<https://missinglink.ai/guides/neural-network-concepts/complete-guide-artificial-neural-networks/>

<https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-tutorial-basic-advanced/>

<https://medium.com/dataseries/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17>

<https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5>

https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

# Deeplizard

# AKTIVACIJSKE FUNKCIJE U NEURALNOJ MREŽI

U ovom poglavlju će se opisati što je to aktivacijska funkcija i zašto se ona koristi u umjetnim neuralnim mrežama. Nakon toga će se opisati i prikazati najčešće korištene aktivacijske funkcije te koji su njihovi međusobni odnosi.

## ZAŠTO SE KORISTE AKTIVACIJSKE FUNKCIJE

Aktivnost u biološkoj neuralnoj mreži, gdje su različiti neuroni aktivirani različitim podražajima, bila je inspiracija za korištenje aktivacijskih funkcija u umjetnim neuralnim mrežama. U biološkoj neuralnoj mreži, neki neuroni su ili aktivirani ili ne. U aktivacijskoj funkciji u umjetnoj neuralnoj mreži, vrijednost neurona može biti bilo koji broj između nula (ili minus jedan) i jedan. Što je vrijednost neurona bliža nuli, to je neuron manje aktiviran. S druge strane, što je vrijednost neurona bliža jedinici, to je neuron više aktiviran.

Aktivacijska funkcija u umjetnoj neuralnoj mreži jest funkcija koja preslikava ulaznu vrijednost, ponderiranu sumu svih ulaznih veza određenog čvora, u odgovarajuću izlaznu vrijednost.

Aktivacijska funkcija na neki određeni način transformira ulaznu vrijednost u broj koji se nalazi između unaprijed određene donje i gornje granice. Obično umjetne neuralne mreže koriste ne-linearne aktivacijske funkcije jer su vrste preslikavanja kod dubokih neuralnih mreža složenije od običnih linearnih preslikavanja i zbog toga što ne-linearne aktivacijske funkcije omogućuju računanje proizvoljno složenih funkcija. Još jedna prednost aktivacijskih funkcija jest ta što aktivacijske funkcije dopuštaju povratnu propagaciju koja je objašnjena u drugom poglavlju.

Obično su aktivacijske funkcije, koje se koriste u skrivenim slojevima, iste za sve skrivene slojeve u određenoj neuralnoj mreži. Nije uobičajeno vidjeti ReLU aktivacijsku funkciju u jednom skrivenom sloju i sigmoidnu aktivacijsku funkciju u drugom skrivenom sloju. Koristi se ili samo ReLU aktivacijska funkcija ili samo sigmoidna aktivacijska funkcija u svim skrivenim slojevima određene mreže.

Kako bi se razumjelo zašto se koriste ne-linearne aktivacijske funkcije, i kako bi se poslije dokazalo da je jedna od aktivacijskih funkcija (ReLU) ne-linearna, prvo je potrebno razumjeti linearne funkcije.

Neka je *f* funkcija na skupu *X*.

Neka su *a* i  *b* elementi skupa *X*.

Neka je *x* neki realan broj.

Funkcija *f* je linearna funkcija ako i samo ako vrijedi:

i ako vrijedi:

.

Važna značajka linearnih funkcija jest da je kompozicija dviju linearnih funkcija također linearna funkcija. To znači da će čak i u dubokim umjetnim neuralnim mrežama, ako se koriste samo linearne funkcije na ulaznim vrijednostima kod prosljeđivanja prema naprijed (eng. *forward propagation*), preslikavanje ulaza na izlaz također biti linearno. Kada bi svaki sloj u neuralnoj mreži koristio samo pristranosti (eng. *bias*) i težine (eng. *weights*), bez aktivacijskih funkcija, cijela mreža bi bila jednaka jednoj linearnoj kombinaciji težina i pristranosti. Drugim riječima, formula neuralne mreže bi se mogla faktorizirati i pojednostavniti na jednostavan linearni regresijski model. Takav model bi mogao rješavati jednostavne linearne ovisnosti, ali ne bi mogao obavljati prave zadatke neuralne mreže kao što su obrada slika i zvukova.

