Primena dubokog Q učenja na automatsko igranje video igara

Matematički fakultet
Univerzitet u Beogradu

Student:
Nikola Milev

Mentor:
Mladen Nikolić

Beograd, 2018.

Sadržaj

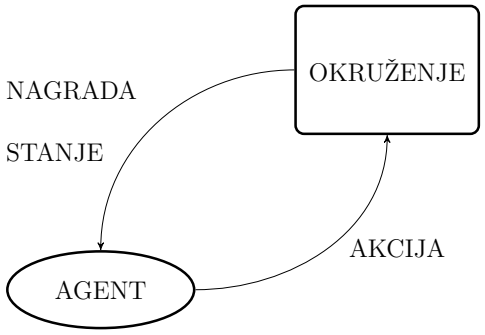
1

Uvod

U maju 1997. godine, Gari Kasparov, tadašnji svetski šampion u šahu, izgubio je meč protiv računarskog sistem pod nazivom "Deep Blue". Skoro dvadeset godina kasnije, program pod nazivom "AlphaGo" pobedio je profesionalnog ljudskog igrača u igri go. Iako su obe igre strateške i igraju se na tabli, između šaha i igre go postoji ogromna razlika. Pravila igre go dosta su jednostavna u odnosu na šah ali je prostor koji opisuje poteze igre go više od 10^{100} puta veći od prostora koji opisuje poteze šaha. Programi koji igraju šah često se zasnivaju na korišćenju stabala pretrage i ovaj pristup jednostavno nije primenljiv na igru go.

Na čemu je onda zasnovan "AlphaGo"? U pitanju je učenje potkrepljivanjem (eng. reinforcement learning). Ovo je vrsta mašinskog učenja koja počiva na sistemu kazne i nagrade. Podrazumeva se da se sistem sastoji od agenta i okruženja u kom agent dela (vrši akcije) i dobija povratne informacije o numeričkoj nagradi i promeni stanja okruženja. Osnovni dijagram ove komunikacije može se videti na slici 1.1. Poput dresiranja psa, nagradama se ohrabruje poželjno ponašanje dok se nepoželjno kažnjava. Cilj jeste ostvariti što veću dugoročnu nagradu. Međutim, agent mora sam kroz istraživanje da shvati kako da dostigne najveću nagradu tako što isprobava različite akcije. Takođe, preduzete akcije mogu da utiču i na nagradu koja se pojavljuje dugo nakon što je sama akcija preduzeta. Ovo zahteva da se uvede pojam buduće nagrade. Pojmovi istraživanja i buduće nagrade su ključni pri učenju potkrepljivanjem.

Pri učenju potkrepljivanjem, najčešće se pretpostavlja da je skup svih mogućih stanja okruženja diskretan. Ovo dozvoljava primenu Markovljevih procesa odlučivanja i omogućuje formalan opis problema koji se rešava, kao i pristupa njegovog rešavanja. Formalno pred-



Slika 1.1: Dijagram komunikacije agenta sa okruženjem pri učenju potkrepljivanjem

stavljanje problema i rešenja dato je u poglavlju 6.

Učenje potkrepljivanjem jedna je od tri vrste mašinskog učenja, pored nadgledanog i nenadgledanog učenja. Pri nadgledanom učenju, sistem dobija skup ulaznih i izlaznih podataka s ciljem da izvrši generalizaciju nad tim podacima i uspešno generiše izlazne podatke od do sada nevidjenih ulaznih podataka. Pri učenju potkrepljivanjem, ne postoje unapred poznate akcije koje treba preduzeti već sistem na osnovu nagrade mora zaključi koji je optimalni sled akcija. Iako široko korišćeno, nadgledano učenje nije prikladno za učenje iz novih iskustava, kada ciljni rezultati nisu dostupni. Kod nenadgledanog učenja, često je neophodno pronaći neku strukturu u podacima nad kojima se uči bez ikakvog predašnjeg znanja o njima. Iako učenje potkrepljivanjem liči i na nadgledano i na nenadgledano učenje, agent ne traži strukturu niti postoji unapred određeno optimalno ponašanje[1] već teži maksimizaciji nagrade koju dobija od okruženja.

