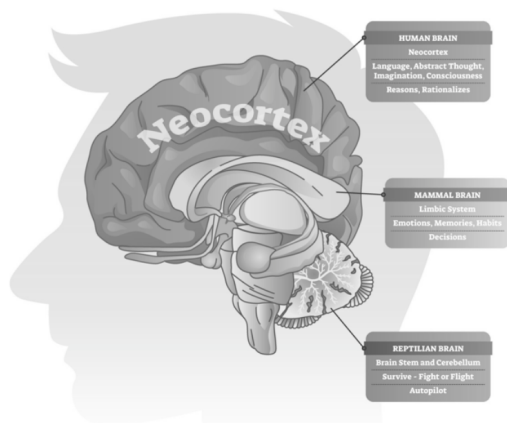


# Predavanje 4

## 3.1 Uvod u neuronske mreže

Motivacija za izučavanje vještačkih neuronskih mreža jeste automatiziranje poslova preko računara. Iako se danas mnoge aktivnosti mogu automatizirati, postoje i one za koje je to još uvijek nemoguće. Da bismo napravili sistem koji može samostalno donositi zaključke ili učiti, moramo prvo pronaći motivaciju u prirodi, odnosno, u **mentalnoj aktivnosti**. Mentalna aktivnost se sastoji od elektrohemijskih aktivnosti moždanih ćelija koje se nazivaju **neuroni**. Inspirisani ovom hipotezom, naučnici su pokušali da naprave sisteme koji rade na bazi **vještačke neuronske mreže** koje mogu na isti ili sličan način procesirati informacije i donositi zaključke. Sistem vještačke inteligencije koji uspješno oponaša rad mozga naziva se **inteligentnim sistem**. Pored ovog naziva, u praksi se javljaju i pojmovi konektivizam, paralelno distribuirano procesiranje i neuronsko računanje.

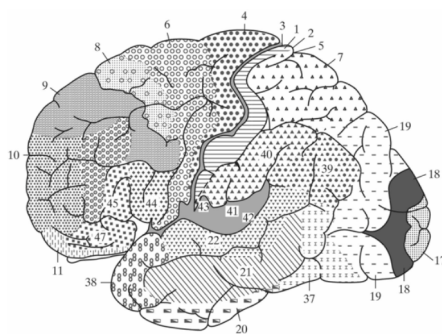
Ljudski mozak je veoma složen središnji dio živčanog sistema čovjeka. Kažemo da se sastoji od četiri osnovna dijela: veliki mozak, srednji mozak, mali mozak i kičmena moždina. Najsloženiji dio predstavlja **neokorteks** (veliki mozak). Neokorteks je dio mozga u kojem možemo da procesiramo jezik, da apstraktno razmišljamo i imamo svijest o nekim pojavama. Drugi dio mozga je **limbički sistem** (srednji mozak) pomoću kojeg spoznajemo emocije, donosimo odluke i čuvamo navike. Treći dio mozga je **moždano stablo** (mali mozak) koji nam omogućava da preživimo u sredini koju manje poznajemo (fight or flight - ili bježi ili se bori). Ovaj dio mozga je zadužen za donošenje odluka na samom licu mjesta.



Slika 3.1: Ljudski mozak

Mozak se sadrži od niza podsistema pomoću kojih čovjek obavlja razne aktivnosti. Prvi dio koji je bitan za spomenuti je **vizuelni korteks** koji je povezan sa sensorima (odnosno sa očima) pomoću kojeg dobijamo vizuelne informacije. Pored toga, postoji i **slušni korteks** (uši) koji služi za primanje zvuka. Zatim imamo **motorni korteks** i **predmotorni korteks** koji su zaduženi za sva kretanja koja obavljamo u svakodnevnim aktivnostima.

**Figure 4** Cytoarchitectural map of the cerebral cortex. The different areas are identified by the thickness of their layers and types of cells within them. Some of the key sensory areas are as follows: Motor cortex: motor strip, area 4; premotor area, area 6; frontal eye fields, area 8. Somatosensory cortex: areas 3, 1, and 2. Visual cortex: areas 17, 18, and 19. Auditory cortex: areas 41 and 42. (From A. Brodal, 1981, with permission of Oxford University Press.)



