Neuronske mreže za početnike

Reni Banov[1], Anđa Valent[2], Judita Anušić[3]

Ovim člankom želja nam je mlade matematičare i zaljubljenike u računalne znanosti upoznati s neuronskim mrežama. Zadnjih dvadesetak godina primjena neuronskih mreža značajno se proširila na različite probleme i pri tome se obično koriste implementirani gotovi modeli. Primjene su često usko specijalizirane, a sami modeli nerijetko su već toliko sofisticirani da mogu obeshrabriti čitatelja početnika.

Naš je cilj objasniti osnovnu matematičku pozadinu neuronskih mreža tako da, uz malu pomoć nastavnika, bude razumljiva učenicima srednjih škola. U dva uvodna poglavlja ukratko ćemo prikazati ciljeve i matematički model neuronskih mreža, a potom ćemo model ilustrirati na jednostavnim primjerima. Jedina matematička predznanja potrebna za razumijevanje sadržaja su: elementarne logičke funkcije (*AND*, *OR*, *XOR*), jednadžba pravca u ravnini i jednadžba ravnine u prostoru, te osnove matričnog računa (množenje vektora matricom).

Uvod

Ideju modeliranja biološke funkcije mozga zasnivaju autori McCulloch i Pitts [1] sredinom prošlog stoljeća, predlažući uporabu logičkih sklopova za opisivanje neuroaktivnosti. Kako je u to doba razvoj računala bio tek u fazi osnovne izvedbe, trebalo je proći neko vrijeme da se razvije model koji bi računalnim programom mogao djelomično opisivati biološku aktivnost te vrste. Krajem pedesetih godina osnove modernih neuronskih mreža za implementaciju računalnim programima postavio je eksperimentalni psiholog Rosenblatt [2] predlažući model Perceptrona (1958). Usprkos tomu što je u svojem radu i kasnijem izvješću Rosenblatt [3] koristio ideju Perceptron modela s više međuslojeva, krajem šezdesetih godina autori Minsky i Papert [4] oštro kritiziraju njegov rad zasnovan na jednostavnom Perceptron modelu, što u konačnici dovodi do znatnog smanjenja novčane potpore za istraživanja na tom području.

U svojoj knjizi Minsky i Papert uzimaju za primjer Perceptron bez međuslojeva, te pokazuju kako se tim modelom ne može riješiti jednostavan problem opisivanja logičke XOR funkcije. Sažetak njihove kritike sadržan je u riječima: *kažeš kako će računala biti svjesna, a ne mogu riješiti XOR problem*. Autori su bili izrazito oštri prema Rosenblattu te

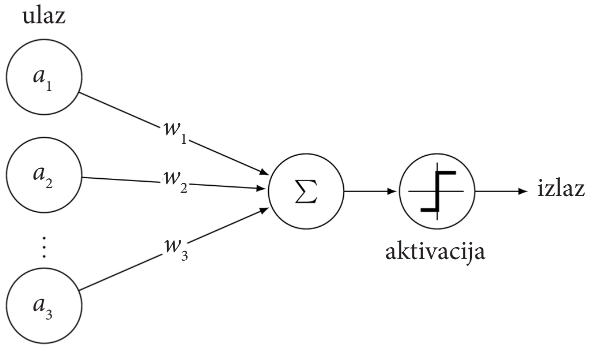
[1]Reni Banov, Tehničko veleučilište u Zagrebu
[2]Anđa Valent, Tehničko veleučilište u Zagrebu
[3]Judita Anušić, studentica Tehničkog veleučilišta u Zagrebu

su ideju Perceptrona okarakterizirali bezvrijednom, što je eksplicitno izrečeno u citatu iz uvoda na četvrtoj stranici njihove knjige: *Most of this writting ... is* ***without scientific value*** *and we will not usually refer by name to the works we criticize.* Samog Rosenblatta kritika je teško pogodila, zbog čega je napustio istraživanja, da bi dvije godine kasnije, na svoj 43. rođendan, tragično preminuo u pomorskoj nesreći u zaljevu Chesapeake.