Učenje potkrepljivanjem ima primene u raznim poljima kao što su industrija, istraživanje podataka, mašinsko učenje (kompanije Gugl (eng. Google) koristi učenje potkrepljivanjem radi automatskog dizajna neuralnih mreža), obrazovanje, medicina i finansije. Ovaj vid mašinskog učenja pokazao se kao dobar i za igranje video igara. U radu objavljenom 2015. godine u časopisu "Nature", "DeepMind" predstavlja sistem koji uči da igra video igre sa konzole Atari 2600, neke čak i daleko bolje od ljudi[2]. U avgustu 2017. godine, OpenAI predstavlja agenta koji isključivo kroz igranje igre i bez predašnjeg znanja o igri stiče nivo umeća dovoljan da pobedi i neke od najboljih ljudskih takmičara u video igri "Dota 2"[3].

U naučnom radu koji je objavila kompanija "DeepMind" u časopisu "Nature" predložen je novi algoritam, DQN (deep Q - network), koji koristi spoj učenja uslovljavanjem i duboke neuronske mreže i uspesno savladava razne igre za Atari 2600 konzolu. Sve informacije dostupne agentu jesu pikseli sa ekrana, trenutni rezultat u igri i signal za kraj igre. Algoritam skladišti prethodna iskustva i umesto učenja neuronske mreže na osnovu samo poslednjih akcija i nagrada, prethodno iskustvo se periodično koristi radi treniranja mreže, nasumičnim uzorkovanjem, smanjujući korelaciju između ulaznih podataka. U sklopu ovog rada, ispitana je struktura algoritma DQN i data je implementacija čije su performanse ispitane na manjoj skali od one date u radu, zbog ograničenih resursa. Takodje je eksperimentisano sa elementima samog algoritma i opisano kako oni utiču na njegovo ponašanje.

[MOZDA NESTO O REZULTATIMA KADA IH BUDE]

U sklopu rada opisani su osnovni pojmovi mašinskog učenja (glava 2), zadržavajući se na neuronskim mrežama uopšte (glava 3) i na konvolutivnim neuronskim mrežama (glava 4). Glava 6 posvećena je učenju potkrepljivanjem dok je algoritam DQN u celosti opisan u glavi 7. U glavi 8 data je implementacija kao i njena evaluacija, dok su eksperimenti i njihovi rezultati opisani u glavi 9.

[1] Postoji optimalno ponašanje ali ono nije poznato agentu na početku učenja.
[2] UBACI NEKU REFERENCU
[3] https://blog.openai.com/dota-2/

2

Mašinsko učenje

Mašinsko učenje počelo je da stiče veliku popularnost devedesetih godina prošlog veka zahvaljujući potrebi i mogućnosti da se uči iz ogromne količine dostupnih podataka i uspešnosti ovog pristupa u tome. Za popularizaciju mašinskog učenja početkom 21. veka najzaslužnije su neuronske mreže, u toj meri da je pojam mašinskog učenja često poistovećen sa pojmom neuronskih mreža. Ovo, naravno, nije tačno; sem neuronskih mreža, postoje razne druge tehnike, kao što su metod potpornih vektora, genetski algoritmi, itd.

Mašinsko učenje nastalo je iz čovekove želje da oponaša prirodne mehanizme učenja kod čoveka i životinja kao jedne od osnovnih svojstava inteligencije i korišćenja dobijenih rezultata u cilju praktične upotrebe. Termin mašinsko učenje prvi je upotrebio pionir veštačke inteligencije, Artur Semjuel[1], koji je doprineo razvoju veštačke inteligencije istražujući igru dame (eng. checkers) i tražeći način da stvori računarski program koji na osnovu iskusva može da savlada ovu igru[2].