Slika 3.2: Podsistemi mozga

Znamo da je mozak sačinjen od velikog broja neurona koji većinski rade paralelno. Pored ove činjenice, poznato je i da postoji više od 100 vrsta različitih neurona. Svaka od ovih vrsta radi vrlo jednostavnu obradu podataka, a vrijeme koje se potroši na obradu je 2ms. Broj neurona u ljudskom mozgu je  $10^{11}$ , a svaki od njih u prosjeku dobija informacije od  $10^3$

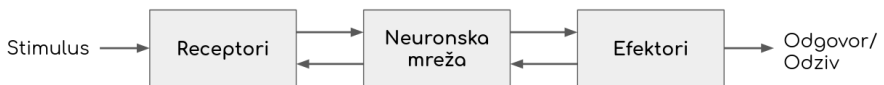
do  $10^4$  drugih neurona. Obrada samih podataka je tolerantna na greške. Ova činjenica govori da neće doći do pada čitavog sistema ukoliko se desi greška u procesu odlučivanja.

Inspirisani različitim istraživanjima na poljima biologije, psihologije i sličnih nauka, naučnici u polju računarskih tehnologija su došli do nove paradigme da se napravi model **vještačkih neuronskih mreža** (engl. *Artificial Neural Networks*). Model se bazira na funkcionalnostima ljudskog mozga. Samim time, javlja se i nova grana računarskih nauka koja se bavi izučavanjem ove nove paradigme koja se naziva **neuro-računarstvo**.

Postoje dva pristupa razvoju vještačke inteligencije. Prvi pristup razvoju predstavlja **simbolički pristup**. Znanje se predstavlja pomoću simbola, te se tim znanjem manipuliše pomoću algoritamskih pravila. Radi se o vrlo ranom pristupu koji nije ispunio početna očekivanja, iako je imao određene uspjehe u oblasti ekspertnih sistema. Razlog ovoga je što svako znanje ne možemo predstaviti formalnim pravilima jer ne znamo tačno na koji način ljudski mozak radi.

**Konektivistički pristup** je drugi pristup razvoja vještačke inteligencije. Zasniva se na izgradnji sistema arhitektura slične arhitekturi mozga koji uči **samostalno** na bazi iskustva.

Mozak je izuzetno složen, nelinearan i paralelan računar. Zadaci prepoznavanja ili percepcije se rutinski izvode za otprilike 100-200ms.

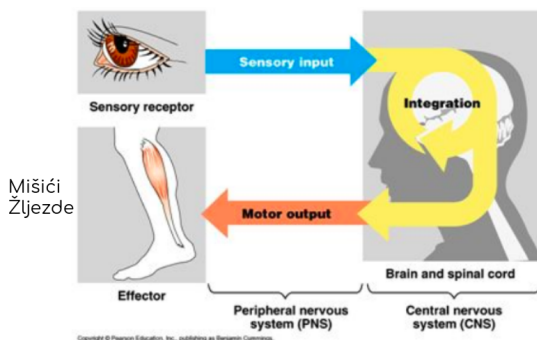


Slika 3.3: Opći model neuronske mreže

Na samom početku modela imamo stimulse koji dolaze iz okoline. Ove podatke primaju **receptori** koji ih dalje prosljeđuju **neuronskoj mreži**. Zatim, neuronska mreža donosi neke odluke i šalje ih **efektorima** koji mogu da daju određenu reakciju (generisanje razgovora, pokret ruke i slično). Te informacije se šalju u okolinu kao **odgovor**. Također, na osnovu određenih informacija iz okoline možemo analizirati efektore drugih ljudi te prosljeđivati te informacije neuronskoj mreži. Upravo navedeno predstavlja **povratnu spregu** koja se konstantno odvija.

Jedna osobina mozga koja se posebno izdvaja je **plastičnost mozga**. Ova osobina omogućava nervnom sistemu da se prilagodi svojoj okolini. Iako se smatra da ova osobina postaje slabija kroz odrastanje, istraživanja kažu da je moguće održavati plastičnost mozga konstantnim osvježavanjem znanja.

Na sljedećoj slici je vizuelno prikazano primanje podataka preko receptora. Receptori šalju određene informacije preko senzorskog ulaza do ljudskog mozga. Na ovom mjestu se informacije procesiraju, te se određene odluke šalju preko centralnog nervnog sistema do različitih motornih izlaza (mišići ili žlijezde).



Slika 3.4: Prikaz povratne sprege

Vještačka neuronska mreža je mašina koja je dizajnirana da modelira način na koji mozak izvršava određene zadatke ili funkcije od interesa. U istraživanju se fokusiramo na neuronske mreže koje izvode **korisne proračune** kroz **proces učenja**.

