{p(O_i)}{p(O_j)}. \tag{6.13}$$

Ľubovoľná lineárna transformácia normálneho rozdelenia je tiež normálne rozdelenie. Túto skutočnosť využijeme pre určenie rozdelenia pravdepodobnosti $p(x_k|O_i)$. Zobrazenie rozdelenia príznakov $\mathbf{x}$ pre triedu O_i na k-tu súradnicu x_k má rozdelenie

$$\sim N(\mathbf{e_k}^T\mathbf{m_i}, \mathbf{e_k}^T\mathbf{\Sigma e_k}) =: N(m_{ki}, \Sigma_k) \tag{6.14}$$

kde $\mathbf{e_k}$ je nulový vektor s jednotkou na k-tej pozícii a Σ_k je k-ty prvok hlavnej diagonály matice $\mathbf{\Sigma}$.

Pre logaritmus distribučnej funkcie $p_N(x;\mu,\sigma^2)$ normálneho rozdelenia $N(\mu,\sigma^2)$ platí

$$\log(p_N(x;\mu,\sigma^2)) = -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 - \log(\sigma\sqrt{2\pi}). \tag{6.15}$$

Pre $\log \frac{p(x_k|O_i)}{p(x_k|O_j)}$ z rovnice (6.13) potom máme

$$\log \frac{p(x_k|O_i)}{p(x_k|O_j)} = \log \frac{p_N(x_k; m_{ki}, \Sigma_k)}{p_N(x_k; m_{kj}, \Sigma_k)} = \tag{6.16}$$

$$= -\frac{1}{2}\left\{ \frac{(x_k - m_{ki})^2}{\Sigma_k} - \frac{(x_k - m_{kj})^2}{\Sigma_k} \right\} = \tag{6.17}$$

$$= \frac{1}{2\Sigma_k}\left(2x_k(m_{ki} - m_{kj}) + (m_{kj}^2 - m_{ki}^2) \right) \tag{6.18}$$

Z posledného riadku môžeme vidieť, že ide o afinnú transformáciu príznaku x_k. Konkrétne ide o posunutie výrazu $\frac{m_{ki}-m_{kj}}{\Sigma_k} x_k$. Z (6.13) vyplýva, že $\log \frac{p(O_i|x_k)}{p(O_j|x_k)}$ je tiež posunutím toho istého výrazu. V analyzovanom prípade majú teda rozdiely logitov dané ako $\log \frac{p(O_i|x_k)}{p(O_j|x_k)}$ normálne homoskedastické rozdelenia pravdepodobnosti. Optimálna hranica medzi triedami i a j má preto v priestore príznakov podobu lineárnej nadroviny.

Parametrické kombinačné metódy, ktorých činnosť je popísaná rovnicou (6.4), vytvárajú lineárnu kombináciu rozdielov logitov pre jednotlivé kombinované klasifikátory. Ak $m_{ki} \neq m_{kj}$ pre $i \neq j$, potom lineárnym kombinovaním (6.13) dokážeme vyjadriť ľubovoľnú lineárnu nadrovinu. Optimalizovaním, ktoré vykonávame pri tréningu parametrických kombinačných metód, tak nájdeme optimálnu rozdeľujúcu nadrovinu medzi triedami O_i a O_j v $\mathbf{R}^3$.

V prípade, že pre niektorú dvojicu tried $i \neq j$ platí $m_{ki} = m_{kj}$, môžeme použiť zistenia Lineárnej diskriminačnej analýzy (LDA). Konkrétne, že normála k LDA nadrovine je daná ako $\mathbf{\Sigma}^{-1}(\mathbf{m_i} - \mathbf{m_j})$ [8]. Z rovnosti $m_{ki} = m_{kj}$ tak vyplýva, že koeficient pre x_k v optimálnej oddeľujúcej nadrovine je nulový. Aj v takomto prípade teda môžeme pomocou trénovania v kombinačnej metóde nájsť optimálnu rozdeľujúcu nadrovinu.

Ukázali sme, že parametrické kombinačné metódy v analyzovanom prípade získajú optimálne párové klasifikátory. V ďalšom kroku sú tieto párové klasifikátory kombinované pomocou párovej zväzovacej metódy, ktorá vytvorí výsledný viactriedny klasifikátor. Párové zväzovacie metódy sa pokúšajú nájsť približné riešenie, v praxi väčšinou nekonzistentnej sústavy Bradley-Terryho rovníc (5.15). Keďže v analyzovanom prípade získame optimálny klasifikátor pre každú dvojicu tried, systém Bradley-Terryho rovníc bude konzistentný. V takomto prípade všetky používané párové zväzovacie metódy tento systém úspešne vyriešia a získajú tak optimálnu viactriednu klasifikáciu.

Táto analýza bola vykonaná za predpokladu homoskedastického viacrozmerného

normálneho rozdelenia. Takýto predpoklad v praxi väčšinou neplatí, preto okrem LDA testujeme ako parametrickú kombinačnú metódu aj logistickú regresiu.

Ďalším predpokladom je kombinovanie členov ansámblu s komplementárnymi informáciami. Takáto situácia môže nastávať pri klasifikovaní multimodálnych dát, kde jednotlivé klasifikátory spracovávajú dáta z odlišných senzorov [84]. Opísaná analýza bola vykonaná až po dokončení experimentov popísaných v nasledujúcej kapitole. Experimenty s kombinovaním klasifikátorov poskytujúcich komplementárne informácie preto zostávajú ako možné smerovanie ďalšieho výskumu.

Kapitola 7

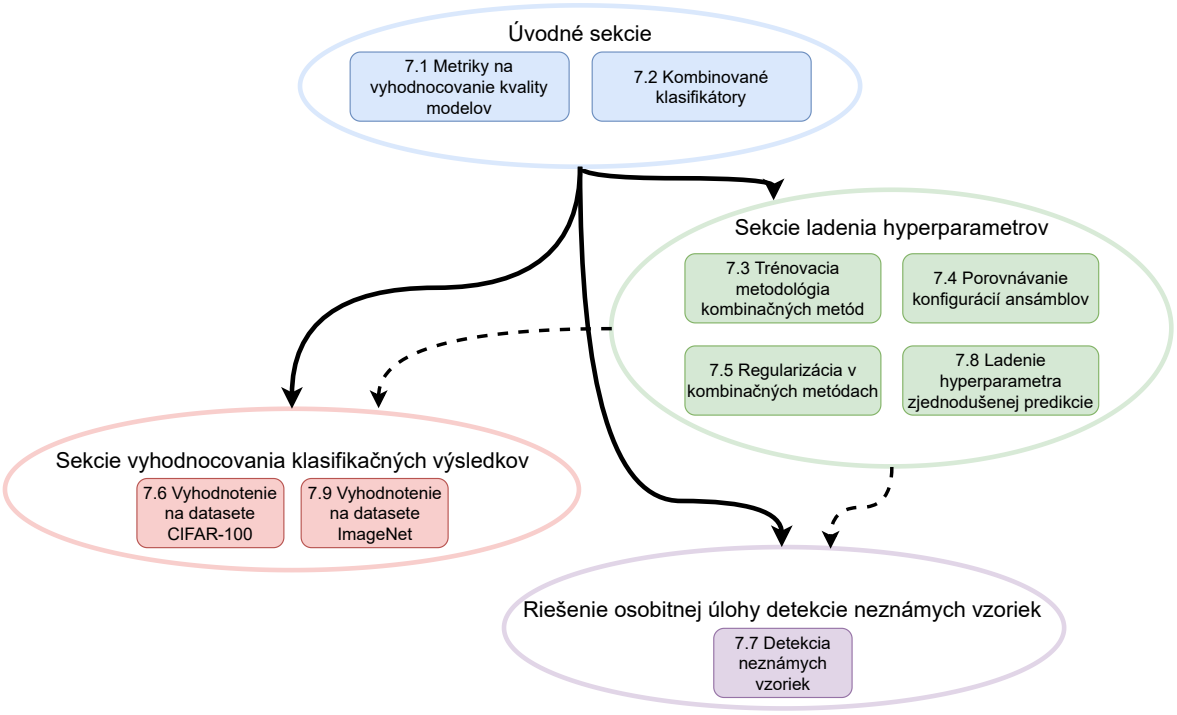
Popis experimentov a výsledkov

Lineárny vážený ansámbel (WLE), ktorý testujeme je tvorený kombináciou kombinačnej metódy a párovej zväzovacej metódy. Pre lepšiu prehľadnosť a stručnosť v nasledujúcich sekciách označujeme WLE ako konfigurácia kombinačná metóda + párová zväzovacia metóda. Teda napríklad ansámbel zložený z kombinačnej metódy **logreg** a párovej zväzovacej metódy **sbt** označíme ako konfigurácia **logreg + sbt**.

Organizácia tejto kapitoly je nasledovná. V sekcii 7.1 popisujeme metriky, podľa ktorých vyhodnocujeme kvalitu pravdepodobnostných klasifikácií. V sekcii 7.2 poskytujeme prehľad klasifikátorov, ktoré kombinujeme pomocou WLE vo vykonávaných experimentoch.

Ďalšie sekcie sa venujú samotným experimentom. Ako prvej sa venujeme v sekcii 7.3 trénovacej metodológii kombinačných metód. Po jej odladení porovnávame jednotlivé konfigurácie WLE v sekcii 7.4. Pri porovnaní sme zistili sľubné výsledky pre niektoré výpočtovo náročné kombinačné metódy založené na logistickej regresii. V sekcii 7.5 preto skúmame možné úpravy týchto metód, ktoré by zefektívnili výpočet. V sekcii 7.6 vyhodnocujeme výsledky niekoľkých vybraných odladených konfigurácií WLE na upravenom datasete CIFAR-100. V sekcii 7.7 skúmame možnosti detekcie neznámych vzoriek s pomocou merania nekonzistencie výstupov kombinovaných klasifikátorov prostredníctvom dodatočného výstupu z párových zväzovacích metód. V sekcii 7.8 sa venujeme riešeniu úloh s veľkým počtom tried a testujeme prístup, ktorý zefektívňuje priebeh predikcie. Na záver v sekcii 7.9 testujeme vybrané konfigurácie s odladenými hyperparametrami na datasete ImageNet.

Pre prehľadnosť sme organizáciu tejto kapitoly vykreslili v podobe diagramu 7.1.



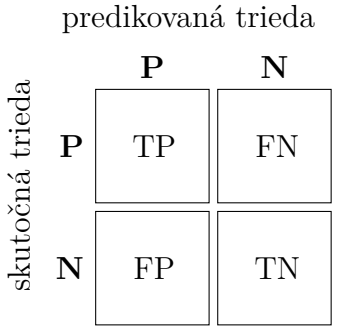
Obrázok 7.1: Diagram organizácie a nadväznosti sekcií kapitoly Experimentov.

7.1 Metriky na vyhodnocovanie kvality modelov

Pri vyhodnocovaní experimentov používame niekoľko metrík kvality. Klasická metrika pri klasifikácii je presnosť klasifikácie (ang. accuracy). Presnosť vyjadruje pre akú časť z klasifikovaných vzoriek bola výsledná trieda správne určená. Pri pravdepodobnostných klasifikátoroch existuje aj zovšeobecnenie presnosti na viac ako jednu najpravdepodobnejšiu triedu. Toto zovšeobecnenie sa nazýva *top-k* presnosť a vyjadruje u akej časti klasifikovaných vzoriek sa správna trieda nachádza medzi k triedami s najvyššou predikovanou pravdepodobnosťou. Pri datasete ImageNet sa štandardne využívajú dve presnosti, *top-1* a *top-5*. Dôvod je najmä ten, že na obrázkoch v tomto datasete je často veľa rôznych objektov, označený je ale len podľa jedného z nich. Klasifikátor nemusí vždy správne odhadnúť, podľa ktorého z nich bol obrázok označený.

Ďalšia metrika, ktorá sa tiež často používa pri trénovaní neurónových sietí, je pokutová funkcia metódy maximálnej vierohodnosti (ang. negative log likelihood) (NLL).

Pre praktické použitie klasifikátora je potrebné aby bol dobre kalibrovaný. Kalibrácii neurónových sietí sme sa venovali v kapitole 3. Mieru kalibračnej chyby meria metrika odhad kalibračnej chyby (ang. estimated calibration error) (ECE). Túto metriku počítame pomocou postupu popísaného v [28], konkrétne postupom s rovnakou



Obrázok 7.2: Matica zámen

početnosťou vzoriek v jednotlivých intervaloch.

V sekcii 7.7 sa venujeme detekcii vzoriek, ktoré nepatria do rozdelenia, na ktorom bol klasifikátor natrénovaný (ang. out-of-distribution detection). Takúto úlohu je možné interpretovať ako dvojtriednu klasifikačnú úlohu. Kvalitu algoritmov zameraných na dvojtriednu klasifikáciu je možné vyhodnocovať podľa špecializovaných metrík odvodených z matice zámen (ang. confusion matrix). Matica zámen pre dvojtriednu klasifikačnú úlohu je zobrazená na obrázku 7.2, triedy sú tu označené ako pozitívna (P) a negatívna (N). Matica obsahuje štyri početnosti:

- TP - skutočne pozitívny (ang. true positive) - prvky, ktoré sú pozitívne a boli označené ako pozitívne,
- FN - falošne negatívny (ang. false negative) - prvky, ktoré sú pozitívne, ale boli označené ako negatívne,
- FP - falošne pozitívny (ang. false positive) - prvky, ktoré sú negatívne, ale boli označené ako pozitívne,
- TN - skutočne negatívny (ang. true negative) - prvky, ktoré sú negatívne a boli označené ako negatívne.

Z matice zámen je odvodené množstvo metrík, my sa venujeme najmä trom:

- úplnosť (ang. recall alebo true positive rate (TPR)) = $\frac{TP}{TP+FN}$,
- precíznosť (ang. precision) = $\frac{TP}{TP+FP}$,
- pravdepodobnosť falošného poplachu (ang. false positive rate (FPR)) = $\frac{FP}{TN+FP}$.

Algoritmy, ktoré na OOD detekciu využívame poskytujú ako svoj výstup mieru príslušnosti do jednej z tried P, alebo N. To nám umožňuje zvoliť si prahovú hodnotu (ang. threshold) a do danej triedy zaradiť prvky, ktoré majú mieru príslušnosti vyššiu ako zvolená prahová hodnota. Pre rôzne prahové hodnoty tak získame rôzne výsledky klasifikácie s rôznymi hodnotami sledovaných metrík. Takéto klasifikátory sa charakterizujú pomocou kriviek vytvorených z bodov, kde prvá súradnica bodu je hodnota jednej metriky a druhá súradnica hodnota inej metriky pre tú istú prahovú hodnotu. Často využívaná je krivka vytvorená dvojicou metrík FPR a TPR, ktorá sa nazýva operačná charakteristika prijímača (ROC z ang. Receiver Operating Characteristic). Ďalšia často využívaná krivka tvorená metrikami precíznosť a úplnosť, nazývaná PR krivka, môže poskytnúť cenné informácie najmä v prípade nerovnomerného zastúpenia pozitívnych a negatívnych vzoriek v datasete [85].

Tieto krivky poskytujú užitočné informácie o skúmanom klasifikátore, je ale ťažké na ich základe porovnávať viacero klasifikátorov. Preto sa využíva tiež metrika plocha pod krivkou, táto metrika sa označuje ako AUROC pre krivku ROC a AUPRC pre krivku PR. Pre ROC krivku je možné spočítať plochu pod ňou nájdením bodov, ktoré tvoria konvexný obal a následnou aplikáciou lichobežníkovej metódy. Pre PR krivku nie je korektné použiť lichobežníkovú metódu, pretože tá spája body lineárnymi úsečkami a v PR priestore takéto úsečky nepredstavujú dosiahnuteľné klasifikácie [85]. Autori v [85] vyvinuli algoritmus, ktorý dokáže korektne spočítať AUPRC. Tento algoritmus spočíva v nájdení bodov konvexného obalu ROC krivky, transformácii týchto bodov do PR priestoru a následnom interpolovaní medzi získanými bodmi. Použitá interpolácia zohľadňuje pomer TP a FP medzi spracovávanými bodmi.

7.2 Kombinované klasifikátory

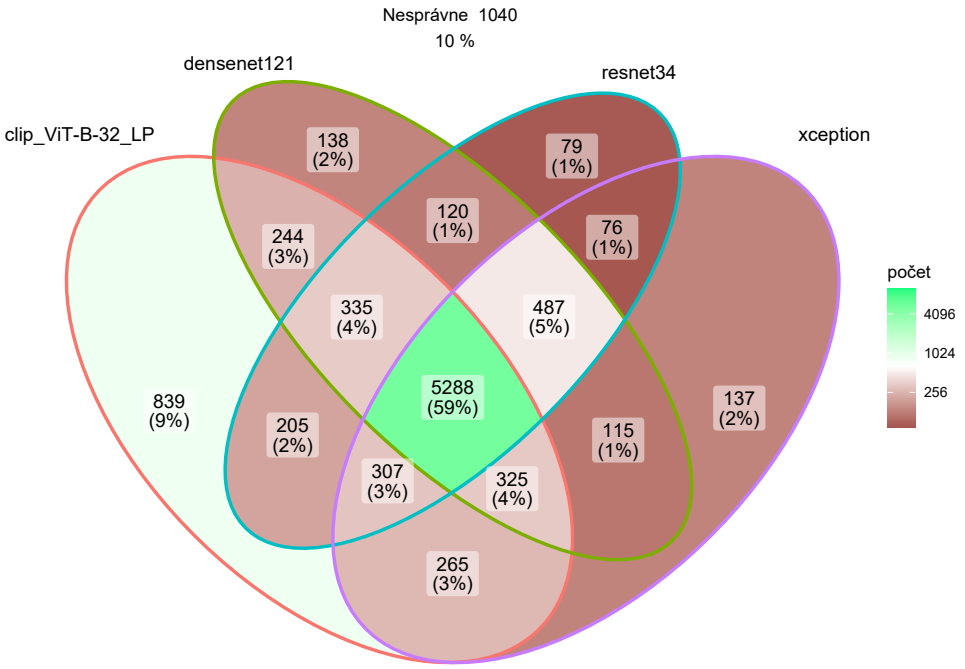
V nasledujúcich testoch vykonaných na datasetoch CIFAR-10 a CIFAR-100 používame na zostavovanie ansámblov niekoľko neurónových sietí. Ide o siete s rozličnými architektúrami, aby sme zabezpečili dostatočnú diverzitu kombinovaných predikcií. Neurónové siete, ktoré používame sú: googlenet [86], seresnet34 [87], resnext101 [88], stochasticdepth50 [89], resnet34 [90], densenet121 [91] a xception [92]. Tieto siete sa trénujú klasickým spôsobom na trénovacej množine datasetu, pre ktorý sú použité. Ok-

rem týchto sietí sme použili tiež predtrénovaný extraktor príznakov clip [93], konkrétne s architektúrami siete ViT-B-32 a ViT-B-16. Na výstupoch tohto extraktora príznakov sme na príslušnej trénovacej množine natrénovali multinomiálnu regresiu. V experimentoch označujeme tieto modely ako clip_ViT-B-32_LP a clip_ViT-B-16_LP, kde LP je z ang. linear probe. Výstupy týchto modelov používame rovnakým spôsobom ako výstupy ostatných neurónových sietí.

Pre znázornenie diverzity sietí sme vykreslili Vennov diagram správnych predikcií štyroch z nich. Diagram pre dataset CIFAR-100 je zobrazený na obrázku 7.3. Z diagramu je zrejmé, že každá zo sietí úspešne klasifikovala niekoľko vzoriek, ktoré sa nepodarilo úspešne klasifikovať žiadnej inej z uvažovaných sietí.

V sekcii venujúcej sa detekcii neznámych vzoriek využívame klasifikátory predtrénované na kompletnom datasete ImageNet21k [33] a aplikujeme ich na dataset CIFAR-10 resp. CIFAR-100 dotrénovaním pomocou techniky nazývanej prenos učenia (ang. transfer learning). Ide konkrétne o konvolučné siete ResNet 50x1 a ResNet 101x3 [94] (v experimentoch označované ako R50x1 a R101x3) dostupné v repozitári [95], MLP Mixer-B/16 [96] a obrazové transformery ViT-B_16 a R50+ViT-B_16 [20](v experimentoch označované ako M_B16, B16 a R50_B16) dostupné v repozitári [97].

V experimentoch na datasete ImageNet1k [34] používame obrazové transformery ViT-Ti_16, ViT-S_16, ViT-B_16, ViT-B_32 [20], MLP Mixer-B/16 [96] a kombináciu konvolučnej siete a obrazového transformera R26+S/32 [98]. Všetky tieto siete sú dostupné v repozitári [97]. V experimentoch sú označované ako Ti16, B16, S16, B32, M_B16 a R26_S32. Tieto neurónové siete boli predtrénované na kompletnom datasete ImageNet21k [33] a nami prispôsobené na použitie na datasete ImageNet1k dotrénovaním poslednej vrstvy. Tento postup je technicky ekvivalentný odstráneniu poslednej vrstvy a jej nahradeniu multinomiálnou regresiou.



Obrázok 7.3: Vennov diagram správnych predikcií neurónových sietí na datasete CIFAR-100. Percentuálne údaje vyjadrujú podiel zo všetkých správne klasifikovaných dát ktoroukoľvek zo sietí. Ako môžeme z obrázka vidieť, najväčší počet obrázkov správne klasifikovaných len jednou zo sietí pripadá na model clip.

7.3 Trénovacia metodológia kombinačných metód

Cieľ experimentov v sekcii: Určiť veľkosť trénovacej množiny pre kombinačné metódy a vyšetriť potrebu oddelenej trénovacej množiny pre kombinačné metódy a pre kombinované klasifikátory.

Parametrické kombinačné metódy vyžadujú tréning, ktorý je možné realizovať rôznymi stratégiami a s rôznymi nastaveniami hyperparametrov. Aby sme mohli jednotlivé kombinačné metódy objektívne porovnávať, bolo najprv potrebné určiť pre ne vhodnú tréningovú konfiguráciu. Pre veľké množstvo experimentov spojených s konfigurovaním kombinačných metód sme tieto experimenty vykonávali na pomerne malých datasetoch CIFAR-10 a CIFAR-100.

7.3.1 Veľkosť trénovacej množiny

Pre tieto experimenty používame klasifikátory googlenet, resnext101, seresnet34, stochasticdepth50, clip_ViT-B-32_LP a clip_ViT-B-16_LP.

Ako prvé potrebujeme určiť veľkosť trénovacej množiny pre parametrické kombinačné metódy. Za týmto účelom sme vykonávali experimenty s kombinačnými metódami **lda**, **logreg** a **logreg_no_interc**. Každú kombinačnú metódu sme testovali v kombinácii s každou zo štyroch párových zväzovacích metód **m1**, **m2**, **bc** a **sbt**. Experimentálne sme zistili, že pre väčšinu dvojíc tried nie je splnený predpoklad normálneho rozdelenia nezávislých premenných, na ktorom stavia metóda lineárna diskriminačná analýza. Výsledky kombinačnej metódy **lda** preto uvažujeme s menšou váhou na finálne rozhodnutie.

Pre vykonanie experimentu sme stanovili niekoľko veľkostí trénovacích množín pre CIFAR-10 a niekoľko pre CIFAR-100. Následne sme pre každú z týchto veľkostí náhodne vybrali z trénovacej množiny desať podmnožín danej veľkosti tak, aby bolo zachované rovnomerné rozloženie jednotlivých tried v týchto množinách. Na každej z týchto vybratých podmnožín sme natrénovali ansámble s použitím jednotlivých kombinačných metód. Každý z týchto ansámblov sme potom otestovali s použitím každej párovej zväzovacej metódy na testovacej množine zodpovedajúceho datasetu.

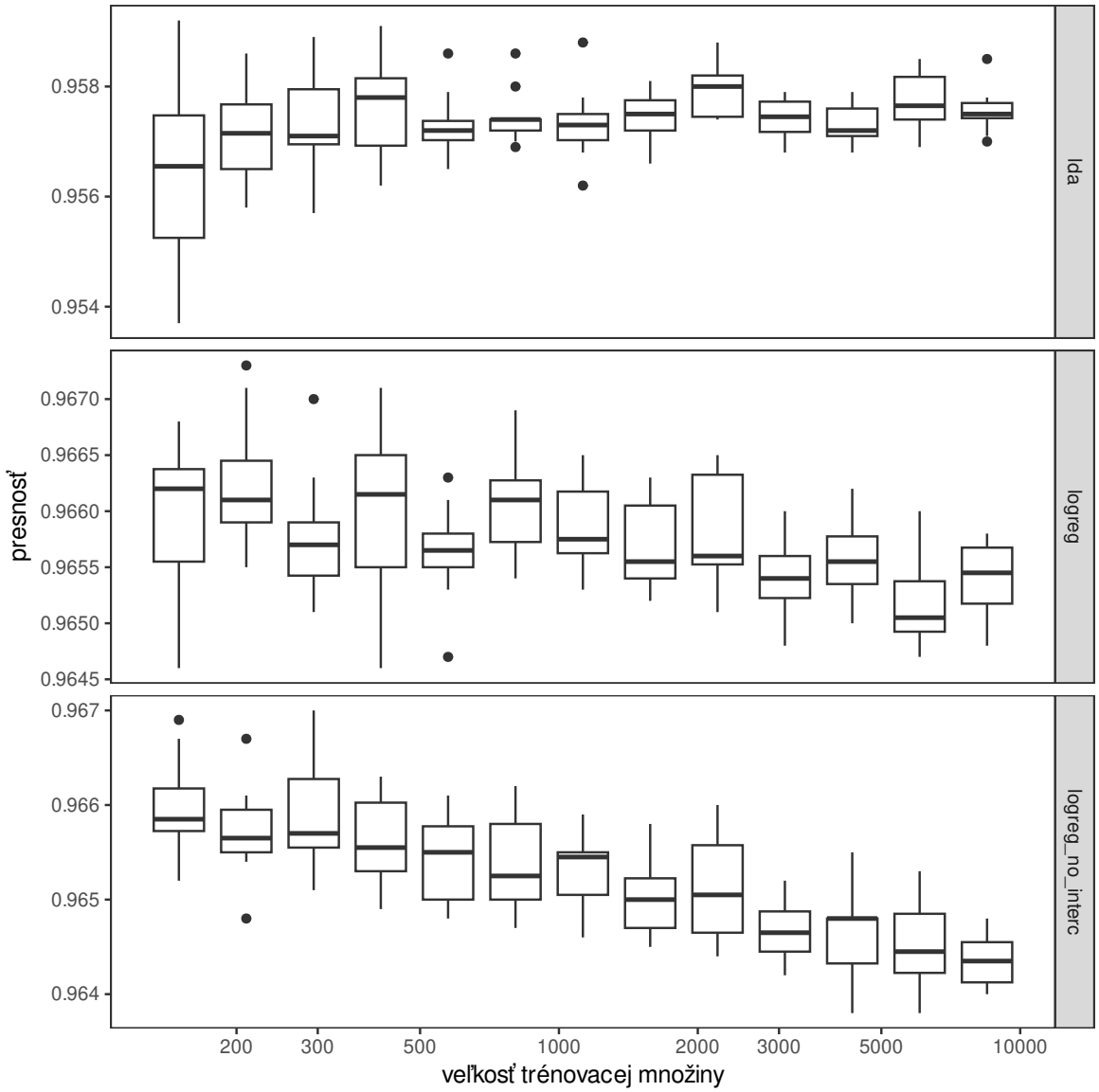
Na obrázku 7.4 je zobrazená presnosť na datasete CIFAR-10 pre ansámble s použitím párovej zväzovacej metódy **bc**, ktorá mala v tomto teste najvyššiu presnosť.

Môžeme vidieť, že so zväčšujúcou sa veľkosťou trénovacej množiny má pre kombinačnú metódu **lda** presnosť stúpavý trend a pre **logreg** metódy klesavý trend. Rozdiely v mediáne presností pre rôzne veľkosti trénovacej množiny sú však malé, len okolo 0.15%, čo predstavuje 15 obrázkov v testovacej množine datasetu CIFAR-10. Okrem presnosti však uvažujeme aj metriky NLL a ECE, ktoré sú zobrazené na obrázkoch 7.5 a 7.6. Na rozdiel od presnosti sa obe tieto metriky snažíme minimalizovať. Ako môžeme na obrázkoch vidieť, pre kombinačnú metódu **lda** sú hodnoty týchto metrík podstatne horšie ako pre kombinačné metódy **logreg**, zameriame sa preto na výsledky kombinačných metód **logreg**. S rastúcou veľkosťou trénovacej množiny majú obe tieto metriky najprv krátky klesavý trend nasledovaný stúpavým trendom. Metrika ECE začína stúpať pri veľkosti trénovacej množiny okolo 300 a metrika NLL pri veľkosti okolo 500.

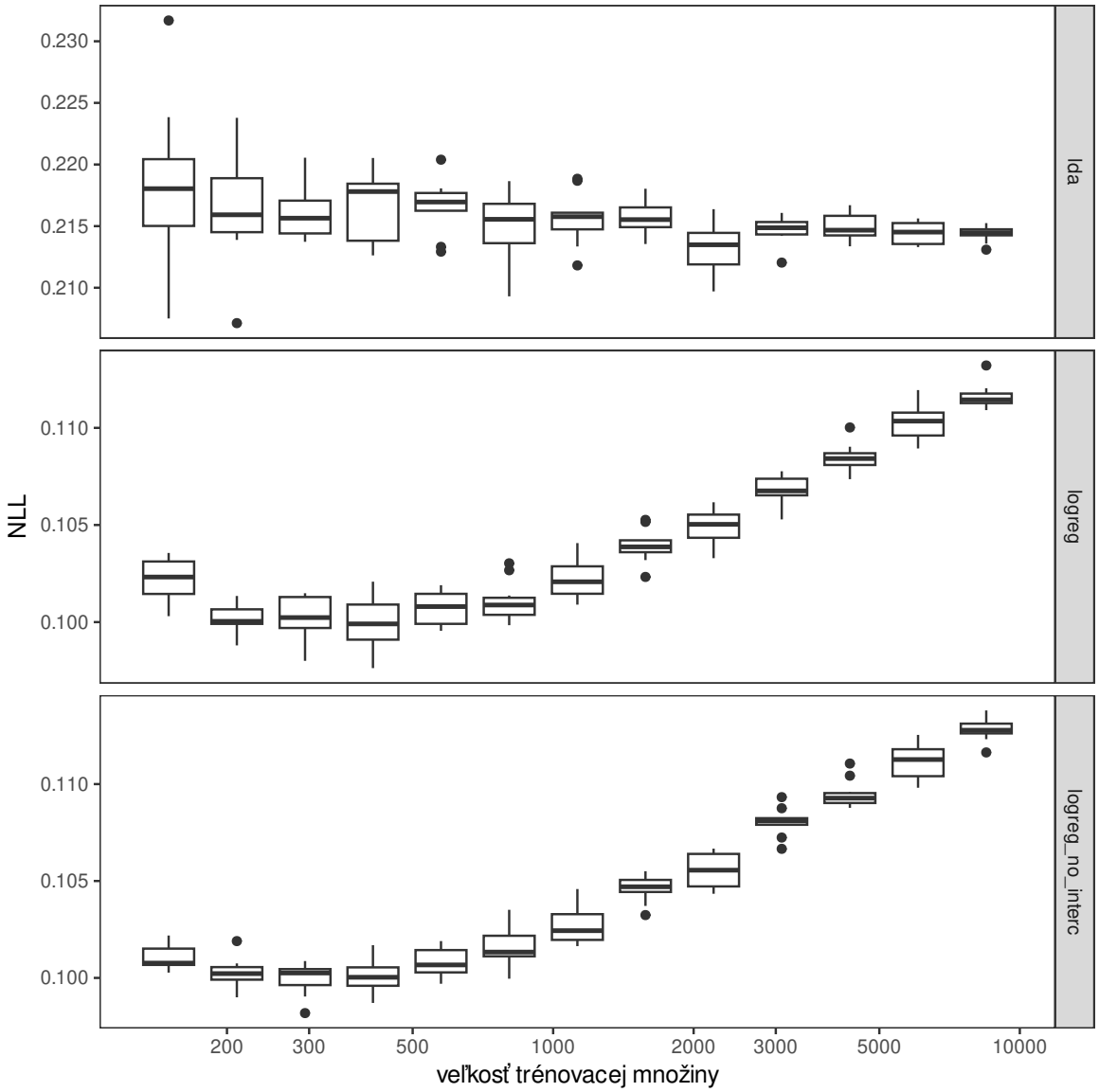
Na datasete CIFAR-100 dosahovala v tomto experimente tiež najvyššiu presnosť párová zväzovacia metóda **bc**. Presnosť, NLL a ECE pre túto metódu sú zobrazené na obrázkoch 7.7, 7.8 a 7.9. Rozdiely v mediánoch presností pre presnejšie kombinačné metódy **logreg** ani v tomto prípade neprekračujú 0.15%. Presnosť pre kombinačné metódy **lda** a **logreg** má s rastúcou veľkosťou trénovacej množiny stúpavý trend a pre kombinačnú metódu **logreg_no_interc** klesavý trend. Metriky NLL a ECE sú znova výrazne lepšie pre kombinačné metódy **logreg** a s rastúcou veľkosťou trénovacej množiny majú jasný klesavý trend.

S uvážením výsledkov pre oba datasety a so zahrnutím praktickej výpočtovej náročnosti trénovania sme sa rozhodli pre parametrické kombinačné metódy používať pre dataset CIFAR-10 trénovaciu množinu veľkosti 500 a pre dataset CIFAR-100 množinu veľkosti 5000. Tieto veľkosti sú konzistentné v tom, že pri trénovaní klasifikátorov na dvojiciach tried pripadá na každú z tried 50 vzoriek.

