Tartu Ülikool

Loodus- ja täppisteaduse valdkond

Arvutiteaduse instituut

Informaatika õppekava

Kirill Grjaznov

Dirichlet’ kalibreerimismeetodi analüüs

Bakalaurusetöö

Juhendaja: Meelis Kull

Kaasjuhendaja: Markus Kängsepp

Tartu 2020

Dirichlet´ kalibreerimismeetodi analüüs

Lühikokkuvõte:

Masinõppes on klassifitseerimismeetodite üheks probleemiks see, et klassifikaatorid väljastavad liiga enesekindlad tõenäosused. Probleemi lahenduseks on kalibreerimine ehk ennustatud tõenäosuste korrigeerimine. Bakalaureusetöös analüüsitakse Dirichlet´ kalibreerimismeetodit. Töö käigus uuriti kalibreerimismaatriksi muutumist läbi klassifikaatori treenimisprotsessi, selle mõju tulemustele erinevatel treenimisetappidel ning interpreteeriti kalibreerimismaatriksi elementide olemust. Töös kirjeldati, kuidas toimub kalibreerimine Dirichlet´ kalibreerimismeetodiga ning kuidas kalibreerimismaatriks näitab ja parandab klassifikaatori enesekindlust. Eksperimentides kasutati *ResNet110*, *Wide ResNet32* ja *DenseNet40* klassifikaatoreid ning CIFAR-10 andmestikku. Analüüsi tulemusena leiti, et klassifikaatorid olid liiga enesekindlad terve treenimisprotsessi käigus ning Dirichlet´ kalibreerimismeetod parandab enesekindlust ja kalibreeritust igal treenimisetapil.

Võtmesõnad:

Klassifikaatori kalibreerimine, klassifikaatori enesekindlus, tehisintellekt

CERCS: P176 Tehisintellekt

Analysis of Dirichlet calibration method

Abstract:

In machine learning, one of the problems with classification methods is that classifiers give too confident probabilities. The solution to the problem is calibration which performs a correction on the predicted probabilities. In this bachelor's thesis, the Dirichlet calibration method is analyzed. The change of the calibration matrix was studied through the classifier training process, its effect on the results at different training stages, and the nature of the elements of the calibration matrix was interpreted. The paper described how the calibration is performed with the Dirichlet calibration method and how the calibration matrix shows and improves the confidence of the classifier. The experiments were performed on deep neural networks with the architectures

ResNet110, *Wide ResNet32* and *DenseNet40* classifiers and on the CIFAR-10 dataset. The analysis showed that the classifiers were over confident throughout the whole training process, and the Dirichlet calibration method improves confidence at each stage of the training process.

Keywords:

Calibration in classification, classifier confidence, artificial intelligence

CERCS: P176 Artificial intelligence

Sisukord

Sissejuhatus

Tänapäeval on olemas tehnoloogiad, mille abil võib ennustada, näiteks, kas pildi peal on koer või kass ehk millisesse klassi kuulub antud objekt. Sellist probleemi nimetatakse klassifikatsiooniks ning mudelit, mis hakkab ennustama, klassifikaatoriks. Enne seda, et mudel hakkaks ennustama, peab seda treenima ehk andma sellele näidised, näiteks, tuhat pilti koeraga ja tuhat pilti kassiga. Peale seda võib anda klassifikaatorile sisendiks pildi ning ta ennustab, kas see on koer või kass.

Klassifikaatorite mudelite, mis ennustavad rohkem kui kahe klassi peal, ehk mitmeklassiliste (*ingl. k. multiclass*) klassifikaatorite väljundiks on tõenäosuste jaotus. Jaotus näitab, kui kindel on mudel, et antud sisend kuulub mingile klassile. Probleemiks on see, et praegused mudelid on liiga kindlad oma ennustustes [1], näiteks, mudel väljastab mingi klassi kohta tõenäosuse 0.9, aga mudeli tegelik täpsus selle klassi kohta ning sellise ennustamisega on 0.6. Teisisõnu, mudel arvab keskmiselt 0.9 tõenäosusega, et pildi peal on koer, aga sajast ennustamisest sama tulemusega on tegelikult ainult 60 pilti koeraga. On väga tähtis, et mudel näitaks realistlike tõenäosusi selleks, et inimene saaks otsuseid tehes toetuda nendele või autonoomsel süsteemil oleksid realistlikumad hinnangud. Kui saadud tõenäosused ei vasta tõele, siis süsteem, näiteks, isejuhtiv auto, võib teha valesid otsuseid, mis võib tuua suuri probleeme.

Probleemi lahenduseks on mudelite kalibreerimine ehk saadud tõenäosuste korrigeerimine lõppfaasis. Kalibreerimismeetodeid on mitu, mõned neist on kirjeldatud artiklis „On Calibration of Modern Neural Networks" [2]. Autorite poolt pakutud kalibreerimismeetod temperatuuri skaleerimine (*temperature scaling*) andis paremad tulemused võrreldes maatriks-skaleerimisega (*matrix scaling*), vektor-skaleerimisega (*vector scaling*), meetodiga *BBQ* ja mitmete muude kalibreerimismeetoditega. Need meetodid parandavad ennustatud klassi täpsust, aga samas ei paranda ülejäänud klasside tõenäosust ehk kalibreerimisel paraneb dominantne tõenäosus, kuid ülejäänud tõenäosuste korrigeerimine ei ole piisav. Vajalik on aga terve tõenäosuste jaotuse täpsus ehk mitte ainult ennustatud klassi tõenäosus, vaid kõikide klasside tõenäosused mistahes sisendi kohta, mida ei parandata välja toodud meetoditega [1].

Artikkel „Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration" [1] pakub välja Dirichlet´ kalibreerimismeetodi, mis on parem kui

temperatuuri skaleerimise meetod ülaltoodud aspektis. Meetod treenib parameetrilist kalibreerimisfunktsioonide perekonda, mida saab esitada maatrikskujul. Selle koefitsiendid mõjutavad lõpptulemust ning mingil määral arvestavad klasside vaheliste seostega. Bakalaureusetöö autori eesmärgiks on saada teada, kuidas kalibreerimismaatriks muutub mudeli treenimise protsessis, interpreteerida selle käitumist ning uurida, kuidas kalibreerimine mõjutab tulemusi erinevatel treenimisetappidel. Huvitav on saada teada, mis treenimisetapil läheb klassifikaator liiga enesekindlaks ning mil viisil toimub kalibreerimine Dirichlet´ meetodiga. Töö üheks eesmärgiks on veel kontrollida hüpoteesi, et klassifikaator on varasematel treenimisetappidel liiga tagasihoidlik ning mingist hetkest hakkab olema liiga enesekindel.

