Strojno učenje prepoznavanja rukopisne numeracije

Slota, Marko

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:131:837566

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-23**

Filozofski fakultet
Sveučilišta u Zagrebu

Repository / Repozitorij:

ODRAZ - open repository of the University of Zagreb
Faculty of Humanities and Social Sciences

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2019./2020.

Marko Slota

Strojno učenje prepoznavanja rukopisne numeracije

Završni rad

Mentori:
dr. sc. Ivan Dunđer
prof. dr. sc. Sanja Seljan

Zagreb, lipanj 2020.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

_____

(potpis)

Sadržaj

1. Uvod 1
2. Pristupi strojnom učenju 2
2.1. Algoritmi za učenje 2
2.1.1. Nadzirano učenje 2
2.1.2. Propagacija unatrag 3
2.1.3. Nenadzirano učenje 5
2.1.4. Polu-nadzirano učenje 5
2.2. Algoritmi sličnosti 6
2.2.1. Algoritmi regresije 6
2.2.2. Algoritmi stabla odlučivanja 7
2.2.3. Metoda KNN 8
2.2.4. Umjetne neuronske mreže 11
2.2.5. Konvolucijske neuronske mreže 12
2.2.6. Stroj potpornih vektora 14
2.3. Pojačano učenje 16
3. Problemi pri raspoznavanju znamenki 17
4. Istraživanje 21
4.1. Primjer softverskog rješenja 21
4.2. Rezultati 27
4.2.1. Prvi rezultat 27
4.2.2. Drugi rezultat 28
4.3. Diskusija 29
5. Zaključak 31
6. Literatura 32
Popis slika 34

Popis tablica 35

Sažetak 36

Summary 37

1. Uvod

Termin strojnog učenja podrazumijeva skupove podataka i računalnih algoritama čija se esencijalna ideja temelji na sabiranju i procesiranju što veće količine pojedinih podataka u svrhu mogućnosti razvitka umjetne inteligencije tako da program samostalno može učiti.

U ovome će radu biti pokrivena tematika o različitim oblicima pristupa strojnome učenju, a naglasak će biti na algoritmima čija je zadaća prepoznavanje rukopisne numeracije sa što većim postotkom točnosti bez obzira na to o kakvome se načinu pisanja znamenaka radi. Cilj ovoga rada je prikazati zašto je primjena pojedinih algoritama za učenje i algoritama sličnosti pogodna za rješenje problema vezanih uz tematiku strojnog učenja prepoznavanja rukopisne numeracije. Tematika prepoznavanja i stvaranja samih predikcija pri radu s ručno napisanim oblicima brojčanih znamenaka smatra se dobro pokrivenim područjem ove znanosti. Sama znanost zasnovana je na strojnom učenju, pa tako i dubokom učenju iako postoje i dalje nerješivi oblici problema koji se često pojavljuju, te koji će također biti opisani u ovome radu. Glavni je cilj ovoga rada ukazati na postojanje više različitih vrsta algoritama i metoda koje se mogu koristiti u obliku klasifikatora kada se radi o modelima na kojima se zasniva čitava kompleksnost tematike strojnog učenja, pa tako i prepoznavanja rukopisne numeracije.

Rad je podijeljen u dva osnovna dijela: teorijski i praktični. U teorijskom dijelu analizirani su pristup strojnom učenju i problem u raspoznavanju znamenki. U praktičnom dijelu prikazano je i analizirano programsko rješenje u Pythonu za prepoznavanje rukopisne numeracije kroz odabranu metodu strojnog učenja. Kroz poglavlja i potpoglavlja koja doprinose daljnjem grananju navedenih tipova algoritama prikazan je način rada tih algoritama. Vidljivi su i rezultati točnosti i pogreške u obliku postotka za svaki algoritam nad kojim je vršeno istraživanje ovoga tipa. U završnom dijelu rada prikazan je praktičan primjer softverskog rješenja ove problematike u obliku koda napisanog u programskom jeziku Python uz detaljna objašnjenja istog popraćena stvarnim prikazima u obliku slikovitog izričaja izlaznih podataka. Na kraju slijede zaključak, popis literature, popis slika i tablica te sažeci s ključnim riječima na hrvatskom i engleskom.

2. Pristupi strojnom učenju

Uzimajući u obzir samu prirodu problema prepoznavanja rukopisne numeracije čiji je cilj biti riješen putem uporabe strojnog učenja, algoritme strojnog učenja po Rokandu (2019)[1] dijeli se na:

- *algoritme za učenje (engl. Learning algorithms)*,
- *algoritme sličnosti (engl. Similarity algorithms)*.

Algoritmi za učenje (*engl. learning algorithms*) još se dijele na one za nadzirano učenje (*engl. supervised learning*), za nenadzirano učenje (*engl. unsupervised learning*), te za polunadzirano učenje (*engl. semi-supervised learning*). Algoritmi sličnosti (*engl. similarity algorithms*) se pak još granaju na algoritme regresije (*engl. regression algorithms*), algoritme stabla odlučivanja (*engl. decision tree algorithms*), umjetne neuronske mreže (*engl. artificial neural networks*) i stroj potpornih vektora (*engl. support vector machine*). Valja također navesti i podržano/pojačano učenje (*engl. reinforcement learning*) koje je zapravo *specijalizirano potpolje strojnog učenja, koje je poznato kao približno dinamičko programiranje ili neuro-dinamičko programiranje.*[2]

2.1. Algoritmi za učenje

Osnovni algoritmi koji se koriste u sustavima umjetne inteligencije upravo su algoritmi za učenje (*engl. learning algorithms*). Njihova je primarna uloga prilagodba na samu prirodu problema i njegovo okruženje. Algoritmima za učenje smatrani su algoritmi koji imaju mogućnost samostalnog stvaranja obrazaca na temelju podataka nad kojima ih se trenira te zatim i učenja.

2.1.1. Nadzirano učenje

Polazišna točka nadziranog strojnog učenja (*engl. supervised learning*) jest učenje na principu funkcije koja je prikladna u odnosu na zadani problem. *Funkcija uzima ulazni podatak (X) i*

[1] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[2] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)

preslikava ga na izlazni podatak (Y)[3], što dovodi do postavljanja jednadžbe Y=f(X). Tada se događa razdvajanje na dva seta podataka u set koji se koristi za treniranje i testni set. Ulazni set podataka je označen i koristi se kao oblik pripreme za matematički model koji služi kasnije u postupcima i procesima treniranja. Taj stvoreni matematički model nadalje analizira testne podatke te je zadužen za stvaranje izvedenih funkcija. Jedan od primjera nadziranog strojnog učenja jest primjer strojnog prevođenja prikazan u Dunđer (2015)[4].

Problemi nadziranog učenja mogu se kasnije implementirati i grupirati kao problemi klasifikacije i regresije. U problemima klasifikacije, izlazni podaci kategorizirani su u posebne grupe, dok su kod regresijskih problema izlazni podaci stvarna vrijednost; „*Supervised learning problems can be further grouped into classification problems and regression problems. In a classification problem, an output is categorized into a specific group, whereas in the regression problem an output is a real value.*“[5]

Prema Rokandu (2019)[6] vidljivo je kako je novonastale oblike problema kod nadziranog strojnog učenja moguće svrstati pod klasifikacijske probleme i probleme regresije, gdje je kod klasifikacijskih problema izlazni podatak kategoriziran u oblik specifične grupe, dok je u području problema regresije izlazni podatak zapravo stvarna vrijednost. Algoritmi čija je svrha rješavanje problema koji se vežu uz područje nadziranog strojnog učenja su *Support Vector Machines, Linear Regression, Logistic Regression, Naive Bayes, Decision Trees, K — Nearest Neighbour i Neural Networks.*[7]

2.1.2. Propagacija unatrag

Algoritam koji se koristi u strojnom učenju u obliku neuronskih mreža pri nadziranom strojnom učenju jest algoritam propagacije pogreške unatrag, poznati i kao algoritam povratne

---

[3] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[4] Dunđer, I. Sustav za statističko strojno prevođenje i računalna adaptacija domene. Doktorska disertacija. Sveučilište u Zagrebu, 2015.
[5] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[6] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[7] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)

propagacije, algoritam unatražnog rasprostiranja ili unatražnog širenja (*engl. backpropagation*). Algoritam propagacije pogreške unatrag, nazivan i skraćeno *backprop,* koristi podatke o trošku koji mogu biti obrađeni u neuronskoj mreži te se koriste za izračun gradijenata; „*The back-propagation algorithm (Rumelhart et al., 1986a), often simply called backprop, allows the information from the cost to then flow backward through the network in order to compute the gradient.*“[8]

