Matematički fakultet, Beograd

Klasifikacija životinja na osnovu određenih atributa upotrebom neuronskih mreža

Marija Filipović, Đorđe Dimović

Februar 2019

SADRŽAJ

UVOD

Veštačke neuronske mreže predstavljaju sisteme sačinjene od međusobno povezanih jedinica, neurona, nastalih po uzoru na biološki neuron, koji šalju poruke jedni drugima. Neuroni unutar mreže su povezani granama. Svaka grana ima svoju težinu izraženu kao brojčanu vrednost. Grane između neurona su podložne promenama u zavisnosti od stečenog iskustva tokom učenja mreže, što veštačke neuronske mreže čini sposobnim za adaptacije i "učenje". Mreža se sastoji od ulaznog I izlaznog sloja, između kojih može biti nijedan, jedan ili više skrivenih slojeva. Kada mreža uči podaci ulaze u mrežu putem ulaznog sloja, on pobuđuje slojeve skrivenih jedinica, da bi potom stigli do izlaznog sloja. Međutim, kako bi mreža naučila da klasifikuje I prepoznaje pojmove, ona mora dobiti povratnu informaciju. Povratna informacija je rezultat poređenja dobijenog ishoda sa željenim ishodom, I ta novonaučena informacija doprinosi da se u sledećoj iteraciji dobijeni ishod približi željenom. Ova promena ishoda se ostvaruje menjanjem težine grana, što omogućava ceo fenomen "učenja" mreže. Na osnovu ovoga možemo primetiti da se veštačka neuronska mreža može koristiti za klasifikaciju podataka, I štaviše, da u njoj ima veliku primenu.

Klasifikacija podataka spada u tehnike nadgledanog učenja, koja za ulazni podatak ima skup slogova, gde je svaki slog sačinjen od skupa atributa i jednog specijalnog atributa koji predstavlja oznaku klase. Potebno je naći klasifikacioni model koji preslikava svaki skup atributa u jednu od predefinisanih oznaka klasa. Ovaj model nastaje u toku procesa treniranja, dok se ispravnost modela odnosno provera njegove efikasnosti nad podacima koji mu nisu prethodno poznati vrsi u fazi testiranja. Da bi se ovo omugućilo ulazni skup podataka se deli na dva dela, jedan za trening, koji je po pravilu veći i sačinjen od otrpilike 80 procenata ulaznog skupa, i jedan za testiranje koji se sastoji od ostatka podataka iz početnog skupa. Cilj je dodeliti slogove koji nisu prethodno poznati što je preciznije moguće jednoj od klasa.

Kada se mreža istrenira korišćenjem podataka iz trening skupa, uvode se novi podaci, kod kojih nedostaje atribut po kome se vrši klasifikacija. Ovim se postiže da mreža na osnovu prethodno stečenog znanja treniranjem može u fazi testiranja da zaključuje koje instance pripadaju kojoj klasi, bez prethodnog znanja o tim instancama. Mreža „zapamti" koje su osobine odnosno vrednosti atributa karakteristične za određene klase u fazi treninga, što je čini sposobnom da u fazi testiranja prepozna iste osobine kod novih instanci i u skladu sa tim odredi klasu. Uspešnost učenja mreže zavisi od mnogih faktora, neki se tiču samih parametara mreže, a neki čak i baze nad kojom mreža uči. Kombinovanjem odgovarajućih postavki može se postići velika efikasnost mreže pri klasifikaciji.

Danas veštačke neuronske mreže imaju širok domen primene u različitim naukama. Mogu se koristiti za prepoznavanje oblika, rukopisa, govora, za predviđanja kretanja cena na tržištu, kompresovanje podataka, kriminološka istraživanja, psihijatrijske procene, analizu medicinskih testova itd. Budući da je veštačka neuronska mreža sposobna da na osnovu prisustva ili odsustva nekih osobina izraženih atributima odluči da li instanca pripada ili ne pripada nekoj klasi, ona se može primenjivati u medicini, kako bi se na osnovu prisustva simptoma utvrdilo postojanje bolesti. Jednu takvu primenu možemo da prikažemo narednim primerom.