Töö jaguneb neljaks peatükiks. Esimeses peatükis uuritakse klassifitseerimismudeli treenimisprotsessi, nimelt, millal mudel hakkab muutuma enesekindlaks. Teisel etapil käsitletakse Dirichlet´ kalibreerimismeetodi olemust, kirjeldatakse kalibreerimismaatriksi elemente. Kolmandas peatükis vaadeldakse, kuidas muutub kalibreerimismaatriks läbi terve klassifikaatori treenimise protsessi. Viimases peatükis uuritakse kalibreerimise mõju tulemustele läbi terve treenimisprotsessi.

1.Põhimõisted ja tähistused

Klassifikaator $\hat{p}: X \to \Delta_k$ võtab sisendiks objekti $x \in X$ ning väljastab tõenäosuste vektori $\Delta_k = \{(q_1, q_2, .., q_k) \in [0,1]^k | \sum_{i=1}^{k} q_i = 1\}$, kus q_i on tõenäosus, et objekt *x* kuulub klassi *i*.

Kalibreerimine on saadud tõenäosuste korrigeerimine selliseks, et oodatavad täpsused oleksid realistlikumad.

Artikli [1] autorid toovad välja kaks mõistet, mis iseloomustavad erinevaid kalibreerimise tasemeid, mis on loetletud tugevuse järgi kahanevalt:

1) Klassifikaator on **kalibreeritud klasside kaupa (*classwise-calibrated*)**, kui iga klassi osakaal antud ennustamisel on võrdne selle klassi jaoks ennustatud tõenäosusega.

$$P(Y = i \mid \hat{p}_i(X) = q_i) = q_i \quad \text{iga } i = 1, 2, \ldots, \text{k}$$

2) Klassifikaatoril on **kalibreeritud enesekindlusega (*confidence-calibrated*),** kui mistahes sisendi puhul ennustatud klassi tõenäosus on võrdne oodatava täpsusega.

$$P(\mathrm{Y} = \operatorname{argmax}(\hat{p}(\mathrm{X})) \mid \max(\hat{p}(\mathrm{X})) = \mathrm{c}) = c \qquad \text{iga } c \in [0,1]$$

Kalibreerituse hindamiseks jaotatakse ennustatud klasside tõenäosused vahemikkudeks, näiteks, [0, 0.1], (0.1, 0.2], (0.2, 0.3] …..(0.9, 1]. Nii saab arvutada *Expected Calibration Error* (**ECE)** ehk kalibreerimise vea, mille järgi võime hinnata kui hästi on mudel kalibreeritud:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|,$$

kus M on tõenäosuste vahemikude arv, B_m on need objektide indeksid, mille ennustatud tõenäosus on vahemikus $(\frac{m-1}{M}, \frac{m}{M}]$ ning n on kõikide objektide arv.

ECE saadakse arvutades saadud täpsuse ($acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i)$, kus $\hat{y}_i$ on ennustatud klass ning y_i tegelik klass) ja oodatava täpsuse ($conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \max(\hat{p}(X_i))$) vahet igal tõenäosuse vahemikul, saadud vahe korrutatakse sisendite osakaaluga sellel vahemikul ning leitakse summat [2]. Seega, mida väiksem on ECE, seda paremini on kalibreeritud enesekindlus.

Selleks et hinnata **kalibreerimist klasside kaupa,** on vaja arvutada ECE iga klassi jaoks ning arvutada keskmine [1] (**classwise-ECE ehk ECEcw**).

Juhul kui klassifikaator on *softmax´*ga närvivõrk, siis viimase kihi väljundiks on *k* suurune vektor (kus *k* on klasside arv), mida töödeldakse *softmax*-funktsiooniga (**σ**), mis väljastabki tõenäosuste jaotuse. Kalibreerimisel muudetakse saadud tõenäosuste vektorit ning lõpuks töödeldakse jälle *softmax*-funktsiooniga, et saada kalibreeritud tõenäosusi.

Softmax-funktsioon võimaldab klassifitseerida rohkem kui kaks klassi, sest annab tõenäosuste jaotuse, kus ennustataval klassil on kõrgeim tõenäosus, seepärast kasutatakse seda tihti klassifikaatoritel ning ta annab head tulemused [3]. *Softmax* funktsiooni valem on järgmine:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}, \text{ kus i = \{1, 2 ..., k\} ja z = } (z_1, z_2 \ldots, z_k) \in \mathbb{R}^k$$

Entroopia (*ing. k. information entropy*) näitab, kui palju informatsiooni saame antud tõenäosuste jaotuselt *q* [4]. Entroopia valem on järgmine [5]:

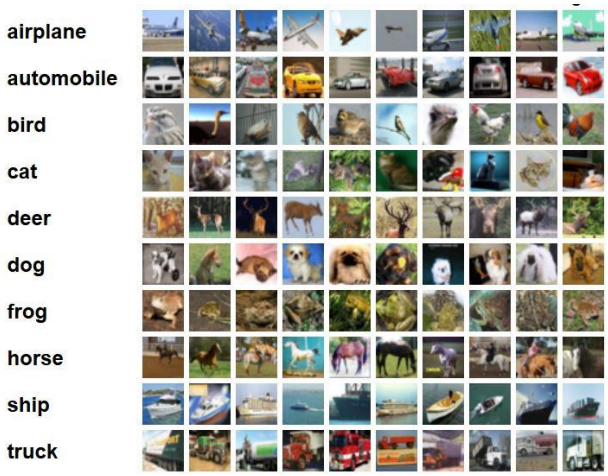
$$e(q) = -\sum_{i=1}^{k} q_i \ln q_i \text{ , kus } q = (q_1, q_2, \ldots, q_k)$$

Ristentroopia (*Kullback-Leibler divergence*) näitab, kui palju informatsiooni kaotame, kui liigume tõenäosuste jaotusest q tõenäosuste jaotusele p [4]. Ristentroopia valem on järgmine:

$$D_{KL}(p, q) = \sum_{i=1}^{k} p_i \ln \frac{p_i}{q_i}$$

2.Klassifikaatori treenimisprotsess

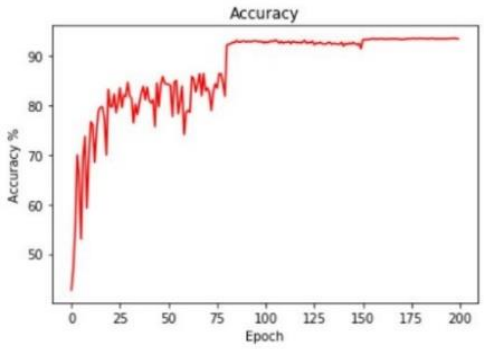
2.1.Mudelid ja andmestik



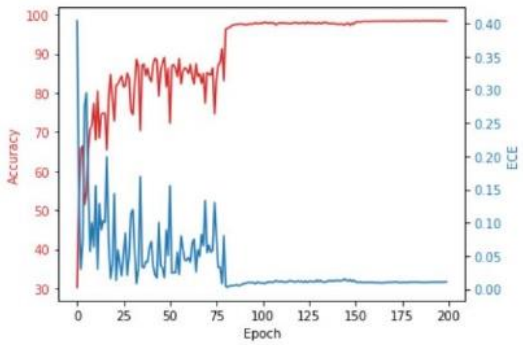
Tabel 1. CIFAR-10 näidispildid klasside kaupa [6]

