

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Labai panašių neuroninių klasių atskyrimas neuroniniais tinklais

(Separation of very similar neuronal classes by neural networks)

Kursinis darbas

Atliko:	3 kurso 5 grupės studentai Miglė Vaitulevičiūtė	(parašas)
Darbo vadovas:	dr. Vytautas Valaitis	(parašas)

Vilnius – 2018

TURINYS

ĮVADAS	3
1. NEURONINIS TINKLAS	4
1.1. Neuroninio tinklo sudėtis	4
1.2. Neuroninio tinklo veikimas	4
1.3. Aktyvavimo funkcijos	5
1.4. Nuostolio funkcijos	5
1.5. Optimizavimo funkcijos	6
2. VGG	7
2.1. Fine-tune	7
3. EKSPERIMENTAS	8
3.1. Technologijos	8
3.1.1. ImageNet	8
3.1.2. Keras	8
REZULTATAI IR IŠVADOS	9
LITERATŪRA	10

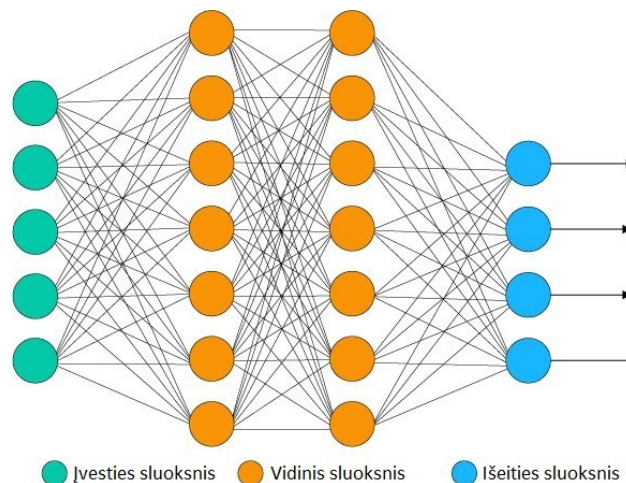
Įvadas

1. Neuroninis tinklas

Pagal apibendrintą žmogaus smegenų veikimą buvo sugalvoti neuroniniai tinklai [GBC16]. Bendrai žmogaus smegenys turi šimtus milijardų neuronų, kurie yra sujungti sinapsėmis. Per tuos neuronus sklinda elektroniniai impulsai perduodantys informaciją - taip žmonės gali atpažinti objektus, garsus ir t.t. Neuroniniai tinklai veikia panašiai. Jie irgi turi daug besijungiančių neuronų, kurie gauna informaciją ir pagal tą informaciją gali nuspręsti koks objektas yra paveikslėlyje. Tačiau ties tuo ir baigiasi žmogaus smegenų ir neuroninių tinklų panašumas, kadangi neuroniniai tinklai yra kompiuterinė simuliacija - matematinis algoritmas su aritmetiniais kintamaisiais. Ši simuliacija yra suvokiama tik žmogui, kuris suprogramavo neuroninį tinklą, pačiam tinklui simuliacija nieko nereiškia, nuovokos neįgauna.

1.1. Neuroninio tinklo sudėtis

Neuroninis tinklas yra sluoksnių rinkinys (1 pav.) - neuronų grupė sudaro sluoksnį, kuris yra sujungtas tarpusavyje su kitais sluoksniais. Vienas iš sluoksnių privalo būti įvesties sluoksnis, kuris atitinkamai pagal užduotį gali gauti įvairios formos informaciją - paveikslukai, vaizdo medžiaga, garsas ir t.t. Ši informacija yra reikalinga tam, kad tinklas galėtų ją išanalizuoti ir išmokti. Tuo tikslu, kad vėliau gavęs panašią informaciją galėtų ją atpažinti - tam reikalingas išeities sluoksnis. Jis yra priešingame neuroninio tinklo gale negu įeities sluoksnis. Tarp anksčiau apibūdintų sluoksnių yra įvairaus dydžio sluoksnių sistema, kuri atlieka pagrindinį darbą [Woo18].



1 pav. Sluoksnių rinkinys

1.2. Neuroninio tinklo veikimas

Jungtys tarp neuronų yra pateiktos skaitine išraiška ir vadinama svoriu. Kuo didesnis šis svoris tuo didesnę įtaką turi vienas neuronas kitam. Vienam neuronui yra pateikiama visų prieš jį buvusių neuronų informacija ir jungčių svoriai. Kiekvieno neurono informacija yra sudauginama su jo svoriu ir visi šie duomenys yra sudedami tarpusavyje. Taip iš vektoriaus gaunamas vienas

rezultatas ir jei šis rezultatas tinka aktyvavimo funkcijai, jis yra perduodamas tolimesniems neuronams. Tokio tipo veikimo dizainas yra vadinamas „feedforward” tinklu.

Tačiau jungčių svoriai nėra pastovūs. Kai neuroninis tinklas mokosi, galutinis rezultatas yra lyginamas su tikėtinu teisingu rezultatu, jei šie rezultatai skiriasi, svoriai yra keičiami atitinkamai tai vadinama „backpropagation”. Tokiu būdu yra gerinamas rezultatas ir mažinamas skirtumas tarp tikėtino ir gauto atsakymų.