**Algoritam učenja** je postupak koji se koristi za izvođenje procesa učenja. Zanimljiva je činjenica je da biološka neuronska mreža *može modificirati vlastitu topologiju* koja je motivirana činjenicom da ljudske moždane ćelije prirodno mogu "umrijeti", te da se mogu formirati nove sinaptičke veze koje ne uključuju ove neurone. Računska snaga neuronske mreže predstavlja masivno distribuiranu i paralelnu strukturu koja ima **sposobnost učenja** (generaliziranja).

Proces učenja se sastoji od dvije faze rada:

- **faza učenja** (treniranja)
- **faza zaključivanja ili obrade podataka**

Učenje je iterativan pristup u kojem nekom sistemu predstavljamo skup ulaznih podataka (uzoraka, iskustva) i informaciju o očekivanom izlazu. Pri ovom procesu je potrebno da postepeno prilagođavamo koeficijente (veze) između neurona.

**Epoha** je jedna faza ili stadij u kojem se predstavljaju neki ulazni podaci.

Dakle, unutar jedne epohe tačno definišemo koliko uzoraka ćemo proslijediti sistemu na osnovu kojih će on dalje učiti. Razlikujemo tri vrste učenja:

1. **pojedinačno učenje** (engl. *online*) - učenje se dešava nakon svakog predstavljenog uzorka
2. **učenje s minigrupama** (engl. *mini-batches*) - učenje se dešava nakon više predloženih uzoraka
3. **grupno učenje** (engl. *batch*) - učenje se dešava tek nakon svih predloženih uzoraka

Proces učenja omogućava odlučivanje o nekim budućim događajima na osnovu posmatranja određene okoline ili pravljenja opservacija o svijetu. Učenje može biti trivijalan proces (kao što je biranje broja na telefonu) ili može biti složen proces (nove teorije o univerzumu).

Mogućnost učenja je bitna iz mnogo razloga. Jedan od njih je činjenica da je nemoguće predvidjeti sve moguće scenarije u kojima će se program (agent) naći. Pored toga, nemoguće je i predvidjeti sve promjene u vremenu koje će se desiti. Kao primjer možemo navesti program koji treba da predviđa stanje na tržištu dionica. Ovakav program mora imati mogućnost da se prilagodi na nove promjene onda kada one nastanu. Iako je moguće napraviti sistem koji nema mogućnost ažuriranja na neku novu okolinu, on će se većinu vremena smatrati nepotrebnim jer ima ograničenu upotrebu. Dalje, mogućnost učenja je bitna jer je neka znanja nemoguće isprogramirati. Primjer ovakvog problema jeste pisanje programa za prepoznavanje lica. Pravljenje ovakvog algoritma nije bilo uspješno jer su programi imali tačnost od oko 60%. Međutim, dolaskom dubokih neuronskih mreža, posao pisanja ovakvih algoritama se znatno olakšao.

**Induktivno učenje** je sposobnost pronalaženja generalne funkcije ili pravila na osnovu tačno određenih ulazno/izlaznih parova podataka.

Postoji nekoliko različitih načina učenja u odnosu na **povratnu spregu** (engl. *feedback*) u procesu učenja. Dakle, ukoliko nekom sistemu prosljedimo neku informaciju, on će donijeti određene zaključke. Ako sistem ima mogućnost da ima povratnu spregu o tim zaključcima, on tu informaciju može koristiti kako bi ažurirao svoje znanje.

U odnosu na povratnu spregu koju možemo da dobijemo, proces učenja možemo podijeliti na tri načina učenja:

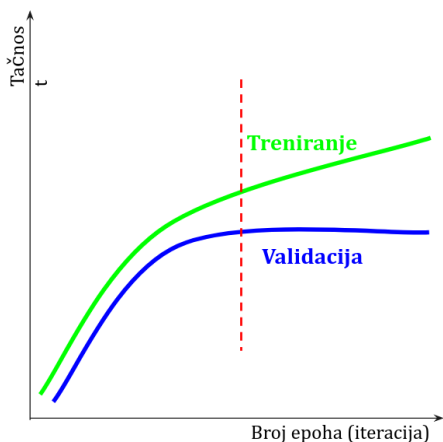
1. **nadzirano** (engl. *supervised*) - u procesu učenja postoje ulazno/izlazni parovi koji pomažu programu/agentu da uči iz podataka
2. **nenadzirano** (engl. *unsupervised*) - ne postoji povratna sprega u procesu učenja (iz podataka se izvlači funkcija koja odgovara samom zadatku)

### 3. **podržano** (engl. *reinforcement*) - učenje na bazi nagrada i kazni

Proces učenja se uvijek izvršava nad nekim **skupom podataka** (engl. *dataset*). Tok procesa se stalno prati kroz **provjeru tačnosti** (engl. *learning accuracy*) nad skupom podataka za treniranje, zatim nad skupom za validaciju i skupom za testiranje. Ako se krećemo u dobrom pravcu (tačnost se povećava), to znači da su arhitektura i hiperparametri dobro određeni.

