# PARAMETRI I HIPERPARAMETRI

www.deeplizard.com

Već se dosta toga spomenulo u vezi parametara koji se mogu naučiti u neuralnoj mreži. Međutim, nije se dao iscrpni popis što to sve neuralna mreža može naučiti.

U ovom poglavlju će se definirati što je parametar koji se može naučiti u neuralnoj mreži. Nakon toga će se opisati kako se broj parametara može izračunati te će se na kraju taj izračun pokazati na jednostavnom primjeru.

Parametar koji se može naučiti u neuralnoj mreži je svaki parametar čiju vrijednost treniranja mreža uči tijekom procesa. Te parametre se može nazivati i parametrima koji se mogu trenirati jer se njihova optimalna vrijednost uči tijekom procesa učenja.

Broj parametara, koje mreža može naučiti, računa se za svaki sloj pojedinačno te se svi brojevi parametara zbroje kako bi se dobio ukupan broj parametara.

Kako bi se izračunao ukupan broj parametara u jednom sloju, prvo je potrebno izračunati nekoliko stvari:

1. Broj ulaza u dani sloj
2. Broj izlaza iz danog sloja
3. Sadrži li, ako da, koliko, sloj pristranosti

Ovdje je potrebno napomenuti da se u ovom slučaju misli na potpuno povezanu mrežu koju čine gusti slojevi.

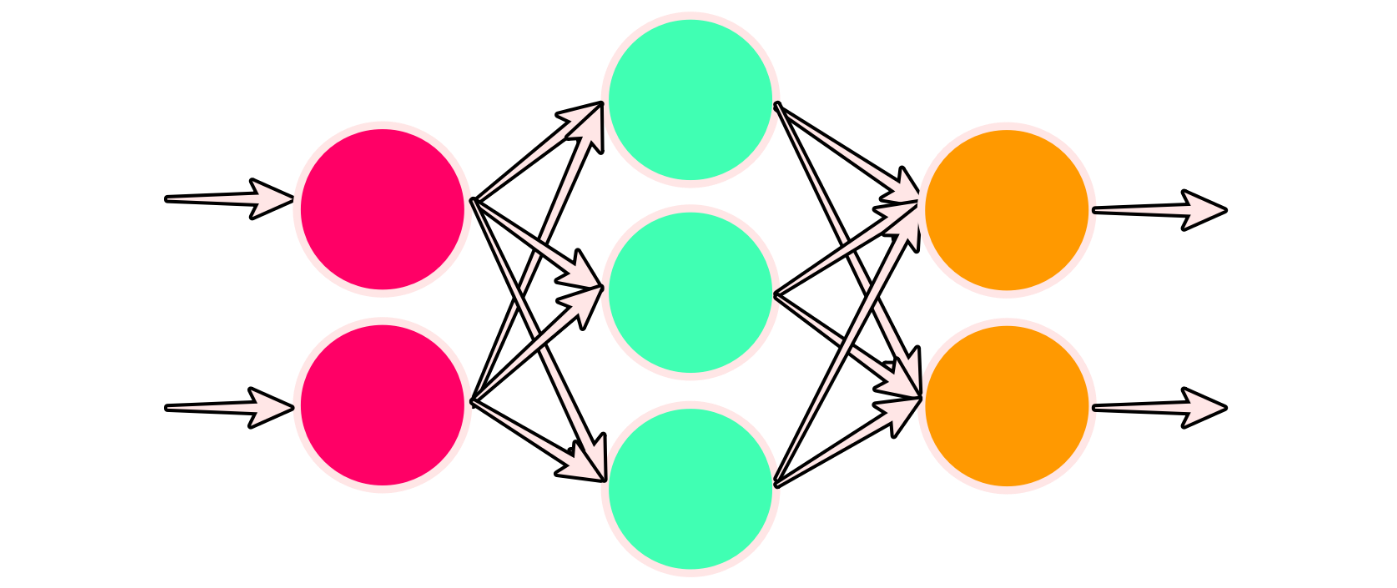
Nakon što se izračunaju brojevi s gore navedene liste, kako bi se izračunao broj parametara, broj ulaza u sloj se pomnoži s brojem izlaza iz čvora . Pod brojem izlaza iz sloja se misli na broj čvorova koji se nalaze u danom sloju kojem se računa broj parametara. Rezultat umnoška jest broj težina veza koje ulaze u taj sloj.

Nakon toga se provjerava sadrži li sloj *bias*-e za svaki čvor. Ako sadrži, broj *bias*-a se jednostavno zbroji s prije dobivenim umnoškom (broj težina). Broj *bias*-a je jednak broju čvorova u danom sloju.