## SIGMOIDNA AKTIVACIJSKA FUNKCIJA

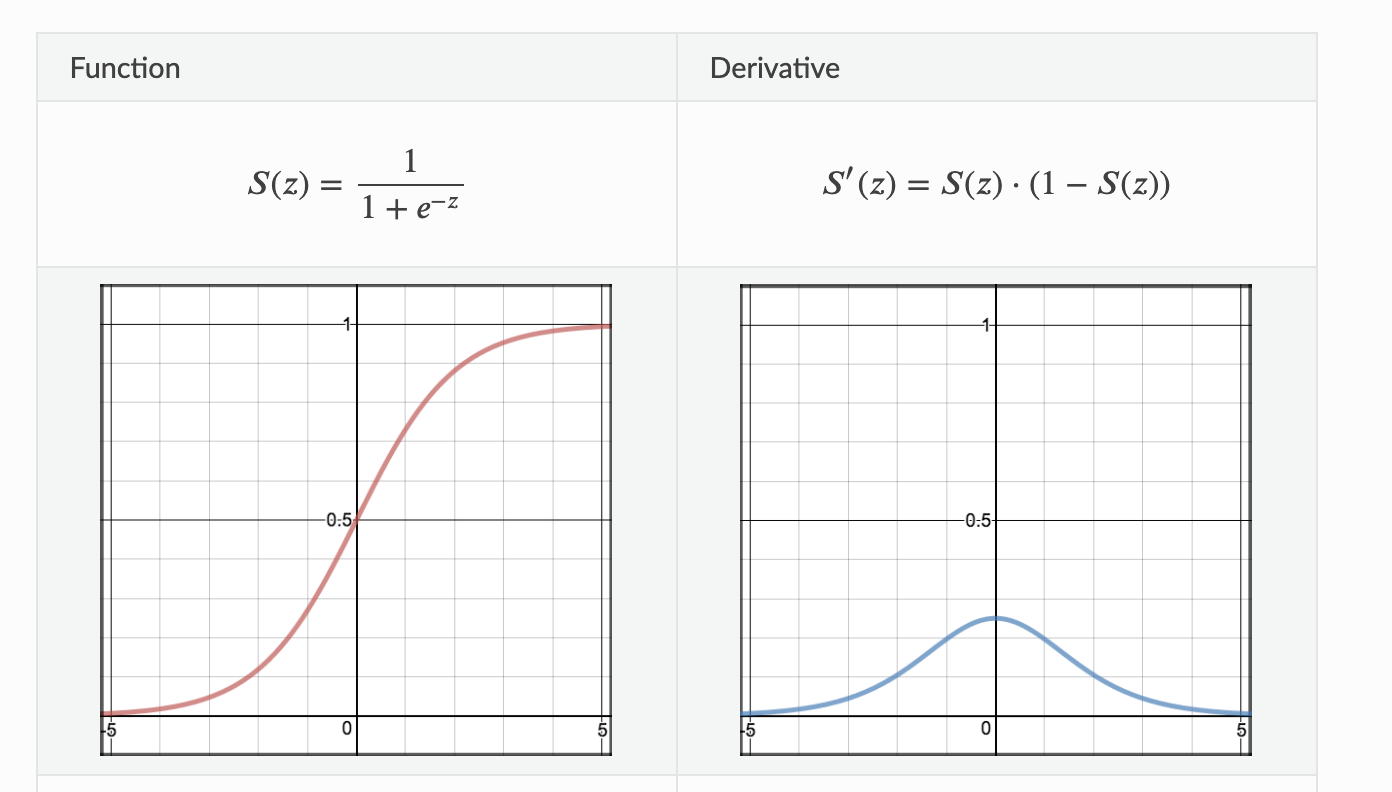
<https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function>

<https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5>

Sve sigmoidne funkcije imaju sposobnost preslikavanja niza brojeva u mali opseg između 0 i 1ili -1 i 1. Tako se sigmoidna funkcija može koristiti za pretvorbu realnog broja u vrijednost koja se potom tumači kao vjerojatnost. Zbog toga se sigmoidna funkcija, osim u skrivenim slojevima, može koristiti i u izlaznom sloju gdje pretvara izlazni rezultat modela u rezultat vjerojatnosti s kojim se onda može lakše raditi i koji se može lakše protumačiti. Zbog svog 'S' oblika na grafu nosi naziv sigmoidna funkcija. Taj naziv se često odnosi na naziv 'logistička sigmoidna funkcija'.

Matematička formula sigmoidne funkcije glasi:

Grafički prikaz sigmoidne funkcije i njene derivacije prikazan je na slici.



Slika 1: sigmoidna aktivacijska funkcija i njena derivacija

Sigmoidna funkcija je ne-linearna, kontinuirano diferencirana, monotona funkcija. Glavna prednost joj je to što je jednostavna i dovoljno dobra kao funkcija za klasifikaciju. S druge strane, veliki nedostatak joj je što uzrokuje problem 'nestajućeg gradijenta' (eng. *vanishing gradient problem*) jer joj vrijednost nije centrirana oko nule. Zbog toga ažurirane vrijednosti mogu ići predaleko u različitim smjerovima. Nadalje, korištenjem sigmoidne funkcije, teže je provesti optimizaciju te računanje u skrivenim slojevima oduzima puno vremena.

