Matematika i neuronske mreže

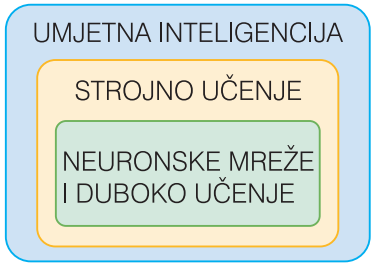
Ela Gračan i Nela Bosner, Zagreb

U ovom ćemo članku dati pregled osnovnih pojmova vezanih za umjetne neuronske mreže te opisati matematički model neurona. Naravno, zagrepst ćemo samo po površini ove teme i u pojednostavljenim crtama predstaviti vam jednu lijepu primjenu matematike u području računarstva.

Uvod

Pojam strojnog učenja i umjetne inteligencije često budi ljudsku maštu te svi odmah pomisle na robote koji imaju vlastitu volju ili na slične filmske interpretacije. A zapravo je riječ o algoritmima, o složenim programskim kôdovima i sofisticiranoj uporabi računala.

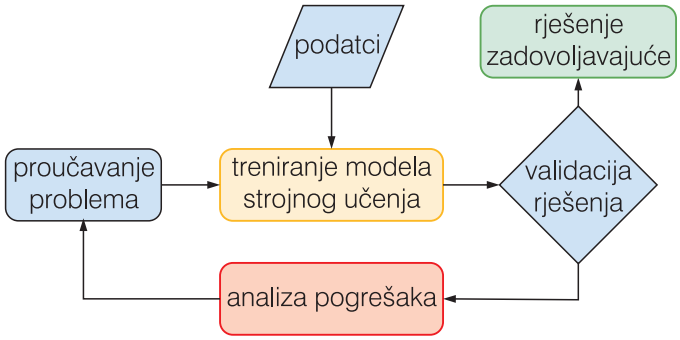
Umjetnom inteligencijom smatramo svaki nebiološki sustav koji ima sposobnost rješavanja dosad neviđenog problema [9]. **Strojno učenje** grana je umjetne inteligencije koja se bavi razvojem metoda i algoritama s pomoću kojih računalo (bez ljudske pomoći) nizom iteracija poboljšava obavljanje nekog zadatka. Područje strojnog učenja koje se oslanja na neuronske mreže naziva se **duboko učenje** te su ta dva pojma usko povezana i često se koriste naizmjenično (slika 1).



Slika 1. Neuronske mreže programski su kôd za strojno učenje

Uobičajeno je da se neki projekt u strojnom učenju odvija u sljedećim koracima (slika 2) [4]:

- analiza podataka
- izbor modela za učenje
- učenje na podatcima
- testiranje i validacija
- primjena modela na nepoznatim podatcima.



Slika 2. Pristup problemu strojnog učenja

Najvažniji je od gornjih koraka učenje na podatcima, tj. treniranje modela, te je to korak koji zahtijeva najviše vremena i računalne snage. U toj fazi učenja očekujemo samostalnu prilagodbu pa-

Ela Gračan, mag. inf. et math., Zagreb, elagracan@gmail.com
doc. dr. sc. Nela Bosner, Zavod za numeričku matematiku i znanstveno računanje, PMF – Matematički odsjek Sveučilišta u Zagrebu, nela.bosner@math.hr

rametara neuronske mreže s ciljem da mreža što točnije predvidi neku dosad neviđenu vrstu danog problema. No navedimo najprije osnovne pojmove.

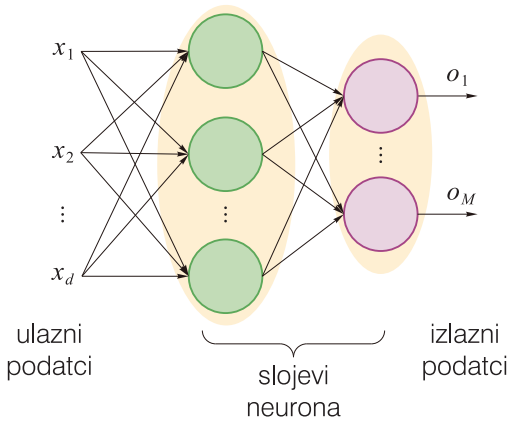
Osnovni pojmovi

Čovjekova je sposobnost paralelne obrade informacija i obavljanja raznih funkcija, kao što su primjerice prepoznavanje lica i razumijevanje prirodnog jezika, rezultat složene strukture neurona u ljudskom mozgu. Biološka neuronska mreža sastoji se od otprilike 10^{11} neurona međusobno povezanih sinapsama. Dio neuronske strukture je urođen, dok se ostatak razvija tijekom života. Te se promjene očituju u osnaživanju, odnosno slabljenju nekih sinaptičkih veza.