Mašinsko učenje može se definisati kao disciplina koja se bavi izgradnjom prilagodljivih računarskih sistema koji su sposobni da poboljšaju svoje performanse koristeći informacije iz iskustva [3]. No, u biti mašinskog učenja leži generalizacija, tj. indukcija. Dve vrste zaključivanja, indukcija[4] i dedukcija[5] imaju svoje odgovarajuće discipline u sklopu veštačke inteligencije: mašinsko učenje i automatsko rezonovanje. Kao što se indukcija i dedukcija razlikuju, i mašinsko učenje i automatsko rezonovanje imaju različite oblasti primene. Automatsko rezonovanje zasnovano je na matematičkoj logici i koristi se kada problem čovek relativno lako može formulisati ali ga, često zbog velikog prostora mogućih rešenja, ne može jednostavno rešiti. U ovoj oblasti, neophodno je dobiti apsolutno tačna rešenja, ne dopuštajući nikakav nivo greške. Pri mašinskom učenju, teže je formalno definisati problem jer postoji relativno visok nivo apstrakcije. Čovek neke od ovih problema lako rešava a neke ne. Ukoliko je neophodno napraviti sistem koji prepoznaje lica na slikama, kako definisati problem? Od čega se tačno sastoji lice? Kako prepoznati elemente lica? Metodama automatskog rezonovanja bilo bi nemoguće definisati ovaj problem i rešiti ga. Mašinsko učenje, s druge strane, pokazalo se kao dobar pristup. Ono što je još karakteristično za mašinsko učenje jeste da rešenje ne mora biti savršeno tačno, iako se tome teži, i nivo prihvatljivog odstupanja

[1] https://en.wikipedia.org/wiki/Machine_learning – da li da citiram Wiki ili njihov izvor?

[2] http://infolab.stanford.edu/pub/voy/museum/samuel.html – kako citirati izvor sa veba?

[3] http://poincare.matf.bg.ac.rs/~janicic/courses/vi.pdf – pretpostavljam da stavim referencu ka literaturi gde će knjiga biti navedena?

[4] Indukcija – zaključivanje od pojedinačnog ka opštem

[5] Dedukcija – zaključivanje od opšteg ka konkretnom

zavisi od problema i konteksta primene.

Ova oblast je kroz manje od 20 godina od popularizacije postala deo svakodnevnice. U sklopu društvene mreže Fejsbuk (eng. Facebook) implementiran je sistem za prepoznavanje lica koji preporučuje profile osoba koje se nalaze na slikama. Razni veb servisi koriste metode mašinskog učenja radi stvaranja sistema za preporuke (artikala u prodavnicama, video sadržaja na platformama za njihovo gledanje, itd). i sistema za detekciju prevara. Mnoge firme koje se bave trgovinom na berzi imaju sisteme koji automatski trguju deonicama. U medicini, jedna od primena mašinskog učenja jeste za uspostavljanje dijagnoze. Još neke primene su u marketingu, za procesiranje prirodnih jezika, bezbednost, itd.

2.1 Vrste mašinskog učenja

Kada se priča o određenoj vrsti mašinskog učenja, podrazumevaju se vrste problema, kao i načini za njihovo rešavanje. Prema problemima koji se rešavaju, mašinsko učenje deli se na tri vrste: nadgledano učenje (eng. supervised learning), nenadgledano učenje (eng. unsupervised learning) i učenje potkrepljivanjem (eng. reinforcement learning). Iako se podela mnogih autora sastoji samo iz nadgledanog i nenadgledanog učenja, postoji razlika između učenja potkrepljivanjem i preostale dve vrste. U nastavku su dati opisi pristupa nadgledanog i nenadgledanog učenja. Učenju uslovljavanjem, kao centralnoj temi ovog rada, posvećeno je više pažnje u poglavlju 6.

2.1.1 Nadgledano mašinsko učenje

Pri nadgledanom mašinskom učenju, date su vrednosti ulaza i izlaza koje im odgovaraju za određeni broj slučajeva. Sistem treba na osnovu već datih veza za pojedinačne parove da ustanovi kakva veza postoji između tih parova i izvrši generalizaciju, odnosno, ukoliko ulazne podatke označimo sa x a izlazne sa y, sistem treba da odredi funkciju f takvu da

$$y \approx f(x)$$

Pri uspešno rešenom problemu nadgledanog učenja, funkcija f davaće tačna rešenja i za podatke koji do sada nisu viđeni. Ulazne vrednosti nazivaju se atributima (eng. features) a izlazne ciljnim promenljivima (eng. target values). Ovim opisom nije određema dimenzionalnost ni za ulazne ni za izlazne promenljive, iako je dimenzija izlazne promenljive uglavnom 1. Funkcija f naziva se modelom.