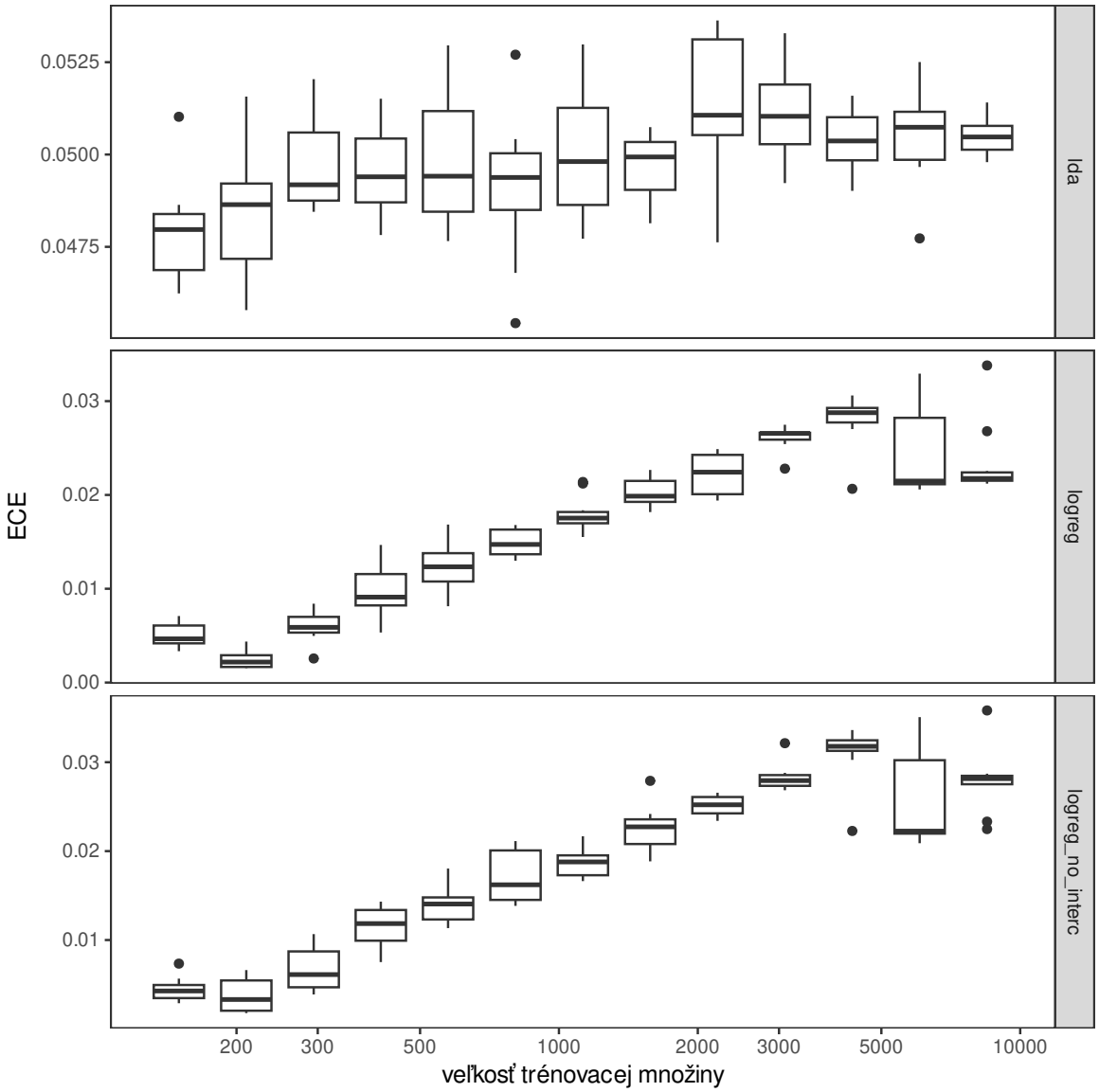
Výsledky pre ostatné párové zväzovacie metódy sú dostupné v prílohe. V týchto výsledkoch je možné pozorovať numerickú nestabilitu párovej zväzovacej metódy **sbt** najmä v kombinácii s kombinačnou metódou **lda**.



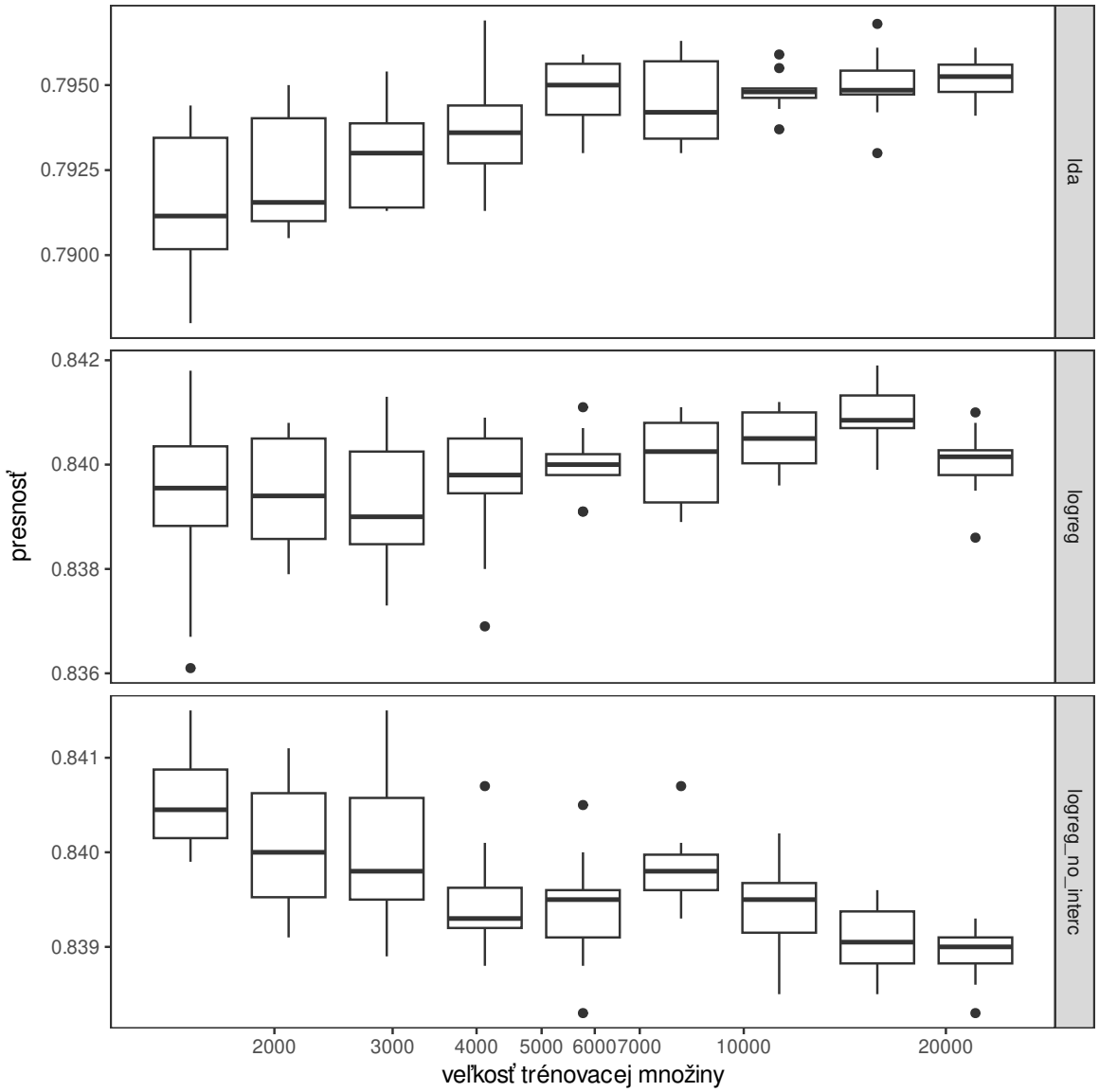
Obrázok 7.4: Presnosť ansámblu s rôznymi veľkosťami trénovacej množiny na datasete CIFAR-10 s párovou zväzovacou metódou, **bc**, ktorá mala v tomto teste najvyššiu presnosť.



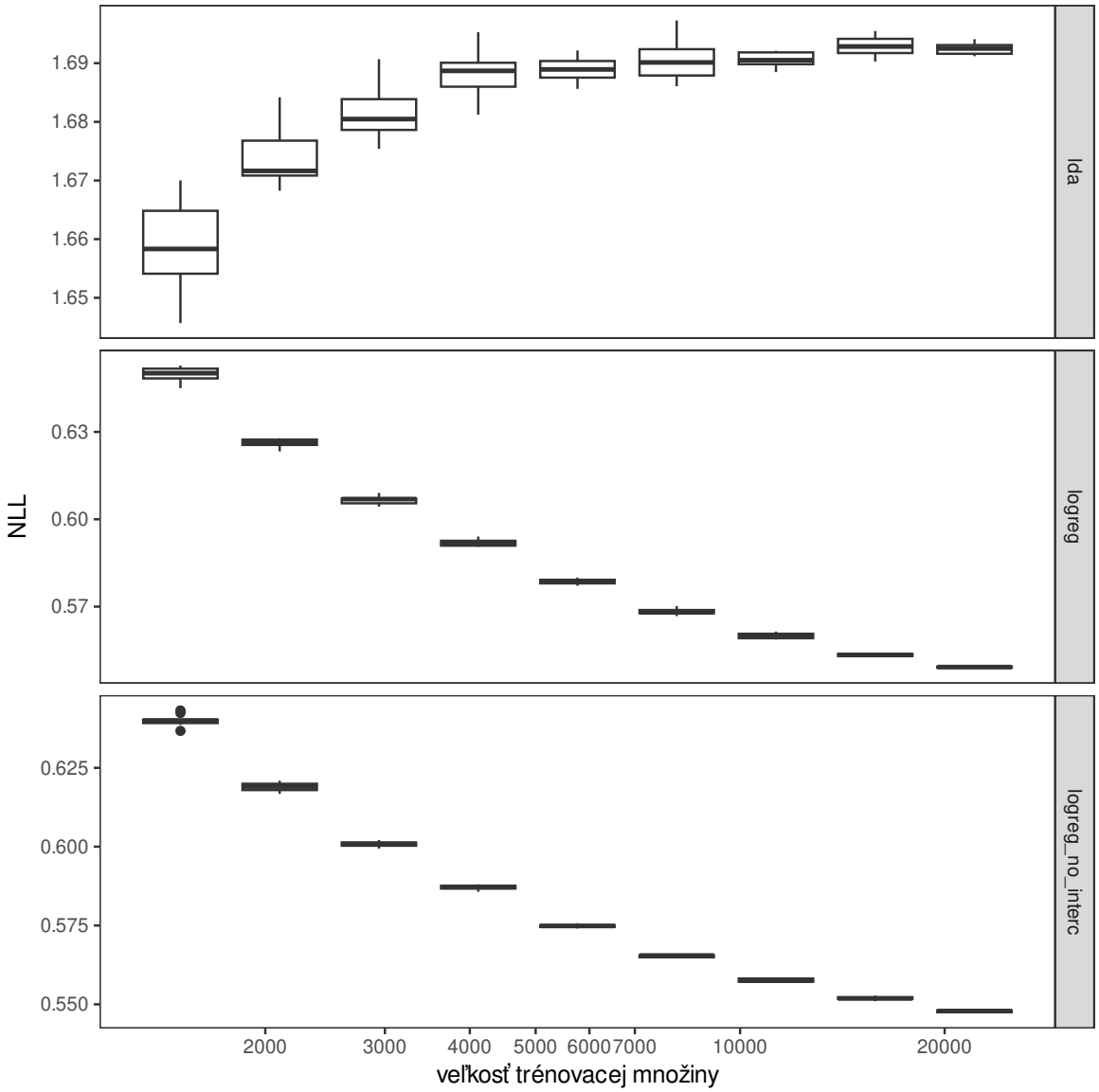
Obrázok 7.5: NLL ansámblu s rôznymi veľkosťami trénovacej množiny na datasete CIFAR-10 s párovou zväzovacou metódou, **bc**, ktorá mala v tomto teste najvyššiu presnosť.



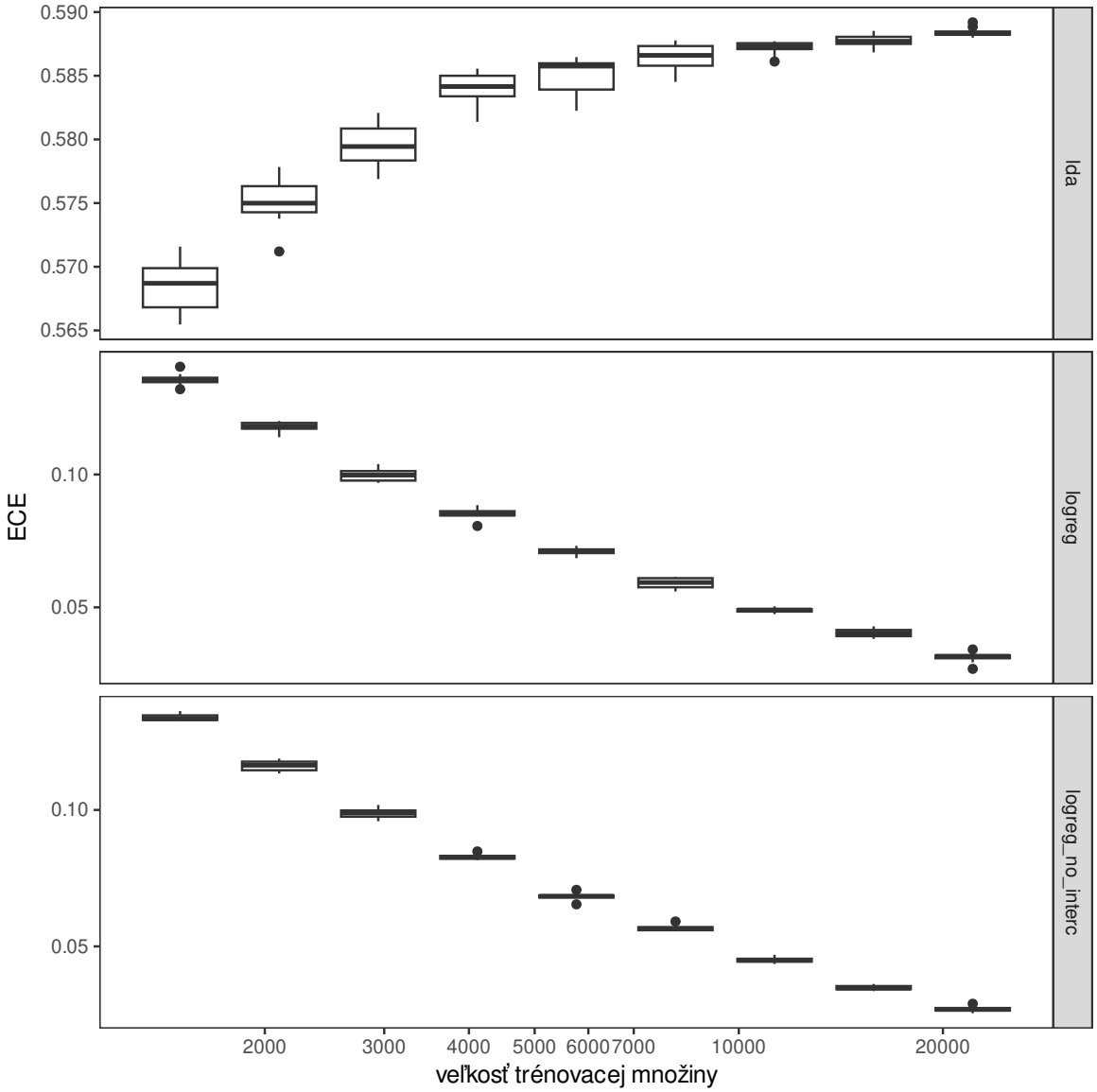
Obrázok 7.6: ECE ansámblu s rôznymi veľkosťami trénovacej množiny na datasete CIFAR-10 s párovou zväzovacou metódou, **bc**, ktorá mala v tomto teste najvyššiu presnosť.



Obrázok 7.7: Presnosť ansámblu s rôznymi veľkosťami trénovacej množiny na datasete CIFAR-100 s párovou zväzovacou metódou, **bc**, ktorá mala v tomto teste najvyššiu presnosť.



Obrázok 7.8: NLL ansámblu s rôznymi veľkosťami trénovacej množiny na datasete CIFAR-100 s párovou zväzovacou metódou, **bc**, ktorá mala v tomto teste najvyššiu presnosť.



Obrázok 7.9: ECE ansámblu s rôznymi veľkosťami trénovacej množiny na datasete CIFAR-100 s párovou zväzovacou metódou, **bc**, ktorá mala v tomto teste najvyššiu presnosť.

7.3.2 Výber trénovacej množiny

Pri voľbe trénovacej množiny pre kombinačné metódy vyvstáva otázka, či trénovanie na rovnakej množine, na ktorej boli natrénované použité neurónové siete má negatívny vplyv na výstupy vytvoreného ansámblu. Za účelom zodpovedania tejto otázky sme navrhli nasledovný experiment.

Trénovaciu množinu datasetov CIFAR-10 a CIFAR-100 sme náhodne rozdelili na polovice tak, aby jednotlivé triedy boli rovnomerne zastúpené v oboch poloviciach. Jednu z týchto polovíc sme označili ako trénovaciu a druhú ako validačnú. Pre CIFAR-10 sme toto rozdelenie realizovali raz, pre CIFAR-100 desať-krát. Na trénovacej polovici sme natrénovali neurónové siete resnet34, densenet121 a xception. Na trénovacej polovici sme tiež natrénovali multinomiálnu regresiu pre extraktor príznakov clip_ViT-B-32, v experimentoch označený ako clip_ViT-B-32_LP.

Následne sme trénovaciu aj validačnú polovicu náhodne rozdelili na časti veľkosti 500 pre CIFAR-10 a 5000 pre CIFAR-100 tak, aby boli jednotlivé triedy v týchto častiach zastúpené rovnomerne. Takto sme získali 50 trénovacích a 50 validačných podmnožín pre oba datasety CIFAR. Na každej z týchto podmnožín sme natrénovali ansámbel s kombinačnými metódami **lda**, **logreg**, **grad_m1**, **grad_m2** a **grad_bc**. Každý z týchto ansámblov sme otestovali s použitím štyroch párových zväzovacích metód **m1**, **m2**, **bc** a **sbt**.

Pri vykonávaní experimentov sme zistili, že preferencia pre validačnú, alebo trénovaciu množinu nie je konzistentná pri vytváraní ansámblu z rôznych podmnožín štyroch dostupných klasifikátorov. Experiment sme preto vykonali na všetkých možných aspoň dvojprvkových podmnožinách štyroch uvedených sietí. Získali sme tak pre každú z 11 podmnožín kombinovaných neurónových sietí 50 ansámblov natrénovaných na trénovacej množine a 50 natrénovaných na validačnej množine. Pre každú z 11 podmnožín kombinovaných neurónových sietí sme potom testovali nulovú hypotézu, že stredná hodnota metriky pre ansámble natrénované na trénovacej množine je rovná strednej hodnote tej istej metriky pre ansámble natrénované na validačnej množine. Túto hypotézu sme testovali na hladine významnosti 1%. Test sme vykonali pre metriky presnosť, NLL a ECE.

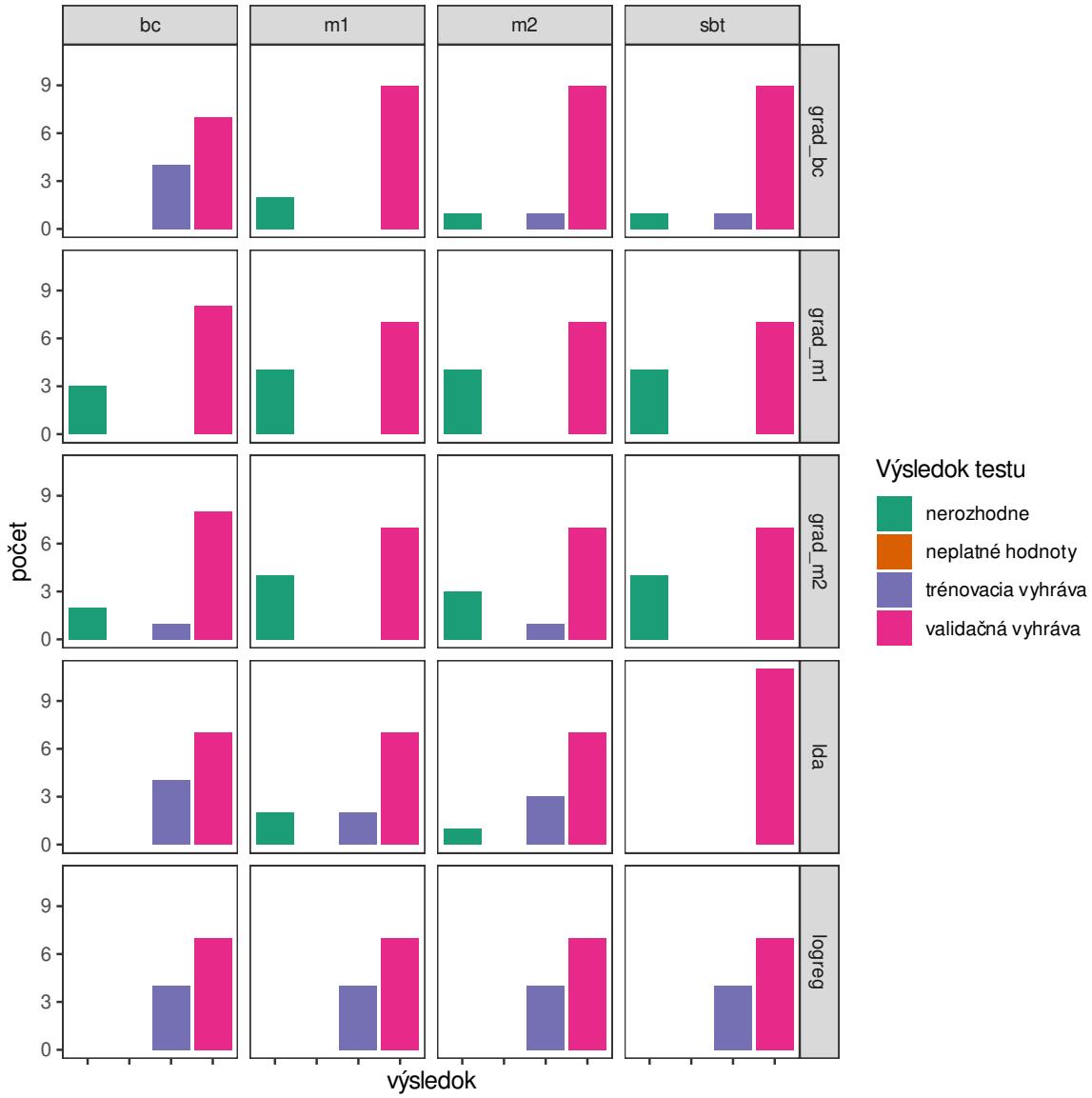
Pre každú z 11 podmnožín kombinovaných klasifikátorov sme výsledok testu vyhodnotili ako:

- nerozhodný, ak sme nulovú hypotézu nezamietli,
- validačná vyhráva, ak sme nulovú hypotézu zamietli v prospech alternatívnej hypotézy, že metrika má lepšiu hodnotu pre ansámble trénované na validačnej množine,
- trénovacia vyhráva, ak sme nulovú hypotézu zamietli v prospech alternatívnej hypotézy, že metrika má lepšiu hodnotu pre ansámble trénované na trénovacej množine,
- neplatné hodnoty, ak sú vo výstupe neplatné hodnoty (nestabilita párovej zväzovacej metódy **sbt**).

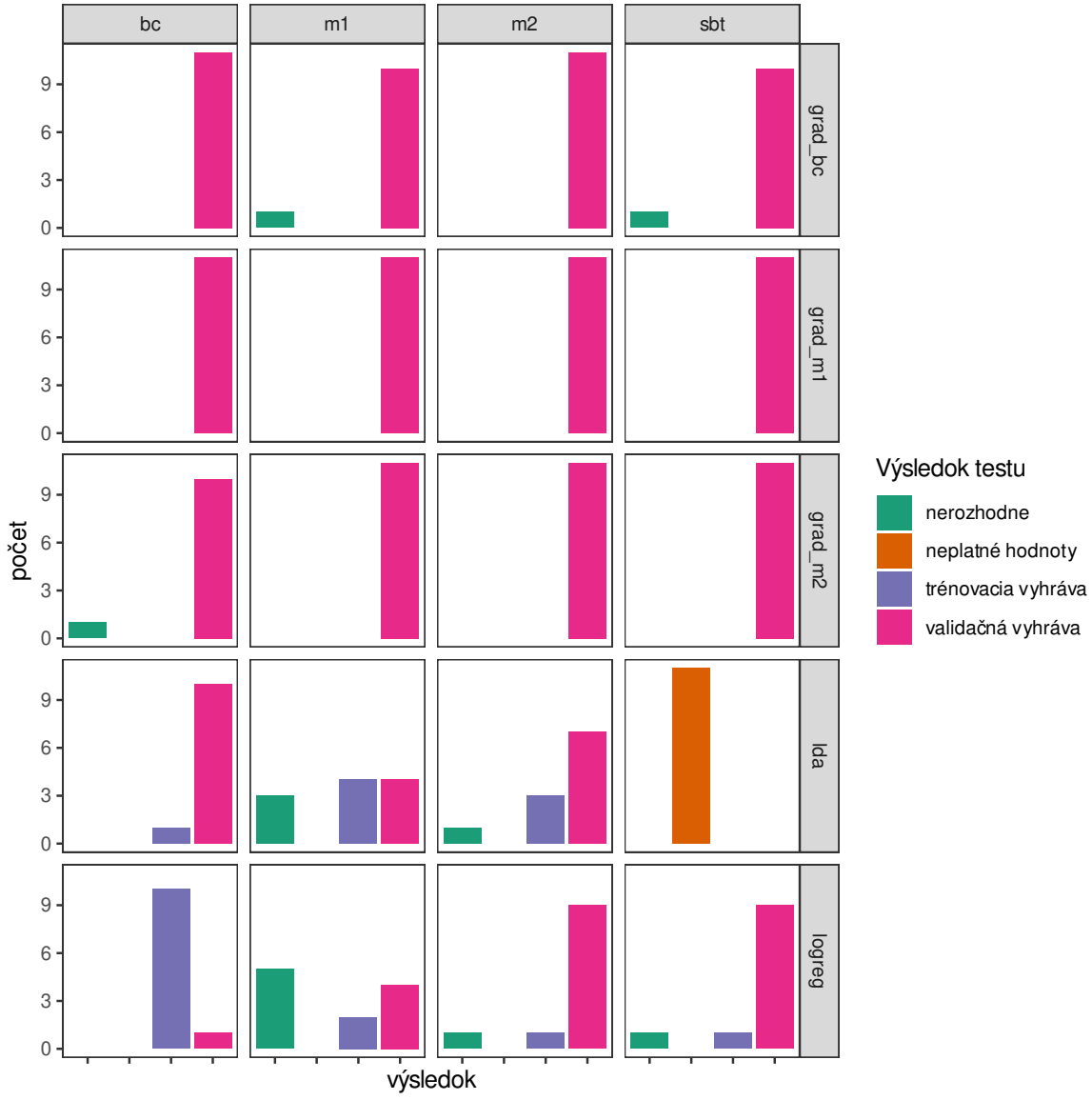
Tento experiment nemodeluje dôverne rozhodnutie či je vhodné pred trénovaním členov ansámblu odobrať z trénovacej množiny osobitnú množinu pre trénovanie kombinačnej metódy. Za účelom presného zodpovedania tejto otázky by sme potrebovali každú z kombinovaných neurónových sietí natrénovať 50-krát na zmenšenej trénovacej množine a 50-krát na úplnej trénovacej množine. Trénovanie na úplnej trénovacej množine by mohlo viesť ku kvalitnejším členom ansámblu a ovplyvniť tak aj jeho výslednú kvalitu. Takýto postup by ale predstavoval neúnosnú výpočtovú záťaž pri tréningu neurónových sietí. Naviac by výsledky takéhoto experimentu mali malú prenositeľnosť na iné datasety s odlišným počtom trénovacích vzoriek, kde by odobratie 50 vzoriek pre každú triedu mohlo predstavovať odlišné relatívne ochudobnenie trénovacej množiny.

CIFAR-10

Výsledky pre CIFAR-10 a metriku presnosť sú zobrazené na obrázku 7.10. Ako môžeme vidieť, pre všetky konfigurácie kombinačných a párových zväzovacích metód vyhráva trénovanie na validačnej množine. Pri gradientových metódach je výsledok, okrem konfigurácie **grad_bc + bc**, jednoznačne v prospech trénovania na validačnej množine. V prípade kombinačnej metódy **logreg** je výsledok o niečo menej jednoznačný. V niekoľkých prípadoch sme tu dosiahli štatisticky významne lepšiu presnosť pri trénovaní na trénovacej množine, vo väčšine prípadov ale stále vyhráva trénovanie na validačnej množine. Výsledky pre metriku ECE sú zobrazené na obrázku 7.11. Vo väčšine prípadov sú výsledky rovnaké ako pre presnosť, výnimkou sú konfigurácie **lda**



Obrázok 7.10: Výsledky štatistických testov pre metriku presnosť na datasete CIFAR-10.



Obrázok 7.11: Výsledky štatistických testov pre metriku ECE na datasete CIFAR-10.

+ m1 a **logreg + bc**. Presnosť ale bola aj v týchto prípadoch lepšia pre trénovanie na validačnej množine.

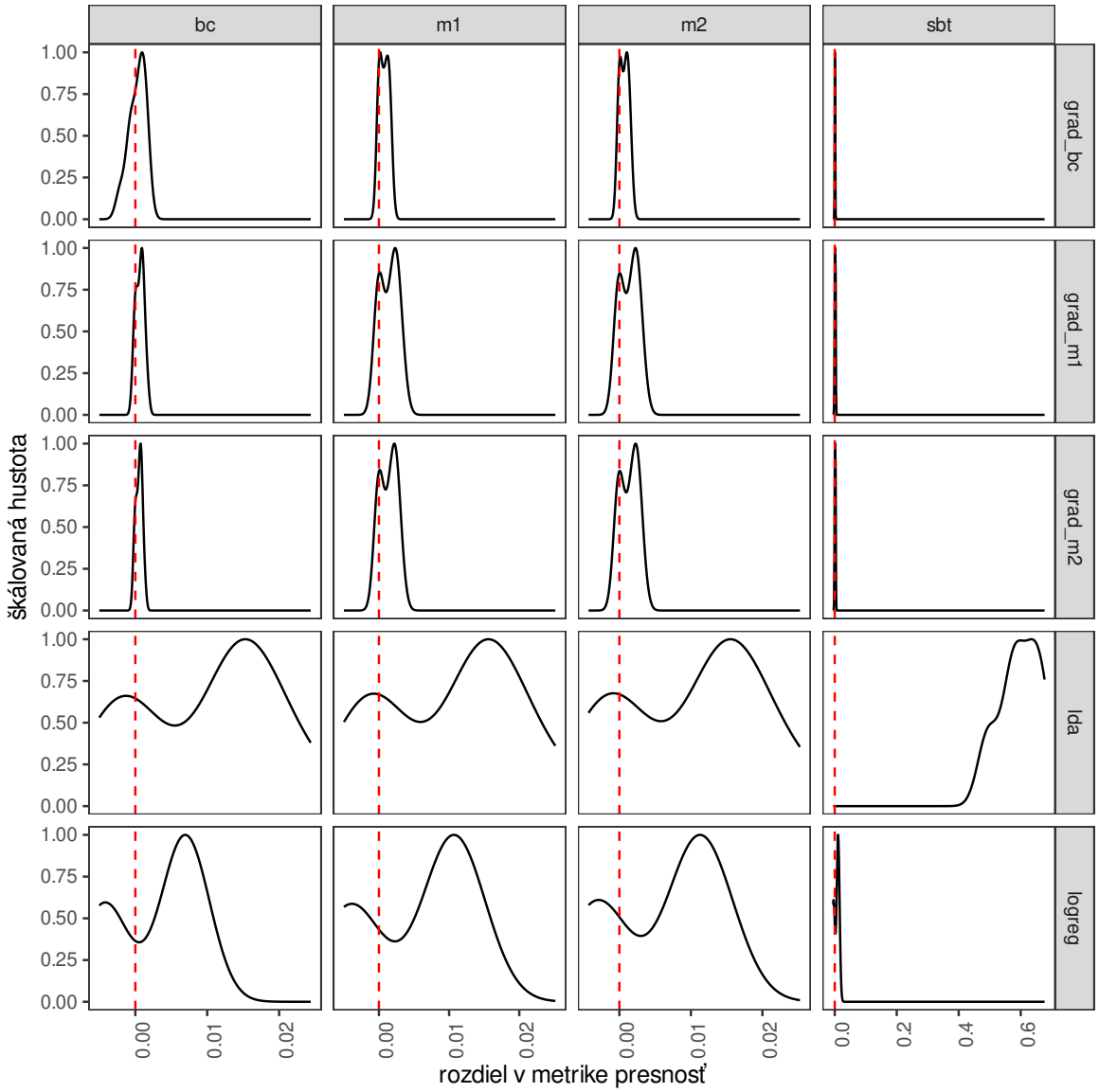
Pri študovaní jednotlivých množín kombinovaných klasifikátorov sme si pre jednotlivé metriky všimli, že v prípadoch vyhodnotených ako "validačná vyhráva" býva medzi priemernými hodnotami metriky pre ansámble trénované na validačnej množine a pre ansámble trénované na trénovacej množine väčší rozdiel ako v prípadoch vyhodnotených ako "trénovacia vyhráva". Túto situáciu vizualizujeme pomocou rozloženia rozdielov v priemernej hodnote metriky pre ansámble trénované na validačnej množine a pre ansámble trénované na trénovacej množine. Priemerná hodnota metriky je počítaná cez jednotlivé trénovacie resp. validačné podmnožiny. Pre metriku presnosť je toto rozloženie zobrazené na obrázku 7.12. Ako je z obrázku zrejmé, záporné rozdiely sú menšie a menej početné ako kladné. Táto nerovnováha je výraznejšia pri kombinačných metódach **lda** a **logreg**. Pre konfiguráciu **lda + sbt** nie sú zobrazené dáta z dôvodu numerickej nestability párovej zväzovacej metódy **sbt** pri trénovaní na trénovacej množine v tejto konfigurácii. Získané výsledky hovoria jednoznačne v prospech trénovania na validačnej množine.