Töös vaadeldakse kolm erinevat klassifikaatori mudelit, mis on pärileviga (*img. k. feed-forward*) sügavad tehisnärvivõrgud: *ResNet110* [7], *Wide ResNet32* [8] ja *DenseNet40* [9]. Samu arhitektuure vaadeldi ka artiklis [1] ning käesolevas töös on kasutatud üldiselt sama treenimisprotsess. Oluline on teha kindlaks, et kalibreerimine käitub sarnaselt, olenemata sellest, kuidas saadakse tõenäosusi. Treenimisprotsess kestis kahel esimesel mudelil 200 epohhi (*epochs*) ning kolmandat mudelit treeniti 300 epohhi, et näha, kas on muutusi üleliigsel treenimisel. Andmestik, mille peal treeniti mudeleid, on CIFAR-10 [6]. Andmestik koosneb piltidest (32x32 pikslit), mis on jaotatud 10. klassi (lennuk, auto, lind jne). Kokku on 60000 pilti, igas klassis 6000 pilti. Treenimise teostas Markus Kängsepp, andmed jagati juhuslikult treening- (45000 pilti), valideerimis- (5000 pilti) ning testandmeteks (10000 pilti) [10].

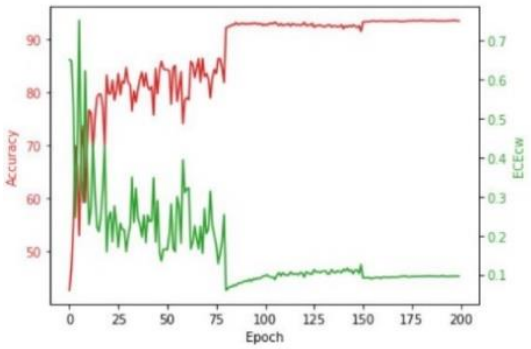
2.2. Treenimisprotsess



Joonis 1. Resnet 110 täpsus testandmetel treenimisprotsessi käigul. (accuracy – täpsus, epoch - epohh)



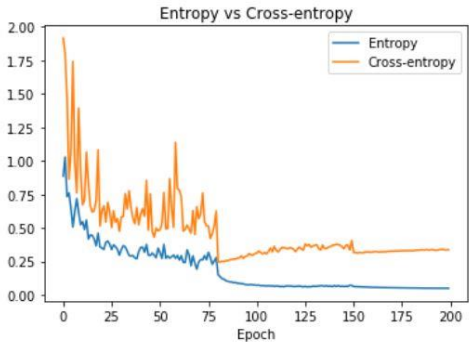
Joonis 2. Resnet110 täpsus ja ECE testandmetel treenimisprotsessi käigul.



Joonis 3. Resnet110 täpsus ja ECEcw testandmetel treenimisprotsessi käigul.

Treenimise käigus oli vajalik uurida, kuidas muutub mudeli täpsus ning kalibreeritus. Joonisel 1 on näha, et *Resnet110* puhul täpsus kõigub treenimise käigus ning täpsus muutub järsult treenimisel kahes kohas. See tähendab, et mudel otsis lokaalseid miinimume kasutades erinevaid õpisamme (*learning rate*). *Resnet110* puhul muutus õpisamm 80. ja 150. epohhil. Seda oli vaja meeles pidada järgneval uurimisel, sest ka kalibreerimismaatriksite muutused ei ole sujuvad, kuna nad sõltuvad ennustatud tõenäosustest, mis omakorda sõltuvad mudeli olekust. Huvitav märkus, joonistel 2 ja 3 on näha, et ECE ja ECEcw muutuvad koos täpsusega. Joonistel on näha, kuidas muutub ECE ja ECEcw võrreldes täpsusega treenimise käigus. Seal, kus täpsus väheneb suurenevad ka ECE ja ECEcw. ECE suurus ei ütle, kas mudel on liiga enesekindel või mitte, sest oodatav täpsus võib olla ka väiksem tegelikkusest ehk mudel võib olla liiga tagasihoidlik. Järgmises alapeatükis uurime enesekindlust lähemalt.

2.3. Enesekindlus



Joonis 4. Resnet110 entroopia vs ristentroopia treenimisprotsessi käigul testandmetel (entropy - entroopia, cross-entopy - ristentroopia).

Mudeli liigne enesekindlus ongi põhiline probleem, mida lahendab kalibreerimine. Selleks, et vaadata kui enesekindel on mudel, võib mõõta ennustamise tõenäosuste vektori keskmist entroopiat testandmetel ning võrrelda seda keskmise ristentroopiaga, mis saadakse ennustatud tõenäosuste vektori ja tegeliku klassi *one-hot-encoding´*u kujul. Mida väiksem entroopia, seda suurem on ennustatud klassi tõenäosus, mis võib põhjustada liigset enesekindlust. Kui see on ristentroopiast väiksem, siis mudel on liiga enesekindel. Mudel on kalibreeritud, kui need on võrdsed. See on põhjustatud sellest, et entroopia näitab, kui palju informatsiooni me keskmisel saame tõenäosuste jaotuselt, aga ristentroopia näitab, kui palju informatsiooni me keskmiselt kaotasime [4, 5]. Joonisel 4 on näha, et entroopia ja ristentroopia vahe suureneb treenimisprotsessi käigus ning entroopia on koguaeg madalam, mis tähendab, et mudel muutub liiga enesekindlaks. Kusjuures 80. epohhil leidis mudel hea lokaalse miinimumi ning pärast hakkas muutuma liiga enesekindlaks.