Skup svih mogućih funkcija odgovarajuće dimenzionalnosti bio bi previše veliki za pretragu i zbog toga se uvode pretpostavke o samom modelu. Pretpostavlja da je definisan skup svih dopustivih modela i da je potrebno naći najpogodniji element tog skupa. Najčešće je taj skup određen parametrima, tj. uzima se da funkcija zavisi od nekog parametra w koji je u opštem slušaju višedimenzioni i tada se funkcija označava sa $f_w(x)$.

Neophodno je uvesti funkciju greške modela (eng. loss function), odnosno funkciju koja opisuje koliko dati model dobro određuje izlaz za dati ulaz. Ova funkcija se najčešće označava sa L i $L(y, f_w(x))$ predstavlja razliku između željene i dobijene vrednosti za pojedinačni par

promenljivih. No, nijedan par promenljivih nije dovoljan za opis kvaliteta modela već treba naći funkciju koja globalno ocenjuje odstupanje modela od stvarnih vrednosti. U praksi, podrazumeva se postojanje uzorka:

$$D = \{(x_i, y_i)|i = 1, ..., N\}$$

i uvodi se empirijski rizik, odnosno sledeća funkcija:

$$E(w, D) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_w(x_i))$$

koja se još naziva prosečnom greškom. Neretko se skup D ne navodi već se njegovo postojanje podrazumeva. Uobičajeno, algoritmi nadgledanog mašinskog učenja zasnivaju se na minimizaciji prosečne greške. Ipak, treba imati u vidu da ovaj pristup nije teorijski zagarantovan i da to zavisi od skupa modela po kom se vrši minimizacija.

Postoje dva osnovna tipa nadgledanog mašinskog učenja:

- Klasifikacija
- Regresija

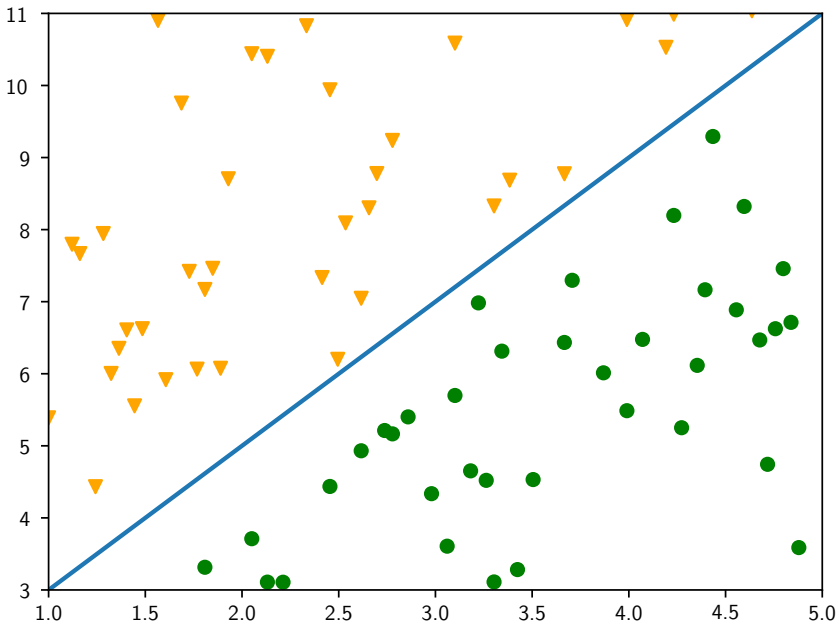
Klasifikacija (eng. classification) predstavlja oblast mašinskog učenja gde je cilj predvideti klasu u kojoj se ciljna promenljiva nalazi. Neki od primera klasifikacije su svrstavanje slika na one koje sadrže ili ne sadrže lice, označavanje nepoželjne (spam) elektronske pošte i prepoznavanje objekata na slikama. Najjednostavniji primer klasifikacije može se videti na slici 2.1, gde su trouglovima označeni podaci iznad prave $y = 2x + 1$ a krugovima podaci ispod date prave.

