

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**SEMINAR**

**Vizualizacija umjetne neuronske mreže u  
JavaScriptu**

*Matija Bertović*

Voditelj: *Marko Čupić*

Zagreb, Travanj, 2018.

## **Sadržaj**

1. Uvod.....	1
2. Opis umjetne neuronske mreže.....	3
3. Prikaz implementacije.....	9
4. Zaključak.....	11
5. Literatura.....	12

## 1. Uvod

Ideja proučavanja i pokušaja implementacije umjetnih neuronskih mreža proizlazi iz naše svakodnevice. Razmislimo li gdje se danas obrađuje najveća količina podataka, doći ćemo do jednostavnog zaključka, a to je živčani sustav živih organizama. Živčani sustav se sastoji od velikog broja živčanih stanica (neurona) koji su povezani u jednu vrlo složenu mrežu. Neuroni obrađuju podatke tako da šalju različite živčane impulse ostalim neuronima s kojima su povezani. Upravo taj složeni i zanimljivi sustav poslužio je kao motivacija za razvoj novog pristupa izgradnji tehničkih sustava, poznatog kao *konektivistički pristup*. Konektivistički pristup temelji se na izgradnji sustava koji, umjesto da se njegovo ponašanje programira, sam uči na temelju prijašnjeg iskustva.

Pokušavajući oponašati rad živčanog sustava grade se jednostavne jedinice (umjetni neuroni) koje se zatim povezuju u složenije paralelne strukture nazvane umjetne neuronske mreže (engl. *Artificial Neural Networks*). Upravo proučavanjem umjetnih neuronskih mreža bavi se *neuro-računarstvo* kao grana *računarstva* iz skupine *mekog računarstva*.

Umjetne neuronske mreže široku primjenu nalaze u području *klasifikacije i funkcijske regresije*. Klasifikacija je postupak određivanja klase kojoj ulazni uzorak pripada, a funkcijska regresija je postupak pri kojem se na temelju poznatih podataka pokušava aproksimirati funkcija koja ih opisuje.

Umjetne neuronske mreže se konkretno koriste u raznim područjima kao što su:

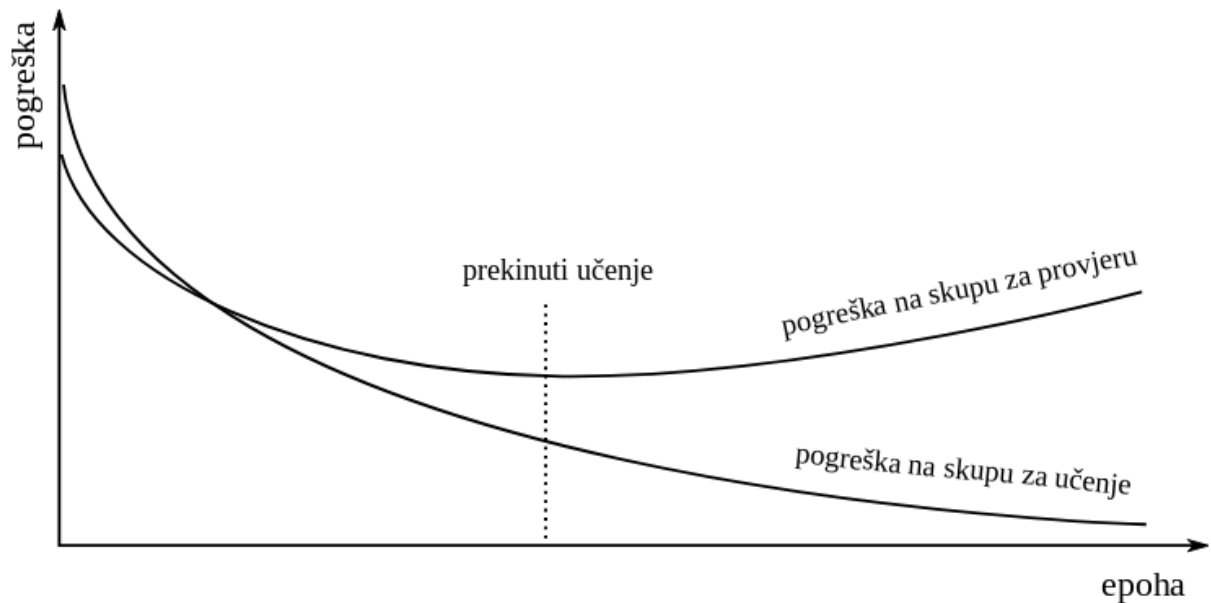
- **Financije.** Koriste se kao pomoć u donošenju raznih odluka u području financija na temelju povijesnih podataka.
- **Kompresija slika.**
- **Obrada jezika.** Koriste se za pretvaranje teksta u govor i obrnuto.
- **Obrada signala.** Koriste se primjerice za otkrivanje izvora signala, uklanjanje šumova u signalima, detekciju signala.
- **Prepoznavanje pisanih znakova.** Prepoznavanje rukom pisanih ili tiskanih znakova i njihova obrada i pretvorba u tekst na računalu.