Model nad kojim vršimo proces učenja mora da ima sposobnost **generalnog zaključivanja** nad skupom za testiranje, odnosno, nad podacima koje mreža nikad nije vidjela. Negativna pojava koja se javlja u procesu učenja je **pretreniranost** (engl. *overfitting*) nad podacima za treniranje. Dešava se da model previše dobro procjenjuje zadatke u procesu treniranja, te iz ovog razloga on nema dobre rezultate nad podacima za testiranje. Dakle, potrebno je pronaći određeni balans između ova dva procesa.

Na grafu je prikazan odnos broja epoha (iteracija) u odnosu na tačnost. Obično se kaže da je broj iteracija povezan sa tačnošću (što je veći broj iteracija, veća je i tačnost). Međutim, kako ne bi došlo do pretreniranosti, proces treniranja je potrebno pratiti i zaustaviti u trenutku kada tačnost počinje stagnirati ili opadati na skupu za validaciju.



Slika 3.5: Odnos broja epoha i tačnosti

Pošto su neuronske mreže pretežno vezane za nadzirano učenje, u nastavku će se opisati zadatak ovog načina učenja.

Neka je poznat skup podataka za treniranje:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

Svaki  $x_i$  predstavlja ulazni podatak koji može biti slika, niz brojeva i slično. Za ove podatke imamo labelu  $y_i$  koja je generisana nama nepoznatom

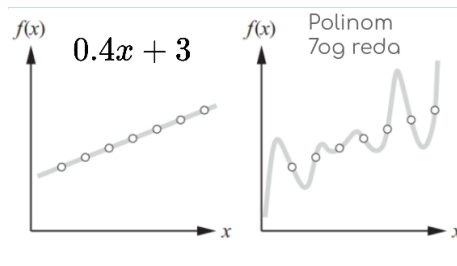
funkcijom  $y = f(x)$ . Zadatak je naći funkciju  $h$  koja vrši aproksimaciju nepoznate funkcije  $y$ . Funkcija  $h$  se naziva **hipoteza** i traži se u prostoru hipoteza.

Dakle, proces učenja je pretraživanje prostora svih hipoteza  $H$  i pronalaženje one hipoteze koja će imati zadovoljavajuće rezultate i nad podacima izvan skupa za treniranje (**testni skup podataka**). Kažemo da hipoteza ima sposobnost generaliziranja ukoliko ispravno izvrši predikciju vrijednosti  $y$  nad testnim skupom podataka.

Kao što je već rečeno, zadatak klasifikacije je da se definiše podjela postojećih podataka na određene klase. Drugi zadatak koji se javlja jeste **regresija** pri čemu je izlaz iz ovog algoritma broj (naprimjer ukoliko provjeravamo temperaturu određenog dana).

Ukoliko imamo neku funkciju hipoteze  $h$  koja može konzistentno da opiše sve podatke nekog skupa, onda se ona naziva **konzistentnom hipotezom**. Problem s kojim se susrećemo jeste kako odabrati najbolju hipotezu. Sami izbor ovisi o testnim podacima, odnosno, ukoliko oni leže na funkciji hipoteze, onda se bira ta hipoteza i kaže se da je to idealan slučaj.

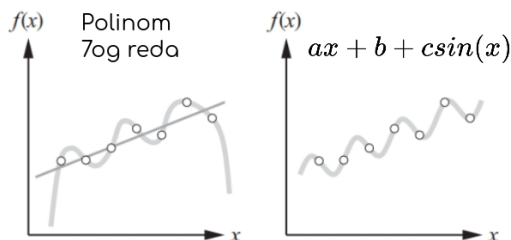
Sljedećim grafom je prikazan niz tačaka i date su funkcije koje ih opisuju. Možemo zaključiti da i jedna i druga funkcija jako dobro opisuju date podatke, te se obe mogu koristiti kao klasifikatori podataka. Međutim, jedna je kompleksnija (polinom sedmog reda) a druga jednostavnija (linearna funkcija). Iz skupa odabranih hipoteza, potrebno je izvršiti ispravan izbor.