### KORIŠTENJE SIGMOIDNE AKTIVACIJSKE FUNKCIJE

Glavno područje strojnog učenja, gdje je bitno korištenje sigmoidne funkcije, je područje korištenja logističkog regresijskog modela. Model logističke regresije se koristi kako bi se procijenila vjerojatnost na binarnom nivou, kao što je, na primjer, 'živo' ili 'neživo', 'lažno' ili 'istinito' i slično. Model daje vrijednost koja se nalazi između 0 i 1.

Razlog zašto se baš (logistička) sigmoidna funkcija koristi u logističkog regresiji je zbog činjenica da funkcija uvijek vraća vrijednost između 0 i 1, a logistička regresija je izvedena iz pretpostavke da su podaci iz obe klase normalno distribuirani.

### USPOREDBA SIGMOIDNE I ReLU FUNKCIJE

U modernim umjetnim neuralnim mrežama običaj je vidjeti, umjesto sigmoidne neuralne funkcije, ReLU aktivacijsku funkciju.

ReLU aktivacijska funkcija donosi nekoliko glavnih prednosti u odnosu na sigmoidnu funkciju. Glavna prednost jest da se ReLU funkcija puno brže računa. Nadalje, aktivacijski potencijal se u biološkim neuralnim mrežama ne minjenja za negativne ulaze. Zbog toga se čini da ReLU funkcija bolje oponaša biološke neuralne funkcije.

Još jedna u nizu prednosti se odnosi na pozitivne vrijednosti . ReLU funkcija ima konstantni gradijent 1, dok sigmoidna funkcija ima gradijent koji brzo konvergira prema 0. Ovo svojstvo čini neuralnu mrežu, koja ima sigmoidnu aktivacijsku funkciju, sporom za učenje.

## TANH AKTIVACIJSKA FUNKCIJA

## https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

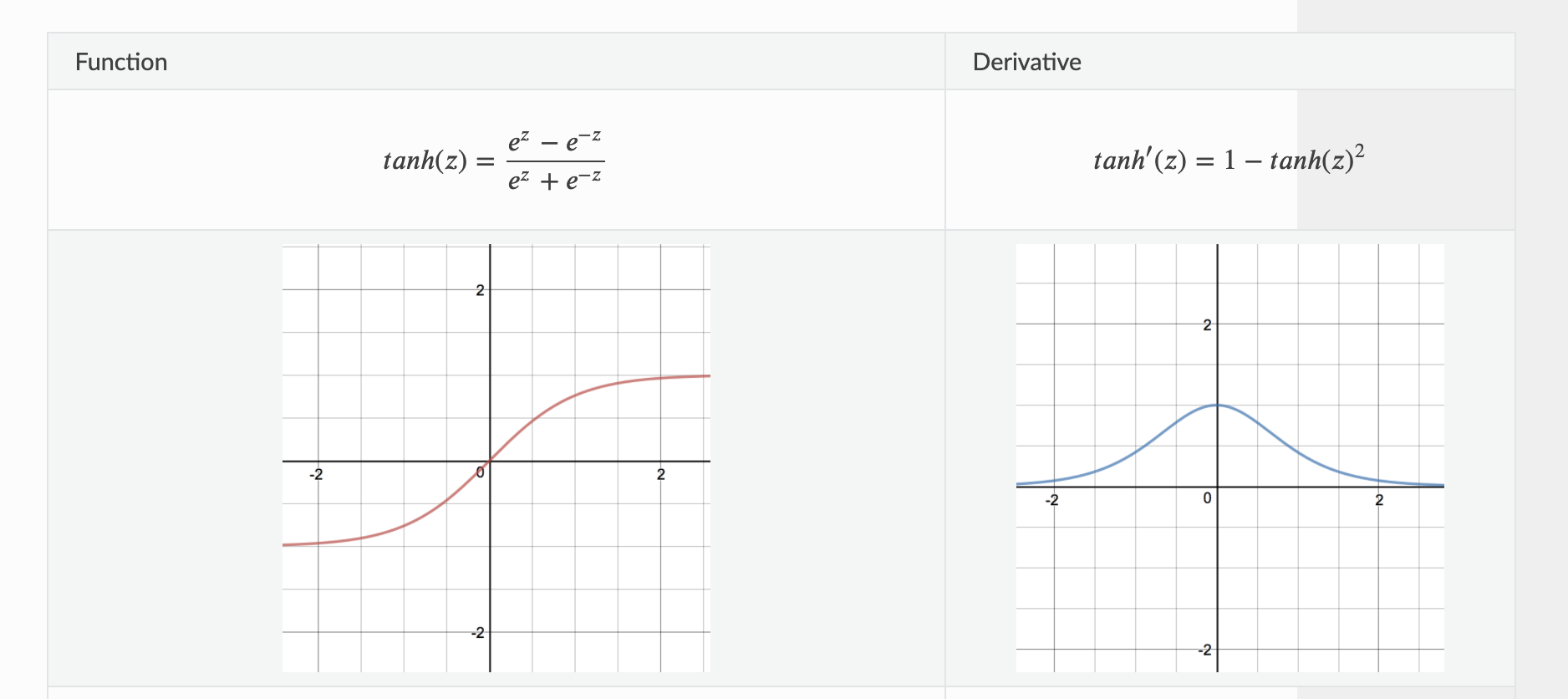
<https://paperswithcode.com/method/tanh-activation>