Regresija se odnosi na skup problema (i rešenja) u kojima je ciljna promenljiva neprekidna. Na primer, cene nekretnina mogu se predvideti na osnovu površine, lokacije, populacije koja živi u komšiluku, itd. Često korišćena vrsta regresije jeste linearna regresija. U slučaju linearne regresije, podrazumeva se da je funkcija $f_w(x)$ linearna u odnosu na parametar w. Iako se ovo na prvi pogled čini kao prilično jako ograničenje, to nije slučaj; kako za atribute ne postoji zahtev za linearnosti, oni pre pravljenja linearne kombinacije mogu biti proizvoljno transformisani. Primer linearne regresije jeste aproksimacija polinomom:

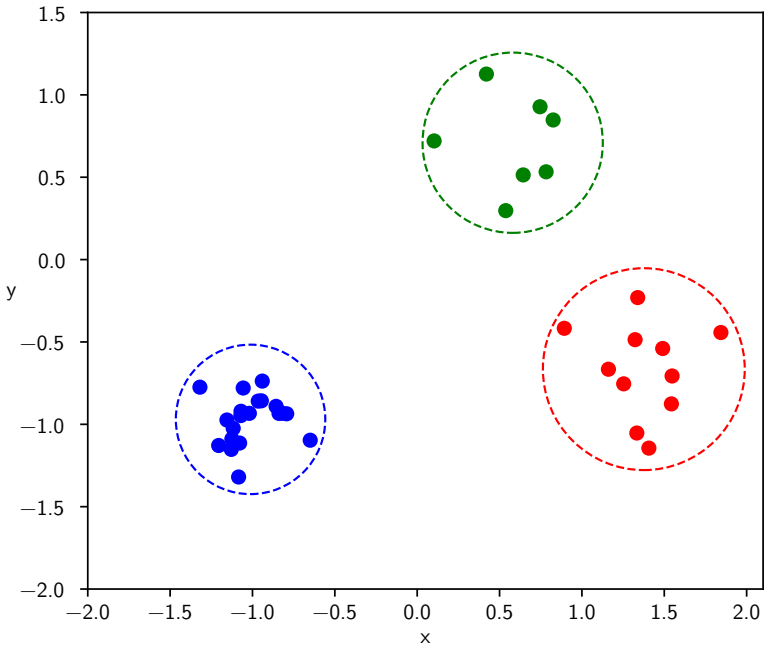
$$f_w(x) = w_0 + \sum_{i=1}^{N} w_i x^i$$

2.1.2 Nenadgledano mašinsko učenje

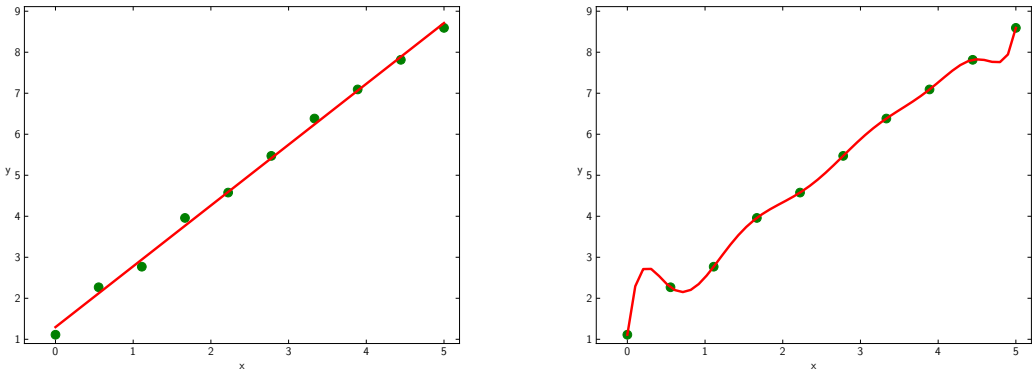
Nenadgledano učenje obuhvata skup problema (i njihovih rešenja) u kojima sistem prihvata ulazne podatke bez izlaznih. Ovo znači da sistem sam mora da zaključi kakve zakonitosti važe u podacima. Kako nije moguće odrediti preciznost sistema pa je cilj naći najbolji model u odnosu na neki kriterijum koji je unapred zadat. Jedan primer nenadgledanog mašinskog učenja je klasterovanje: sistem grupiše neoznačene podatke u odnosu na neki kriterijum koji nije unapred poznat. Svaka grupa (klaster) sastoji se iz podataka koji su međusobno slični i različiti od elemenata preostalih grupa u odnosu na taj kriterijum. Jednostavan primer klasterovanja po numeričkim atributima x i y može se videti na slici 2.2.