Konačni rezultat jest broj parametara u danom gustom sloju. Isti proces se ponavlja za svaki sloj u neuralnoj mreži. Konačan broj parametara se računa tako što se zbroje parametri za svaki pojedini sloj.

Neka je dana potpuno povezana neuralna mreža ima tri sloja sa sljedećim brojem čvorova:

1. Ulazni sloj s 2 čvora
2. Skriveni sloj s 3 čvora
3. Izlazni sloj s 2 čvora



Slika 28: primjer potpuno povezane neuralne mreže s tri sloja

Na kraju, neka mreža sadrži pristranosti. To znači da svaki čvor u skrivenom i izlaznom sloju sadrži pristranost.

Broj parametara se računa na sljedeći način:

1. Ulazni sloj ne sadrži parametre koji se mogu naučiti jer se on sastoji samo od ulaznih podataka. Izlaz iz tog sloja će se samo smatrati ulazom u sljedeći sloj.
2. Broj ulaza u skriveni sloj jest broj čvorova u prethodnom, odnosno ulaznom, sloju. To je broj 2. Broj izlaza iz danog, skrivenog, sloja jednak je broju čvorova u danom sloju. To znači da imamo tri izlaza iz danog sloja. Pomnože se broj ulaza u sloj i broj izlaza iz sloja kako bi se dobio broj težina veza. Dobiven je broj 6. Kako bi se dobio ukupan broj parametara za dani sloj, umnošku se doda broj pristranosti koji je jednak broju čvorova u danom sloju, odnosno 3. Ovaj sloj ima ukupno 9 parametara koji se mogu naučiti.
3. Broj ulaza u izlazni sloj je 3 jer je broj čvorova u prethodnom, skrivenom, sloju 3. Broj izlaza iz izlaznog sloja je 2 jer ima toliko izlaznih čvorova. Umnožak ta dva sloja 6. Na kraju se umnošku doda broj pristranosti i dobije se da je ukupan broj parametara u izlaznom sloju broj 8.
4. Ukupan broj parametara koji mreža može naučiti se dobije zbrajanjem broja parametara skrivenog sloja i broja parametara izlaznog sloja. Mreža može naučiti ukupno 17 parametara. Tijekom procesa treniranja, SGD će učiti i optimizirati svih 17 parametara.

## PARAMETRI KOJI SE MOGU NAUČITI U KONVOLUCIJSKOJ NEURALNOJ MREŽI

U prethodnom poglavlju je opisano kako se računa broj parametara koje može naučiti potpuno povezana neuralna mreža s gustim slojevima. U ovom poglavlju će se opisati kako se taj broj računa kada se radi s konvolucijskom neuralnom mrežom.

Prvo će se objasniti što su to parametri koje može naučiti konvolucijska neuralna mreža te kako se ukupan broj tih parametara može izračunati. Na kraju će biti dan jednostavan primjer kako se taj broj računa u knovolucijskoj neuralnoj mreži.

Kao i kod standardne neuralne mreže, broj parametara u konvolucijskoj neuralnoj mreži se računa za svaki sloj posebno te se ukupan broj parametara dobije tako da se zbroji broj parametara za svaki pojedinačni sloj.

Kod gustih slojeva, broj parametara se računao na sljedeći način:

ulazi \* izlazi + pristranosti

Međutim, konvolucijska mreža obično ne posjeduje guste slojeve. Isto tako, za razliku od standardnih neuralnih mreža, konvolucijska mreža sadrži filtere, koji se još nazivaju kernelima. Kada se strukturira kovnolucijska mreža, potrebno je odrediti broj i veličinu filtera koji će se koristiti i taj će broj poslije ući u izračun broja parametara.

Potrebno je voditi računa o tim, gore navedenim, stvarima jer se, također gore navedena, formula modificira s obzirom na to:

1. je li prethodni sloj konvolucijskog sloja bio gusti sloj. Ako je, ulaz u konvolucijski sloj je jednostavno broj čvorova u prethodnom gustom sloju.
2. Je li prethodni sloj konvolucijskog sloja bio konvolucijski sloj. Ako je, broj ulaza u sloj je broj filtera koje posjeduje taj prethodni konvolucijski sloj.

Isto tako, izlaz iz konvolucijskog sloja je broj koji se dobije množenjem broja filtera s njihovim dimenzijama. Konačno, broj pristranosti će biti jednak broju filtera u danom konvolucijskom sloju.

Tako je sačuvana općenita postavka za broj parametara koji se mogu naučiti, umnožak broja ulaza s brojem izlaza koji se potom zbroji s brojem pristranosti.

Neka se konvolucijska neuralna mreža sastoji od ulaznog sloja, dva skrivena sloja i gustog izlaznog sloja:

1. ulazni sloj
2. skriveni konvolucijski sloj
3. skriveni konvolucijski sloj
4. gusti izlazni sloj.