3. Dirichlet´ kalibreerimismeetod

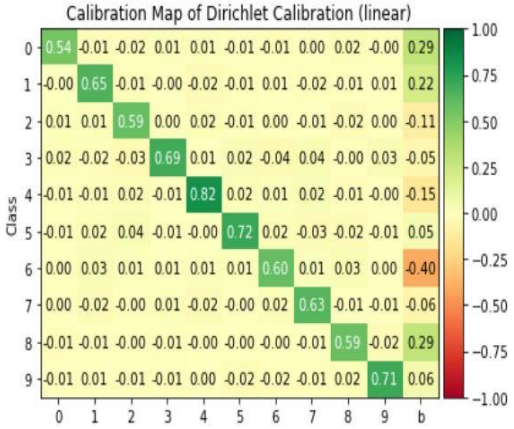
3.1. Kalibreerimismeetodi kirjeldus

Dirichlet´ kalibreerimismeetod seisneb selles, et treenitakse k x k kalibreerimismaatriks W, kus k on klasside arv ning vabaliikmete vektor b suurusega k (joonis 5). Ennustatud tõenäosuste vektori q, kus q_i on klassi i ennustatud tõenäosus ja $i = (1,2\ldots.,k)$, kalibreeritakse funktsiooniga:

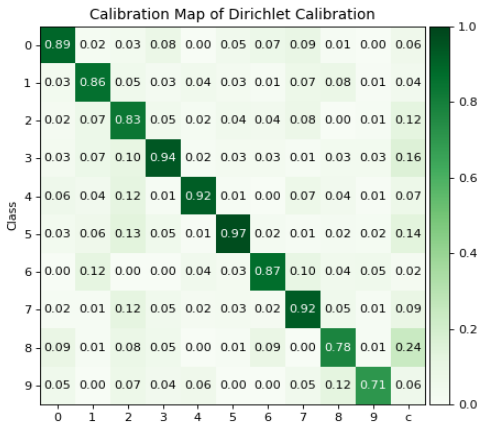
$$\mu_{DirLinear}(q; W; b) = \sigma(W \ln q + b),$$

kus σ on *softmax* funktsioon, ning saadakse kalibreeritud tõenäosuste vektori. On olemas mitu kalibreerimisfunktsiooni kuju. Ülaltoodud kuju nimetati lineaarseks ning selle eelis on lihtne treenimine neuronkihina [1]. Kalibreerimismaatriksi saadakse treenides üheainsa neuronkihti, mille sisendiks on k suurune tõenäosuste vektor ning väljundiks on kalibreeritud tõenäosused, eraldi valideerimisandmetel ehk peale seda kui klassifikaator on treenitud, saadakse ennustatud tõenäosuste vektorid igal objektil valideerimisandmestikul. Kalibreerimismaatriksit treenitakse saadud tõenäosuste ja nende tegelike klasside peal.

3.2. Kalibreerimismaatriksi kanooniline kuju



Joonis 5. Dirichlet´ kalibreerimismaatriksi lineaarne kuju (W, b). Esimesed kümme veergu on maatriks W ning viimane veerg on vabaliikmete vector b (class - klass).



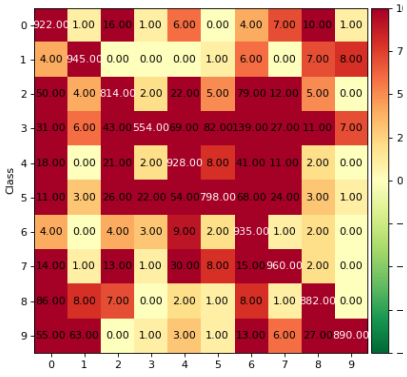
Joonis 6. Dirichlet´ kalibreerimismaatriksi kanooniline kuju (A, c).

Ülaltoodud lineaarset kuju on raske interpreteerida, sest sama kalibreerimisfunktsiooni saab esitada mitmeti, seega viiakse maatriks kanoonilisele kujule, mille eelis on ühesus. Kalibreerimisfunktsioon sellisel kujul on:

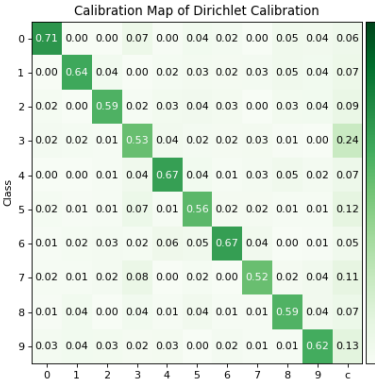
$$\mu_{Dir}(q; A; c) = \sigma(A \ln \frac{q}{u} + \ln c),$$

kus A on k x k maatriks, kus iga element saadakse $a_{ij} = w_{ij} - min_i\, w_{ij}$ (kus i on rida ja j on veerg), ning vabaliikmete vektor c, mis on saadud $c = \sigma(W \ln u + b)$, kus $u = (1/k, 1/k \ldots. 1/k)$ suurusega k (joonis 6). Vektor c on tõenäosuste vektor, mis näitab kuhu nihutatakse kalibreeritud tõenäosusi. Mida suurem on c_i seda rohkem nihutatakse tõenäosusi klassi i poole. Kanooniline kuju on samaväärne lineaarse kujuga ning seda on mugavam analüüsida, sest sama kalibreerimisfunktsiooni saab lineaarsel kujul esitada mitmeti, aga kanoonilisel kujul ainult ühtemoodi. Vaatame kalibreerimismaatriksi elemente lähemalt.

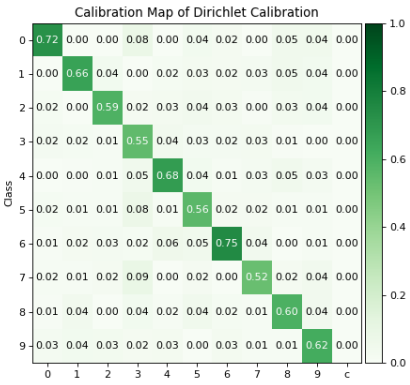
3.2.1. Vabaliikmed



Joonis 7. DenseNet40 klassifikaatori segadusmaatriks testandmetel 84. epohhill.



Joonis 8. DenseNet40 klassifikaatori kalibreerimismaatriks 84. epohhil.



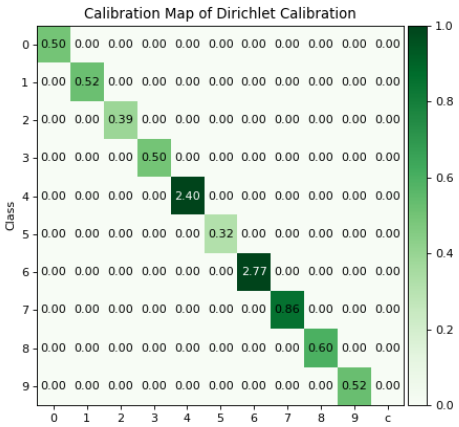
Joonis 9. DenseNet40 klassifikaatori kalibreerimismaatriks 84. epohhil ilma vabaliikmeta.