Umjetna neuronska mreža prije iskorištavanja mora proći kroz fazu učenja. U toj fazi, ona obrađuje razne uzorke iz skupa uzoraka za učenje i time mijenja jačine veza između pojedinih neurona, baš kao što živčane stanice stvaraju nove veze u procesu našeg učenja. Ona može podešavati jačine svojih veza nakon svakog uzorka ili tek nakon cijelog skupa uzoraka te se prema tome učenje dijeli na *pojedinačno učenje* (engl. *on-line learning*) i *grupno učenje* (engl. *batch learning*). *Iteracija* je predočavanje jednog uzorka mreži, a *epoha* predočavanje cijelog skupa uzoraka.

Ako neuronskoj mreži prilikom učenja predočimo preveliki skup uzoraka, ona nakon nekog vremena prestane popravljati sposobnost generalizacije i podatke kreće učiti napamet. To, svakako, ne želimo te je od velike važnosti uočiti trenutak kada mreža kreće gubiti sposobnost generalizacije i tada prekinut učenje.

U svrhu praćenja sposobnosti umjetne neuronske mreže, skup uzoraka dijelimo na tri dijela: *skup za učenje* (engl. *training set*), *skup za provjeru* (engl. *validation set*) i *skup za testiranje* (engl. *testing set*). Mreži u fazi učenja predočavamo uzorke iz skupa za učenje, a povremeno ju testiramo na skupu za provjeru. Važno je naglasiti da prilikom provjere mreže nad skupom za provjeru nju držimo fiksnom, odnosno ona ne mijenja svoje jačine veza među neuronima. To činimo iz tog razloga što onda

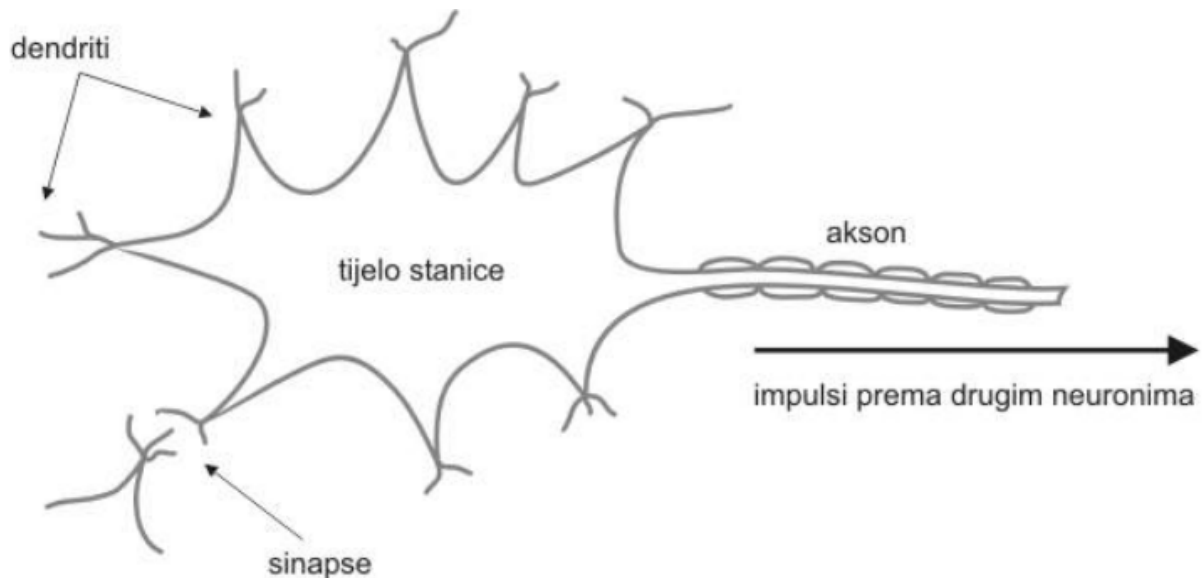
možemo pratiti kako mreža mijenja svoje performanse. Ako ona napamet nauči neke smetnje i šumove iz skupa za učenje, krenut će povećavati svoju pogrešku nad skupom za provjeru i mi ćemo to moći detektirati te prekinuti učenje. Na poslijetku, kada je postupak učenja gotov, kvalitetu mreže provjeravamo na skupu za testiranje, a ta se provjera radi samo jednom. Opisani postupak prikazan je na sljedećoj krivulji kretanja pogrešaka na skupu za učenje i skupu za testiranje na slici 1 (preuzeto iz [1]).



