Vladimir Lunić

Poređenje tačnosti metoda K-najbližih komšija i neuronske mreže na prepoznavanje cifara

Na MNIST skupu podataka rukom napisanih cifara koji, se sastoji iz 60000 slika veličine 28×28 piksela, urađena je analiza glavnih komponenti kako bi se smanjila veličina podataka na 3, 6, 8 ili 10 glavnih komponenti. Nakon toga su ti podaci klasifikovani putem algoritama K-najbližih komšija, i neuronske mreže. Algoritam K-najbližih komšija je imao prosečnu tačnost od 88% sa najmanjom tačnošću od 85% sa 3 komšije i 3 glavne komponente, a najveću tačnost od 92% sa 10 komšija i 10 glavnih komponenti. Neuronska mreža je imala prosečnu tačnost od 26%, pri črmu najmanju tačnost od 21% u varijanti sa tri glavne komponente, 1000 iteracija učenja i 10 neurona u skrivenom sloju, a najveću tačnost od 32% sa 10 komponenti, 10000 iteracija učenja i 20 neurona u skrivenom sloju. U radu se pokazalo da je, nakon redukovanja broja dimenzija putem analize glavnih komponenti, bolje koristit algoritam K-najbližih komšija nego neuronsku mrežu.

Uvod

Problem čitanja rukom napisanih cifara postoji dosta dugo, sortiranje pošte je rađeno ručno, radnici su gledali svako pismo ponaosob i stavljali ga u kutije za odgovarajuće brojeve. Takođe, ovaj problem se javlja i kod veštačke inteligencije, kada želimo da što tačnije znamo koji su brojevi u pitanju. Danas je problem veoma dobro izučen, razrađeni su algoritmi izuzetne efikasnosti, izučavano je sve od linearnih klasifikatora do jako kompleksnih neuronskih mreža (Le Cun *et al*. 1990).

Neuronske mreže se koriste u raznim situacijama – prepoznavanju objekata, prepoznavanju reči, pretvaranju teksta u govor, razumevanju govora i mnogim drugim, pri čemu se koriste različite vrste neuronskih mreža. Algoritam K-najbližih komšija (eng. K-nearest neighbors) koristi se, pored prepoznavanja cifara, i u prepoznavanju dokumenata sa sličnim temama, sistemima za preporuku (eng. recommender systems), gde se prati šta se nekoj osobi sviđa, pa prave preporuke, tražeći slične stvari koje bi se takvoj osobi verovatno svidele.

U ovom istraživanju se upoređivala tačnost neuronske mreže i algoritma K-najbližih komšija (KNK), nakon što je urađena analiza glavnih komponenti (eng. principal component analysis) na skupu slika rukom napisanih cifara iz baze MNIST (Modified National Institute of Standards and Technology database, LeCun *et al*. 1998). MNIST je baza podataka rukom napisanih cifara koja se sastoji iz 60000 cifara za vežbanje i 10000 za testiranje. Podaci dolaze u obliku matrica veličine 28×28 piksela, gde je svaka vrednost od 784 piksela broj između 0 i 255. Na skupu za vežbanje urađena je analiza glavnih komponenti (AGK u daljem tekstu). AGK se, između ostalog, koristi za prepoznavanje lica (Turk i Pentland 1991), tako što analizira komponente i uzima najvažnije, glavne, komponente, praveći „prosečno lice" u odnosu na koje se dalje porede druga lica. Na sličan način, AGK koji je rađen nad skupon cifara za vežbanje dao je „prosečne cifre" koje se dalje koriste u neuronskoj mreži i algoritmu K-najbližih komšija.

Svi programi su pisani u programskom jeziku Python 3.6.

Vladimir Lunić (2000), Novi Sad, Vladike Ćirića 28/8, učenik 4. razreda Gimnazije „Isidora Sekulić" u Novom Sadu

MENTOR: Stefan Nožinić, softverski inženjer, Šabac

Metode

Analiza glavnih komponenti. Analiza glavnih komponenti (AGK) je postupak u statistici koji ortogonalno transformiše podatke koji su delom korelisani u nekorelisane, odnosno linearno nezavisne vektore. Dobijeni nekorelisani vekori nazivaju se glavne komponente.

