Logistička i softmaks regresija (opis metoda, implementacija, primena nad podacima, komparativna analiza) i njihova primena kao aktivacione funkcije u neuronskim mrežama

Student: Danilo Veljović, broj indeksa: 1120

Novembar 6, 2020

Sadržaj

Glava 1

Uvod

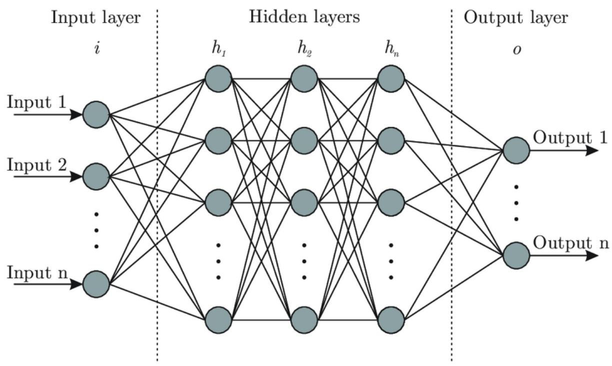
Termini poput veštačke inteligencije i mašinskog učenja su sve popularniji u savremenoj kulturi. Preko medija sve češće čujemo nagoveštaje vodećih naučnika iz ove oblasti da će sistemi koji koriste veštačku inteligenciju zameniti ljude na mnogim radnim mestima. Priča se da će prvo nestati najrepetitivniji poslovi poput vozača kamiona, radnika u skladištima, prodavaca i dr. Treba postaviti suštinsko pitanje zašto je to tako. Zašto će prvo rutinski poslovi nestati prvi? Odgovor na to pitanje daje prave smernice kada neko pokušava da shvati šta je to mašinsko učenje.

Mašinsko učenje vuče korene iz statistike. Slobodno rečeno, mašinsko učenje se može shvatiti kao primena statističkih metoda nad velikom količinom podataka, da bi se izvukle nekakve pravilnosti iz podataka. Te pravilnosti uočava model i opisuje ih različitim statističkim terminima. Model se može shvatiti kao parametrizovana funkcija koja može da „uči" iz podataka. Podaci se u mašinskom učenju dele na trening i test skupove. Trening skupovi imaju određene vrednosti za atribute i imaju izlaznu labelu klase kojoj pripadaju ili neke kontinualne vrednosti koju uzimaju. Model uzima jedan po jedan podatak iz trening skupa, ili čitavu grupu, i proba da napravi predikciju koje bi izlazne vrednosti mogle da odgovaraju tim ulazima. Kada vidi koliko njegova procena odudara od vrednosti labele za taj podatak, on se ispravi svoje parametre. Ovaj proces se naziva treniranje modela. Kada model optimizuje svoje parametre, i kada se odradi evaluacija modela, prelazi se na testiranje. Kada se model testira njemu se daju instance podataka koje nikad nije video. Za te instance on sada predviđa izlazne labele.

Posle ovog kratkog objašnjenja procesa mašinskog učenja, lako se može naslutiti odgovor na prethodno postavljeno pitanje o poslovima koje će prvo „progutati" inteligentni sistemi. Uzmimo primer radnika koji radi u skladištu i pomera kutije od mesta A do mesta B. Robot može da posmatra radnika koji nosi kutije i da zaključi da radnik na mestu A podigne kutiju, nosi je do mesta B i tu je spušta. To je njegov „trening skup" podataka. Kod ovakvog tipa inteligentih sistema najčešće se koriste konvolucione neuronske mreže za računarski vid, pomoću kojih sistem prima podatke o tome šta se dešava oko njega. Princip ostaje isti, sistem preko kamera, prima ulazne podatke, i ovo služi kao trening skup podataka. Trenira određen model, i kada se završi taj proces on može da izvede zadatak prenosa kutija. Ako se od sistema zahteva da radi nešto novo, slično prethodnom, on će probati to da uradi na jedan način, i u zavisnosti od ishoda „učiće" o tome gde je pogrešio, odnosno optimizovaće model.

Tehnike mašinskog učenja su srž inteligentnih sistema. Neki od važnijih modela mašinskog učenja su:

- *Klasifikacija* predviđa klasu koja odgovara podatku
- *Regresija* predviđa kontinualnu vrednost koja odgovara podatku
- *Klasterizacija* je tehnika nenadgledanog učenja. Za razliku od tehnika nadgledanog učenja, o kojima je do sad bilo reči, kod nenadgledanog učenja nemamo informaciju o tome koji podatak ima kakvu klasu/vrednost. Zadatak klasterizacije je da one podatke koji su „slični" (mera sličnost se definiše kao parametar tehnike) grupiše zajedno.
- *Neuronske mreže* su tehnika koja je donela revoluciju u svet mašinskog učenja i omogućila stvari poput računarskog vida i procesuiranja prirodnih jezika. Ideja za ovu tehniku dobijena je iz ideje ljudskog mozga. Cela neuronska mreža se sastoji iz veštačkih neurona, koji su međusobno povezani. Na slici 1.1 dat je prikaz jedne neuronske mreže. Neuronska mreža se sastoji iz tri sloja, *ulaznog* sloja, mnoštvo *skrivenih* slojeva i *izlaznog sloja.* Ideja kod funkcionisanja neuronske mreže je da početni slojevi prepoznaju jako male delove sistema, npr kod računarskog vida prepoznaju ivice, kod procesuiranja prirodnih jezika prepoznaju pojedina slova. Skriveni slojevi prepoznaju složenije pojave, npr geometrijske figure ili tela, ili reči i rečenice. Na kraju izlazni neuroni se aktiviraju ako je prepoznato nešto. Primera radi, mreža može da se trenira da prepoznaje slike mačaka ili da prepoznaje da li je neka rečenica nosi pozitivna ili negativna osećanja. Ako mreža prepozna sliku mačke, ili ako je određena rečenica ima pozitivna osećanja, izlazni neuron se aktivira.



Slika 1.1: Arhitektura neuronske mreže

Jedna od tehnika koja se koristi za klasifikaciju se naziva *logistička regresija.* Iako u nazivu sadrži reč *regresija* ova tehnika se ne koristi za predviđanje kontinualne vrednosti, već za predviđanje da li neki podatak pripada nekoj klasi ili ne. U nastavku rada biće reči o matematičkim osnovama logističke regresije, o funkciji koja je vrlo slična logističkoj regresiji i vrlo često se koristi u sličnim situacijama - *softmaks regresiji.* Implementiraćemo logističku i softmaks regresiju u programskom jeziku Python, primeniti je nad skupom podataka i evaluirati njihove performanse i napraviti komparativnu analizu obe tehnike. Na kraju rada biće dat prikaz primene logističke i softmaks regresije kao aktivacione funkcije („*threshold functions*") kod neuronskih mreža.

Glava 2

Teorijske osnove

2.1 Kratak pregled potrebnih pojmova iz verovatnoće i statistike

Kako bi se pravilno razumela logistička regresija potrebno je pre toga dobro poznavati njene matematičke osnove. U nastavku je data kratka rekapitulacija potrebnih pojmova iz verovatnoće i statistike koji su potrebni za njeno razumevanje. Da bi se lakše svi pojmovi dati u nastavku, razmatrani su samo u okvirima diskretnih slučajnih promenljivih.

2.1.1 Verovatnoća

Verovatnoća predstavlja odnos između broja ishoda u kome je neki događaj ispunjen i broja svih mogućih ishoda.

$$P = \frac{\textit{broj pozitivnih ishoda događaja}}{\textit{broj svih mogućih ishoda}} \tag{2.1}$$

Primer 1. Primer jednog događaja može biti bacanje novčića. Pozitivan ishod je kada se padne glava. Broj svih mogućih ishoda u jednom bacanju je 2 (može se pasti ili pismo ili glava). Kada zamenimo ove vrednosti u prethodnu jednačinu dobijamo:

$$P = \frac{1}{2} = 0.5 \tag{2.2}$$

Iz jednačine se vidi da je verovatnoća da u jednom bacanju novčića dobijemo glavu 0.5, odnosno 50 %.

2.1.2 Šansa

Šansa predstavlja odnos između verovatnoće da se događaj desio i verovatnoće da se događaj nije desio.

$$\textit{šansa} = \frac{\textit{P(događaj se desio)}}{\textit{P(događaj se nije desio)}} = \frac{p}{\textit{1-p}} \tag{2.3}$$

Ako se vratimo na primer 1 bacanja novčića, vidimo da je šansa da se događaj desi, odnosno da se padne glava jednak:

$$\textit{šansa(glava)} = \frac{0.5}{0.5} = 1, tj.\textit{1:1} \tag{2.4}$$

Zaključujemo da je šansa da se padne glava ista kao i šansa da se ne padne glava, tj 1 prema 1.

Odnos šansi se definiše kao:

$$\textit{odnos šansi} = \frac{\textit{šansa}_1}{\textit{šansa}_1} \tag{2.5}$$

Primer 2. Uzmimo dva novčića, jedan koji je fer i drugi za koji se zna da je verovatnoća da se padne glava 0.7. Naći odnos šansi ova dva novčića za događaj kada se padne glava.
Za fer novčić važi:

$$P(glava) = \frac{1}{2} = 0.5 \tag{2.6}$$

$$\textit{šansa(glava)}_{fer} = \frac{0.5}{0.5} = 1, \textit{tj.1:1} \tag{2.7}$$

Za „nefer” novčić važi:

$$P(glava) = 0.7 \tag{2.8}$$

$$\textit{šansa(glava)}_{nefer} = \frac{0.7}{0.3} = 2.333 \tag{2.9}$$

Odnos šansi definišemo kao:

$$\textit{odnos šansi} = \frac{\textit{šansa(glava)}_{nefer}}{\textit{šansa(glava)}_{fer}} = \frac{2.333}{1} = 2.333 \tag{2.10}$$

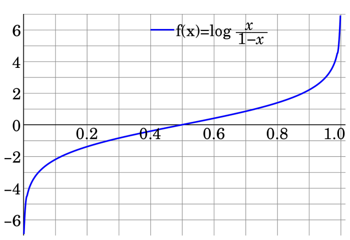
Vidi se da su šanse da se padne glava na „(”nefer) novčiću veća nego da se dobije glava na „(”fer) novčiću 2.333 puta.