Zdravlje srca se može proučavati pomoću EKG snimaka ili merenjem srčanog pulsa, na osnovu kojih bi se mogli primetiti indikatori koji sinaliziraju prisustvo određenih bolesti srca. Međutim, ovakvi signali nisu statični, oni se mogu javiti u bilo kom vremenskom periodu, zbog čega bi trebalo proučavanje EKG

rezultata i praćenje varijacija srčanog pulsa bolesnika vršiti u višečasovnom periodu. Varijacije srčanog pulsa, koje mogu biti indikator bolesti ili upozorenje za moguć razvitak srčane bolesti, mogu biti prisutne sve vreme tokom tokom jednog dana ili se naprasno javiti u nekom trenutku tokom dana. Danas se promena srčanog ritma koristi kao osnovni signal za podelu abnormalija srčanog rada u osam klasa. Tumačenje pojava abnormalija iskrslih u određenom periodu tokom par sati u velikom skupu podataka može biti vrlo naporan posao. Međutim, upotreba veštačke neuronske mreže za klasifikaciju ovakvih podataka može biti jako korisno u dijagnostici. Mreža korišćena za klasifikaciju ima pored standardna dva sloja – ulaznog i izlaznog, i dva skrivena sloja, a težine grana su inicijalizovane na slučajan način, a tokom procesa učenja menjanje su radi dobijanja što preciznijeg rezultata. Za učenje mreže korišćen je metod propagacije unazad (Back Propagation Algotihm, BPA). Kao ulazni podaci koriste se tri parametra dobijena iz merenja srčanog pulsa. Ucenje mreže se vršilo pomoću standardne sigmoidne aktivacione funkcije. Izlazni sloj sastoji se od tri neurona s vrednostima jednakim 0 ili 1, to je omogućilo dobijanje 8 kombinacija od 000 do 111, koje predstavljaju kodove za osam željenih klasa. Rezultati primene pokazuju veliki stepen efikasnosti klasifikatora, sa nivoom tačnosti od 80-85%. [1]

Primećujemo da je primena veštačke neuronske mreže kao klasifikatora radi dobijanje zaključaka iz datih podataka jako praktična, a pokazala se i kao vrlo efikasna. Našim projektom želimo da pokažemo učinak veštačke neuronske mreže u klasifikaciji podataka iz baze sa pretežno brojčanim atributima.

OPIS I REŠENJE PROBLEMA

Opis baze i cilja projekta

Projekat se bavi klasifilacijom podataka pomoću vestačke neuronske mreže iz baze podataka koja sadrži atribute kategoričkog i numeričkog tipa. Baza sadrži informacije o životinjskim vrstama, njihove nazive i osobine, i najzad nazive životinjskih grupa kojima pripadaju. Cilj je klasifikovati životinjske vrste u njima odgovarajuće grupe na osnovu prisustva ili odsustva navedenih osobina. Atribut kategoričkog tipa predstavlja naziv vrste, dok je ciljni atribut po kome klasifikujemo numerički, pri čemu svaki broj označava po jednu grupu životinja. Svi ostali atributi predstavljaju osobine životinja, i mogu imati vrednosti isključivo 1 ili 0, pri čemu je 1 za prisustvo, a 0 za odsustvo date osobine. Pored ovakvih atributa koji predstavljaju osobine, baza sadrži jedan atribut koji označava broj nogu životinje, i može uzimati vrednosti 0, 2, 4, 5, 6 ili 8. Isprobavajući različite načine pretprocesiranja podataka, ovaj atribut ćemo kasnije podeliti na 6 novih atributa, gde će za svaku životinjsku vrstu jedan od njih moći da uzme vrednost 1, dok će ostali biti 0. Ovu promenu atributa vršimo kako bi nam podaci s kojima mreža radi bili usaglašeni, a i u svrhu eksperimentisanja i praćenja dobijenih rezultata pri ovakvoj promeni.

Rešenje problema

Projekat je rađen u programskom jeziku Python 2.7. Neke od ključnih biblioteka koje su korišćene: pandas – za rad sa csv fajlovima, sklearn – za pretprocesiranje podataka, matplotlib – za vizuelizaciju podataka i biblioteka keras za rad sa veštačkom neuronskom mrežom.