Vabaliige $\boldsymbol{c_i}$ tõstab klassi i tõenäosust sõltuvalt suurusest, näiteks, vabaliikmete vektor (0.2, 0.5, 0.3) suurendab kõige rohkem teise klassi tõenäosust. Kalibreerimisel see tähendab, et mida suurem on $\boldsymbol{c_i}$ seda vähem klassifikaator ennustas klassi i, kui see oli vajalik. Joonisel 7 on näha mudeli segadusmaatriks (*confusion matrix*) 84. epohhil ning joonisel 8 on selle mudeli kalibreerimismaatriks. Segadusmaatriksi element e_{ij} näitab, mitu korda mudel ennustas klassi j,

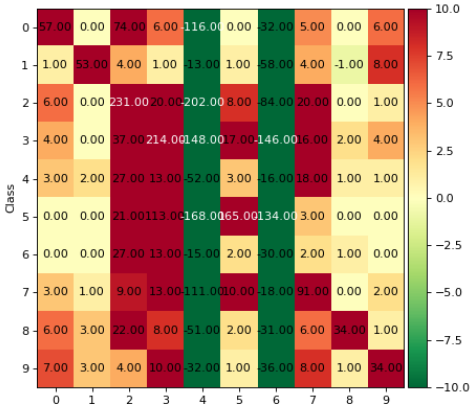
kui tegelikult oli see klassi *i* objekt. On näha, et mudel eksis kõige rohkem kolmanda klassi puhul (neljas rida) ehk ennustas seda liiga vähe kordi seal, kus oli vaja, sest selle rea elemendid (välja arvatud peadiagonaali element) on suured. Seega vastav vabaliige kalibreerimismaatriksil on suurem, et tõsta eelnevalt klassi 3 tõenäosust.

Joonisel 9 on näidatud sama kalibreerimismaatriks, aga mis on treenitud ilma vabaliikmeta. On näha, et vabaliikmete olemasolu eriti ei muuda maatriksi rakendamise tulemust, kuigi koos vabaliikmetega täpsus suureneb.

3.2.2. Diagonaal



Joonis 10. DenseNet40 klassifikaatori kalibreerimismaatriks ainult diagonaalsete elementidega. 2. epohhil.



Joonis 11. Eelmise joonise kalibreerimismaatriksi tulemus. Segadusmaatriksite vahe pärast kalibreerimist testandmetel.

Uurime alguses ainult diagonaali ehk treenime kalibreerimismaatriksi ilma teiste elementidega (joonis 10). Kuna $Ax = Wx + const1$, $\ln \sigma(x) = x + const2$ mistahes x puhul, kus *const1* ja *const2* on konstantsed vektorid (kõik arvud on võrdsed), ning $\sigma(v + const) = \sigma(v)$, kus v on mistahes vektor ja *const* konstantne vektor [1], siis selleks, et oleks lihtsam interpreteerida diagonaalsete elementide mõju, viime kalibreerimisfunktsiooni kujule:

$$\sigma\left(A \ln \frac{q}{u} + \ln c\right) = \sigma(A \ln q - A \ln u + \ln c)$$
$$= \sigma(A \ln q - W \ln u + const1 + \ln \sigma(W \ln u + b))$$
$$= \sigma(A \ln q - W \ln u + const1 + W \ln u + b + const2) = \sigma(A \ln q + b)$$

Kuna vabaliikmeid antud juhul ei ole, siis vaatleme funktsiooni $\sigma(A \ln q)$. Joonisel 11 on näidatud segadusmaatriksite vahe ehk kuidas muutus segadusmaatriks pärast kalibreerimist. On näha, et mida suurem diagonaalne element (võrreldes teistega), seda vähem ennustatakse antud klassi pärast kalibreerimist. See on põhjustatud sellest, et ln q_i on negatiivne, sest $q_i < 1$ (i=0,1..9), seega mida suurem on diagonaalne element a_i, seda negatiivsem on $a_i \ln q_i$, järelikult vähendatakse klassi i tõenäosust rohkem. Vaatleme kolme huvitavat olukorda:

1) Kui kõik diagonaalsed elemendid on võrdsed ühega, siis kalibreerimist ei toimu ehk tõenäosused on juba kalibreeritud.

$\sigma(A * \ln q) = \sigma(a * \ln q) = q$ kus $a = 1$ ehk praegu vaatleme olukorra, kus A koosneb ainult võrdsetest diagonaalsetest elementidest, mis on samaväärne korrutamisega mingi arvu a-ga.

2) Juhul kui kõik elemendid on võrdsed ja suuremad kui üks ($a > 1$), siis igal sisendil suurendatakse dominantse klassi tõenäosust, sest mida väiksem tõenäosus q_i, seda negatiivsem on $ln\ q_i$, järelikult tegur muudab seda rohkem. Teisisõnu, väiksed tõenäosused järsult langevad ning *softmax* funktsiooni tõttu suur tõenäosus kasvab.

Tõestus: nagu näitab punkt 1, siis $a = 1$ korral kalibreerimist ei toimu.

$$\sigma(a \ln q) = \sigma(\ln q)$$

$$\ln q = (\ln q_1, \ln q_2, \dots, \ln q_k)$$

Oletame, et q_i ($i \in (1,2,3,\dots,k)$) on suurim, seega $0 \geq \ln q_i > \ln q_j$, kus $j = (1,2,3\dots,k)$ ning $j \neq i$ ehk kõik ülejäänud liikmed. Kui me suurendame a (ehk $a > 1$), siis märkame,

$$\ln q_i - a * \ln q_i < \ln q_j - a * \ln q_j$$

Arvestades sellega, et $\sigma(x + const) = \sigma(x)$ siis liidame pärast korrutamist saadud vektorile konstantse vektori suurusega k arvudega $c = \ln q_i - a * \ln q_i$.

$$(a \ln q_1, a \ln q_2, \dots a \ln q_k) + (c, c, c, \dots, c)$$

Tulemuseks saame vektori, kus jäi alles klassi i liige $\ln q_i$ ning kõik teised liikmed läksid väiksemaks võrreldes vektoriga $(\ln q_1, \dots, \ln q_k)$, sest:

$$a \ln q_i + c = \ln q_i$$

$$a \ln q_j + c < \ln q_j$$

See tähendab, et pärast *softmax* funktsiooni dominantne tõenäosus suureneb:

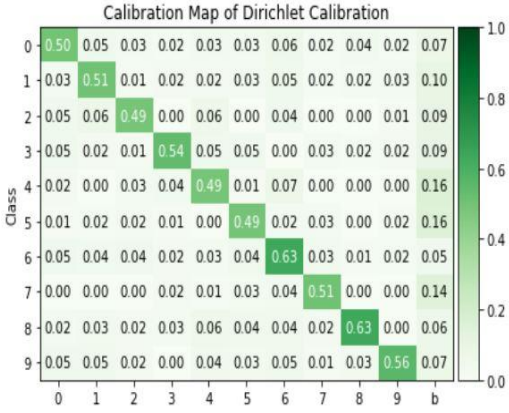
$\sigma_i(\ln q) = \frac{e^{\ln q_i}}{\sum_{l=1}^{k} e^{\ln q_l}} < \sigma_i(a \ln q) = \frac{e^{\ln q_i}}{(\sum_{l=1, l \neq i}^{k} e^{a \ln q_l + c}) + e^{\ln q_i}}$, sest nimetaja läks väiksemaks.