Takva je kritika dovela do nepunih dvadeset godina stagnacije istraživanja ideje Perceptrona, koja su ponovno pokrenuta sredinom osamdesetih u radovima brojnih autora [5,6], od kojih izdvajamo samo nekolicinu: Rumelhart, McClelland, Anderson, Rosenfeld, Hopfield, Lippmann, itd. Navedeni autori u svojim radovima postavljaju osnove modernih neuronskih mreža zasnovanih na višeslojnom modelu Perceptrona, koje svoj pravi procvat doživljaju posljednjih godina pojavom snažnih računala s mogućnošću paralelne obrade podataka. Nedvojbeno je ideja Perceptrona značajno pridonijela utjecaju razvoja računalstva pa ju je Lewis [7] zasluženo uvrstio na popis ideja koje su odredile budućnost računalne znanosti. U ovom ćemo radu prikazati interpretaciju modela višeslojnih neuronskih mreža pomoću elementarne matematike.

Višeslojne neuronske mreže

Osnovni Perceptron model zasniva se na McCulloch-Pitts opisu jednog biološkog neurona. U biološkom neuronu signal iz drugih neurona prenosi se putem sinaptičkih veza na dendrite te doprinosi povećanju ili smanjenju njegove pobude. Nakon što dostigne određenu razinu pobude, neuron odašilje vlastiti izlazni signal koji se putem aksona prenosi na druge neurone. Sinaptičke veze između dendrita uspostavljaju se elektrokemijskim putem i tipično su ostvarene između 10^3 – 10^4 susjednih neurona [8]. Budući da se moždana kora sastoji od prosječno 10^{11} neurona, možemo zaključiti kako mozak predstavlja složen paralelni sustav obrade informacija. Vrijeme prijenosa signala sinaptičkom vezom tipično iznosi nekoliko milisekunda, tako da, primjerice, prepoznavanje lica druge osobe koje traje tek sekundu čini proces u moždanoj kori od nekoliko stotina paralelnih koraka. S obzirom na to da se sinaptičkom vezom prenosi mala količina informacija (nekoliko bita), jasno je kako je takav paralelni sustav iznimno učinkovit i optimiziran evolucijskim razvojem.



Slika 1. McCulloch-Pitts model biološkog neurona

Vratimo se na McCulloch-Pitts model biološkog neurona. Model možemo zapisati jednostavno pomoću skalarnog umnoška vektora **(a)** ulaznih aktivacija i vektora **(w)** intenziteta sinaptičkih veza (težine)

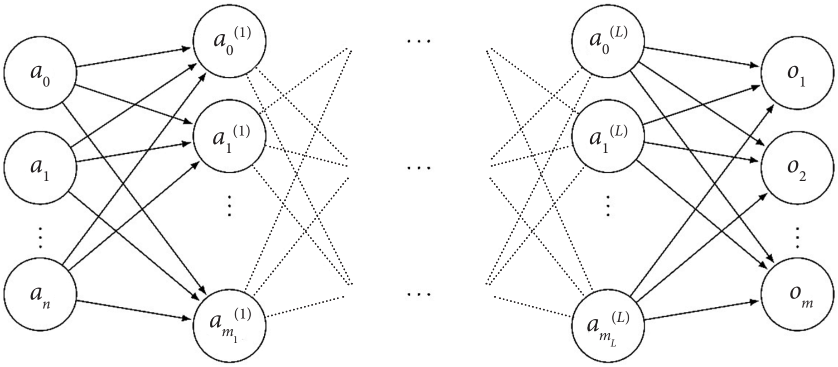
$$\mathbf{w}^T \cdot \mathbf{a} = \sum_{i=1}^{n} w_i a_i$$

te realne funkcije σ koja simulira aktivaciju neurona uz dodatno normiranje izlaznog signala. U modernim implementacijama neuronskih mreža aktivacijske funkcije izabiru se prema prirodi problema za koji se priprema neuronska mreža te se najčešće upotrebljavaju: po dijelovima konstantna, po dijelovima linearna, logistička (sigmoidna, hiperbolni tangens), itd. Uobičajeno se na vrijednost skalarnog umnoška dodaje posmak b (eng. *bias*) najčešće sa svrhom poboljšanja performansi neuronske mreže. Stoga model jednog neurona možemo opisati pomoću vrijednosti funkcije