CIFAR-100

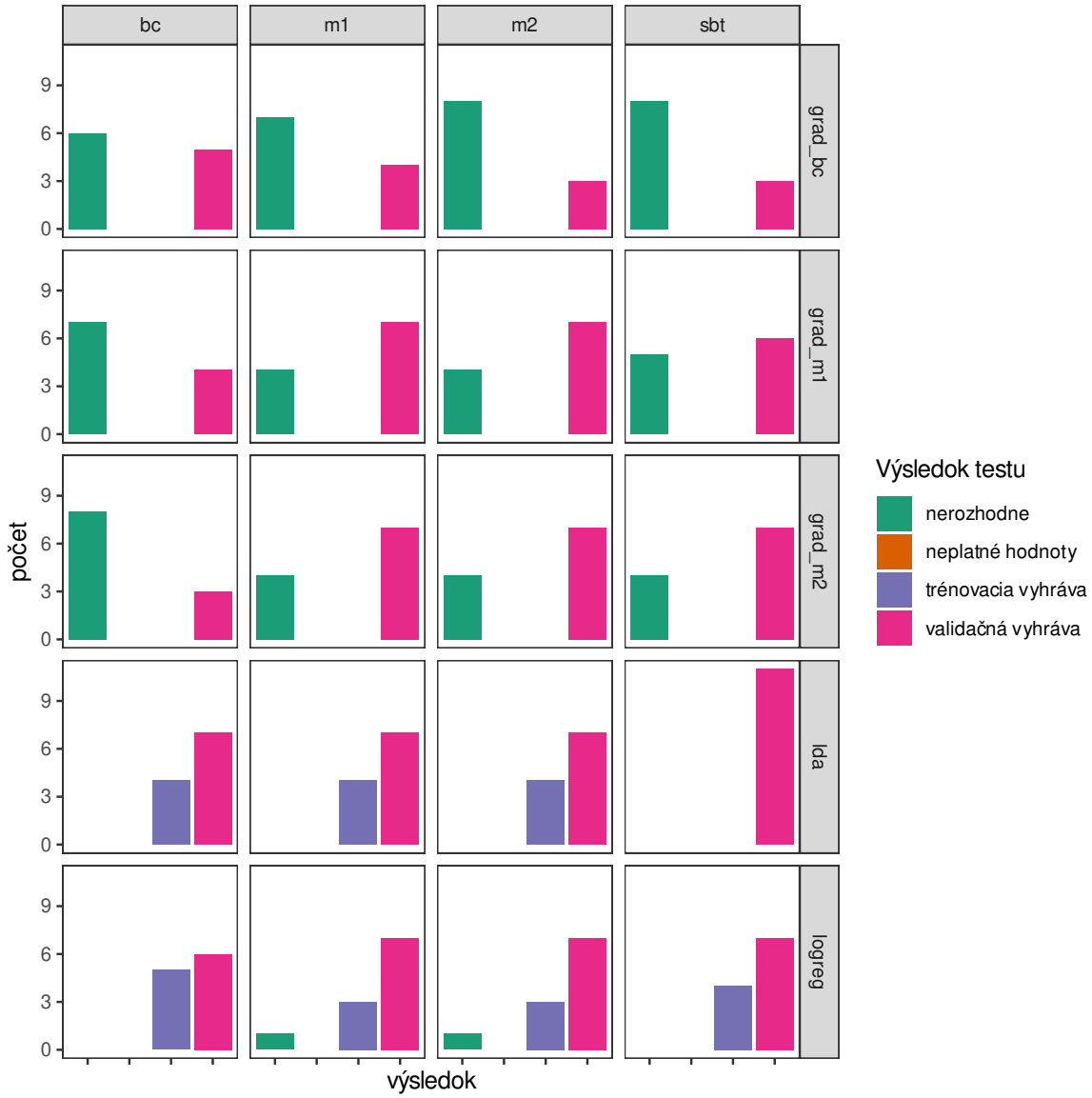
Výsledky štatistických testov pre CIFAR-100 a metriku presnosť sú zobrazené na obrázku 7.13. Ako môžeme vidieť, aj tu pre všetky konfigurácie kombinačných a párových zväzovacích metód vyhráva trénovanie na validačnej množine. Výsledky testov pre metriku ECE sú zobrazené na obrázku 7.14. V prípade datasetu CIFAR-100 hovorí ECE v prospech trénovania na validačnej množine najmä pri kombinačných metódach **lda** a **logreg**. Gradientové metódy sú viac nerozhodné, konfigurácia **grad_m1 + bc** dokonca hovorí v prospech trénovania na trénovacej množine. Aj pre tento dataset sme vykreslili tiež rozdiely v priemeroch jednotlivých metrík pre trénovanie na validačnej množine a pre trénovanie na trénovacej množine. Tieto rozdiely sú zobrazené na obrázku 7.15. Pre dataset CIFAR-100 je z tohto obrázku zrejmá ešte vyššia výhodnosť trénovania na validačnej množine a to najmä pre kombinačné metódy **lda** a **logreg**.

Na základe získaných výsledkov sme sa rozhodli všetky kombinačné metódy trénovať na validačnej množine.

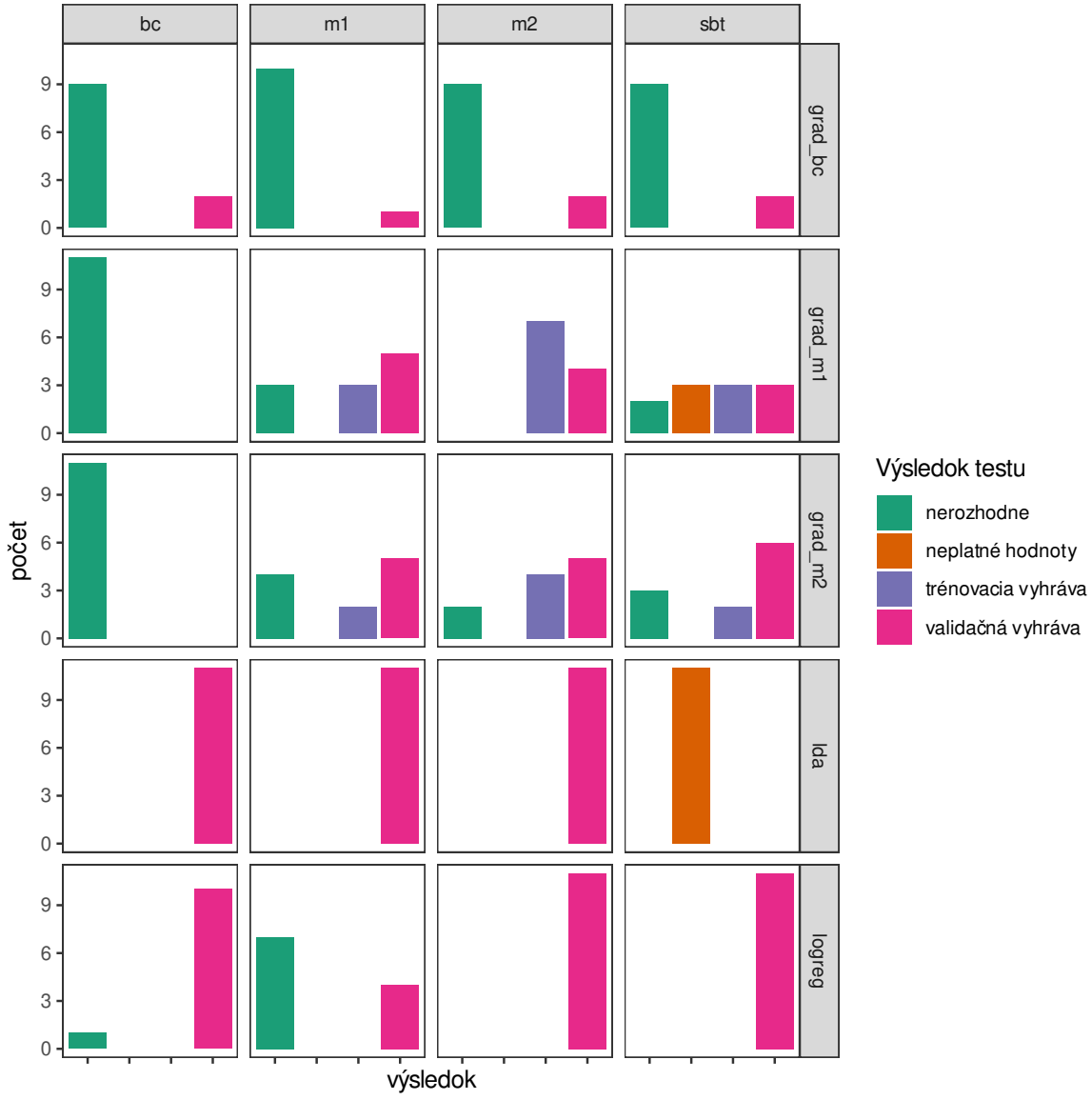
Grafy pre ostatné metriky a tiež individuálne grafy pre jednotlivé podmnožiny



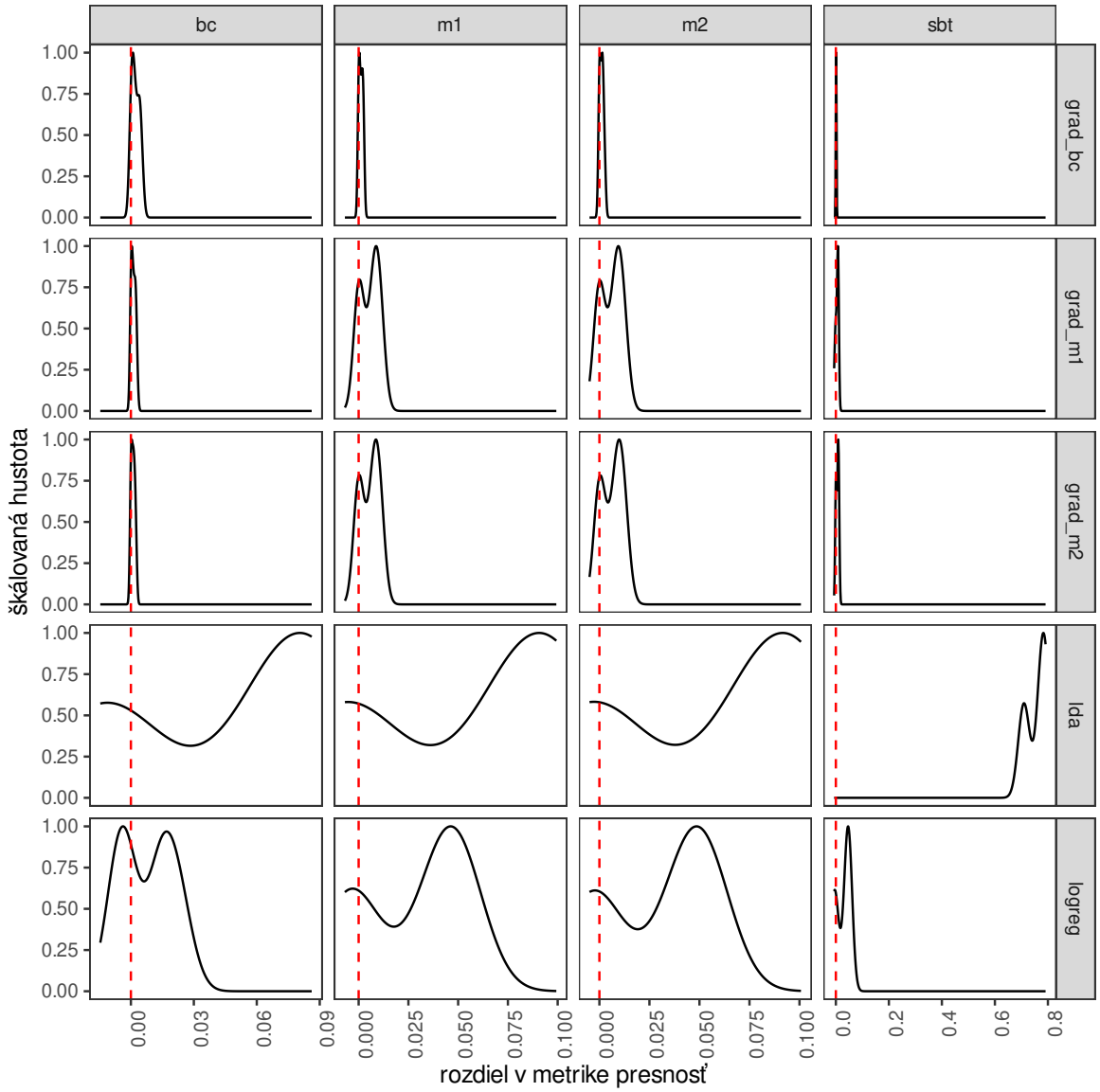
Obrázok 7.12: Distribúcia rozdielov medzi priemernou hodnotou metriky presnosť pre ansámble trénované na validačnej množine a pre ansámble trénované na trénovacej množine. Každý rozdiel prislúcha jednej z 11 kombinácií kombinovaných klasifikátorov. Červenou čiarou je znázornený nulový rozdiel. Záporné rozdiely predstavujú prípady keď priemerná presnosť ansámblov trénovaných na trénovacej množine bola vyššia ako priemerná presnosť ansámblov trénovaných na validačnej množine. Kladné rozdiely predstavujú opačný prípad. Grafy pre párovú zväzovaciu metódu **sbt** sú skreslené z dôvodu numerickej nestability a neplatných hodnôt vo výstupe.



Obrázok 7.13: Výsledky štatistických testov pre metriku presnosť na datasete CIFAR-100.



Obrázok 7.14: Výsledky štatistických testov pre metriku ECE na datasete CIFAR-100.



Obrázok 7.15: Distribúcia rozdielov v metrike presnosť na datasete CIFAR-100. Detailné vysvetlenie je v popise obrázku 7.12.

kombinovaných klasifikátorov sú dostupné v prílohe.

Výsledok experimentov v sekcii: Vhodnú veľkosť trénovacej množiny kombinačných metód sme určili ako 50 vzoriek na jednu triedu. Na základe výsledkov štatistických testov pre presnosť a ECE pri trénovaní kombinačných metód na oddelenej množine vs na rovnakej množine na ktorej boli trénované kombinované klasifikátory sme určili, že trénovanie na oddelenej množine je vhodnejšie.

7.4 Porovnávanie konfigurácií ansámblov

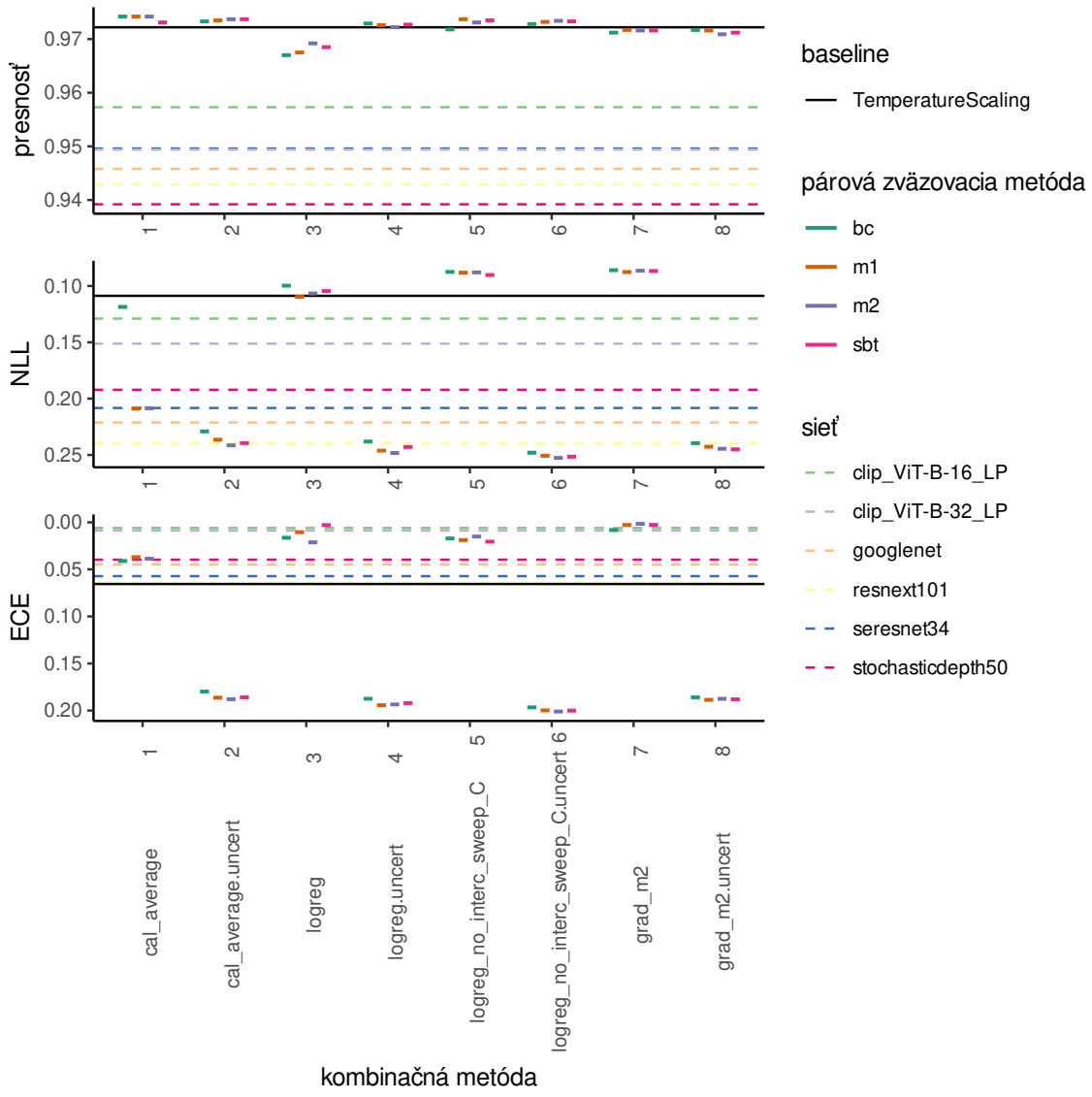
Cieľ experimentov v sekcii: Vytvoriť prehľad kvality všetkých navrhnutých WLE konfigurácií, najmä aby bolo možné pre ďalšie experimenty vybrať len tie, ktoré dávajú sľubné výsledky.

V tejto časti uvádzame výsledky experimentov zameraných na porovnanie ansámblov tvorených kombináciami kombinačných metód a párových zväzovacích metód (označované ako konfigurácie ansámblov). Pre overenie konkurencieschopnosti navrhovaných ansámblov sme pre porovnanie ako baseline implementovali jednoduchú, ale často používanú ansámblovú metódu, ktorá výslednú pravdepodobnostnú predikciu spočíta ako priemer kalibrovaných predikcií jednotlivých klasifikátorov. Pre kalibrovanie jednotlivých klasifikátorov používame metódu teplotné škálovanie (ang. temperature scaling). Teplotné škálovanie pracuje s jedným parametrom, baseline metóda teda používa rovnaký počet parametrov ako je počet kombinovaných klasifikátorov. V experimentoch označujeme túto metódu ako **TemperatureScaling**.

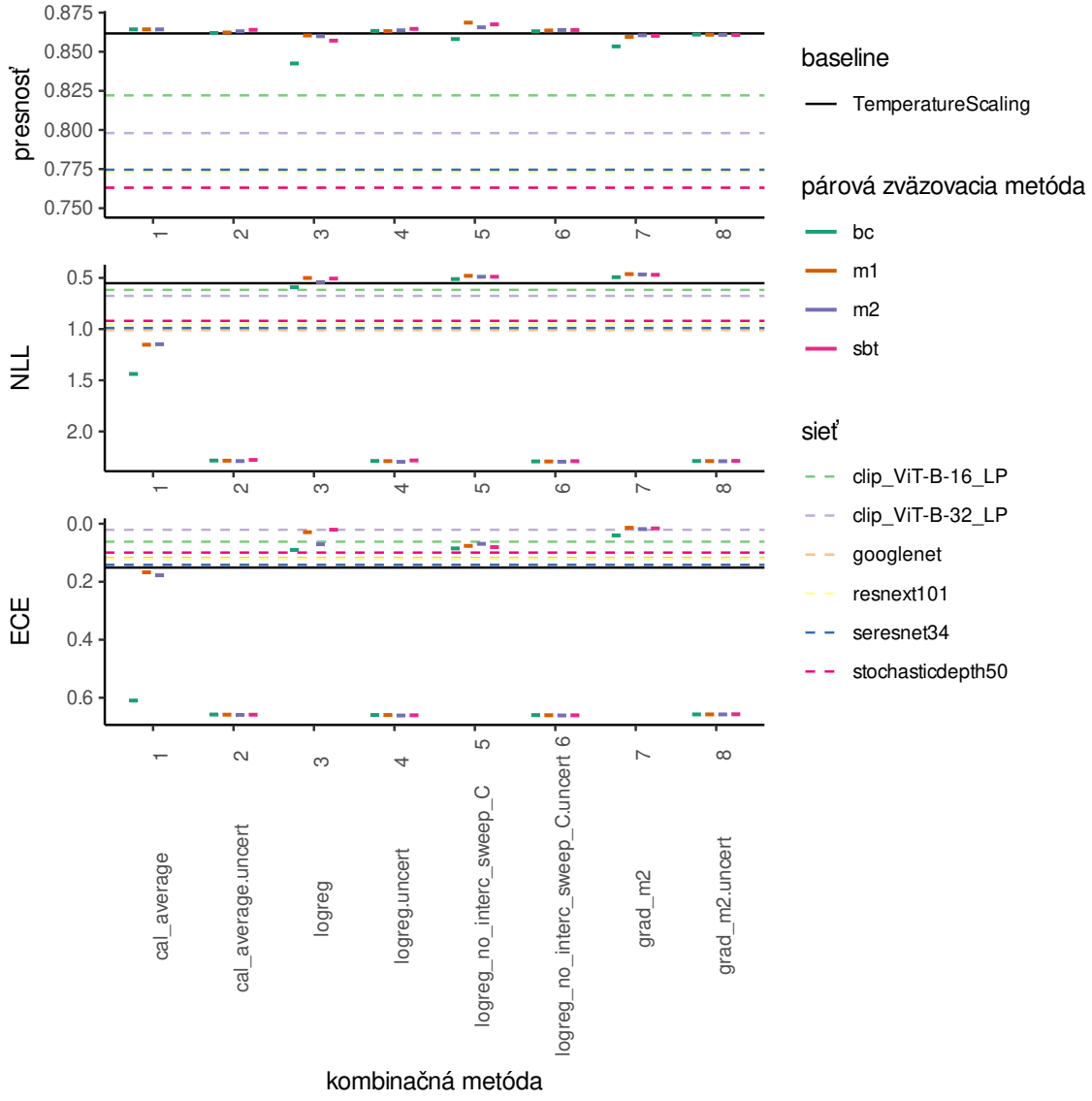
Z dôvodu zlých výsledkov a nesplnených predpokladov normality prediktorov v tomto teste už neuvažujeme kombinačnú metódu **lda**.

V tomto experimente sme použili siete googlenet, stochasticdepth50, resnext101, seresnet34, clip_ViT-B-30_LP a clip_ViT-B-16_LP. Testy sme vykonali na všetkých aspoň dvojprvkových podmnožinách množiny týchto sietí.

Testy sme vyhodnocovali individuálne pre jednotlivé množiny kombinovaných sietí a potom tiež súhrnne pre všetky množiny. Grafy s výsledkami pre individuálnu kombináciu zahŕňajúcu všetky siete sú zobrazené pre CIFAR-10 na obrázku 7.16 a pre CIFAR-100 na obrázku 7.17. Testovali sme 22 kombinačných metód, pre prehľadnosť grafy zobrazujú len tie metódy, ktoré dosiahli najlepšie výsledky.



Obrázok 7.16: Porovnanie vybraných konfigurácií ansámblov kombinujúcich šesť sietí na datasete CIFAR-10. Vertikálne osi pre metriky NLL a ECE sú obrátené - lepšie hodnoty sú hore.



Obrázok 7.17: Porovnanie vybraných konfigurácií ansámblov kombinujúcich šesť sietí na datasete CIFAR-100. Vertikálne osi pre metriky NLL a ECE sú obrátené - lepšie hodnoty sú hore.

Pre oba datasety si môžeme všimnúť, že všetky ansámblové metódy vrátane baseline metódy dosiahli v metrike presnosť zlepšenie oproti všetkým kombinovaným sieťam. (Výnimkou je jedine konfigurácia **cal_average + sbt**, ktorá dosiahla presnosť okolo 50% a nie je ju v grafe vidieť.) Tvrdenie o zlepšení ale neplatí pre metriky NLL a ECE. Najmä pri metrike ECE môžeme u väčšiny ansámblových metód pozorovať zhoršenie oproti kombinovaným sieťam. Z grafov je tiež viditeľný výrazný negatívny vplyv modifikácie kombinačných metód **uncert** na metriky NLL a ECE, aj napriek tomu, že v niektorých prípadoch táto modifikácia prináša zlepšenie presnosti. U oboch datasetov sa v metrike ECE podarilo dosiahnuť zlepšenie oproti dobre kalibrovaným modelom **clip** len niekoľkým konfiguráciám využívajúcim kombinačné metódy **grad**, alebo kombinačnú metódu **logreg**. Pre oba datasety si môžeme všimnúť dobré výsledky v metrike presnosť pre jednoduchú kombinačnú metódu **cal_average** a jej modifikáciu **cal_average.uncert**. Tieto dobré výsledky sa ale neprenášajú do metrík NLL a ECE. Za zmienku stoja tiež dobré výsledky bezparametrickej kombinačnej metódy **average** pre metriky NLL a ECE. Podobné grafy sme vytvorili pre všetky testované množiny kombinovaných sietí a sú dostupné v prílohe.

Vyhodnotenie kvality jednotlivých kombinačných metód sme vykonali na základe všetkých testovaných podmnožín kombinovaných klasifikátorov. Hodnotu metriky dosiahnutú ansámblovou predikciou sme porovnávali s najlepšou hodnotou danej metriky medzi kombinovanými klasifikátormi a tiež s priemernou hodnotou danej metriky medzi kombinovanými klasifikátormi. Tieto rozdiely sme vypočítali pre všetky testované množiny kombinovaných klasifikátorov. Rozdiely sme počítali tak, aby kladná hodnota znamenala zlepšenie oproti kombinovaným klasifikátorom. Priemerné hodnoty týchto rozdielov sú v nasledujúcich tabuľkách označené ako "Zlepšenie metriky oproti priemeru" a "Zlepšenie metriky oproti najlepšej". Pre všetky sledované metriky uvádzame 20 najúspešnejších konfigurácií ansámblov. Pre dataset CIFAR-10 sú výsledky pre presnosť zobrazené v tabuľke 7.1, pre NLL v tabuľke 7.2 a pre ECE v tabuľke 7.3. Pre dataset CIFAR-100 sú výsledky pre presnosť v tabuľke 7.4, pre NLL v tabuľke 7.5 a pre ECE v tabuľke 7.6.

Pre CIFAR-10 bolo najväčšie zvýšenie presnosti dosiahnuté kombinačnou metódou **cal_average** a jej modifikáciou **cal_average.uncert**. Za ňou nasledovali rôzne verzie kombinačnej metódy **logreg**, niektoré z nich tiež s rozšírením **uncert**. Medzi prvými

Tabuľka 7.1: CIFAR-10. Ansámblové konfigurácie s najvyšším priemerným zvýšením presnosti oproti kombinovaným sieťam.

Poradie	**Metóda**	**Zlepšenie presnosti oproti priemeru**	**Zlepšenie presnosti oproti najlepšej**
1	cal_average + bc	0.0214	0.0153
2	cal_average + m1	0.0214	0.0153
3	cal_average + m2	0.0214	0.0153
4	cal_average + sbt	0.0214	0.0153
5	cal_average.uncert + m2	0.0213	0.0152
6	cal_average.uncert + sbt	0.0213	0.0152
7	cal_average.uncert + m1	0.0212	0.0151
8	cal_average.uncert + bc	0.0210	0.0150
9	logreg_no_interc.uncert + bc	0.0205	0.0144
10	logreg_no_interc.uncert + m1	0.0204	0.0143
11	logreg_no_interc_sweep_C.uncert + m1	0.0203	0.0143
12	logreg_no_interc + bc	0.0203	0.0143
13	logreg_no_interc_sweep_C.uncert + sbt	0.0203	0.0143
14	logreg_no_interc_sweep_C.uncert + m2	0.0203	0.0142
15	logreg_sweep_C.uncert + m2	0.0203	0.0142
16	logreg_no_interc_sweep_C.uncert + bc	0.0203	0.0142
17	logreg_sweep_C.uncert + sbt	0.0202	0.0141
18	logreg_no_interc_sweep_C + m1	0.0201	0.0140
19	logreg.uncert + bc	0.0201	0.0140
20	logreg_sweep_C.uncert + bc	0.0200	0.0140
31	**baseline - TemperatureScaling**	**0.0196**	**0.0135**

Tabuľka 7.2: CIFAR-10. Ansámblové konfigurácie s najvyšším priemerným znížením NLL oproti kombinovaným sieťam. Vyššie číslo znamená väčšie zníženie a teda väčšie zlepšenie.

Poradie	**Metóda**	**Zníženie NLL oproti priemeru**	**Zníženie NLL oproti najlepšej**
1	logreg_no_interc + bc	0.0901	0.0482
2	grad_bc + bc	0.0872	0.0453
3	grad_m2 + m2	0.0870	0.0451
4	grad_m2 + sbt	0.0869	0.0450
5	grad_m2 + bc	0.0865	0.0446
6	grad_bc + sbt	0.0865	0.0446
7	grad_bc + m1	0.0856	0.0437
8	grad_bc + m2	0.0856	0.0437
9	grad_m2 + m1	0.0853	0.0434
10	grad_m1 + bc	0.0849	0.0430
11	logreg_sweep_C + m2	0.0847	0.0428
12	grad_m1 + m2	0.0845	0.0426
13	average + bc	0.0840	0.0421
14	average + sbt	0.0830	0.0411
15	average + m2	0.0830	0.0411
16	average + m1	0.0830	0.0411
17	grad_m1 + sbt	0.0826	0.0407
18	grad_m1 + m1	0.0823	0.0404
19	logreg_no_interc + m1	0.0798	0.0379
20	logreg_no_interc_sweep_C + m2	0.0797	0.0378
30	**baseline - TemperatureScaling**	**0.0719**	**0.0300**

Tabuľka 7.3: CIFAR-10. Ansámblové konfigurácie s najvyšším priemerným znížením ECE oproti kombinovaným sieťam. Vyššie číslo znamená väčšie zníženie. Záporné čísla predstavujú zvýšenie, teda zhoršenie.

Poradie	**Metóda**	**Zníženie ECE oproti priemeru**	**Zníženie ECE oproti najlepšej**
1	logreg + sbt	0.0315	0.0085
2	logreg_no_interc + sbt	0.0305	0.0075
3	logreg + m1	0.0298	0.0068
4	grad_m1 + m1	0.0289	0.0059
5	grad_m1 + sbt	0.0286	0.0057
6	grad_m1 + m2	0.0277	0.0048
7	logreg_no_interc + m1	0.0277	0.0047
8	grad_m2 + m2	0.0266	0.0036
9	grad_m2 + sbt	0.0260	0.0030
10	grad_m2 + m1	0.0258	0.0028
11	logreg_no_interc + bc	0.0253	0.0023
12	logreg_no_interc + m2	0.0253	0.0023
13	grad_bc + bc	0.0225	-0.0005
14	grad_bc + m1	0.0216	-0.0014
15	logreg + bc	0.0215	-0.0015
16	logreg_sweep_C + m2	0.0215	-0.0015
17	grad_bc + sbt	0.0214	-0.0016
18	grad_bc + m2	0.0210	-0.0020
19	grad_m2 + bc	0.0207	-0.0023
20	prob_average + bc	0.0204	-0.0026
42	**baseline - TemperatureScaling**	**-0.0160**	**-0.0390**

sa tiež nachádzajú metódy s ladením regularizačného parametra **sweep_C**. Baseline metóda sa nachádza na 31. mieste a nad ňou sa podarilo umiestniť väčšine konfigurácií využívajúcich logistickú regresiu. Pod baseline metódou skončili gradientové metódy a tiež metódy kombinujúce pravdepodobnosti **prob_average** a **cal_prob_average**. V tabuľkách 7.2 a 7.3 pre metriky NLL a ECE vidíme, že kombinačnej metóde **cal_average** úspešnej v zlepšení presnosti sa v týchto metrikách nepodarilo umiestniť medzi prvými. Najlepšie výsledky pre metriky NLL a ECE dosiahli gradientové kombinačné metódy **grad** a niektoré kombinačné metódy logistickej regresie **logreg**. Metódy využívajúce modifikáciu **uncert** dosiahli v metrikách NLL a ECE horšie výsledky ako priemer kombinovaných klasifikátorov. V metrike ECE dosiahla horšie výsledky ako priemer kombinovaných klasifikátorov tiež baseline ansámblová metóda. So zohľadnením všetkých metrík dosahovali najlepšie výsledky metódy využivajúce logistickú regresiu.

Pri datasete CIFAR-100 môžeme pozorovať podobné správanie kombinačných metód ako pri datasete CIFAR-10. V zlepšení presnosti sú znovu dominantné kombinačné metódy **cal_average** a metódy zo skupiny **logreg**. V zlepšení metrík NLL a ECE si najlepšie počínajú gradientové kombinačné metódy a kombinačné metódy logistickej regresie. Pri datasete CIFAR-100 dosahuje dobré zlepšenie v metrikách NLL a ECE tiež bezparametrická kombinačná metóda **average**.

Pre lepšiu prehľadnosť sme výsledky vykreslili aj vo forme grafov. Zobrazujeme kombinačné metódy, ktoré sa umiestnili v zlepšení v niektorej metrike medzi prvými troma. Graf pre dataset CIFAR-10 je na obrázku 7.18 a pre CIFAR-100 na obrázku 7.19. Z oboch grafov sú zrejmé najmä potiaže s metrikou ECE, kde väčšina metód vykazuje horšie výsledky ako najlepšia kombinovaná sieť. Kalibráciu si dokázali zachovať konfigurácie využívajúce kombinačné metódy **logreg**, **logreg_no_interc** a metódy **grad**, v presnosti tieto metódy ale zaostávajú za baseline metódou. Je však nutné podotknúť, že baseline metóda kalibráciu výrazne zhoršuje. Z grafu 7.18 si tiež môžeme všimnúť dobré výsledky pre všetky metriky na datasete CIFAR-10 dosahované párovou zväzovacou metódou **bc** v kombinácii s kombinačnými metódami **cal_average** a **logreg_no_interc**. Tieto výsledky sa však neprenášajú na dataset CIFAR-100, kde najmä pre metriky NLL a ECE párová zväzovacia metóda **bc** v rovnakých konfiguráciách dosahuje výrazne horšie výsledky ako ostatné párové zväzovacie metódy.

Zo zobrazených boxplotov pre metriku presnosť je vidieť, že pre oba datasety CIFAR

Tabuľka 7.4: CIFAR-100. Ansámblové konfigurácie s najvyšším priemerným zvýšením presnosti oproti kombinovaným sieťam.