1.3. Aktyvavimo funkcijos

Aktyvavimo funkcijų yra įvairių, kadangi sprendžiant tam tikrą problemą yra geriau naudoti vienas funkcijas, o kitas problemas - kitas funkcijas. Pagrindė yra dviejų tipų aktyvavimo funkcijos - tiesinės ir netiesinės. Tačiau tiesinės nėra tokios populiarios, kadangi jos neleidžia įvesčiai būti lanksčiai. Nors tiesinė funkcija labai dažnai naudojama išeities sluoksnyje. Netiesinės funkcijos dažniausiai naudojamos vidiniuose sluoksniuose. Šiuo metu labiausiai naudojama yra ReLU, kadangi naudojant šią funkciją mokymo rezultatai nuolat gerėja, tačiau ReLU funkcijos sprausumas nesuteikia efektyvumo tinklui [XWC⁺15].

Aktyvavimo funkcijos yra skirstomos į:

- Tiesinė:
 - Žingsninė (binarinė) - išėjimas yra 0 arba 1.
- Netiesinė:
 - Sigmoidinė - išėjimas intervale [0; 1].
 - Hiperbolinio tangento - išėjimas intervale [-1; 1].
 - Minkštojo maksimumo - sunormuoja išėjimo vektorius į 1.
 - ReLU - išėjimas intervale [0; begalybė].

1.4. Nuostolio funkcijos

Kai neuroninis tinklas mokosi, jo gaunami rezultatai gali labai skirtis nuo tikėtinų rezultatų. Todėl nuostolio funkcija apskaičiuoja kaip stipriai skiriasi gautas rezultatas nuo tikėtino. Kuo didesnis nuostolis tuo toliau nuo teisingo atsakymo yra neuroninis tinklas [Dav15]. Paprasčiausia ir dažniausiai naudojama nuostolio funkcija yra vidutinio kvadrato klaida. Ši funkcija apskaičiuoja kvadratinį skirtumą tarp tikėtino ir gauto rezultatų. Tačiau šios funkcijos vienas iš didesnių trūkumų - neproporcingas išskyrimas didelių rezultatų. Kadangi funkcija didėja kvadratiniai, o ne tiesiniai, kai gaunamas rezultatas tolsta nuo tikėtino rezultato.

Priklausomai nuo to kokią problemą yra bandoma išspręsti yra naudojamos skirtingos funkcijos. Viena iš problemų yra klasifikacijos - dažniausiai išeities rezultatas yra tikimybės vertė $f(x)$. Bendrai, funkcijos reikšmės dydis parodo gauto rezultato tikslumą. Dauguma klasifikacijos nuostolių funkcijos stengiasi maksimaliai padidinti tikslumą [Agr17].

Kelios klasifikacijos nuostolio funkcijos:

- Binarinė kryžiaus entropija.
- Neigiama registravimo tikimybė.

- Maržos klasifikatorius.
- Minkštų maržų klasifikatorius.

1.5. Optimizavimo funkcijos

Optimizavimo funkcijos naudojamos vidinių tinklo parametrų atnaujinimui, kad sumažinti gaunamų rezultatų netikslumą. Visos optimizavimo funkcijos gali būti suskirtos į du tipus - nuolatinio mokymosi greičio ir prisitaikančio mokymosi.

Nuolatinio mokymosi greičio funkcijos turi hiperparametrą - mokymosi greitį. Jis privalo būti nustatytas, tačiau pasirinkti tinkamą mokymosi greitį gali būti sudėtinga - pasirinkus per mažą vidiniai parametrai gali labai lėtai konverguoti, o pasirinkus per didelį parametrams gali trukdyti konverguoti ir priversti nuostolio funkciją svyruoti apie minimumą arba diverguoti. Šio tipo funkcijos turi panašų hiperparametrą - momentą - kuris didina mokymosi greitį, kai jis artėja prie minimumo.

Vienos iš pagrindinių problemų nuolatinio mokymosi greičio funkcijų, kad jos privalo turėti nustatytus hiperparametrus iš anksto ir jie labai stipriai priklauso nuo modelio ir sprendžiamos problemos. Dar vienas trūkumas, kad toks pats mokymosi greitis yra pritaikomas visiems vidinių parametrų atnaujinimams.

Prisitaikančio mokymosi funkcijos turi atskirus kiekvieno parametro mokymosi greičio metodus, kurie teikia euristikos metodą, nereikalaujant brangaus darbo rankiniu būdu nustatant hiperparametrus mokymosi greičiui. Tačiau šios funkcijos generalizuoja blogiau negu nuolatinio mokymosi greičio funkcijos, nors ir mokymosi metu pasirodo geriau [WRS⁺17].

2. VGG

2.1. Fine-tune

3. Ekspermentas

3.1. Technologijos

3.1.1. ImageNet

3.1.2. Keras

Rezultaitai ir išvados

Literatūra

- [Agr17] Apoorva Agrawal. Loss functions and optimization algorithms. <https://medium.com/data-science-group-iitr/loss-functions-and-optimization-algorithms-demystified-bb92daff331c>. 2017.
- [Dav15] Cameron Davidson-Pilon. *Bayesian methods for hackers*. Addison-Wesley Professional, 2015.
- [GBC16] Ian Goodfellow, Yoshua Bengio ir Aaron Courville. *Deep learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [Woo18] C. Woodford. Neural networks. <https://www.explainthatstuff.com/introduction-to-neural-networks.html>. 2018.
- [WRS⁺17] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro ir B. Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *Arxiv e-prints*, 2017. eprint: 1705.08292 (stat.ML).
- [XWC⁺15] Bing Xu, Naiyan Wang, Tianqi Chen ir Mu Li. Empirical evaluation of rectified activations in convolutional network. *Corr*, abs/1505.00853, 2015. arXiv: 1505.00853. URL: <http://arxiv.org/abs/1505.00853>.