Slika 1: Kretanje pogrešaka pri učenju umjetne neuronske mreže

## 2. Opis umjetne neuronske mreže

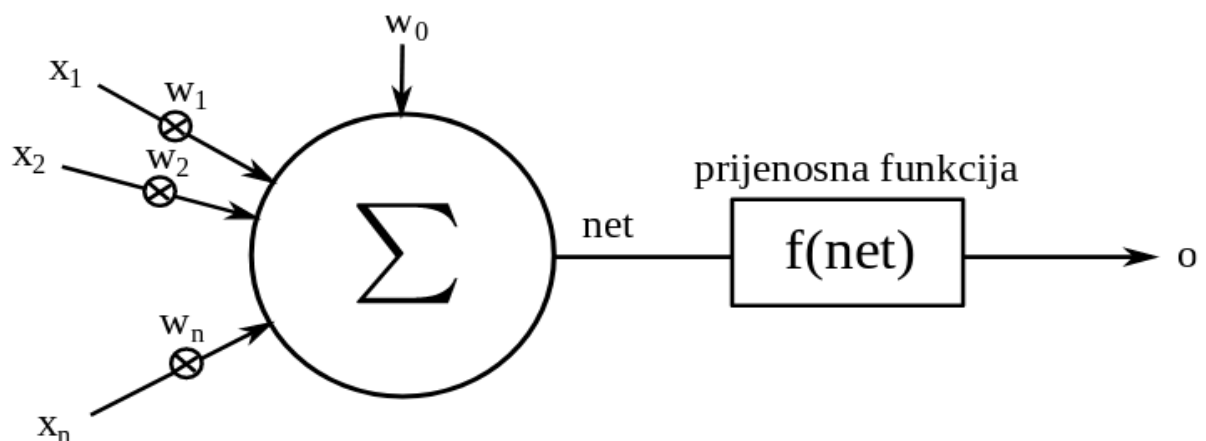
Umjetna neuronska mreža sastoji se od određenog broja umjetnih neurona. Zato bi prije opisa kompletne mreže valjalo opisati sami umjetni neuron. Umjetni neuron simulira biološki neuron. Prisjetimo se građe biološkog neurona (slika 2, preuzeto iz [1]).



Slika 2: Biološki neuron.

U *tijelu stanice* nalazi se jezgra. Na rubovima stanice nalaze se *dendriti* koji primaju impulse okolnih neurona. Preko *aksona*, dugačkog kraka stanice, ona šalje impulse prema drugim neuronima. Ovisno o jačini *sinaptičkih veza*, mijenja se sposobnost prijenosa impulsa.

Tim znanstvenika Warren McCulloch i Walter Pitts definirali su 1943. godine model umjetnog neurona na temelju strukture biološkog neurona (slika 3, preuzeto iz [1]).



Slika 3: Model umjetnog neurona

Umjetni neuron ima ulaze koji oponašaju dendrite. Oni su prikazani varijablama  $x_1, x_2, \dots, x_n$ . Preko njih neuron prima podražaje ili od drugih neurona ili iz vanjskog svijeta. Svaki od tih ulaza utječe na neuron, a težine  $w_i$  određuju jačine tih utjecaja. Težine mogu biti pozitivne i negativne. Djelovanje pojedinog ulaza na neuron određeno je umnoškom  $x_i \cdot w_i$ .

Tijelo stanice umjetnog neurona određuje ukupni potencijal stanice i generira ga na svom izlazu, tj. na ulazu aksona. Ukupni potencijal označava se oznakom *net* a računa se sumom:

$$net = \sum_{i=1}^n x_i \cdot w_i + w_0 \quad (2.1)$$

Možemo definirati da svaki naš neuron ima još i jedan ulaz  $x_0$  čija vrijednost će uvijek biti 1 te onda formulu (2.1) možemo zapisati kao:

$$net = \sum_{i=0}^n x_i \cdot w_i = \vec{x} \cdot \vec{w} \quad (2.2)$$

Gdje su  $\vec{x}$  i  $\vec{w}$   $(n + 1)$ -dimenzionalni vektori, a *net* njihov skalarni produkt.