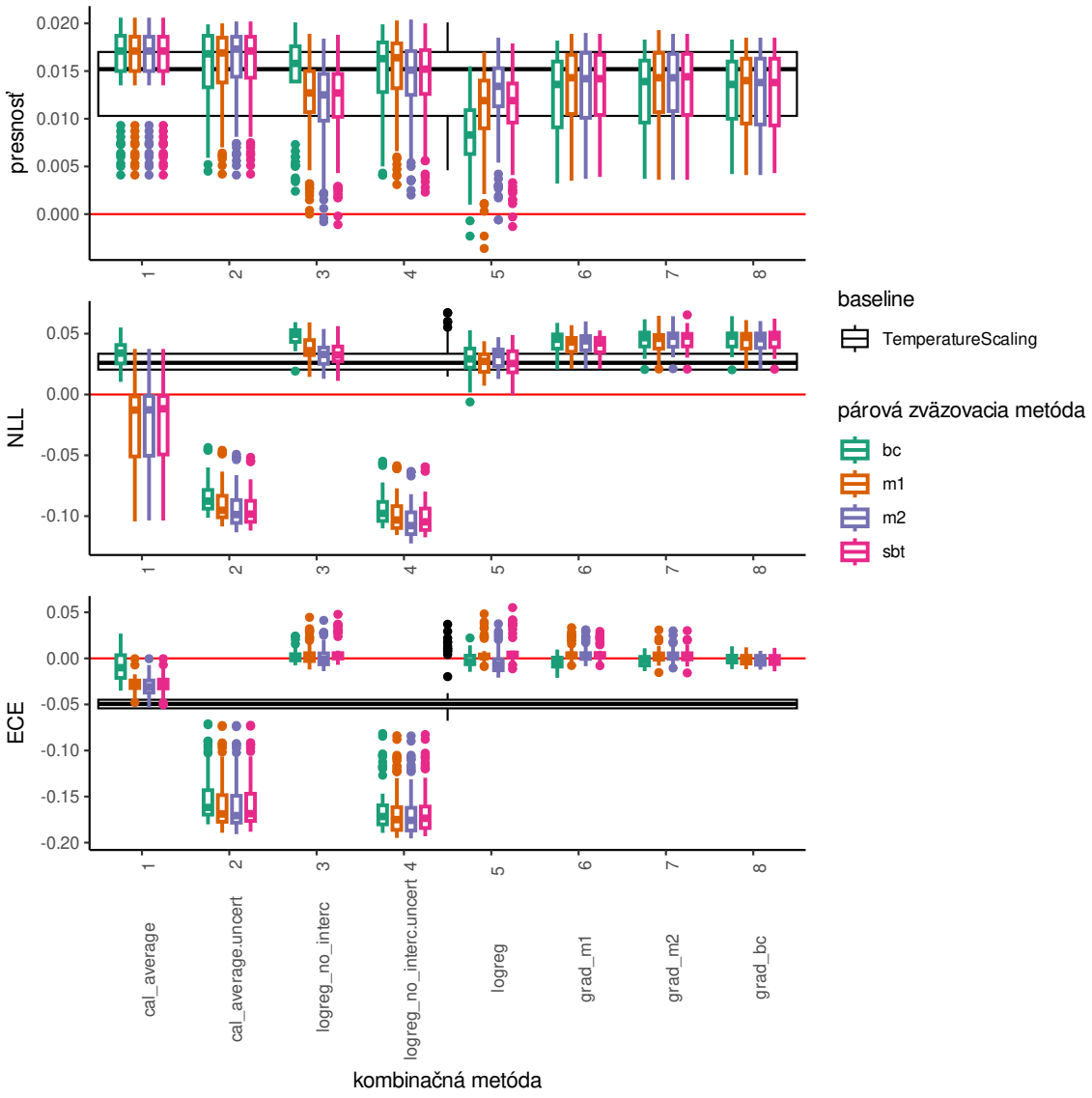
Poradie	**Metóda**	**Zlepšenie presnosti oproti priemeru**	**Zlepšenie presnosti oproti najlepšej**
1	logreg_no_interc_sweep_C + sbt	0.0653	0.0413
2	logreg_no_interc_sweep_C + m1	0.0651	0.0411
3	cal_average + m1	0.0646	0.0407
4	cal_average + m2	0.0646	0.0407
5	cal_average + bc	0.0646	0.0407
6	logreg_no_interc_sweep_C + m2	0.0644	0.0404
7	logreg_sweep_C + m2	0.0640	0.0400
8	logreg_sweep_C + sbt	0.0635	0.0396
9	logreg_no_interc.uncert + sbt	0.0635	0.0396
10	logreg_no_interc.uncert + m2	0.0633	0.0394
11	logreg.uncert + sbt	0.0633	0.0393
12	logreg.uncert + m2	0.0632	0.0392
13	cal_average.uncert + sbt	0.0629	0.0390
14	logreg_no_interc_sweep_C.uncert + sbt	0.0629	0.0389
15	logreg_no_interc_sweep_C.uncert + m2	0.0628	0.0389
16	logreg_no_interc + m1	0.0627	0.0388
17	logreg_sweep_C.uncert + m2	0.0627	0.0388
18	logreg_no_interc.uncert + m1	0.0627	0.0388
19	logreg_sweep_C.uncert + sbt	0.0627	0.0388
20	cal_average.uncert + m2	0.0626	0.0387
57	**baseline - TemperatureScaling**	**0.0611**	**0.0372**

Tabuľka 7.5: CIFAR-100. Ansámblové konfigurácie s najvyšším priemerným znížením NLL oproti kombinovaným sieťam. Vyššie číslo znamená väčšie zníženie.

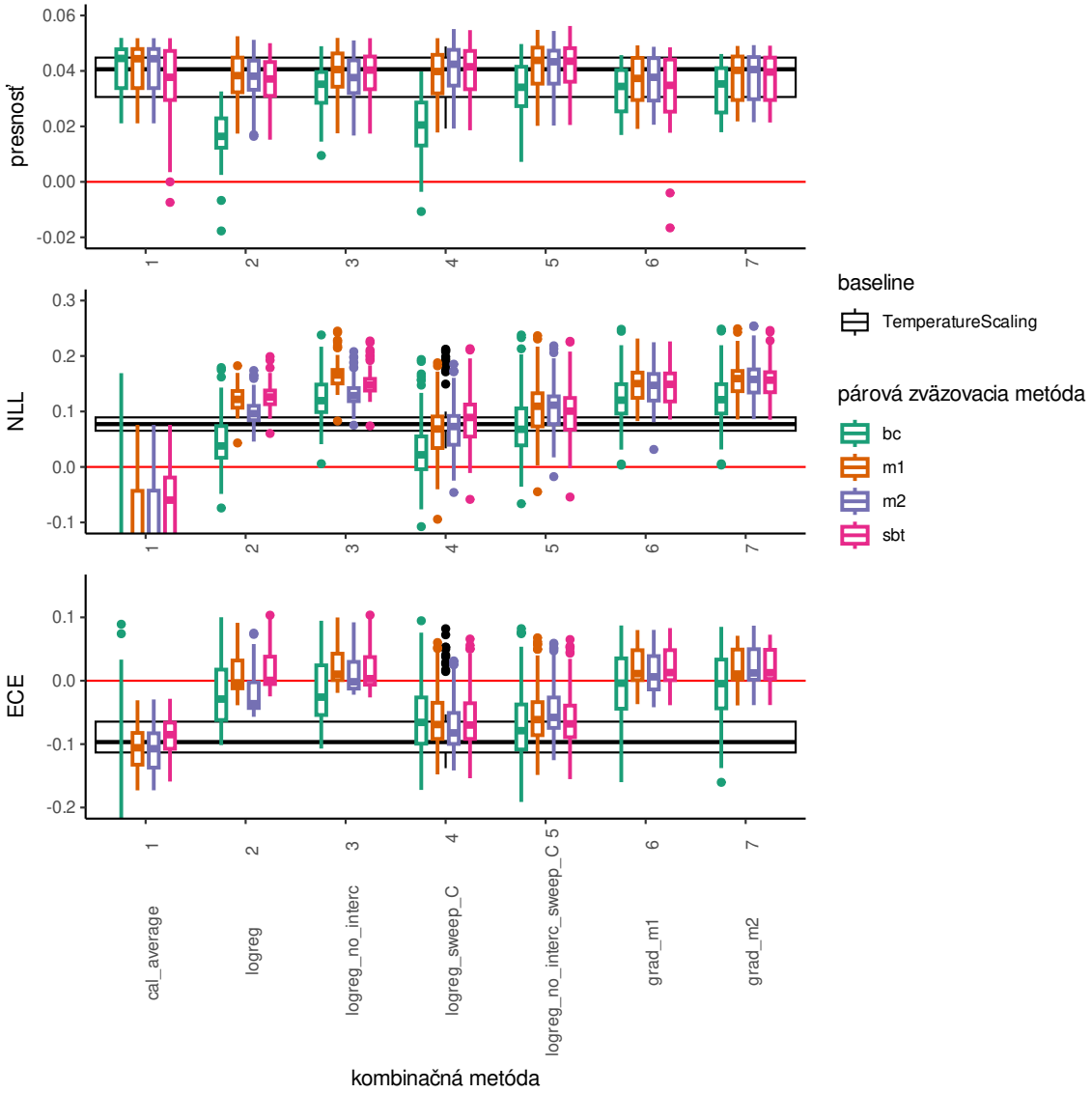
Poradie	**Metóda**	**Zníženie NLL oproti priemeru**	**Zníženie NLL oproti najlepšej**
1	logreg_no_interc + m1	0.3352	0.1687
2	grad_m2 + m2	0.3253	0.1588
3	grad_m2 + m1	0.3241	0.1577
4	grad_m2 + sbt	0.3216	0.1552
5	logreg_no_interc + sbt	0.3194	0.1529
6	grad_m1 + m1	0.3158	0.1493
7	grad_m1 + m2	0.3102	0.1437
8	grad_bc + m2	0.3044	0.1380
9	grad_bc + sbt	0.3044	0.1379
10	grad_bc + m1	0.3038	0.1374
11	logreg_no_interc + m2	0.3017	0.1353
12	average + sbt	0.3016	0.1352
13	average + m2	0.3016	0.1352
14	average + m1	0.3016	0.1352
15	grad_bc + bc	0.2990	0.1325
16	logreg + sbt	0.2941	0.1277
17	logreg_no_interc + bc	0.2917	0.1253
18	grad_m2 + bc	0.2904	0.1240
19	grad_m1 + bc	0.2902	0.1238
20	logreg + m1	0.2893	0.1229
26	**baseline - TemperatureScaling**	**0.2597**	**0.0933**

Tabuľka 7.6: CIFAR-100. Ansámblové konfigurácie s najvyšším priemerným znížením ECE oproti kombinovaným sieťam. Vyššie číslo znamená väčšie zníženie. Záporné čísla predstavujú zvýšenie, teda zhoršenie.

Poradie	**Metóda**	**Zníženie ECE oproti priemeru**	**Zníženie ECE oproti najlepšej**
1	logreg_no_interc + m1	0.0725	0.0269
2	grad_m2 + m2	0.0700	0.0243
3	grad_m1 + m1	0.0693	0.0237
4	grad_m2 + sbt	0.0689	0.0233
5	logreg_no_interc + sbt	0.0682	0.0226
6	grad_m2 + m1	0.0681	0.0224
7	logreg + sbt	0.0674	0.0217
8	grad_m1 + m2	0.0613	0.0156
9	logreg_no_interc + m2	0.0600	0.0144
10	grad_bc + m2	0.0501	0.0045
11	grad_bc + sbt	0.0500	0.0043
12	grad_bc + m1	0.0499	0.0043
13	average + sbt	0.0489	0.0033
14	average + m1	0.0489	0.0033
15	average + m2	0.0489	0.0033
16	logreg + m1	0.0484	0.0028
17	grad_bc + bc	0.0435	-0.0022
18	average + bc	0.0372	-0.0084
19	grad_m1 + bc	0.0362	-0.0094
20	grad_m2 + bc	0.0361	-0.0096
36	**baseline - TemperatureScaling**	**-0.0261**	**-0.0717**



Obrázok 7.18: CIFAR-10. Graf zlepšení dosiahnutých testovanými ansámblovými konfiguráciami v sledovaných metrikách oproti najlepšej kombinovanej sieti. Červenou vodorovnou čiarou je znázornené nulové zlepšenie. Čierny boxplot v pozadí vyjadruje zlepšenie dosiahnuté baseline metódou.



Obrázok 7.19: CIFAR-100. Graf zlepšení dosiahnutých testovanými ansámblovými konfiguráciami v sledovaných metrikách oproti najlepšej kombinovanej sieti. Červenou vodorovnou čiarou je znázornené nulové zlepšenie. Čierny boxplot v pozadí vyjadruje zlepšenie dosiahnuté baseline metódou. Zobrazenie na zvislej osi je pre lepšiu detailnosť zdola obmedzené, niektoré konfigurácie dosahujúce zlé výsledky preto nemusí byť vidieť.

rozdiely medzi WLE ansámblami a baseline ansámblom nie sú výrazné. Rozdiely pre metriky NLL a ECE sú v niektorých prípadoch výraznejšie. Na tomto mieste jednotlivé metódy nevyhodnocujeme pomocou štatistických testov, keďže stále sme len v procese výberu vhodných konfigurácií WLE ansámblu.

Výsledok experimentov v sekcii: Pokiaľ nám ide hlavne o presnosť, na datasetoch CIFAR dosahuje najlepšie výsledky kombinačná metóda **cal_average**. Ak nám ide o všetky tri sledované metriky, tak ako najlepšia voľba vychádzajú kombinačné metódy založené na logistickej regresii, najmä **logreg_no_interc**. V metrikách ECE a NLL sa na popredných miestach umiestňujú okrem metód založených na logistickej regresii aj gradientové metódy, najmä **grad_m2**, tie ale majú horšiu presnosť.

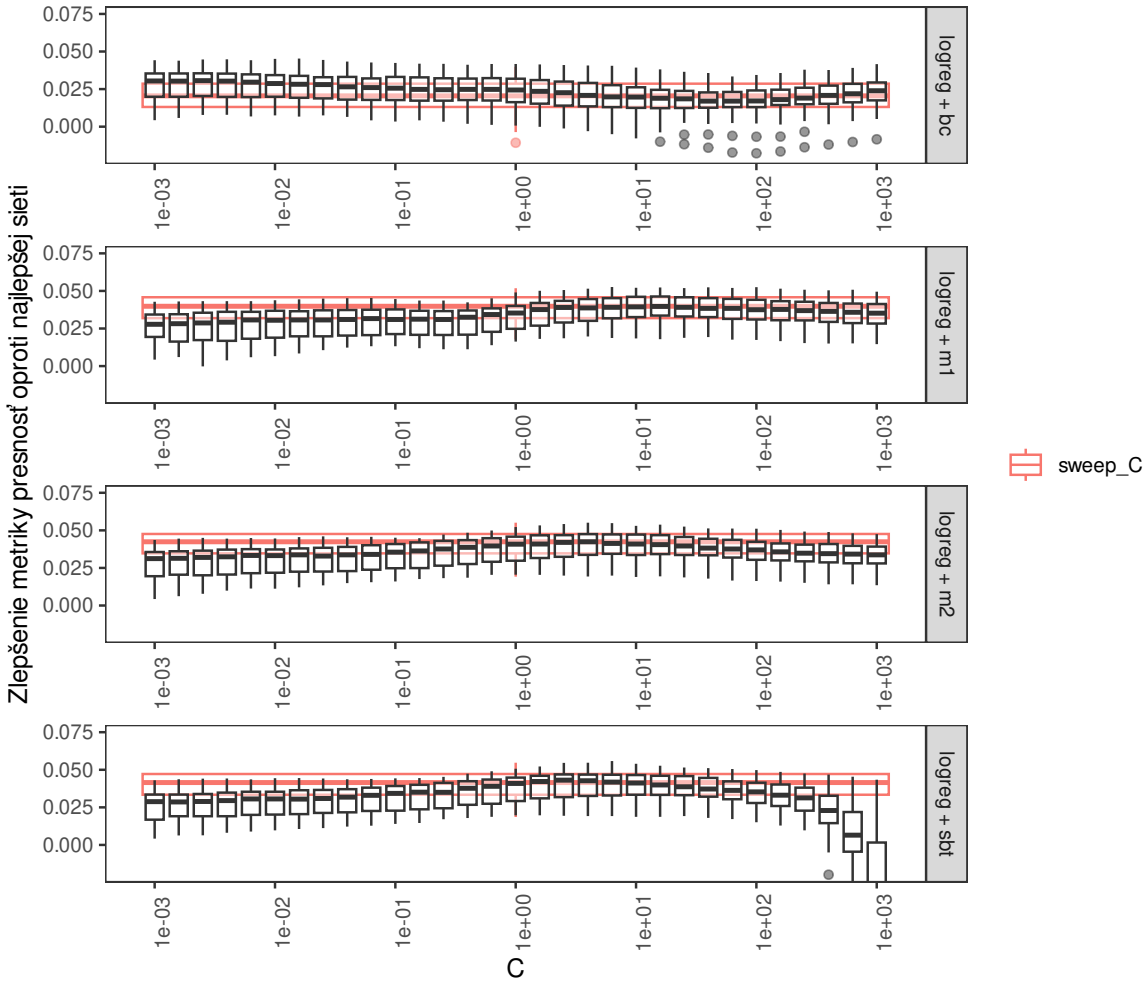
7.5 Regularizácia v kombinačných metódach

Cieľ experimentov v sekcii: Určiť vhodné hodnoty regularizačného koeficientu pre kombinačné metódy založené na logistickej regresii a pre gradientové kombinačné metódy.

7.5.1 Regularizácia v kombinačnej metóde logistická regresia

Kombinačné metódy založené na logistickej regresii a využívajúce ladenie koeficientu C miery regularizácie sú v experimentoch označené príponou **sweep_C**. Tieto metódy sa na oboch datasetoch CIFAR a vo všetkých metrikách vyskytujú na popredných miestach. Trénovanie takýchto kombinačných metód je ale náročné z dôvodu individuálneho ladenia parametra C pre každú dvojicu tried. Preto sme sa rozhodli vykonať experiment zameraný na nájdenie jednotnej hodnoty parametra C pre všetky dvojice tried.

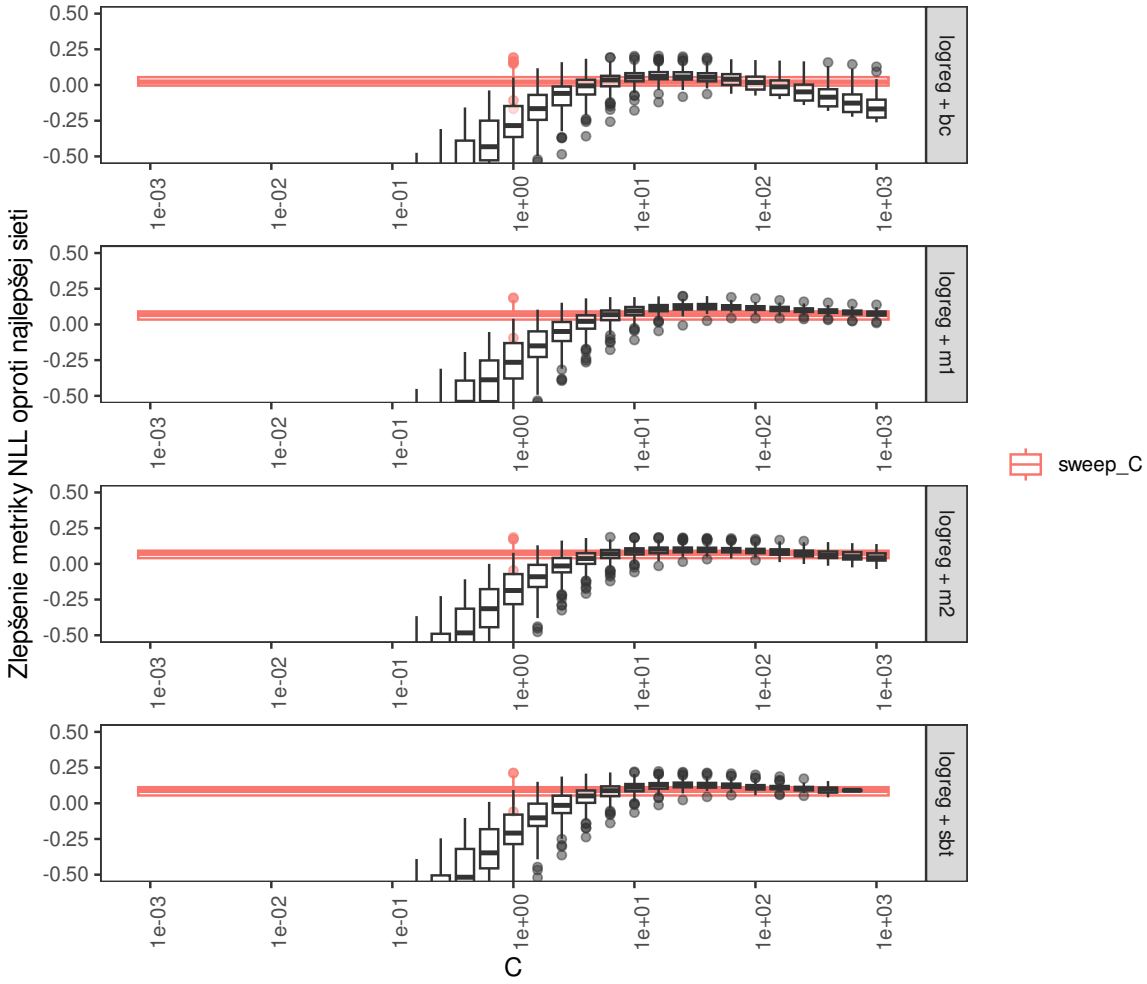
Experimenty s ladením regularizačného koeficientu C logistickej regresie sme vykonávali pre kombinačné metódy **logreg**, **logreg.uncert**, **logreg_no_interc** a **logreg_no_interc.uncert**. Testy sme vykonali na oboch datasetoch CIFAR a testovali sme hodnoty koeficientu $C \in \{10^{-3.0}, 10^{-2.8}, 10^{-2.6}, \ldots, 10^{2.6}, 10^{2.8}, 10^{3.0}\}$, teda spolu 31 hodnôt. Rovnako ako v predchádzajúcej časti sme ansámblové modely trénovali na všetkých aspoň dvojprvkových podmnožinách klasifikátorov googlenet, stochasticdepth50, resnext101, seresnet34, clip_ViT-B-30_LP a clip_ViT-B-16_LP. Kvalitu



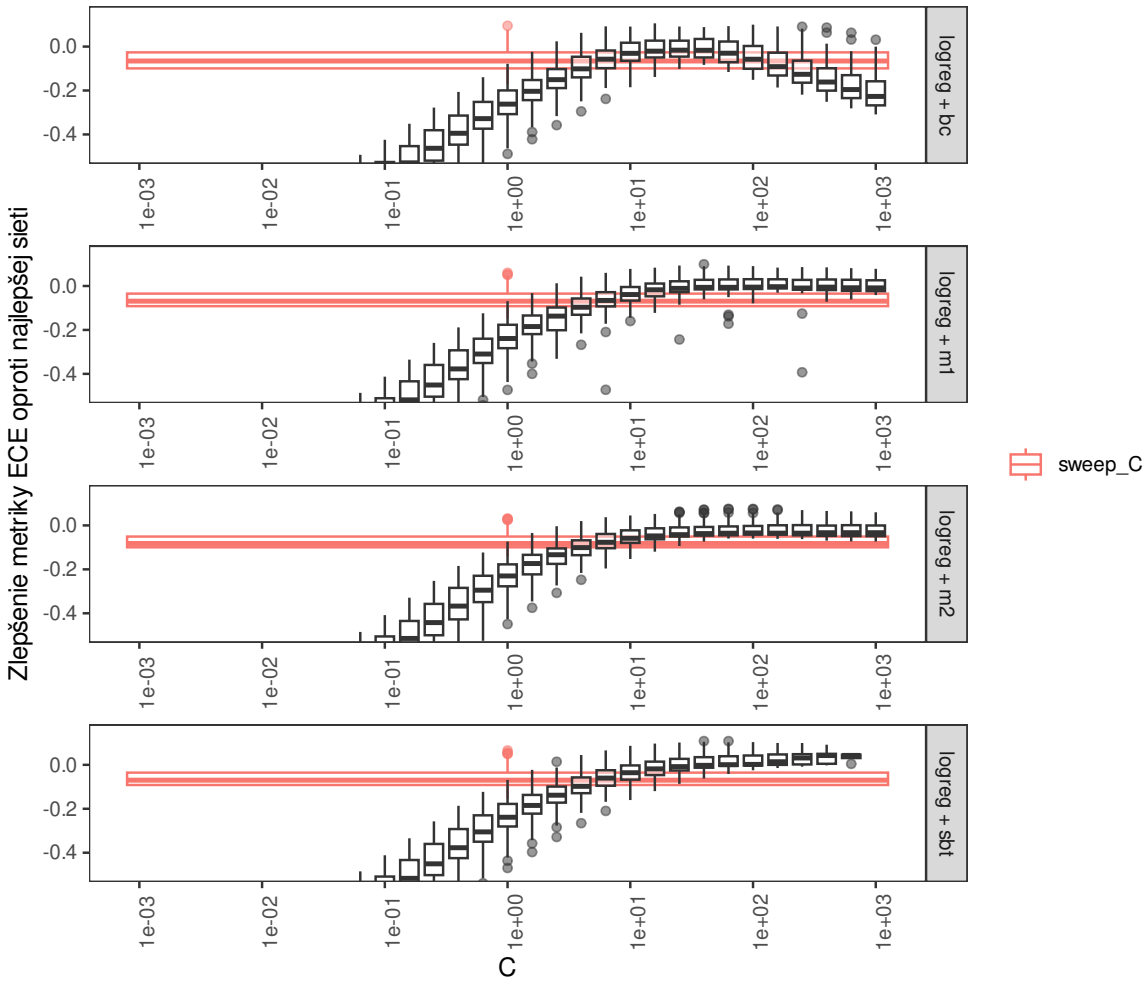
Obrázok 7.20: CIFAR-100. Výsledky kombinačnej metódy **logreg** pre metriku presnosť s rôznymi nastaveniami regularizačného parametra C. Červený boxplot znázorňuje zlepšenia dosiahnuté verziou kombinačnej metódy **logreg__sweep__C**. Nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.

ansámblových modelov sme vyhodnocovali pomocou rozdielu medzi hodnotou metriky dosiahnutou ansámblom a najlepšou hodnotou metriky spomedzi kombinovaných klasifikátorov. Výsledky na datasete CIFAR-100 pre kombinačnú metódu **logreg** sú zobrazené na grafoch 7.20, 7.21 a 7.22.

Ako môžeme z grafov vidieť, pre metriku presnosť boli najlepšie výsledky dosiahnuté pre hodnoty C medzi $10^{0.6}$ a $10^{1.4}$. Párová zväzovacia metóda **bc** sa z tohto pravidla vymyká, ale zlepšenie presnosti s jej použitím je takmer o 2.5% menšie ako s použitím ostatných párových zväzovacích metód, preto ju budeme uvažovať s menšou váhou na finálne rozhodnutie. Metrike NLL najlepšie vyhovujú nastavenia parametra C $10^{1.0}$ a vyššie. Metrika ECE dosahuje najlepšie výsledky pre hodnoty C, medzi $10^{1.0}$



Obrázok 7.21: CIFAR-100. Výsledky kombinačnej metódy **logreg** pre metriku NLL s rôznymi nastaveniami regularizačného parametra C. Červený boxplot znázorňuje zlepšenia dosiahnuté verziou kombinačnej metódy **logreg__sweep__C**. Nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.



Obrázok 7.22: CIFAR-100. Výsledky kombinačnej metódy **logreg** pre metriku ECE s rôznymi nastaveniami regularizačného parametra C. Červený boxplot znázorňuje zlepšenia dosiahnuté verziou kombinačnej metódy **logreg__sweep__C**. Nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.

Tabuľka 7.7: Zvolené hodnoty regularizačného parametra C pre kombinačné metódy využívajúce logistickú regresiu.

Kombinačná metóda	**Zvolená hodnota parametra** C
logreg	$10^{1.2}$
logreg_no_interc	$10^{1.2}$
logreg.uncert	$10^{1.8}$
logreg_no_interc.uncert	$10^{1.6}$

a $10^{2.0}$. Pri datasete CIFAR-10 sme pozorovali najlepšie výsledky pre podobné, alebo o trochu nižšie hodnoty parametra C, ako výslednú hodnotu sme pre kombinačnú metódu **logreg** preto zvolili $10^{1.2}$. Podobnými úvahami sme určili vhodné hodnoty hyperparametra C aj pre ostatné kombinačné metódy využívajúce logistickú regresiu. Ich hodnoty sú uvedené v tabuľke 7.7. Grafy pre ostatné kombinačné metódy a pre oba datasety je možné nájsť v prílohe. Vo všetkých prípadoch sme pozorovali, že nastavenie jednotného parametra regularizácie C pre všetky trénované modely logistickej regresie nemá negatívny vplyv na činnosť výsledného ansámblového modelu oproti jeho individuálnemu nastavovaniu pomocou **sweep_C** variantu kombinačných metód.

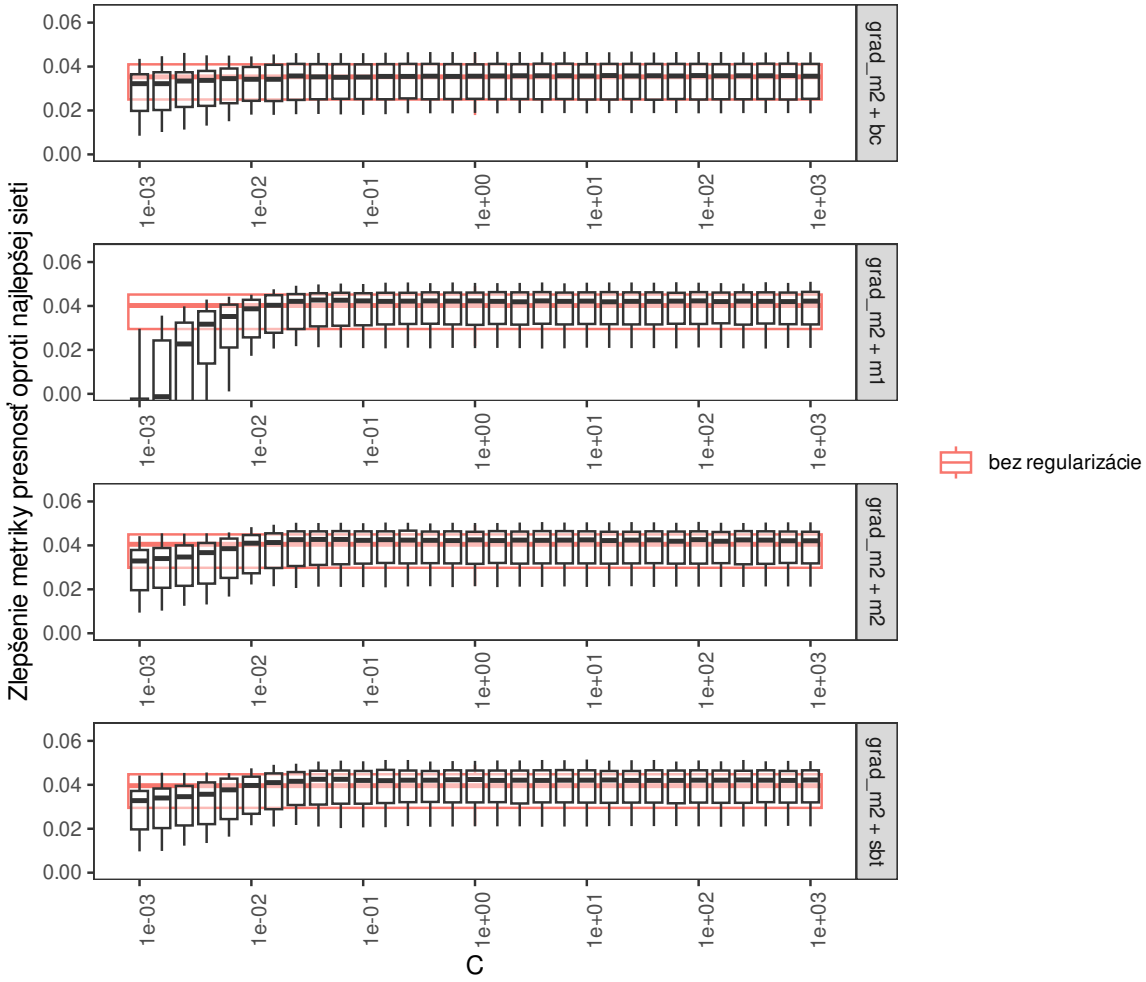
Pri nízkej regularizácii sme v niektorých prípadoch pozorovali numerickú nestabilitu párovej zväzovacej metódy **sbt**, regularizácia teda napomáha aj numerickej robustnosti výsledného modelu.

7.5.2 Regularizácia v gradientových kombinačných metódach

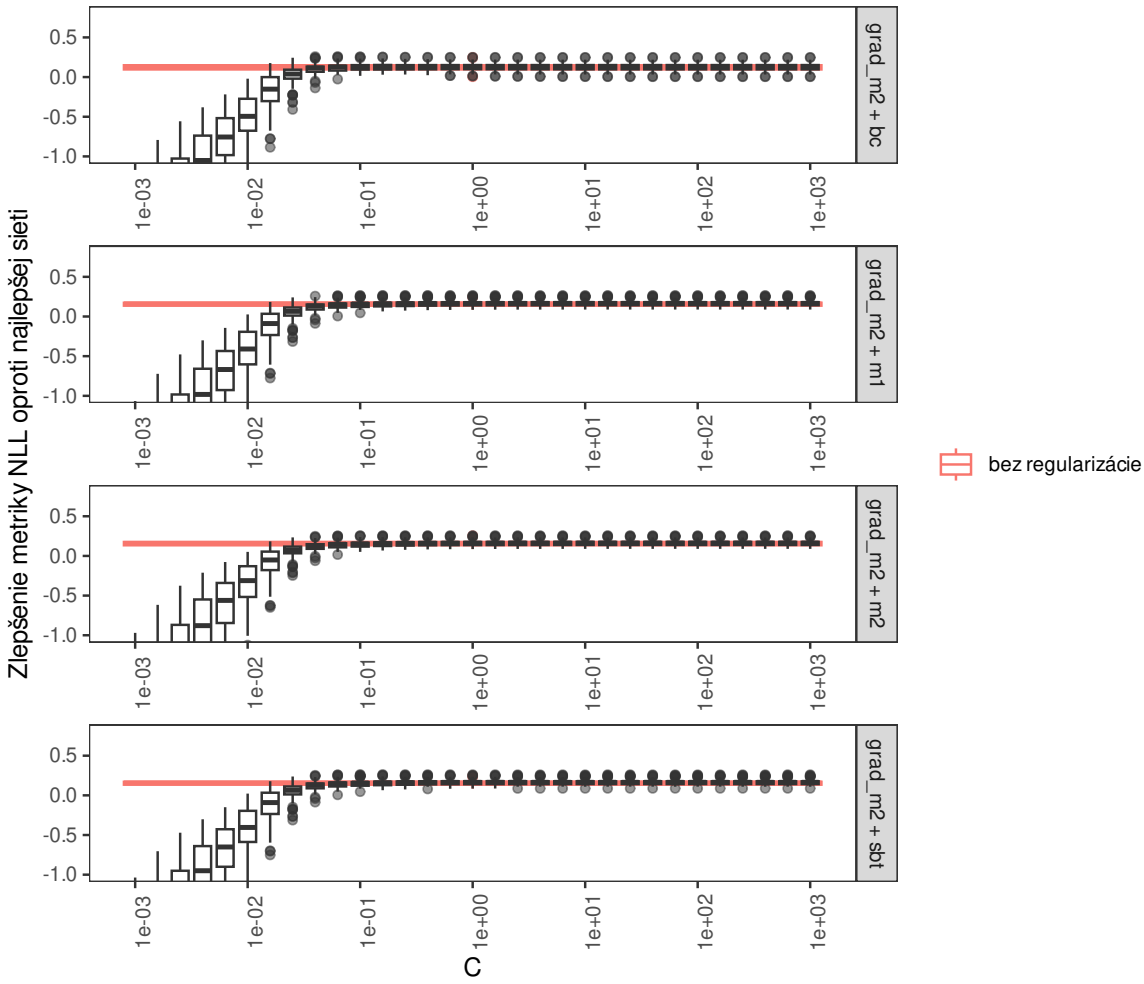
Vzhľadom k dobrým výsledkom dosiahnutým pomocou regularizácie s kombinačnými metódami využívajúcimi logistickú regresiu sme implementovali regularizáciu aj pre gradientové kombinačné metódy. Rovnako, ako pri kombinačných metódach založených na logistickej regresii, aj tu sme použili l_2 regularizáciu, pričou miera regularizácie je určená parametrom C a nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.