3) Kui kõik elemendid on võrdsed ning väiksemad kui üks, siis on analoogiline, aga vastupidine olukord. Tõestus on analoogiline, kus võrratus punktis 2 muutub:

$$\ln q_i - a * \ln q_i > \ln q_j - a * \ln q_j \quad \text{sest } a < 1$$

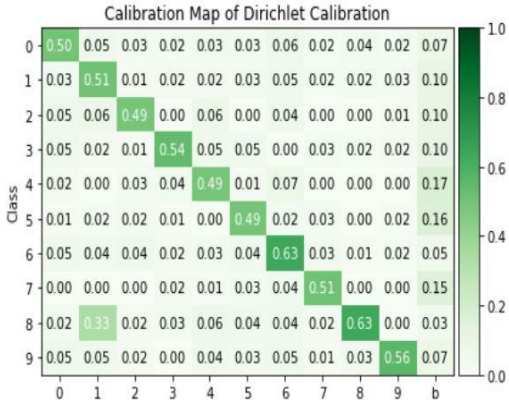
Kokkuvõtvalt, kui diagonaalsed elemendid on väiksemad ühest, siis mudel on liiga enesekindel ehk kalibreerimisel vähendatakse dominantset tõenäosust ja jaotatakse seda teiste klasside vahel.

3.2.3. Diagonaali väljaspool elemendid



Joonis 12. DenseNet40 klassifikaatori kalibreerimismaatriks 123. epohhil.

Joonis 13. Segadusmaatriksite vahe pärast kalibreerimist testandmetel 123. epohhil.



Joonis 14. Suurendatakse element a_{81} 0.3 võrra.

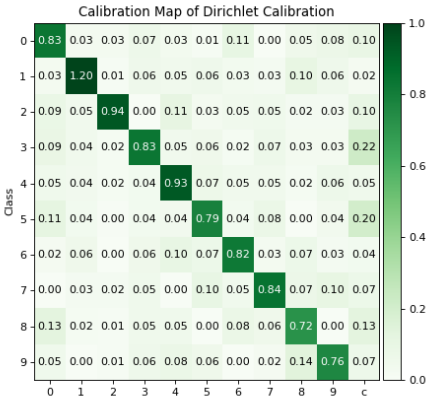
Joonis 15. Kalibreerimise tulemus pärast maatriksi muutust testandmetel.

Diagonaali väliste elementide interpreteerimiseks vaatame jooniseid 12-15. Joonised 12 ja 13 näitavad kalibreerimismaatriksi koos kõikide elementidega ja segadusmaatriksite vahet pärast kalibreerimist. Kui me suurendame elemendi a_{81} 0.3 võrra (joonis 14), siis väheneb klassi 8 ennustamiste arv, eriti, kui see oli õige ennustamine (joonis 15). Järelikult vähendati klassi 8 tõenäosust üldjuhul ning eriti siis, kui klassi 8 tõenäosus oli suur. Nagu eelnevalt järeldatud, siis suurem element vähendab vastava klassi tõenäosust. Kuna element mõjutab ainult ühe liidetava maatriksi korrutamisel ning arvestades varasemad tulemused, siis võib järeldada, et element a_{ij} ($i \neq j$) vähendab klassi i tõenäosust kui klassi j tõenäosus on väike. See põhjendab saadud tulemuse, sest tõesti, kui ennustatakse klassi 8, siis tema tõenäosus on suurim, järelikult teiste klasside tõenäosused on väiksed, mille hulgas on ka klass 1. Tulemusena vähendatakse klassi 8 tõenäosust

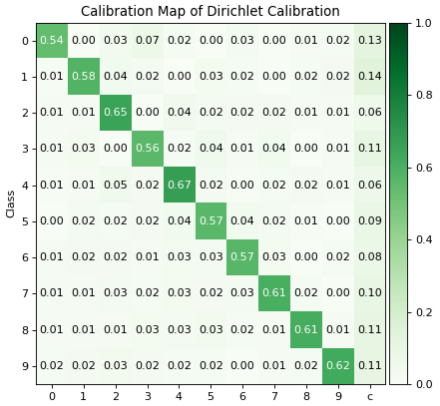
iga kord, kui ta on suur. Sellist klasside seost võib kirjeldada nii, kui mudel arvab, et pildi peal ei ole pliiats, siis vähendame tõenäosust, et see on pastapliiats, sest need klassid on väga sarnased.

Kuna samasugune arutuskäik kehtib ka diagonaalsete elementide jaoks ehk mida väiksem on q_i seda negatiivsem on $a_{ii} \ln q_i$, seega võib üldistada reegli ka maatriksi diagonaalsete elementide jaoks ehk maatriksi element a_{ij} vähendab klassi *i* tõenäosust, mida väiksem on klassi *j* tõenäosus.

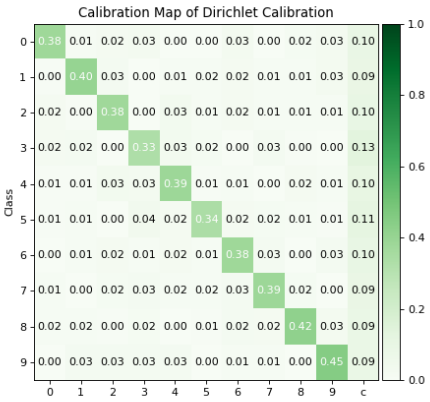
4. Kalibreerimismaatriksi muutused klassifikaatori treenimise käigus



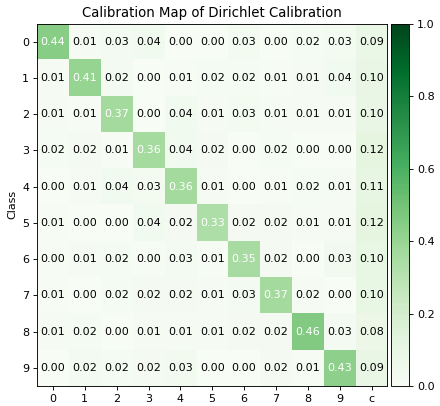
Joonis 16. DenseNet40 klassifikaatori kalibreerimismaatriks 1. epohhil.