Akson umjetnog neurona predstavlja prijenosnu funkciju (engl. *transfer function*) koja određuje izlaz *o* neurona:

$$o = f(net) \quad (2.3)$$

Danas se koristi više različitih prijenosnih funkcija kao što su npr. funkcija identiteta, funkcija skoka, sigmoidalna funkcija, funkcija tangens hiperbolni. Grafovi navedenih funkcija prikazani su na slici 4 (preuzeto iz [1]). Sigmoidalna prijenosna funkcija definirana je izrazom

$$sigm(net) = \frac{1}{1 + e^{-net}} \quad (2.4)$$

koja je zanimljiva jer se njena derivacija može prikazati pomoću nje same kao

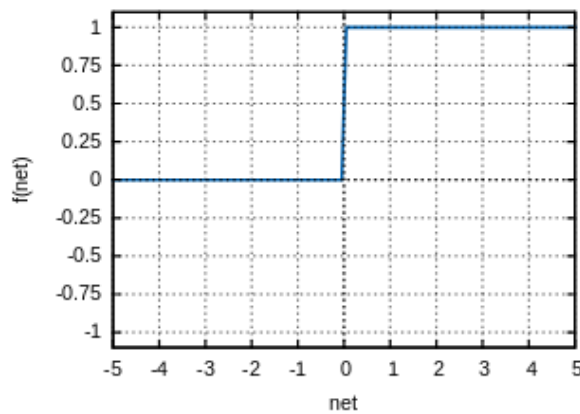
$$\frac{d \, sigm(net)}{d \, net} = sigm(net) \cdot (1 - sigm(net)) \quad (2.5)$$

Poseban slučaj umjetnog neurona koji kao prijenosnu funkciju koristi funkciju skoka naziva se *TLU-perceptron* (engl. *Threshold Logic Unit*). Težine perceptrona podešavaju se njegovim učenjem. Učenje je opisano *algoritmom učenja perceptrona* (preuzet iz [1]):

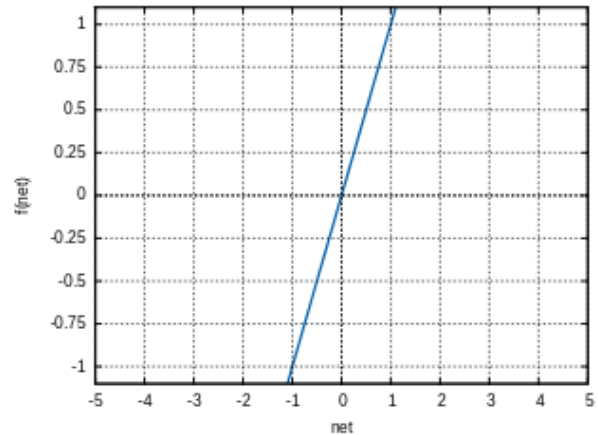
1. Ciklički prolazi kroz svih  $N$  uzoraka za učenje, jedan po jedan.
2. Klasificiraj trenutni uzorak.
  - (a) Ako se klasificira korektno, ne mijenjaj težine i
    - i. ako je to  $N$ -ti uzastopni uzorak klasificiran korektno, prekini učenje jer perceptron uz trenutni skup težina sve klasificira korektno,

- ii. inače prijeđi na sljedeći uzorak.
- (b) Ako se ne klasificira koretno, korigiraj težine perceptrona prema sljedećem izrazu, pri čemu je  $t$  željena vrijednost na izlazu neurona, a  $o$  je dobivena vrijednost:

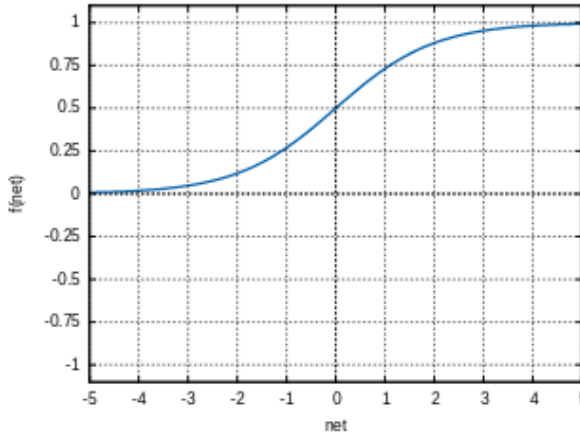
$$w_i(k+1) \leftarrow w_i(k) + \eta \cdot (t - o) \cdot x_i \quad (2.6)$$



