

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Kursinis darbas

**Objektų atpažinimas ir sekimas kompiuterinės tomografijos
vaizduose**

(Object detection and tracking in computed tomography)

Atliko: 3 kurso Programų sistemų studentas

Paulius Milmantas (parašas)

Darbo vadovas:

Linas Petkevičius (parašas)

Vilnius
2020

Įvadas

Pastaraisiais metais į medicinos sritį labai sparčiai skerbiasi informacinės technologijos. Atliekant įvairias diagnostikas ar tiriant ligas, vaistus yra pasitelkiama programinių sistemų pagalba. Tačiau didžiąją radiologo darbo dalį sprendimus turi priimti jis pats be jokios programinės įrangos pagalbos, nors dažniausiai yra apibrėžti tam tikri ligų aptikimų algoritmatai, kaip vėžinių lastelių aptikimas, lūžiai ir taip toliau. Mano darbo tikslas yra prisidėti prie programinės įrangos kūrimo radiologams ir sukurti programą, kuri aptinka plaučius, kad vėliau tai galėtų būti panaudota tolimesnei IT plėtrai medicinos srityje.

Šiandien viena iš labiausiai perspektyvių sričių informacinių technologijų srityje yra giliojo mokymosi metodai. Jie padeda duomenyse aptikti sunkiai pastebimus dėsningumus ir nuspėti išeities rezultatus su tam tikra klaidos tikimybe. Šiame darbe kalbėsiu vieną iš šių metodų: dirbtinius neuroninius tinklus. Juos pasirinkau nes kompiuterinės tomografijos vaizdai turi labai daug informacijos ir juos apdoruoti naudojant logines taisykles yra per daug sunku. Tačiau net ir jiems ši užduotis yra per sudėtinga, jeigu tiriamos yra įvairios mutacijos arba nedažnos ligos, nes su kiekvienu nauju nematytu ligos atveju, yra reikalinga apmokyti modelį, o tai reikalauja daug duomenų.

Pasirinktam mano dirbtinių neuroninių tinklų metodui viena iš galimų alternatyvų yra "sprendimo medžiai". Tai yra prižiūravimo mokymosi metodas skirtas duomenų klasifikacijai ir duomenų regresijai. Šis metodas naudoja taisyklių rinkinį. Norint gauti mažą šio metodo paklaidą, turime aprašyti vis daugiau šių taisyklių. Taisyklės paprastai būna aprašomos naudojant "if, else" aprašus. Mano nagrinėjamame atvejyje duomenys yra labai dideli ir įvairūs, todėl aprašyti visas šias taisykles užtruktų neproporcingai daug laiko ir tam tikrų radiologijos žinių.

Turinys

Ivadas	1
1. Dirbtinio neuroninio tinklo sudėtis	3
1.1. Bazinė struktūra	3
1.2. Aktyvacijos funkcijos	4
1.3. GPU naudojimas	4
1.4. Poskyris.....	4
1.4.1. Skirsnis	4
1.4.1.1. Straipsnis	4
1.4.2. Skirsnis	4
2. Skyrius	5
2.1. Poskyris.....	5
2.2. Poskyris.....	5
Išvados	6
Literatūra	7
Priedas Nr.1	
Priedas Nr.2	

1. Dirbtinio neuroninio tinklo sudėtis

1.1. Bazinė struktūra

Daugelį dirbtinių neuronų tinklų sudaro panašios struktūrinės dalys:

1. Įvesties sluoksnis: tai dalis kuri priima įvestį ir perduoda kitiems sluoksniams.
2. Išvesties sluoksnis: tai dalis, kuri naudoja aktyvacijos funkciją kuri grąžina galutinį tinklo rezultatą: tikimybių rinkinį, kuris parodo kokia tikimybė, kad objektas atitinka tam tikrą klasę.
3. Paslėptas sluoksnis: perduoda svorius iš praeito sluoksnio į sekantį.
4. Susijungimai ir svoriai: tarp kiekvieno neurono, kuris yra susijungęs, turi savo svorį, pagal kurį yra pakeičiama perduodama reikšmė.
5. Aktyvacijos funkcija: tai funkcija, kokia turi būti neurono išvestis.
6. Mokymosi taisyklė: apibrėžia, kaip tinkle keičiasi svoriai, kad tinklas išvestų norimus rezultatus.

Dirbtinį neuroninį tinklą taip pat sudaro svoriai ir tendenciškumas. Šie du parametrai yra tinklo automatiškai sugeneruojami ir apmokant tinklą pagal duomenis, svoriai ir tendenciškumas yra automatiškai tinklo keičiami. Tendenciškumas apibrėžia kaip tinklo išvestis yra nutolusi nuo tikrosios reikšmės. Tendenciškumas priartina tinklo išvestį prie tikrųjų reikšmių. Svoriai nurodo koks stiprus yra ryšys tarp neuronų, taip nurodant, kaip tinklo reikšmės kinta pačiame tinkle. Jeigu tinklui yra paduodama reikšmė, o pirmasis tinklo neuronas, kuris priima reikšmę turi mažą svorį, tai reiškia, jog galutiniams rezultatui ši reikšmė neturi daug įtakos. Jeigu svoris yra didelis, tai duota reikšmė turi daug įtakos rezultatui.

Svoriai ir tendenciškumas yra tinklo tvarkomi treniravimo proceso metu. prižiūrimasis tinklo treniravimas vyksta dviem žingsniai: priekine ir atgaline sklaida.