Le Cun i suradnici (1989)[9] izvršili su istraživanje na temu primjene strojnog učenja prepoznavanja rukopisne numeracije u svrhu prepoznavanja poštanskih brojeva. Poštanski broj obično se sastoji od pet znamenaka i svaka ga osoba ručno piše drugačijim načinom, pri čemu je teško postići da svačiji izričaj pojedine znamenke bude u skladu sa standardnim načinom pisanja iste, već se često uvelike razlikuje. Pri rješavanju ovoga problema koriste se ulazni podaci u obliku sličica koje prikazuju ručno napisane brojke tako da je sama znamenka istaknuta kao skupina piksela crne boje na bijeloj površini čime se postiže zadovoljavajuća razina kontrasta. Baza podataka nad kojom se vršilo treniranje algoritma propagacije pogreške unatrag sačinjena je od 9298 nizova brojeva digitaliziranih iz ručno napisanih poštanskih brojeva koji se pojavljuju u poštanskome sustavu pokrajine Buffalo u New Yorku. Svaki primjerak pisanja poštanskog broja napisan je u različitim oblicima fonta i veličine te je napisan s različitim razinama brige o stilskom izričaju; „*The digits were written by many different people, using a great variety of sizes, writing styles and instruments, with widely varying levels of care*.“[10] Neuronska mreža korištena za ovo istraživanje jest mreža sačinjena od više slojeva u kojoj su sve veze prilagođene, a radi se nad modelom propagacije unatrag. Svim je vezama ograničeno spajanje tek u prvih nekoliko slojeva neuronske mreže kako bi se postigao učinak lokalizacije jer je cilj izbjeći prevelik broj parametara potrebnih za rad neuronske mreže. Putem ovoga pristupa dolazi do načina rada u kojem se značajka detektora za jedan komad slike ručno napisane brojke može koristiti i za druge dijelove slike. Rješenje problema je skenirati sliku koja je ulazni podatak s jednim neuronom koji ima lokalno receptivno polje, te pohraniti stanja tog neurona na odgovarajuće lokacije u sloju koji se naziva značajskom mapom (*engl. feature map*); „*solution to this problem is to scan the input image with a single neuron that has a local receptive field, and store the states of this neuron in corresponding locations in a layer called*

[8] Mashimo, *Back-propagation dor neural network* // *Recalling the perception.* 2015. URL: https://mashimo.wordpress.com/2015/09/13/back-propagation-for-neural-network/ (25.06.2020.)
[9] Le Cun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, J. D. Handwritten Digit Recognition with a Back-Propagation Network. AT&T Bell Laboratories, Holmdel, N. J. 07733, 1989.
[10] Le Cun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, J. D. Handwritten Digit Recognition with a Back-Propagation Network. AT&T Bell Laboratories, Holmdel, N. J. 07733, 1989.

a feature map."[11] Rezultati ovog istraživanja ukazali su na postotak pogreške sustava u iznosu od 1.1%, dok su rezultati istraživanja provedenim na čitavim setom podataka pokazali da se radi o postotku pogreške nad prepoznavanjem ručno napisane numeracije u iznosu od 3.4%.

2.1.3. Nenadzirano učenje

Razliku između nadziranog strojnog učenja (*engl. supervised learning*) i nenadziranog strojnog učenja (*engl. unsupervised learning*) čini to što set ulaznih podataka kod nenadziranog učenja nije niti označen, niti klasificiran ili kategoriziran na bilo koji način što dovodi do principa rada u kojem korišteni matematički model ima zadaću identificirati svaki oblik sličnosti u setu podataka. Na osnovu toga matematički model nadalje pokušava utvrditi postojeću strukturu ulaznih podataka. Kao što problemi kod nadziranog učenja imaju mogućnost grananja u budućnosti, tako se i problemi kod nenadziranog učenja mogu nadalje opisati kao problemi grupiranja (*engl. clustering problems*) i problemi asocijacije (*engl. association problems*). Jedan od primjera grupiranja prikazan je u istraživanju Seljan i suradnici (2020)[12] koji su izradili model grupiranja za 32 EU zemlje (*engl. clustering*) uz primjenu algoritma K-means za korištenje e-usluga s obzirom na dob i spol.

Problemi grupiranja temelje se na pokušavanju otkrivanja nerazdvojivosti u grupiranju vezanom uz set podataka, dok se problemi asocijativnosti svode na pokušaj generaliziranja pravila koja su korištena kao oblik opisa velikih dijelova seta podataka. Algoritmi za strojno učenje koji se koriste u području nenadziranog strojnog učenja su *K-means, Neural Networks, Linear Discriminant Analysis Principal Component Analysis, i Apriori Algorithm*[13].

2.1.4. Polu-nadzirano učenje

Kao što i sam naziv govori, polu-nadzirano strojno učenje (*engl. semi-supervised learning*) temelji se na principima rada i nadziranog i nenadziranog učenja. Ulazni set podataka pri radu s polu-nadziranim učenjem sačinjavaju i označeni i neoznačeni podaci. Uobičajeni je jest omjer taj da se radi o velikoj količini neoznačenih podataka, te podosta manjoj količini onih podataka koji su označeni. Matematički model koji se koristi pri polu-nadziranom strojnom učenju

[11] Le Cun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, J. D. Handwritten Digit Recognition with a Back-Propagation Network. AT&T Bell Laboratories, Holmdel, N. J. 07733, 1989.
[12] Seljan, S.; Miloloža, I., Pejić Bach, M. e-Government in European Countries: Gender and Ageing Digital Divide. Interdisciplinary Management Research XVI, 1563-1584
[13]Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)

koristi označene podatke u svrhu prepoznavanja strukture onih neoznačenih i na taj način dolazi do mogućnosti da učini predikciju. Kao i kod nadziranog učenja, problemi polu-nadziranog učenja kasnije imaju mogućnost biti grupirani kao oblici klasifikacijskih problema i problema regresije. Primjer moguće primjene polu-nadziranog strojnog učenja jest u analizi govora iz razloga što je označavanje auditornih datoteka postupak koji zahtjeva veliku količinu ljudskih resursa.[14]

2.2. Algoritmi sličnosti

Algoritmi sličnosti (*engl. similarity algorithms*) dobili su naziv po svojstvu grupacije na osnovu sličnosti vlastitih funkcija; „*Similarity algorithms are grouped by similarity of their function*".[15] Čest je slučaj da se algoritmi koji pripadaju ovoj skupini algoritama još detaljnije mogu objasniti u obliku podskupina.

2.2.1. Algoritmi regresije

Korištenje algoritama regresije u strojnom učenju zasnovanom na statističkom modelu nastalo je preuzimanjem regresijskih metoda zbog mogućnosti oblikovanja veza između varijabli. Putem višestrukih iteracija, algoritmi regresije imaju mogućnost postići to da su veze između varijabli čvršće i stabilnije, što kasnije dovodi do pojave točnije predikcije izlaznih podataka. „*Statistical machine learning has co-opted regression methods because of*

the modeling the relationship between variables. Regression algorithms can iteratively refine these relationships to predict a better outcome."[16]

Algoritmi regresije nadalje se dijele na algoritme linearne regresije (*engl. linear regression*) i na algoritme logističke regresije (*engl. logistic regression*). Linearna regresija smatra se jednim od osnovnih pristupa pri bavljenju analizama koje se bave predikcijama. Postoje dvije teze kojima se bavi područje regresije, a to su radi li set varijabli zadužen za predikciju uistinu dobar posao u smislu stvaranja valjane predikcije za izlazni podatak i koja je točno varijabla važan

[14] Rodriguez, J. *Understanding Semi-supervised Learning*. 2017. URL:https://medium.com/@jrodthoughts/understanding-semi-supervised-learning-a6437c070c87 (28.08.2020.)
[15] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[16] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)

faktor u stvaranju predikcije za izlazni podatak; *„(1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable? (2) Which variables in particular are significant predictors of the outcome variable.“*[17]

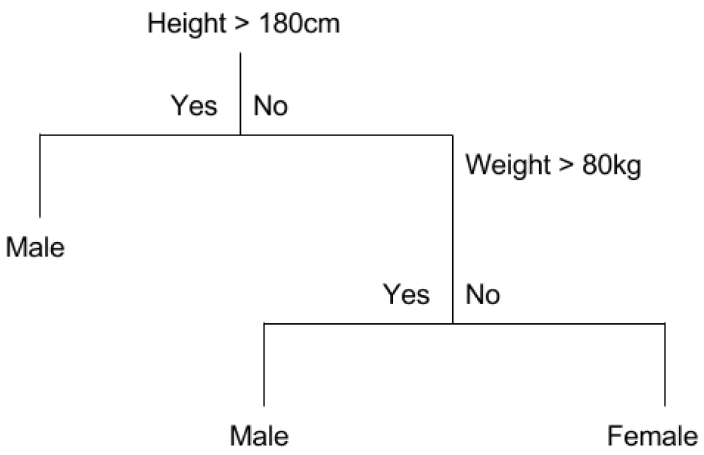
Navedene teze koriste se u svrhu pojašnjavanja veza između one varijable koja je zavisna i jedne ili više varijabli koje djeluju nezavisno što dovodi do toga da je regresiju moguće koristiti kako bi se utvrdila razina učinkovitosti nezavisne varijable ili više njih. Također, na taj je način moguće utvrditi i kolika je veličina promjene koja se dogodi u zavisnoj varijabli pri stvaranju promjena u jednoj ili više nezavisnih varijabli.

Logistički tip regresije se pak s druge strane bavi problematikom binarne klasifikacije. Kao glavni cilj logističkog tipa regresije valja navesti to da ona pokušava izraziti važnost svake ulazne varijable, no kod predikcije izlaznih podataka koristi se nelinearna funkcija čiji je naziv logistička funkcija (*engl. logistic function*).