<https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5>

Tanh je poboljšana sigmoidna funkcija. Opseg tanh funkcija je od -1 do 1 te, kao i sigmoidna funkcija, ima 'S' oblik.

Matematička formula tanh funkcije glasi:

Grafički prikaz tanh funkcije i njene prve derivacije prikazan je na donjoj slici:



Slika 2: grafički prikaz tnh funkcije i njene prve derivacije

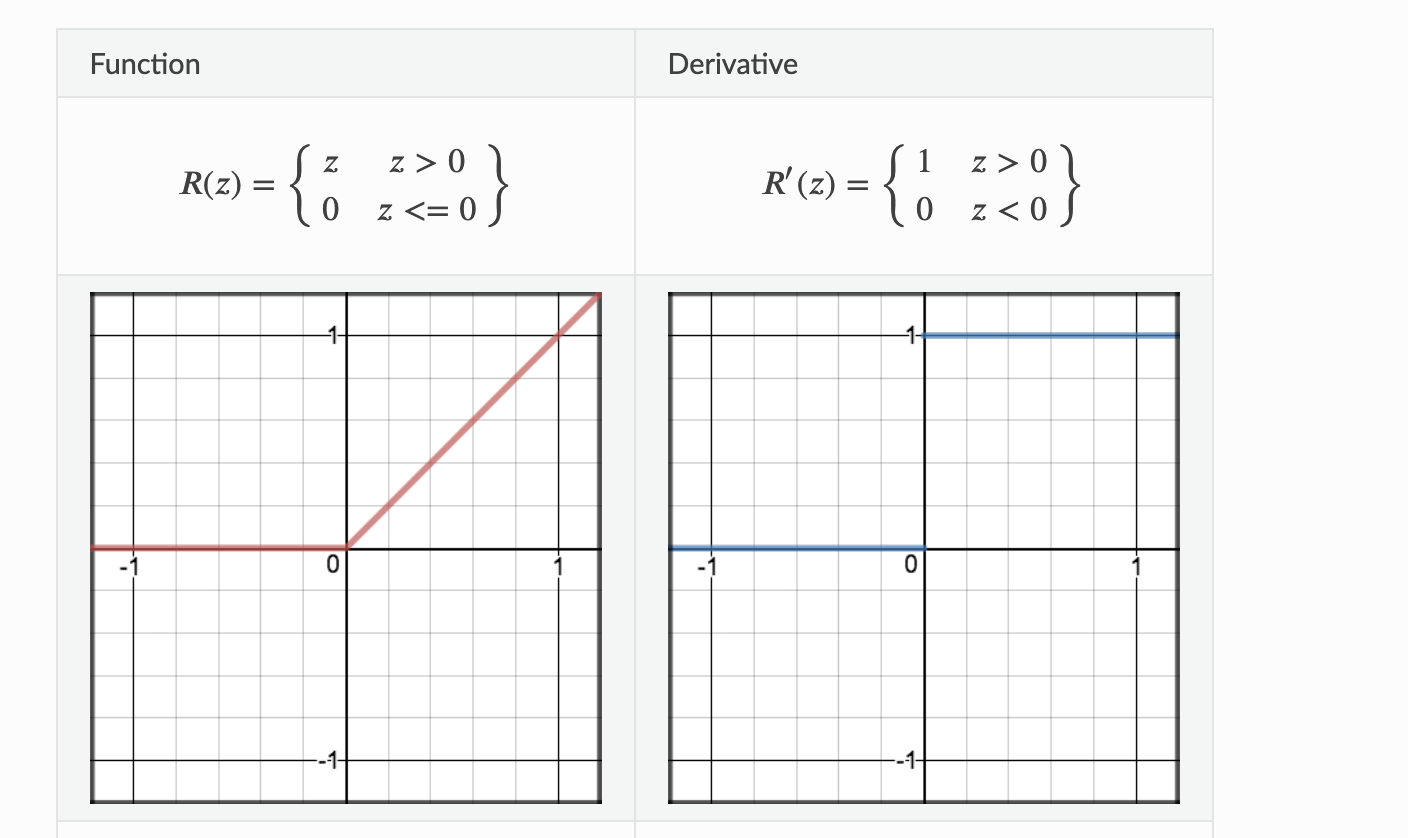
Prednost Tanh nad sigmoidnom funkcijom je što će negativne vrijednosti biti preslikane u izrazito negativne vrijednosti, a vrijednosti, koje imaju približnu vrijednost 0, će biti preslikane u vrijednosti koje su blizu 0 i tako rješava problem sigmoidne funkcije. Nadalje, pokazala se da daje rezultate na više-slojnim neuralnim mrežama. Međutim, kao i sigmoidna funkcija, uzrokuje problem nestajućeg gradijenta, zbog čega se danas preferira korištenje ReLU funkcije.