TODO

Atgalinė sklaida vyksta, kai tinklas išveda rezultatą, patikrinama ar rezultatas yra teisingas ir atitinkamai tinklas reaguoja į tai. Inicijuojant tinklą svoriai yra nustatomi atsitiktinai. Kiekvienam duomenų rinkiniui yra stebima tinklo išvestis ir tikrinama kokia ji turėtų būti. Padaryta klaida yra perduodama tinklui atgal, per praeitus sluoksnius, einant nuo išvesties sluoksnio. Taip tinklo svoriai yra keičiami atsižvelgiant į padarytą klaidą. Kai klaidos yra padaromos pakankamai mažos ir jos yra mažesnės už duotą ribinę reikšmę, yra laikoma, kad tinklas baigė mokymosi procesą.

1.2. Aktyvacijos funkcijos

1.3. GPU naudojimas

Šiame darbe bus naudojamas GPU, atliekant tinklo treniravimą. Tai yra spartesnis būdas už CPU naudojimą, nes CPU turi kelis galingus branduolius, o GPU žymiai daugiau. Giliajame mokyje, naudojami aritmetiniai veiksmai yra ganėtinai paprasti, todėl juos apdoroti gali ir CPU ir GPU. Tačiau GPU turi daugiau branduolių, kas leidžia atlikti daugiau matematinių veiksmų paralizuotai, negu CPU. Taip yra pasiekiamas didesnis treniravimo našumas.

Didesniam treniravimo našumui pastebėti naudojant GPU, negu CPU, buvo atliktas paprastas bandymas: naudojant Google tensorflow struktūrą, apmokame sistemą atpažinti kur yra žmogaus plaučiai kompiuterinės tomografijos vaizduose su CPU ir GPU. Vidutiniškai apdorojant 60 megabaitų duomenų, vienam ciklui tiklas su CPU resursais užtrunka 6 sekunes, o naudojant GPU vidutiniškai vienas žingsnis užtrunka 0.3 sekundės.

Šiame darbe naudojamas Tensorflow karkasas naudoja NVIDIA Cuda sistemą. CUDA yra paralelinio skaičiavimo platforma, naudojama su grafikos plokštėmis. Ji kiekvieną skaičiavimą paleidžia ant atskiros gijos. Visi duomenys yra suskirstyti į vienos, dviejų ir trijų dimensijų blokus. Kiekvienams blokas gali turėti virš 512 gijų. Visos gijos kurios operuoja viename bloke dalinasi bendra GPU atmintimi. Jeigu norima optimizuoti programą ir paleisti jos skaičiavimus ant GPU, tada reikia susiskaičiuoti kiek vyks paralelinių operacijų ir kiek iš jų dalinsis bendra atmintimi. Pagal apskaičiuotą kiekį nurodome kiek mes norime turėti blokų ir kiek kiekvienas blokas turės atskirų gijų. Prieš kiekvieną operaciją, siekiant pasiekti optimalų našumą, jeigu reikia įkeliamo kintamuosiu į bendrą bloko atmintį. Po atliktų skaičiavimų išvestus rezultatus turime iškelti iš bendros atminties.

1.4. Poskyris

Citavimo pavyzdžiai: cituojamas vienas šaltinis [PPP01]; cituojami keli šaltiniai [Pav05; PPP+02; PPP03; PPP04; STU+02; STU01; STU03; STU04; Sur05].

1.4.1. Skirsnis

1.4.1.1. Straipsnis

1.4.2. Skirsnis

2. Skyrius

2.1. Poskyris

2.2. Poskyris

Išvados

Išvadose ir pasiūlymuose, nekartojant atskirų dalių apibendrinimų, suformuluojamos svarbiausios darbo išvados, rekomendacijos bei pasiūlymai.

Literatūra

- [Pav05] A. Pavardonis. *Magistrinio darbo pavadinimas*. Magistrinis darbas, Universiteto pavadinimas, 2005.
- [PPP+02] A. Pavardenis, B. Pavardonis, C. Pavardauskas ir D. Pavardinskas. Straipsnio pavadinimas. *Rinkinio pavadinimas*, p.p. 3–15, Miestas, šalis. Leidykla, 2002.
- [PPP01] A. Pavardenis, B. Pavardonis ir C. Pavardauskas. Straipsnio pavadinimas. *Žurnalo pavadinimas*, IV:8–17, 2001.
- [PPP03] A. Pavardenis, B. Pavardonis ir C. Pavardauskas. *Knygos pavadinimas*. Leidykla, Miestas, šalis, 2003. 172 psl.
- [PPP04] A. Pavardenis, B. Pavardonis ir C. Pavardauskas. Elektroninės publikacijos pavadinimas. <http://example.com/kelias/iki/straipsnio.pdf>, 2004. 45 KB, tikrinta 2015-02-01.
- [STU+02] A. Surname, B. Tsurname, C. Usurname, and D. Vsurname. Article title. In *Conference book title*, pp. 3–15, City, country. Publisher, 2002.
- [STU01] A. Surname, B. Tsurname, and C. Usurname. Article title. *Journal Title*, IV:3–15, 2001.
- [STU03] A. Surname, B. Tsurname, and C. Usurname. *Book title*. Publisher, City, country, 2003. 172 p.
- [STU04] A. Surname, B. Tsurname, and C. Usurname. Online publication title. <http://example.com/path/to/the/article.pdf>, 2004. 45 KB, accessed 2015-02-01.
- [Sur05] A. Surname. *Ttitle fo PhD thesis*. PhD thesis, Title of university, 2005.

Priedas Nr. 1

Niauroninio tinklo struktūra

1 pav. Paveikslėlio pavyzdys

Priedas Nr. 2

Eksperimentinio palyginimo rezultatai

1 lentelė. Lentelės pavyzdys

Algoritmas	\bar{x}	σ^2
Algoritmas A	1.6335	0.5584
Algoritmas B	1.7395	0.5647