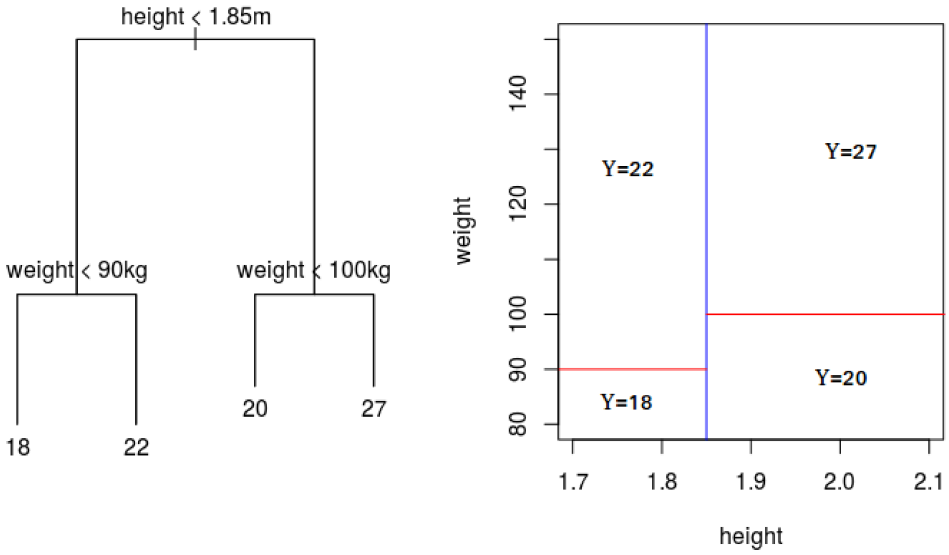
2.2.2. Algoritmi stabla odlučivanja

Kod algoritama stabla odlučivanja (*engl. decision tree*) početni korak jest stvaranje modela za stvaranje odluke. Princip rada kod stabala odlučivanja sličan je binarnim stablima. Kao temelj postavlja se teza na koju je moguće odgovoriti u dva ili više smjerova. Ovisno o tome u kojem smjeru je pohranjen odgovor, postavlja se nova teza koja također ima dva ili više ishoda što u konačnici dovodi do daljnjih grananja. Svaka donesena odluka od strane algoritma ovisna je o prethodnoj odluci i stvara mogućnost više smjerova odluka u grananju koje nakon toga slijedi. „*Kada ciljana varijabla u stablu odluke ima mogućnost primiti odvojeni set vrijednosti, tada se radi o klasifikacijskom stablu, a kada dođe do slučaja da je ciljana varijabla stalna, tada se radi o regresijskom stablu.*“[18] Način na koji algoritam klasifikacijskog stabla odlučivanja donosi odluku prikazan je na slici pod nazivom „Slika 1“, a slika pod nazivom „Slika 2“ nastoji prikazati način na koji odluku donosi algoritam regresijskog stabla odlučivanja.

[17] Statistics Solutions, *What is Linear Regression*. 2013. URL:https://www.statisticssolutions.com/what-is-linear-regression/ (27.06.2020.)
[18] Chakure, A. *Decision Tree Clasification*. 2019. URL:https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac (29.08.2020.)



Slika 1, Primjer klasifikacijskog stabla odlučivanja[19]



Slika 2, Primjer regresijskog stabla odlučivanja[20]

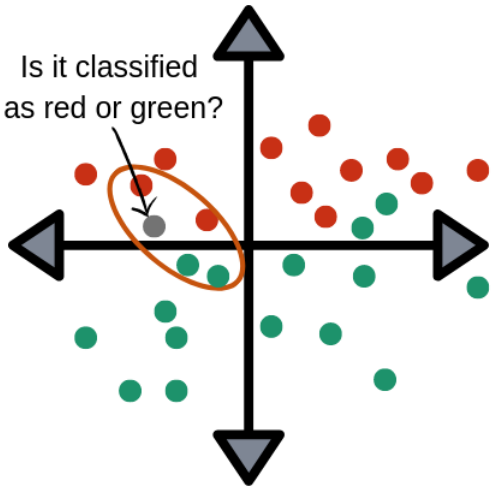
2.2.3. Metoda KNN

Prema Schott (2019)[21] neparametarska metoda ili klasifikator KNN (*engl. K nearest neighbors*) koristi se kako bi se riješili klasifikacijski problemi i problemi regresije. Ova metoda ima epitet algoritma za strojno učenje koji je spor iz razloga što se svi izračuni događaju tek pri zadnjoj fazi klasifikacije. Pošto se radi o jednostavnoj metodi strojnog učenja kojom se može postići

[19] Chakure, A. *Decision Tree Clasification*. 2019. URL:https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac (29.08.2020.)
[20] Chakure, A. *Decision Tree Clasification*. 2019. URL:https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac (29.08.2020.)
[21] Schott, M. *K-Nearest Neighbors (KNN) Algorithm for Machine Learning*. 2019. URL:https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26 (26.08.2020.)

prepoznavanje rukopisne numeracije, u primjeni ovakvih algoritama ne postoji eksplicitan oblik faze ranijeg treniranja algoritma nad podacima za treniranje i algoritam ne vrši generalizaciju nad setom podataka korištenim za treniranje. Ova se metoda koristi kada se radi nad velikim skupom ulaznih podataka.

Što se tiče samog algoritma u području prepoznavanja ručno napisane numeracije, izrazito je važno izračunati metriku udaljenosti između testnih podatkovnih jedinica promatranja i svih označenih podatkovnih jedinica promatranja i zatim označene podatkovne jedinice promatranja poredati u rastućem nizu s obzirom na metriku distance. Zatim algoritam odabire vrhovnu K označenu podatkovnu točku te ju uspoređuje s klasnim oznakama kako bi se utvrdila razina sličnosti. Kod odabira parametara u korištenju ovoga algoritma, važno je uzeti u obzir veličinu podataka nad kojima se vrši istraživanje. Najbolji odabir za parametar K ovisi o podacima. Što je veća vrijednost parametra K više se reducira efekt šuma prilikom klasifikacije, a dolazi do jasnijih granica između klasa; „*Best choice of K depends on data. The larger value of K reduce the effect of noise on classification but makes the decision boundaries between classless distinct.*“[22] Dakle, što je manja vrijednost K, veća je mogućnost pojave šuma sa jasnom razdvojenošću među klasama. „Slika 3“ slikovno i pojednostavljeno prikazuje princip rada KNN-a.



[22] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)

Slika 3, Jednostavan prikaz rada KNN-a[23]

Nadalje, od velike je važnosti što je više moguće smanjiti broj manje potrebnih ili nepotrebnih značajki. Naime, kada dođe do pojave da postoje u potpunosti suvišne značajke, valja ih ukloniti iz razloga što bez viška značajki postoji velika mogućnost da proces klasifikacije bude točnije i bolje odrađen. Točnost u obliku postotka nad testnim podacima, točnost u obliku postotka nad podacima za treniranje, vrijeme potrebno za treniranje i vrijeme potrebno za testove za uporabu ovog algoritma za klasificiranje u obliku algoritma za strojno učenje prepoznavanja rukopisne numeracije izračunao je Beniwal (2018)[24] u svojem radu *Handwritten Digit Recognition using Machine Learning*. Baza podataka nad kojom je vršeno ovo istraživanje je MNIST baza podataka koja je detaljnije opisana u ovome radu u poglavlju „Istraživanje“. Beniwal je došao do saznanja da točnost prepoznavanja nad testnim slikama ručno napisane numeracije iznosi 96.67%, a da točnost prepoznavanja nad podacima za treniranje iznosi 97.88%. Vrijeme koje je bilo potrebno ovoj vrsti klasifikatora za provođenje prepoznavanja nad podacima za treniranje bilo je 25 minuta, a vrijeme potrebno za prepoznavanje nad testnim podacima bilo je 15 minuta. Prikaz ovih vrijednosti vidljiv je u tablici pod nazivom „Tablica 1“

Tablica 1, Rezultati KNN-a[25]

Parametri	**KNN**
Točnost (testne slike)	96.67%
Točnost (slike za treniranje)	97.88%
Vrijeme (treniranje)	25 minuta
Vrijeme (testno)	15 minuta

[23] Schott, M. *K-Nearest Neighbors (KNN) Algorithm for Machine Learning*. 2019. URL:https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26 (26.08.2020.)
[24] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[25] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)

2.2.4. Umjetne neuronske mreže

Ideja stvaranja umjetnih neuronskih mreža u svrhu stvaranja mogućnosti samostalnog strojnog učenja temelji se na principu rada neuronskih mreža u ljudskome mozgu. Radi se o veoma kompleksnim sklopovima koji su sačinjeni od višestrukih algoritama koji imaju za zadaću procesirati iznimno velike i složene skupove podataka. Svaku je neuronsku mrežu moguće podijeliti na tri glavna sloja koji su sačinjeni od sloja ulaznih podataka (*engl. input layer*), skrivenih slojeva (*engl. hidden layer*) i od sloja izlaznih podataka (*engl. output layer*). Skriveni slojevi mogu imati više slojeva; „*the input layer, hidden layers, and the output layer. Hidden layers can have multiple layers.*“[26]

Svaki se sloj može opisati kao skupina čvorova koji zajedno čine umjetne živce. Svaki umjetno stvoreni čvor koji čini živac povezan je sa čvorom koji se nalazi u sljedećem sloju. „*Duboko učenje (engl. deep learning) zasnovano je upravo na stvaranju neuronskih mreža.*“[27] Algoritmi koji se koriste za duboko učenje su *Perceptron, Back-Propagation, Deep Boltzmann Machine (DBM) and Convolutional Neural Network (CNN).*[28]

„*Ideja rada ovakvog tipa algoritma za strojno učenje u primjeni za prepoznavanje rukopisne numeracije započinje korištenjem slučajnog odabira pri procesu inicijalizacije težine te implementacijom širenja. Potom je potrebno izračunati trošak. Slijedi korak u kojem se vrši procjena povratnog širenja kako bi se stvorila mogućnost za izračunom djelomičnih derivata. Zatim se vrši provjera gradijenata (engl. gradient check) što podrazumijeva metodu koja se koristi za brojčanu provjeru derivata izračunatih od strane samoga koda kako bi se uvjerili da je implementacija uspješno izvršena. Zatim slijedi isključivanje opcije provjere gradijenata.*“[29] Korištenjem gradijentnog spuštanja ili neke druge funkcije zadužene za optimizaciju moguće je minimizirati funkciju troškova. Važna stavka pri radu sa ovim oblikom algoritma jest to da je broj izlaznih podataka u obliku jedinica zapravo i sami broj klasa.

