

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – 1. stopnja

Laura Guzelj Blatnik

**Nevronske mreže z vzratnim razširjanjem napak v
funkcijskem programskem jeziku**

Delo diplomskega seminarja

Mentor: prof. dr. Ljupčo Todorovski
Somentor: asist. dr. Aljaž Osojnik

Ljubljana, 2020

KAZALO

1. Uvod	4
2. Umetne nevronske mreže	4
2.1. Biološko ozadje	4
2.2. Zgradba nevronskih mrež	4
2.3. Nevron	5
2.4. Delovanje perceptrona	5
2.5. Lastnosti nevronskih mrež	5
2.6. Uporaba nevronskih mrež	5
3. Učenje	6
3.1. Pravilo delta	6
3.2. Ideja vzratnega razširjanja napake	6
3.3. Vzratno razširjanje napake	6
3.4. Lastnosti vzratnega razširjanja napake	8
4. Funkcijski programski jezik OCaml	8
5. Praktični del	8
6. Zaključek	8
Slovar strokovnih izrazov	8
Literatura	8

**Nevronske mreže z vzvratnim razširjanjem napak v funkcijskem
programskem jeziku**

POVZETEK

Angleški prevod slovenskega naslova dela

ABSTRACT

Math. Subj. Class. (2010):

Ključne besede:

Keywords:

1. UVOD

Človeški možgani so kompleksen organ predvsem zaradi nešteti funkcij, ki jih opravljajo. In zgolj vprašanje časa je bilo, kdaj bodo znanstveniki skoraj neskončne zmožnosti možganov prenesli v računalništvo. Ko so poskušali idejo uresničiti, so si predvsem želeli strukture, ki se bo - podobno kot možgani - sposobna učiti, odzivati na spremembe in prepoznati neznane situacije. Tako je 1949 Donal Hebb v svojem delu prvič predstavil idejo umetnih nevronske mrež, kjer se je zgledoval predvsem po možganih. Dvoslojni perceptron se je rodil nekoliko kasneje, leta 1962 je ta pojem prvi omenil Rosenblatt. Z razvojem storjne opreme se je zanimanje za nevronske mreže zopet povečalo v osemdestih letih prejšnjega stoletja. Leta 1986 je več raziskovalcev neodvisno drug od drugega vpelje teorijo o večslojnih nevronske mrežah in vzvratnem razširjanju napake. Kljub širokemu spektru uporabe, ki jih nudijo nevronske mreže je interes zanje kasneje nekoliko upadel. V zadnjih letih pa so nevronske mreže zopet vroča tema, predvsem zaradi njihove zmožnosti prepoznavanja slik. Do danes so umetne nevronske mreže močno napredovale, namesto običajnih mrež se vedno bolj uporabljajo globoke nevronske mreže, ki v nekaterih lastnostih celo prekašajo možgane. Kljub vsemu pa so možgani sposobni masrčičesa, česar računalniki ne bodo nikoli.

Diplomska naloga se poglobi v umetne nevronske mreže z vzvratnim razširjanjem napake. Primer take mreže sem tudi implementirala v funkcijskem programskem jeziku OCaml.

2. UMETNE NEVRONSKE MREŽE

2.1. Biološko ozadje. Kot sugerira že samo ime, se umetne nevronske mreže v marsičem zgledujejo po človeških možganih. Osnovni gradniki možganov so nevroni, ki so med seboj povezani s sinapsami. Prav te goste povezave so ključne za delovanje živčevja. Skozi naše življenje se sinapse spreminjajo po jakosti in številu, ko se učimo se ustvarjajo nove povezave med nevroni, že obstoječe pa postajajo močnejše. Nevron se aktivira le, ko po sinapsi do njega pride signal s točno določeno frekvenco, ta impulz nato nevron posreduje sosednjim nevronom. Človeške možgane sestavlja 10^{10} nevronov, vsak ima približno 10^4 sinaps. Umetne nevronske mreže načeloma sestavlja veliko manj nevronov, posledično so zato pri izvajanju operacij veliko hitreje od možganov. Kot bomo videli v nadaljevanju, je osnovna ideja umetnih nevronske mrež zelo podobna živčevju v možganih. Lahko bi rekli, da so umetne nevronske mreže zgolj abstraktna poenostavitev delovanja možganov.