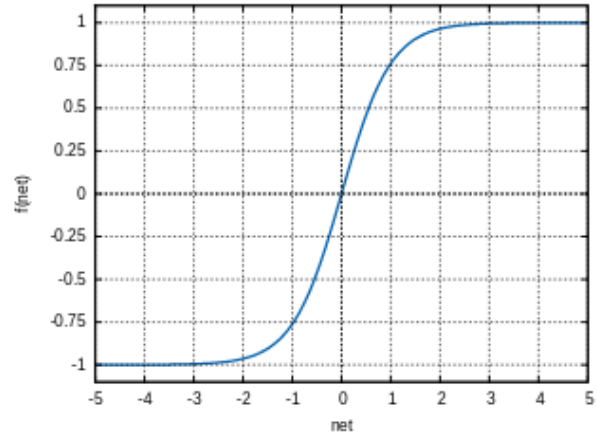
(a) Funkcija skoka  $f(net) = \text{step}(net)$



(b) Funkcija identiteta  $f(net) = net$



(c) Sigmoidalna funkcija  $f(net) = \text{sigm}(net)$



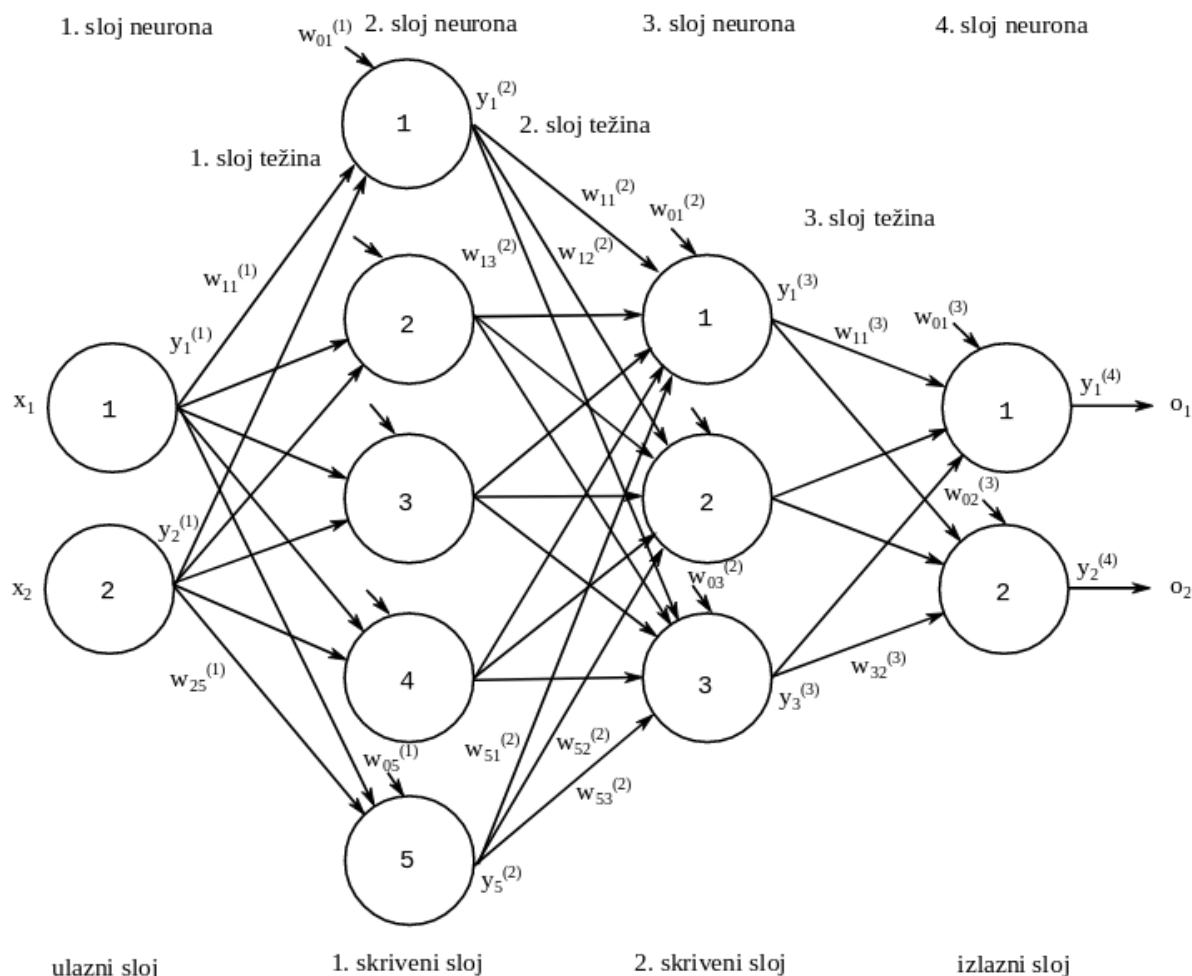
(d) Funkcija tangens hiperbolni  $f(net) = \text{tanh}(net)$

Slika 4: Različite prijenosne funkcije.