---

[26] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[27] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[28] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)
[29] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)

Beniwalova testiranja (2018)[30] nad ovim oblikom metode za postizanje prepoznavanja ručno napisanih znamenaka dovela su do zaključka kako korištenje neuronskih mreža (*engl. Neural Networks, NN*) pri izračunima dobivenim nad računanjem točnosti za testne podatke daje postotak pogreške od 3.15%. Uporabom podataka za treniranje utvrđeno je da se radi o postotku pogreške točnosti u iznosu od 2.17%. Vrijeme koje je bilo potrebno neuronskoj mreži kako bi dala rezultate vezane uz podatke za treniranje iznosilo je 35 minuta, dok je za obradu testnih podataka ovome klasifikatoru bilo potrebno 20 minuta. Tablični prikaz rezultata dobivenih na temu strojnog učenja prepoznavanja rukopisne numeracije korištenjem neuronskih mreža na temelju MNIST baze podataka Beniwala (2018)[31] vidljiv je u tablici pod nazivom „Tablica 2".

Tablica 2, Rezultati NN-a[32]

Parametri	NN
Točnost (testne slike)	3.15% (pogreška)
Točnost (slike za treniranje)	2.17% (pogreška)
Vrijeme (treniranje)	35 minuta
Vrijeme (testno)	20 minuta

2.2.5. Konvolucijske neuronske mreže

Oblik algoritma koji je moguće koristiti za duboko učenje, pa tako i za problematiku vezanu uz prepoznavanje rukopisne numeracije, a u srži je napredan oblik neuronske mreže, naziva se konvolucijskom neuronskom mrežom (*engl. Convolutional Neural Network, CNN*).

„*Saznanja o ovakvome tipu algoritma dovode do mogućnosti obavljanja raznih zadataka u kraćem vremenskom periodu i sa većim postotkom točnosti u odnosu na druge tipove klasifikatora*."[33] Primjena ove metode jest upravo u prepoznavanju slika i video zapisa te u

[30] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[31] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[32] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[33] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)

obradi prirodnog jezika. Korištenje konvolucijskih neuronskih mreža proširilo se kako društvena mreža Facebook koristi neuronske mreže za vlastiti automatski algoritam za označavanje; „*Use of CNN have spread as Facebook uses neural nets for their automatic tagging algorithms*“[34].

Razne varijante ovoga pristupa koriste i druge velike internetske korporacije poput giganata kao što su Amazon, Pinterest i Instagram. Prenošenje slike u obliku parametra i daljnja predviđanja je li neki uvjet vezan uz klasifikaciju slike ispunjen ili nije, područje je istraživanja i rješavanja uz pomoć ovoga algoritma. Uz ovu metodu moguće je relativno brzo doći do prepoznavanja obrazaca i generalizacije proizašle iz prijašnjih saznanja, što dovodi do zaključka da je ovaj pristup moguć za rješavanje problema vezanih uz strojno učenje prepoznavanja rukopisne numeracije.

Za oblik primjene u prepoznavanju obrazaca učenjem važno je znati da računalo preuzima sliku u obliku liste vrijednosti piksela. Svakome je broju dodijeljena vrijednost u rasponu od 0 do 255 s obzirom na intenzitet piksela u zadanom trenutku. Glavna bit jest ta da nakon preuzimanja slikovnog oblika ulaznog podatka algoritam ima mogućnost stvaranja pretpostavke ili predikcije o izlaznome podatku i može opisati kolika je vjerojatnost da slika pripada određenoj klasi. Postupak primjene ovog algoritma u obliku strojnog učenja prepoznavanja rukopisne numeracije jest da se ulazni podaci u obliku slika razdvoje na manje dijelove koji čine segmente velike slike. Svaki se taj razdvojeni i umanjeni dio nadalje implementira u sitan dio neuronske mreže. Posebno se označavaju segmenti koji se razlikuju od drugih. Potom je potrebno pohraniti rezultate segmenata iz razloga što nipošto nije poželjno unijeti pogrešan indeks izvorne datoteke. Tako se vrši postavljanje rezultata u mrežu istog rasporeda kao i originalna slika.

Istraživanje točnosti i potrebnog vremena uporabom ove vrste algoritma nad MNIST bazom podataka od strane Beniwala (2018) pokazalo je da konvolucijske neuronske mreže daju rezultat točnosti predikcije nad testnim slikama u iznosu od 98.72%. Izračunata je i točnost dobivena korištenjem podataka za treniranje same mreže, a ona je iznosila vrlo visokih 99.78%. Vrijeme koje je bilo potrebno konvolucijskoj neuronskoj mreži kako bi obradila podatke za treniranje bilo je 70 do 90 minuta. Za obradu testnih podataka bilo je potrebno 40 minuta. Tablični prikaz rezultata dobivenih koristeći algoritam konvolucijske neuronske mreže u svrhu

[34] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)

strojnog učenja prepoznavanja rukopisne numeracije nad MNIST bazom podataka od strane Beniwala (2018) nalazi se u tablici pod nazivom „Tablica 3“. Detaljan prikaz u obliku pojašnjenja samoga koda za ovaj oblik pristupa strojnom učenju nalazi se u ovome radu u poglavlju „Istraživanje“.

Tablica 3, Rezultati CNN-a[35]

Parametri	CNN
Točnost (testne slike)	98.72%
Točnost (slike za treniranje)	99.78%.
Vrijeme (treniranje)	70-90 minuta
Vrijeme (testno)	40 minuta

2.2.6. Stroj potpornih vektora

Sljedeća metoda pristupa strojnom učenju jest putem stroja potpornih vektora (*engl. support vector machine*). „*Korištenje ovog oblika sustava za pristup strojnome učenju ima puno potencijala u rješavanju problema vezanih sa klasifikacijom i regresijom, a usko je povezan uz pristup nadziranog strojnog učenja*.“[36] Kod ovoga pristupa, u setu podataka koji se koristi za treniranje svaka je podatkovna jedinica promatranja označena na način da joj je dodijeljena pripadnost u klasu 0 ili 1; „*In training dataset, each data point is marked whether it belongs to class 0 or class 1.*“[37] Vektor tada trenira model koji je stvoren na bazi seta podataka stvorenog za treniranje i zatim stvara predikciju pripadaju li nove podatkovne točke onoj klasi označenoj sa 0 ili označenoj sa 1.

Kako bi sa sigurnošću raspodijelio sve podatkovne jedinice promatranja, pristup potpornih vektora pokušava pronaći odgovarajući koeficijent koji služi za raspodjelu podatkovnih točaka na osnovu klase koja proizlazi iz hiperravnine. Hiperravnina se može definirati kao potpostor u prostoru koji dijeli taj sami prostor na dva razdvojena dijela prostora; „*A hyperplane in an n-*

[35] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[36] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[37] Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)

dimensional Euclidean space is a flat, n-1 dimensional subset of that space that divides the space into two disconnected parts.“[38] Točke koje služe za definiranje prostora hiperravnine dijele naziv sa imenom ovoga pristupa i nazivaju se vektorima potpore. Izvorni, engleski naziv za metodu klasifikatora u obliku stroja potpornih vektora glasi *Support Vector Machine* (SVM). Pri korištenju ovog algoritma za rješavanje problema prepoznavanja ručno napisane numeracije korištenjem strojnog učenja važno je provesti ispravan oblik identifikacije hiperravnine koja služi za bolje razdvajanje dviju klasa i zatim uspješno pronaći najdulji mogući oblik udaljenosti koji se odnosi na najbliže podatkovne točke i hiperravnine dok se mjera te udaljenosti izražava u obliku margine. Zaključno, potrebno je pronaći odgovarajuću hiperravninu s maksimalnom mogućom marginom koja je jednaka s obje strane. Hiperravnina s većom marginom je više robusna, dok se manja margina mijenja za pogrešnu klasifikaciju; „*Hyperplane with higher margin is more robust, whereas low margin has changed for misclassification*.“[39]

Stroj potpornih vektora odabire klasifikator iznimno točno s obzirom na maksimalnu marginu i ova je metoda algoritma podosta gruba za klasifikator te ima značajku kojom ignorira točke podataka koje se znaju razlikovati naspram ostalih opažanja. Tada je klasifikator u potrazi za hiperravninom s maksimalnim oblikom hiperravnine. U Beniwalovu istraživanju nad MNIST bazom podataka koristeći metodu stroja potpornih vektora, došlo je do zaključka kako ovaj algoritam ima postotak točnosti u iznosu od 97.91% kada se radi o podacima u obliku testnih slika ručno napisanih brojki. Pri izračunu točnosti klasifikatora dok je radio s podacima za treniranje, rezultat je bio 99.91% što je iznimno dobar rezultat s gotovo neznatnim postotkom pogreške. Vrijeme koje je bilo potrebno klasifikatoru kada je obrađivao podatke za treniranje, bilo je 19 minuta, a za obradu testnih podataka bilo mu je potrebno samo 11 minuta. Rezultati Beniwalova istraživanja vezano u ovaj algoritam klasifikacije tablično su prikazana u tablici pod nazivom „Tablica 4“.