Ova funkcija se najčešće koristi za klasifikaciju između dvije klase.

## ReLU AKTIVACIJSKA FUNKCIJA

<https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>

ReLU aktivacijska funkcija je kratica od *rectified linear unit*, koja je izražena matematičkom formulom: . To je jedna od najboljih i najčešće korištenih aktivacijskih funkcija koja se danas koristi (pogotovo u konvolucijskim neuralnim mrežama) u odnosu na sigmoidnu i TanH funkciju jer se uspješno suočava s problemom nestajućeg gradijenta (eng. *vanishing gradient problem*). Grafička reprezentacija ReLU aktivacijske funkcije i njene derivacije prikazane su na donjoj slici.



Slika 3: ReLU aktivacijska funkcija i njena derivacija

Kao što se može vidjeti na slici, ReLU aktivacijska funkcija je linearna za sve pozitivne vrijednosti i nula za sve negativne vrijednosti. To znači nekoliko stvari:

1. ReLU funkcija se jednostavno računa, stoga će modelu trebati manje vremena za treniranje i kasnije donošenje odluka.
2. Brže konvergira. Linearnost ReLU funkcije znači da njen nagib nema 'visoravan' kada x poprimi veliku vrijednost.
3. Slabo je aktivirana. S obzirom da ReLU poprima vrijednost nula za sve negativne ulazne vrijednosti, postoji vjerojatnost da se neki neuroni u mreži nikada neće aktivirati, što je u većini slučajeva dobro.

U biološkim neuralnim mrežama, gdje postoji na milijarde neurona, nisu svi neuroni aktivni u isto vrijeme jer imaju različite uloge i aktiviraju ih različiti podražaji. Slaba aktiviranost u modelima rezultira u bržim i boljim predviđanjima i manjem *overfitting*-u.

### DOKAZ DA JE ReLU NE-LINEARNA FUNKCIJA

Kako bi se dokazalo da je ReLU aktivacijska funkcija ne-linearna funkcija, pokazat će se da ReLU ne uspijeva biti linearna funkcija. Kao što je već spomenuto, aktivacijske funkcije su ne-linearne funkcije.

Za svaki realan broj *x*, funkcija *f* je definirana tako da je:

Neka se pretpostavi da je *a* realan broj za koji vrijedi .

Koristeći pretpostavku da vrijedi , može se vidjeti da je:

i da je:

Radi ovoga dolazimo do zaključka:

Tako je pokazano da funkcija *f*, odnosno ReLU funkcija, ne uspijeva biti linearna funkcija.

### UMIRUĆI ReLU

Kao što je već prije spomenuto, oskudnost ReLU funkcije je u većini slučajeva poželjna jer to rezultira u bržim i boljim rezultatima te manjim *overfitting*-om. Međutim, negativna strana toga što ReLU poprima vrijednost nula za sve negativne vrijednosti je ta što se može pojaviti problem koji se naziva 'umirući ReLU' (eng. *dying ReLU*).