Povezivanjem umjetnih neurona tvorimo umjetnu neuronsku mrežu. Postoji više različitih oblika umjetnih neuronskih mreža, a mi ćemo se baviti *potpuno povezanim unaprijednim slojevitim umjetnim neuronskim mrežama* (engl. *Fully-connected Feed-forward Multi-layered*). To je mreža u kojoj su neuroni grupirani u slojeve, svaki neuron iz sloja  $i$  dobiva na ulaz pobude svih neurona isključivo iz sloja  $i - 1$  te u mreži ne postoje ciklusi u obradi podataka.

Mreža se sastoji od slojeva umjetnih neurona. Prvi sloj mreže sadrži neurone koji ne obavljaju nikakvu funkciju nego je njihov zadatak na svoj izlaz direktno staviti podatak koji su dobili na ulazu. Slijede skriveni slojevi koji na ulaze dobivaju izlaze svih neurona iz prethodnog sloja te na izlaz postavljaju podatak obrađen prema gore opisanom postupku. Posljednji sloj je izlazni sloj. Na izlazima neurona izlaznog sloja očitavamo konačni podatak kojega je mreža izračunala. S obzirom na brojeve neurona u svakom sloju, arhitekturu mreže možemo jednostavnije zapisati navođenjem tih brojeva. Na primjer, arhitekturu četveroslojne mreže koja u ulaznom sloju ima 2, svakom od dva skrivena sloja po 3 i jedan izlazni umjetni neuron kraće zapisujemo kao  $2 \times 3 \times 3 \times 1$ .

Prije razrade algoritma učenja unaprijednih neuronskih mreža, trebamo dogovoriti oznake pojedinih dijelova mreže. Slojeve neurona i neurone unutar sloja numerirat ćemo počevši od 1. Izlaz  $i$ -tog neurona u  $k$ -tom sloju označit ćemo s  $y_i^{(k)}$ . Rekli smo da izlazi neurona u prvom sloju odgovaraju njihovim ulazima, dakle  $y_i^{(1)} = x_i$ , a izlaze neurona u izlaznom sloju označit ćemo s  $o_i = y_i^{(K)}$  gdje je  $K$  ukupan broj slojeva mreže. Težinu koja se nalazi između izlaza  $i$ -tog neurona u sloju  $k$  i ulaza  $j$ -tog neurona u sloju  $(k + 1)$  označit ćemo sa  $w_{ij}^{(k)}$ , a težinu  $w_0$   $i$ -tog neurona u sloju  $(k + 1)$  označit ćemo sa  $w_{0i}^{(k)}$ . Na slici 5 (preuzeto iz [1]) možemo vidjeti dogovorene oznake nekih dijelova mreže.



Slika 5: Prikaz umjetne neuronske mreže i dogovorenih oznaka



Osnovni algoritam učenja unaprijednih neuronskih mreža naziva se *algoritam propagacije greške unatrag*. U ovom radu ću se baviti isključivo unaprijednim neuronskim mrežama sa sigmoidalnom prijenosnom funkcijom.

Ako dimenzionalnost ulaznog prostora označimo sa  $N_i$ , a izlaznog sa  $N_o$ , skup uzoraka za učenje biti  $\{(x_{1,1}, \dots, x_{1,N_i}) \rightarrow (t_{1,1}, \dots, t_{1,N_o}), \dots, (x_{N,1}, \dots, x_{N,N_i}) \rightarrow (t_{N,1}, \dots, t_{N,N_o})\}$ . Rezultat mreže nakon učitano  $s$ -tog uzorka, ovisno o trenutnim težinama označit ćemo sa  $(o_{s,1}, \dots, o_{s,N_o})$ , a očekivani rezultat sa  $(t_{s,1}, \dots, t_{s,N_o})$ . Kvadratno odstupanje za  $s$ -ti uzorak računamo formulom:

$$E(s) = \frac{1}{2} \sum_{i=1}^{N_o} (t_{s,i} - o_{s,i})^2 \quad (2.7)$$

Ukupnu prosječnu pogrešku koju mreža radi nad svim uzorcima za učenje označit ćemo sa  $E$ , a računati prema formuli:

$$E = \frac{1}{N} \sum_{s=1}^N E(s) = \frac{1}{2 \cdot N} \sum_{s=1}^N \sum_{i=1}^{N_o} (t_{s,i} - o_{s,i})^2 \quad (2.8)$$

Algoritam propagacije greške unatrag (prikazan u nastavku) dobiva se primjenom algoritma gradijentnog spusta na problem minimizacije funkcije pogreške. Izvod za potrebe ovog rada nije neophodno prikazati. (Algoritam je preuzet iz [1])