$$\sigma(\mathbf{w}^T \cdot \mathbf{a} + b).$$

Primijetimo kako je u modelu jednog neurona nelinearnost određena aktivacijskom funkcijom. Naime, odaberemo li za aktivacijsku funkciju po dijelovima konstantnu ili linearnu, dobit ćemo isti takav model. Složeni sustav možemo graditi prijenosom vrijednosti aktivacijske funkcije na druge umjetne neurone. To znači da mrežu umjetnih neurona formiramo množenjem matrice težina **(W)** s vektorom vrijednosti aktivacijske funkcije za vektor **(a)** ulaznih neurona. Slikovito takav model predstavlja paralelni sustav koji simulira aktivaciju sloja bioloških neurona u višeslojnoj neuronskoj mreži. Pomoću vektora i matrica taj model možemo zapisati vrijednostima (realni vektor) funkcije

$$\sigma : \mathbb{R}^n \to \mathbb{R}^m \Rightarrow \mathbf{a}^{(i+1)} = \sigma\left(\mathbf{W}^{(i)}\mathbf{a}^{(i)} + \mathbf{b}^{(i)}\right) \quad \text{za aktivaciju } (i+1) \text{ sloja.}$$



Slika 2. Neuronska mreža s L međuslojeva

Slika 2. prikazuje primjer neuronske mreže koja se sastoji od ulaznog **(a)** sloja, L međuslojeva ($\mathbf{a}^{(1)}\ldots\ \mathbf{a}^{(L)}$) i izlaznog **(o)** sloja. Tipično se na takav način umrežuje više slojeva umjetnih neurona, s time da početni sloj neurona (vektor **a**) predstavlja ulazni, a završni sloj (vektor **o**) izlazni podatak za neuronsku mrežu.

Međuslojevi u mreži nazivaju se skrivenim slojevima (eng. *hidden layer*) te se formiraju s različitim brojem neurona (indeks m_i) ovisno o tipu i aktivnosti koju neuronska mreža simulira. Primjerice, drukčiji oblici neuronskih mreža koriste se za prepoznavanje oblika od onih koji se upotrebljavaju za optimizacijske probleme. Topološke razlike (vidjeti [8]) u obliku mreže određene su prirodom problema koji se rješava te načinom (učenjem) formiranja matrice težina. Važno je napomenuti kako se tijekom procesa učenja formira matrica težina i vektor posmaka za **svaki** sloj neuronske mreže osim za ulazni.

Izvedba funkcija AND i OR neuronskom mrežom

Krenut ćemo od najjednostavnijih primjera neuronskih mreža za logičke funkcije – AND i OR. Dva su razloga zašto krećemo od ove dvije funkcije: prvi je što je riječ o logičkim funkcijama s kojima je upoznat gotovo svaki učenik srednje škole, a drugi je što ih je moguće riješiti jednostavnim Perceptron modelom sa samo jednim ulaznim slojem neurona.

Kao ulazne vrijednosti (domenu) obje logičke funkcije imaju uređene parove (x, y), pri čemu $x, y \in \{0, 1\}$, a izlazne vrijednosti (slika) iz istog su skupa. Uobičajeno ih možemo definirati tablično (tablicom istinitosti):

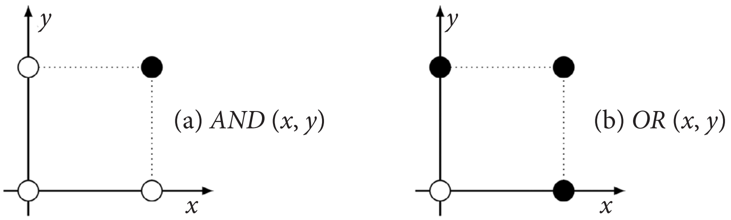
x	y	$AND\ (x, y)$	$OR\ (x, y)$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Tablica 1. Tablica istinitosti logičkih funkcija AND, OR