Slika 3.6: Primjer traženja hipoteza

Jedno pravilo koje se bavi ovim problemom je pravilo **Ockhamove oštrice** (engl. *Ockham's razor*). Ime je dobilo po engleskom filozofu **William of Ockham** iz XIV vijeka. Ovo filozofsko pravilo pokušava da dâ odgovor na pitanje kako odabrati između dvije ili više konzistentnih hipoteza. Jedno od mogućih rješenja je da biramo najjednostavnije pravilo/funkciju u odnosu na konzistentnost sa podacima.

Drugi primjer sličnog problema je predstavljen sa sljedećim skupom podataka. Na samom grafu su prikazane i funkcije koje opisuju zadane podatke. Predstavljene su dvije hipoteze, pri čemu je jedna funkcija polinom sedmog reda, dok je druga funkcija modifikovana sinusoida sabrana sa linearnom funkcijom.



Slika 3.7: Primjer traženja hipoteza

Postavlja se pitanje koja je funkcija bolja. Potrebno je napraviti *kompromis* između složenosti funkcije i mogućnosti generalizacije. Suštinski problem je što ne znamo pravu funkciju koja bi mogla da opiše dati skup podataka. Samim time, teško je reći da li je proces učenja moguće realizirati. Obzirom na to da izbor hipoteze ovisi i o brzini izračunljivosti funkcije  $h$ , dolazimo do zaključka da bismo biramo jednostavnije funkcije kao što to opisuje i Ockhamovo pravilo.

Polinomi većeg reda mogu značajno bolje opisati skup podataka za treniranje. Međutim, bitnije je naći funkciju koja će pokazati zadovoljavajuće rezultate na validacijskom i testnom skupu podataka. **Izbor modela** je proces kojim se bira red polinoma (odnosno **prostor hipoteze**). Zatim se procesom **optimizacije** pronalazi funkcija najbolje hipoteze unutar tog prostora. Da bismo izvršili optimizaciju, moramo definisati i **funkciju gubitka  $L$**  (engl. *loss*).

Recimo da imamo neku funkciju  $f(x) = y$  koja će nam dati pravu labelu. Prvo je potrebno naći hipotezu kojoj šaljemo iste ulazne podatke, a na izlazu dobijamo procjenu funkcije  $y$ . Recimo da je tražena vrijednost 137.035999, a vrijednost koju smo dobili pomoću hipoteze iznosi 137.036. Da bismo procijenili da li je ova funkcija dobra ili ne, definišemo **funkciju gubitka  $L(x, y, \hat{y})$** .

Imamo nekoliko načina izračunavanja vrijednosti gubitka. Prvi način je **apsolutna vrijednost gubitka ( $L_1$ )** data funkcijom

$$L_1(y, \hat{y}) = |y - \hat{y}|$$

dok je drugi način putem **kvadratne vrijednosti greške** definiranom



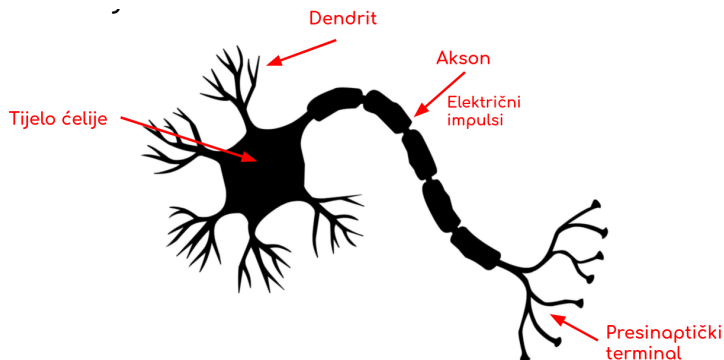
funkcijom

$$L_2(y, \hat{y}) = (y - \hat{y})^2$$

Ovu informaciju o gubitku je moguće dalje propagirati kroz mrežu putem već spomenute backpropagacije.