Za svaku sliku, matrice veličine 28×28 su pretvorene u vektore 1×784 tako što je svaka vrednost početne matrice dodavana na praznu listu, praveći željeni vektor. Nakon toga određena je prosečna vrednost svakog piksela čime je dobijen vektor proseka, odnosno vektor koji ima prosečne vrednosti svih piksela:

$$\mu = \frac{\sum_{k=1}^{n} a_k}{n}$$

gde je u našem slučaju $n = 784$. Dalje oduzimamo vrednost prosečnog piksela od svakog piksela da bismo centrirali sve vrednosti oko njega:

$$d_k = a_k - \mu$$

čime se dobija matrica centriranih piksela. Množenjem matrice centriranih piksela sa transponovanim oblikom sebe dobija se matrica rasejanja. Rešavanjem svojstvenog problema ove matrice dobijaju se njene svojstvene vrednosti i svojstveni vektori. Dobijene svojstvene vrednosti sortiramo od najveće do najmanje. Sortiramo i svojstvene vektore prema svojstvenij vrednosti kojoj odgovaraju. Na osnovu prvih N najvećih svojstvenih vrednosti, bira se N njima pripadajućih svojstvenih vektora koji čine novu ortonormiranu bazu, čime se smanjuje veličina početne matrice sa 60000×784 na 60000×N, gde je N broj manji od 784, i koji sami biramo u zavisnosti od toga koliko želimo da smanjimo veličinu početnih podataka.

Algoritam K-najbližih komšija. K-najbližih komšija (KNK) je algoritam koji se koristi i za klasifikaciju i za regresiju (Peterson 2009). U oba slučaja, podaci se nalaze u $\mathbb{R}^N$ metričkom prostoru. U ovom radu udaljenost se merila euklidskom distancom, a algoritam gleda K najbližih suseda, i u odnosu na njih pravi klasifikaciju, ili određuje granice između objekata u prostoru.

Za prepoznavanje rukom napisanih cifara KNK se koristi kao klasifikacioni algoritam. Nakon AGK, slike iz skupa za obuku su projektovane u N-dimenzionalni prostor (gde je N broj koji se bira tokom AGK). Cifra se klasifikuje tako što se svaki vektor, koji je slika cifre samo redukovane dimenzije, iz skupa pomnoži sa glavnim komponentama dobijenim iz AGK i odredi joj se euklidska distanca u odnosu na sve ostale slike iz skupa za vežbanje. Euklidske distance test slike u odnosu na slike za vežbanje se poređaju po veličini u rastućem nizu. U tom nizu se onda odredi K vektora sa najmanjim distancama od date slike pomoću kojih će se odrediti cifra. Dalje se gleda koliko puta se pojavljuje koja cifra u tih K najmanjih distanci i ona cifra koja se pojavljuje najčešće će biti dodeljena test slici. Na kraju se određuje tačnost algoritma, koja se računa tako što se broj tačno pogođenih cifara podeli sa ukupnim brojem test cifara.

Neuronska mreža. Neuronske mreže (NM) su mreže slične mrežama neurona koje se nalaze u ljudskom mozgu, takve gde jedan neuron istovremeno prima podatke iz neurona koji se nalaze iza njega i prosleđuje informacije sledećem neuronu ili neuronima (Le Cun *et al*. 1990). Neuronske mreže (NM) se obično koriste za klasifikaciju objekata, na primer da li je nešto drvo ili ne. NM to rade tako što imaju skup podataka za vežbanje od kojih „uče" da klasifikuju objekte menjajući svakim ciklusom učenja parametre pomoću kojih prave klasifikaciju. Svaka NM ima najmanje dva sloja, sloj za unos u koji ulaze podaci, i sloj za izalz u kojem ona pravi svoje nagađanje, između njih se mogu nalaziti skriveni slojevi koji doprinose većoj preciznosti mreže. Između slojeva se nalaze sinapse, tj. veze između prethodnog i trenutnog sloja neurona. Svaki neuron ima svoju aktivacionu funkciju i funkciju prenosa kao i svoju težinu i svoju sklonost da bude aktiviran ili ne. Pošto svaki neuron u mreži ima vezu sa svakim iz prethodnog sloja, u njega ulaze sve vrednosti neurona tog sloja pomnožene njihovim težinama. Suma vrednosti tih veza koje dolaze iz prethodnog sloja uz dodavanja naklonosti naziva se prenosna ili transfer funkcija. Vrednost transfer funkcije se dalje pušta kroz aktivacionu funkciju koja smanjuje vrednost aktivacione funkcije na manji interval i to onda postaje nova vrednost neurona. Sve tako dok ne dođe do sloja za izlaz, gde onda ispisuje svoju pretpostavku. To se naziva protok unapred (engl. feed forward), i tako NM pravi svoje klasifikacije za objekte. NM uči tako što projektuje unazad, tj. radi bekpropagaciju (backward prop-

agation): gleda koliko je pogrešila na svom sloju za izlaz i onda podešava težine neurona i njihovu naklonost u odnosu na to.