Geometrijski, vrijednosti obiju funkcija možemo jednostavno prikazati točkama u xy ravnini. Ulazne vrijednosti su vrhovi jediničnog kvadrata, to jest točke s koordinatama

$$(0, 0), (1, 0), (1, 1), (0, 1),$$

a izlazne vrijednosti prikazane su punim (crnim) ili praznim (bijelim) kružićem. Prazan kružić označava vrijednost 0, dok puni kružić označava vrijednost 1 za logičku funkciju u točki (x, y).



Slika 3. AND,OR funkcije kao vrhovi jediničnog kvadrata

Za aktivacijsku funkciju neurona uzet ćemo po dijelovima konstantnu funkciju

$$\sigma(x)=\begin{cases}1 & x>0\\ 0 & x\leq 0\end{cases}.$$

Vratimo se na oznake uvedene u uvodnom dijelu i podsjetimo se da je model jednog neurona opisan pomoću vrijednosti funkcije

$$\sigma(\mathbf{w}^T\cdot\mathbf{a}+b).$$

Ulazne vrijednosti (x, y) možemo zapisati pomoću vektora

$$\mathbf{a}=\begin{bmatrix}x\\ y\end{bmatrix}\qquad x,y\in\{0,1\}$$

a naš je cilj pronaći vektor $\mathbf{w}$ i posmak $b\in\mathbb{R}$ tako da vrijedi

$$\sigma(\mathbf{w}^T\cdot\mathbf{a}+b)=f(\mathbf{a}).$$

Kada vektor težine zapišemo na sljedeći način

$$\mathbf{w}=\begin{bmatrix}w_1\\ w_2\end{bmatrix},$$

zadatak se svodi na problem određivanja brojeva w_1, w_2, b za koje vrijedi

$$f(x,y)=\sigma(w_1x+w_2y+b).$$

Problem sada ima vrlo lijepu geometrijsku interpretaciju koja će nam pomoći da na jednostavan način dođemo do rješenja. Detaljno ćemo ju ilustrirati na primjeru funkcije $f(x,y)=AND(x,y)$, a vjerujemo kako temeljem toga primjera čitatelji mogu samostalno naći rješenje za funkciju $f(x,y)=OR(x,y)$.

Uočimo da vrijednost funkcije σ ovisi isključivo o predznaku izraza w_1x+w_2y+b. Kad je taj izraz pozitivan, funkcija poprima vrijednost 1, a kada je 0 ili negativan, funkcija poprima vrijednost 0. Kako bi se rezultat podudarao s funkcijom $f(x,y)=AND$, naš je zadatak „naštimati” brojeve w_1, w_2, b za koje izraz w_1x+w_2y+b poprima pozitivne vrijednosti u točki koja je na slici 4. označena punim kružićem, te negativne vrijednosti u točkama koje su označene praznim kružićem. Sada, naravno, prepoznajemo da rezultat ovisi o jednadžbi

$$w_1x+w_2y+b=0,$$

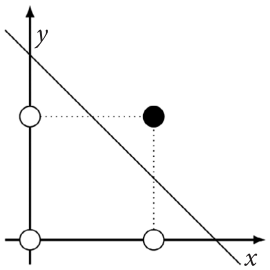
što predstavlja jednadžbu pravca u ravnini. Dakle, trebamo naći jednadžbu bilo kojeg pravca u ravnini koji će razdijeliti „puno” i „prazno” obojene točke. Očigledno rješenje nije jedinstveno, ali je suština da postoji pa prema tome možemo pronaći barem jedno. Primjerice, možemo uzeti

$$w_1 = 1,\ w_2 = 1,\ b = -1.5,$$

čime dobijemo jednadžbu pravca

$$y=-x+1.5,$$

kao što prikazuje Slika 4.