Upravo je biološka građa neurona bila inspiracija za konstrukciju umjetnih neuronskih mreža. Umjetne neuronske mreže apstrakcija su bioloških i puno su jednostavnije građe. Ostvarene su u najvećoj mjeri u obliku **programskog kôda** te nemaju sposobnosti ni približne ljudskom mozgu. Unatoč tomu, one u posljednje vrijeme nalaze primjenu u sve više djelatnosti.

Osnovni građevni element neuronske mreže, prema njezinom biološkom uzoru, nazivamo **neuron**. U literaturi se često koristi i naziv jedinica (engl. *unit*) [7]. Neurone spajaju veze koje imaju svoje **težine**, što poistovjećujemo sa sinapsama i njihovom snagom. Ako je ulazni signal neurona dovoljno jak, neuron će proslijediti signal sljedećim neuronima. Vrijednost koju neuron treba dostići da bi proslijedio signal nazivamo **prag** (engl. *bias*). Upravo su težine veza među neuronima i prag oni parametri mreže koji se prilagođavaju tijekom faze učenja. Iz ove osnovne ideje razvili su se razni modeli neurona koji se uglavnom razlikuju prema tome na koji način prosljeđuju signale.

Neurone grupiramo u **slojeve**, a svi slojevi čine neuronsku mrežu (slika 3). Ulazni se signali ponekad nazivaju **ulaznim slojem**, međutim on se ne sastoji od neurona. Osim ulaznog sloja mreža mora imati i **izlazni sloj**. Ako se mreža sastoji samo od ulaznog



Slika 3. Neuroni povezani u slojeve

sloja (ulaznih podataka) i izlaznog sloja neurona, reći ćemo da je to **jednoslojna neuronska mreža**. Kada mreža između ulaznog i izlaznog sloja ima jedan ili više slojeva neurona, govorimo o **višeslojnoj mreži**, a te slojeve nazivamo **skrivenim slojevima**.

Neuroni u istom sloju nisu međusobno povezani, već je svaki neuron sloja povezan s nekim neuronima u sljedećem sloju. Slika 3 prikazuje način povezivanja neurona u mrežu. Ovakva mreža naziva se **aciklička**, najjednostavnije je građe te je osnova za sve druge arhitekture neuronskih mreža.

Broj neurona u izlaznom sloju, kao i broj ulaza mreže, određen je prirodom problema. Primjerice, ako želimo ulazne parametre podijeliti u M klasa, tada će izlazni sloj takve mreže sadržavati M neurona. Broj neurona u skrivenim slojevima teže je odrediti te se on razlikuje od problema do problema.

Jedan se ulazni podatak može sastojati od više signala koji se nazivaju i **značajke** (engl. *feature*), dok ciljni podatak nazivamo **oznaka** (engl. *label*). Uobičajeno je da se ulazni podatci dijele na **skup za učenje**, **skup za validaciju** i **skup za testiranje** kako bi se osigurao ispravan rad mreže na dosad neviđenim podatcima. Ako ne osiguramo pravilnu podjelu ovih podataka, može doći do problema **pretreniranosti**. To je pojava u kojoj se mreža u potpunosti prilagodi podatcima za treniranje i gubi mogućnost prilagodbe novim vrstama problema

(engl. *overfitting*). Skup podataka za testiranje koristi se samo za evaluaciju mreže te se ti podatci ne smiju koristiti u algoritmima za učenje.

Pri konstrukciji neuronske mreže postoje razni **hiperparametri** koje možemo prilagođavati. Hiperparametri su vrijednosti koje se kroz samo učenje ne mijenjaju, ali utječu na efikasnost učenja mreže, poput primjerice broja slojeva. Usredotočimo li se na jedan hiperparametar, prirodan bi način bio da pokušamo trenirati mrežu nekoliko puta, svaki puta s različitom vrijednošću odabranog hiperparametra i odlučimo se za onaj koji daje najbolje rezultate na skupu za testiranje.

Međutim, kada neuronsku mrežu stavimo u uporabu, može se dogoditi da radi značajno lošije nego što je radila sa skupom podataka za testiranje. Razlog tome je što smo ponovljenim treniranjem i testiranjem mrežu implicitno prilagodili i skupu podataka za testiranje, pa mreža u stvarnosti nema sposobnost prilagođavanja novim podatcima. Rješenje ovog problema je dodatni skup podataka koji nazivamo skup za validaciju.