Slika 2.1: Binarna klasifikacija tačaka u skladu sa položajem u odnosu na pravu $2x + 1$



Slika 2.2: Klasterovanje



Slika 2.3: Primer odabira modela pri linearnoj regresiji polinomom

2.2 Dizajn sistema za mašinsko učenje

Okvirno, koraci u rešavanju problema su sledeći:

- Prepoznavanje problema mašinskog učenja (nadgledano učenje, nenadgledano učenje, učenje potkrepljivanjem);
- Prikupljanje i obrada podataka, zajedno sa odabirom atributa;
- Odabir skupa dopustivih modela;
- Odabir algoritma učenja; moguće je odabrati postojeći algoritam ili razviti neki novi koji bolje odgovara problemu
- Izbor mere kvaliteta učenja;
- Obuka, evaluacija i, ukoliko je neophodno, ponavljanje nekog od prethodnih koraka radi unapređenja naučenog modela

Prilikom odabira modela treba imati na umu vrstu problema koja se rešava, količinu podataka, zakonitosti koje važe u podacima, itd. Slika 2.3 prikazuje razliku između dva modela iz skupa dopustivih modela za linearnu regresiju polinomom nad 10 različitih tačaka. Na levom delu slike prikazan je polinom reda 1 (prava) a na desnom delu prikazan je polinom reda 10. Iako će polinom reda 10 savršeno opisivati 10 tačaka sa slike, vidi se da su one raspoređene blizu prave i, uprkos većem odstupanju takvog modela od podataka za učenje, jasno je da je prava bolji izbor.

2.2.1 Podaci

Mašinsko učenje bavi se generalizacijom nad nepoznatim objektima na osnovu već viđenih objekata. Pod pojmom objekta misli se na pojedinačni podatak koji sistem vidi. Koriste se još i izrazi primerak i instanca. Vrednosti podataka pripadaju nekom unapred zadatom skupu. Podaci mogu biti različitog tipa: numerički ili kategorički. Skupovi koji određuju vrednosti kojima se instance određuju nisu unapred zadati i neophodno ih je odrediti na način pogodan za rešavanje konkretnog problema. Na primer, ukoliko je neophodno razvrstati slike

životinja i biljaka na te dve kategorije, informacija o količini zelene boje na slici može biti prilično korisna, dok pri razvrstavanju vrste biljaka u zavisnosti od lista ovaj podatak skoro nije upotrebljiv (ali podatak o nijansi zelene boje može biti). Dakle, dobar izbor atributa imaće veliki uticaj na kasnije korake učenja. Podaci se sistemu daju kao vektori atributa.

Podaci se neretko pre slanja sistemu obrađuju na neki način; ovaj postupak zove se pretprocesiranje. Postoje mnogi razlozi za pretprocesiranje a glavni cilj jeste da se dobiju objekti nad kojima učenje može da se desi. Međutim, i to zavisi od problema. Nekada će nepotpuni objekti, podaci koji ne sadrže sve informacije neophodne za učenje, biti izbačeni iz skupa podataka koji se razmatra, a u nekom drugom slučaju, i oni će biti korišćeni. Jedan primer pretprocesiranja jeste pretvaranje slike koja je u boji u crno beli zapis.