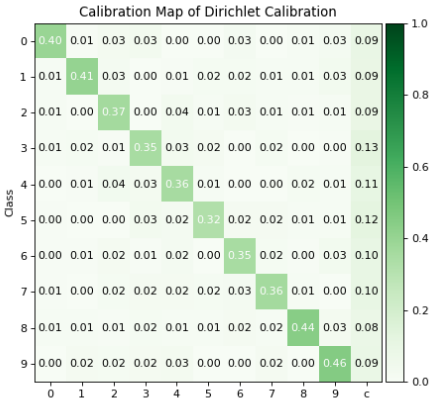
Joonis 17. 60. epohh



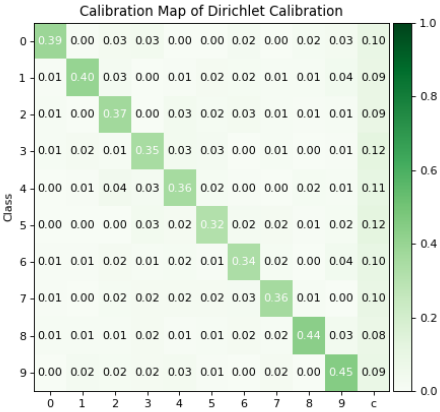
Joonis 18. 206 epohh



Joonis 19. 230. epohh



Joonis 20. 270. epohh



Joonis 21. 300. epohh

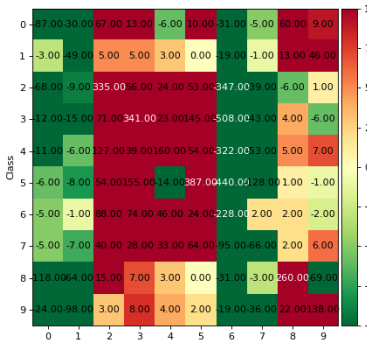
Klassifikaatori treenimise käigus iga epohhi järel salvestati mudeli ennustused valideerimisandmetel ning treeniti selle järgi kalibreerimismaatriks. Kokkuvõtteks on saadud kalibreerimismaatriksid iga klassifikaatori treenimisepohhi tagant, niimoodi kolme mudeli kohta. *Resnet110* ja *Wide ResNet* mudelitel mõlemal on saadud 200 kalibreerimismaatriksit ja 300 *DenseNet* mudeli kohta, et vaadata, kas maatriks muutub üleliigsel treenimisel. Kõikide mudelite kalibreerimismaatriksid muutusid sama malli järgi, nimelt, diagonaalsete elementide väärtused vähenesid ning diagonaali-välised elemendid lähenesid nullile. Joonised 16-21 näitavad kalibreerimismaatrikseid epohhidel 1, 60, 206, 230, 270 ja 300 vastavalt mudeli *DenseNet* treenimisekäigus. On näha, et pärast 200. epohhi kalibreerimismaatriksid on väga sarnased ehk elemendid ainult natuke kõiguvad. Elemendid ei saa olla täiesti võrdsed, sest treenimise käigus muudetakse mudeli olekut, aga samas ei ole näha elementide muutumistendentsi, seega võib järeldada, et üleliigne treenimine üldiselt ei muuda kalibreerimismaatriksit. Tähtis on mainida, et ka ECE ja ECEcw ei lähe suuremaks üleliigsel treenimisel.

Terve treenimise protsessi käigul kalibreerimismaatriksi diagonaalsed elemendid olid väiksemad kui 1. Ainult varasemates epohhides esinesid mõnikord üksikud elemendid, mis on suuremad kui 1, aga seda võib põhjendada klassifikaatori halva lokaalse miinimumiga sellel hetkel, sest tema täpsus kõikus liiga palju (peatükk 2.2). Tulemused näitavad, et klassifikaatori enesekindlus suurenes treenimise käigus ja üldjuhul oli klassifikaator liiga enesekindel.

Diagonaali-välised elemendid lähenesid nullile, mille põhjuseks on see, et ennustamisel dominantse klassi tõenäosus suurenes, seega väliste elementide mõju suurenes, sest teiste klasside tõenäosused vähenesid. Sellises olukorras on vaja väliseid elemente vähendada selleks, et nad ei vähendaks dominantse klassi tõenäosust liiga palju. Teiseks põhjuseks võiks olla see, et klasside vahelised seosed kaovad ära ehk mudel hakkab paremini eristama klasse, aga sellisel juhul diagonaali välised elemendid ei jõua nullini, sest klassifikaator ei ennusta ideaalselt, järelikult klasside vahel on sarnaseid tunnuseid, mille pärast klassifikaator tegi vigu.

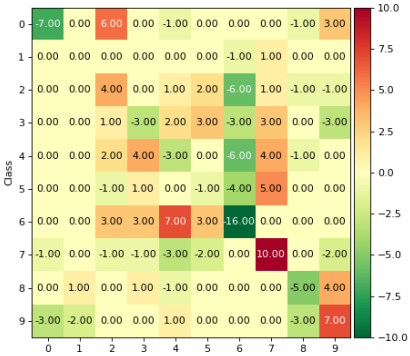
5. Kalibreerimise mõju tulemustele

5.1. Segadusmaatriks



Joonis 22. DenseNet40 klassifikaatori tulemused testandmetel pärast kalibreerimist 1. epohhil

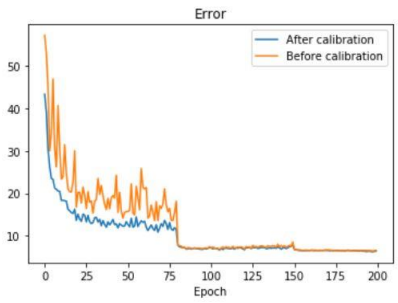
Joonis 23. 100. epohh



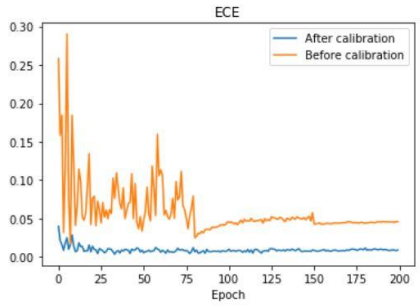
Joonis 24. 200. epohh

Uurides segadusmaatriksite vahet pärast kalibreerimist erinevate epohhide järel (joonised 22-24), võib näha, et mida treenitum on klassifikaator, seda vähem mõjutab kalibreerimine ennustatud klassi. Tõesti, klassifikaatori täpsus suurenes ehk ta tegi vähem vigu, mida võiks parandada kalibreerimine. Varasematel epohhidel (joonis 22) kalibreerimine tihti suurendas mõnede klasside tõenäosused liiga palju ja vähendas teiste klasside tõenäosusi. Selline suur tulemuste muutus võib olla põhjustatud sellest, et algul klassifikaator ei osanud väga eristada klasse ning kalibreerimine muutus lihtsalt klassifikaatori edasiseks treenimiseks ehk kalibreerimine otsis parema lokaalse miinimumi. Põhjuseks on see, et kalibreerimismaatriks saadakse neuronkihi treenimisel, seega liiga halva täpsusega saadud tõenäosusi treenitakse selles kihis edasi.