Slika 4. Geometrijska ilustracija rješenja za AND funkciju

Kako bismo se dodatno uvjerili da naš model neuronske mreže zaista rješava problem *AND* funkcije, rezultat ćemo provjeriti pomoću tablice istinitosti:

x	y	$AND\,(x, y)$	$\sigma\,(x + 1 - 1.5)$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Dakle, pokazali smo kako se *AND* funkcija može modelirati jednostavnim Perceptron modelom sa samo jednim neuronom, pri čemu je

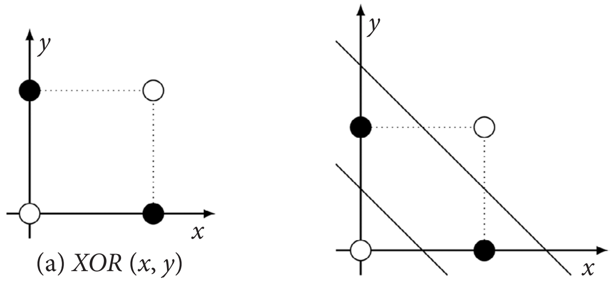
$$\mathbf{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad b = -1.5\,.$$

Vjerujemo da, prateći ovaj detaljno raspisani primjer, zainteresirani čitatelj može samostalno naći rješenje za *OR* funkciju.

Nelinearnost modela neuronske mreže na primjeru XOR funkcije

Nelinearnost modela u neuronskoj mreži, osim na razini aktivacijske funkcije, možemo uvesti na razini ulaznog i/ili drugih slojeva. Uvođenjem nelinearnosti na razini međuslojeva tipično određujemo topologiju neuronske mreže, tj. određujemo način i tip povezivanja neurona u međuslojevima. Na ulaznom nivou nelinearnost se uvodi povećanjem dimenzije ulaznog vektora, tj. dimenzije domene funkcije *f*. Oba načina uvođenja nelinearnosti možemo ilustrirati na jednostavnom *XOR* problemu iz uvoda.

Podsjetimo se da logička funkcija *XOR* za dva ulazna bita daje jedinicu na izlazu samo ako su ulazni bitovi različiti. Ponovno možemo tu funkciju geometrijski prikazati na isti način kao funkcije *AND*, *OR* vrijednostima u vrhovima jediničnog kvadrata. Na Slici 5. vidimo njezin prikaz, te bi pitanje linearnog modela koji će odrediti rezultat *XOR* funkcije bio problem nalaženja jednog pravca koji dijeli kvadrat na dva dijela tako da se u jednom dijelu kvadrata nalaze samo istobojni kružići. Sa slike lako uočavamo kako, za razliku od ranijih primjera za *AND* i *OR* funkcije, za *XOR* funkcije problem nije moguće riješiti jednim pravcem, ali je moguće napraviti podjelu s dva pravca u kojoj svaki dio kvadrata sadrži samo istobojne kružiće. Podjela jednim pravcem odgovara Perceptron modelu jednog neurona koji su kritizirali Minsky i Papert, dok je podjela s dva pravca neuronska mreža s jednim međuslojem.



(a) *XOR* (x, y)

Slika 5. XOR funkcija i geometrijska interpretacija rješenja

Riješimo *XOR* problem s neuronskom mrežom koja ima jedan međusloj. Za aktivacijsku funkciju neurona ponovno uzimamo po dijelovima konstantnu

$$\sigma(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}.$$

Kako bismo odredili matricu težina i vektor posmaka, pogledajmo najprije Sliku 5. koja prikazuje geometrijsko rješenje problema pomoću dva pravca:

$$-x - y + 1.5 = 0, \quad x + y - 0.5 = 0.$$

Postavimo matricu težina i vektor posmaka za srednji i izlazni sloj na sljedeći način:

$$\mathbf{W}^{(0)} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{b}^{(0)} = \begin{bmatrix} 1.5 \\ -0.5 \end{bmatrix}, \quad \mathbf{W}^{(1)} = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} -1.5 \end{bmatrix}$$