Neka ulazni sloj sadrži ulazne podatke slika čije su dimenzije 20x20x3, gdje 20x20 definira širinu i visinu slike, a broj 3 definira broj kanala. Tri kanala indiciraju da su slike u RGB-u te ova tri kanala predstavljaju ulazne značajke u ulalzni sloj.

Prvi konvolucijski sloj sadrži 2 filtera dimenzija 3x3. Drugi konvolucijski sloj sadrži 3 filtera dimenzija 3x3. Izlazni sloj je gusti sloj s 2 čvora.

Neka konvolucijska mreža sadrži pristranosti i neka je pretpostavljeno popunjavanje nulama kako bi se sačuvale dimenzije slika.

Do sada vrijedi:

1. ulazni sloj: slike dimenzija 20x20x3
2. skriveni konvolucijski sloj: 2 filtera dimenzija 3x3
3. skriveni konvolucijski sloj: 3 filtera dimenzija 3x3
4. gusti izlazni sloj: 2 čvora

Ulazni sloj ne sadrži parametre koji se mogu naučiti jer oni samo sadrži ulazne podatke.

Prvi skriveni sloj ima tri ulaza iz ulaznog sloja. Broj izlaza iz tog sloja jest umnožak broja filtera s njihovim dimenzijama. Kako taj sloj sadrži dva filtera, svaki dimenzija 3x3, broj izlaza iz konvolucijskog sloja će biti 2\*3\*3 = 18. Množenjem broja ulaza s brojem izlaza dobije se broj od 3\*18 = 54 težine. Na kraju se taj broj zbroji s brojem pristranosti i dobije ukupan broj od 56 parametara koji se mogu naučiti u prvom skrivenom konvolucijskom sloju.

Drugi skriveni sloj ima dva ulaza iz prethodnog sloja jer prethodni sloj sadrži dva filtera. Broj izlaza iz trenutnog sloja jest umnožak broja filtera s njihovim dimenzijama, to je 3\*3\*3 = 27.

Umnoškom broja ulaza s brojem izlaza se opet dobije broj od 54 težine. Na kraju se tom broju doda broj pristranosti i dobije broj od 57 parametara koje mreža može naučiti u drugom skrivenom konvolucijskom sloju.

S obzirom da je pretpostavljeno da mreža koristi popunjavanje nulama, sačuvane su dimenzije slika koje ulaze u mrežu (20x20). Tako se množenjem ta dva broja dobije broj od 1200 ulaza u izlazni sloj. Sada, s obzirom da je izlazni sloj gusti sloj, broj izlaza je jednak broju čvorova u danom sloju. Množenjem broja ulaza s brojem izlaza, dobije se broj od 2400 težina koje povezuju izlazni i njemu prethodni sloj. Kada se doda broj pristranosti čvorova, dobije se ukupan broj 2402 parametra koje konvolucijska neuralna mreža može naučiti u izalznom sloju.

Kada se zbroje svi parametri iz svakog pojedinog sloja mreže, dobije se ukupan broj od 2515 parametara koje mreža može naučiti za danu strukturu.

Može se vidjeti da je, općenito, proces određivanja broja parametara isti kao i kod standardne neuralne mreže, samo što se mora voditi računa od dodatnim stvarima kao što su broj kanala, broj filtera i njihovih dimenzija.

<https://missinglink.ai/guides/neural-network-concepts/hyperparameters-optimization-methods-and-real-world-model-management/>

Hiperparametri neuronske mreže utječu na funkcioniranje mreže, njenu ispravnost preslikavanja i valjanost. Hiperparametri su neriješeni problem – postoji više načina kako ih optimizirati, od manualnog pokušaja i pogreške do sofisticiranih algoritamskih metoda i ne postoji industrijski konsenzus koja je najbolja praksa.

RAZLIKA IZMEĐU PARAMETRA I HIPERPARAMETRA

U neuronskim mrežama, parametri se koriste kako bi mreža naučila predviđati. Postoje dva tipa parametara:

1. parametri
2. hiperparametri

Parametri se nalaze unutar neuronske mreže. To su na primjer težine veza. Njihove vrijednosti se izračunavaju i uče na temelju primjera za učenje. Ovi parametri se koriste kako bi mreža donosila predviđanja i u produkciji.

Hiperparametri su vanjski parametri neuronske mreže koje postavlja operator neuronske mreže. To su, na primjer, odabir aktivacijske funkcije ili veličina grupe tijekom procesa učenja. Hiperparametri imaju veliki utjecaj na preciznost neuronske mreže. Mogu postojati različite optimalne vrijednosti i nije ih jednostavno otkriti.