## 3.2 Biološki i vještački neuron

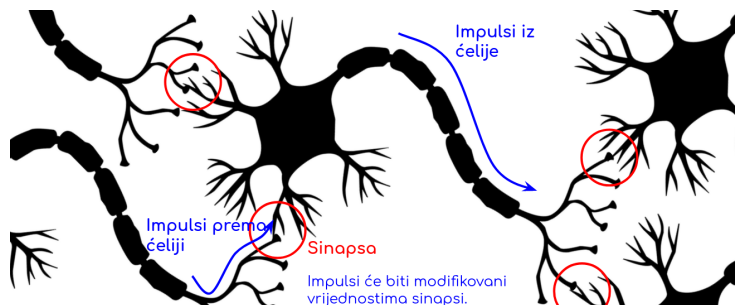
U nastavku će se uvesti osnovne informacije o biološkom i računarskom neuronu. Biološki neuron je ilustrativno predstavljen sljedećim modelom:



Slika 3.8: Biološki neuron

Funkcionalnost ovakvog biološkog neurona je bazirana na različitim istraživanjima. Komponenta svakog neurona je **tijelo ćelije**. Na tijelo ćelije je povezan **akson** kroz koji prolaze elektrohemijski impulsi. Tijelo ćelije prihvata informacije od drugih neurona preko **dendrita**, dok se preko **aksona** šalju informacije ka drugim neuronima. Shodno tome, impulsi se mogu slati prema ćeliji i izvan ćelije. Na krajevima aksona su definisani **presinaptički terminali** koji služe za spajanje sa drugim neuronima. Kada se spoji terminal sa dendritom drugog neurona, dobijamo pojavu **sinapse**.

Jedan neuron nema neku značajnu funkciju, već je potrebno imati određenu arhitekturu neurona.

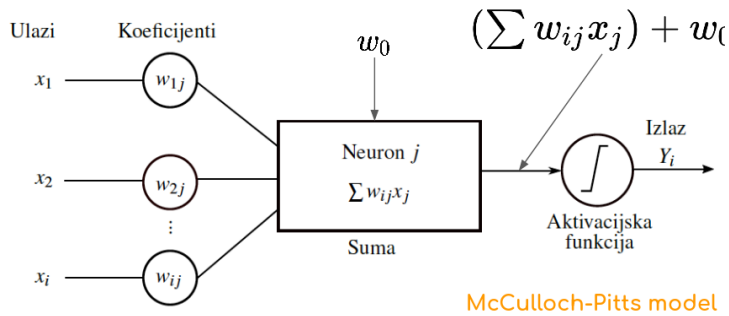


Slika 3.9: Mreža bioloških neurona

Obzirom na to, u realnosti većinom posmatramo niz neurona. Konekcija između dendrita i presinaptičkih terminala omogućava prihvatanje informacija

od drugih neurona. Navedeno ne mora značiti da će doći do slanja impulsa drugom neuronu svaki put kada se ostvari konekcija između njih. Ovom činjenicom se javlja pojava nelinearnog ponašanja čime se definiše **stopa aktiviranja neurona**. Iz ovog razloga je potrebno preći određeni prag da bismo poslali informaciju iz jednog neurona u drugi neuron.

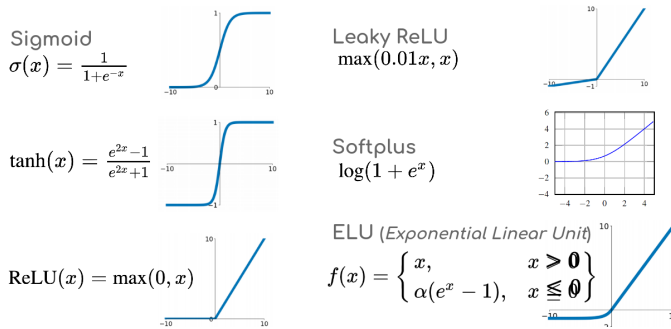
Prvi model biološkog neurona je nastao 1943. godine i naziva se **Threshold Logic Unit** ili **TLU perceptron**. Predstavljen je od strane Warren McCulloch-a i Walter Pitts-a i ujedno predstavlja prvi model biološkog neurona.