Nakon što faza učenja završi i kada mreža postiže zadovoljavajuće rezultate na skupu podataka za validaciju, tada na podatcima za testiranje možemo provjeriti je li spremna za stvarnu uporabu.

Primjena

Neuronske mreže koriste se u raznim područjima kojima je zajednička velika količina podataka i nepostojanje eksplicitnog algoritma za rješavanje problema nad tim podatcima. U posljednjih je nekoliko godina interes za umjetne neuronske mreže ponovno porastao, prvenstveno zbog razvoja računalne moći, što je dovelo do porasta kapaciteta neuronskih mreža i značajnog napretka u njihovu razvoju. Neuronske se mreže tako koriste u zrakoplovstvu, bankarstvu, medicini i robotici te mnogim drugim područjima u kojima pomažu ili samostalno obavljaju razne analize, procjene ili predviđanja.

Jedan od prvih sustava strojnog učenja koji je ušao u svakodnevnu upotrebu i svakodnevno ga koriste milijuni ljudi je filtar neželjene e-pošte (engl. *spam*), odnosno sustav koji na temelju sadržaja poruke elektroničke pošte istu razvrstava u neželjenu, odnosno željenu poštu (engl. *spam or ham*) [4]. Neki od najslikovitijih primjera primjene neuronskih mreža su obrada slike, prepoznavanje govora ili detekcija kartičnih prijevara. Naime, u doba internetske kupovine gdje kupac nije fizički prisutan, nemoguće je provjeriti svaku transakciju te je takav sustav podložan raznim ilegalnim manipulacijama. Neuronske su mreže u mogućnosti uočiti neke sličnosti u nelegalnim transakcijama te ih označiti ili spriječiti [6].

Nadalje, neuronske mreže ugrađene u sustave s optičkim čitačima koriste se za prepoznavanje slova, znamenaka i drugih znakova te se koriste za pretvaranje rukom pisanih, skeniranih ili tiskanih dokumenata u tekst u ASCII ili Unicode formatu [8]. U medicini se neuronske mreže mogu koristiti primjerice za detekciju ili potvrdu srčanog udara iz elektrokardiograma ili za otkrivanje demencije iz elektroencefalograma pacijenata. Koriste se i u obradi govora jer mogu prepoznati jezik kojim govorimo, pa čak i detektirati emociju kao što je ljutnja ili radost.

Matematički model neurona

Građa umjetnog neurona

Ulazni podatci neurona mogu biti vektori, matrice ili čak tenzori. Radi jednostavnosti pretpostavimo da su to vektori iz $\mathbf{R}^d$. Točnije, neka je $d \in \mathbf{N}$ dimenzija pojedinog ulaznog podatka neurona te $N \in \mathbf{N}$ ukupan broj ulaznih podataka. Označimo ulazne podatke s $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ pri čemu je

$$\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kd}], \qquad k = 1, 2, \ldots, N.$$

Sve ulazne podatke možemo promatrati kao matricu:

$$X = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1d} \\ x_{21} & x_{22} & \ldots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \ldots & x_{Nd} \end{bmatrix}.$$

Neka je sada $\mathbf{x} = [x_1, x_2, \ldots, x_d]$ jedan ulazni podatak, odnosno jedan redak matrice X koja se često naziva **matrica dizajna** [3]. Matematički model neurona ima d ulaznih signala te d odgovarajućih težina tih ulaznih signala $w_1, w_2, \ldots, w_d$. Ulaznim signalima neurona smatramo komponente vektora $\mathbf{x}$ koje smo nazvali značajke. Dakle, svakom od ulaznih signala neurona $x_i \in \mathbf{R}$ pridružena je težinska vrijednost $w_i \in \mathbf{R}$, $i = 1, 2, \ldots, d$. Neuron može imati i prag $b \in \mathbf{R}$ koji u jednostavnom modelu predstavlja kriterij za slanje signala. Osim ulaza, težina i praga, neuron ima ugrađenu i **funkciju prijelaza**, odnosno **aktivacijsku funkciju**. Kao ulazni parametar, funkcija prijelaza prima težinsku sumu z koja se često naziva **aktivacija neurona**.

$$z = \sum_{i=1}^{d} x_i w_i + b. \qquad (1)$$

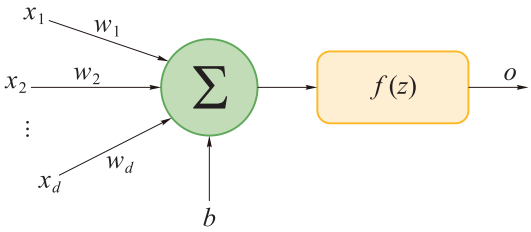
Isti izraz zapisan u matričnom obliku glasi

$$z = \mathbf{x}W + b,$$

pri čemu je

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}.$$

Neuron s ulazom $\mathbf{x}$, težinama W i funkcijom prijelaza f grafički je prikazan na slici 4.