Najjednostavniji način prilikom odabira hiperparametara za neuronsku mrežu je manualna potraga – drugim riječima pokušaj i pogreška.

LISTA HIPERPARAMETARA

1. Količina skrivenih slojeva – dodavanjem više skrivenih slojeva će povećati preciznost mreže, ali isto tako povećava rizik od prenaučenosti.
2. Dropout – koliki će se postotak nasumično odabranih neurona ignorirati tijekom svake epohe kako bi se spriječila prenaučenost.
3. Aktivacijska funkcija – koja bi se aktivacijska funkcija trebala koristiti za obradu ulaznih podataka koji ulaze u svaki pojedini neuron. Aktivacijska funkcija može utjecati na preciznost i brzinu učenja mreže.
4. Inicijalizacija težina – potrebno je postaviti neke inicijalne težine na veze u neuronskoj mreži. Dvije osnovne opcije su postavljanje težina na nulu ili inicijalizacija nasumičnim brojčanim vrijednostima. Kod bilo koje opcije postoji opasnost od problema nestajućeg i eksplodirajućeg gradijenta. Još jedna od opcija je Xavier heuristička inicijalizacija težina.

HIPERPARAMETRI POVEZANI S ALGORITMOM UČENJA

1. Stopa učenja neuronske mreže – koliko brzo algoritam propagacije prema natrag izvodi gradijentni spust. Niža stopa učenja produljuje vrijeme učenja, ali zato izbjegava mogućnost dobivanja pretjeranog rezultata (eng. *overshooting*).
2. Epoha, broj iteracija i veličina grupe – ovi parametri određuju kojom brzinom će se primjeri prosljeđivati mreži tijekom procesa učenja..
3. Algoritam optimizacije – osnovna opcija je stohastički gradijentni spust, ali postoji više opcija. Jedna od njih je algoritam Momentum ili algoritmi Nesterov akcelerirani gradijent, Adam i AdaDelta.

[www.deeplizard.com](http://www.deeplizard.com)

## VELIČINA PODSKUPOVA U UMJETNOJ NEURALNOJ MREŽI

U ovom poglavlju će se opisati što to znači specificirati veličinu podskupa iz sveukupnog skupa podataka za treniranje.

Veličinu podskupa će činiti broj uzoraka iz cjelokupnog skupa podataka za treniranje koji će se kao cjelina proslijediti mreži u jednom trenutku. Podsjetnik, epohu čini jedan prolazak cijelog skupa podataka za treniranje. Podskup i epoha nisu ista stvar.

### PODSKUPOVI U EPOHI

Neka skup podataka za treniranje čini 1000 slika automobila pomoću koji će se model naučiti da razlikuje različite modele automobila. Ako je batch veličina jednaka 10, to znači da će se skup od 10 slika automobila proslijediti modelu u jednom trenutku.

Kako epohu čini jedan prolazak cijelog skupa za treniranje, bit će potrebno 100 podskupova kako bi se završila jedna epoha.

### RAZLOG KORIŠTENJA PODSKUPOVA

Općenito vrijedi da, što je veća veličina podskupa, model će brže završiti svaku epohu tijekom treniranja. To je zbog toga što, ovisno o računalnim resursima, računalo može procesirati toliko više od jednog uzorka u jednom trenutku.

Međutim, čak i ako računalo može obraditi velike podskupove u jednom trenutku, kvaliteta modela može degradirati jer, ako je podskup, koji se šalje modelu u jednom trenutku, dovoljno velik, model možda neće moći dobro generalizirati podatke koje do tada nije vidio.

Općenito, veličina podskupa je još jedan u nizu hiperparametara koji se moraju testirati i podesiti na temelju toga koliko ispravno konkretni model preslikava podatke tijekom učenja. Ovaj parametar se treba testirati i u odnosu na to koliko dobru izvedbu ima samo računalo u smislu iskorištavanja vlastitih resursa kada se koriste različite veličine podskupova.

Na primjer, kada bi veličina podskupa bio relativno veliki broj, kao 100, tada računalo možda neće raspolagati dovoljnom računalnom moći da paralelno obradi svih 100 slika i to bi bio znak da se veličina skupa mora postaviti na neki manji broj.

### MINI-BATCH GRADIJENTNI SPUST

Korištenjem *mini-batch* gradijentnog spusta, gradijent će promijeniti svoju vrijednost nakon prolaska svakog podskupa (per-batch basis) kroz model. Ovo je suprotno od stohastičkog gradijentnog spusta koji implementira promjene gradijenta nakon prolaska svakog uzorka i suprotno je od *batch* gradijentnog spusta koji implementira promjenu gradijenta nakon svake epohe.