2.2. Zgradba nevronske mrež. Nevronske mreže so sestavljene iz nevronov, ki matematično gledano niso nič drugega kot funkcije. Nevroni so med seboj povezani, vsaka povezava ima svojo težo oziroma utež. Nevrone razdelimo v sloje, pomembno je poudariti, da nevroni v istem sloju med seboj niso povezani, vsak nevron je povezan le z vsemi nevroni v sosednjem (oziroma prejšnjem sloju). Nevronska mreža lahko vsebuje le vhodni in izhodni sloj - takrat govorimo o dvoslojni nevronske mreži. Če se med tema slojema skriva še kakšen skriti sloj, potem mreža postane večslojna. Število nevronov v posameznem sloju tako kot tudi število skritih slojev nevronske mreže je odvisno od problema, ki ga nevronska mreža rešuje.

Pri mrežah, ki jih bom obravnavala v diplomski nalogi, so vse povezave usmerjene, čeprav je pomembno omeniti, da obstajajo tudi mreže z dvosmernimi povezavami.

Usmerjeni nevronske mreže rečemo perceptron, glede na število slojev pa ločimo dvoslojni in večslojni perceptron.

2.3. Neuron.

2.4. Delovanje perceptrona. Za začetek vzemimo dvoslojni perceptron z m neuroni v vhodnem sloju, njihova stanja označimo z X_1, X_2, \dots, X_m in n neuroni v izhodnem sloju, označimo jih Y_1, Y_2, \dots, Y_n . Običajno sloj nevronov zapišemo v obliki vektorja:

$$X = (X_1, X_2, \dots, X_m)^T \quad \text{in} \quad Y = (Y_1, Y_2, \dots, Y_n)^T.$$

Uteži med neuroni predstavimo z matriko:

$$M = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1m} \\ W_{21} & W_{22} & \cdots & W_{2m} \\ \cdots & & & \\ W_{n1} & W_{n2} & \cdots & W_{nm} \end{bmatrix}$$

2.4.1. Delovanje nevrona. Neuron prejme vhodno stanje, na njem uporabi funkcijo kombinacije in nato še funkcijo aktivacije.

V vsakem sloju neuroni neodvisno izračunajo svoje izhode in te vrednosti posredujejo nevronom v naslednjem sloju

2.5. Lastnosti nevronske mreže. Nevronske mreže odlikujejo tako pozitivne kot negativne lastnosti. Med pozitivne spadajo:

- (1) paralelizabilnost:
- (2) robustnost:
- (3)

Glavna pomanjkljivost nevronske mreže je določanje topologije mreže. Število nevronov v posameznem skritem sloju in število skritih slojev je odvisno od posameznega primera. Idealno topologijo nevronske mreže je tako težko doseči, do nje lahko pridemo le z ugibanjem. Še ena slaba lastnost pa je nezmožnost razlaganja odločitev. Zaradi kompleksne strukture je vsak rezultat mreže produkt večih neodvisnih operacij nevronov, ki jih ne moremo razložiti.

2.6. Uporaba nevronske mreže. Zaradi razvoja strojne opreme v zadnjih letih so močno napredovale tudi nevronske mreže. Poleg tega so bile razvite tudi hitrejše implementacije algoritmov, posledično je uporaba nevronske mreže danes zelo pogosta v industriji. Pomembno je tudi dejstvo, da nevronske mreže lahko rešujejo najrazličnejše probleme, zato se dandanes nevronske mreže porabljajo na skoraj vseh mogočih področjih. Dva najpomembnejša namena uporabe sta regresija in klasifikacija. Področja industrijske uporabe so:

- prepoznavanje in klasifikacija slik: konkretno podjetje Google uporablja nevronske mreže za prepoznavanje slik, ki jih nato opremi s ključnimi besedami
- prepoznavanje ročno napisanih besedil,
- prepoznavanje govora, tu bi veljalo izpostaviti podjetje Microsoft, ki je razvilo nevronske mreže, ki govorjeno angleščino prevaja v kitajščino
- kontrola kvalitete v proizvodnji,
- v zdravstvu se uporabljajo za določanje pacientove diagnoze
- napovedovanje različnih vrednosti glede na trenutne trende.