1. Sve težine neuronske mreže postavi na slučajne vrijednosti.
2. Ponavljaj dok nije zadovoljen uvjet zaustavljanja
  - (a) Za svaki uzorak  $s: (x_{s,1}, \dots, x_{s,N_i}) \rightarrow (t_{s,1}, \dots, t_{s,N_o})$  iz skupa uzoraka za učenje čini:
    - i. Postavi podatak  $(x_{s,1}, \dots, x_{s,N_i})$  na ulaz mreže.
    - ii. Izračunaj izlaze svih neurona u svim slojevima, od prvog prema posljednjem. Označimo izlaze neurona posljednjeg sloja s  $(o_{s,1}, \dots, o_{s,N_o})$ .
    - iii. Odredi pogrešku svakog od neurona *izlaznog* sloja. Pogrešku  $i$ -tog izlaznog neurona označit ćemo s  $\delta_i^K$  i računati prema izrazu:

$$\delta_i^K = o_{s,i} \cdot (1 - o_{s,i}) \cdot (t_{s,i} - o_{s,i})$$

gdje je  $(t_{s,i} - o_{s,i})$  učinjeno odstupanje između željene vrijednosti i dobivene vrijednosti, a član  $o_{s,i} \cdot (1 - o_{s,i})$  predstavlja derivaciju prijenosne funkcije tog neurona.

- iv. Potom se vraćaj sloj po sloj prema početku mreže. Za  $i$ -ti neuron koji se nalazi u  $k$ -tom sloju (gledamo samo skrivene slojeve), pogrešku neurona računamo kao težinsku sumu pogrešaka neurona kojima on šalje svoj izlaz, pomnoženu derivacijom prijenosne funkcije tog neurona:

$$\delta_i^{(k)} = y_i^{(k)} \cdot (1 - y_i^{(k)}) \cdot \sum_{d \in \text{Downstream}} w_{i,d} \cdot \delta_d^{(k+1)}$$

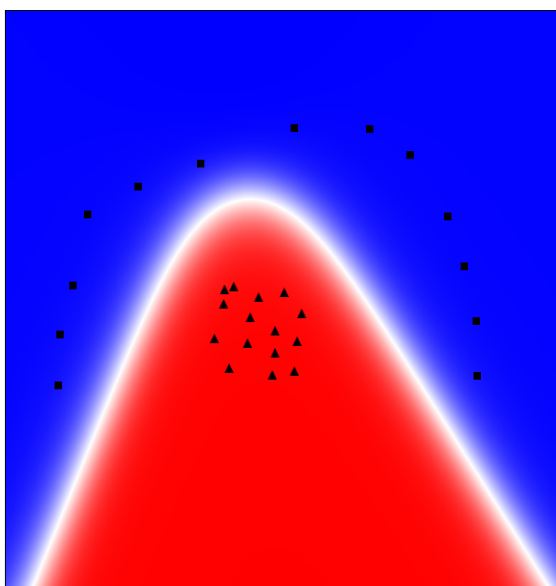
v. Napraviti korekciju svih težina. Težinu  $w_{i,j}^{(k)}$  korigiraj prema izrazu:

$$w_{i,j}^{(k)} \leftarrow w_{i,j}^{(k)} + \eta \cdot y_i^{(k)} \cdot \delta_j^{(k+1)}$$

U formulama je broj  $\eta$  *stopa učenja*. To je unaprijed određeni pozitivni broj dosta malog iznosa. Rekli smo da se na početku sve težine postavljaju na proizvoljne vrijednosti. Ipak, preporuča se da broj ne bude prevelik, a jedno od pravila koje daje dobre rezultate je da se za neuron koji ima  $m$  ulaza svaku težinu postavi na proizvoljni broj iz intervala  $[-2.4/m, +2.4/m]$ .

