Višeslojni perceptroni

Predmet: Neuronske mreže

Mentor: prof. Dr Zora Konjović

Autori: Dejan Stokić, Milan Dobrota, Miloš Janjić

Sadržaj

[1. Uvod u višeslojne perceptrone 3](#_Toc294993923)

[2. Učenje 4](#_Toc294993924)

[2.1. Algoritam prostiranja greške unazad – Backpropagation 4](#_Toc294993925)

[3. Optimizacija strukture veštačke neuronske mreže (pruning) 6](#_Toc294993926)

[3.1 Optimal Brain Damage algoritam 6](#_Toc294993927)

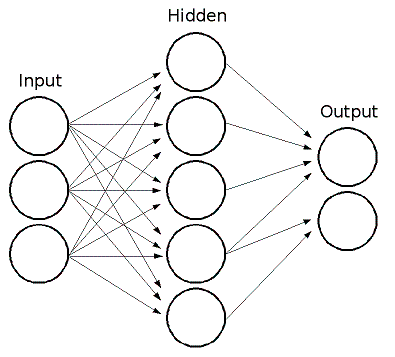
[4. Iris dataset 8](#_Toc294993928)

[5. Implementacija Optimal Brain Damage algoritma 9](#_Toc294993929)

[Reference 10](#_Toc294993930)

# 1. Uvod u višeslojne perceptrone

Višeslojni perceptroni predstavljaju tip veštačkih neuronskih mreža sa jednim ili više skrivenih slojeva. Ovakav tip veštačkih neuronskih mreža je acikličan tj. signal se prostire unapred, od ulaznih neurona prema neuronima skrivenog sloja, sa jednog skrivenog sloja na drugi i sa poslednjeg skrivenog sloja na neurone izlaznog sloja. Struktura višeslojnog perceptrona je prikazana na slici 1.1.

  
Slika 1.1 – struktura višeslojnog perceptrona

Svaki neuron ima nelinearnu aktivacionu funkciju uz pomoću koje se generiše njegov izlaz. Takvi izlazi se putem veze prenose do neurona sledećeg sloja. Svaka veza ima težinu, i ona simbolizuje jačinu biološke veze između dva neurona. Težina veze (), pomnožena sa izlazom (*x*) iz neurona prethodnog sloja se prenosi na svaki od neurona iz sledećeg sloja. Zbir svih ulaza u neuron čini težinsku sumu.

Data formula se odnosi na izračunavanje težinske sume pri prenosu signala od sloja čiji su neuroni označeni sa *j*, ka neuronu označenog sa *i*.

# 2. Učenje

Kao i u slučaju ostalih neuronskih mreža potrebno je izvršiti proces učenja. Postoje dve vrste učenja - sa i bez nagledanja, pri čemu se poslednje više koristi. Da bi se dato učenje izvršilo potrebno je imati obučavajući skup, koji se sastoji od različitih vektora ulaza, sa željenim vektorom izlaza iz celokupne mreže. Neobučena mreža generiše rezultat koji se razlikuje od željenog, što predstavlja grešku. Svrha celokupnog procesa učenja jeste da se ta greška minimizuje do određene granice. Postoje različiti načini redukcije greške. Najčešće korišćeni algoritam jeste algoritam prostiranja greške unazad (Backpropagation Learning Algorithm).

## 2.1. Algoritam prostiranja greške unazad – Backpropagation

Backpropagation algoritam raspoređuje grešku izlaznog sloja na prethodne slojeve, vršeći korekciju težina veza. Greška u izlaznom neuronu označenim sa *j* za ulazni vector *n* predstavljena je sledećom formulom:

gde *d* predstavlja željeni izlaz datog neurona, a *y* njegov trenutni izlaz. Ukupna greška neuronske mreže se računa na sledeći način:

Pre početka obuke potrebno je usvojiti:

* Obučavajući skup (skup ulaznih i izlaznih podataka)
* Vrednost koraka učenja
* Kriterijum za prekid algoritma
* Način korekcije težina sinapsi
* Aktivacionu funkciju (najčešće se koristi sigmoida)
* Inicijalne vrednosti težina sinapsi (obično mali slučajni brojevi)

Pravilo obuke definiše da je promena težina srazmerna negativnom gradijentu greške izlaza:

Da bi se odredila vrednost prethodnog izraza primenjuje se pravilo ulančavanja izvoda:

U slučaju sigmoidne aktivacione funkcije, za l=L-1 (težine izlaznog sloja) može se pisati sledeći izraz:

Kombinovanjem datih formula dobija se:

Gde je, za neurone izlaznog sloja:

dok je za neurone skrivenog sloja:

Proces određivanja gradijenata i korekcija težina se ponavlja dok funkcija greške *E* ne dostigne svoj minimum.

Na osnovu prethodnih izraza se uočava da faktor *δl* zavisi od greške određene na sloju *l+1*, zbog čega se ovaj metod i zove prostiranja greške unazad.