Slika 4. Model neurona

Izlaz neurona je skalar o koji računamo kao

$$o = f(z) = f(\mathbf{x}W + b),$$

gdje je f neka određena funkcija prijelaza [5].

Funkcije prijelaza

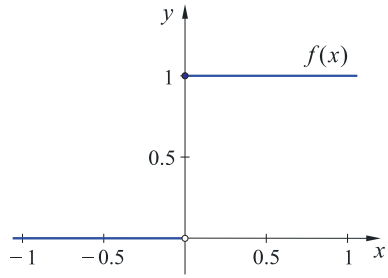
Najjednostavnija funkcija prijelaza je identiteta

$$f(x) = x,$$

odnosno izlaz neurona dan je s $o = \mathbf{x}W + b$. Model neurona koji koristi ovu funkciju naziva se ADALINE (engl. *ADAptive LINear Element*).

Sljedeća funkcija prijelaza je step-funkcija (engl. *hard limit transfer function*) čiji je izlaz 0 ili 1 (slika 5). Time dobivamo model neurona koji nazivamo **TLU** (engl. *Threshold Logic Unit*). Taj se model naziva i **perceptron** te se može koristiti za jednostavne zadatke klasifikacije gdje su podatci linearno odvojivi. Perceptron je jedan od najranije modeliranih neurona i iz njega su nastale razne druge poboljšane inačice [2].

$$f(x) = \begin{cases} 1 & \text{za } x \geq 0, \\ 0 & \text{inače.} \end{cases}$$



Slika 5. Step-funkcija

Funkcija koja ima sposobnost nelinearne klasifikacije je pozitivna linearna funkcija

$$f(x) = \max(0, x).$$

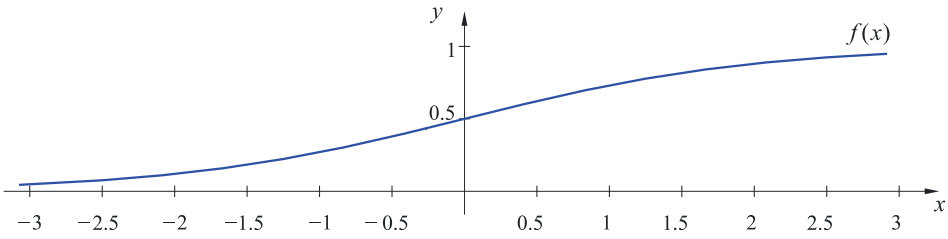
Neuron koji koristi ovu funkciju kao aktivacijsku funkciju naziva se **ReLU** (engl. *Rectified Linear Unit*). Često se koristi i njezina derivabilna inačica $f(x) = \ln(1 + e^x)$.

Nadalje, kao aktivacijska funkcija može se koristiti funkcija tangens hiperbolni:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

odnosno **logistička** funkcija (engl. *logistic sigmoid function*):

$$f(x) = \frac{1}{1 + e^{-cx}}.$$



Slika 6. Logistička funkcija

Njezina je značajka da za svaki $x \in \mathbf{R}$ vrijedi $f(x) \in [0,1]$ te je diferencijabilna pa je pogodna za optimizacijske metode koje koriste gradijent (slika 6).

Tablica 1 daje prikaz najčešćih funkcija prijelaza. Sve navedene funkcije u obzir uzimaju samo aktivaciju neurona na kojem djeluju.

Naziv funkcije	$f(x)$
identiteta	x
step	$\begin{cases} 1 & \text{za } x \geq 0 \\ 0 & \text{inače} \end{cases}$
pozitivna linearna	$\max(0, x)$
tangens hiperbolni	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$
logistička	$\frac{1}{1 + e^{-cx}}$

Tablica 1. Najčešće funkcije prijelaza

Međutim, najvažnija aktivacijska funkcija je funkcija *softmax*. Ona pri izračunu izlaza neurona uzima u obzir aktivaciju *svih* neurona istog sloja. Evo njezine matematičke definicije.

Neka je $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, gdje x_i predstavlja aktivaciju i-tog neurona u nekom sloju od n neurona. Tada je *softmax* funkcija dana s:

$$f(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}}.$$

Upravo je ova funkcija najčešće korištena u problemima klasifikacije jer bilo koji vektor pretvara u distribuciju vjerojatnosti. Zbog toga se najčešće koristi u izlaznom sloju neurona gdje izlazne vrijednosti neurona označavaju vjerojatnost pripadnosti nekoj klasi.

