

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. KONVOLIUCINIS NEURONINIS TINKLAS .....	7
2.1. Konvoliucija .....	7
2.2. Konvoliucinio neuroninio tinklo sluoksniai .....	8
2.2.1. Konvoliucinis sluoksnis .....	8
2.2.2. Sujungimo sluoksnis .....	8
2.2.3. Pilno sujungimo sluoksnis .....	8
2.3. Modelio reguliavimas .....	9
2.4. Architektūros .....	9
3. EKSPERIMENTAS .....	11
3.1. Technologijos .....	11
3.1.1. ImageNet .....	11
3.1.2. Keras .....	11
REZULTATAI IR IŠVADOS .....	12
LITERATŪRA .....	13

# Įvadas

Darbo tikslas: patikrinti mokymosi tikslumo kokybę su skirtingų gylių dirbtiniais neuroniniais tinklais, kai mokymui yra naudojamas mažas paveikslėlių rinkinys.

Užduotys:

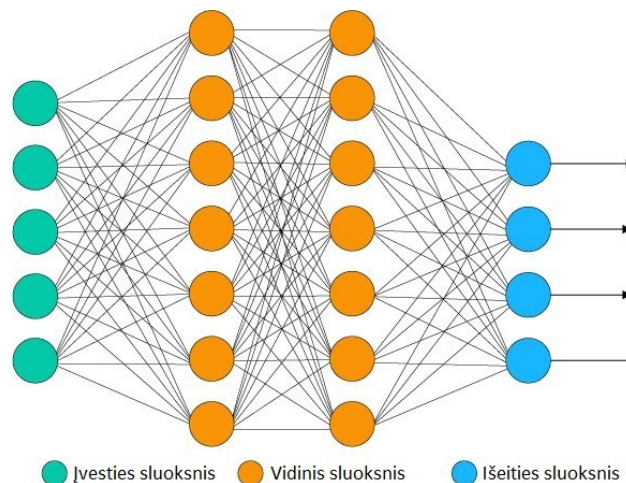
1. Pasirinkti analizuojamą dirbtinį neuroninį tinklą.
2. Pritaikyti ir modifikuoti neuronų tinklą išsikeltam darbo tikslui.
3. Gautą neuronų tinklą su skirtingais gyliais apmokyti.
4. Surasti ir naudoti geriausią optimizavimo funkciją.

# 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 nesuteikia.

## 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 vadinamos 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. 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 nuolatos gerėja, tačiau ReLU funkcijos sprautumas nesuteikia efektyvumo tinklui [XWC<sup>+</sup>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<sup>+</sup>17].

## 2. Konvoliucinis neuroninis tinklas

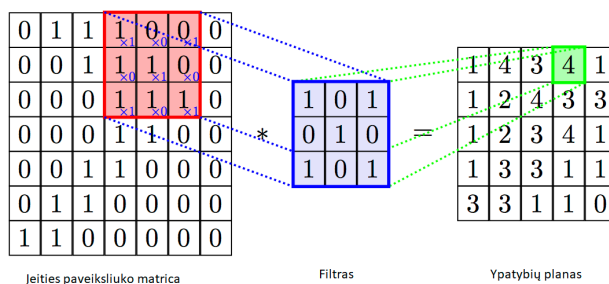
Konvoliuciniai neuroniniai tinklai yra labai panašūs į paprastus neuroninius tinklus (daugiau informacijos skyriuje „Neuroninis tinklas“). Tačiau pagrindinis skirtumas tarp šių tinklų yra, kad konvoliucinio įeities sluoksnis priima tik tai paveikslukus, kurie jei padaryti su standartine skaitmenine kamera, turi tris komponentus - raudoną, žalią ir mėlyną. Šiuos komponentus galima įsivaizduoti kaip tris 2D matricas sudėtas viena ant kitos. Kiekvienos matricos i-osios eilutės ir j-ojo stulpelio elementas atitinka nuotraukos pikselį, kurio reikšmė yra intervale nuo 0 iki 255. Kadangi naudojamos informacijos tipas yra specifinis, tai labai sumažina tinklo parametrų kiekį ir tinklą padaro efektyvesnį.