### 3. Prikaz implementacije

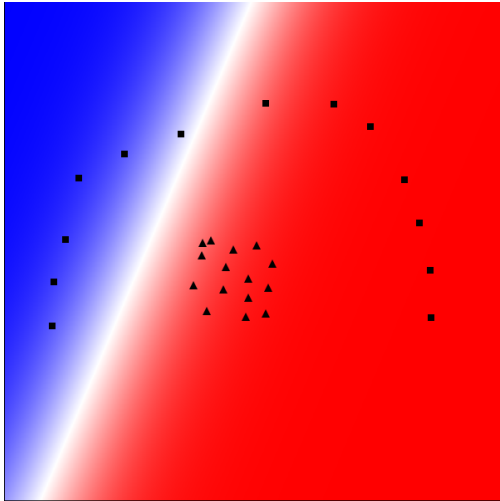
Gore upisanom unaprijednom neuronskom mrežom sa sigmoidalnom funkcijom prijenosa možemo obaviti klasifikaciju uzoraka. Primjer klasifikacije uzoraka prikazat ću programom u JavaScript-u. Pretpostavimo da imamo koordinatni sustav  $[0, 1] \times [0, 1]$ . U njemu možemo izabrati nekoliko točaka i podijeliti ih u dvije različite skupine. Želimo da naša mreža, učenjem na temelju zadanih točaka bude u mogućnosti aproksimirati kojoj će skupini pripadati bilo koja točka sustava. U ovom primjeru koristit ću neuronsku mrežu arhitekture  $2 \times 2 \times 1$ , odnosno imat će dva ulazna neurona, jedan skriveni sloj sa dva neurona i jedan izlazni neuron. Kao izlaznu funkciju koristit ću sigmoidalnu prijenosnu funkciju. Ona daje izlaze na intervalu  $(0, 1)$ . U ovom primjeru, elemente jedne skupine označavat ćemo „kvadratićem” (u daljnjem tekstu skupina  $R_1$ ), a druge „trokutićem” (u daljnjem tekstu skupina  $R_2$ ). Konačni izlaz neuronske mreže nakon učenja možemo vidjeti na slici 6.



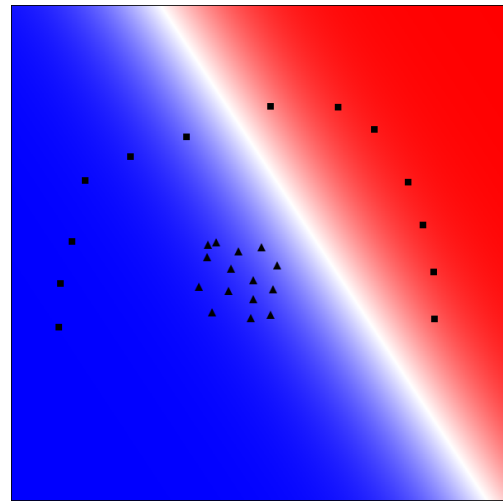
Slika 6: Izlaz neurona izlaznog sloja 0.5, odnosno skupinu  $R_2$  boja je crvena.

Granica između te dvije skupine naziva se *decizijska granica* i područje oko nje označeno je sve svijetlijom, odnosno bijelom bojom. Dakle svaku točku iz „plavog” područja, mreža će klasificirati kao da je dio skupine  $R_1$ , a iz „crvenog” područja kao dio skupine  $R_2$ .

Zanimljivo je pogledati izlaze oba neurona skrivenog sloja (slike 7 i 8). Vidimo da je svaki od ta dva neurona napravio neku (različitu) granicu između ove dvije skupine. Promatrajući sliku 6, vidimo da granica na lijevoj strani slike otprilike odgovara granici koju prikazuje izlaz prvog neurona skrivenog sloja, a granica na desnoj strani slike otprilike odgovara izlazu drugog neurona skrivenog sloja. Konačni izlaz dobiven je kombinacijom tih dviju granica, a na mjestu njihovog spajanja dobiven je obli rub.



Slika 7: Izlaz prvog neurona skrivenog sloja



Slika 8: Izlaz drugog neurona skrivenog sloja

## 4. Zaključak

Rješavajući programski dio naišao sam na jedan problem. Prvo sam imao neskaliране grafove pa su iz tog razloga veličine ulaza u mrežu bile poprilično velike. Zbog velikih ulaza, a nedovoljno malih težina ulaza u neurone, težinska suma tih ulaza bila bi dovoljno velika da djelujući sigmoidalnom funkcijom na nju uvijek dobijemo broj koji je otprilike jednak 1. Iz tog razloga sam većinom dobivao krive rezultate jer se težine nisu dobro ili uopće korigirale. Analiziranjem algoritma, uspio sam pronaći problem. U formuli za uzračun pogreške neurona nalazi se član  $y_i^{(k)} * (1 - y_i^{(k)})$ . Taj član je većinom davao 0 i to je bitno utjecalo na rezultat. Ovaj problem poznat je kao *problem nestajućeg gradijenta* (engl. *vanishing gradient problem*). Što se više argument sigmoidalne funkcije udaljava od 0, to više gore navedeni umnožak teži k 0, dok je za argument 0 umnožak maksimalan i jednak 0.25. Problem sam riješio skaliranjem grafa na  $[0, 1] \times [0, 1]$  čime sam dobio dovoljno male brojeve da rezultat bude zadovoljavajuć. Ovo rješenje je za mene radilo jer imam dovoljno malo slojeva u mreži. Povećanjem broja slojeva navedeni problem sve više dolazi do izražaja.

## **5. Literatura**

[1] Čupić, M. (2017.) Umjetne neuronske mreže.