Postoje i razne druge funkcije koje se mogu koristiti kao prijelazne funkcije u modelu neurona. Sve one primjenjuju se na aktivaciju neurona z danog formulom (1).

Arhitektura neuronskih mreža

Neuronska mreža najjednostavnije arhitekture je **jednoslojna aciklička mreža** (engl. *one layer feedforward network*). Ona sadrži samo jedan sloj neurona, dakle ulazni signali povezani su sa slojem neurona koji je ujedno i izlazni sloj mreže. Nema povratnih veza između neurona, odnosno sve su veze usmjerene od ulaznih signala prema izlaznom sloju. Pritom neuroni istog sloja nisu međusobno povezani [2]. Najistaknutiji su primjeri ove arhitekture mreže ADALINE (funkcija prijelaza je identiteta) i Perceptron (funkcija prijelaza je step-funkcija).

Višeslojne (duboke) acikličke mreže promatramo kao niz slojeva neurona. Svaki je neuron povezan usmjerenom vezom s neuronima u sljedećem sloju. Kažemo da je mreža **potpuno povezana** kada je svaki od neurona u sloju povezan sa svakim neuronom u sljedećem sloju te takav sloj nazivamo **gustim slojem** (engl. *dense layer*). U suprotnom kažemo da je mreža **djelomično povezana** [7].

Treća skupina arhitektura su **mreže s povratnim vezama** (engl. *recurrent networks*). To su mreže u kojima postoji barem jedan **ciklus**, odnosno postoji barem jedan neuron koji je povezan s neuronom u prethodnom sloju. Neke od poznatih mreža ove arhitekture su Hammingova mreža koja ima jedan sloj neurona spojen acikličkim vezama i jedan sloj s povratnim vezama, te Hoppfieldova mreža koja se sastoji samo od sloja s povratnim vezama [5].

Promotrimo na jednostavnim primjerima razliku u mogućnostima između jednoslojne acikličke mreže (perceptrona) i višeslojnih acikličkih mreža.

Perceptron

Perceptronom nazivamo jednoslojnu mrežu neurona koji kao aktivacijsku funkciju koriste step-funkciju, dakle pojedini neuron kao svoj izlaz daje samo 0 ili 1 (TLU neuroni). Perceptron je jedna od najjednostavnijih neuronskih mreža koju je 1957. godine osmislio Frank Rosenblatt [5].

Za ilustraciju konstruirajmo jednostavni perceptron koji predstavlja logički sklop *ILI*. Ulazni podatci mogu biti $(0,0), (0,1), (1,0)$ i $(1,1)$, a izlaz $f(z)$ treba biti $0, 1, 1, 1$, nije teško zaključiti da matrica W treba biti $W = [1,1]^T$ te da je $b = 0$.

Ulazni podatak	Aktivacija	Izlaz
(0,0)	0	0
(0,1)	1	1
(1,0)	1	1
(1,1)	2	1

Tablica 2. Tablica vrijednosti aktivacija i izlaza za sklop *ILI*

Ako želimo dobiti sklop *I*, dovoljno je promijeniti b u $b = -1$ da bismo dobili ispravno rješenje.

Ulazni podatak	Aktivacija	Izlaz
(0,0)	-1	0
(0,1)	0	0
(1,0)	0	0
(1,1)	1	1

Tablica 3. Tablica vrijednosti aktivacija i izlaza za sklop *I*

Međutim, perceptronom ne možemo konstruirati sklop *ISKLJUČIVO ILI* (*XOR*). Već kod takvog jednostavnog zadatka jednoslojna mreža nailazi na probleme i potrebni su nam kompleksniji alati – treba nam višeslojna neuronska mreža.

Višeslojne neuronske mreže

Ove su mreže osnovni gradivni element za mnoge koncepte dubokog učenja. Evo ukratko i njihove matematičke pozadine.

Neka je S ukupan broj slojeva neurona u mreži. On se naziva **dubina modela**, odakle proizlazi i pojam dubokog učenja.