Výsledky pre kombinačnú metódu **grad_m2** na datasete CIFAR-100 a metriky presnosť, NLL a ECE sú zobrazené na obrázkoch 7.23, 7.24 a 7.25. Grafy pre ostatné gradientové kombinačné metódy a dataset CIFAR-10 sú v prílohe.

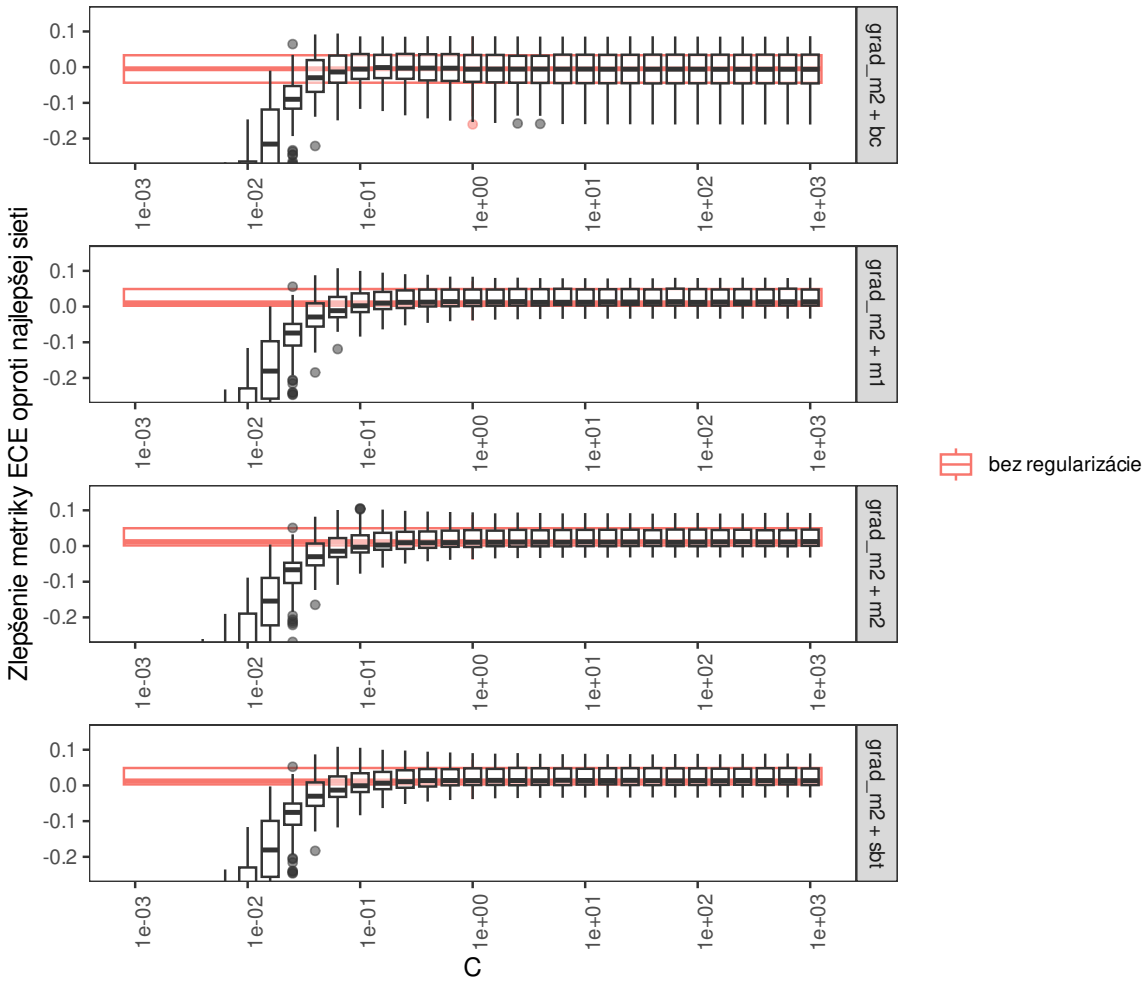
Z výsledkov na obrázku 7.23 môžeme vidieť, že vplyv regularizácie na presnosť je



Obrázok 7.23: CIFAR-100. Výsledky kombinačnej metódy **grad_m2** pre metriku presnosť s rôznymi nastaveniami regularizačného parametra C. Červený boxplot zobrazuje zlepšenia dosiahnuté rovnakou kombinačnou metódou bez regularizácie. Nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.



Obrázok 7.24: CIFAR-100. Výsledky kombinačnej metódy **grad_m2** pre metriku NLL s rôznymi nastaveniami regularizačného parametra C. Červený boxplot zobrazuje zlepšenia dosiahnuté rovnakou kombinačnou metódou bez regularizácie. Nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.



Obrázok 7.25: CIFAR-100. Výsledky kombinačnej metódy **grad_m2** pre metriku ECE s rôznymi nastaveniami regularizačného parametra C. Červený boxplot zobrazuje zlepšenia dosiahnuté rovnakou kombinačnou metódou bez regularizácie. Nižšie hodnoty parametra C vyjadrujú silnejšiu regularizáciu.

Tabuľka 7.8: Zvolené hodnoty regularizačného parametra C pre gradientové kombinačné metódy.

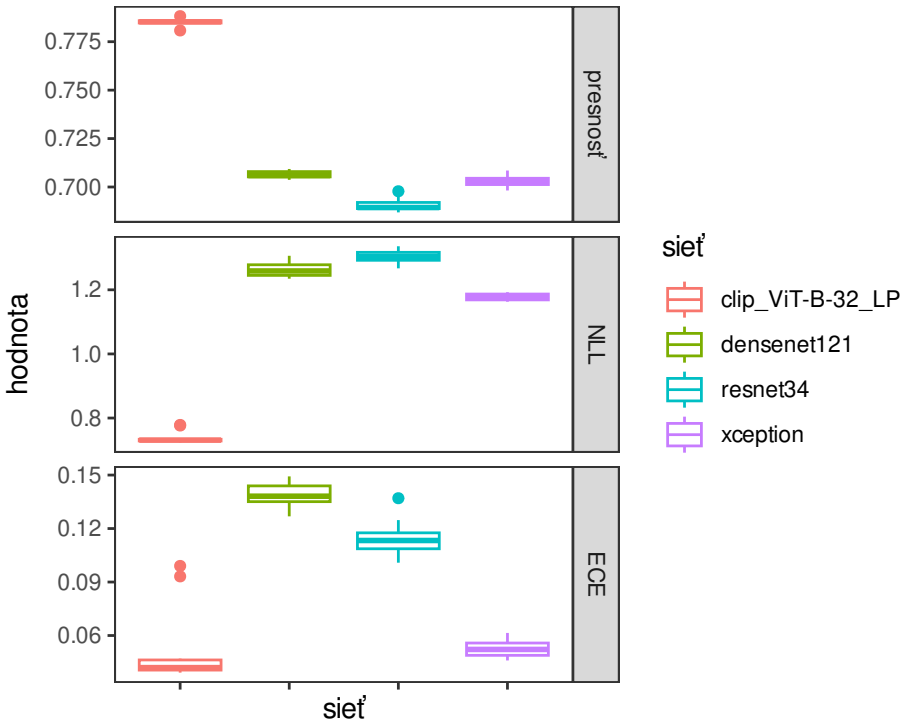
Kombinačná metóda	**Zvolená hodnota parametra** C
grad_bc	$10^{-0.4}$
grad_bc.uncert	$10^{0.0}$
grad_m1	$10^{-0.2}$
grad_m1.uncert	$10^{0.0}$
grad_m2	$10^{-0.6}$
grad_m2.uncert	$10^{0.0}$

malý, s výnimkou párovej zväzovacej metódy **m1**, kde silná regularizácia spôsobuje zhoršenie presnosti. Pre všetky párové zväzovacie metódy je presnosť výsledných modelov pre hodnoty prametra C $10^{-1.6}$ a vyššie prakticky rovnaká. Pri metrikách ECE a NLL spôsobuje vysoká regularizácia zhoršenie výsledkov, pri nižšej regularizácii sa výsledky postupne stabilizujú. Pri metrike ECE dochádza k tejto stabilizácii okolo hodnoty C $10^{-0.6}$ a pri metrike NLL okolo hodnoty C $10^{-1.2}$. Pre dataset CIFAR-10 pozorujeme podobné správanie, stabilizácia výsledkov ale nastáva pri mierne vyšších hodnotách parametra C a teda pri nižšej regularizácii. Ako výslednú hodnotu parametra C pre kombinačnú metódu **grad_m2** volíme $10^{-0.6}$. Pre ostatné gradientové metódy sme určili vhodné hodnoty parametra C podobným spôsobom. Výsledné hodnoty sú zobrazené v tabuľke 7.8. Aj keď regularizácia pri gradientových metódach neprináša výrazné zlepšenie výsledkov, rozhodli sme sa ju naďalej využívať pre jej potenciál pre zvýšenie robustnosti výsledných modelov zlepšením numerickej stability párových zväzovacích metód, ktoré sa prejavilo najmä pri kombinačnej metóde **grad_m1**.

Výsledky experimentov v sekcii: Zvolené hodnoty regularizačného parametra pre kombinačné metódy založené na logistickej regresii sú zobrazené v tabuľke 7.7, pre gradientové kombinačné metódy v tabuľke 7.8.

7.6 Vyhodnotenie na datasete CIFAR-100

Cieľ experimentov v sekcii: Porovnať vybrané konfigurácie odladenej WLE metódy s baseline metódou.



Obrázok 7.26: Metriky sietí trénovaných na polovici trénovacej množiny datasetu CIFAR-100. Pri metrikách NLL a ECE predstavujú nižšie hodnoty lepší výsledok.

V predchádzajúcich sekciách sme odladili trénovaciu metodológiu ansámblovej metódy WLE. Na základe predbežných experimentov na datasetoch CIFAR-10 a CIFAR-100 sme vybrali niekoľko konfigurácií, ktoré dosahovali sľubné výsledky. V týchto experimentoch sa ukázalo, že baseline ansámbel a WLE v metrike presnosť dosahujú podobné výsledky.

Aby bolo možné tieto ansámblové metódy lepšie odlíšiť, vytvárame v tomto experimente náročnejšiu úlohu. Jej náročnosť spočíva v menšom počte trénovacích dát pre členy ansámblu. Trénovanie členov ansámblu realizujeme na polovici trénovacej množiny datasetu CIFAR-100. Kombinačné metódy a baseline ansámbel trénujeme na náhodne vybranej množine veľkosti 5000 vzoriek z nevyužitej polovice trénovacej množiny. Ako členy ansámblu používame neurónové siete resnet34, densenet121, xception a predtrénovaný obrazový transformer clip_ViT-B-32 s dotrénovanou poslednou vrstvou, v experimentoch označený ako clip_ViT-B-32_LP.

Náhodné rozdelenie trénovacej množiny a trénovanie neurónových sietí opakujeme 10-krát. Metriky kombinovaných sietí sú zobrazené na obrázku 7.26. Z grafu je viditeľná jasná prevaha klasifikátora clip_ViT-B-32_LP v metrike presnosť a NLL. Zvyšné

tri klasifikátory majú približne rovnakú presnosť. V metrike ECE nadobúda podobne dobré hodnoty ako clip_ViT-B-32_LP aj klasifikátor xception.

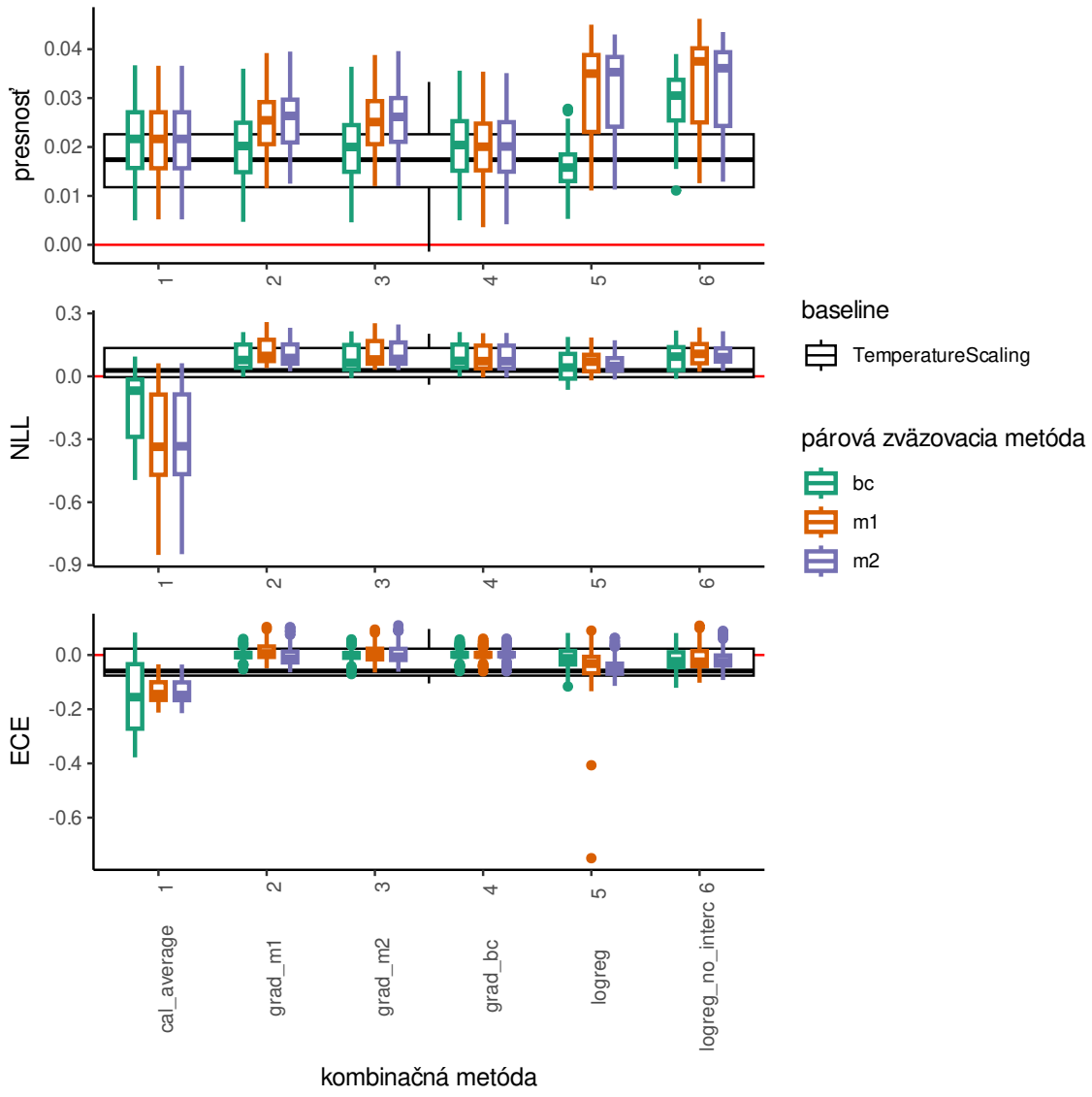
Pri ansámblovaní kombinujeme len siete trénované na rovnakej množine. Získame tak ansámble veľkosti 2 až 4, pričom každý sa opakuje 10-krát pre jednotlivé replikácie trénovania neurónových sietí. Vyhodnotenie metrík pre šesť vybraných konfigurácií WLE ansámblu a ich porovnanie s baseline ansámblom je zobrazené na obrázku 7.27.

Môžeme vidieť, že v metrike presnosť dosahujú najlepšie výsledky kombinačné metódy založené na logistickej regresii, najmä metóda bez absolútneho koeficientu **logreg_no_interc**. Za nimi nasledujú gradientové metódy, najmä **grad_m1** a **grad_m2**. V ostatných metrikách dosahujú gradientové metódy a metódy založené na logistickej regresii podobné výsledky.

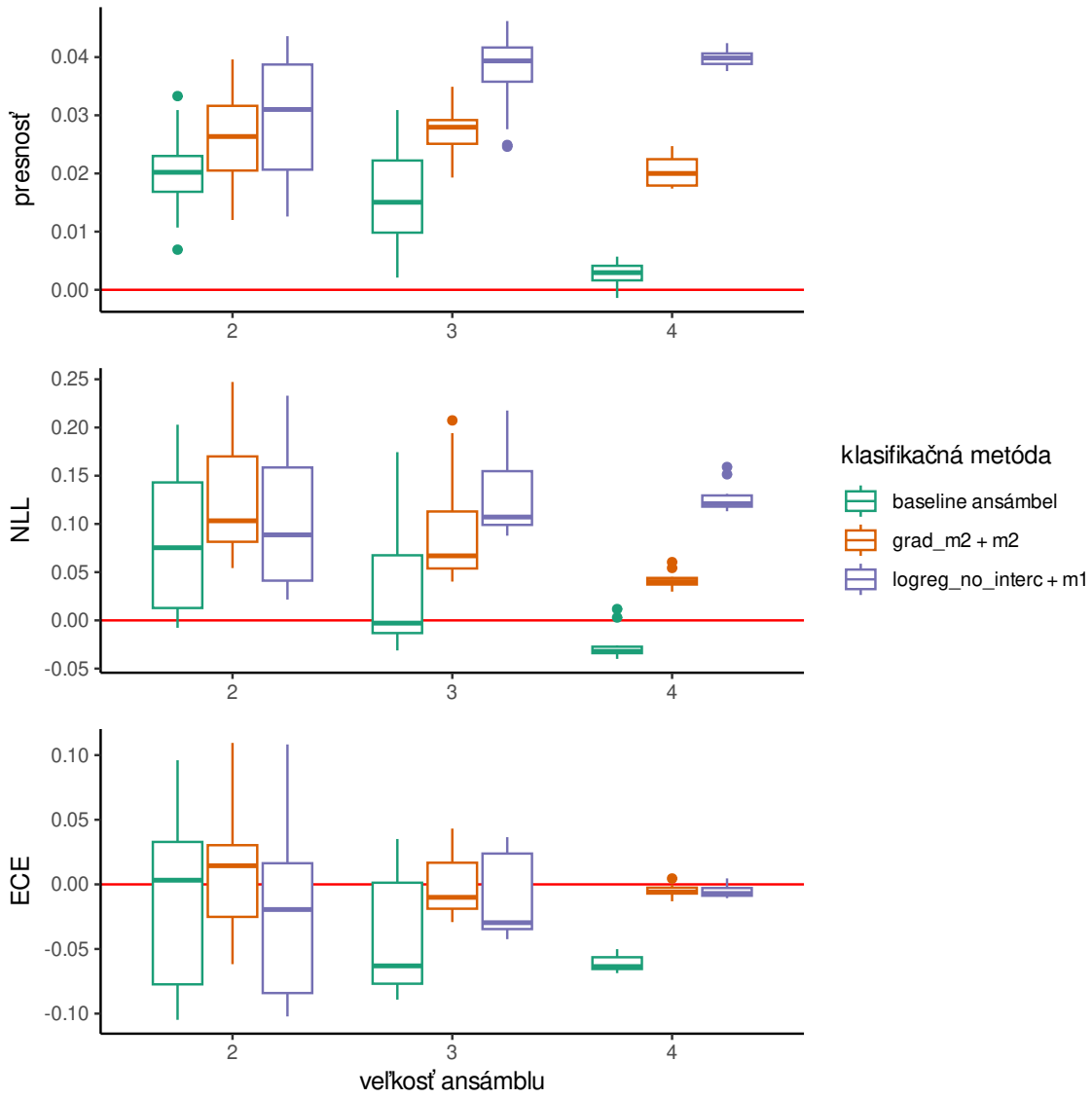
Kombinačná metóda **cal_average**, ktorá priraďuje všetkým dvojiciam tried jedného klasifikátora rovnaký koeficient, dosahuje vo všetkých metrikách horšie výsledky ako ostatné WLE metódy. U všetkých metód, vrátane baseline môžeme v metrike presnosť konštatovať zlepšenie oproti najlepšej kombinovanej sieti. V metrike NLL kombinačná metóda **cal_average** vo väčšine prípadov nedosahuje zlepšenie oproti najlepšej kombinovanej sieti. Ostatné ansámblové metódy vrátane baseline v metrike NLL vo väčšine prípadov dosahujú zlepšenie oproti najlepšej kombinovanej sieti. Pre metriku ECE sú výsledky rôznorodé. Baseline ansámbel nedosahuje vo väčšine prípadov zlepšenie, väčšina WLE metód sa správa o niečo lepšie ako baseline ansámbel. Porovnanie baseline ansámblu a WLE metód vyhodnotíme v závere sekcie pomocou štatistických testov.

Zo zobrazených konfigurácií sme vybrali dve, **logreg_no_interc + m1** a **grad_m2 + m2**, na ktorých vykonáme hlbšiu analýzu. Aby sme lepšie porozumeli správaniu sa testovaných ansámblových metód, zobrazujeme zlepšenia v sledovaných metrikách oproti najlepšiemu členu ansámblu osobitne pre jednotlivé veľkosti ansámblu. Takéto zobrazenie je na obrázku 7.28.

Pre baseline ansámbel je vo všetkých troch metrikách s rastúcou veľkosťou ansámblu viditeľný pokles zlepšenia. Podobný pokles je v menšej miere viditeľný aj pre WLE konfiguráciu **grad_m2 + m2**. Pre WLE konfiguráciu **logreg_no_interc + m1** môžeme pozorovať opačný trend, s rastúcou veľkosťou ansámblu zlepšenie v metrike presnosť a NLL rastie, pre metriku ECE nie je trend jednoznačný.



Obrázok 7.27: Zlepšenie v sledovaných metrikách oproti najlepšiemu členu ansámblu pre vybrané konfigurácie WLE z neurónových sietí trénovaných na polovici datasetu CIFAR-100.



Obrázok 7.28: Zlepšenie v sledovaných metrikách oproti najlepšiemu členu ansámblu pre dve konfigurácie WLE z neurónových sietí trénovaných na polovici datasetu CIFAR-100. Veľkosť ansámblu vyjadruje počet jeho členov.

Ak sa vrátime ku grafu s metrikami kombinovaných sietí na obrázku 7.26, vidíme, že u väčšiny ansámblov prebieha kombinovanie jedného modelu poskytujúceho kvalitnejšie predikcie s niekoľkými slabšími modelmi. Pre lepšie preskúmanie tejto situácie sme rozdelili ansámble z grafu 7.28 na tie, ktoré obsahujú model clip_ViT-B-32_LP a tie, ktoré ho neobsahujú. Zobrazenie pre ansámble zahŕňajúce clip_ViT-B-32_LP je na obrázku 7.29, zobrazenie pre zvyšné ansámble, ktoré ho neobsahujú je na obrázku 7.30.

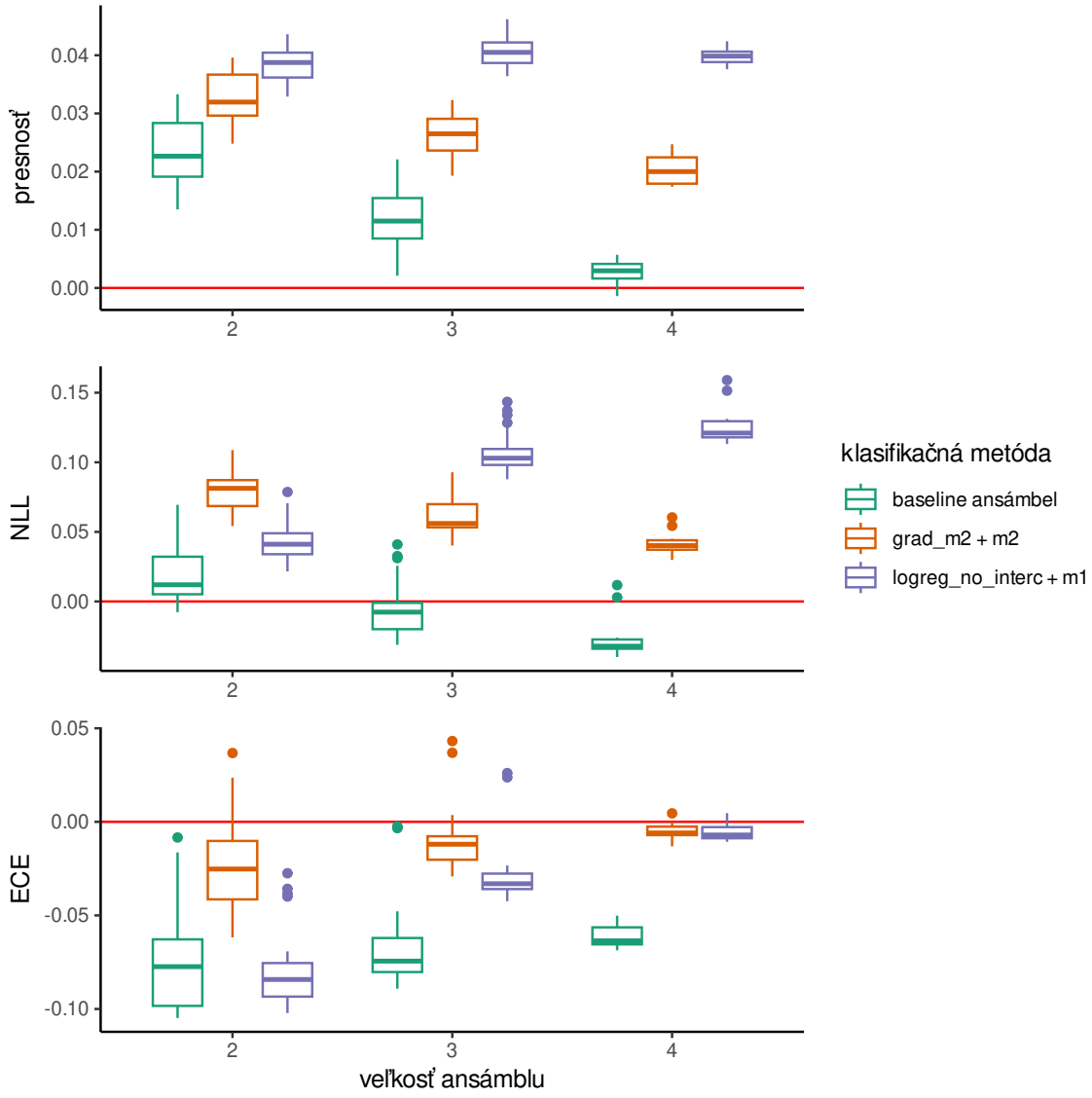
V metrikách presnosť a NLL na obrázku 7.29, zahŕňajúcom model clip, môžeme u baseline ansámblu a WLE ansámblu **grad_m2 + m2** pozorovať podobný pokles zlepšenia s narastajúcou veľkosťou ansámblu ako sme pozorovali na obrázku 7.28 pre všetky kombinácie sietí. Na obrázku 7.30, ktorý neobsahuje model clip, je viditeľný rast zlepšenia. Pre metriku ECE toto pozorovanie neplatí. Všetky tri ansámblové metódy majú pre ansámble zahŕňajúce model clip v metrike ECE rastúci trend.

Z opísaných pozorovaní je vidieť, že pridávanie viacerých slabších klasifikátorov k silnejšiemu klasifikátoru spôsobuje pre ansámblové netódy baseline a **grad_m2 + m2** pokles presnosti a NLL. Malý pokles v presnosti medzi veľkosťami ansámblu 3 a 4 je viditeľný aj u WLE konfigurácie **logreg_no_intercept + m1**. Správanie WLE metód je v tomto ohľade lepšie ako u baseline ansámblu a rozdiel medzi WLE metódami a baseline metódou sa s rastúcou veľkosťou ansámblu zväčšuje.

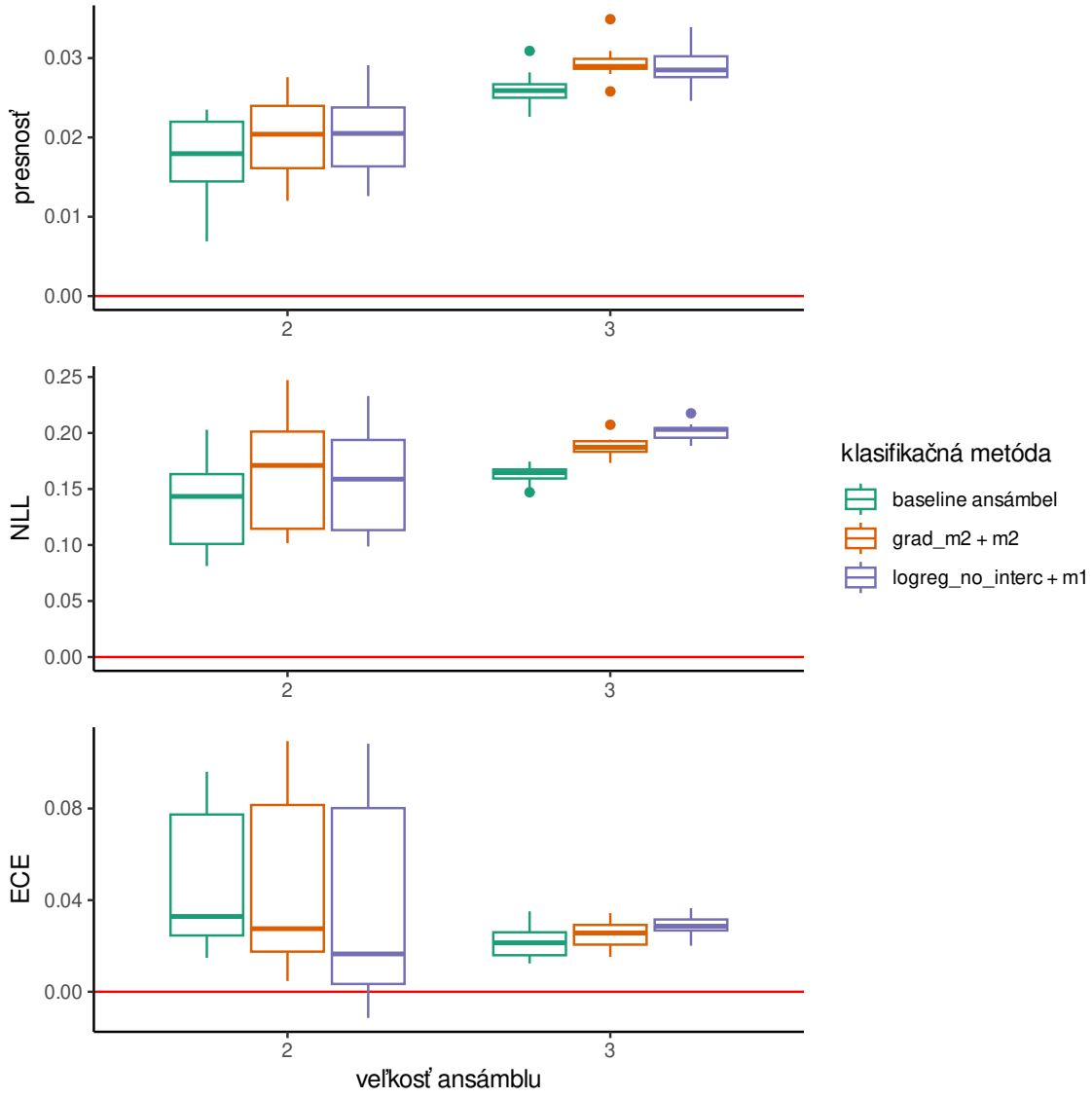
Porovnanie medzi WLE metódami a baseline metódou sme vykonali aj s pomocou štatistických testov. Pre rôzne veľkosti ansámblu sme pozorovali rôzne správanie jednotlivých metód, porovnanie preto vykonávame pre jednotlivé veľkosti ansámblu osobitne. Množiny členov jednotlivých ansámblov nie sú volené pomocou náhodného výberu, používame preto párový permutačný test [99].

Výsledky pre WLE konfigurácie **grad_m2 + m2** a **logreg_no_interc + m1** sú zobrazené v tabuľke 7.9. Výsledky sme vyhodnotili na hladine významnosti 5%. Pre zobrazené dve konfigurácie WLE je z tabuľky zrejmá štatisticky významná prevaha WLE nad baseline ansámblom vo všetkých prípadoch okrem prípadu konfigurácie **logreg_no_interc + m1**, metriky ECE a veľkosti ansámblu 2, pri ktorom dosiahla štatisticky významnú prevahu baseline metóda. Výsledky štatistických testov pre ostatné WLE konfigurácie z obrázku 7.27 sú dostupné v prílohe. Tu k nim poskytneme len súhrnné informácie.

Testy pre všetky testované konfigurácie gradientových kombinačných metód pre



Obrázok 7.29: Zlepšenie v sledovaných metrikách oproti najlepšiemu členu ansámblu pre dve konfigurácie WLE z neurónových sietí trénovaných na polovici datasetu CIFAR-100. Graf zahŕňa len tie ansámble, ktoré obsahujú model clip_ViT-B-32_LP. Veľkosť ansámblu vyjadruje počet jeho členov.



Obrázok 7.30: Zlepšenie v sledovaných metrikách oproti najlepšiemu členu ansámblu pre dve konfigurácie WLE z neurónových sietí trénovaných na polovici datasetu CIFAR-100. Graf zahŕňa len tie ansámble, ktoré neobsahujú model clip_ViT-B-32_LP. Veľkosť ansámblu vyjadruje počet jeho členov.

Tabuľka 7.9: Výsledky štatistických testov porovnania s baseline metódou pre ansámble zložené z neurónových sietí trénovaných na polovici datasetu CIFAR-100.

WLE konfigurácia	**metrika**	**veľkosť ansámblu**	**p-hodnota**	**lepšia metóda**
logreg_no_interc + m1	presnosť	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0019	WLE
	NLL	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0007	WLE
	ECE	2	0.0000	baseline
		3	0.0000	WSL
		4	0.0014	WSL
grad_m2 + m2	presnosť	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0021	WLE
	NLL	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0019	WLE
	ECE	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0017	WLE

všetky metriky a veľkosti ansámblov vyšli v prospech WLE. Testy pre konfigurácie s kombinačnými metódami založenými na logistickej regresii vyšli pre metriky NLL a ECE v niektorých prípadoch pre ansámble veľkosti 2 nerozhodne, alebo v prospech baseline metódy. Testy pre metriku presnosť vyšli pre tieto kombinačné metódy v prospech WLE okrem jedného prípadu konfigurácie **logreg + bc** a veľkosť ansámblu 2, kde test vyšiel v prospech baseline ansámblu.

Výsledok experimentov v sekcii: Vykonané štatistické testy ukázali, že konfigurácie WLE ansámblu s kombinačnými metódami založenými na gradientovom trénovaní dosahujú vo všetkých metrikách lepšie výsledky ako baseline ansámbel. Pri WLE konfiguráciách s kombinačnými metódami založenými na logistickej regresii v niektorých prípadoch veľkosti ansámblu 2 baseline metóda dosahuje lepšie výsledky v metrikách NLL a ECE. Pri kombinačnej metóde **logreg_no_interc** k prevahe baseline metódy dochádza len v prípade metriky ECE. Konfigurácie využívajúce kombinačné metódy založené na logistickej regresii dosahujú väčšie zlepšenie v presnosti ako konfigurácie využívajúce gradientové kombinaćné metódy.