Tablica 4, Rezultati SVM-a[40]

[38] Pupale, R. *Support Vector Machines (SVM) – An Overview*. 2018. URL:https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989 (29.06.2020.)
[39] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
[40] Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)

Parametri	SVM
Točnost (testne slike)	97.91%
Točnost (slike za treniranje)	99.91%.
Vrijeme (treniranje)	19 minuta
Vrijeme (testno)	11 minuta

2.3. Pojačano učenje

Duboko učenje u svrhu pojačanja (*engl. reinforcement learning*) temelji svoj rad na umjetnim neuronskim mrežama i na samoj arhitekturi korištenoj za postizanje pojačanja. S obzirom na to, dolazi do mogućnosti detaljnijeg definiranja funkcija i optimizacije te mapiranja parova stanja i djelovanja; „*That is, it unites function approximation and target optimization, mapping state-action pairs to expected rewards.*“[41] Pojačano učenje još se naziva i neuro-dinamičkim programiranjem. Pojačano strojno učenje nastoji riješiti probleme uzrokovane korelacijom akcija koje se trenutno provode i njihovih zakašnjelih izlaznih podataka. Cilj je postići to da program zna u kojoj mjeri i kada koristiti pojedinu funkciju, dakle da ima sposobnost samostalnog poboljšavanja povećavanjem opsega seta podataka korištenog za treniranje i povećanjem broja iteracija.

[41] Nicholson, C. *A Beginner's Guide to Deep Reinforcement Learning*. 2019. Pathmind, URL:https://pathmind.com/wiki/deep-reinforcement-learning (26.06.2020.)

3. Problemi pri raspoznavanju znamenki

Razvitak umjetne inteligencije kroz relativno kratak vremenski period pokazao je kako tehnološki napreci mogu dovesti do sveukupnog boljitka i olakšanja svakodnevnih radnji koje čovjek čini. Ipak, grana umjetne inteligencije koja se bavi strojnim učenjem nad zadanim modelom, u primjeru za prepoznavanje ručno napisane numeracije, i dalje nije u stanju pružiti postotak točnosti u iznosu od 100%. U primjerima u kojima se radi o ručno napisanim brojkama koje izrazito čitko napisane, algoritam neuronske mreže ne pokazuje nikakve oblike pogreške pri raspoznavanju o kojoj se točno brojci radi. S druge strane, kod slikovnih primjera istih znamenaka koji su tek neznatno modificirani, dakle promijenjeni izgledom, postotak točnih predikcija koje čini neuronska mreža se smanjuje iako je jasno da se radi o primjerima istih ručno napisanih brojki.

Istraživanje koje je proveo Brendel (2018)[42] godine potkrepljuje ovu tvrdnju. Slika pod nazivom „Slika 4“ prikazuje niz koji čini ručno napisani oblik znamenaka 9, 8, 2, 9 i 1. Ispod sličica vidljivo je kakva je bila odluka neuronske mreže, dakle o kojoj je ona brojci zaključila da se radi. U ovome je primjeru jasno da je algoritam točno klasificirao sve brojke.

Slika 4. Uspješna odluka mreže[43]

Slika pod nazivom „Slika 5“ prikazuje modificiranu verziju ikona s ručno napisanim brojkama i ispod svake ikone predikciju koju je neuronska mreža napravila. Na primjeru je vidljivo kako je neuronska mreža klasificirala niz brojki kao 3, 3, 9, 4, 4 iako se radi o istim brojkama kao i na prethodnom primjeru, uz čovjeku gotovo neznatne razlike. Ovime je dokazana pretpostavka

[42] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)
[43] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)

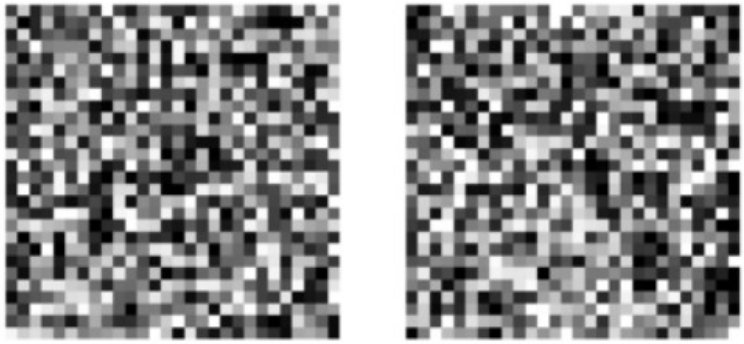
da je primjena umjetne inteligencije u ovom području i dalje u fazi razvitka te nije postigla svoj puni potencijal.

Slika 5. Neuspješna odluka mreže[44]

Ovakvi primjeri utječu na donošenje odluka kada se radi o algoritmima čija je zadaća vršenje prepoznavanja pojedinih objekata i semantičke segmentacije slika pri uporabi algoritma za filtriranje neželjene pošte u elektroničkoj pošti, te prepoznavanje govora.

Kao što je poznato u primjerima svake nove tehnologije koja danas uživa napredak, tako i za ovaj oblik vrijedi teza da se sve većim razvitkom stvara i sve veća mogućnost za napad na sustave koji koriste neuronske mreže iz razloga što je sada poznato da su ovakvi primjeri njihova slabost. Tehnološkim napretkom dolazi i do pojave napretka u sustavima koji žele naštetiti novoj tehnologiji. Drugi primjer pogreške koji neuronska mreža čini pri pokušaju klasifikacije ručno napisanih brojki jest kod sličica nad kojima nije moguće odrediti o kojoj se znamenci zapravo radi. Niti jedna znamenka nije napisana, već se radi o nasumičnoj skupini piksela različitih nijansi boja. Ipak, neuronska mreža u tom slučaju sa sigurnošću tvrdi da se radi o ručno napisanoj znamenci 0. Slika pod nazivom „Slika 6“ prikazuje dva primjera ove teze iz Brendelova istraživanja (2018)[45].

[44] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)
[45] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)



Slika 6. Zavaravanje neuronske mreže[46]

Ovakvi primjeri neprepoznatljivih slika daju naglasak na probleme koji zahvaćaju neuronske mreže kada se radi o području klasifikacije. Neuronska se mreža ponaša u potpunosti krivo u ovakvim situacijama gdje se kod ulaznih podataka ne radi o stvarnoj znamenci, već o prikazu šuma. Ovaj je tip problema na koji se nailazi prepoznat kao jedan od najvećih problema s kojima se razvitak umjetne inteligencije suočava pri implementaciji dubokog učenja na svakidašnje radnje.

Ideja rješenja ovog problema leži u učenju neuronske mreže da raspoznaje deblje linije od tanjih uzoraka. Na taj način i ljudski mozak razabire radi li se o nekome slovu ili brojci bez obzira na smetnje koje se istovremeno nalaze na samoj slici slova ili znamenke. Kod primjera brojke 7, ljudski mozak prepoznaje da se radi o dvije deblje linije, pri čemu je jedna dulja, a druga kraća. Ako se uspije postići da neuronska mreža umjetno misli na taj način, šanse za stvaranje pogrešne predikcije su vrlo male, bez obzira na eventualni šum. Takav bi model vrlo vjerojatno bio u mogućnosti točno klasificirati brojke napisane u sličicama koje se nalaze na slici pod nazivom „Slika 5“.

Kod primjera stvaranja generativnog modela neuronske mreže koja se uči prepoznavanju brojke 0, važno je postići da se u mreži koristi što više različitih ulaznih podataka za treniranje, dakle ručno napisanih znamenki u obliku sličica koje su različitih debljina i nagiba. Sljedeći je korak da neuronska mreža na osnovu tih podataka samostalno započne generirati razne varijante primjera napisane brojke 0 i takvim stvaranjem umjetnog iskustva moguće je postići da je postotak pogreške vrlo malen ili čak nepostojeći. Ovakav model strojnog učenja

[46] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)

primjenjiv je za svaku znamenku. Pri uvođenju novih ulaznih podataka neuronskoj mreži koja je zasnovana na ovome modelu, dolazi do visokog postotka točnosti klasifikacije svake pojedine sličice koja sadrži ručno napisani oblik brojčane znamenke. Ovakav oblik procedure obično se naziva *postupak analize po sintezi*[47], iz razloga što se analizira sadržaj slike s obzirom na model koji ga najbolje sintetizira; „*This procedure is typically called analysis-by-synthesis, because we analyse the content of the image according to the model that can best synthesise it*“[48]. Standardni oblici neuronskih mreža nemaju mehanizme koji šalju povratnu informaciju o tome sliči li zapravo brojka iz seta ulaznih podataka ikojoj klasi brojki prema kojima algoritam klasificira ulazne podatke. Iz toga je zaključak kako se uobičajenim neuronskim mrežama može samo vjerovati, jer ne postoji način kako bi se provjerila njihov predikcija. Uvođenjem postupka analize po sintezi, dolazi do mogućnosti da neuronska mreža provede dodatan korak prije donošenja samoga zaključka u kojoj klasi pripada sličica ulaznog podatka. Kako bi rezultati predikcija u svakome slučaju bili točni u iznosu od 100%, potrebno je provoditi istraživanja nad generativnim modelima savršenih oblika, a ne nad mutnim oblicima sličica kako je prikazano u slikama pod nazivima „Slika 4“ i „Slika 5“. Ipak, s obzirom i na tu dozu mutnosti, ovakav primjerak klasifikatora u mogućnosti je klasificirati ručno napisanu numeraciju u vrlo visokim postocima, po Brendelovu istraživanju i više od 99% što je objektivno gledano vrlo visok postotak. Klasifikator s implementiranim postupkom analize po sintezi za razliku od klasifikatora bez istog čini predikciju o kojoj se znamenci radi vrlo uvjereno bez obzira na to je li prisutan ikakav oblik šuma. Iako implementacija ovakvog modela stvara veći postotak točnosti predikcije, ovaj model se i dalje uvelike razlikuje od načina percepcije brojki kod ljudskoga mozga.