Promotrimo način računanja u l-tom sloju neurona, gdje je $l = 1, 2, \ldots, S$. Neka je $r_l \in \mathbf{N}$ broj neurona u tom sloju te neka je $r_{l-1} \in \mathbf{N}$ broj neurona iz prethodnog sloja (odnosno dimenzija d ulaznog podatka ako promatramo prvi sloj). Sa o^{l-1} označit ćemo izlaz prethodnog sloja neurona.

$$o^{l-1} = [o_1^{l-1}, o_2^{l-1}, \ldots, o_{r_{l-1}}^{l-1}].$$

U slučaju kada je $l = 1$, imamo

$$o^0 = \mathbf{x} = [x_1, x_2, \ldots, x_d].$$

Matrica težina l-tog sloja, $W_l \in M_{r_{l-1} r_l}(\mathbf{R})$ dana je sa

$$W_l = \begin{bmatrix} w_{11}^l & w_{12}^l & \cdots & w_{1r_l}^l \\ w_{21}^l & w_{22}^l & \cdots & w_{2r_l}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{r_{l-1}1}^l & w_{r_{l-1}2}^l & \cdots & w_{r_{l-1}r_l}^l \end{bmatrix}.$$

Vektor praga b^l dan je sa

$$b^l = [b_1^l, b_2^l, \ldots, b_{r_l}^l].$$

Sa z^l označavat ćemo vektor težinskih suma l-tog sloja (aktivaciju l-tog sloja):

$$z^l = [z_1^l, z_2^l, \ldots, z_{r_l}^l],$$

pri čemu je

$$z_j^l = \sum_{i=1}^{r_{l-1}} w_{ij}^l \, o_i^{l-1} + b_j^l.$$

Matrično, vektor b^l možemo dodati kao zadnji redak matrice W_l te izlaz prethodnog sloja proširiti jedinicom:

$$W_l' = \begin{bmatrix} W_l \\ b^l \end{bmatrix}, \qquad \tilde{o}^{l-1} = [\, o^{l-1} \quad 1 \,].$$

Tada je ekvivalentno

$$z^l = \tilde{o}^{l-1} W_l'.$$

Za zadanu aktivacijsku funkciju l-tog sloja f_l, izlaz sloja o^l dan je sa

$$\begin{aligned} o^l &= [o_1^l, o_2^l, \ldots, o_{r_l}^l] \\ &= [f_l(z_1^l), f_l(z_2^l), \ldots, f_l(z_{r_l}^l)]. \end{aligned}$$

Sada možemo konstruirati mrežu s jednim skrivenim slojem koja računa isključivo *ILI*. Matrica dizajna dana je sa

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Ciljni je izlaz y dan sa

$$y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Skriveni sloj imat će dva neurona $r_1 = 2$ te će koristiti pozitivnu linearnu aktivacijsku funkciju $f_1(x) = \max(0, x)$. Izlazni sloj imat će jedan neuron te će njegov izlaz biti jednak aktivaciji, odnosno aktivacijska funkcija bit će identiteta.

Neka je

$$W_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad b_1 = [\, 0 \quad -1 \,].$$

Aktivacija i izlaz prvog sloja $z_1 = XW_1 + b_1$ i $o_1 = f_1(z_1)$ dani su sa

$$z_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}, \qquad o_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}.$$

Neka je sada

$$W_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \qquad b_2 = 0.$$

Dobivamo izlaz $o = o_2 = z_2 = o_1 W_2 + b_2$

$$o = y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Ulazni podatak	Aktivacija prvog sloja	Izlaz prvog sloja	Aktivacija drugog sloja	Izlaz drugog sloja
(0,0)	(0, -1)	(0,0)	0	0
(0,1)	(1, 0)	(1,0)	1	1
(1,0)	(1, 0)	(1,0)	1	1
(1,1)	(2, 1)	(2,1)	0	0

Tablica 4. Tablica vrijednosti aktivacija i izlaza za sklop *isključivo ILI*

Želimo li s pomoću višeslojne neuronske mreže obraditi složenije ulazne podatke ili signale velike dimenzionalnosti, poput neke slike ili zvuka, potrebno je dodatno popraviti i doraditi njezine mogućnosti djelovanja. Nadogradnja višeslojnih neuronskih mreža su **konvolucijske neuronske mreže**.

Konvolucijske neuronske mreže

Višeslojnu neuronsku mrežu nazivamo **konvolucijskom** ako uz određen broj potpuno povezanih slojeva neurona sadrži barem jedan **konvolucijski sloj**. Obično se konvolucijski sloj koristi kako bismo iz ulaza, primjerice slike, dobili bolji prikaz skrivenih reprezentacija. Taj bolji prikaz naziva se **mapa značajki**. Pritom taj model neuronske mreže mora biti invarijantan na translacije objekata na slici i mora imati mogućnost uočavanja lokalnih povezanosti na slici.

Zamislimo da ulazni podatak više nije vektor, već crno-bijela slika, odnosno matrica $x \in M^{mn}(\mathbf{R})$. Umjesto matrice težina W, konvolucijski sloj koristi matricu $\omega \in M^{pr}(\mathbf{R})$ koju nazivamo **filtar** ili **jezgra** (engl. *kernel*).