2.2.2 Evaluacija modela

Nakon obučavanja (treniranja), neophodno je izvršiti evaluaciju dobijenog modela. Na koji god način se ovo izvršava, podaci korišćeni za obučavanje ne smeju se koristiti za evaluaciju modela. Često se pribegava podeli podataka na skupove za obučavanje i za testiranje. Skup za obučavanje obično iznosi dve trećine skupa ukupnih podataka. No, kako različite podele skupa mogu izazvati dobijanje različitih modela, slučajno deljenje nije najbolji izbor. Često korišćena tehnika jeste unakrsna validacija. Ovaj pristup podrazumeva podelu skupa podataka D na K podskupova približno jednake veličine, S_i za $i = 1, ..., K$. Tada se za svako i model trenira na skupu $D \setminus S_i$ a evaluacija se vrši pomoću podataka iz S_i. Posle izvedenog postupka za sve i, kao konačna ocena uzima se prosečna ocena svakog od K treniranja i evaluacija modela. Za vrednosti K uobičajeno se uzimaju vrednosti 5 ili 10. Ovaj metod vodi pouzdanijoj oceni kvaliteta modela.

2.3 Problemi pri mašinskom učenju

Kao što je podrazumevano pri pomenu pojma generalizacije, nije dovoljno odrediti funkciju koja dobro određuje izlazne vrednosti na osnovu promenljivih nad kojima se uči već je poželjno i novim ulaznim podacima dodeliti tačnu izlaznu vrednost. Odavde se može videti da je primer lošeg sistema za mašinsko učenje onaj sistem koji će izuzetno dobro naučiti da preslikava ulazne vrednosti iz skupa za učenje u odgovarajuće izlazne vrednosti ali u situaciji kada se iz tog skupa izađe neće davati zadovoljavajuće rezultate. Ovaj problem ima svoje ime: preprilagođavanje. Postoji i problem potprilagođavanja, koji podrazumeva da se sistem nije dovoljno prilagodio podacima. I preprilagođavanje i potprilagođavanje predstavljaju veliki problem ukoliko do njih dođe. Primer preprilagođavanja može se videti na slici 2.3. Polinomom stepena 10 model se savršeno prilagodio podacima za trening ali neće biti u stanju da izvrši generalizaciju za nove podatke.

Na još jedan od mogućih problema nailazi se u slučaju neprikladnih podataka. Moguće je da ulazni atributi ne daju dovoljno informacija o izlaznim. Takođe je moguće da podataka jednostavno nema dovoljno. U ovom slučaju, sistem ne dobija dovoljno bogat skup informacija kako bi uspešno izvršio generalizaciju. S druge strane, moguće je da postoji prevelika količina podataka. Tada se pribegava pažljivom odabiru podataka koji se koriste za učenje ali ovo u opštem slučaju treba izbegavati jer su podaci izuzetno vredan element procesa mašinskog učenja. Još jedan problem vezan za podatke može biti njihova nepotpunost. Na

primer, moguće je da u nekim instancama postoje nedostajuće vrednosti atributa.

Kako je najčešče potrebno pretprocesirati podatke u sklopu procesa mašinskog učenja, moguće je da u ovom postupku dođe do greške. Primera radi, prilikom rada sa konvolutivnim neuronskim mrežama, o kojima će biti reči u jednom od narednih glava, nekada se slike u boji pretvaraju u crno-bele. Ako se primeni transformacija koja onemogućuje razlikovanje objekata koji su različiti u početnoj slici a razlikovanje je neophodno za ispravno učenje sistema, tada proces treniranja neće teći kako je planirano.

Problem može da nastane i ukoliko nije odabran pravi algoritam za učenje, ukoliko se loše pristupilo procesu optimizacije, prilikom lošeg procesa evaluacije i, naravno, prilikom loše implementacije algoritma. Sve ove prepreke moguće je prevazići ali je jasno da je neophodno biti izuzetno pažljiv prilikom celog procesa mašinskog učenja.

3

Neuronske mreže

4

Konvolutivne neuronske mreže

5

Markovljevi procesi odlučivanja

6

Učenje potkrepljivanjem

7

DQN

8

Detalji implementacije

9

Eskperimentisanje sa elementima algoritma DQN