Ulazni vektor predstavlja vrijednosti dviju varijabla x, y, tj. dimenzija ulaznog prostora je $n = 2$, dok je dimenzija izlaznog sloja $m = 1$, te je neuronska mreža određena transformacijama

međusloj $\quad \mathbf{a}^{(1)} = \sigma\left(\mathbf{W}^{(0)}\mathbf{a} + \mathbf{b}^{(0)}\right)$

izlazni sloj $\quad \mathbf{o} = \sigma\left(\mathbf{W}^{(1)}\mathbf{a}^{(1)} + \mathbf{b}^{(1)}\right).$

Odredimo funkciju (model)

$$f : \mathbb{R}^2 \to \mathbb{R}$$

koja je definirana ovom neuronskom mrežom. Za ulazni vektor

$$\mathbf{a} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad x, y \in \{0,1\}$$

najprije izračunamo izraz za međusloj mreže

$$\mathbf{a}^{(1)} = \begin{bmatrix} \sigma(-x-y+1.5) \\ \sigma(x+y-0.5) \end{bmatrix}$$

te njegovim uvrštenjem u transformaciju za izlazni sloj dobijemo konačni oblik tražene funkcije

$$f(x,y) = \sigma\big(\sigma(-x-y+1.5)+\sigma(x+y-0.5)-1.5\big)$$

Možemo jednostavno provjeriti kako dobivena funkcija zaista rješava *XOR* problem. U sljedećoj tablici dane su njezine vrijednosti za ulazni vektor:

x	y	$f(x,y)$
0	0	0
0	1	1
1	0	1
1	1	0

Riješimo sada isti problem uvođenjem nelinearnosti na ulaznom sloju neurona, tj. riješimo *XOR* problem s modelom Perceptrona. Uz istu aktivacijsku funkciju σ povećajmo dimenziju ulaznog vektora s varijablom xy, tako da vektor ulaznih neurona sada glasi:

$$\mathbf{a} = \begin{bmatrix} x \\ y \\ xy \end{bmatrix} \qquad x, y \in \{0,1\},$$

dok su vektor težina i posmak zadani s

$$\mathbf{w} = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad b = -0.5\,.$$

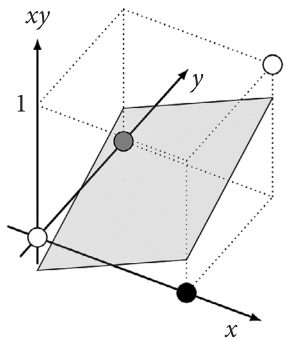
Perceptron model koji rješava *XOR* problem tada je određen izrazom

$$\sigma\left(\mathbf{w}^T \cdot \mathbf{a} + b\right)$$

te predstavlja funkciju

$$f(x,y) = \sigma(x+y-2xy-0.5)$$

koja također ima istu tablicu vrijednosti kao i prethodna funkcija. Iako su analitički različite, ove dvije funkcije ipak određuju isti skup vrijednosti neuronske mreže. Ujedno je jedno od važnih svojstava neuronskih mreža da kroz postupak učenja dolazimo do različitih funkcija (modela) koje rješavaju zadani problem. Naravno, moglo bi se postaviti pitanje kako odabrati (i naučiti) neuronsku mrežu za rješavanje određenih klasa problema. O tome, međutim, nekom drugom prilikom!



Slika 6. Perceptron rješenje XOR problema

Geometrijska interpretacija toga modela prikazana je na Slici 6. Perceptron model zapravo je ravnina određena s tri točke u prostoru:

$$A\left(1,0,\frac{1}{4}\right), B\left(0,1,\frac{1}{4}\right), C\left(1,1,\frac{3}{4}\right)$$

pa njezin vektor normale predstavlja vektor težine za Perceptron model.

Napomena. Uvođenjem dodatnog neurona (varijabla xy) na ulaznom sloju u Perceptron modelu ustvari je primijenjena tehnika jezgrenih metoda (eng. *kernel method*). Na taj način Perceptron model postaje nelinearni model za rješavanje *XOR* problema, bez potrebe za uvođenjem međusloja. Jezgrene metode popularne su za definiranje bogate klase konvolucijskih neuronskih mreža [9].