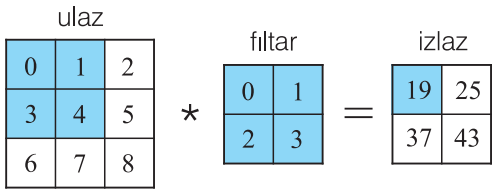
Dvodimenzionalna mapa značajki h računa se na sljedeći način:

$$h = \omega * x,$$

pri čemu $*$ označava operaciju konvolucije te vrijedi:

$$h(i,j) = (\omega * x)(i,j) = \sum_{k=0}^{p-1} \sum_{l=0}^{r-1} x(i+k, j+l)\omega(k,l). \tag{2}$$

Zapravo se koristi formula koja se naziva **unakrsna korelacija**, ali zbog sličnosti s formulom konvolucije* mreža nosi takav naziv [10]. Na slici 7 vidimo primjer računanja jednog elementa mape značajki.



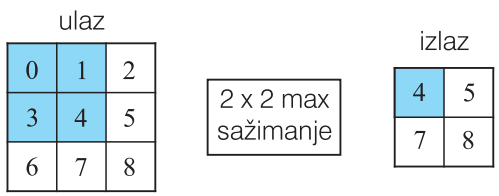
Slika 7. Primjer izračuna konvolucije [15]

Prednost konvolucijskog sloja u odnosu na potpuno povezan sloj očituje se u tome što su veze između neurona puno rjeđe zbog naglaska na lokalnu povezanost pa je samim time smanjen broj težina veza koje se tijekom učenja prilagođavaju. Ako je potrebno, na dobivenu se mapu značajki također može primijeniti aktivacijska funkcija.

Važan pojam povezan uz konvolucijski sloj je **pomak** (engl. *stride*). Kada primjenjujemo formulu (2) na matricu x, počinjemo od gornjeg lijevog elementa matrice x te se pomičemo desno i dolje dok ne prođemo sve dijelove matrice. Tako se u primjeru na slici 7 primjerice računa $0 \cdot 0 + 1 \cdot 1 + 3 \cdot 2 + 4 \cdot 3 = 19$. Pomak određuje za koliko se elemenata matrice želimo pomicati. Može biti različito za horizontalni i vertikalni smjer.

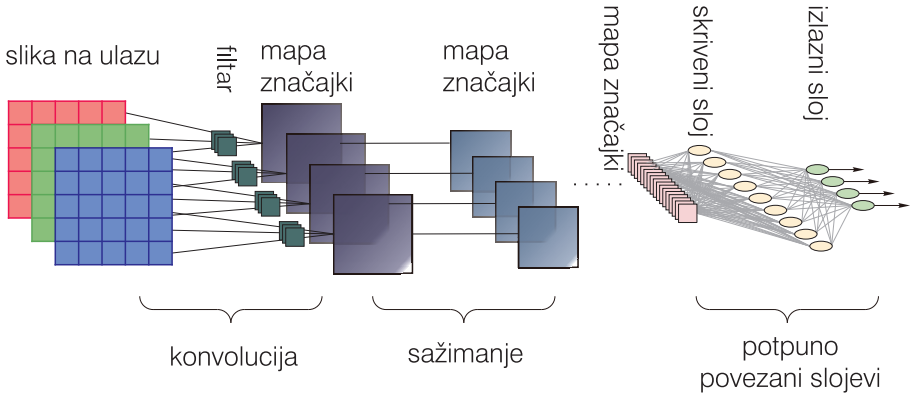
U višeslojnim se neuronskim mrežama uz konvolucijski sloj često nalazi još i **sloj sažimanja**. Taj sloj služi za smanjenje rezolucije mape značajki dobivene u konvolucijskom sloju, korištenjem **funkcije sažimanja**. Funkcija sažimanja djeluje na bliske vrijednosti mape značajki i pretvara ih u jednu vrijednost.

Kao i u konvolucijskom sloju, za sloj sažimanja također definiramo pomak. Uz to je potrebno definirati i **okvir sažimanja**, odnosno odrediti dimenziju podmatrice na koju se primjenjuje funkcija sažimanja. Na slici 8 vidimo primjer sažimanja maksimalnom vrijednošću s pomakom 1 te okvirom sažimanja 2×2.