Pretprocesiranje

Prvi korak algoritma jeste pretprocesiranje podataka. Podaci koje očekujemo na izlazu iz mreže su numeričkog tipa, i uzimaju vrednosti iz intervala [1, 7]. Međutim, umesto da imamo jedan neuron u izlaznom sloju koji kao izlaz ima jedan ceo broj iz ovog intervala, pogodnije je da u izlaznom sloju postoji 7 neurona, pri čemu će samo jedan moći da ima izlaznu vrednost 1, a svi ostali vrednost 0. Svaki od ovih 7 neurona simbolizuje po jednu klasu životinja, i u zavisnosti od toga koji neuron uzima vrednost 1 određujemo klasu životinje. Ovo se ostvaruje korišćenjem one-hot-encoding metode.

Atribut koji označava ime životinje nije nam potreban za učenje mreže zbog toga što je jedinstveno za svaku vrstu, te ga nismo ni koristili u ovoj fazi. Naknadno ćemo ga korisititi u svrsi vizuelizacije.

Ulazni skup smo podelili na trening i test skup, veličina 70 i 30 procenata polaznog skupa.

```
def preprocess(dataset):
    y = dataset.iloc[:, -1].values

    onehotencoder = OneHotEncoder()
    y = onehotencoder.fit_transform(y.reshape(-1, 1)).toarray()
    y = np.asarray(y, dtype=int)

    X = dataset.iloc[:, :17].values

    return X, y
```

Slika 1. – funkcija pretprocesiranja

Učenje mreže

Nakon pretprocesiranja, pomoću funkcija biblioteke keras konstruišemo mrežu. Mreža ima dva sloja, ulazni i izlazni, zbog toga što ovakva mreža daje dovoljno dobre rezultate u našem slučaju. Ulazni sloj ima 16 ulaza, za 16 osobina životinja. Broj izlaza iz ovog sloja je 7, što je istovremeno i broj ulaza za izlazni sloj. Rekli smo da je broj izlaza iz mreze, tj. iz izlaznog sloja takođe 7. Inicijalne vrednosti težina grana su postavljene na nasumičan način. Kao aktivaciona funkcija ulaznog sloja korišćena je RelU (Rectified Linear Unit) funkcija, $f(x) = \max(0, x)$, koja nam odgovara zbog toga što u bazi nemamo negativne vrednosti koje bi bile zanemarene korišćenjem ove funkcije. Takođe, pri testiranju se pokazalo da ovakva postavka daje

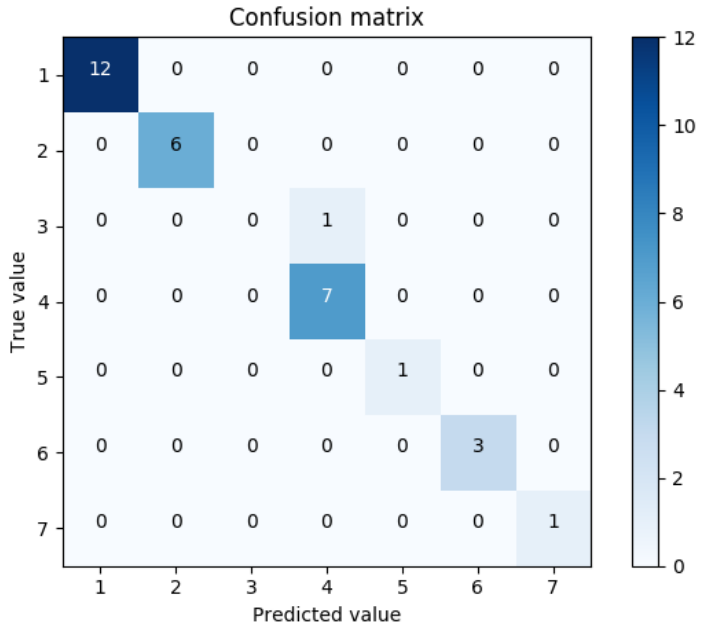
dobre rezultate. Aktivaciona funkcija izlaznog sloja je sigmoidna funkcija, $f(x) = 1/(1 + e^{-x})$, koja slika na interval [0, 1].

```
model = Sequential()

model.add(Dense(units=7, init='uniform', activation='relu', input_dim=16))
model.add(Dense(units=7, init='uniform', activation='sigmoid'))

model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(X_train, y_train, batch_size=16, epochs=400)

y_pred_train = model.predict(X_train)
y_pred_test = model.predict(X_test)
```

Slika 2. – instanciranje mreže i postavljanje slojeva

Testiranje i vizuelizacija

Sa prethodno navedenim postavkama, neuronska mreža se pokazala veoma dobro. Najbolja precuznost za trening skup iznosila je 97.14%, dok je preciznost za test skup iznosila 96.77%.