[47] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)
[48] Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)

4. Istraživanje

Jedan od ključnih koraka pri stvaranju softverskog rješenja za problematiku strojnog učenja prepoznavanja rukopisne numeracije jest mogućnost korištenja velike baze podataka koja sadrži stotine slikovnih verzija brojki koje su napisane od strane raznih ljudi i, takoreći, istrenirati neuronsku mrežu da dobije mogućnost i sposobnost prepoznavanja o kojoj se brojci radi.

U ovome primjeru bit će korištena MNIST baza podataka slikovno prikazane i ručno pisane numeracije koja sadrži 60 tisuća raznih slikovnih primjera brojki koje su napisane od strane gotovo 5 stotina različitih ljudi. Svi različiti slikovni primjeri brojki preuzeti iz ove baze dolaze zajedno s oznakama koje govore o kojoj se zapravo brojci radi kako ne bi došlo do zabune pri interpretaciji. Kada neuronska mreža „zapamti“ da se radi o pojedinoj brojci na osnovu ovih primjera, trebala bi biti u mogućnosti prepoznati o kojoj se brojci radi bez obzira koja osoba napiše koju brojku, dakle imati sposobnost točno detektirati rukopisnu numeraciju iako ona možda nema potpunu sličnost kao i ista brojka napisana u standardnom digitalnom obliku. Ovdje se radi o dekadskom brojevnom sustavu što znači da je sve brojeve moguće napisati sustavom brojevnih znamenaka od 0 do 9. MNIST baza podataka korištena za ovo istraživanje preuzeta je sa službene stranice. [49]

Kako bi programski kod koji slijedi u nastavku radio, potrebno je izvršiti sljedeću konfiguraciju programskog jezika Python; prvo, instalirati najaktualnije verzije paketa Numpy i Matplotlib. Zatim, instalirati paket Theano 0.8.2 i paket Lasagne 0.2dev1.

4.1. Primjer softverskog rješenja

Sljedeći je korak postavljane neuronske mreže s potrebnim slojevima i čvorovima te provesti strojno učenje kako bi stekla sposobnost sama prepoznati rukopisnu numeraciju. Kod je preuzet sa internetske stranice *Github* od Ansari (2019)[50] Provjera se izvršava prvotno na samo jednoj slici kako bi se uvidjelo je li izlazni podatak (*engl. output*) uistinu točno pogođena vrijednost, a zatim i na 10 tisuća slika također preuzetih iz MNIST baze podataka. Ovaj korak omogućuje

[49] MNIST baza podatakarukopisne numeracije, URL: http://yannlecun.com/exdb/mnist/ (02.07.2020.)
[50] Ansari, A. *From-0-to-1-Machine-Learning-NLP-Python-Cut-to-the-Chase.* 2019. Github, sekcija 15, URL:https://github.com/PacktPublishing/From-0-to-1-Machine-Learning-NLP-Python-Cut-to-the-Chase/blob/master/Section%2015/DigitRecognition-Python3.py (15.07.2020.)

precizno mjerenje točnosti neuronske mreže i potvrđuje u kojoj je mjeri strojno učenje uspješno provedeno.

```
# coding: utf-8

# In[1]:

import numpy as np

import os
```

Sljedeći prikaz koda potrebnog za stvaranje algoritma koji prepoznaje rukopisnu numeraciju bavi se preuzimanjem datoteka sa ranije navedene web stranice koja sadrži potrebnu bazu podataka, te pohranjuje iste na lokalni disk (*engl. local disc*).

```
def load_dataset():

 def download(filename, source='http://yann.lecun.com/exdb/mnist/'):

      print ("Downloading ",filename)

      import urllib.request

      urllib.request.urlretrieve(source+filename,filename)

import gzip
```

Potrebno je uvjeriti se da dio lokalnog diska nije već popunjen podacima. Sljedeći korak podrazumijeva liniju koda koja otvara *zip* oblik datoteke, što u sljedećim linijama dovodi do problema pitanja jednodimenzionalnih lista koji je rješiv na način da se liste pretvore u slikovni oblik. Ideja je postaviti veličinu slike kao jednobojnu sliku dimenzija 28x28 piksela. Prva dimenzija predstavlja broj slika, što dovodi do izvođenja drugih brojeva slika uz pomoć ostalih dimenzija i dužina korištenih lista. Druga dimenzija prikazuje brojku kanala, dok se treća i četvrta odnose na veličinu slike. Posljednja linija čini pretvorbu vrijednosti bajta (*engl. byte*) u decimalnu brojčanu vrijednost (*engl. float*) u rasponu oznaka (*engl. index*) od prvog do drugog mjesta (uključivo).

```
def load_mnist_images(filename):
    if not os.path.exists(filename):
        download(filename)
    with gzip.open(filename,'rb') as f:
        data=np.frombuffer(f.read(), np.uint8, offset=16)
        data=data.reshape(-1,1,28,28)
        return data/np.float32(256)
```

Naredna linija koda vrši ponovno čitanje datoteka koje su u binarnome obliku. Potom slijedi postavljanje *numpy* liste cijelih brojeva, dakle brojčanu vrijednost koja odgovara slikama.

```
def load_mnist_labels(filename):
    if not os.path.exists(filename):
        download(filename)
    with gzip.open(filename,'rb') as f:
        data = np.frombuffer(f.read(),np.uint8,offset=8)
    return data
```

Idući segment koda odnosi se na preuzimanje i čitanje testnih podataka korištenih za vježbu neuronske mreže što obuhvaća pojedine slike s oznakama.

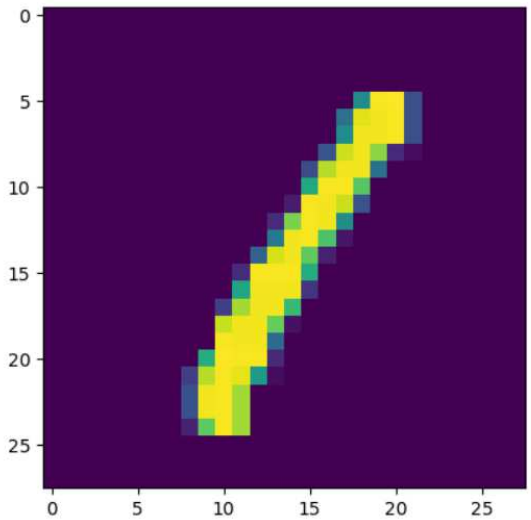
```
X_train = load_mnist_images('train-images-idx3-ubyte.gz')
  y_train = load_mnist_labels('train-labels-idx1-ubyte.gz')
  X_test = load_mnist_images('t10k-images-idx3-ubyte.gz')
  y_test = load_mnist_labels('t10k-labels-idx1-ubyte.gz')
 return X_train, y_train, X_test, y_test
```

```
# In[2]:

X_train, y_train, X_test, y_test = load_dataset()
```

Koristeći Python skup paketa *matplotlib* dolazi do mogućnosti pojedinačnih pregleda slika.

```
# In[3]:

import matplotlib

matplotlib.use('TkAgg')

import matplotlib.pyplot as plt

plt.show(plt.imshow(X_train[3][0]))
```

Zadnja linija u prethodno navedenom bloku koda omogućuje prikaz primjera broja iz baze podataka. „Slika 7“ prikazuje sliku te znamenke.



Slika 7. Primjer znamenke iz baze podataka

Iduća važna stavka jest postavljanje neuronske mreže s potrebnim slojevima i pripadajućim čvorovima kako bi se mreža mogla istrenirati za uspješno prepoznavanje rukopisne numeracije.

Slijedi postavljanje Python paketa kojima se postiže mogućnost mreže da ima sposobnost stvarnog učenja.

```
# In[16]:
import lasagne
import theano
import theano.tensor as T
```

Kreiranje ove neuronske mreže svodi se na implementaciju dvaju skrivenih slojeva koji sadrže po osam stotina čvorova, dok će izlazna vrijednost sadržavati deset čvorova numeriranih brojkama od 0 do 9, pri čemu svaki čvor dobiva vrijednost u rasponu od 0 do 1. Predviđena izlazna vrijednost proizlazi iz čvora s najvećom vrijednošću. Postavljeno je također odstupanje u vrijednosti od 20% u smislu otpuštanja rubova koji se nalaze između ulaznih vrijednosti i nadolazećih slojeva. Naknadno ovaj korak dovodi do pojave potpune povezanosti svih rubova. Ovaj sloj je inicijaliziran određenom težinom te je primijenjena *Glorot uniform* shema koja pospješuje brzinu inicijalizacije. Zatim slijedi ispuštanje u vrijednosti od 50% koje se odnosi na skriveni sloj. Daljnjim postavljanjem slojeva dobiva se jednako rješenje.

```
def build_NN(input_var=None):
l_in = lasagne.layers.InputLayer(shape=(None,1,28,28),input_var=input_var)
l_in_drop = lasagne.layers.DropoutLayer(l_in,p=0.2)
l_hid1= lasagne.layers.DenseLayer(l_in_drop,num_units=800,
                        nonlinearity=lasagne.nonlinearities.rectify,
                        W=lasagne.init.GlorotUniform())
l_hid1_drop = lasagne.layers.DropoutLayer(l_hid1,p=0.5)
l_hid2= lasagne.layers.DenseLayer(l_hid1_drop,num_units=800,
                        nonlinearity=lasagne.nonlinearities.rectify,
```

```
                    W=lasagne.init.GlorotUniform())
 l_hid2_drop = lasagne.layers.DropoutLayer(l_hid2,p=0.5)
```