2.1.3 Bernulijeva raspodela

2.2 Logistička regresija

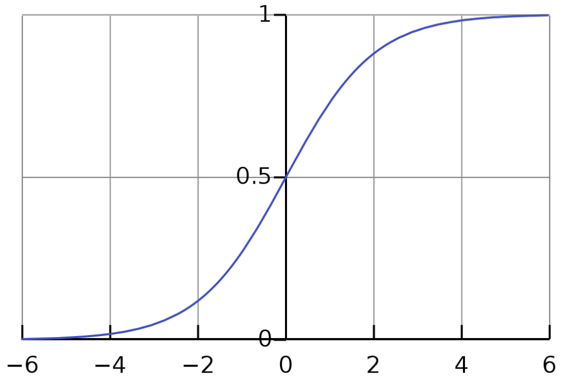
Zavisna promenljiva u logističkoj regresiji ima Bernulijevu raspodelu i ima nepoznatu verovatnoću p. Logistička regresija procenjuje nepoznatu verovatnoću p za neku linearnu kombinaciju nezavisnih promenljivih (najčešće su to elementi ulaznog vektora). Zatim na osnovu procenjene verovatnoće p daje izlaz 1, ako je verovatnoća veća od 0.5 odnosno 0, ako je verovatnoća manja od 0.5. Procena verovatnoće p se često obelažava sa $\hat{p}$.
Potrebno je naći vezu između nezavisnih promenljivih na ulazu i Bernulijeve raspodele. Ta veza se predstavlja *logit* funkcijom. Logit funkcija mapira linearnu kombinaciju nezavisno promenljivih na ulazu na Bernulijevu raspodelu u domenu od 0 do 1. Logit funkcija predstavlja prirodni logaritam odnosa šansi.

$$\textit{logit(p)} = \textit{ln}\left(\frac{p}{\textit{1-p}}\right) \tag{2.11}$$



Slika 2.1: Logit funkcija

Logit funkcija je data na slici 2.1. Domen funkcije je interval $(0, +1)$, i definisana je svuda sem u 0 i 1. Kodomen funkcije je interval $(-\infty, +\infty)$. Pošto je potrebno da nama argumenti mogu da uzmu bilo koju realnu vrednost, a da izlaz bude u intervalu između 0 i 1, potražićemo inverznu funkciju ovoj. Ta funkcija se naziva *logistička* funkcija. Izvođenje je dato u jednačini 2.12.



h

Slika 2.2: Logistička funkcija

$$
\begin{aligned}
ln\left(\frac{p}{\textit{1-p}}\right) &= \alpha, \alpha \in \mathbb{R} \\
\frac{p}{1-p} &= e^{\alpha} \\
p &= e^{\alpha}(1-p) \\
p + pe^{\alpha} - e^{\alpha} &= 0 \\
p(1+e^{\alpha}) - e^{\alpha} &= 0 \\
p &= \frac{e^{\alpha}}{1+e^{\alpha}}
\end{aligned}
\tag{2.12}
$$

$$
logit^{-1}(\alpha) = \frac{e^{\alpha}}{1+e^{\alpha}} = \frac{1}{1+e^{-\alpha}} \tag{2.13}
$$

Grafik logističke funkcije dat je na slici 2.2. Vidimo da je sada domen funkcije $(-\infty, +\infty)$, a kodomen $(0, +1)$. Ovakva funkcija se naziva *sigmoidnom* funkcijom (takođe postoji naziv i „S”-kriva) i ona daje broj uvek između 0 i 1.
Kada logistička regresija nađe verovatnoću $\hat{p}$ kao funkciju ulaznih parametara $\mathbf{x}^{\mathbf{T}}\ \theta$, gde je $\mathbf{x} = \mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{n}$ a $\theta = \theta_1, \theta_2, ..., \theta_\mathrm{n}$, ona na osnovu te verovatnoće može da da predikciju da li taj podatak pripada pozitivnoj klasi.

$$
\hat{y} = \begin{cases} 1, & \hat{p} > 0.5 \\ 0, & \hat{p} < 0.5 \end{cases} \tag{2.14}
$$

2.3 Softmaks regresija

asfsdfasffdafsdfasdsdfsdfsdfasdfasdfdfsaasdfdfas asd f sdf sdf asdf sdfa asdf asdfasdfsdfsdfasdfasdfasdfsdfsdf

Glava 3

Implementacija

3.1 Logistička regresija

3.2 Softmaks regresija

Glava 4

Analiza rezultata

Glava 5

Primena kod neuronskih mreža