Korisno je znati kako neuronska mreža samo s jednim neograničenim međuslojem, bez obzira koju aktivacijsku funkciju koristimo, može aproksimirati bilo koju drugu funkciju. Taj važan teorem (vidjeti [10]) dokazan je davne 1989. godine na samim početcima primjene neuronskih mreža. Nešto je kasnije dokazano kako neuronska mreža s neograničenim brojem međuslojeva, koji imaju najmanje $n+m+2$ neurona, može također aproksimirati svaku funkciju.

Pogled matematičara na problem pamćenja i poopćenja

Neuronske mreže pripadaju metodama umjetne inteligencije (eng. *Artificial Intelligence*). Drugim riječima, namjena im je simulirati određene biološke funkcije mozga. Osnovnu jedinicu moždane kore čine neuroni koje možemo smatrati nekom vrstom jednostavnog računala premda je puno sporiji od modernih računala. Međutim, mozak brzinu nadoknađuje velikim brojem neurona te na mozak možemo gledati kao na složeni sustav paralelnih računala. Osim toga, mozak ima jedno važno svojstvo – ne rađamo se s unaprijed ugrađenim znanjima, već se naš mozak razvija *učenjem* temeljem našeg *iskustva*.

Na sličan način funkcioniraju i neuronske mreže. Temeljem ulaznih podataka prolaze *proces učenja* čiji je cilj pripremiti ih za primjenu na konkretnim problemima. Možemo slobodno reći kako se neuronskim mrežama primarno koristimo za *predviđanje* rezultata. Iz navedenog slijedi kako za neuronsku mrežu u osnovi razlikujemo dva problema: problem *pamćenja* kroz fazu učenja i problem *poopćenja* kroz fazu primjene.

Tijekom faze učenja u neuronskoj se mreži formira matrica težina i parametri aktivacijske funkcije na osnovi zadanih vrijednosti neurona na ulaznom sloju i očekivane (točne) vrijednosti neurona na izlaznom sloju. Ovisno o tipu i namjeni neuronske mreže primjenjuju se različiti modeli učenja povratnom vezom, a djelomičan popis može se pronaći u literaturi [8,9]. Budući da ulazni sloj neurona predstavlja realni vektor $\mathbf{a} \in \mathbb{R}^n$, dok izlazni sloj predstavlja realni vektor $\mathbf{o} \in \mathbb{R}^m$, zaključujemo kako je faza učenja neuronske mreže zapravo postupak nalaženja funkcije (modela)

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

za koju će na zadanom skupu ulazno-izlaznih parova vrijediti

$$f(x) = y, \forall (x, y) \in D \subset \mathbb{R}^n \times \mathbb{R}^m$$

Idealno bi bilo kada bi se kroz fazu učenja u neuronskoj mreži „zapamtio” cjelokupan zadani skup za učenje *D* (eng. *training set*). U praksi se, naravno, ideal rijetko kada dostiže (osim za male *D* skupove), stoga se kroz fazu učenja traži funkcija koja će minimizirati odstupanje od zadanih vrijednosti. Drugim riječima, faza učenja predstavlja određivanje funkcije *f* za zadanu funkciju pogreške

$$f_e : D \times \mathbb{R}^m \to \mathbb{R}$$

koja je definirana na trojkama $(x, y, f(x))$. Tipičan primjer za funkciju pogreške predstavlja kvadrat razlike

$$f_e(x, y, f(x)) = (f(x) - y)^2$$

te se pomoću nje može definirati problem pamćenja kao problem određivanja funkcije *f* koja minimizira sljedeći model na zadanom skupu

$$\sum_{(x,y) \in D} f_e(x, y, f(x))$$

Dakako, postavlja se pitanje kako minimizirati takav model za neuronsku mrežu. Njegovo rješavanje predstavlja postupak određivanja matrice težina i parametara za odabranu funkciju *f* iz obitelji aktivacijskih funkcija kod neuronske mreže. Primjerice, za po dijelovima konstantne ili linearne aktivacijske funkcije relativno je jednostavno naći parametre koji daju globalni minimum, dok se kod sigmoidnih i drugih nelinearnih aktivacijskih funkcija najčešće gradijentnim postupkom određuje lokalni minimum, a samo je ponekad nađeni lokalni minimum ujedno i globalni. Rješavanjem problema pamćenja možemo „naučiti” neuronsku mrežu da zapamti sve točne odgovore za jednostavnije probleme, primjerice *XOR* problem iz uvoda.