NM u ovom istraživanju ima onoliko ulaznih neurona koliko se glavnih vektora odabralo. Sloj za izlaz ima 10 neurona, tako da ako je upaljen 5. neuron, to znači da je NM klasifikovala podatak kao da je cifra 4. NM takođe ima i jedan skriveni sloj sa različitim brojem neurona. Aktivaciona funkcija korišćena na skrivenom sloju je ReLU (Aleksander i Morton 1990), a aktivaciona funkcija korišćena za izlazni sloj je sigmoidna funkcija (Shamra 2017). Funkcija greške je srednja kvadratna funkcija greške (Levinson 1946). Neuronska mreža je iterativni algoritam, što znači da mora da prođe više puta kroz svoj algoritam, da bi što bolje naučila, i postigla što veću tačnost. Ponavljanjem algoritma neuronska mreža uči tako što radi gradijentni spust (gradient descent) u N-dimenzionalnom prostoru, u svakoj iteraciji tražeći minimum unutar N-dimenzionalnog prostora. Pomoću gradijentnog spusta i koeficijenta učenja, algoritam podešava težine i naklonosti neurona ka upaljenosti i ugašenosti. Broj iteracija za učenje je varirao između 1000 i 10000, a stopa učenja je 0.0001. Za bekpropagaciju je korišćen serijski gradijentni spust (batch gradient descent, Ruder 2016).

Rezultati i diskusija

Baza podataka na kojoj je rađeno prepoznavanje je MNIST, baza rukom napisanih cifara koji se sastoji iz 60000 zapisa. Kod oba metoda, KNK i neuronske mreže, kao prvi parametar koji se menjao (i od kojeg zavisi tačnost) bio je broj glavnih komponenti, broj komponenti koji je proučavan je 3, 6, 8 i 10 komponenti, gde uz povećanje komponenti se povećavala i tačnost oba algoritma. Zatim se za algoritam KNK proveravala tačnost sa 3, 5 i 10 komšija, gde uz povećanje broja komšija se povećavala i tačnost. Dok se za neuronsku mrežu proveravala tačnost na 1000, 2000, 5000 i 10000 iteracija učenja. Takođe je proveravana tačnost sa 10, 15 i 20 neurona u skrivenom sloju. Uz povećanje oba se povećavala i tačnost neuronske mreže.

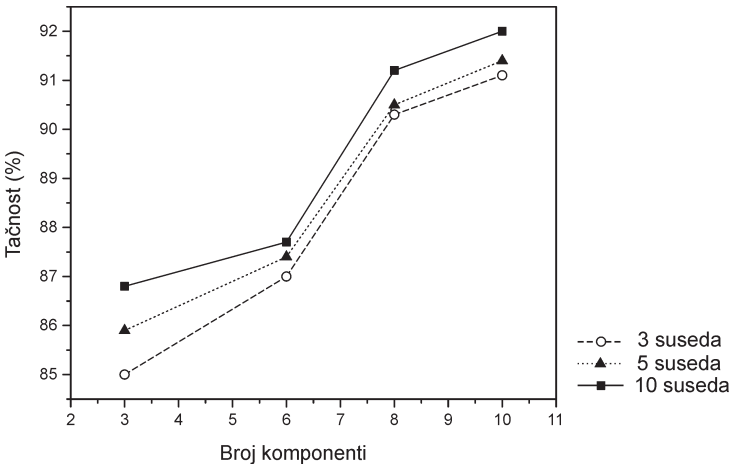
Tačnost algoritma KNK je rasla sa porastom broja komšija i broja glavnih komponenti (slika 1) dobijajući maksimalnu proverenu tačnost od 92%. Tačnost se verovatno povećava sa povećanjem broja komponenti i komšija.

Tačnost neuronske mreže se takođe povećava sa povećanjem komponenti, iteracija učenja i broja neurona u skrivenom sloju (slika 2). Maksimalna proverena tačnost neuronske mreže iznosila je 32%, posle čega je počela da opada opet zbog overfitovanja modela. Prosek neuronske mreže je iznosio približno 26%.