ReLU neuron je 'mrtav' ako je zapeo na negativnoj strani vrijednosti zbog čega uvijek ima vrijednost nula. Kako je nagib ReLU funkcije za negativne vrijednosti nula, jednom kada neuron poprimi negativnu vrijednost, malo je vjerojatno da će se oporaviti od te vrijednosti. Takvi neuroni ne pridonose u preslikavanju ulaza na izlaz te postaju praktički beskorisni. Tijekom vremena se može dogoditi da veliki dio mreže ne radi ništa (u nekim slučajevima čak oko 40%).

Zbog toga se, osim ReLU funkcije, znaju koristiti njene izvedenice koje nastoje zaobići taj problem. Neke od njih će biti opisane u zasebnim poglavljima. Neke od izvedenica su:

1. Leaky ReLU
2. Parametric ReLU (PReLU)
3. ReLU-6
4. Concatenated ReLU (CReLU)

## LEAKY ReLU AKTIVACIJSKA FUNKCIJA

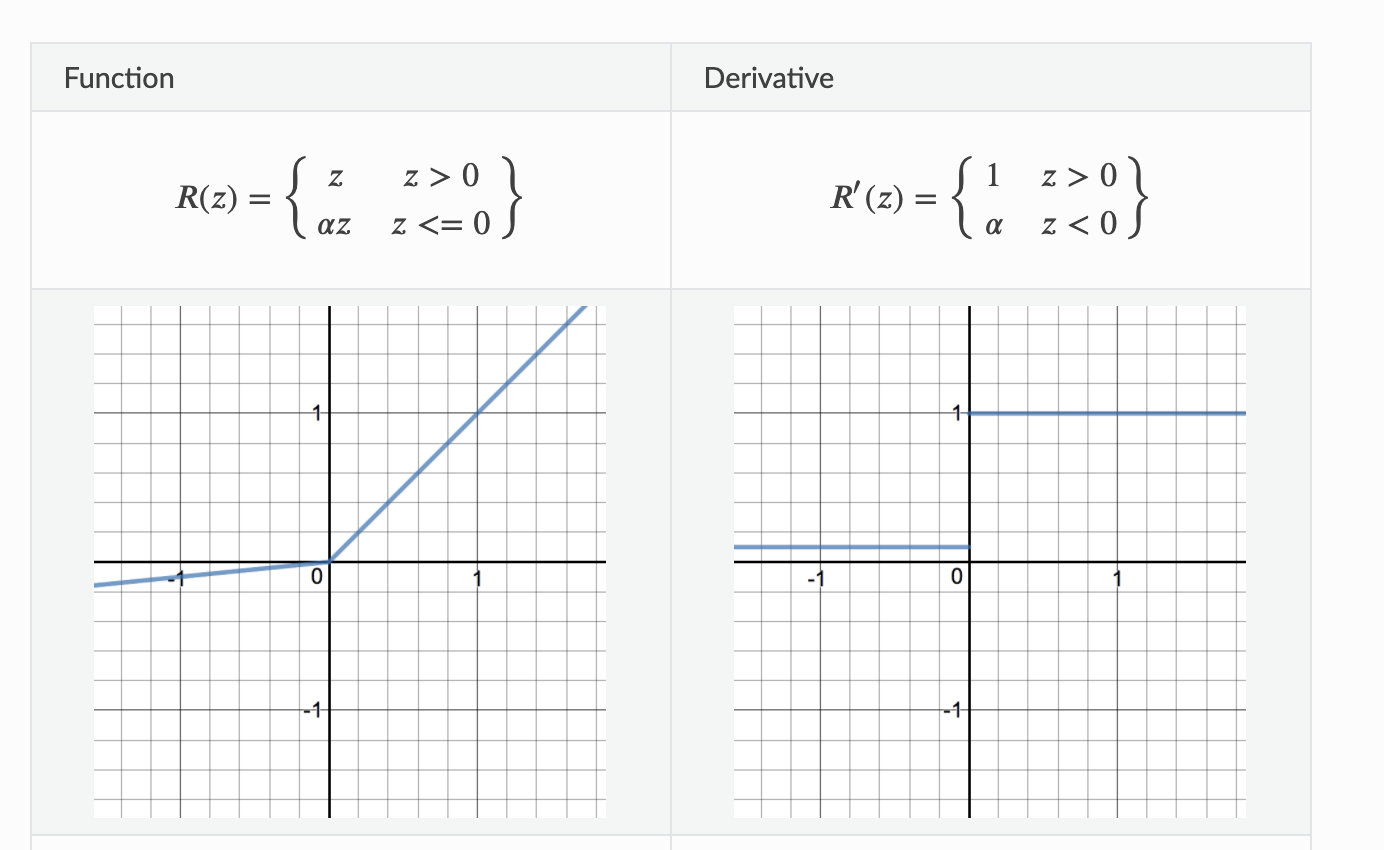
<https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5>

https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

*Leaky* ReLU aktivacijska funkcija je izvedenica ReLU funkcije. Izvedena je s ciljem da riješi problem 'umirućeg ReLU-a'. Ona, za razliku od ReLU funkcije, ima lagani pozitivni nagib u negativnom dijelu vrijednosti i tako omogućuje izvedbu povratne propagacije i za negativne vrijednosti. To znači da je opseg *leaky* ReLu funkcije .