7.7 Detekcia neznámych vzoriek

Cieľ experimentov v sekcii: Preskúmať potenciál nekonzistencie kombinovaných predikcií určený vnútornými ukazovateľmi párových zväzovacích metód pre detekciu neznámych vzoriek a porovnať tento prístup so zaužívaným prístupom **MSP**.

Ako bolo spomenuté v sekcii 6.2, párové zväzovacie metódy umožňujú kvantifikovať mieru nekonzistencie spracovávaných párových pravdepodobností. V tejto sekcii skúmame možnosť použitia takto vyjadrenej miery nekonzistencie ako kvantifikátora neistoty pre účel detekcie neznámych vzoriek (ang. out of distribution detection). Ako baseline sme si zvolili zaužívanú metódu najvyššej predikovanej pravdepodobnosti (ang. maximum softmax probability (MSP)), ktorá sa napriek svojej jednoduchosti ukázala ako vysoko úspešná pre viacero aplikácií [81].

Experimenty vykonávame na benchmarkoch CIFAR-10 vs CIFAR-100 a CIFAR-100 vs CIFAR-10. Pri oboch benchmarkoch sú testované klasifikátory natrénované na prvom datasete a testuje sa rozlíšenie testovacej množiny prvého datasetu od testovacej množiny druhého. Testovacia množina druhého datasetu je teda považovaná za

”mimo trénovacieho rozdelenia” (ang. out of distribution (OOD)) a testovacia množina prvého datasetu za ”v trénovacom rozdelení” (ang. in distribution (IND)). Na porovnávanie jednotlivých metód používame ROC a PR krivky a plochy pod týmito krivkami AUROC a AUPR popísané v sekcii 7.1.

OOD detekciu pri použití individuálnych sietí a tiež pri použití baseline ansámblu realizujeme pomocou **MSP**. Pri WLE porovnávame dva prístupy na detekciu OOD a to neistotu vyjadrenú párovou zväzovacou metódou a aplikáciu **MSP** na pravdepodobnostný výstup ansámblu.

Na základe výsledkov z predchádzajúcich experimentov sme množinu testovaných kombinačných metód obmedzili na: **grad_m2** a **logreg_no_interc**. Množinu párových zväzovacích metód sme tiež obmedzili na: **m1**, **m2** a **bc**. Testujeme všetkých 6 konfigurácií týchto kombinačných a párových zväzovacích metód. Ako je uvedené v sekcii 7.2 experimenty vykonávame s piatimi neurónovými sieťami: R50x1, R101x3, B16, R50_B16 a M_B16. Testujeme všetky aspoň dvojprvkové podmnožiny týchto sietí.

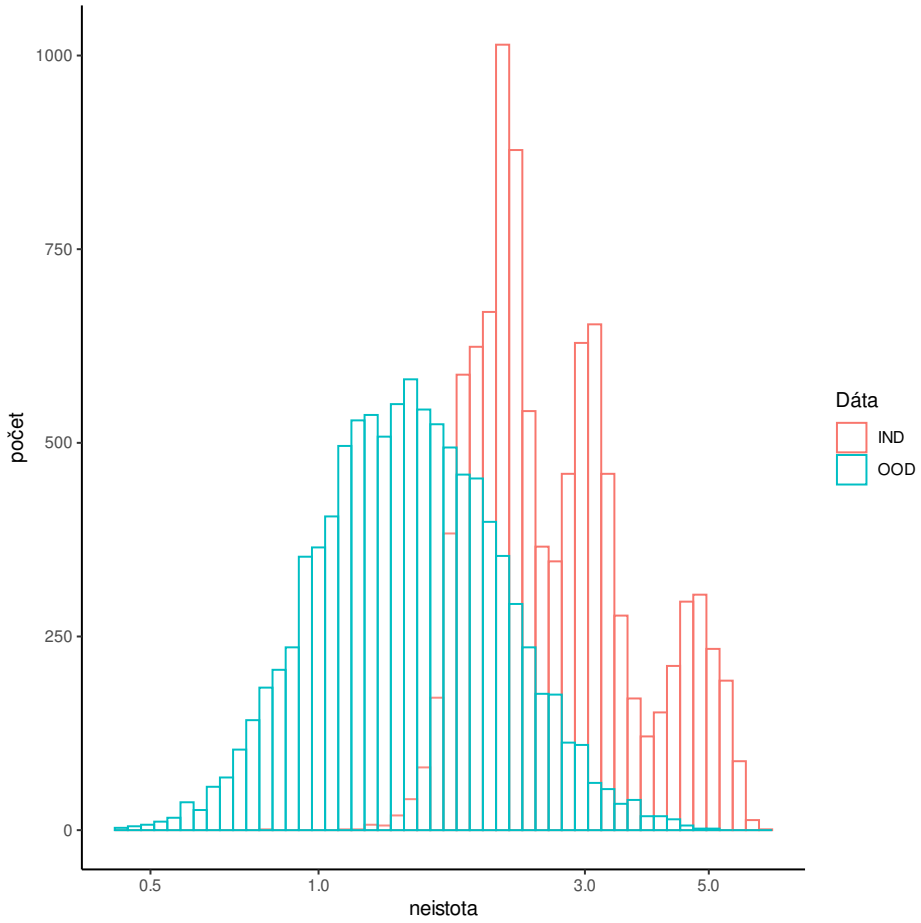
7.7.1 Rozdelenie neistoty párových zväzovacích metód

V tejto podsekcii skúmame rozdelenie neistoty vyjadrenej párovými zväzovacími metódami pre OOD a IND vzorky. Pre každú konfiguráciu a množinu kombinovaných sietí sme rozdelenie neistoty zobrazili ako histogram. Z týchto histogramov sme vyvodili nasledujúce pozorovania.

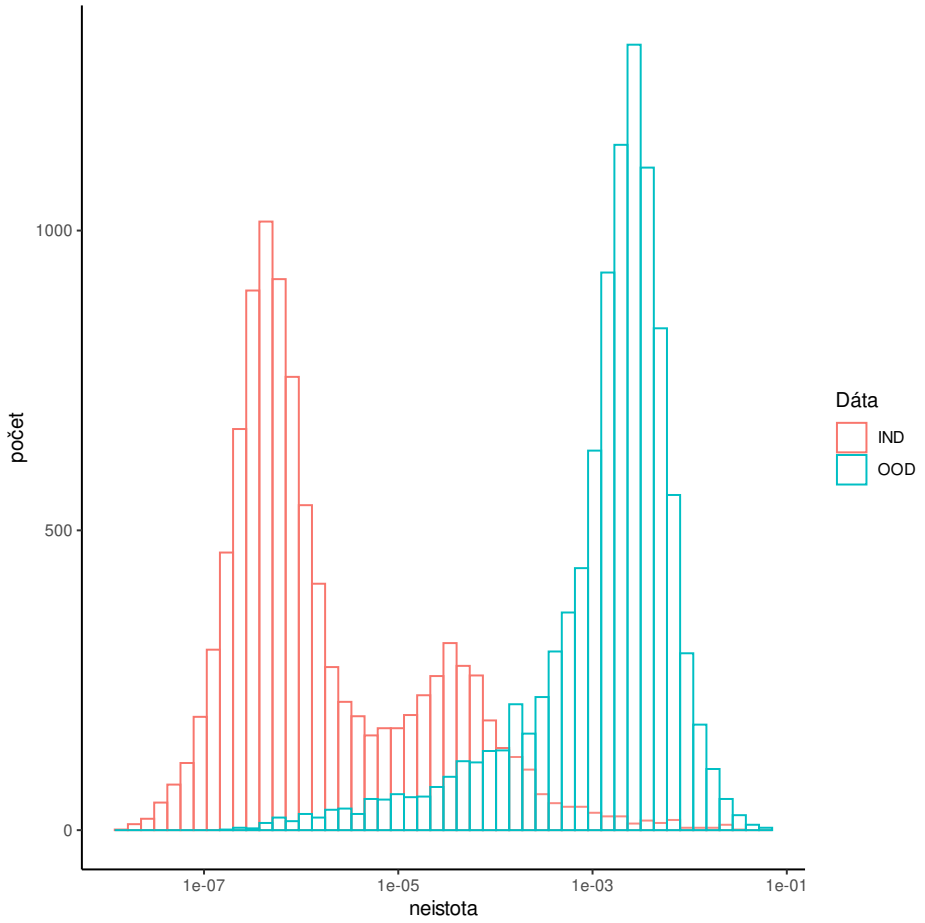
Neistota vyjadrená párovou zväzovacou metódou **bc** má zvláštne a nečakané správanie, kde IND vzorky majú často vyššiu neistotu ako OOD vzorky. Toto pozorovanie platí pre obe testované kombinačné metódy a pre oba benchmarky CIFAR-10 vs CIFAR-100 ako aj CIFAR-100 vs CIFAR-10. Príklad takejto situácie môžeme vidieť na obrázku 7.31.

Párová zväzovacia metóda **m1** má pri použití s kombinačnou metódou **logreg_no_interc** tiež vyššiu neistotu pre IND vzorky ako pre OOD. Pri použití s kombinačnou metódou **grad_m2** sú rozdelenia korektné a väčšinou jasne oddeliteľné.

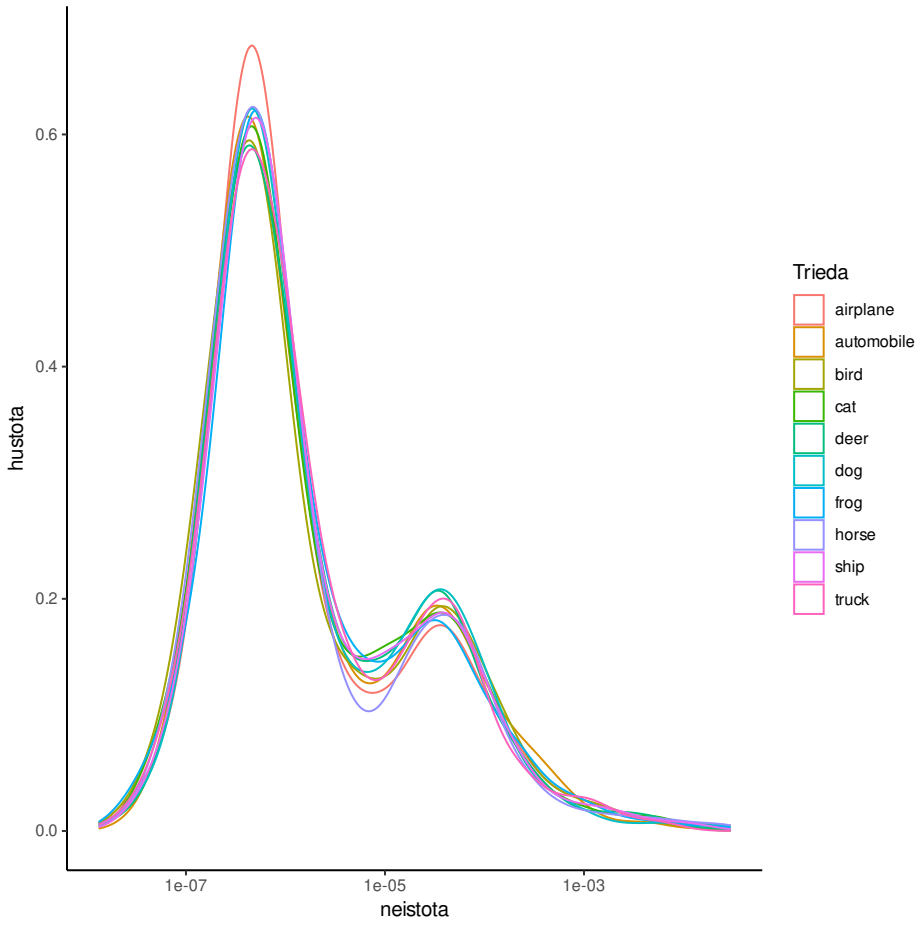
Párová zväzovacia metóda **m2** poskytuje v konfigurácii s oboma kombinačnými metódami správne usporiadané a vo väčšine prípadov dobre separovateľné rozdelenia. Ukážku takéhoto prípadu môžeme vidieť na obrázku 7.32.



Obrázok 7.31: Neistota vyjadrená párovou zväzovacou metódou v konfigurácii **logreg_no_interc + bc** na testovacej úlohe CIFAR-10 vs CIFAR-100 s použitím všetkých 5 sietí. Detekcia OOD pomocou tejto metódy nie je realizovateľná, keďže IND dáta majú vyššiu neistotu ako OOD dáta.



Obrázok 7.32: Neistota vyjadrená párovou zväzovacou metódou v konfigurácii **logreg_no_interc + m2** na testovacej úlohe CIFAR-10 vs CIFAR-100 s použitím všetkých 5 sietí. Detekcia OOD pomocou tejto metódy je realizovateľná, keďže rozdelenia sú rozlíšiteľné.

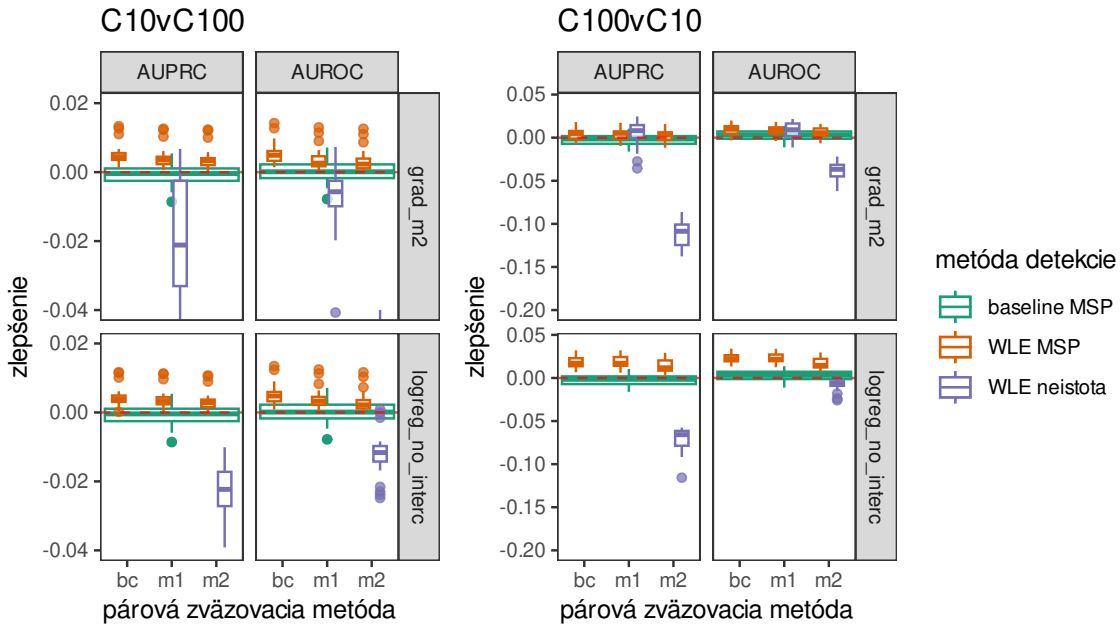


Obrázok 7.33: Rozdelenie neistoty pre IND vzorky separované po triedach na datasete CIFAR-10 pri kombinovaní všetkých piatich sietí pomocou konfigurácie **logreg_no_interc + m2**.

V niektorých prípadoch je rozdelenie neistoty pre IND vzorky bimodálne, alebo aj trimodálne. Tieto prípady sme vyšetrili z hľadiska zastúpenia jednotlivých tried v celom rozsahu rozdelenia neistoty. Na grafe 7.33 analyzujeme ten istý ansámbel, ktorého neistota bola zobrazená na obrázku 7.32. Ako môžeme vidieť, zastúpenie jednotlivých tried je rovnomerné a príčinu multimodality sa nám nepodarilo odhaliť.

7.7.2 Vyhodnotenie kvality detekcie OOD vzoriek

Vizuálna kontrola histogramov ukázala, že párové zväzovacie metódy **m1** a **m2** by mohli poskytnúť použiteľnú funkcionalitu detekcie OOD. S baseline metódou detekcie OOD **MSP** sme metódy založené na neistote porovnávali pomocou plôch pod krivkami ROC a PR. Metódu **MSP** sme aplikovali na výstup našej ansámblovej metódy, na



Obrázok 7.34: Zlepšenie metrík AUROC a AUPRC pre jednotlivé testované metódy detekcie neznámych vzoriek oproti najlepšej z kombinovaných sietí. Metódy, ktoré na grafe nie je vidieť dosahovali horšie výsledky. Na ľavom grafe sú zobrazené výsledky pre benchmark CIFAR-10 vs CIFAR-100 a na pravom grafe pre CIFAR-100 vs CIFAR-10. Červenou čiarkovanou čiarou je znázornené nulové zlepšenie.

výstupy jednotlivých sietí a tiež na výstup baseline ansámblu **TemperatureScaling**. Plochy pod oboma krivkami ROC aj PR dávajú vo väčšine prípadov podobné výsledky, čo sa dá očakávať, keďže testovacie datasety majú rovnaký počet IND vzoriek ako OOD vzoriek.

Podobne ako pri ostatných metrikách, aj tu porovnávame výsledky dosiahnuté testovanými metódami s výsledkami dosiahnutými najlepšou z kombinovaných sietí (s použitím metódy **MSP**). Plochy pod krivkami v nasledujúcich grafoch sú zobrazené v podobe zlepšenia oproti ploche pod zodpovedajúcou krivkou pre tú z kombinovaných sietí, ktorá si v danej metrike počínala najlepšie. Spoločne zobrazujeme výsledky pre detekciu pomocou neistoty z párových zväzovacích metód WLE ako aj pre aplikáciu **MSP** na baseline ansámbel a na výstupy WLE ansámblu. Toto porovnanie je pre obe riešené úlohy zobrazené na obrázku 7.34. Baseline ansámbel dosahuje podobné výsledky ako najlepšia z kombinovaných sietí. Metódy využívajúce neistotu z párových zväzovacích metód dosahujú stabilné zlepšenie len v konfigurácii **grad_m2 + m1** a len pri

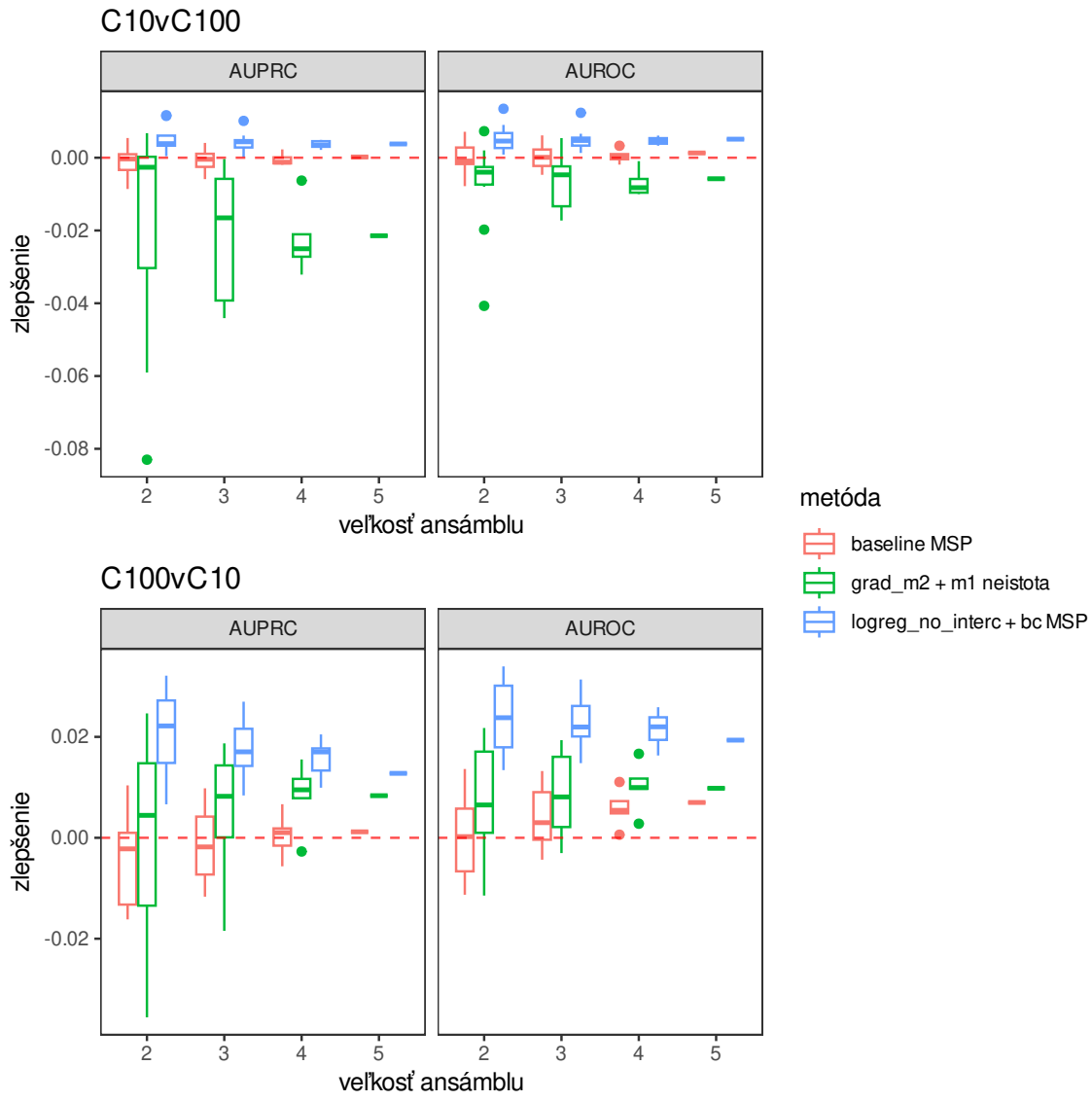
úlohe CIFAR-100 vs CIFAR-10. Aplikácia **MSP** na výstupy WLE ansámblu dosahuje zlepšenie vo všetkých konfiguráciách, pre kombinačnú metódu **logreg_no_interc** o niečo výraznejšie ako pre kombinačnú metódu **grad_m2**.

Analogicky k predchádzajúcim experimentom aj tu vyšetríme správanie sa skúmaných metód v závislosti od veľkosti ansámblu. Pre túto analýzu sme vybrali metódu využívajúcu neistotu z párovej zväzovacej metódy v konfigurácii **grad_m2 + m1** a aplikáciu **MSP** na WLE konfiguráciu **logreg_no_interc + bc**. Zobrazenie zlepšenia v sledovaných metrikách pre jednotlivé veľkosti ansámblu je na obrázku 7.35. Z obrázku môžeme pozorovať, že výsledky pre baseline ansámbel s **MSP** sú vo väčšine prípadov pre rôzne veľkosti ansámblu stabilné, zreteľnejší rast zlepšenia je viditeľný len pre metriku AUROC pri úlohe CIFAR-100 vs CIFAR-10. V metódach využívajúcich WLE ansámbel nie sú viditeľné zrejmé vzory, ktoré by sa opakovali v oboch metrikách, alebo pre obe riešené úlohy.

Z praktického dôvodu malého počtu vzoriek pre ansámble veľkosti 4 a 5 a tiež z dôvodu, že sme nenašli špecifické vzory správania sa jednotlivých metód pre rôzne veľkosti ansámblu vykonáme štatistické porovnanie testovaných metód spoločne pre všetky veľkosti ansámblov. Porovnanie vykonávame medzi metódami využívajúcimi WLE ansámbel a baseline ansámblom s použitím **MSP**. Rovnako ako pri vyhodnocovaní klasifikačných experimentov aj tu používame párový permutačný test [99]. Výsledky pre dve vybrané WLE metódy sú zobrazené v tabuľke 7.10. Výsledky boli vyhodnotené na hladine významnosti 5%. Z tabuľky je vidieť, že metóde založenej na neistote z párovej zväzovacej metódy sa podarilo baseline ansámbel s **MSP** prekonať len na úlohe CIFAR-100 vs CIFAR-10.

Metóda, pri ktorej sme aplikovali **MSP** na výstup WLE prekonala baseline v oboch metrikách a na oboch úlohách.

Výsledky testov pre ostatné metódy zobrazené v 7.34 sú dostupné v prílohe. V týchto výsledkoch v oboch metrikách a na oboch riešených úlohách prekonáva baseline ansámbel s **MSP** aplikácia metódy **MSP** na každú z WLE konfigurácií zobrazených na obrázku 7.34. Pri použití neistoty z párových zväzovacích metód prekonáva WLE metóda baseline ansámbel s **MSP** len pri použití konfigurácie **grad_m2 + m1** aj to len na úlohe CIFAR-100 vs CIFAR-10. V ostatných prípadoch vyhráva baseline ansámbel s **MSP**.



Obrázok 7.35: Zlepšenie metrík AUROC a AUPRC oproti najlepšej z kombinovaných sietí pre dve vybrané WLE metódy a baseline ansámbel pri použití na detekciu neznámych vzoriek. Červenou čiarkovanou čiarou je znázornené nulové zlepšenie. Veľkosť ansámblu vyjadruje počet jeho členov.

Tabuľka 7.10: Výsledky štatistických testov porovnania metód detekcie neznámych vzoriek založených na WLE s aplikovaním metódy **MSP** na baseline ansámbel.

WLE metóda	**úloha**	**metrika**	**p-hodnota**	**lepšia metóda**
logreg_no_interc + bc MSP	C10 vs C100	AUROC	0.0000	WLE
		AUPRC	0.0000	WLE
	C100 vs C10	AUROC	0.0000	WLE
		AUPRC	0.0000	WLE
grad_m2 + m1 neistota	C10 vs C100	AUROC	0.0000	baseline
		AUPRC	0.0000	baseline
	C100 vs C10	AUROC	0.0049	WLE
		AUPRC	0.0336	WLE

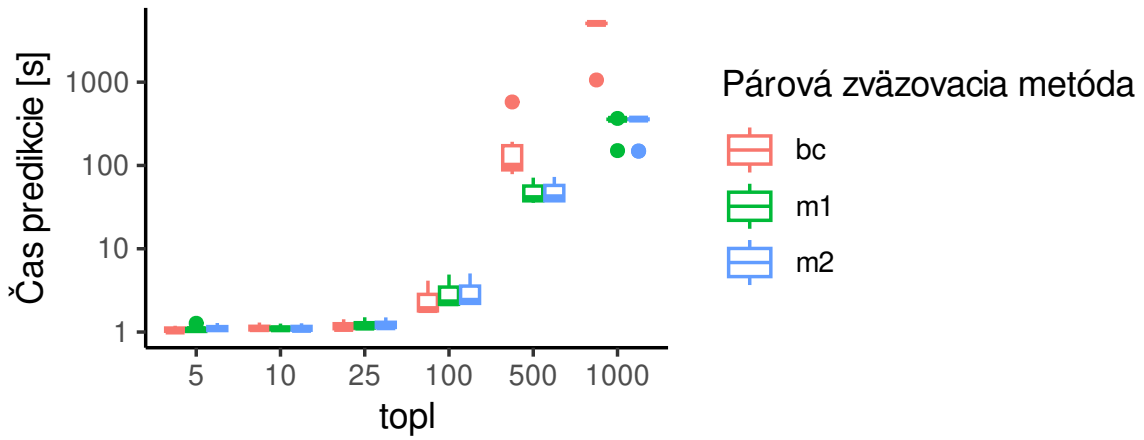
Výsledok experimentov v sekcii: Detekcia OOD pomocou neistoty z párových zväzovacích metód priniesla zlepšenie oproti baseline metóde len v konfigurácii **grad_m2 + m1**, a to len na úlohe CIFAR-100 vs CIFAR-10. Zistili sme však, že WLE ansámbel produkuje výsledky, ktoré po aplikácii metódy **MSP** umožňujú dobrú OOD detekciu prekonávajúcu aplikáciu **MSP** na baseline ansámbel.

7.8 Ladenie hyperparametra zjednodušenej predikcie

Cieľ experimentov v sekcii: Určiť vhodné hodnoty hyperparametra *topl* zavedeného v sekcii 6.3 pre jednotlivé konfigurácie WLE.

Experimenty v predchádzajúcich sekciách boli realizované na 10 a 100 triednej úlohe. Aby sme overili použiteľnosť WLE na väčšej úlohe, pre nasledujúci experiment sme zvolili dataset ImageNet1k [34]. Tento dataset obsahuje obrázky zaradené do 1000 rôznych tried.

Pri úlohe s takýmto počtom tried je relevantná úprava, ktorú sme navrhli v sekcii 6.3. Táto úprava zjednodušuje proces predikcie, pričom tréning kombinačných metód ostáva nezmenený. Úprava je riadená hyperparametrom *topl*. Nižšie hodnoty *topl* predstavujú väčšie zjednodušenie, hodnota rovná počtu tried problému reprezentuje

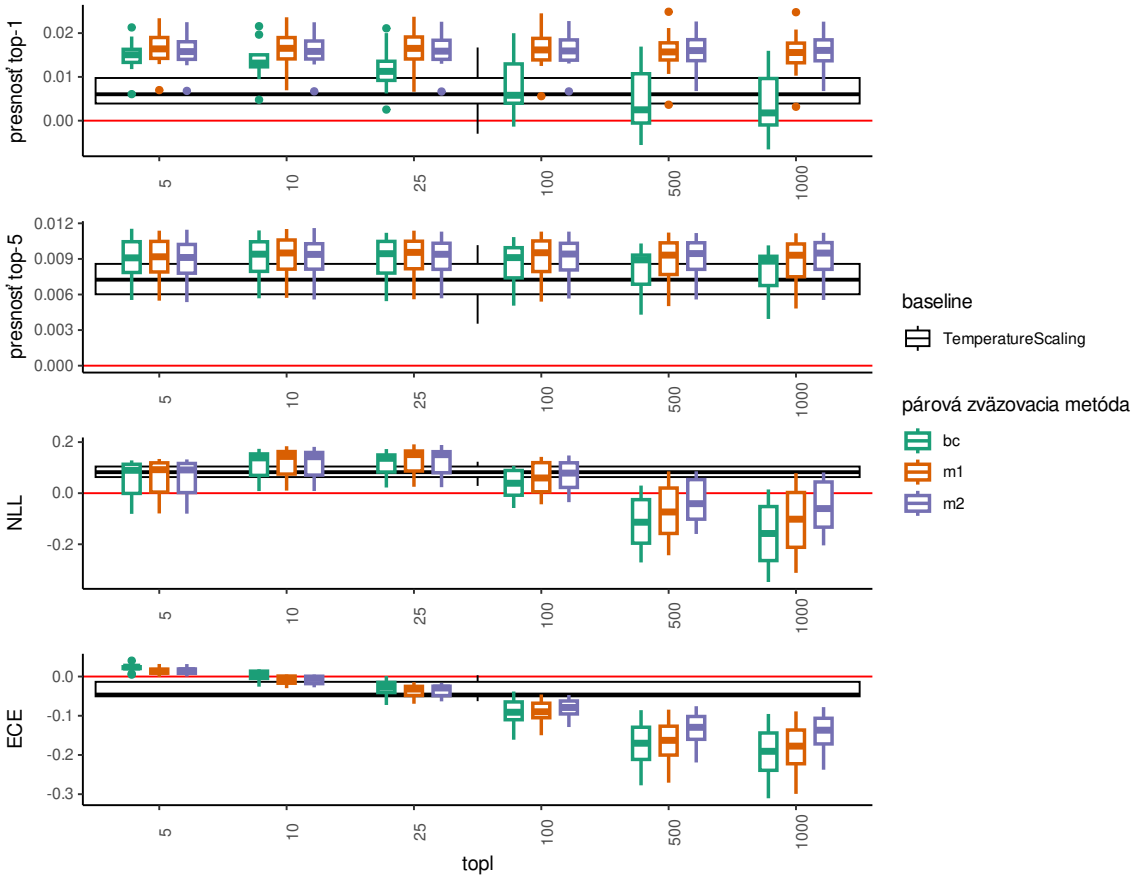


Obrázok 7.36: Čas predikcie ansámblu, bez času inferencie členov ansámblu, pre rôzne hodnoty hyperparametra *topl*. Zobrazené hodnoty sú pre spracovanie celej validačnej množiny (50000 prvkov) na datasete ImageNet1k. Zobrazené konfigurácie využívajú kombinačnú metódu **logreg_no_interc**.

vykonanie predikcie bez úpravy. Aby bolo ladenie hyperparametra korektné, odrezali sme z trénovacej množiny datasetu ImageNet1k dve validačné množiny. Obe množiny obsaujú 50 prvkov pre každú triedu, majú teda 50000 prvkov. Na prvej validačnej množine trénujeme kombinačné metódy a na druhej testujeme kvalitu výstupov WLE pre rôzne hodnoty *topl*. Na rozdiel od predchádzajúcich sekcií pridávame metriku *presnosť top-5*, štandardne používanú pri datasete ImageNet. Táto metrika vyhodnocuje v akej časti testovaných vzoriek sa správna trieda nachádza medzi 5 triedami s najvyššou predikovanou pravdepodobnosťou. Klasickú presnosť tu označujeme ako *presnosť top-1*.

V experimente kombinujeme štyri klasifikátory: B16, S16, M_B16 a R26_S32, ktoré sú bližšie popísané v sekcii 7.2. WLE trénujeme na všetkých aspoň dvojprvkových podmnožinách tejto množiny klasifikátorov. Na základe výsledkov z predchádzajúcich sekcií sme sa rozhodli otestovať kombinačné metódy **cal_average**, **grad_m2** a **logreg_no_interc** v kombinácii s párovými zväzovacími metódami **m1**, **m2** a **bc**. Pre hyperparameter *topl* testujeme hodnoty: 5, 10, 25, 100, 500 a 1000, pričom hodnota 1000 zodpovedá uvažovaniu všetkých tried.

Pre približnú predstavu o časovej úspore s využitím nižších hodnôt hyperparametra *topl* sme vykreslili časy predikcie pre validačnú množinu, teda 50000 prvkov. Tieto časy sú zobrazené na obrázku 7.36. Ako môžeme vidieť, úspora dosahuje dva až tri rády.



Obrázok 7.37: Zlepšenie v sledovaných metrikách oproti najlepšej sieti pre kombinačnú metódu **logreg__no__interc** a pre rôzne hodnoty hyperparametra *topl* na datasete ImageNet1k. Červená čiara predstavuje nulové zlepšenie a čierny boxplot reprezentuje výsledky baseline ansámblovej metódy.

Výsledky sme vyhodnocovali v kontexte zlepšenia oproti najlepšej z kombinovaných sietí. Výsledok pre kombinačnú metódu **logreg__no__interc** je zobrazený na grafe 7.37, grafy pre ostatné kombinačné metódy sú v prílohe. Z grafu môžeme vidieť pre párovú zväzovaciu metódu **bc** klesajúcu presnosť so zvyšujúcou sa hodnotou *topl*. Pre metriky NLL a ECE môžeme pozorovať podobný pokles u všetkých párových zväzovacích metód. Pri metrike NLL je viditeľné mierne zlepšenie medzi hodnotami *topl* 5 a 10. Na základe zobrazených výsledkov volíme pre kombinačnú metódu **logreg__no__interc** hodnotu hyperparametra *topl* 10.