Zaključak

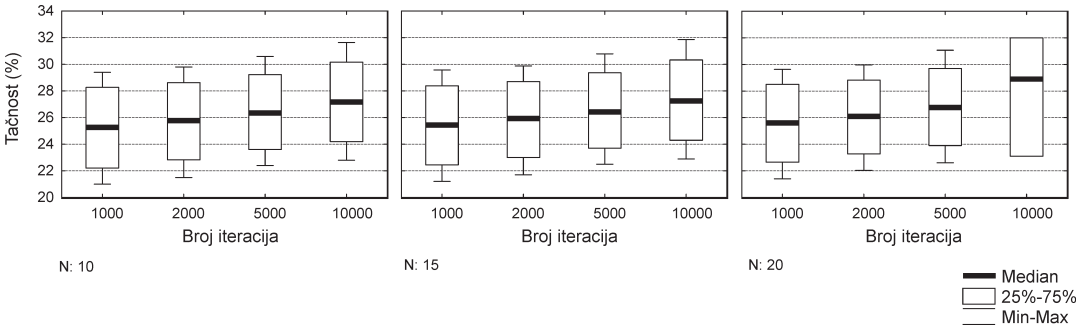
Kao što se može videti iz rezultata, algoritam K-najbližih komšija je pokazao veću tačnost u odnosu na neuronsku mrežu. Iz toga se može zaključiti da uz analizu glavnih komponenti kao pretprocesiranje je efikasnije i tačnije da se koristi algoritam K-najbližih komšija.

Na bazi podataka umesto samo AGK, moglo bi se raditi uklanjanje šuma i ispravljanje za bolji



Slika 1. Zavisnost tačnosti algoritma KNK od broja glavnih komponenti kada *K* iznosi 3, 5 i 10

Figure 1. Accuracy of the KNN algorithm with respect to principal components for *K* equaling 3, 5 and 10



Slika 2. Tačnost prepoznavanja neuronske mreže u zavisnosti od broja iteracija za 10, 15 i 20 neurona u skrivenom sloju

Figure 2. Accuracy of the Neural network algorithm with respect to the number of learning iterations for 10, 15 and 20 hidden neurons

izgled slike, pa tek onda AGK za bolje glavnih komponente. Neuronska mreža može da bude sa većim brojem skrivenih slojeva i većim brojem neurona u skrivenim slojevima, a za veću tačnost treba probati i sa konvolucionim neuronskim mrežama.

Literatura

Aleksander I., Morton H. 1990. *An introduction to neural computing*. New York: Van Nostrand Reinhold

Le Cun Y., Bottou L., Bengio Y., Haffner P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86** (11): 2278.

Le Cun Y., Boser B. E., Denker J. S., Henderson D., Howard R. E., Hubbard W. E., Jackel L. D. 1990. Handwritten digit recognition with a back-propagation network. U *Advances in Neural Information Processing Systems, NIPS 1989* (ur. D. Touretzky). Denver: Morgan Kaufmann, str. 396-404.

Levinson N. 1946. The Wiener (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics*, **25** (1-4): 261.

Peterson L. E. 2009. K-nearest neighbor. *Scholarpedia*, **4** (2): 1883.

Ruder S. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747v2 [cs.LG]. Dostupno na: https://arxiv.org/pdf/1609.04747.pdf

Shamra S. 2017. Activation functions in neural networks. Dostupno na: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Turk M., Pentland A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, **3** (1): 71.

Vladimir Lunić

K-Nearest Neighbors and Neural Network Accuracy Comparison on Digit Recognition

The MNIST dataset of handwritten digits consists of 60000 pictures in which every picture is made up of 784 pixels in the shape of a 28×28 matrix. The principal component analysis (hereafter PCA) algorithm was done as preprocessing on the images. After PCA, the largest 3, 5 and 10 principal components were taken for further use. Every digit from the dataset was then multiplied by the principal components, giving us less dimensions per picture. The K-nearest Neighbors algorithm (hereafter KNN) was then applied on the newly made dataset. The KNN classifier works by calculating the Euclidean distance in N-dimensional space between the new point in the N-dimensional space (which is the picture of the digit we want to classify multiplied by the

principal vectors) and all the other points (the pictures from the dataset also multiplied by the principal vectors). The picture is classified by a majority vote of its neighbors (the *K* shortest distances), with the picture being assigned to the digit most common among its K-nearest neighbors. The accuracy of the KNN algorithm was in the range between 85 and 92 percent.

Lastly, the dataset, also multiplied by the principal components, was classified by a neural network. The neural network consisted of an input layer made up of *N* neurons (where *N* is the number of principal components), a hidden layer of varying depth and varying amount of neurons and lastly an output layer consisting of 10 neurons (one for each digit). The accuracy of the neural network was between 21 and 31 percent, showing that the KNN algorithm has a better accuracy by a margin of 60 percent for this type of preprocessing.