Matematička formula *Leaky* ReLU funkcije glasi:

Grafički prikaz *leaky* ReLU funkcije kao i njene prve derivacije prikazan je na donjoj slici:



Slika 4: grafički prikaz leaky ReLU funkcije i njene prve derivacije

Međutim, *leaky* ReLU ne nudi konzistentna predviđanja za negativne vrijednosti. Tijekom propagacije prema naprijed, ako je stopa učenja postavljena na preveliki broj, uzrokovat će *overshooting* i tako 'ubiti' neuron.

Zbog toga se ideja *leaky* ReLU-a može proširiti. Umjesto toga da se *x* pomnoži nekim konstantnim pojmom, može se pomnožiti s hiper-parametrom kojeg mreža može naučiti tijekom treniranja. Ova izvedenica *leaky* ReLU-a je poznata kao *parametric* ReLU.

## SWISH AKTIVACIJSKA FUNKCIJA

<https://towardsdatascience.com/swish-booting-relu-from-the-activation-function-throne-78f87e5ab6eb>

<https://medium.com/@neuralnets/swish-activation-function-by-google-53e1ea86f820>

Aktivacijske funkcije imaju dugu povijest. Prvo se koristila sigmoidna funkcija zbog svoje jednostavne derivacije, opsega između 0 i 1 te zbog svog probabilističkog oblika. Tanh funkcija se smatrala zamjenom za 'običnu' sigmoidnu funkciju jer je ulazne vrijednosti preslikavala u vrijednosti koje se nalaze između -1 i 1. Međutim, te dvije funkcije su u većini slučajeva zamijenjene ReLU aktivacijskom funkcijom.

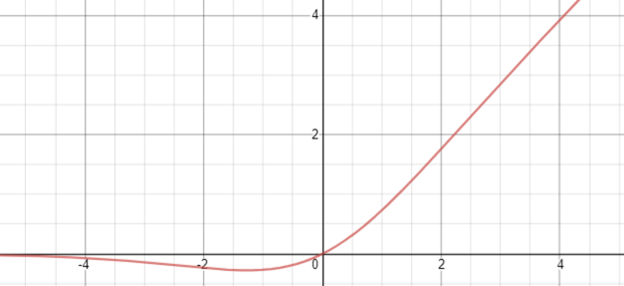
Kao i ReLU, Swish, aktivacijska funkcija razvijena od strane Google razbojnog tima, je omeđena dolje. To znači da, kako se vrijednost približava negativnoj vrijednosti, se približava nekoj konstantnoj vrijednosti. Isto tako je, kao i ReLU, neograničena od gore (kako se vrijednost približava pozitivnoj vrijednosti, se približava nekoj beskonačnoj vrijednosti). Ali, za razliku od ReLU funkcije, Swish je 'glatka', odnosno nema iznenadne promjene gibanja.

Matematička formula Swish aktivacijske funkcije glasi:

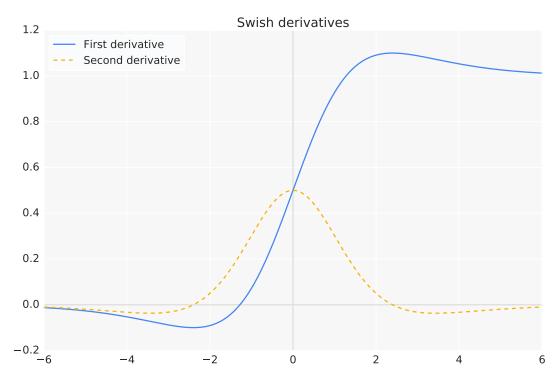
odnosno:

Derivacija Swish funkcije je onda:

Grafički prikaz Swish funkcije kao i grafički prikaz njene derivacije nalaze se na slikama.



