

Fazi neuronska mreža

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Katarina Savičić, Bojana Ristanović
mi16261@alas.matf.bg.ac.rs, mi16045@alas.matf.bg.ac.rs

23. septembar 2020

Sažetak

U ovom radu će biti predstavljena fazi neuronska mreža. Konkretnije, ovaj metod je korišćen kao metod klasifikacije podataka, koji u trening fazi koristi min-max algoritam, a u test fazi koristi k najbližih suseda. Nakon opisa algoritma, prikaza je njegova praktična primena. Na kraju, rezultati su upoređeni sa autorima rada *Understanding Fuzzy Neural Network using code and animation*.

Ključne reči: fazi logika, neuronske mreže, min-max, knn, klasifikacija.

Sadržaj

1	Uvod	2
2	Fazi logika	2
3	Neuronske Mreže	2
4	Fazi min-max klasifikator	2
4.1	Hiperboks (eng. <i>Hyperbox</i>)	2
4.2	Algoritam	3
4.2.1	Faza proširenja	3
4.2.2	Faza kontrakcije	3
4.3	Modifikacije algoritma	3
5	Rezultati	4
6	Zaključak	4
	Literatura	4

1 Uvod

Fazi neuronska mreža (eng. *Fuzzy neural network*) je sistem za učenje koji pronalazi parametre fazi sistema koristeći tehnike aproksimacije neuronskih mreža. To je hibridni inteligentni sistem koji kombinuje tehnike rasuđivanja fazi logike sa tehnikama učenja neuronskih mreža.[2]

Fazi min-max klasifikator (eng. *The fuzzy min-max (FMM)*) je sistem koji formira hiperboksove za klasifikaciju i predviđanje. U ovom radu je pokušana modifikacija FMM-a korišćenjem algoritma k najbližih suseda (eng. *k-nearest neighbors algorithm (k-NN)*) u procesu predviđanja klasa prosleđenih podataka.

2 Fazi logika

Kod klasične logike, promenljive mogu uzeti jedino vrednosti 1 ili 0 (tačno ili netačno). Kod fazi logike se skup vrednosti proširuje, pri čemu se dozvoljava da promenljive mogu uzeti realne vrednosti unutar nekog intervala. Drugim rečima, pretpostavlja se da ne mora sve biti u potpunosti istinito ili u potpunosti neistinito, već se dozvoljava određen nivo delimične istine. Na primer, za nešto možemo reći da je najverovatnije istina, pa odgovarajuća promenljiva može imati vrednost 0.9, a u nešto drugo možemo imati veliku sumnju, ali ostavljati malu verovatnoću da bude istinito, pa odgovarajuća logička promenljiva tada može imati vrednost 0.1. [1]

Da li nešto pripada skupu možemo odrediti pomoću funkcije pripadnosti. Ona kod klasične logike ima vrednost 0 ili 1, zato što nešto može ili ne da pripada skupu. Kod fazi logike ova funkcija ima vrednost između 0 i 1. [1]

3 Neuronske Mreže

4 Fazi min-max klasifikator

Osnovna ideja je da se na osnovu funkcije pripadnosti koja se koristi u Fazi logici može odrediti kojoj klasi pripada instanca koju klasifikujemo. U ovom slučaju instanca je uređeni par realnih brojeva, koji se može predstaviti kao tačka u koordinatnom sistemu. Klasa instance će biti ona za koju funkcija pripadnosti daje najveću vrednost. Postavlja se pitanje kako predstaviti fazi skup.[3]

4.1 Hiperboks (eng. *Hyperbox*)

Fazi skup možemo definisati kao pravougaonik čije dužine stranica mogu uzimati vrednost od 0 do 1. Ovaj skup je takav da ako se tačka nalazi unutar pravouganičara, vrednost njene funkcije pripadnosti je 1. Pravougaonik je definisan svojom najmanjom (donjom levom) i najvećom (gornjom desnom) tačkom. Te tačke ćemo označiti redom slovima v i w .

Vrednost funkcije pripadnosti za svaku tačku koja je van pravouganičara se smanjuje sa povećanjem udaljenosti tačke od pravouganičara. Formula za računanje funkcije pripadnosti nekom pravouganičaru, tj. fazi skupu, je data u nastavku:

$$b_j(A_h) = \frac{1}{2n} \sum_{i=1}^n [\max(0, 1 - \max(0, \gamma \min(1, a_{hi} - w_{ji}))) + \max(0, 1 - \max(0, \gamma \min(1, v_{ji} - a_{hi})))] \quad (1)$$

A_h obrazac za koji računamo funkciju pripadnosti, b_j fazi skup a n dimenzija (u našem slučaju 2). γ je hiperparametar koji se naziva osetljivost ili stopa smanjenja i može se koristiti za kontrolisanje vrednosti funkcije pripadnosti.