Slika 3. – Matrica konfuzije test podataka

	Names	Predict_Type	Origin_Type	Correct
0	moth	6	6	True
1	termite	6	6	True
2	sole	4	4	True
3	leopard	1	1	True
4	worm	7	7	True
5	squirrel	1	1	True
6	crow	2	2	True
7	haddock	4	4	True
8	swan	2	2	True
9	stingray	4	4	True
10	seal	1	1	True
11	chub	4	4	True
12	girl	1	1	True
13	fruitbat	1	1	True
14	lion	1	1	True
15	catfish	4	4	True
16	raccoon	1	1	True
17	frog	5	5	True
18	kiwi	2	2	True
19	rhea	2	2	True
20	vulture	2	2	True
21	antelope	1	1	True
22	cheetah	1	1	True
23	seasnake	4	3	False
24	wallaby	1	1	True
25	ladybird	6	6	True
26	cavy	1	1	True
27	porpoise	1	1	True
28	herring	4	4	True
29	skimmer	2	2	True
30	tuna	4	4	True

Slika 4. – Podaci iz test skupa, prikaz kako su klasifikovani i njihova stvarna klasa

EKSPERIMENTALNI REZULTATI

Do prethodno navedenih rezultata došli smo eksperimentisanjem sa parametrima funkcija, poput broja epoha, broja slojeva, različitih postavki aktivacionih funkcija, promena veličina test i trening skupa itd. U uvodu smo napomenuli da su svi atributi sa kojima mreža uči numeričkog tipa s mogućim vrednostima 0 ili 1, izuzev jednog. Taj atribut predstavlja broj nogu određene životinjske vrste, te smo odlučili da ga razložimo na 6 atributa za svaku od mogućih različitih vrednosti. Ovo je pretežno promena u pretprocesiranju podataka. Međutim ove izmene nisu donele značajnu promenu u efikasnosti mreže.

```
def preprocess(dataset):
    y = dataset.iloc[:, -1].values

    onehotencoder = OneHotEncoder()
    y = onehotencoder.fit_transform(y.reshape(-1, 1)).toarray()
    y = np.asarray(y, dtype=int)

    X = dataset.iloc[:, :17].values
    legs = onehotencoder.fit_transform(X[:, 13].reshape(-1, 1)).toarray()
    legs = np.asarray(legs, dtype=int)
    Xnew = np.append(X, legs, axis=1)
    X = np.delete(Xnew, 13, axis=1)

    return X, y
```

Slika 5. – Promena funkije za pretprocesiranje

U procesu razvijanja programa primetili smo da drugačiji izbor aktivacione funkcije donosi promenu u efikasnosti rada mreže. U koliko kao aktivacionu funkciju ulaznog sloja postavimo sigmoidnu funkciju, mreža daje lošije rezultate. Pri ovakvim postavkama preciznost test skupa je pala na oko 80%.