Slika 3.10: McCulloch-Pitts-ov model neurona

Jedinica koja definiše prag (engl. *Threshold*) zapravo predstavlja aktivacijsku funkciju. McCulloch-Pitts-ov model se sastoji od više elemenata. Na samom početku se nalaze ulazi koje čine neki drugi neuroni ( $x_i$ ). Veze između ovih ulaznih neurona i biološkog neurona  $j$  su omogućene preko koeficijenata koji se ažuriraju u samom procesu učenja. Zatim, imamo tijelo neurona koje se implementira kao suma skaliranih vrijednosti ulaza. Na ovu sumu se dodaje određena vrsta otklona (engl. *bias*)  $w_0$  koja predstavlja neku predefinisanu vrijednost i može biti pozitivna ili negativna vrijednost (vrlo često ima vrijednost 1). Vrijednost na izlazu iz tijela neurona predstavlja **funkciju sume** koja se dalje proslijedi na procesiranje **aktivacijskoj funkciji** (engl. *threshold logic unit*). Aktivacijska funkcija donosi odluku da li šaljemo informaciju do narednog neurona. Ukoliko nećemo vršiti slanje određene informacije, izlaz će biti nula, a u suprotnom slučaju je jedan. U realnosti postoje kompleksnije aktivacijske funkcije koje daju više izlaza. Aktivacijska funkcija je nelinearna funkcija i unosi nelinearnost u računarski model. Dakle, model će generisati određeni izlaz koji prelazi prag definisan u odnosu na odabranu aktivacijsku funkciju.

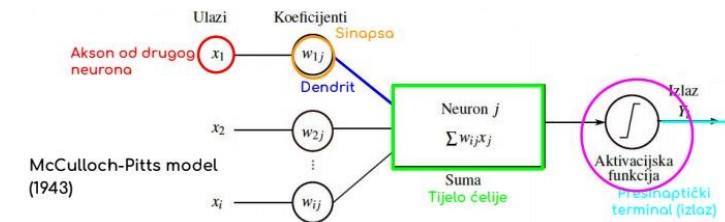
Neke aktivacijske funkcije koje se mogu iskoristiti su date u nastavku:



Slika 3.11: Primjer aktivacijskih funkcija

Izbor funkcije zavisi od vrste problema koji se rješava. Najčešće se koristi funkcija  $\text{ReLU}(x)$  jer ona vrši najjednostavniju operaciju. Unutar ove funkcije se traži maksimalna vrijednost između nule i izlaza koji dobijemo iz tijela neurona. Ukoliko je pozitivna vrijednost, izvršiti će se proslijeđivanje informacija, a ukoliko je rezultat nula, neće se ništa desiti. Pored jednostavnosti, funkcija daje i zadovoljavajuće rezultate.

Preko računarskog modela biološkog neurona je moguće predstaviti biološki neuron sa svim spomenutim konceptima. Ulaz u model će predstavljati aksone od drugog neurona. Zatim, sinapsu možemo predstaviti putem koeficijenata, dok će veza između koeficijenata i tijela ćelije biti prikazana preko dendrita. Tijelo ćelije se predstavlja preko već opisane sume, a rezultat se šalje na presinaptički terminal.



Slika 3.12: Model biološkog preko računarskog modela neurona

U narednoj tabeli, možemo vidjeti kako se neki dijelovi biološkog neurona mapiraju na vještačke neurone. **Soma** ili **ćelija** se obično mapira na čvor (engl. *Node*) i predstavlja osnovnu ćeliju neurona. **Dendriti** koji se koriste za slanje informacija drugim neuronima su zapravo ulazi. Zatim, aksoni predstavljaju izlaze iz neurona. Sinapse predstavljaju konekcije između određenih elemenata i određene su koeficijentima. Koeficijent se određuje pomoću snage i smjera, te se na osnovu toga utvrđuje da li se povećava ili smanjuje funkcija veze. Dalje, posmatrajući jedinicu biološkog neurona,

možemo reći da su oni dosta spori dok su s druge strane vještački neuroni jako brzi. Međutim, nemoguće je imati onoliki broj neurona koliko ih postoji u ljudskom mozgu jer je treniranje takve mreže jako kompleksno.

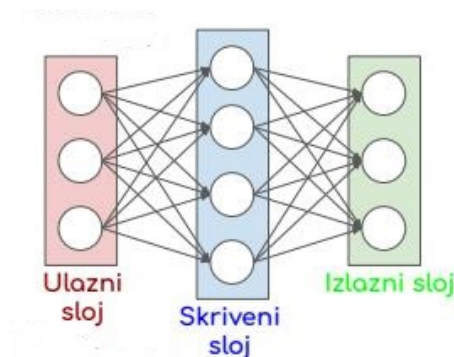
Bioški	Vještački
Soma	Čvor
Dendridi	Ulazi
Akson	Izlaz
Sinapsa	Koeficijent
Sporo	Brzo
Veliki broj neurona ( $10^9$ )	Nekoliko neurona (desetina ili stotine hiljada)

Slika 3.13: Tabela poređenja neurona

Kao što je već rečeno, da bismo mogli izvršiti neku korisnu klasifikaciju, potrebno je više neurona. Ti neuroni se obično organizuju u tri vrste slojeva: ulazni, skriveni i izlazni. Ukoliko se radi o kompleksnijim modelima, postojati će više skrivenih slojeva i u tom slučaju se radi o **dubokoj mreži**.