Kod neuronske mreže imamo prvi sloj koji predstavlja ulazne čvorove za podatke. Sledeći sloj su čvorovi koji predstavljaju hiperbokseve, tj. svaki čvor predstavlja jedan hiperboks koji pripada nekoj klasi i definisan je dvema tačkama. Hiperboks je prethodno pomenuti pravouganičar. Poslednji sloj čvorova predstavlja izlaz, odnosno klasu. U poslednjem sloju može biti više čvorova, u zavisnosti od broja postojećih klasa. Kada gledamo vrednost klase, svaki čvor koji predstavlja klasu definisan je najvećom vrednošću funkcije pripadnosti od svih funkcija pripadnosti računate za sve hiperbokseve koje pripadaju datoj klasi. Dakle, svaka klasa može imati jedan ili više hiperbokseva, vrednost u čvoru je maksimalna vrednost funkcije pripadnosti razmatranog podatka tim hiperboksevima. Na kraju se porede vrednosti u svim čvorovima koji predstavlja klase i bira se klasa koja ima najveću vrednost. [3]

4.2 Algoritam

Kako bismo sproveli pomenuti način klasifikacije, potrebno je da definišemo hiperbokseve za sve klase. To ćemo uraditi na osnovu skupa za treniranje, odnosno istreniraćemo klasifikator tako što ćemo napraviti hiperbokseve za postojeće klase. Algoritam se sastoji iz dve faze: proširenja i kontrakcije. [3]

4.2.1 Faza proširenja

Imamo instancu X_h iz trening skupa, koja pripada klasi Y . Prvo proverimo sve hiperbokseve koji pripadaju klasi Y i računamo funkciju pripadnosti za svaki. Hiperboks sa najvećom funkcijom pripadnosti je najpogodniji za proširenje. Neka B_j bude jedan takav hiperboks. Pre nego što ga proširimo računamo kriterijum ekspanzije dat sledećom formulom:

$$n\theta >= \sum_{i=1}^n (\max(w_{ji}, x_{hi}) - \min(v_{ji}, x_{hi})) \quad (2)$$

gde je θ hiper parametar koji predstavlja kriterijum proširenja i kontroliše maksimalno proširenje dozvoljeno za hiperboks. Ako zadovoljava kriterijum, hiperboks se širi na sledeći način:

$$v_{ji}^{new} = \min(v_{ji}^{old}, x_{hi}) \quad w_{ji}^{new} = \max(w_{ji}^{old}, x_{hi}) \quad (3)$$

gde su v i w minimalna i maksimalna tačka hiperboksa. Hiperboks se čiri tako da je X_h uključena u region.

Ukoliko kriterijum širenja nije zadovoljen, pravi se novi hiperboks za klasu Y , takav da su mu minimalna i maksimalna tačka jednake i imaju vrednost X_h . Kada proširimo hiperboks, može doći do preklapanja sa drugim hiperboksom. Ovo nije toliko problem ako je u pitanju hiperboks iz iste klase, ali jeste ukoliko je u pitanju hiperboks koji pripada nekoj drugoj klasi. Zbog toga imamo fazu kontrakcije. [3]

4.2.2 Faza kontrakcije

Pronašli smo hiperboks koji smo proširili i sada treba da proverimo da li postoji preklapanje sa drugim hiperboksom. Dozvoljeno je preklapanje između hiperbokseva iste klase, pa treba samo da proverimo za hiperbokseve drugih klasa.

Proveru da li postoji preklapanje vršimo preko tačaka koje određuju hiperbokseve. Gledamo po svim dimenzijama da li postoji preklapanje i beležimo rezultate. Onda tražimo dimenziju po kojoj je preklapanje najmanje. Slučajevi koji pokrivaju proveru preklapanja su dati u kodu. Kada nađemo dimenziju po kojoj je preklapanje minimalno, hoćemo da smanjimo hiperboks po toj dimenziji. Slučajevi koji pokrivaju način smanjenja hiperboksa takođe se nalaze u kodu i neće biti predstavljeni ovde.

Poenta celog algoritma je naći odgovarajući hiperboks, proširiti ga ako je to moguće, ako ne naći drugi koji je moguće proširiti, ili dodati novi, proveriti da li postoji preklapanje i izvršiti kontrakciju ukoliko postoji. [3]

4.3 Modifikacije algoritma

U originalnom algoritmu se za određivanje klase uzima samo najveća vrednost klase, koja predstavlja najveću vrednost funkcije pripadnosti za hiperbokseve te klase. Mi smo algoritam izmenili tako da ne gleda jednu najveću vrednost već uzima u obzir nekoliko vrednosti. Umesto da gledamo funkcije pripadnosti i tražimo najveće vrednosti, gledali smo rastojanje svake tačke od hiperboksa i tražili najmanje rastojanje. Pretpostavili smo da rastojanje od hiperboksa možemo posmatrati kao rastojanje od prave koja prolazi kroz tačke koje određuju hiperboks. Sortirali smo rastojanja od svih hiperbokseva rastuće i primenili metod k najbližih suseda. Metod smo primenili tako što smo uzeli k najmanjih rastojanja, gledali kojim klasama pripadaju hiperboksevi i kao klasu instance uzeli najbrojniju klasu među hiperboksevima.

5 Rezultati

6 Zaključak

Literatura

- [1] Fazi logika. on-line at: <http://poincare.matf.bg.ac.rs/~stefan/ri/3.html>.
- [2] Fuzzy Neural Network. on-line at: http://www.scholarpedia.org/article/Fuzzy_neural_network.
- [3] Min-Max Fuzzy Classifier. on-line at: <https://medium.com/@apbetahouse45/understanding-fuzzy-neural-network-with-code-and-graphs-263d1091d773>.