Objektų atpažinimas paveikslukuose yra sudėtingas dėl šių iššukių:

- Segmentavimas - paveikslukai gali atvaizduoti įvairias scenas, kuriose gali būti pavaizduota daug objektų, kurie vienas kita gali dalinai uždengti.
- Šviesa - pikselių intensyvumas gali būti paveiktas šviesos šaltinio ar pačio objekto.
- Deformacija - objektai gali būti deformuoti įvairiais būdais, pavyzdžiui, kiekvieno žmogaus ranka parašyti skaičiai skiriasi.
- Galimybės - objektų klasės dažnai nustatomos pagal tai kaip patys objektai yra naudojami, pavyzdžiui, kėdės yra objektai sukurti sėdėti, tačiau jos gali turėti įvairų dizainą.
- Žvilgsnio taškas - keičiant vietą iš kurios yra žiūrima gali keistis objekto forma, informacija šokinėja per įeities sluoksnio dimensiją (t.y. pikselius).

### 2.1. Konvoliucija

Konvoliucija yra matematinė operacija, kuri apibūdina taisyklę, kuri parodo kaip reikia sujungti du informacijos rinkinius [PG17]. Paveikslukų analizėje, statinė ir pagrindinė funkcija yra įeities paveikslukas, kuris yra analizuojamas, o antroji, judanti funkcija, žinoma kaip filtras, nes ji išskiria paveiksluko ypatybę. Abi funkcijos yra susietos daugyba (2 pav.).



2 pav. Konvoliucijos veikimas

Tačiau konvoliuciniai tinklai turi daug filtrų, kurie pereina per vieną paveiksluką, kiekvienas išskirdamas skirtingą paveiksluko ypatybę. Pirmuose sluoksniuose šiuos filtrus galima įsivaizduoti kaip horizontalių, vertikalinių ar įstrižių linijų filtrus, kurie sukuria paveikslėlio kraštų planą. Tinklas paima visus filtrus, gabaliukus paveikslukų ypatybių vietų, ir juos sudeda į planą, kuris parodo ypatybės vietą. Mokydamasis skirtingų proporcijų ypatybių, tinklas leidžia lengvai kurti greitą ypatybių atpažinimą.

## **2.2. Konvoliucinio neuroninio tinklo sluoksniai**

Neuroniniai tinklai tai yra sluoksnių rinkinys, kuris turi įėjties, vidinius ir išėjties sluoksnius. Tačiau priklausomai kokio tipo neuroninis tinklas vidiniai sluoksniai gali skirtis. Konvoliuciniai neuroniniai tinklai turi tris pagrindinius sluoksnių tipus, kurie sudaro vidinį sluoksnį. Šie tipai yra konvoliucinis, sujungimo ir pilno sujungimo sluoksniai.

### **2.2.1. Konvoliucinis sluoksnis**

Šis sluoksnis yra pagrindinis konvoliucinio neuroninio tinklo sluoksnis, kuris atlieka daugiausia skaičiavimų, nustato visas paveiksluko ypatybes. Kadangi, įėjties informacija (paveikslukas) yra didelės dimencijos neefektyvu visų neuronų sujungti vienus su kitais. Todėl neuronai yra sujungiami su lokaliu informacijos kiekiu, kuris yra lygus filtro dydžiui ir vadinamas erdvinio mastu [Li15].

Neuronų kiekis po konvoliucijos (ypatybių plano dydis) yra nustatomas trimis parametrais:

- Gylis - atitinka filtrų skaičių.
- Žingsnis - pikselių kiekis, kuris parodo per kiek reikia slinkti filtro matricą per įėjties informacijos matricą.
- Nulių pamušalas - įėjties informacijos matricos kraštus užpildyti nuliais.

### **2.2.2. Sujungimo sluoksnis**

Periodiškai sujungimo sluoksnis yra įterpiamas tarp konvoliucinių. Pagrindinis sluoksnio tikslas yra laipsniškai mažinti erdvinį filtruojamo paveiksluko mastą. Šis veikslas yra atliekamas tam, kad sumažinti parametų ir skaičiavimų kiekį bei kontroliuoti perjungimą. Sujungimo sluoksnis veikia nepriklausomai nuo kiekvieno gabalėlio gylio ir keičia jo dydį erdviškai, naudodamas MAX operaciją. Dažniausiai šis sluoksnis yra naudojamas su 2x2 dydžio filtru - kas antras po konvoliucijos gauto gabaliuko kiekvienas gylio sluoksnis yra mažinamas per pusę ties ilgiu ir pločiu, taip yra atsikratoma 75 procentų aktyvacijų. Po šios operacijos gabaliuko gylis nepasikeičia.