Slika 8. Primjer pretvorbe mape značajki sažimanjem maksimalnom vrijednošću [15]

Slika 9 prikazuje uobičajeni način konstrukcije konvolucijske neuronske mreže. Nakon konvolucijskih



Slika 9. Uobičajena struktura konvolucijske neuronske mreže [16]

* Konvolucija je integral koji opisuje količinu preklapanja jedne funkcije s drugom dok se jedna pomiče preko druge (vidi `http://mathworld.wolfram.com/Convolution.html`).

slojeva dolaze skriveni potpuno povezani slojevi, a potom izlazni sloj.

Učenje mreže

Spomenimo još na kraju pravila i algoritme za prilagođavanje slojeva neurona. To se često naziva **treniranje**, odnosno **učenje mreže**. Razlikujemo tri načina učenja:

- **Učenje pod nadzorom**
  Algoritmu za učenje predaju se parovi (*ulaz*, *izlaz*) koji se nazivaju skup podataka za učenje. Učenje se odvija tako da se trenutačni izlaz mreže za dani ulaz uspoređuje s ciljanom vrijednošću izlaza te se na temelju toga parametri mreže prilagođavaju. Najčešći problemi koji se rješavaju učenjem pod nadzorom su predviđanje (regresija) i klasifikacija.
- **Učenje bez nadzora**
  Težine i pragovi prilagođavaju se samo s pomoću ulaznih podataka. Ova se vrsta učenja uglavnom koristi za grupiranje podataka u konačan broj grupa.
- **Učenje uz poticaj**
  Algoritam umjesto izlaznih podataka dobiva ocjenu za uspjeh te prilagođava težine i pragove s ciljem popravljanja dobivene ocjene.

Jedno od prvih pravila učenja naziva se **Hebbovo pravilo** koje je 1949. godine predstavio neuroznanstvenik Donald Hebb i predstavlja jedan od mogućih mehanizama za prilagodbu sinaptičkih veza između neurona koji se od tada koristi za prilagodbu parametara mreže. Hebbovo pravilo proizlazi iz Hebbova postulata koji glasi: "Kada akson jednog neurona neprestano podražuje drugi neuron, tada se među njima događa promjena tako da se učinkovitost prvog neurona povećava". Drugim riječima, veze među neuronima osnažuju se ako jedan neuron često podražava drugi ili obratno [5].

Neuronske mreže najveću primjenu nalaze u području učenja pod nadzorom. Učenje se tada svodi na optimizacijski problem funkcije koja mjeri točnost ili gubitak. Najpoznatiji algoritam za učenje naziva se **metoda gradijentnog spusta** te **algoritam unazadne propagacije** koji služi za izračun gradijenta u višeslojnim mrežama. Njima i brojnim drugim algoritmima i metodama bave se brojni znanstvenici, programeri, logičari i matematičari. A temelj svih ovih koncepata jest matematika – viša, kompliciranija, laicima nerazumljiva, ali uvijek vrijedna divljenja.

LITERATURA

1/ Ela Gračan (2020.): *Prepoznavanje uzoraka pomoću neuronskih mreža*, diplomski rad, PMF – Matematički odsjek Sveučilišta u Zagrebu, `https://urn.nsk.hr/urn:nbn:hr:217:771401`.

2/ B. Dalbelo Bašić, M. Čupić i J. Šnajder (2008.): *Umjetne neuronske mreže*, `https://www.fer.hr/_download/repository/UmjetneNeuronskeMreze.pdf`.

3/ I. Goodfellow, Y. Bengio i A. Courville (2016.): *Deep Learning*, MIT Press, `http://www.deeplearningbook.org`.

4/ A. Géron (2019.): *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, `https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/`.

5/ M. T. Hagan, H. B. Demuth, M. H. Beale i O. De Jesús (2015.): *Neural Network Design*, 2nd Edition, `http://hagan.okstate.edu/NNDesign.pdf`.

6/ R. Patidari, L. Sharma (2011.): *Credit Card Fraud Detection Using Neural Network*, `https://pdfs.semanticscholar.org/0419/c275f05841d87ab9a4c9767a4f997b61a50e.pdf`.

7/ Udacity i Google (2018.): *Intro to TensorFlow for Deep Learning*, `https://clašroom.udacity.com/courses/ud187`.

8/ Wikipedia (2013.), *Optical Character Recognition*, `https://en.wikipedia.org/wiki/Optical_character_recognition`.

9/ Wikipedia (2019.): *Umjetna inteligencija*, `https://hr.wikipedia.org/wiki/Umjetna_inteligencija`.

10/ Aston Zhang, Zachary C. Lipton, Mu Li i Alexander J. Smola (2020.): *Dive into Deep Lear-ning*, `https://d2l.ai`.

11/ S. Šegvić (2019.): *Duboko učenje, predavanja* @FER, `http://www.zemris.fer.hr/~šegvic/du/`.