Na základe podobných úvah, sme vybrali hodnoty *topl* aj pre ostatné kombinačné metódy. Pre všetky tri testované kombinačné metódy nám ako najvhodnejšia znížená hodnota *topl* vyšla 10. Predikciu s vybranými zníženými hodnotami *topl* ďalej označu-

jeme ako stratégia *fast*. Každú WLE konfiguráciu testujeme aj s *topl* hodnotou 1000, tieto výsledky označujeme ako stratégia *full*.

Výsledok experimentov v sekcii: Pre všetky tri testované kombinačné metódy sme zvolili ako vhodnú zníženú hodnotu *topl* 10.

7.9 Vyhodnotenie na datasete ImageNet

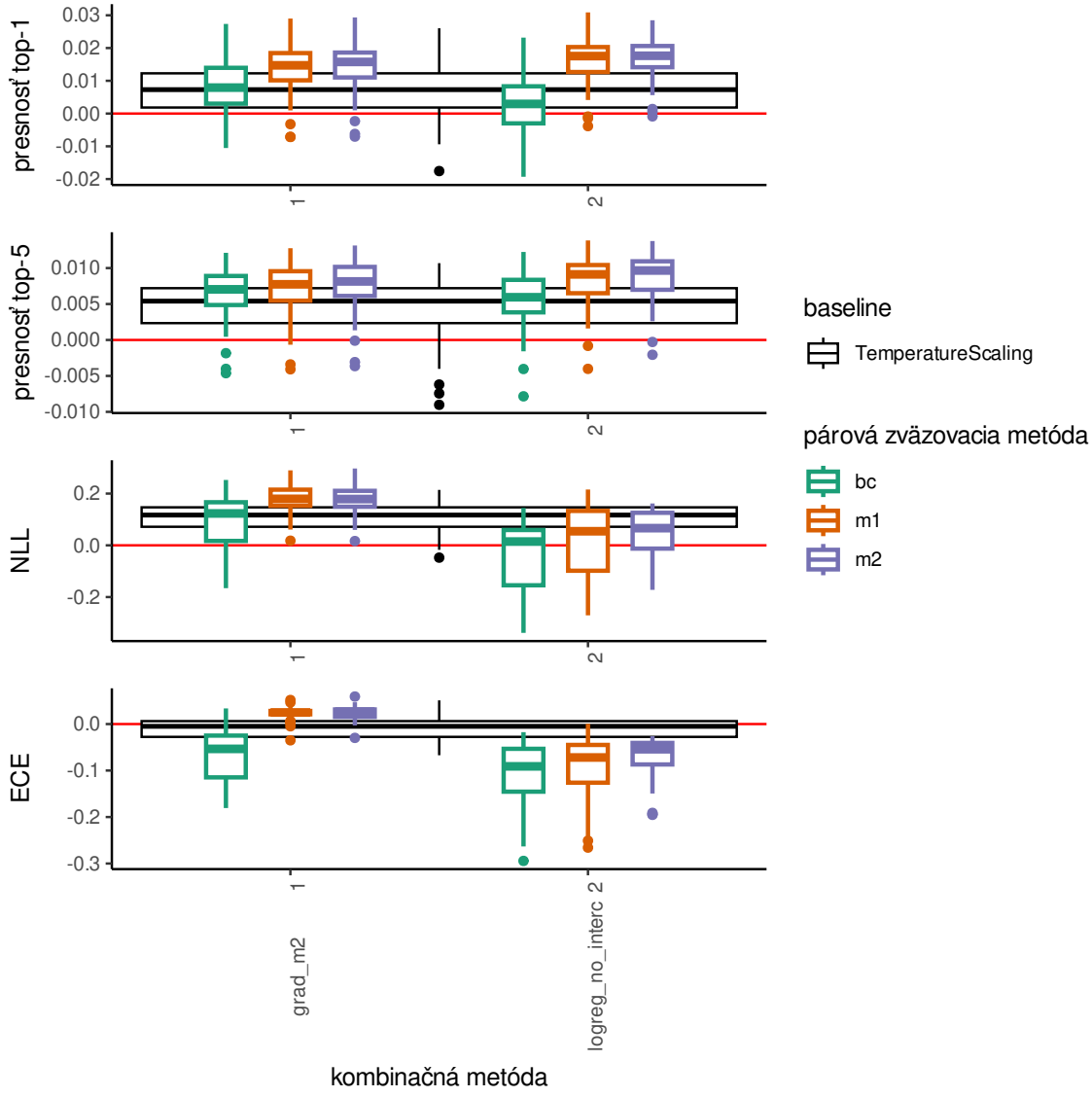
Cieľ experimentov v sekcii: Vyhodnotiť kvalitu vybraných konfigurácií WLE ansámblu s odladenými hyperparametrami na datasete ImageNet1k.

Pomocou experimentov v predchádzajúcich sekciách sme vytvorili trénovaciu metodológiu WLE a stanovili sme vhodné hodnoty pre jeho hyperparametre. Vybrali sme tiež konfigurácie kombinačných a párových zväzovacích metód, ktoré dávali z hľadiska sledovaných metrík najlepšie výsledky. V tejto sekcii testujeme vybrané konfigurácie WLE na datasete ImageNet1k, pričom výsledky vyhodnocujeme na oficiálnej validačnej množine. (Pozn. testovacia množina pre tento dataset nie je verejne dostupná, na jej mieste sa používa validačná množina.)

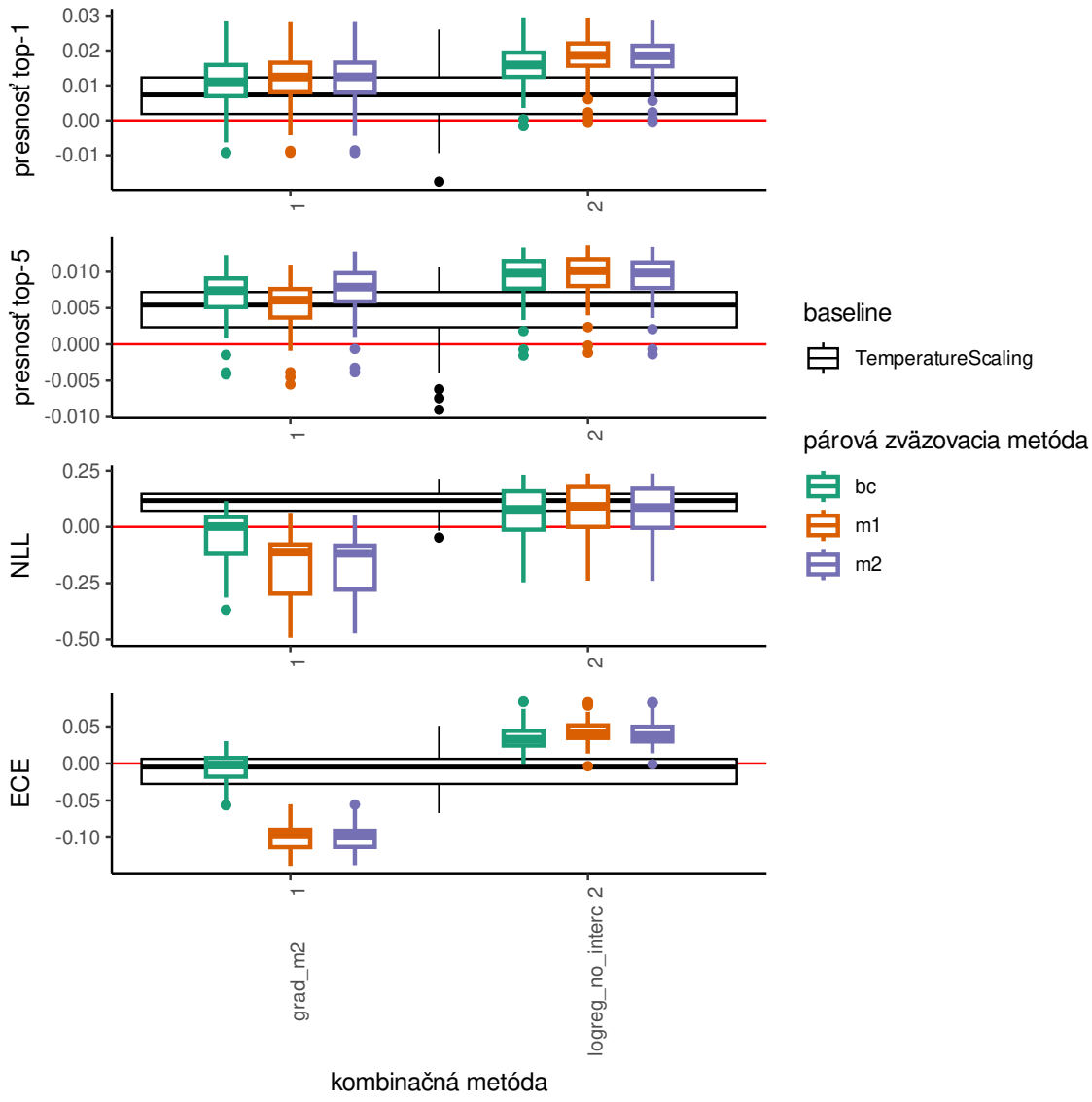
Ku klasifikátorom z predchádzajúcej sekcie sme pridali ďalšie dva, ansámble teda budujeme zo šiestich klasifikátorov B16, S16, M_B16, R26_S32, B32 a Ti16. Testujeme konfigurácie zložené z kombinačných metód **grad_m2** a **logreg_no_interc** a z párových zväzovacích metód **bc**, **m1** a **m2**. Ansámble vytvárame s využitím všetkých aspoň dvojprvkových podmnožín kombinovaných klasifikátorov.

Na základe výsledkov z predchádzajúcej sekcie tu vyhodnocujeme dve verzie predikcie a to *fast* a *full*. Súhrnné výsledky zlepšenia oproti najlepšej kombinovanej sieti pre verziu *full* sú na obrázku 7.38 a pre verziu *fast* na obrázku 7.39. Z grafu 7.38 pre verziu *full* môžeme vidieť dobré výsledky pre konfigurácie **grad_m2 + m2** a **logreg_no_interc + m1**, ktoré sme skúmali pri testovaní na datasete CIFAR-100 v sekcii 7.6. Pri oboch kombinačných metódach sú medzi párovými zväzovacími metódami **m1** a **m2** malé rozdiely. Párová zväzovacia metóda **bc** dosahuje o niečo horšie výsledky.

Na grafe 7.39 pre verziu *fast* vidíme pre kombinačnú metódu **logreg_no_interc** malé rozdiely medzi jednotlivými párovými zväzovacími metódami. Pre kombinačnú metódu **grad_m2** sú rozdiely výraznejšie a jednotlivé konfigurácie sa správajú odlišne



Obrázok 7.38: Zlepšenie v sledovaných metrikách oproti najlepšej z kombinovaných sietí na datasete ImageNet1k s prístupom predikcie *full*.



Obrázok 7.39: Zlepšenie v sledovaných metrikách oproti najlepšej z kombinovaných sietí na datasete ImageNet1k s prístupom predikcie *fast*.

ako s prístupom *full*. Pre metriky NLL a ECE pri predikčnom prístupe *fast* dosahuje párová zväzovacia metóda **bc** lepšie výsledky ako párové zväzovacie metódy **m1** a **m2**. Pri predikčnom prístupe *full* tomu bolo naopak. Pre metriky presnoti dosahuje metóda **bc** podobné výsledky ako **m2**.

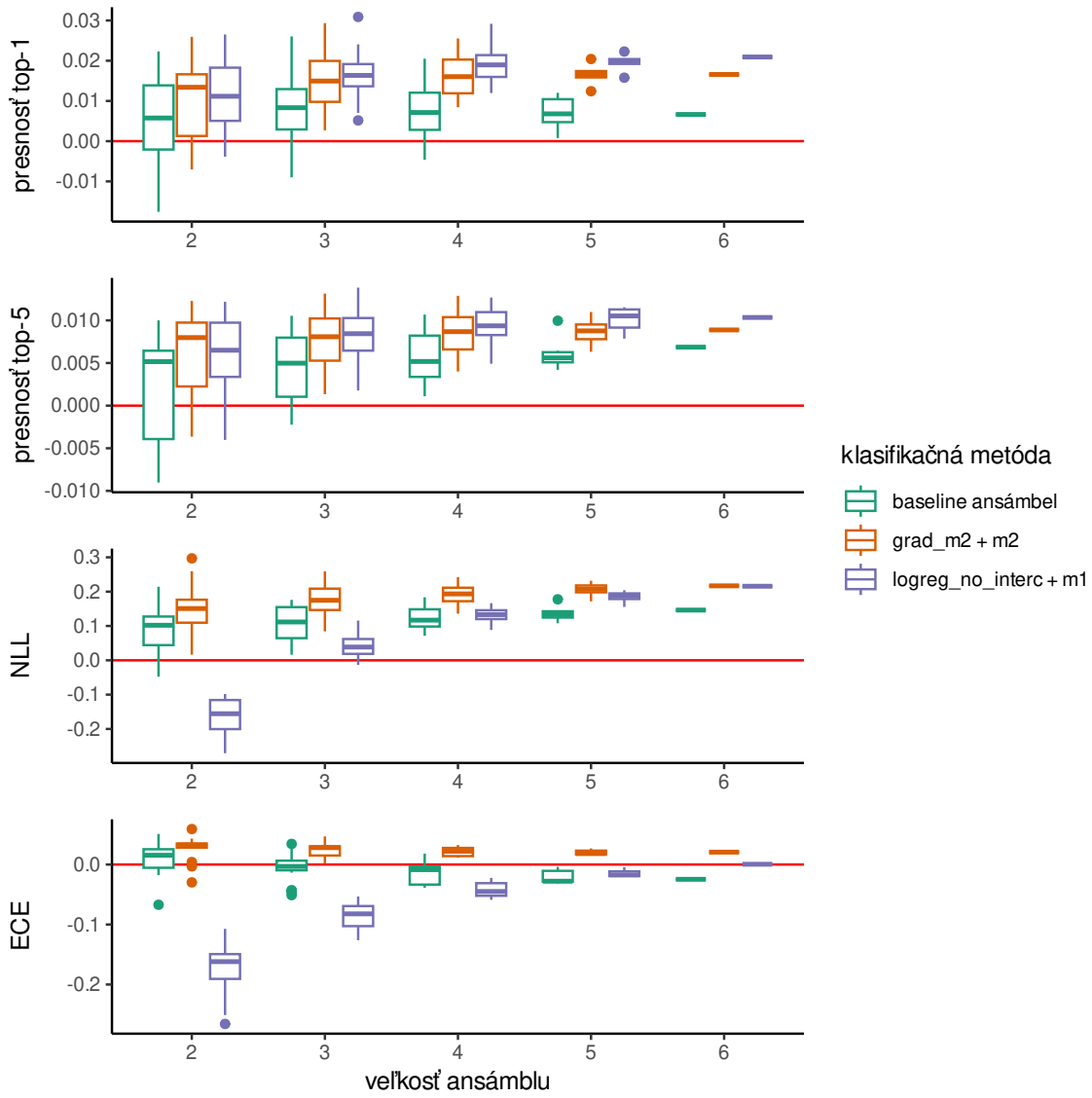
Pre konzistentnosť budeme pre obe stratégie predikcie ďalej analyzovať konfigurácie **grad_m2 + m2** a **logreg_no_interc + m1**. Pre tieto konfigurácie vyšetríme správanie WLE v závislosti od veľkosti ansámblu.

Graf pre jednotlivé veľkosti ansámblov so stratégiou predikcie *full* je na obrázku 7.40. Baseline ansámbel dosahuje vo všetkých metrikách pre rôzne veľkosti ansámblu prevažne stabilné výsledky. Pre metriku presnosť top-1 a v menšej miere aj pre presnosť top-5 môžeme pozorovať s rastúcou veľkosťou ansámblu zväčšujúcu sa prevahu WLE konfigurácií nad baseline ansámblom. Podobné správanie je vidieť aj pri metrikách NLL a ECE, tu ale konfigurácia **logreg_no_interc + m1** začína pri malých veľkostiach ansámblu so zhoršením oproti najlepšej sieti.

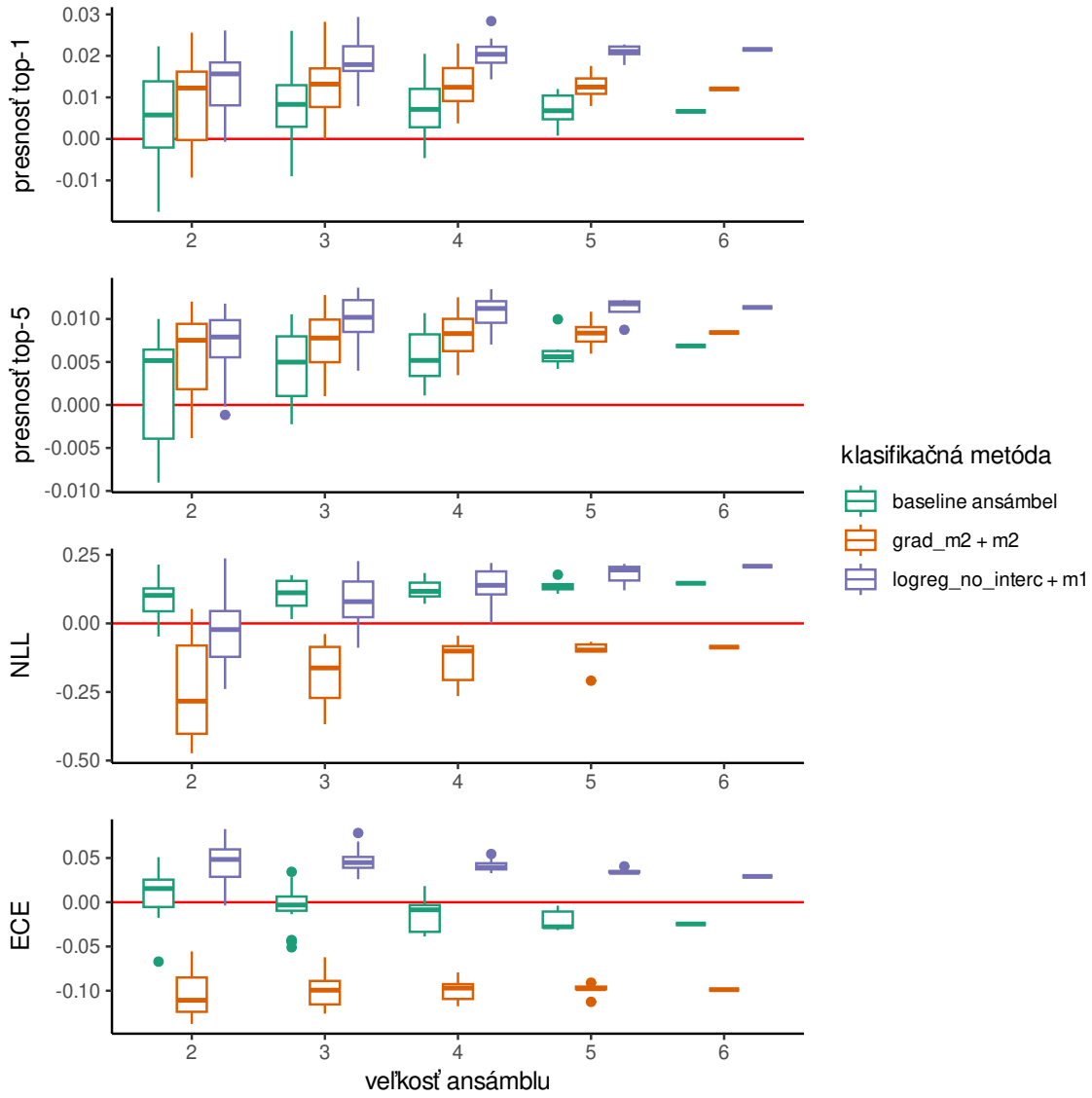
Na grafe 7.41 je rovnaké zobrazenie pre stratégiu predikcie *fast*. Na tomto grafe pozorujeme pre metriky presnosti podobné správanie ako pri stratégii *full*. Pre metriky NLL a ECE sa správanie WLE metód zmenilo. Konfigurácia **logreg_no_interc + m1** dosahuje stabilné zlepšenie oproti najlepšej sieti pre väčšinu prípadov okrem metriky NLL pre veľkosť ansámblu 2. Konfigurácia **grad_m2 + m2** nedosahuje v týchto metrikách zlepšenie oproti najlepšej sieti.

Z týchto pozorovaní vidíme, že jednotlivé WLE konfigurácie sa správajú pre rôzne veľkosti ansámblu odlišne. Štatistické porovnanie s baseline ansámblom preto vykonáme pre jednotlivé veľkosti ansámblu osobitne. Pre veľkosti ansámblu 5 a 6 nemáme dostatok vzoriek na vykonanie štatistických testov. Pre lepšie vizuálne porovnanie týchto prípadov vykresľujeme zlepšenie v jednotlivých metrikách, ktoré dosiahli WLE konfigurácie oproti baseline ansámblu. Ide o hodnoty, ktoré sú štandardne spracované v párových štatistických testoch. Zobrazenie pre stratégiu predikcie *full* je na obrázku 7.42 a pre stratégiu *fast* na obrázku 7.43. Vo vizuálnom hodnotení sa obmedzíme na veľkosti ansámblu 5 a 6, ostatné veľkosti budú vyhodnotené pomocou štatistických testov.

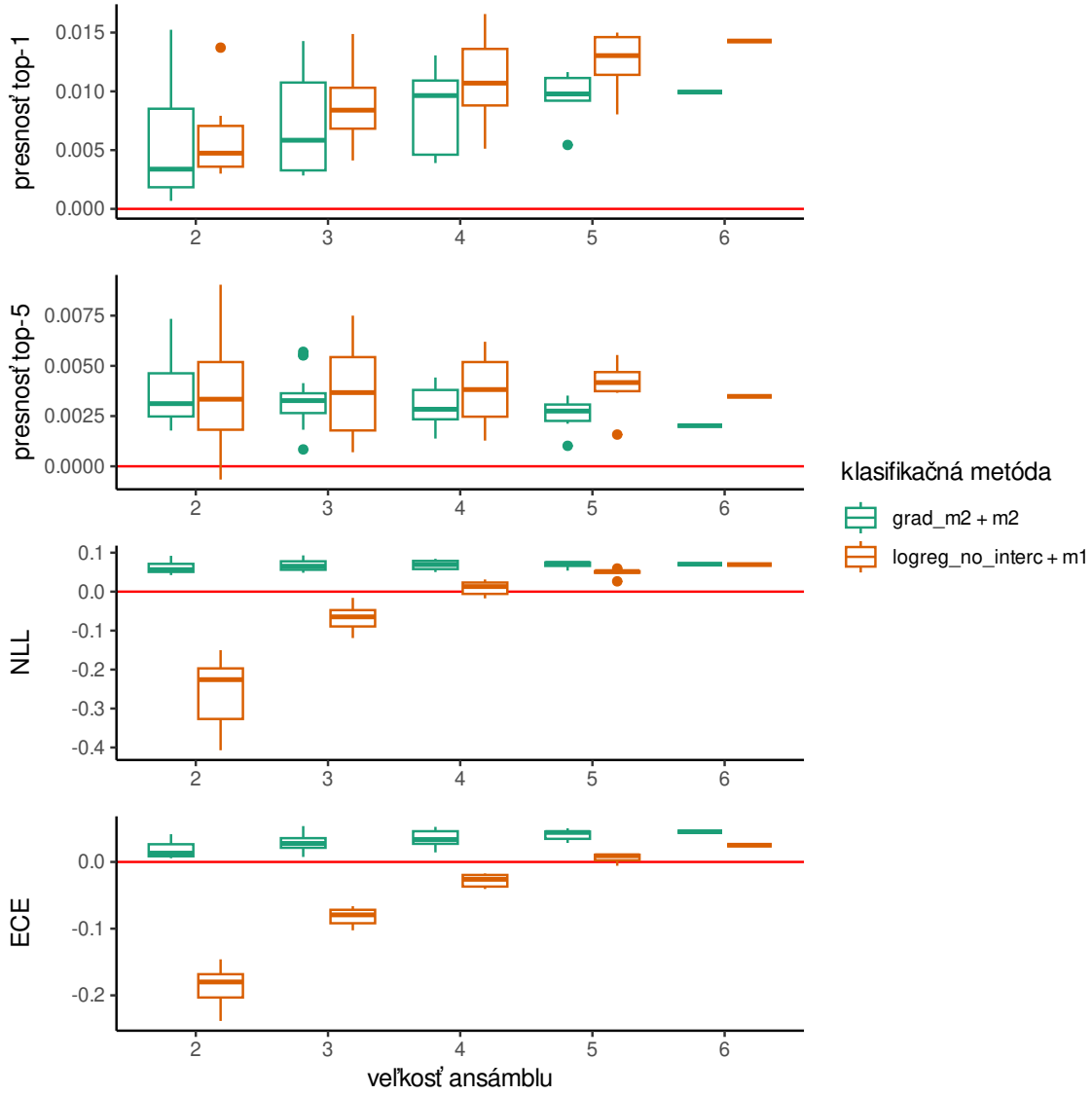
Pre stratégiu predikcie *full* je zlepšenie jasne viditeľné vo všetkých prípadoch okrem konfigurácie **logreg_no_interc + m1** pre veľkosť ansámblu 5 pri metrike NLL, kde



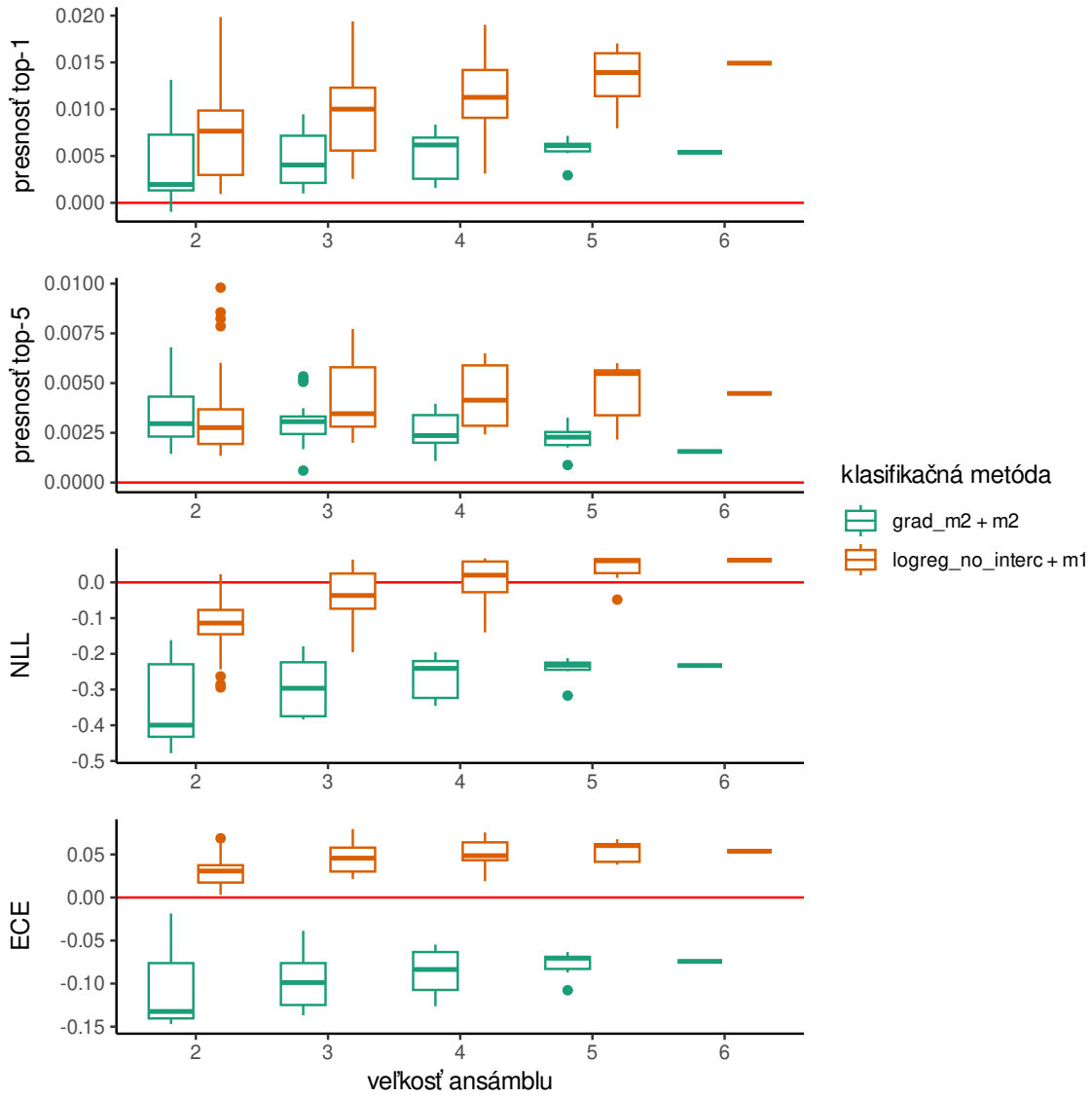
Obrázok 7.40: Zlepšenie v sledovaných metrikách oproti najlepšej z kombinovaných sietí pre dve vybrané WLE konfigurácie v závislosti od veľkosti ansámblu na datasete ImageNet1k s prístupom predikcie *full*.



Obrázok 7.41: Zlepšenie v sledovaných metrikách oproti najlepšej z kombinovaných sietí pre dve vybrané WLE konfigurácie v závislosti od veľkosti ansámblu na datasete ImageNet1k s prístupom predikcie *fast*.



Obrázok 7.42: Zlepšenie v sledovaných metrikách oproti baseline ansámblu pre dve vybrané WLE konfigurácie v závislosti od veľkosti ansámblu na datasete ImageNet1k s prístupom predikcie *full*.



Obrázok 7.43: Zlepšenie v sledovaných metrikách oproti baseline ansámblu pre dve vybrané WLE konfigurácie v závislosti od veľkosti ansámblu na datasete ImageNet1k s prístupom predikcie *fast*.

táto WLE konfigurácia dosahuje podobné výsledky ako baseline.

Pre stratégiu predikcie *fast* WLE konfigurácia **grad_m2 + m2** nedosahuje zlepšenie oproti baseline ansámblu v metrikách NLL a ECE. V ostatných prípadoch dosahujú WLE ansámble jasne viditeľné zlepšenie.

Výsledky štatistických testov pre dve vybrané WLE konfigurácie a veľkosti ansámblu 2, 3 a 4 sú zobrazené v tabuľke 7.11 pre stratégiu predikcie *full* a v tabuľke 7.12 pre stratégiu predikcie *fast*. Podobne ako pri vyhodnotení na datasete CIFAR-100, aj tu sme použili párový permutačný test [99]. Výsledky sme vyhodnotili na hladine významnosti 5%.

Z výsledkov je vidieť, že pri predikčnom prístupe *full* WLE konfigurácia **grad_m2 + m2** vyhráva vo všetkých metrikách pre všetky veľkosti ansámblu. WLE konfigurácia **logreg_no_interc + m1** má pri predikčnom prístupe *full* problémy s metrikami NLL a ECE a to najmä pre menšie veľkosti ansámblov. V metrikách presnosti prekonáva baseline ansámbel.

Pri predikčnom prístupe *fast* WLE konfigurácia **grad_m2 + m2** prekonáva baseline ansámbel len v metrikách presnosti. Pre metriky NLL a ECE nedosahuje lepšie výsledky ako baseline pre žiadnu veľkosť ansámblu. WLE konfigurácia **logreg_no_interc + m1** dosahuje pri predikčnom prístupe *fast* lepšie výsledky. Táto konfigurácia vyhráva nad baseline ansámblom vo všetkých metrikách okrem NLL pri veľkostiach ansámblu 2 a 3, kde výsledky vychádzajú v prospech baseline ansámblu a nerozhodne.

Na základe týchto výsledkov odporúčame použitie WLE konfigurácie **grad_m2 + m2** s predikčnou stratégiou *full* v prípade ak pracujeme s menším ansámblom veľkosti 2, alebo 3 a záleží nám na metrike NLL. V ostatných prípadoch odporúčame použitie WLE konfigurácie **logreg_no_interc + m1** s predikčným prístupom *fast*.

Výsledky štatistických testov pre ostatné WLE konfigurácie sú dostupné v prílohe. Tu poskytneme len zhrnutie získaných výsledkov. Pri oboch predikčných prístupoch a pre obe testované kombinačné metódy nespôsobujú zámeny párových zväzovacím metód **m1** za **m2** a naopak takmer žiadne zmeny vo výsledkoch štatistických testov. Použitie párovej zväzovacej metódy **bc** spôsobuje zhoršenie výsledkov najmä pre metriky NLL a ECE, v niektorých prípadoch aj pre presnosť.

Výpočtová náročnosť trénovania ako aj predikcie s predikčným prístupom *full* je

Tabuľka 7.11: Výsledky štatistických testov porovnania vybraných WLE konfigurácií a baseline ansámblu na datasete ImageNet1k pre stratégiu predikcie *full*. Výsledky sú vyhodnotené na hladine významnosti 5%.

WLE konfigurácia	**metrika**	**veľkosť ansámblu**	**p-hodnota**	**lepšia metóda**
grad_m2 + m2	presnosť top-1	2	0.0002	WLE
		3	0.0000	WLE
		4	0.0000	WLE
	presnosť top-5	2	0.0002	WLE
		3	0.0001	WLE
		4	0.0000	WLE
	NLL	2	0.0001	WLE
		3	0.0000	WLE
		4	0.0001	WLE
	ECE	2	0.000	WLE
		3	0.0000	WLE
		4	0.0001	WLE
logreg_no_interc + m1	presnosť top-1	2	0.0002	WLE
		3	0.0000	WLE
		4	0.0002	WLE
	presnosť top-5	2	0.0003	WLE
		3	0.0000	WLE
		4	0.0002	WLE
	NLL	2	0.0001	baseline
		3	0.0000	baseline
		4	0.0458	WLE
	ECE	2	0.0000	baseline
		3	0.0000	baseline
		4	0.0000	baseline

Tabuľka 7.12: Výsledky štatistických testov porovnania vybraných WLE konfigurácií a baseline ansámblu na datasete ImageNet1k pre stratégiu predikcie *fast*. Výsledky sú vyhodnotené na hladine významnosti 5%.

WLE konfigurácia	**metrika**	**veľkosť ansámblu**	**p-hodnota**	**lepšia metóda**
grad_m2 + m2	presnosť top-1	2	0.0001	WLE
		3	0.0000	WLE
		4	0.0001	WLE
	presnosť top-5	2	0.0001	WLE
		3	0.0000	WLE
		4	0.0000	WLE
	NLL	2	0.0000	baseline
		3	0.0000	baseline
		4	0.0000	baseline
	ECE	2	0.0001	baseline
		3	0.0000	baseline
		4	0.0000	baseline
logreg_no_interc + m1	presnosť top-1	2	0.0001	WLE
		3	0.0000	WLE
		4	0.0001	WLE
	presnosť top-5	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0000	WLE
	NLL	2	0.0022	baseline
		3	0.897	-
		4	0.0015	WLE
	ECE	2	0.0000	WLE
		3	0.0000	WLE
		4	0.0001	WLE

pre PWE ansámbel kvadratická v počte tried riešenej úlohy. To môže predstavovať obmedzenie pri riešení veľkých úloh. Pri našom experimente sa ukázalo, že pri riešení klasifikačnej úlohy s 1000 triedami zostáva čas tréningu aj predikcie na praktických hodnotách. Experimenty sme vykonávali s použitím grafického akcelerátora NVIDIA GeForce RTX 2080 Ti. Dotrénovanie neurónových sietí bolo realizované s použitím dvoch takýchto akcelerátorov, na výpočty ansámblu bol použitý jeden.

