Fakultet elektrotehnike i računarstva

Zavod za primjenjeno računarstvo

**Napredni algoritmi i strukture podataka**

2. laboratorijska vježba

Valerio Franković 0036471547

Zagreb, 10.12.2016.

# Zadatak

**Zadaci za 17 bodova**

1. 1)  Riješiti zadatak s neuronskom mrežom iz skupine za 5 bodova, ali pisanjem vlastitog programa (poželjno C, C++, C# ili Java). Pri tome nije potrebno grafički prikazati izlaz mreže (naravno, nije ni zabranjeno), već je kao dokaz uspješnosti dovoljno izračunati neku objektivnu mjeru (brojčani pokazatelj).

Posebno razrađena, domišljata i uspješna rješenja zadataka iz skupine za 11 bodova. Procjena asistenta!

# Rješenje zadatka

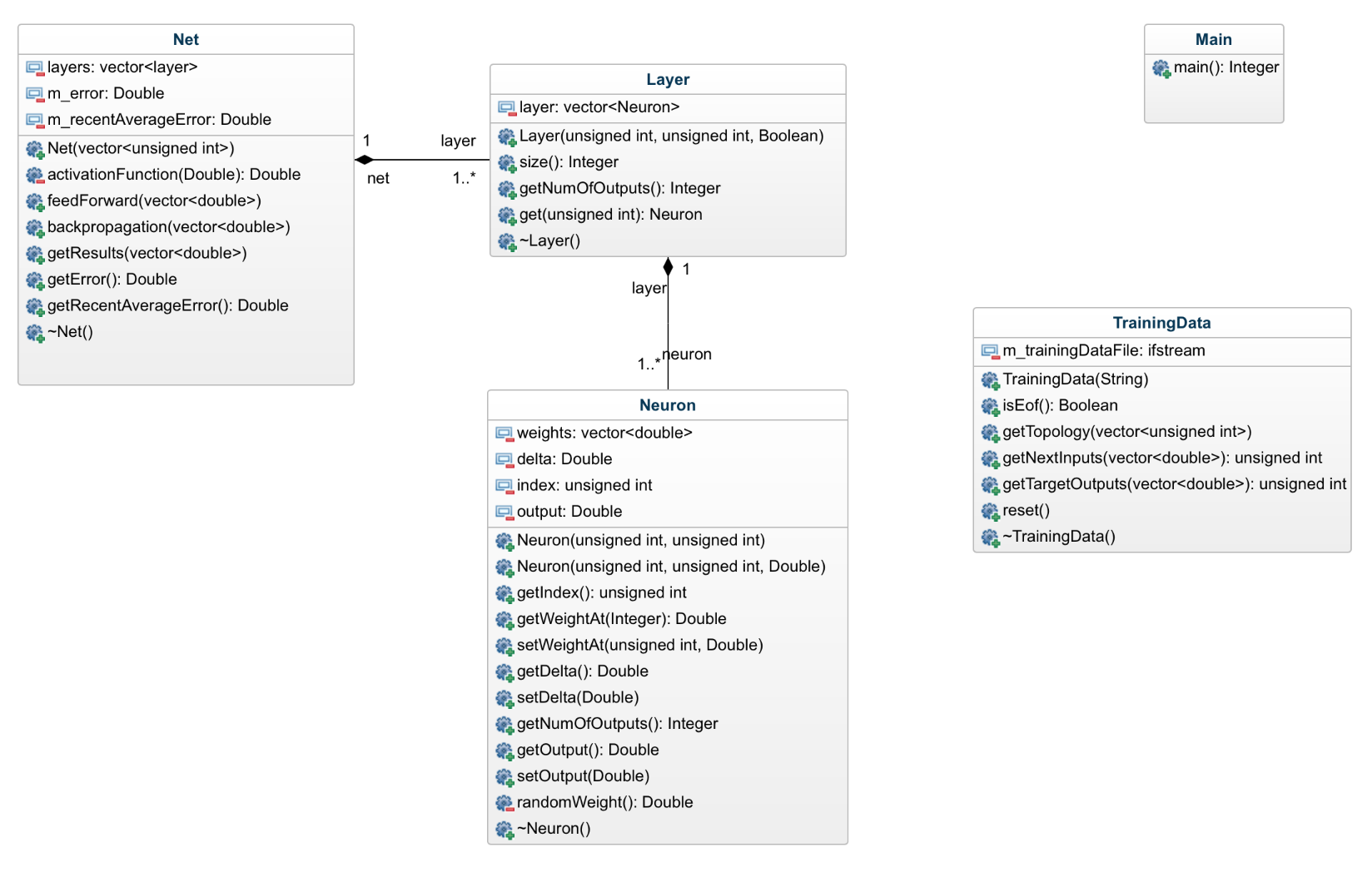
## Teorijski uvod

Umjetna neuronska mreža (engl. *artifical neural network*) je programski ili sklopovski sustav građen od jednakih, jednostavnih i međusobno povezanih dijelova (čvorova) koji može obavljati raznovrsne i složene zadaće ovisno o parametrima dijelova i veza među njima. Ideja umjetnih neuronskih mreža temelji se na uvjerenju da rad ljudskoj mozga (uspostavljanje veza mežu neuronima) može biti imitiran koristeći silicij i žice kao zamjenu za neurone i dendrite; sastavljene su od mnogo čvorova koji imitiraju biološke neurone. Neuroni su međusobno spojeni poveznicama te komuniciraju jedan s drugim. Grupirani su u slojeve koje dijelimo na ulazni, skriveni i vanjski.