Dažniausiai yra naudojamos dvi šio sluoksnio variacijos. Pirmasis yra vadinamas persidengiantis sujungimas, kur filtro dydis yra lygus 3 ir žingsnis yra lygus 2. O kitas dažniau naudojamas turi filtro dydį lygų 2 ir žingsnį taip pat 2. Sujungimo sluoksniai su labai dideliais parametrais yra labai destruktuvūs.

### **2.2.3. Pilno sujungimo sluoksnis**

Konvoliucinio ir sujungimo sluoksnių išėjties yra aukšto lygio ypatybės, kurios yra gautos iš įėjties paveiksluko. Pilno sujungimo sluoksnis yra sujungtas su visais neuronais iš sluoksnio buvusio prieš jį. Šio sluoksnio tikslas yra panaudojant tas ypatybes, kurios yra gautos iš prieš tai buvusių sluoksnių, nustatyti kokioms klasėms priklauso įėjties paveikslukas pagal mokymo informacijos rinkinį, kai neuroninio tinklo problema yra klasifikacija [Kar16]. Jei šiam sluoksniui yra naudojama minkštojo maksimumo funkcija tuomet sudėtis visų gautų galimybių turi būti lygi 1. Minkštojo maksimumo funkcija priima vektorių įvertinimų ir jį suspaudžia į vektorius, kuriame



yra klasių tikimybių įvertinimai intervale nuo 0 iki 1, kur tikimybė arčiausiai vieneto reiškia, kad labiausiai užtikrintas dėl tos klasės.

## 2.3. Modelio reguliavimas

Pilnas dirbtinio neuroninio tinklo apmokymas gali užtrukti labai ilgą laiką ir išnaudoti daug resursų. Todėl yra kai kurios įstaigos arba žmonės, kurie apmoko savo tinklą ir jo svorius bei reikšmes, vadinamą modeliu, pateikia visuomenei, tačiau šis modelis yra nepritaikytas individualiai žmogaus užduočiai. Modelį reikia reguliuoti - iš naujo apmokyti paskutinius sluoksnius su individualios užduoties parametrais.

Daugelis dirbtinių neuroninių tinklų apmokytų su natūraliais paveikslukais turi fenomeną. Pirmuosiuose sluoksniuose jie išmoka ypatybių panašių į Gaboro filtrą (tiesinis filtras naudojamas tekstūroms analizuoti) ir spalvų dėmes. Šios pirmojo sluoksnio ypatybės nepriklauso nuo duomenų rinkinio, bet yra bendros ir tinkamos daugeliui duomenų rinkinių ir užduočių [DBLP:journals/corr/YosinskiCBL14]. Dėl šio fenomeno galima naudoti modelius neapmokytus su specifiniu duomenų rinkiniu, bet jį minimaliai modifikuoti, kitoms užduotims spręsti, kas leidžia sutaupyti resursų bei turėti mažesnę duomenų rinkinį.

## 2.4. Architektūros

Konvoliuciniai neuroniniai tinklai turi keletą skirtingų architektūrų, kurios yra naudojamos atinkamai pagal sprendžiamą problemą. 1 lentelėje pateikta informacija apie įvairias architektūras.

Pavadinimas	Metai	Parametrų kiekis	
LeNet	1998	60 000	Geriausiai atpažysta ranka parašty
AlexNet	2012	60 000 000	Vei
GoogLeNet/Inception	2014	4 000 000	Vidiniai sluoksniai sudėti paraleliai
VGGNet	2014	138 000 000	Panaš
ResNet	2015	25 000 000	Turi labai daug sluoksnių, sudėtų vienas po kito, kurie

### **3. Ekspermentas**

#### **3.1. Technologijos**

##### **3.1.1. ImageNet**

##### **3.1.2. Keras**

## Rezultatai ir išvados

Darbo rezultatai:

- Pasirinktas tiriamas neuroninis tinklas - VGG.
- Rasta geriausia ir tinkamiausia optimizavimo funkcija - XX.
- VGG buvo pritaikytas ir modifikuotas išsikeltam tikslui.
- Neuroninis tinklas buvo apmokytas.

Darbo 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>.
- [Kar16] Ujjwal Karn. An intuitive explanation of convolutional neural networks. <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>. 2016.
- [Li15] Fei-Fei Li. Convolutional neural networks. <https://cs231n.github.io/convolutional-networks/>. 2015.
- [PG17] Josh Patterson ir Adam Gibson. *Deep learning*. O'Reilly Media, Inc., 2017.
- [Woo18] C. Woodford. Neural networks. <https://www.explainthatstuff.com/introduction-to-neural-networks.html>. 2018.
- [WRS<sup>+</sup>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<sup>+</sup>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>.