Slika 5: grafički prikaz Swish aktivacijske funkcije

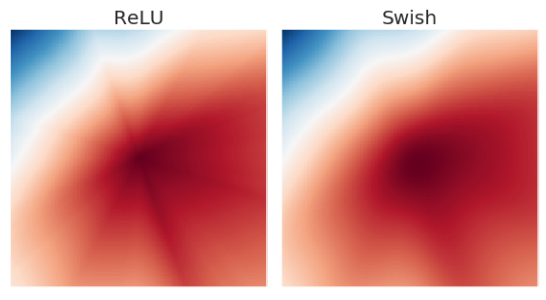


Slika 6: grafički prikaz prve (plavo) i druge (žuto) derivacije Swish funkcije

### SVOJSTVA SWISH FUNKCIJE

Neomeđenost funkcije je poželjna kod svake aktivacijske funkcije jer se time izbjegava dugo vrijeme učenja s vrijednostima gradijenta koje su blizu 0, kao što su sigmoidna i tanh funkcije koje su ograničene od gore i od dolje i zbog čega se mreže moraju oprezno inicijalizirati.

Imati funkciju koja je omeđena od dolje, kao što je ReLU funkcija, daje prednost tijekom procesa učenja jer se odbacuju velike negativne ulazne vrijednosti. Nadalje, uglađenost funkcije pomaže kod optimiziranja i poopćavanja neuralne mreže. Na prikazanoj slici su uspoređene ReLU i Swish funkcija gdje se vidi kako Swish funkcija ima bolju uglađenost.



Slika 7: prikaz uglađenosti ReLU i Swish aktivacijske funkcije

Pejzažna glatkoća direktno korelira s pejzažem greški, što je on glađi, to je lakše pronaći minimum.

Eksperimenti, koje je proveo Googleov razvojni tim, pokazali su da Swish funkcija daje bolje rezultate od ReLU funkcije na dubljim modelima. Na primjer, samo zamjenom ReLU funkcije Swish funkcijom, poboljšana je točnost klasifikacije ImageNet baze podataka za 0.9% i NASANetA za 0.6%. Nadalje, Swish ima bolje izvođenje od ReLU kod modela koji imaju između 40 i 50 slojeva, kada optimizacija postaje teška. Što se tiče veličine podskupova (eng. *batch*), obje funkcije bilježe pad u kvaliteti izvođenja, ali i tada Swish ima bolju izvedbu od ReLU.

## SOFTMAX AKTIVACIJSKA FUNKCIJA

<https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5>

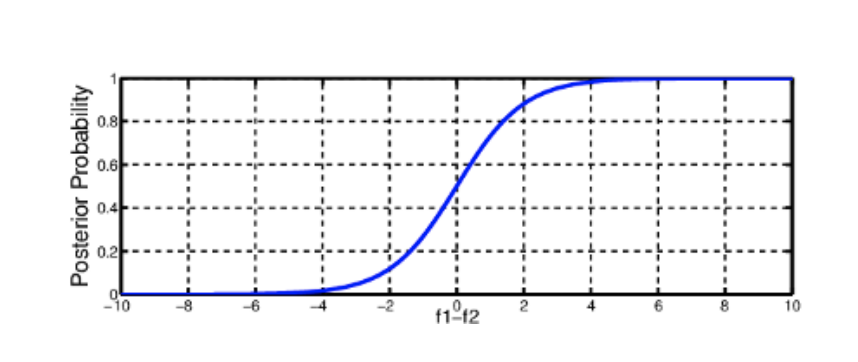
<https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>

Još poznata i kao 'softargmax' funkci ili više-klasna logistička regresija, softmax funkcija je funkcija koja prima vektor od K realnih vrijednosti i vraća vektor K realnih vrijednosti čiji zbroj daje vrijednost 1. Ulazne vrijednosti mogu biti pozitivne, negativne, 0, veće od jedan, manje od jedan, ali će ih softmax funkcija uvijek transformirati u vrijednosti koje se nalaze između 0 i 1 tako da mogu biti interpretirane kao vjerojatnosti.

Matematička formula softmax funkcije slična je onoj sigmoidne funkcije i glasi:

Sličnost je u tome što softmax funkcija prima vektor, dok sigmoidna funkcija prima skalar. Dapače, sigmoidna funkcija je posebni slučaj softmax funkcije kada je ulazne vrijednosti potrebno klasificirati u dvije izlazne klase.

Grafički prikaz sigmoidne prikazan je na donjoj slici:



Slika 8: grafički prikaz softmax funkcije

Softmax funkcija se najčešće koristi na kraju neuralne mreže gdje je potrebno izlazne rezultate prikazati kao vjerojatnosti da ulaz pripada određenom izlazu.