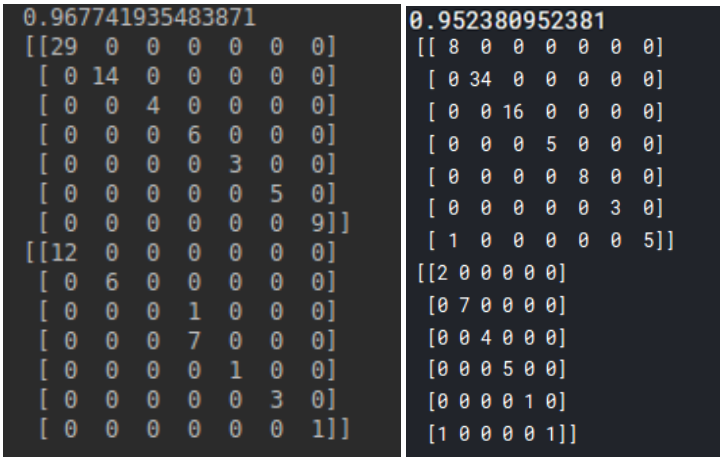
Zaključili smo da mreža daje dobre rezultate sa brojem epoha oko 400. Povećanje broja epoha nam nije od značaja za preciznost rada mreže. Na primer, ukoliko broj epoha postavimo na 2000, tačnost nije veća od tačnosti koja je postignuta sa 400 epoha. Zaključujemo da su dobijeni rezultati sa malim brojem epoha dovoljno dobri po tačnosti, te nema razloga podizati broj epoha. Međutim, povećanje broja epoha na 2000 dovodi do smanjenja greske sa oko 0.5 na 0.0000001. Povećanje epoha je dakle ipak značajno, ali u svrhu smanjenja napravljene greške pri klasifikaciji. Ukoliko smanjimo broj epoha na npr. 300, preciznost klasifikacije opada na oko 65%. Iako je preciznost sa manjim brojem epoha zadovoljavajuća, čestim testiranjem rada mreže primetili smo da sa ovim brojem epoha dobijeni rezultati mogu da variraju, i to ne za male vrednosti. Radi izbegavanja ovakvog nepredividovog ponašanja mreže i veće pouzdanosti bilo bi ipak bolje držati se većeg broja epoha.

Poređenje sa drugim radovima

U potrazi za radovima koji obrađuju slične teme naišli smo na rad koji rešava isti problem. Ova mreža je isprogramirana takotakoše u python-u, međutim drugačije aktivacione funkcije, broj epoha i pretprocesiranje podataka u odnosu na našu mrežu. Korišćene su akrivacione funkcije reLU i softmax. Mreža ima tri sloja sa 21 neuronom u skrivenom sloju. Broj epoha je dvostruko manji od našeg.

Kada uporedimo vreme izvršavanja, uprkos većem broju epoha naša mreža ima mnogo kraće vreme izvršavanja – 5 sekundi u proseku, u poređenju sa datom mrežom čije je vreme izvršavanja oko 68 sekundi. Čak i kada se broj epoha naše mreže znatno poveća – sa 400 na 2000 epoha, vreme izvršavanja iznosi oko 23 sekunde, što je i dalje značajno manje u poređenju sa datom mrežom.

Iako data mreža ima sloj više u odnosu na našu, preciznost je slična. Zaključujemo da je ovako zbog same baze, njenih podataka i malog broja instanci.



Slika 6. – Levo je prikaz preciznosti i matrice konfuzije naše mreže, desno od date mreže

Karakteristike sistema

RAM memorija: 6GB
Procesor: Intel Core i3 M370 2.4GHz x 4
OS: Ubuntu 16.4
Grafičko okruženje: PyCharm 20018.3.4

ZAKLJUČAK

Povećanjem broja epoha i dodavanjem skrivenih slojeva stabilnost mreže je veća – u našem primeru sa 400 epoha i 2 sloja se pokazalo da, iako mreža daje zadovoljavajuće rezultate, njena preciznost klasifikacije može dosta da varira. Dodavanje skrivenih slojeva i više epoha bi moglo da otkloni ovako nepredvidivo ponašanje mreže. Preciznost mreže je već dovoljno velika, te je možda bolje koncentrisati se na unapređenje mreže smanjivanjem greške koju pravi, i rađenjem na njenoj pouzdanosti umesto dodatnog povećanja preciznosti.

LITERATURA

[1] R. Acharya, A. Kumar, P. S. Bhat, C. M. Lim, S. S. Lyengar, N. Kannathal, S. M. Krishnan, Classification of cardiac abnormalities using heart rate signals, https://link.springer.com/article/10.1007/BF02344702

[2] Sava Gavran, Veštačke neuronske mreže u istraživanju podataka: pregled i primena, Master rad, 2016., Univerzitet u Beogradu Matematićki fakultet

[3] Saravanan K, S.Sasithra, Review of classification based on artificial neural networks, International Journal of Ambient Systems and Applications (IJASA) Vol.2, No.4, December 2014

[4] Charu C. Aggarwal, Data Minning, The Textbook, Springer, 2015.