Vilniaus Universitetas

Matematikos ir Informatikos fakultetas

Programų Sistemų katedra

**Konvoliucinio neuroninio tinklo apmokymas**

Taikomųjų programų kūrimo mobiliesiems įrenginiams ir autonominėms sistemoms 2 laboratorinis darbas

Atliko: 4kurso 5 grupės studentas Jurgis Kargaudas

2018-01-11

**Turinys**

[**Įvadas** 3](#_Toc503437509)

[**1.** **Konvolucinio neuroninio tinklo mokymas skirtingomis konfigūracijomis** 4](#_Toc503437510)

[**1.1.** **Pradiniai duomenys ir konvoliucinis neuroninis tinklas** 4](#_Toc503437511)

[**1.2.** **Neuroninio tinklo mokymas** 4](#_Toc503437512)

[**2.** **Neuroninio tinklo praplėtimas įtraukiant papildomą vaizdų klasę** 5](#_Toc503437513)

[**2.1.** **Papildomos vaizdų klasės gavimas** 5](#_Toc503437514)

[**2.2.** **Neuroninio tinklo testavimas su nauja vaizdų aibe** 5](#_Toc503437515)

[**Priedai** 7](#_Toc503437516)

**Įvadas**

Šio darbo tikslas yra apmokyti konvoliucinį neuroninį tinklą atpažinti skirtingas vaizdų klases, stebėti kintantį jo efektyvumą keičiant konfigūracijas bei išbandyti tinklą papildant duomenis nauja vaizdų klase. Darbe naudojamos klasės – katės ir šunys. Papildoma, vėliau pridedama vaizdų klasė – antys. Neuroniniam tinklui modeliuoti naudojama *Keras* biblioteka.

1. **Konvolucinio neuroninio tinklo mokymas skirtingomis konfigūracijomis**
   1. **Pradiniai duomenys ir konvoliucinis neuroninis tinklas**

Neuroninis tinklas šiame darbe turėjo atpažinti kačių ir šunų paveikslėlius. Duomenų aibė buvo sudaryta iš 10000 treniravimo nuotraukų kiekvienai klasei bei 1000 testavimo nuotraukų kiekvienai klasei (iš viso 22000 paveikslėlių). Naudoti *Kaggle* platformos skelbto *Cats vs Dogs* konkurso paveikslėliai.

Vaizdų atpažinimui naudojantis *Keras* biblioteka buvo paruoštas konvoliucinis neuroninis tinklas. Pradinio tinklo architektūra:

* 3 konvoliuciniai blokai (*konvoliucinis* sluoksnis, *ReLu* aktyvacijos sluoksnis, *max-pooling* sluoksnis).
* Tinklo išplokštinimas (*Flatten*)
* 2 pilnai sujungti blokai su *Dropout* tarp jų.
  1. *Dense*(64) su *ReLu* aktyvacija;
  2. *Dropout*(0.5);
  3. *Dense*(1) su *sigmoid* aktyvacija.

Pradinio tinklo kodą galima rasti Priede Nr.1.

* 1. **Neuroninio tinklo mokymas**

Kiekvienas mokymas turėjo 10 epochų. Tinklas mokytas 4-is kartus – vieną kartą su pradine architektūra ir 3 kartus konfigūruojant tinklo parametrus. Žemiau pateikiama rezultatai, gauti po kiekvieno mokymo apibūdinanti tinklo pakitimą nuo pradinio tinklo ir gautą tikslumą po mokymo.

* + 1. **Pradinio tinklo apmokymas**

Apmokius tinklą su pradiniais parametrais ir architektūra, gauti žemiau matomi duomenys.

Paveikslas 1. Pradinio tinklo mokymosi tikslumas

* + 1. **Tinklo apmokymas pakeitus max\_pooling dydį**

Po pirmo tinklo apmokymo, atliktas tinklo parametrų pakeitimas – max\_pooling dydis pakeistas iš 2 į 3. Žemiau pateikiami gauti rezultatai:

Paveikslas 2. Tinklo mokymosi tikslumas, kai max\_pooling dydis lygus 3

Iš rezultatų galima matyti, kad tinklo mokymasis pagerėjo – pagerėjo tiek testavimo duomenų atpažinimas, tiek loss funkcijos reikšmė. Šis pakeitimas bus naudojamas ir toliau.

* + 1. **Tinklo apmokymas pakeitus konvoliucinių blokų kiekį**

Kita išbandyta konfigūracija – mažesnis konvoliucinių blokų kiekis. Iš tinklo išimtas vidurinis konvoliucinis blokas (palikti tik 2 konvoliuciniai blokai). Gauti mokymosi rezultatai:

Paveikslas 3. Tinklo mokymosi tikslumas išėmus 1-ą konvoliucinį bloką

Diagramoje matyti, kad tiek loss funkcijos, tiek tikslumo rezultatai sumažėjo, tad išimtas konvoliucinis blokas grąžintas į tinklo architektūrą.