# 3. Optimizacija strukture veštačke neuronske mreže (pruning)

Mreže koje se koriste za rešavanje praktičnih problema veoma često postaju velike i kompleksne. Iz tih razloga, javlja se potreba za redukciju broja nepotrebnih neurona u unutrašnjosti mreže (skrivenim slojevima). Jedan od načina za optimizaciju strukture veštačke neuronske mreže jeste metoda Optimal Brain Damage.

## 3.1 Optimal Brain Damage algoritam

Osnovna ideja Optimal Brain Damage algoritma jeste da se redukcijom broja neurona u skrivenim slojevima može uprostiti celokupna struktura mreže, uz minimalno povećanje greške, ili, u nekim slučajevima, njeno smanjenje. Ovakva optimizacija najviše dolazi do izražaja u slučajevima gde je obučavajući skup ograničene veličine. Dokazano je, i praktično, i teorijski, da kompleksne mreže imaju problema sa uopštavanjem rezultata, dok mreže manje veličine često nemaju sposobnost da precizno predstave obučavajući skup. Takva konstantacija opravdava postojanje ovog algoritma.

Suština Optimal Brain Damage algoritma se sastoji u uklanjanju veza sa najmanjom važnošću. One veze koje imaju najmanji uticaj na promenu greške pri učenju se mogu odstraniti, posle čega se mreža može ponovo obučavati. Ovakva procedura se može ponavljati sve dok je greška mreže manja od unapred odredjene maksimalne željene greške. U slučaju prekoračenja, mreža se vraća u prethodno stanje.

Da bi se utvrdilo koja veza ima najmanju važnost, posmatra se analitička promena greške mreže u slučaju njenog uklanjanja. Promena greške *E* je predstavljena sledećom formulom:

gde *hii* predstavlja dijagonalni element Hesijanove matrice, dok je *u* element parametarskog vektora.

Da bi se izvršilo izračunavanje elemenata Hesijanove matrice potrebno je odrediti stanje mreže. Stanje neurona *xi* je odredjeno sledećom jednačinom:

gde *f* predstavlja aktivacionu funkciju. Ukupni ulaz u neuron (težinska suma) predstavlja zbir pojedinačnih ulaznih težinskih suma određenog neurona.

Dijagonalni član Hesijanove matrice je predstavljen sledećom formulom:

Dati sabirak se može izračunati na sledeći način:

U određenim slučajevima, umanjilac prethodne jednačine se može zanemariti.

Optimal Brain Damage procedura se dakle sastoji iz sledećih koraka:

1. Odabiranje početne arhitekture mreže.
2. Obučavanje mreže dok se ne postigne prihvatljiva ukupna greška.
3. Računanje Hesijanove matrice za svaki parametar.
4. Računanje važnosti za svaki parametar.
5. Brisanje parametra sa najmanjom važnošću.
6. Ponavljanje koraka 2-6 sve dok se greška nalazi u prihvatljivim granicama.

# 4. Iris dataset

Iris dataset (skup podataka) predstavlja jednu od najpoznatijih baza koje postoje u literature vezanoj za prepoznavanje šablona. Skup podataka predstavljaju tri klase od po 50 instanci, gde svaka klasa ukazuje na jednu vrstu biljke perunike (iris). Jedna klasa je linearno odvojiva od ostale dve, dok te dve preostale nisu međusobno linerano odvojive. Kreator ovog skupa podataka je Ronald Aylmer Fisher, koji ga je predstavio 1936. godine kao primer diskriminantne analize.

# 5. Implementacija Optimal Brain Damage algoritma

Pri realizaciji Optimal Brain Damage algoritma korišćen je softverski paket *Neuroph* pisan u Java programskom jeziku. Ovaj algoritam je primenjen nad objektom klase MultiLayerPerceptor. Pre svake iteracije vrši se čuvanje postojeće strukture mreže u datoteku, sve dok greška ne postane neprihvatljiva, u kom slučaju se vrši učitavanje prethodno sačuvane strukture. Korišćena je sigmoidna aktivaciona funkcija. Neuronska mreža ima četiri ulaza i tri izlaza i jedan skriven sloj. Klasifikacija je ostvarena na sledeći način:

Setosa – 1,0,0

Versicolor – 0,1,0

Virginica – 0,0,1

Optimalan broj neurona nakon primene Optimal Brain Damage algoritma nije konstantan. Razlog te pojave jeste upravo stohastična priroda obučavanja mreže.

# Reference

[1] Yann Le Cun, John S. Denker, Sara A. Solla – Optimal Brain Damage

[2] R.A. Fisher - Iris Data Set (1936) (<http://archive.ics.uci.edu/ml/datasets/Iris>)

[3] Neuroph – (<http://neuroph.sourceforge.net/>)

[4] Multilayer perceptron – (<http://en.wikipedia.org/wiki/Multilayer_perceptron>)

[5] Stephan Dlugosz - Multi-layer Perceptron Networks for Ordinal Data Analysis