Ulazni sloj sadrži ulazne podatke. Svaki podatak se može prikazati preko određenih atributa. Stoga, ulazni sloj će imati onoliko podataka koliko on ima atributa. Iako se ulazi predstavljaju kao neuroni, oni nemaju mogućnost računanja dok neuroni u skrivenom i izlaznom sloju posjeduju ovu mogućnost.

U sljedećem primjeru, svaki ulaz je povezan sa svakim neuronom u skrivenom sloju, te je svaki neuron iz skrivenog sloja povezan sa izlaznim slojem. Ovaj model predstavlja **unaprijed formiranu neuronsku mrežu sa više slojeva**, te nema povratnu spregu.



Slika 3.14: Slojevi neuronskih mreža

Uzmimo kao primjer binarnu klasifikaciju. U tom slučaju, izlazni sloj ima samo jedan neuron koji može biti ili nula ili jedan. Dakle, za potrebe klasifikacije ovog modela, jedan sloj je sasvim dovoljan. Povećavanjem broja klasa povećava se i broj izlaza.

Sličnosti između bioloških i vještačkih neurona su simbolične, te računarski neuron ne predstavlja stvarnu reprezentaciju biološkog neurona. Dakle, u realnosti ne možemo tačno replicirati biološki neuron u računarskim sistemima iz više razloga. Biološki neuroni imaju izuzetno **kompleksnu topologiju** (vezu između neurona). Dendriti u biološkim neuronima mogu izvršavati složene nelinearne proračune, što je teško precizno opisati u računarskom sistemu. Pored toga, pojava sinapse predstavlja **složeni nelinearni dinamički sistem** (nisu pojedinačne težine). S druge strane, računarski neuroni su organizovani u **slojeve regularnog formata** i primarno koriste operacije sabiranja i množenja. Ovi slojevi predstavljaju apstraktne vještačke tvorevine koje trebaju da preslikaju stvarnu strukturu bioloških neurona.

Nakon objave McCulloch-Pitts-ovog modela biološkog neurona, 1949. godine je definisano i **Hebbovo pravilo učenja**. Ovo pravilo je uveo **Donald O. Hebb** i ono glasi:

"Kad je akson neurona A dovoljno blizu da aktivira neuron B i to ponavlja veći broj puta, dolazi do metaboličkih promjena tako da se povećava efikasnost neurona A u aktiviranju neurona B."

Nakon njega, psiholog **Gunther S. Stent** je proširio Hebbovo pravilo 1973. godine:

"Ako jedan neuron ne utiče na drugog, onda sinapsa među njima postaje slabija ili se potpuno eliminiše."

Dakle, pravilo govori da što su dva neurona bliže jedan drugom da imamo veću mogućnost promjene koeficijenata između ta dva neurona. Odnosno, što više ponavljamo neku materiju, neuroni koji definišu znanje se češće mijenjaju. Samim time, ukoliko imamo dva neurona koji ne utiču jedan na drugog, sinapsa će biti slabija.

Rosenblatt 1958. godine uzima ova dva pravila i McCulloch-Pitts-ov model, te definiše **pravilo učenja perceptrona** koje glasi:

1. Ciklično se prolazi kroz svih  $N$  uzoraka za učenje, jedan po jedan.
2. Izvrši se klasifikacija trenutnog uzorka.

(a) Ako je klasifikacija ispravna, ne mijenjaj koeficijente

- i. ako je  $N$ -ti uzastopni uzorak klasificiran ispravno, prekini proces učenja

- ii. inače pređi na sljedeći uzorak
- (b) Ako je klasifikacija neispravna, izvrši korekciju vrijednosti koeficijenta perceptrona prema sljedećoj formuli

$$w_i(k+1) = w_i(k) + \eta(y - \hat{y})x_i$$

gdje je  $w_i$  vrijednost trenutnog koeficijenta,  $y - \hat{y}$  razlika između stvarne i predviđene labele

Parametar  $\eta$  se naziva **stopa učenja** i može da poprimi različite vrijednosti. Naprimjer, može biti pozitivan veoma mali broj koji određuje koliko treba ažurirati trenutnu vrijednost koeficijenta. Ako je broj premali, postupak učenja će napredovati veoma sporo. S druge strane, ako je preveliki, postupak može divergirati u neželjenom smjeru.