* + 1. **Tinklo apmokymas su vienu pilnai sujungtu sluoksniu**

Paskutinės išbandytos sąlygos – kitoks pilnai sujungtų sluoksnių kiekis tinkle. Šiuo atveju, vietoj 2 pilnai sujungtų sluoksnių paliktas 1-as pilnai sujungtas sluoksnis.

Paveikslas 4. TInklo mokymosi tikslumas, palikus 1 pilnai sujungtą sluoksnį

Iš grafiko galima matyti, kad mokymosi metu atsirado gana dideli šuoliai tarp skirtingų epochų, vertinant atpažinimo tikslumo ir loss funkcijos reikšmes. Kadangi tinklas tapo mažiau nuspėjamas, šio pakeitimo atsisakyta.

1. **Neuroninio tinklo praplėtimas įtraukiant papildomą vaizdų klasę**
   1. **Papildomos vaizdų klasės gavimas**

Norint patikrinti neuroninio tinklo naujų vaizdų klasės atpažinimą skirtingomis sąlygomis (nieko nežinant apie aibę arba jau apmokytam su ja), siekta surasti papildomą vaizdų klasę. Ieškota tokių paveikslėlių, kurie priklausytų kitai klasei, tačiau būtų panašūs bendromis savybėmis – spalvotos nuotraukos su gyvūnais.

Pasinaudojus Flickr platformos suteikiama API prieiga, sukurta Python programa, kuri siunčia paveiksliukus su nurodytu užrašu (angl. *tag*) iš Flickr puslapio (Priedas nr. 2). Išbandžius keletą skirtingų užrašų, naujai vaizdų klasei pasirinktos antys. Jų iš viso parsiųsta 1500. Atrinkus su antimis nesusijusius paveiksliukus, palikta 1100 failų. Kadangi parsiųsti failai buvo gana dideli, paveiksliukų kokybė sumažinta ir palikti panašaus į pradinių klasių dydžio failai (10-300 KB).

Siekiant padidinti turimą ančių aibę, duomenys buvo augmentuoti. Pasitelkus *Keras* paveiksliukų apdorojimo biblioteką, iš kiekvieno failo buvo pagaminti 5 failai su tam tikrais pakeitimais (apvertimu, ištempimu, pasukimu). Tokiu būdu gauta 5000 paveikslėlių mokymui ir 500 paveikslėlių testavimui.

* 1. **Neuroninio tinklo testavimas su nauja vaizdų aibe**

Neuroninis tinklas išbandytas dviem scenarijais – jam nežinant apie naują vaizdų klasę (t.y., mokiusis tik su katėmis ir šunimis) bei žinant (mokiusis su katėmis, šunimis ir antimis). Tikslas – įsitikinti, kad tinklas atpažįsta antis prastai, kai nėra jų matęs ir atpažįsta gerai, kai yra išmokytas jas atpažinti.

Ančių paveikslėliai buvo padalinti taip: 1000 paveikslėlių mokymuisi ir 100 paveikslėlių testavimui. Prieš atliekant naujus mokymus, tinklas buvo šiek tiek pakeistas, norint jį pritaikyti kategorinei klasifikacijai (prieš tai taikyta binarinė klasifikacija): paskutinis pilnai sujungtas sluoksnis iš  pakeistas į. Žemiau pateikiami gauti rezultatai skirtingomis mokymo sąlygomis:

Lentelė 2. Neuroninio tinklo mokymas papildžius duomenis nauja vaizdų klase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mokymo ID | Mokymo duomenys | Testavimo duomenys | Treniravimo duomenų  atpažinimo  tikslumas | Testavimo duomenų atpažinimo tikslumas |
| 1 | 6000 kačių paveiksliukų  6000 šunų paveiksliukų | 1000 kačių paveiksliukų  1000 šunų paveiksliukų  100 ančių paveiksliukų | 0.861 | 0.687 |
| 2 | 6000 kačių paveiksliukų  6000 šunų paveiksliukų  1000 ančių paveiksliukų | 1000 kačių paveiksliukų  1000 šunų paveiksliukų  100 ančių paveiksliukų | 0,855 | 0,821 |

Iš gautų rezultatų galima įsitikinti, kad tinklas, nesupažindintas su nauja vaizdų klase mokymo metu, negeba atpažinti naujos vaizdų klasės (dėl ko krenta atpažinimo tikslumas). Jei tinklas mokomas su nauja vaizdų klase, jis sėkmingai geba atpažinti visas tris vaizdų klases – kates, šunis ir antis.

# **Priedai**

Priedas Nr. 1

Pradinis konvoliucinis neuroninis tinklas faile *catsdogs.py.*

Priedas Nr.2

Kodas, skirtas naudojantis *Flickr* API parsiųsti paveikslėlius su nurodytu užrašu (*tag*) iš socialinio tinklo *Flickr* pateikiamas faile *img\_scraper.py.*