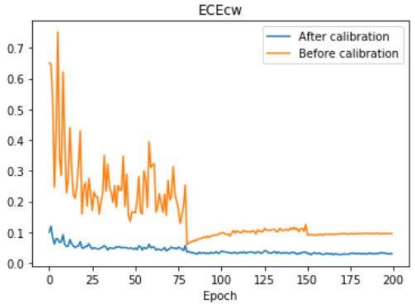
5.2. Error, ECE ja ECEcw



Joonis 25. Resnet110 error testandmetel.



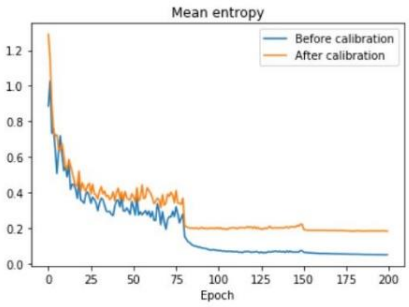
Joonis 26. Resnet110 ECE testandmetel.



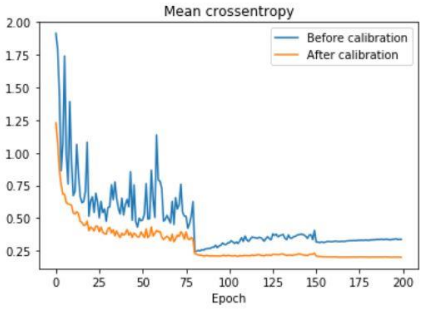
Joonis 27. Resnet110 ECEcw testandmetel.

Kalibreerimise ülesanne on korrigeerida ennustatud tõenäosusi ehk parandada ennustamiste ECE ja ECEcw. Nagu näidatud joonistel 26 ja 27, Dirichlet´ kalibreerimismeetod parandab tulemusi tervel treenimisprotsessil ning ka täpsus paraneb, sest nad on tugevas korrelatsioonis (peatükk 1.2).

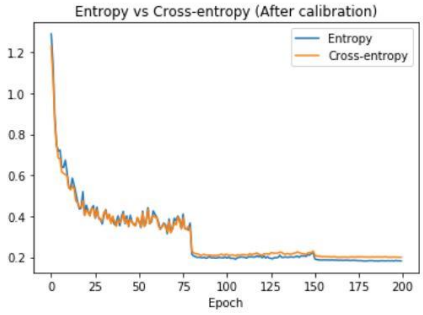
5.3. Enesekindluse muut



Joonis 28. Resnet110 keskmine entroopia testandmetel treenimisprotsessil



Joonis 29. Resnet110 keskmine ristentroopia testandmetel treenimisprotsessil.



Joonis 30. Resnet110 entroopia vs ristentroopia testandmetel pärast kalibreerimist.

Kui enne kalibreerimist oli klassifikaator liiga enesekindel (peatükk 2.3), siis oli huvitav vaadata, kuidas enesekindlus muutus pärast kalibreerimist. Joonised 28 ja 29 näitavad, et kalibreerimine suurendab ennustuste keskmist entroopiat ning ristentroopia langeb. Entroopia suureneb, sest kalibreerimisel vähendatakse dominantse klassi tõenäosust ning see jaotatakse teiste klasside vahel, sest kalibreerimismaatriksi diagonaalsed elemendid olid väiksemad kui 1 (peatükk 4.1). Ristentroopia vähenes suuremas osas selle pärast, et täpsus suurenes pärast kalibreerimist ning valedel ennustamistel ennustatud klassi tõenäosus muutus väiksemaks. Joonis 30 näitab keskmist

entroopiat ja ristentroopiat pärast kalibreerimist tervel treenimisprotsessil. On näha, et need on peaaegu võrdsed, mis tähendab, et Dirichlet´ kalibreerimismeetod lahendas liigse enesekindluse probleemi. Võrdlemiseks võib vaadata tulemust enne kalibreerimist joonisel 4.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli interpreteerida Dirichlet´ kalibreerimismaatriksit, vaadata, kuidas muutub kalibreerimismaatriks läbi klassifikaatori treenimisprotsessi ning kuidas see mõjutab ennustamist.

Töö käigus uuriti kalibreerimismaatriksit erinevatel klassifikaatori treenimis-epohhidel. Analüüsi käigus leiti, et kalibreerimismaatriksi element a_{ij} vähendab klassi *i* tõenäosust, mida väiksem on klassi *j* tõenäosus. Vabaliikmete vektor *c* näitab, mis klasside puhul klassifikaator tegi rohkem vigu ehk ei ennustanud neid õigesti, seega vabaliige c_i tõstab eelnevalt klassi *i* tõenäosust.

Töös kasutati kolme klassifikaatori mudeli: *Resnet110*, *Wide ResNet32* ja *DenseNet40*. Kõigi kolme mudeli treenimisprotsessil muutus kalibreerimismaatriks sama malli järgi. Diagonaalsed elemendid läksid väiksemateks ja olid väiksemad kui 1, diagonaali-välised elemendid lähenesid nullile. Tulemus näitab, et klassifikaatorid olid liiga enesekindlad terve treenimisprotsessi käigus, kui mitte võtta arvesse algseid epohhe, mil klassifikaatori täpsus kõikus liiga palju.

Dirichlet´ kalibreerimine parandas ennustamisi terve klassifikaatori treenimisprotsessi käigul ehk igal epohhil täpsus läks paremaks, enesekindlus vähenes võrreldes sellega, mis oli enne kalibreerimist, ECE ja ECEcw läksid madalamaks.

Kirjandus

[1] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song ja Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration, 2019.

[2] Chuan Guo, Geoff Pleiss, Yu Sun, Kilian Q. Weinberger. On Calibration of Modern Neural Networks. 2017.

[3] Mahmood Hamza. The Softmax Function, Simplified. Towards Data Science. 2018.

[4] Andy Thomas. An introduction to entropy, cross entropy and KL divergence in machine learning.

[5] Meelis Kull, Peter Flach. Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration. 2015.

[6] Alex Krizhevsky, Vinod Nair, Geoffrey Hinton. The CIFAR-10 dataset. Google.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 2015.

[8] Sergey Zagoruyko, Nikos Komodakis. Wide Residual Networks. 2016.

[9] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks. 2016.

[10] Markus Kängsepp. Convolutional Neural Networks Training and Logits Saving. https://github.com/markus93/convnets_logits?fbclid=IwAR1CANyQDYipkTEcHRe2-s7vWSCdEPnD8TP7VXCJ8edL7NupqLLjycBFeZo

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Kirill Grjaznov,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Dirichlet´ kalibreerimismeetodi analüüs**,

   mille juhendajad on Meelis Kull ja Markus Kängsepp,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Kirill Grjaznov

08.05.2020