Izlazni sloj sadrži deset jedinica. Najveća vrijednost izražena između 0 i 1 pri izlaznim podacima predstavlja predikciju brojke od strane neuronske mreže. Dolazi do vraćanja prvog sloja, no s obzirom na to da su svi slojevi međusobno povezani, osigurana je mogućnost vraćanja čitave mreže. Napokon dolazi egzaktan dio koji čini sustav u kojemu mreža uči kako pravilno napraviti predikciju, to jest kako prepoznati rukopisnu numeraciju na osnovu primjera kojima je već bila podložna. Prazne liste dobivaju ulogu mjesta za testne podatke koji su dani mreži kao primjeri za učenje. Prazna jednodimenzionalna lista cijelih brojeva predstavlja oznake. Također, potrebno je postaviti *error* funkciju koja će reći mreži kako da se ažurira s obzirom na vrijednost funkcije. Pozivanje funkcije koja se nalazi u posljednjem redu sljedećeg bloka koda služi treniranju neuronske mreže.

```
l_out = lasagne.layers.DenseLayer(l_hid2_drop, num_units=10,
                    nonlinearity = lasagne.nonlinearities.softmax)
return l_out
input_var = T.tensor4('inputs')
target_var = T.ivector('targets')
network=build_NN(input_var)
prediction = lasagne.layers.get_output(network)
loss = lasagne.objectives.categorical_crossentropy(prediction, target_var)
loss = loss.mean()
params = lasagne.layers.get_all_params(network, trainable=True) # Current value of all the
#weights
updates = lasagne.updates.nesterov_momentum(loss, params, learning_rate=0.01,
momentum = 0.9)
train_fn = theano.function([input_var, target_var], loss, updates=updates)
```

4.2. Rezultati

Rezultati koda bit će prikazani u nekoliko različitih varijanti s obzirom na to da je već kreirana neuronska mreža. Testiranje je obavljeno od strane kanala Machine Learning TV (2018)[51] u videu pod nazivom *Machine Learning for Handwritten Digit Recognition – Part 3.* U nadolazećim potpoglavljima bit će prikazana tri različita primjera rješenja dobivenih pokretanjem koda koji služi za učenje prepoznavanja rukopisne numeracije.

4.2.1. Prvi rezultat

Prvo se postavlja funkcija koja će predvidjeti izlazno rješenje za sliku. Za provjeru predikcije vezane uz jednu sliku postavlja se funkcija koja prosljeđuje vrijednost mreži koja je trenirana. Prilikom preuzimanja MNIST baze podataka, dobiven je pristup dvama kompletima slika od kojih je jedan komplet za treniranje, a drugi je testni komplet. Potrebno je aplicirati funkciju nad jednom slikom koja je prva po redu u testnom kompletu.

```
# In[27]:
test_prediction = lasagne.layers.get_output(network)
val_fn = theano.function([input_var],test_prediction)
val_fn([X_test[0]])
```

„Slika 8“ prikazuje rješenje gore navedenih funkcija. Izlistano je deset brojeva u vrijednosti od 0 do maksimalno 1. Najveća vrijednost nalazi se na indeksu 7.

```
Out[27]: array([[ 0.12529882,  0.03095455,  0.10531944,  0.08858692,  0.06188799,
                  0.08941018,  0.09295384,  0.1472574 ,  0.11782097,  0.14050988]])
```

Slika 8. Prikaz liste koja prikazuje deset brojeva u rasponu vrijednosti od 0 do 1[52]

[51] Machine Learning TV, *Machine Learning for Handwritten Digit Recognition – Part 3*. 2018. URL:https://www.youtube.com/watch?v=yGrzu9LU15E (30.08.2020.)
[52] Machine Learning TV, *Machine Learning for Handwritten Digit Recognition – Part 3*. 2018. URL:https://www.youtube.com/watch?v=yGrzu9LU15E (30.08.2020.)

Potrebno je također provjeriti stvarnu vrijednost znamenke.

 # In[28]:

y_test[0]

„Slika 9“ prikazuje da se uistinu radi o znamenci 7.

```
Out[28]: 7
```

Slika 9. Dokaz prikaza stvarne slike[53]

4.2.2. Drugi rezultat

Slijedi provjera koja se vrši nad uzorkom od deset tisuća preuzetih slika što će dovesti do izračuna preciznosti neuronske mreže u obliku postotka na način da se postavi funkcija koja prima slike i njihove oznake te se računa na osnovu stvarnih oznaka slika. Potrebno je izvršiti provjeru indeksa najveće vrijednosti u svakom testnom predviđanju i usporediti ga sa stvarnom vrijednošću.

 # In[31]:

test_prediction = lasagne.layers.get_output

test_acc=T.mean(T.eq(T.argmax(test_prediction, axis=1),target_var),dtype=theano.config.floatX)

acc_fn = theano.function([input_var,target_var],test_acc)

acc_fn(X_test,y_test)[54]

[53] Machine Learning TV, *Machine Learning for Handwritten Digit Recognition – Part 3*. 2018. URL:https://www.youtube.com/watch?v=yGrzu9LU15E (30.08.2020.)
[54] Ansari, A. *From-0-to-1-Machine-Learning-NLP-Python-Cut-to-the-Chase.* 2019. Github, sekcija 15, URL:https://github.com/PacktPublishing/From-0-to-1-Machine-Learning-NLP-Python-Cut-to-the-Chase/blob/master/Section%2015/DigitRecognition-Python3.py (15.07.2020.)

Prethodno navedeni dio koda dao je izračun koji prikazuje da se radi o točnosti u iznosu od 31.48% što je vidljivo na slici pod nazivom „Slika 10“. Postoji mogućnost dobivanja manjeg postotka dodavanjem funkcije *deterministic*, no i ovaj postotak daje zadovoljavajuću vrijednost.

```
Out[31]: array(0.3148)
```

Slika 10. Postotak pogreške[55]

4.3. Diskusija

Spektar primjena tehnologije koja podrazumijeva korištenje neuronskih mreža u svrhu učenja i kasnijeg samostalnog prepoznavanja numeracije je širok. MNIST baza podataka koja je korištena u primjeru neuronske mreže prikazane u ovome radu baza je podataka koja je podatkovno izrazito velika što iziskuje posjedovanje računala dovoljnog kapaciteta kako bi programski kod uspješno izvršio svoje zadaće. Istraživanje je potrebno pokrenuti lokalno na vlastitom računalu pomoću vlastitog GPU-a ili pomoću *cloud* servisa poput *Amazon Web Service (AWS)*, *Google Cloud Platform* ili *DigitalOcean*. Postoji i opcija u kojoj se kao set podataka nad kojim se vrši testiranje ne koristi MNIST baza podataka, već neki drugi set podataka, ovisno o željenoj vrsti primjene.

Daljnji razvitak sustava za prepoznavanje brojki može doprinijeti razvitku sveukupnog područja umjetne inteligencije. Primjerice, razvitkom pojedine teze dolazi do njenog testiranja, pri čemu je rezultate testiranja moguće usporediti s drugim pristupima istom problemu. Na taj način dolazi do uvida u prednosti i mane svakog sustava koji je korišten kako bi se riješio problem teze koja je testirana. Daljnja primjena vodi do mogućnosti korištenja sustava za prepoznavanje brojki u obliku testiranja učinkovitosti računalne platforme ili hardverske arhitekture.

Primjer u kojem je potreban sustav za prepoznavanje ručno napisanih brojki jest primjer poštanskoga sustava u pokrajini Buffalo u New Yorku prikazan u ovome radu u poglavlju pod

[55] Machine Learning TV, *Machine Learning for Handwritten Digit Recognition – Part 3*. 2018. URL:https://www.youtube.com/watch?v=yGrzu9LU15E (30.08.2020.)

nazivom „*propagacija unatrag*“. Domene u kojima razvitak ovakve tehnologije daje doprinos su također sigurnosni sustavi i automobilska industrija. Registarske oznake automobila sastoje se od slova i brojki, gdje u primjeru sigurnosnog mjerenja brzine vožnje automobila i nastojanja prepoznavanja potencijalnog vozača koji automobil vozi brzinom većom od propisane, važnost u prepoznavanju brojki imaju upravo sustavi bazirani na strojnom učenju prepoznavanja numeracije.

Prepoznavanje dijela registarske oznake moguće je pak provesti korištenjem sustava za optičko prepoznavanje znakova (*engl. Optical Character Recognition, OCR*). OCR je računalni softver koji prepoznaje nedigitalno znakovlje i potom ga pretvara u digitalno, što je problematika koja je kroz primjenu poveziva sa sustavima za strojno učenje prepoznavanja rukopisne numeracije u raznim domenama. Primjer prepoznavanja slova i brojki na registarskim oznakama automobila pripada domeni u kojoj su korišteni posebni računalni hardver i softver čija je zadaća automatsko prepoznavanje i digitalno zapisivanje registarskih oznaka bez potrebe čovjeka. Uz primjenu u obliku detekcije prebrzih vozača, postoji i primjena u obliku dodjele parkirnog mjesta na privatnom parkiralištu osobi kojoj to mjesto pripada, kada se govori o primjerice poslovnoj sferi primjene.