Vidjeli smo u prethodnom dijelu kako je faza učenja neuronske mreže upućena na problem pamćenja. Međutim, neuronska mreža najčešće se koristi za predviđanje rezultata nakon faze učenja. Slično definiciji skupa za učenje možemo definirati skup vrijednosti na kojem se provjerava točnost predviđanja rezultata iz neuronske mreže. Uz korištenje iste funkcije pogreške neuronska se mreža u fazi primjene tipično dodatno trenira na tom skupu. Drugim riječima, faza predviđanja za neuronske mreže čini problem određivanja funkcije (najčešće korekcija postojeće)

$$\tilde{f} : \mathbb{R}^n \to \mathbb{R}^m \Rightarrow \tilde{f}\Big|_D = f$$

na skupu za koji nije trenirana. Stoga problem poopćenja neuronske mreže predstavlja problem minimizacije modela

$$\sum_{(x,y) \notin D} f_e(x, y, \tilde{f}(x))$$

Nakon što smo definirali dva osnovna problema kod neuronskih mreža, postavlja se pitanje o čemu zapravo ovisi uspješnost predviđanja neuronske mreže. Kroz primjenu neuronskih mreža uvidjelo se kako točnost predviđanja značajno ovisi o nelinearnosti implementiranog modela, ali to ipak ostavljamo za neku drugu priliku.

Zaključak

Popularnost neuronskih mreža i zapanjujući rezultati njihove primjene za rješavanje raznovrsnih problema ipak u sebi skrivaju zrno matematike. Naravno, napredne matematičke tehnike koje se primjenjuju u neuronskim mrežama zahtijevaju dugotrajnije izučavanje i veću posvećenost matematičkome radu, ali vjerujemo kako krajnji rezultat opravdava uloženi trud. Osim toga područje istraživanja matematičkih modela i računalnih algoritama koji se primjenjuju u neuronskim mrežama iznimno je dinamično te pruža puno prostora za znanstveni i stručni napredak. Osobito bismo istaknuli kako napredak u primjeni neuronskih mreža, između brojnih pitanja iz toga područja, uvodi tipična matematička pitanja:

- konvergencija tijekom faze učenja
- određivanje nužnog i dovoljnog skupa parametara za model
- odabir dimenzije međuslojeva
- probabilistički gradijentni algoritam učenja
- analiza kompleksnosti primijenjenih algoritama

Stoga još jednom napominjemo kako je upoznavanje s neuronskim mrežama iz matematičke perspektive istovremeno koristan i radostan zadatak.

Literatura

1. W. S. McCulloch, W. Pitts. A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, Vol. 5, 1943, pp. 115-133
2. F. Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, Vol. 65(6), 1958, pp. 386–408
3. F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Cornell Aeronautical Laboratory, Report No. VG-1196-6-8, 1961
4. M. Minsky, S. Papert. *Perceptrons: An Introduction to Computational Geometry*, MIT Press, 1969
5. D.E. Rumelhart, J. L. McClelland. *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press, 1986
6. J. A. Anderson, E. Rosenfeld. *Neurocomputing: Foundations of Research*, MIT Press, 1988
7. H. R. Lewis. *Ideas That Created the Future: Classic Papers of Computer Science*, MIT Press, 2021
8. A. Jain, J. Mao, K. M. Mohiuddin. Artificial Neural Networks: A Tutorial, *Computer*, Vol. 29(3), 1996, pp. 31-44
9. C. C. Aggarwal. *Neural Networks and Deep Learning*, Springer, 2018
10. K. Hornik, M. Stinchcombe, H. White. Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, Vol. 2, 1989, pp. 359-366