## Implementacija

Neuronska mreža je isprogramirana u programskom jeziku *C++* koristeći *Eclipse CDT*.

Korišteno je četiri razreda – *Net*, *Layer*, *Neuron* i *TrainingData* te metoda *main*. Razred *Net* je programska apstrakcija neuronske mreže. Sadrži vektor slojeva te metode koje omogućuju rad mreže, poput *feed forward* i *backpropagation*. Razred *Layer* sadrži vektor neurona, a svaki pojedini neuron sadrži vrijednosti poput težina, izlaza te nekih pomoćnih variabli i metoda. Programske knjižnice koje su upotrijebljene radi implementacije su *<iostream>*, *<cmath>*, *<vector>*, *<cstdlib>*, *<fstream>*, *<sstream>*. UML dijagram klasa može se vidjeti u nastavku.



*Slika 2.1 - UML dijagram razreda*

### Unaprjeđivanje

Unaprjeđivanje signala se izvodi tako da se za izlazne vrijednosti ulaznih neurona postave vrijednosti uzorka koji se želi evaluirati. Zatim se na temelju njih, svakom pojedinom neuronu u svakom pojedinom sloju izračuna *net* vrijednost koja se provuče kroz sigmoidu. Za izlaz se uzima izlaz signala na izlaznim neuronima.

Pseudokod je dan u nastavku:

**Net::feedForward(inputVals):**

postavi neuronima ulaznog sloja izlaze na inputVals

za svaki sloj počevši od skrivenog

dohvati trenutni i prijasnji sloj

za svaki neuron u sloju

izračunaj net po formuli

postavi izlaz na net provučen kroz sigmoidu

### Backpropagation

Težine osvježavamo formulom:

,

gdje je

.

Raspisivanjem i deriviranjem se dođe do formula za ažuriranje težina koje dijelimo na formulu za ažuriranje težina neurona izlaznog sloja i formulu za ažuriranje težina neurona skrivenog sloja.

Težine neurona prema izlaznom sloju ažuriramo po sljedećoj formuli:

,

gdje je

.

Težine neurona prema nekom od slojeva skrivenog sloja ažuriramo po sljedećoj formuli:

,

gdje je

U ovoj zadaći se težine ažuriraju stohastički po formuli:

za svaki uzorak iz skupa.

Pseudokod je dan u nastavku:

**Net::backpropagation(outputVals):**

krenuvši od posljednjeg do prvog sloja

svim neuronima izračunaj delte i ujedno ažuriraj težine

### Prikaz rezultata

Prikaz rezultata se vrši tako da se metodi prenese vektor u koji se trebaju upisati izlazne vrijednosti neurona izlaznog sloja. Pseudokod je dan u nastavku:

**Net::getResults(resultVals):**

isprazni ulazni vektor

dodaj u njega sve izlazne vrijednosti neurona izlaznog sloja

prikazi na ekranu vrijednosti tog vektora

### Dohvaćanje ulaznih vrijednosti

Ulazne vrijednosti se dohvaćaju iz tekstualne datoteke koja mora biti posebno strukturirana. Konkretna, prvi linija u datoteci mora sadržavati strukturu mreže, a ostali redci sadrže informacije o ulaznim i izlaznim vrijednostima. *Main* metoda spremi datoteku u *wrapper* *TrainingData*. Zatim za fiksan broj ponavljanja provede postupak treniranja mreže nakon čega se ista testira.

# Zaključak

Umjetne neuronske mreže možemo primijeniti na razne probleme poput regresije i klasifikacije. Neke od prednosti umjetnih neuronskih mreža su rekordna točnost na cijeli set problema uključujući prepoznavanje slika i zvuka, analiza teksta i tokova vremena itd. No, nedostatci su što ih može biti teško namjestiti da dobro uče, time ih je teško *debugirati*. Drugo, ne objašnjavaju zašto dolaze do određenih zaključaka te treće, računalno su vrlo zahtjevne.

# Literatura

[1] N. Hlupić, D. Kalpić. 2009. Neuronske mreže (*pdf*). Dostupno na: [http://www.fer.unizg.hr/\_download/repository/Neuronske\_mreze.pdf](http://www.fer.unizg.hr/_download/repository/Strukture_podataka.pdf)

[2] M. Čupić, B. D. Bašić, M. Golub, *Neizrazito, evolucijsko i neuroračunarstvo*. 2013, pp. 131-168.