U automobilskoj industriji dolazi do razvitka sustava za autonomnu vožnju automobila bez operacija koje čini čovjek. To podrazumijeva mogućnost automobila da bez ičije asistencije samostalno prepoznaje ostale sudionike u prometu kao što su ostala vozila, pješaci ili pojedine barijere na koje je moguće naići u cestovnom prometu. Također, takav automobil mora biti u mogućnosti prepoznati znakove u prometu što uključuje i znakove koji upozoravaju ili daju naredbe u pisanom obliku riječi poput znaka „STOP“ i znakove kojima se određuje dozvoljena brzina vožnje koji su zapisani u brojčanom obliku. Važnost sustava za prepoznavanje napisane numeracije i sustava za prepoznavanje znakova uopće u ovoj primjeni leži u tome da vozilo pomoću takvog sustava uistinu prepozna o čemu se radi i potom usporedi te podatke sa svojom ugrađenom bazom podataka. Zatim, sustav zadužen za autonomiju automobila iz navedenim podataka zaključuje kakav oblik radnje je dužan poduzeti kako bi osigurao sigurnu vožnju za vozača automobila i za ostale sudionike u prometu.

Razvitkom sustava koji koriste strojno učenje prepoznavanja rukopisne numeracije i sustava za prepoznavanje znakova općenito, dolazi do razvitka i cjelokupnog područja umjetne inteligencije. Time pak dolazi i do stvaranja raznih oblika problematike u kojima je primjena nove tehnologije umjetne inteligencije moguća i nužna.

5. Zaključak

U ovome radu je prikazano kako upravo umjetno stvorene neuronske mreže već postižu iznimno velike postotke točnosti pri rješavanju problema vezanih za strojno učenje prepoznavanja rukopisne numeracije. Sustavi imaju sposobnost prepoznati o kojoj se točno znamenci radi bez obzira na to je li ona uistinu čitko napisana. Ovo je dokaz o tome kako umjetna inteligencija zaista ima mogućnost učenja i opravdano je pretpostaviti kako bi u zahtjevnijim područjima mogla zamijeniti čovjeka u određenim aktivnostima koje se ponavljaju. Ipak, ovo je područje i programiranja koje se svodi na formiranje ideje i na njezinu implementaciju. Razni pristupi koji imaju mogućnost korištenja na zadanu temu pokazali su da su u mogućnosti davanja rezultata s veoma sitnim međusobnim varijacijama što dovodi do zaključka kako svi do sada osmišljeni sustavi za stvaranje predikcija ove vrste imaju perspektivu postizanja rezultata zavidne točnosti. I dalje se kao problem nameće raspoznavanje i klasificiranje slikovnog prikaza znamenaka kod ulaznih podataka u kojima postoji šum određene vrste ili raznolikost prikaza, iako i za ove probleme već postoje određena rješenja te je veoma izvjesno da će i taj segment biti pokriven daljnjim radom i istraživanjima. Područje umjetne inteligencije kao takve i dalje se smatra područjem znanosti koje je u razvitku i u čiji potencijal je čovječanstvo tek zakoračilo. Ideja vodilja kojom se smatra postizanje stvaranja sustava sa neuronskim mrežama sličnog principa rada kao i kod ljudskog mozga već je i sada dovela do zaključka kako je takvo nešto uistinu moguće postići. Možda ne još danas, no u budućnosti zasigurno da.

6. Literatura

1. Ansari, A. *From-0-to-1-Machine-Learning-NLP-Python-Cut-to-the-Chase.* 2019. Github, sekcija 15, URL:https://github.com/PacktPublishing/From-0-to-1-Machine-Learning-NLP-Python-Cut-to-the-Chase/blob/master/Section%2015/DigitRecognition-Python3.py (15.07.2020.)
2. Beniwal, H. *Handwritten Digit Recognition using Machine Learning*. 2018. URL:https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (14.06.2020.)
3. Brendel, W. *AI still fails on robust handwritten digit recognition (and how to fix it)*. 2018. URL:https://medium.com/bethgelab/ai-still-fails-on-robust-handwritten-digit-recognition-and-how-to-fix-it-a432d84ede18 (01.07.2020.)
4. Chakure, A. *Decision Tree Clasification*. 2019. URL:https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac (29.08.2020.)
5. Dunđer, I. Sustav za statističko strojno prevođenje i računalna adaptacija domene. Doktorska disertacija. Sveučilište u Zagrebu, 2015.
6. Mashimo, *Back-propagation dor neural network // Recalling the perception.* 2015. URL: https://mashimo.wordpress.com/2015/09/13/back-propagation-for-neural-network/ (25.06.2020.)
7. Le Cun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, J. D. Handwritten Digit Recognition with a Back-Propagation Network. AT&T Bell Laboratories, Holmdel, N. J. 07733, 1989.
8. Machine Learning TV, *Machine Learning for Handwritten Digit Recognition – Part 3*. 2018. URL:https://www.youtube.com/watch?v=yGrzu9LU15E (30.08.2020.)
9. MNIST baza podataka rukopisne numeracije, URL:http://yann.lecun.com/exdb/mnist/ (02.07.2020)
10. Nicholson, C. *A Beginner's Guide to Deep Reinforcement Learning*. 2019. Pathmind, URL:https://pathmind.com/wiki/deep-reinforcement-learning (26.06.2020.)
11. Preuzimanje paketa Lasagne, URL:https://github.com/Lasagne/Lasagne/archive/master.zip (30.08.2020.)

12. Preuzimanje paketa Theano i Lasagne, URL:https://raw.githubusercontent.com/Lasagne/Lasagne/master/requirements.txt (30.08.2020.)

13. Pupale, R. *Support Vector Machines (SVM) – An Overview*. 2018. URL:https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989 (29.06.2020.)

14. Rodriguez, J. *Understanding Semi-supervised Learning*. 2017. URL:https://medium.com/@jrodthoughts/understanding-semi-supervised-learning-a6437c070c87 (28.08.2020.)

15. Rokand, Brij *Machine Learning Approaches and Its Applications*. 2019. URL:https://medium.com/datadriveninvestor/machine-learning-approaches-and-its-applications-7bfbe782f4a8 (22.06.2020.)

16. Schott, M. *K-Nearest Neighbors (KNN) Algorithm for Machine Learning*. 2019. URL:https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26 (26.08.2020.)

17. Seljan, S.; Miloloža, I., Pejić Bach, M. e-Government in European Countries: Gender and Ageing Digital Divide. Interdisciplinary Management Research XVI, 1563-1584

18. Statistics Solutions, *What is Linear Regression*. 2013. URL:https://www.statisticssolutions.com/what-is-linear-regression/ (27.06.2020.)

Popis slika

Slika 1. Primjer klasifikacijskog stabla odlučivanja

Slika 2. Primjer regresijskog stabla odlučivanja

Slika 3. Jednostavan prikaz rada KNN-a

Slika 4.Uspješna odluka mreže

Slika 5. Neuspješna odluka mreže

Slika 6. Zavaravanje neuronske mreže

Slika 7. Primjer znamenke iz baze podataka

Slika 8. Prikaz liste koja prikazuje deset brojeva u rasponu vrijednosti od 0 do 1

Slika 9. Dokaz prikaza stvarne slike

Slika 10. Postotak točnosti

Popis tablica

Tablica 1, Rezultati KNN-a

Tablica 2, Rezultati NN-a

Tablica 3, Rezultati CNN-a

Tablica 4, Rezultati SVM-a

Strojno učenje prepoznavanja rukopisne numeracije

Sažetak

Završni rad na temu strojnog učenja prepoznavanja rukopisne numeracije detaljno prikazuje podjelu algoritama zaduženih za strojno učenje na način da vrši uvid u same pristupe strojnom učenju i načine na koje se ti pristupi mogu koristiti kada dođe do problematike prepoznavanja ručno napisanih brojčanih znamenaka. Svi navedeni pristupi nadalje su razgranati kako bi se postigao jasan pregled najčešćih oblika klasifikatora zaduženih za istraživanja na ovu temu. Ovaj rad daje uvid u postotak točnosti i pogreške dobivenih na osnovu istraživanja vršenih za svaki pojedini algoritam koji se koristi u obliku sustava za klasifikaciju korištenih ulaznih podataka. Svi rezultati dobiveni testiranjem navedenih algoritama tablično su prikazani i detaljno objašnjeni. Također, vršen je i prikaz eventualnih problema koji se događaju pri stvaranju predikcija o kojoj se ručno napisanoj znamenci uistinu radi, kao i koja su eventualna rješenja za iste. U završnome dijelu rada prikazano je softwaresko rješenje ta temu ove problematike u obliku koda napisanog u programskom jeziku Python.

Ključne riječi: Strojno učenje, Rukopisna numeracija, Algoritmi, Klasifikatori

Handwritten digit recognition using machine learning

Summary

The final work on theme of handwritten digit recognition using machine learning shows in detail the division of machine learning algorithms by insight into the machine learning approaches and ways those approaches can be used in problems related to handwritten numbering. All of the above approaches are further explained in order to achieve a clear overview of the most common forms of classifiers in charge of research on this topic. This paper provides insight into the percentage of accuracy and error obtained based on the research performed for each individual algorithm it uses in the form of a system for classifying the input data used. All results obtained by testing this algorithm are tabulated and published in detail. Also, a presentation of possible problems that occur when creating a prediction about which handwritten digit is truly used, as well as possible solutions for the same. The final part of the paper presents a software solution on the topic in the form of code written in the Python programming language.

Key words: Machine learning, Handwritten numbering, Algorithms, Classifiers