Mašinsko učenje

Istraživanje koncepta veštačkih neuronskih mreža i biblioteka



Milan Stošić

TAble of  
Contents

neuronske mreže Kako rade? 1

Pojednostavljeno 1

Konfigrisanje Crne kutije 2

Treniranje 3

TensorFlow vs PyTorch 4

Korisni linkovi 4

# neuronske mreže Kako rade?

Za početak da objasnimo način rada samih neuronskih mreža.

Diagram

Description automatically generated

## Pojednostavljeno

Prvo ćemo sakriti srednji deo. Sve počinje sa podacima za treniranje koji se sastoje od skupa ulaznih podataka i tačnog rezultata. Posao sakrivenog dela(crne kutije) u sredini je da prodje kroz sve te podatke dovoljno puta dok ne možemo reći da je naštelovan da kad mu damo novi set ulaznih podataka on sam može pretpostaviti rezultat.

## Konfigrisanje Crne kutije

* Koliko će slojeva imati crna kutija?  
  Za jednostavne probleme jedan a za komplikovanije probleme uglavnom više.
* Koliko će neurona imati svaki sloj?  
  - Ne postoji tačan odgovor na ovo pitanje, ako imamo puno ulaznih a malo izlaznih podataka dobar potez bi bio da izaberemo neki broj iz sredine ako imamo npr. 7 ulaznih a 2 izlazna podataka 3 do 4 neurona bi bilo u redu. Definitivno želimo da imamo vise nego 2 puta manje neurona nego ulaznih podataka kako bi izbegli overfitting(nećemo dobijati tačne rezultate)
* Aktivaciona funkcija  
  - Npr. Sigmoid koja svodi vrednost izmedju 0 i 1 ili Tanh koja svodi vrednost izmedju -1 i 1
* Stopa učenja i momentum  
  - Stopa učenja određuje koliko će neki korak u treniranju uticati na težine i biase. Mala stopa učenja znači da će nakon izvesnog vremena mreža biti tačna dok visoka stopa učenja znači da će nakon par koraka mreža pretpostaviti da zna dovoljno i biti sklonija greškama  
  - Momentum određuje koliko će neki prethodni rezultati uticati na težine i biase. Npr. iz prethodnih rezultata se može primetiti da neka iniciajlna težina uvek vraća veliki broj kao rezultat tako da ćemo nastaviti da je smanjujemo bez obzira na sve ostalo.
* Iteracije i željena stopa grešaka(treniranje se zaustavlja kada dostignemo jedno ili drugo)

## Diagram Description automatically generatedTreniranje

Za početak pravimo mozak koji ne zna ništa. Svakoj vezi dodeljujemo random weight a svakom neuronu bias. Test primer sa slike ima 2 ulazna podatka, npr x1 i x2, i 2 moguća rezultata, y1 i y2. Recimo da u našem testnom skupu podataka imamo da x1=25 i x2=15 daju da je y1=1 i y2=0. Za svaki neuron se računa vrednost tako što se sabiraju vrednosti neurona sa leve strane pomnožene težinom svoje veze a zatim se na taj zbir dodaje bias npr.

Neuron 1: (x1 \* .1 + x2 \* .8) + .71 . Ovaj korak se ponavlja za svaki neuron u crnoj kutiji. Nakon što smo dobili naš set random pretpostavki prosleđujemo ih aktivacionoj funkciji, ovaj deo je jako bitan jer se rad isa podacima koji nisu linearni. Dosli smo do trenutka kad je nasa mreža došla do svoje prve pretpostavke, npr. y1=0.3512 i y2=0.7891. Možemo primetiti da je ova pretpostavka skroz pogrešna. Sada je vreme za vraćanje unazad, krenućemo unazad korak po korak i da podešavamo parametre, težine i biase, a koje od njih ćemo menjati zavisi od konfiguracije. Prvi koraci u konfiguraciji koji utiču na podešavanje parametara su stopa učenja i momentum. Kada biramo podešavanja za konfiguraciji bitno je imati u vidu koliko brzo želimo da treniramo našu mrežu i koliko želimo da ona bude tačna, bitno je naći balans. Nakon što uradimo prethodne korake za sve podatke iz našeg testnog skupa možemo reći da smo završili prvu iteraciju a zatim određujemo da li želimo još iteracija ili smo zadovoljni stopom grešaka.

## TensorFlow vs PyTorch

* Razlika između verzija 1.0 i 2.0 kod TensorFlow-a, mora se paziti prilikom pretraživanja na internetu jer moguće je da rešenje za neki problem ne bude tačno jer je iz verzije 1.0 a to nije naznačeno nigde.
* Dosta je sporiji od PyTorch-a
* PyTorch je zastupljeniji kod profesionalnih istraživanja
* TensorFlow ima bolju i laksu integraciju sa web i mobilnim aplikacijama
* Sve više velikih kompanija poput OpenAI, Microsoft, Tesla itd. Se opredeljuje za PyTorch sto ga mozda cini perspektivnijim za ucenje
* Mnogo vise treniranih SOAT modela je dostupno u PyTorchu ali kako mi treba da treniramo nase modele mislim da je ova tacka manje bitna

Moj izbor je TensorFlow zbog lakse integracije i zbog vise dostupnih tutoriala.

## Korisni linkovi

<https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/>

<https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi>

<https://www.youtube.com/watch?v=GvQwE2OhL8I&ab_channel=LearnCode.academy>

<https://www.youtube.com/watch?v=NuJB-RjhMH4>