Dotrénovanie najväčšej použitej neurónovej siete B16 trvalo viac ako 29 hodín. Inferencia pre 50000 obrázkov validačnej množiny trvala s tou istou sieťou približne 609 sekúnd. Časy tréningu a predikcie PWE, ktoré ďalej diskutujeme nezahŕňajú čas inferencie neurónovými sieťami, len samotné spracovanie výstupov sietí ansámblom.

Tréning kombinačnej metódy **grad_m2** pre ansámbel veľkosti 5 trval približne 30 minút. Pre kombinačnú metódu **logreg_no_interc** trval ten istý tréning približne 14 sekúnd. Ako môžeme vidieť, pri uvažovaní tréningu niekoľkých neurónových sietí ako členov ansámblu zaberá tréning samotného PWE ansámblu len malý zlomok celkového trénovacieho času.

Predikcia pre 50000 prvkov validačnej množiny pri použití konfigurácie **logreg_no_interc + m1** a predikčného prístupu *full* trvala pre ansámbel veľkosti 5 približne 160 sekúnd. Rovnaká predikcia s použitím predikčného prístupu *fast* trvala približne 1.3 sekundy. Podobne ako pri tréningu, aj tu môžeme vidieť, že spracovanie ansámblom predstavuje len malú časť celej inferencie.

Výsledok experimentov v sekcii: Na základe vykonaných experimentov a štatistických testov sme vybrali dve konfigurácie WLE, ktoré konzistentne prekonávajú baseline ansámbel v sledovaných metrikách. Konfigurácia **grad_m2 + m2** má lepšie správanie pri predikčnom prístupe *full* a prekonáva baseline ansámbel vo všetkých sledovaných metrikách pre všetky veľkosti ansámblov. Konfigurácia **logreg_no_interc + m1** má lepšie správanie pri predikčnom prístupe *fast*. V prípade ansámblov veľkosti 2 a 3 neprekonala baseline ansámbel v metrike NLL, v ostatných prípadoch dosiahla lepšie výsledky ako baseline ansámbel. Vo všeobecnosti odporúčame konfiguráciu **logreg_no_interc + m1** s predikčným prístupom *fast* pre rýchlejší čas jej trénovania aj predikcie a taktiež pre väčšie zlepšenie presnosti oproti konfigurácii **grad_m2 + m2**.

Kapitola 8

Záver

V práci sme sa venovali klasifikácii obrazu pomocou strojového učenia. V prvých kapitolách poskytujeme prehľad existujúcich metód. Začíname jednoduchými klasifikačnými metódami, ktoré nie sú špeciálne prispôsobené klasifikácii obrazu, ale majú význam pre ansámblovú metódu, ktorú navrhujeme. Ďalej sa venujeme klasifikačným metódam špeciálne určeným na klasifikáciu obrazu. Kladieme dôraz na najnovšie metódy, preto zahŕňame moderné architektúry konvolučných neurónových sietí a obrazové transformery. V ďalších kapitolách skúmame aktuálny stav ansámblových postupov v klasifikácii a získavame poznatky, ktoré neskôr využijeme pri vytváraní novej ansámblovej klasifikačnej metódy. Zvláštnu pozornosť venujeme párovým ansámblovým modelom, ktoré využívame aj v nami navrhnutej ansámblovej metóde Vážený lineárny ansámbel (WLE).

Hlavný cieľ práce

V kapitole 6 navrhujeme novú ansámblovú klasifikačnú metódu, ktorá využíva binarizáciu, lineárne klasifikačné metódy a metódy využívané v párových ansámbloch. Navrhujeme viacero verzií tejto metódy, niektoré z nich vyžadujú trénovanie. V kapitole 7 popisujeme experimenty a výsledky prostredníctvom ktorých sme vytvorili metodológiu trénovania a použitia jednotlivých verzií vrátane odladenia ich hyperparametrov. Tiež tu porovnávame jednotlivé verzie navrhnutej metódy na troch datasetoch s počtom tried 10, 100 a 1000. Prostredníctvom týchto experimentov sme splnili hlavný cieľ práce, ktorý sa skladá z nasledujúcich bodov.

Efektívne implementovať navrhnutú parametrickú párovú kombinačnú metódu.

Metódu WLE sme implementovali kompletne pomocou tenzorových operácií, vrátane procesu trénovania kombinačných koeficientov. Takáto implementácia umožňuje použiť WLE aj na dataset s 1000 triedami aj napriek kvadratickej zložitosti vzhľadom k počtu tried pri trénovaní. Takéto použitie sme demonštrovali v sekcii 7.9. Navrhli sme tiež rozšírenie, ktoré umožňuje zrýchliť proces predikcie pri datasete s 1000 triedami až o dva rády s minimálnym dopadom na kvalitu získanej predikcie. Toto rozšírenie je popísané v sekcii 6.3 a otestované v sekcii 7.8.

Pre trénovanie na datasete s 1000 triedami vyžaduje vytvorená implementácia kvalitný grafický akcelerátor, pretože je pomerne pamäťovo náročná. Pre trénovanie ansámblu na úlohe s 1000 triedami pri kombinovaní viac ako štyroch klasifikátorov metóda vyžaduje aspoň 20GB grafickej operačnej pamäte.

Zvoliť vhodnú metódu na vytváranie diverzity použiteľnú pre konvolučné neurónové siete.

V experimentoch sa nám osvedčilo kombinovať rôzne architektúry neurónových sietí. V priebehu posledných rokov si získal veľkú pozornosť výskum v oblasti neurónových sietí nazývaných transformery. Architektúra transformerov bola vo forme obrazových transformerov prispôsobená na prácu s obrazovými dátami a teda aj na klasifikáciu obrazu. Okrem rôznych konvolučných neurónových sietí využívame preto aj niekoľko obrazových transformerov.

Okrem klasického trénovania neurónových sietí využívame aj metódu prenosu učenia. Pri tejto metóde sú predtrénované modely neurónových sietí na cieľovej úlohe len dotrénované, čo vyžaduje menej dát a kratší trénovací čas.

Zostaviť a otestovať metodiku trénovania kombinačnej metódy.

Vo finálnej podobe metódy WLE sme navrhli niekoľko alternatívnych konfigurácií, všetky z nich vyžadujú trénovanie. Metodiku ich trénovania sme experimentálne budovali v sekcii 7.3. Ako najvhodnejšie sa ukázalo používať trénovaciu množinu veľkosti 50 vzoriek na triedu riešenej klasifikačnej úlohy, pozostávajúcu zo vzoriek, ktoré neboli použité pri trénovaní členov ansámblu.

Otestovať správanie zostavenej ansámblovej metódy na vhodne zvolených datasetoch.

Metódu WLE sme otestovali na troch datasetoch rôznej veľkosti. V testoch na datasetoch CIFAR-10 a CIFAR-100 zdokumentovaných v sekcii 7.4 sme porovnávali jednotlivé konfigurácie WLE. Cieľom týchto testov bolo obmedzenie konfigurácií WLE vstupujúcich do ďalších experimentov len na tie, ktoré dávajú dobré výsledky.

Následne sme na datasete CIFAR-100 v sekcii 7.6 porovnávali vybrané konfigurácie s populárnou ansámblovou metódou priemeru kalibrovaných predikcií. Výsledky štatistických testov ukázali, že metóda WLE dosahuje lepšie výsledky ako porovnávaný baseline ansámbel takmer vo všetkých testovaných prípadoch. Najlepšie výsledky dosahovali kombinačné metódy **logreg_no_interc** a **grad_m2** v konfiguráciách s párovými zväzovacími metódami **m1** a **m2**.

S rovnakým baseline ansámblom sme metódu WLE porovnávali aj na datasete ImageNet1k v sekcii 7.9. V tomto prípade sme testovali aj úpravu predikčného procesu WLE umožňujúcu rýchlejšiu predikciu. Konfigurácie WLE opäť prekonali baseline ansámbel vo väčšine testovaných prípadov. Najlepšie výsledky dosahovali tie isté konfigurácie ako v testoch na datasete CIFAR-100. Pri použití s úpravou pre zrýchlenú predikciu mala lepšie správanie kombinačná metóda **logreg_no_interc**.

Vo vykonaných experimentoch sme pozorovali dobré vlastnosti baseline ansámblu s ktorým sme metódu WLE porovnávali. Napriek svojej jednoduchosti dosahoval baseline ansámbel zlepšenie v presnosti oproti najlepšej z kombinovaných sietí takmer vo všetkých testovaných prípadoch.

Vedľajší cieľ práce

Párové zväzovacie metódy umožňujú kvantifikovať nesúlad medzi kombinovanými predikciami, čo dáva možnosť využiť ich na detekciu neznámych vzoriek a vytvorenie klasifikátora schopného zdržať sa predikcie. V sekcii 7.7 sme pre našu ansámblovú metódu otestovali aj takéto rozšírenie a porovnali sme ho so štandardne využívanou metódou detekcie neznámych vzoriek MSP. Tieto experimenty napĺňajú vedľajší cieľ práce zložený z nasledovných bodov.

Navrhnúť a implementovať funkcionalitu umožňujúcu vytvorenému klasifikátoru zdržať sa klasifikácie.

Teoretický popis tohto rozšírenia je dostupný v sekcii 6.2. Implementovali sme ho pre párové zväzovacie metódy **m1**, **m2** a **bc**.

Otestovať implementovanú funkcionalitu na vhodne zvolených úlohách.

Funkcionalitu zdržania sa klasifikácie, resp. detekcie neznámych vzoriek sme otestovali na dvoch úlohách využívajúcich datasety CIFAR-10 a CIFAR-100. Obe tieto úlohy trénujú klasifikátor na jednom z datasetov CIFAR a následne testujú jeho schopnosť rozoznať vzorky z testovacej množiny datasetu na ktorom bol trénovaný od vzoriek z testovacej množiny druhého CIFAR datasetu. Popis experimentov a výsledkov je dostupný v sekcii 7.7. Implementovanú funkcionalitu sme prostredníctvom metrík AUROC a AUPRC porovnávali s populárnou metódou MSP. Ukázalo sa, že neistota z párových zväzovacích metód neposkytuje dostatočne vhodné informácie na vykonávanie OOD detekcie. Pri experimentoch sme ale zistili, že aplikovanie metódy MSP na výstupy WLE poskytuje lepšiu OOD detekciu, ako aplikovanie MSP na výstupy baseline ansámblu.

Diskusia

Za hlavnú limitáciu pri použití WLE metódy považujeme potrebu oddelenej trénovacej množiny pre trénovanie kombinačných metód. Pri trénovaní metód hlbokého strojového učenia je spravidla potrebné odladiť hodnoty niekoľkých hyperparametrov na oddelenej validačnej množine. Po odladení hyperparametrov je možné pre dosiahnutie čo najlepších výsledkov znovu natrénovať metódu hlbokého učenia s použitím získaných hodnôt hyperparametrov na kompletnej trénovacej množine zahŕňajúcej aj validačnú množinu. Otestovanie podobného postupu pre použitie WLE metódy je jedným z možných smerovaní ďalšieho výskumu.

Pri teoretickej analýze v sekcii 6.4 sme zistili dobré teoretické vlastnosti WLE metódy, ktoré by sa mohli prejaviť pri kombinovaní klasifikátorov poskytujúcich komplementárne informácie. Analýza bola vykonaná po dokončení experimentov v práci, preto návrh vykonaných experimentov tieto vlastnosti nevyužíva. Vhodným spôso-

bom na otestovanie týchto vlastností by mohla byť klasifikácia multimodálnych dát, pri ktorej sa využíva kombinovanie klasifikátorov trénovaných na odlišných skupinách príznakov [84]. Klasifikácia multimodálnych dát je preto ďalším možným smerovaním budúceho výskumu.

Architektúra WLE metódy technicky umožňuje kombinovanie klasifikátorov zameraných na odlišné podmnožiny tried riešeného problému. Bolo by teda možné identifikovať problematické podmnožiny tried a natrénovať klasifikátory, ktoré by sa na ne špecializovali. S použitím WLE by takéto klasifikátory mohli byť kombinované so štandardnými klasifikátormi zameranými na celý problém. Takéto použitie metódy WLE je taktiež možným smerovaním ďalšieho výskumu.

Zoznam použitej literatúry

[1] R. A. FISHER. „The use of multiple measurements in taxonomic problems“. In: *Annals of Eugenics* 7.2 (1936), s. 179–188. DOI: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

[2] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984. ISBN: 9781315139470. DOI: https://doi.org/10.1201/9781315139470.

[3] Bernhard Boser, Isabelle Guyon a Vladimir Vapnik. „A Training Algorithm for Optimal Margin Classifier“. In: *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory* 5 (aug. 1996), s. 144–152. DOI: 10.1145/130385.130401.

[4] Sepp Hochreiter a Jürgen Schmidhuber. „Long Short-term Memory“. In: *Neural computation* 9 (dec. 1997), s. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[5] Y. LeCun et al. „Backpropagation Applied to Handwritten Zip Code Recognition“. In: *Neural Computation* 1.4 (dec. 1989), s. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541.

[6] Franco Scarselli et al. „The Graph Neural Network Model“. In: *IEEE Transactions on Neural Networks* 20.1 (2009), s. 61–80. DOI: 10.1109/TNN.2008.2005605.

[7] Maurice Vogels et al. „P. F. Verhulst's “notice sur la loi que la populations suit dans son accroissement” from correspondence mathematique et physique. Ghent, vol. X, 1838“. In: *Journal of Biological Physics* 3.4 (dec. 1975), s. 183–192. ISSN: 1573-0689. DOI: 10.1007/BF02309004.

[8] Trevor Hastie, Robert Tibshirani a Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. 2. vyd. Springer Science & Business Media, 2009. ISBN: 978-0-387-84858-7. DOI: https://doi.org/10.1007/978-0-387-84858-7.

[9] Gareth James et al. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN: 1461471370.

[10] John Platt et al. „Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods“. In: *Advances in large margin classifiers* 10.3 (1999), s. 61–74.

[11] Trevor Hastie, Robert Tibshirani et al. „Classification by pairwise coupling“. In: *Annals of statistics* 26.2 (1998), s. 451–471. DOI: https://doi.org/10.1214/aos/1028144844.

[12] Ting-Fan Wu, Chih-Jen Lin a Ruby C Weng. „Probability estimates for multi-class classification by pairwise coupling“. In: *Journal of Machine Learning Research* 5.Aug (2004), s. 975–1005. DOI: https://dl.acm.org/doi/10.5555/1005332.1016791.

[13] Chih-Chung Chang a Chih-Jen Lin. „LIBSVM: a library for support vector machines“. In: *ACM transactions on intelligent systems and technology (TIST)* 2.3 (2011), s. 1–27. DOI: https://doi.org/10.1145/1961189.1961199.

[14] Warren S McCulloch a Walter Pitts. „A logical calculus of the ideas immanent in nervous activity“. In: *The bulletin of mathematical biophysics* 5.4 (1943), s. 115–133. DOI: https://doi.org/10.1007/BF02478259.

[15] F. Rosenblatt. „The perceptron: a probabilistic model for information storage and organization in the brain.“ In: *Psychological review* 65 6 (1958), s. 386–408. DOI: https://doi.org/10.1037/h0042519.

[16] Alekseĭ Grigor'evich Ivakhnenko a Valentin Grigor'evich Lapa. *Cybernetics and forecasting techniques*. Zv. 8. American Elsevier Publishing Company, 1967.

[17] Paul Werbos. „Beyond regression:" new tools for prediction and analysis in the behavioral sciences“. In: *Ph. D. dissertation, Harvard University* (1974).

[18] G. Cybenko. „Approximation by superpositions of a sigmoidal function“. In: *Mathematics of Control, Signals and Systems* 2 (1989), s. 303–314. DOI: https://doi.org/10.1007/BF02551274.

[19] Kurt Hornik. „Approximation capabilities of multilayer feedforward networks“. In: *Neural Networks* 4.2 (1991), s. 251–257. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(91)90009-T.

[20] Alexey Dosovitskiy et al. „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale“. In: *International Conference on Learning Representations*. 2021.

[21] Sepp Hochreiter. „Untersuchungen zu dynamischen neuronalen Netzen“. In: *Diploma, Technische Universität München* 91.1 (1991).

[22] Bing Xu et al. *Empirical Evaluation of Rectified Activations in Convolutional Network*. 2015. DOI: https://doi.org/10.48550/arXiv.1505.00853.

[23] Glenn W Brier et al. „Verification of forecasts expressed in terms of probability“. In: *Monthly weather review* 78.1 (1950), s. 1–3. DOI: https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

[24] Oludare Isaac Abiodun et al. „State-of-the-art in artificial neural network applications: A survey“. In: *Heliyon* 4.11 (2018), e00938. DOI: https://doi.org/10.1016/j.heliyon.2018.e00938.

[25] Ashish Vaswani et al. „Attention is All you Need“. In: *Advances in Neural Information Processing Systems*. Ed. I. Guyon et al. Zv. 30. Curran Associates, Inc., 2017.

[26] Xiangning Chen, Cho-Jui Hsieh a Boqing Gong. „When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations“. In: *International Conference on Learning Representations*. 2022.

[27] Chuan Guo et al. „On Calibration of Modern Neural Networks“. In: ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, s. 1321–1330. DOI: https://dl.acm.org/doi/10.5555/3305381.3305518.

[28] Rebecca Roelofs et al. „Mitigating bias in calibration error estimation“. In: *arXiv preprint arXiv:2012.08668* (2020). DOI: https://doi.org/10.48550/arXiv.2012.08668.

[29] Kartik Gupta et al. „Calibration of Neural Networks using Splines“. In: *International Conference on Learning Representations*. 2021.

[30] Alex Krizhevsky, Geoffrey Hinton et al. „Learning multiple layers of features from tiny images“. In: *Technical report* (2009).

[31] Antonio Torralba, Rob Fergus a William T. Freeman. „80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.11 (2008), s. 1958–1970. DOI: 10.1109/TPAMI.2008.128.

[32] Christiane Fellbaum a George Miller. *WordNet : an electronic lexical database*. 1998. ISBN: 9780262272551.

[33] Jia Deng et al. „ImageNet: A large-scale hierarchical image database“. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, s. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[34] Olga Russakovsky et al. „ImageNet Large Scale Visual Recognition Challenge“. In: *International Journal of Computer Vision* 115 (2015), s. 211–252. DOI: https://doi.org/10.1007/s11263-015-0816-y.

[35] „Confident Learning: Estimating Uncertainty in Dataset Labels“. In: *J. Artif. Int. Res.* 70 (máj 2021), s. 1373–1411. ISSN: 1076-9757. DOI: 10.1613/jair.1.12125.

[36] Curtis G Northcutt, Anish Athalye a Jonas Mueller. „Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks“. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. 2021.

[37] Lucas Beyer et al. „Are we done with ImageNet?“ In: *ArXiv* abs/2006.07159 (2020).

[38] David Rolnick et al. „Deep Learning is Robust to Massive Label Noise“. In: *ArXiv* abs/1705.10694 (2017).

[39] C. Sun et al. „Revisiting Unreasonable Effectiveness of Data in Deep Learning Era“. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, okt. 2017, s. 843–852. DOI: 10.1109/ICCV.2017.97.

[40] Dhruv Mahajan et al. *Exploring the Limits of Weakly Supervised Pretraining*. Tech. spr. 2018.

[41] Ashfaqur Rahman, Daniel V. Smith a Greg Timms. „A Novel Machine Learning Approach Toward Quality Assessment of Sensor Data“. In: *IEEE Sensors Journal* 14.4 (2014), s. 1035–1047. DOI: 10.1109/JSEN.2013.2291855.

[42] H. Erdoğan et al. „Multi-modal Person Recognition for Vehicular Applications“. In: *Multiple Classifier Systems*. Ed. Nikunj C. Oza et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, s. 366–375. ISBN: 978-3-540-31578-0. DOI: https://doi.org/10.1007/11494683_37.

[43] Vladimir Svetnik et al. „Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules“. In: *International Workshop on Multiple Classifier Systems*. Springer. 2004, s. 334–343. DOI: https://doi.org/10.1007/978-3-540-25966-4_33.

[44] Erinija Pranckeviciene, Richard Baumgartner a Ray Somorjai. „Using domain knowledge in the random subspace method: Application to the classification of biomedical spectra“. In: *International Workshop on Multiple Classifier Systems*. Springer. 2005, s. 336–345. DOI: https://doi.org/10.1007/11494683_34.

[45] Mohamed Hosni et al. „Reviewing ensemble classification methods in breast cancer“. In: *Computer methods and programs in biomedicine* 177 (2019), s. 89–112. DOI: https://doi.org/10.1016/j.cmpb.2019.05.019.

[46] L.K. Hansen a Peter Salamon. „Neural Network Ensembles“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12 (nov. 1990), s. 993–1001. DOI: 10.1109/34.58871.

[47] Thomas G. Dietterich. „Ensemble Methods in Machine Learning“. In: *Multiple Classifier Systems*. MCS '00. Berlin, Heidelberg: Springer-Verlag, 2000, s. 1–15. ISBN: 3540677046. DOI: https://doi.org/10.1007/3-540-45014-9_1.

[48] Avrim L. Blum a Ronald L. Rivest. „Training a 3-node neural network is NP-complete“. In: *Neural Networks* 5.1 (1992), s. 117–127. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80010-3.

[49] Gavin Brown et al. „Diversity Creation Methods: A Survey And Categorisation“. In: *Information Fusion* 6 (mar. 2005), s. 5–20. DOI: 10.1016/j.inffus.2004.04.004.

[50] William B. Yates a Derek Partridge. „Use of methodological diversity to improve neural network generalisation“. In: *Neural Computing & Applications* 4.2 (1996), s. 114–128. DOI: https://doi.org/10.1007/BF01413747.

[51] Balaji Lakshminarayanan, Alexander Pritzel a Charles Blundell. „Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles“. In: *Advances in Neural Information Processing Systems*. Ed. I. Guyon et al. Zv. 30. Curran Associates, Inc., 2017.

[52] Stefan Lee et al. *Why M Heads are Better than One: Training a Diverse Ensemble of Deep Networks*. 2015.

[53] Stanislav Fort, Huiyi Hu a Balaji Lakshminarayanan. „Deep ensembles: A loss landscape perspective“. In: *arXiv preprint arXiv:1912.02757* (2019). DOI: https://doi.org/10.48550/arXiv.1912.02757.

[54] David Opitz a Jude Shavlik. „Generating Accurate and Diverse Members of a Neural-Network Ensemble“. In: *Advances in Neural Information Processing Systems*. Ed. D. Touretzky, M.C. Mozer a M. Hasselmo. Zv. 8. MIT Press, 1995.

[55] Md M Islam, Xin Yao a Kazuyuki Murase. „A constructive algorithm for training cooperative neural network ensembles“. In: *IEEE Transactions on neural networks* 14.4 (2003), s. 820–834. DOI: https://doi.org/10.1109/TNN.2003.813832.

[56] Wenjia Wang, P. Jones a D. Partridge. „Diversity between Neural Networks and Decision Trees for Building Multiple Classifier Systems“. In: *Multiple Classifier Systems*. 2000. DOI: https://doi.org/10.1007/3-540-45014-9_23.

[57] W. Langdon, S. Barrett a B. Buxton. „Combining Decision Trees and Neural Networks for Drug Discovery“. In: *Genetic Programming*. Springer Berlin Heidelberg, 2002, s. 60–70. DOI: https://doi.org/10.1007/3-540-45984-7_6.

[58] James Large, Jason Lines a Anthony Bagnall. „A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates“. In: *Data Mining and Knowledge Discovery* 33.6 (júl 2019), s. 1674–1709. ISSN: 1573-756X. DOI: 10.1007/s10618-019-00638-y.

[59] Leo Breiman. „Bagging predictors“. In: *Machine Learning* 24.2 (aug. 1996), s. 123–140. ISSN: 1573-0565. DOI: https://doi.org/10.1007/BF00058655.

[60] R.E. Schapire. „The strength of weak learnability“. In: *30th Annual Symposium on Foundations of Computer Science*. 1989, s. 28–33. DOI: 10.1109/SFCS.1989.63451.

[61] Yoav Freund a Robert E Schapire. „A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting“. In: *Journal of Computer and System Sciences* 55.1 (1997), s. 119–139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504.

[62] Nikunj C Oza a Kagan Tumer. *Dimensionality reduction through classifier ensembles*. Tech. spr. NASA, 1999.

[63] Nikunj C Oza a Kagan Tumer. „Input decimation ensembles: Decorrelation through dimensionality reduction“. In: *International Workshop on Multiple Classifier Systems*. Springer. 2001, s. 238–247. DOI: http://dx.doi.org/10.1007/3-540-48219-9_24.

[64] Yuansong Liao a John Moody. „Constructing heterogeneous committees using input feature grouping: Application to economic forecasting“. In: *Advances in neural information processing systems* 12 (1999), s. 921–927.

[65] Gao Huang et al. *Snapshot Ensembles: Train 1, Get M for Free*. 2017.

[66] Timur Garipov et al. *Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs*. Ed. S. Bengio et al. 2018.

[67] L. Rokach. „Ensemble-based classifiers“. In: *Artificial Intelligence Review* 33 (2009), s. 1–39. DOI: https://doi.org/10.1007/s10462-009-9124-7.

[68] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014. ISBN: 1118315235978-1-118-91454-0. DOI: https://doi.org/10.1002/9781118914564.

[69] K. M. Ting a I. H. Witten. „Issues in Stacked Generalization“. In: *Journal of Artificial Intelligence Research* 10 (máj 1999), s. 271–289. ISSN: 1076-9757. DOI: 10.1613/jair.594.

[70] Saso Džeroski a Bernard Ženko. „Is combining classifiers with stacking better than selecting the best one?“ In: *Machine learning* 54.3 (2004), s. 255–273. DOI: https://doi.org/10.1023/B:MACH.0000015881.36452.6e.

[71] Alexander K Seewald a Johannes Fürnkranz. „An evaluation of grading classifiers“. In: *International symposium on intelligent data analysis*. Springer. 2001, s. 115–124. DOI: http://dx.doi.org/10.1007/3-540-44816-0_12.

[72] Robert A Jacobs et al. „Adaptive mixtures of local experts“. In: *Neural computation* 3.1 (1991), s. 79–87. DOI: 10.1162/neco.1991.3.1.79.

[73] Leo Breiman et al. „Heuristics of instability and stabilization in model selection“. In: *Annals of Statistics* 24.6 (1996), s. 2350–2383. DOI: 10.1214/aos/1032181158.

[74] Philippe REFREGIER a François VALLET. „PROBABILISTIC APPROACH FOR MULTICLASS CLASSIFICATION WITH NEURAL NETWORKS“. In: *Artificial Neural Networks*. Ed. Teuvo KOHONEN et al. Amsterdam: North-Holland, 1991, s. 1003–1006. ISBN: 978-0-444-89178-5. DOI: https://doi.org/10.1016/B978-0-444-89178-5.50005-1.

[75] David Price et al. „Pairwise Neural Network Classifiers with Probabilistic Outputs“. In: *Advances in Neural Information Processing Systems*. Ed. G. Tesauro, D. Touretzky a T. Leen. Zv. 7. MIT Press, 1994.

[76] S.A. Zahorian a Z.B. Nossair. „A partitioned neural network approach for vowel classification using smoothed time/frequency features“. In: *IEEE Transactions on Speech and Audio Processing* 7.4 (1999), s. 414–425. DOI: 10.1109/89.771263.

[77] Ondrej Such, Stefan Benus a Andrea Tinajová. „A New Method to Combine Probability Estimates from Pairwise Binary Classifiers“. In: *Conference on Theory and Practice of Information Technologies*. 2015, s. 194–199.

[78] Ondrej Šuch a Santiago Barreda. „Bayes covariant multi-class classification“. In: *Pattern Recognition Letters* 84 (2016), s. 99–106. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2016.08.014.

[79] Thomas G Dietterich a Ghulum Bakiri. „Solving multiclass learning problems via error-correcting output codes“. In: *Journal of artificial intelligence research* 2 (1994), s. 263–286. DOI: https://doi.org/10.1613/jair.105.

[80] R. Duncan Luce. *Luce's choice axiom*. revision #121550. 2008. DOI: 10.4249/scholarpedia.8077.

[81] Dan Hendrycks a Kevin Gimpel. „A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks“. In: *International Conference on Learning Representations*. 2017.

[82] Ondrej Šuch et al. „Pairwise coupling of convolutional neural networks for better explicability of classification systems“. In: *arXiv preprint arXiv:1911.03645* (2019). DOI: https://doi.org/10.48550/arXiv.1911.03645.

[83] René Fabricius, Ondrej Šuch a Peter Tarábek. „Deep neural network ensembles using class-vs-class weighting“. Unpublished work. 2023.

[84] Ethem Alpaydin. „Classifying Multimodal Data“. In: *The Handbook of Multimodal-Multisensor Interfaces: Signal Processing, Architectures, and Detection of Emotion and Cognition - Volume 2*. Association for Computing Machinery a Morgan & Claypool, 2018, s. 49–69. ISBN: 9781970001716.

[85] Jesse Davis a Mark Goadrich. „The Relationship between Precision-Recall and ROC Curves“. In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, s. 233–240. ISBN: 1595933832. DOI: 10.1145/1143844.1143874.

[86] Christian Szegedy et al. „Going deeper with convolutions“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, s. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[87] Jie Hu, Li Shen a Gang Sun. „Squeeze-and-excitation networks“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, s. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[88] Saining Xie et al. „Aggregated residual transformations for deep neural networks“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, s. 1492–1500. DOI: http://dx.doi.org/10.1109/CVPR.2017.634.

[89] Gao Huang et al. „Deep networks with stochastic depth“. In: *European conference on computer vision*. Springer. 2016, s. 646–661. DOI: https://doi.org/10.1007/978-3-319-46493-0_39.

[90] K. He et al. „Deep Residual Learning for Image Recognition“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jún 2016, s. 770–778. DOI: 10.1109/CVPR.2016.90.

[91] G. Huang et al. „Densely Connected Convolutional Networks“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, júl 2017, s. 2261–2269. DOI: 10.1109/CVPR.2017.243.

[92] F. Chollet. „Xception: Deep Learning with Depthwise Separable Convolutions“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, júl 2017, s. 1800–1807. DOI: 10.1109/CVPR.2017.195.

[93] Alec Radford et al. „Learning Transferable Visual Models From Natural Language Supervision“. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. Marina Meila a Tong Zhang. Zv. 139. Proceedings of Machine Learning Research. PMLR, júl 2021, s. 8748–8763.

[94] Alexander Kolesnikov et al. „Big Transfer (BiT): General Visual Representation Learning“. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*. Glasgow, United Kingdom: Springer-Verlag, 2020, s. 491–507. ISBN: 978-3-030-58557-0. DOI: 10.1007/978-3-030-58558-7_29.

[95] Alexander Kolesnikov et al. *Big Transfer*. 2022.

[96] Ilya Tolstikhin et al. „MLP-Mixer: An all-MLP Architecture for Vision“. In: *Advances in Neural Information Processing Systems*. Ed. A. Beygelzimer et al. 2021.

[97] Alexey Dosovitskiy et al. *Vision Transformer and MLP-Mixer Architectures*. 2022.

[98] Andreas Peter Steiner et al. „How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers“. In: *Transactions on Machine Learning Research* (2022). ISSN: 2835-8856.

[99] Salvatore S. Mangiafico. *Summary and Analysis of Extension Program Evaluation in R*. 1.20.01. New Brunswick, NJ: Rutgers Cooperative Extension, 2016.

Zoznam publikácií

ADF Innovating instruction of communication theory with machine learning and speech analysis - ICETA 2020: 18th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings / Jakab, František [editor]. – 1. vyd. – Denver (USA) : Institute of Electrical and Electronics Engineers, 2020. – ISBN 978-0-7381-2366-0, s. [680-686].

Šuch Ondrej (45%), Fabricius René (45%), Klimo Martin (5%), Juhár Jozef (5%)

AFD Ensemble classification methods - Mathematics in science and technologies: proceedings of the MIST conference 2021 / Bachratá, Katarína; Bohiniková, Alžbeta. – 1. vyd. – [S.l.] : [s.n.], 2021. – ISBN 9798748088183, s. [26-36].

Fabricius René (100%)

ADF Detection of vowel segments in noise with ImageNet neural network architectures - TRANSCOM 2021: 14th International Scientific Conference on Sustainable, Modern and Safe Transport : (2021) Transportation Research Procedia, 55. - ISSN 23521457, s. [1289-1295].

Fabricius René (50%), Šuch Ondrej (50%)

ADN Two objective public service system design problem - Communications: scientific letters of the University of Žilina. - ISSN 1335-4205. - Roč. 23, č. 4 (2021), s. [68-75].

Janáček Jaroslav (25%), Kohání Michal (25%), Grygar Dobroslav (25%), Fabricius René (25%)

ADC Public service system design with conflicting criteria - IEEE Access: practical innovations, open solutions. - ISSN 2169-3536. - Roč. 9 (2021), s. 130665-130679.

Janáček Jaroslav (50%), Fabricius René (50%)

ADF Introducing students to out-of-distribution detection with deep neural networks - ICETA 2022: 20th IEEE International conference on emerging elearning technologies and applications : Information and communication technologies in learning : proceedings. 2020. – ISBN 979-8-3503-2033-6, s. [627-633].

Šuch Ondrej (50%), Fabricius René (45%), Tarábek Peter (5%)

Bridging performance gap between minimal and maximal SVM models - Transactions on Machine Learning Research - ISSN 2835-8856. (2023) https://openreview.net/forum?id=SM1BkjGePI

Šuch Ondrej, Fabricius René

Zoznam príloh

Prílohy

Príloha A: Obsah digitálneho priečinka

Priložený digitálny priečinok obsahuje grafy výsledkov experimentov, ktoré neboli zobrazené v texte práce. Grafy sú rozdelené podľa sekcií, resp. podsekcií kapitoly 7 nasledovne:

- Podsekcia 7.3.1 v priečinku *trenovacia_mnozina_velkost*
- Podsekcia 7.3.2 v priečinku *trenovacia_mnozina_vyber*
- Sekcia 7.5 v priečinku *cifar_regularizacia*
- Sekcia 7.4 v priečinku *cifar_porovnanie_konfiguracii*
- Sekcia 7.6 v priečinku *cifar_vyhodnotenie*
- Sekcia 7.7 v priečinku *cifar_ood*
- Sekcia 7.8 v priečinku *imagenet_topl*
- Sekcia 7.9 v priečinku *imagenet_vyhodnotenie*.