3. UČENJE

Najprej je smiselno definirati, kaj pojem učenje sploh pomeni. Po [VSTAVI VIR] za učenje potrebujemo sistem, ki strmi k izpolnitvi določene naloge oziroma cilja. Pred učenjem sistem ni sposoben zadostno opraviti naloge. S ponavljanjem določenih opravil poskušamo sistem pripraviti do tega, da bo deloval bolje, kjer je bolje lahko hitreje, ceneje, bolj pravilno... Učenje lahko torej definiramo kot zaporedje ponovitev, kjer pri vsaki ponovitvi skušamo zmanjšati napako tako, da bi se zastavljenemu cilju čimbolj približali.

Obstaja kar nekaj različnih pravil, kako umetno nevronske mreže naučiti pravilnega delovanja. V svoji diplomski nalogi sem se osredotočila na posplošeno pravilo delta oziroma vzvratno razširjanje napake, ki je nekako standarden algoritem pri večslojnih perceptronih.

3.1. Pravilo delta. Pravilo delta se uporablja pri učenju dvoslojnih perceptronov. Če je mreža večslojna, potem govorimo o vzvratnem razširjanju napake. Najprej si pogledajmo kako deluje pravilo delta, saj je vzvratno razširjanje napake samo posplošitev pravila delta.

3.2. Ideja vzvratnega razširjanja napake. Učenje umetne nevronske mreže poteka tako, da iz začetnih poljubno izbranih uteži (pomembno je le, da niso vse uteži nastavljene na 0), uteži popravimo tako, da bo mreža iz vhodnih podatkov znala napovedati izhod. Da mrežo naučimo pravilnega delovanja potrebujemo učne primere - torej vhodne podatke in željene izhode za te vrednosti. Za vzvratnim razširjanjem napake stoji povsem preprosta ideja. Najprej uteži med nevroni poljubno nastavimo, nato na izbranem testnem primeru izračunamo za koliko se je naša mreža zmotila glede na pričakovan izhod. Tako dobimo napako s pomočjo katere lahko vrednosti na utežeh popravimo tako, da minimaliziramo dobljeno napako. To počnemo od izhodnega sloja nevronov proti vhodnemu (od tod tudi ime - vzvratno razširjanje napake oziroma backpropagation v angleščini). Postopek ponavljamo na ostalih testnih primerih dokler se uteži ne ustalijo. Takrat se učenje konča.

3.3. Vzvratno razširjanje napake. Vzemimo splošen večslojen perceptron z X_{N_X} nevroni v vhodnem sloju in Y_{N_Y} nevroni v izhodnem sloju. Nevronska mreža naj sestoji iz m skritih slojev, $m > 0$, vsak skriti sloj pa naj vsa buje N_k nevronov, kjer velja $1 \leq k \leq m$. Nevrone v skritih slojih označimo sledeče: H_{ij} , kjer število i ponazarja sloj v katerem se nevron nahaja, število j pa zaporedno številko nevrona v tem sloju.

Najprej za dani učni primer izračunamo izhodne vrednosti, tako da na vsakem nevronu uporabimo funkcijo aktivacije in kombinacije. Za i -ti nevron v prvem skitem sloju lahko zapišemo:

$$H_{1i}(l) = f \left(\sum_{j=1}^{N_X} W_{ij}^{(1)} X_j(l) \right)$$

Zaradi poenostavitve označimo še:

$$A1i(l) = \sum_{j=1}^{N_X} W_{ij}^{(1)} X_j(l)$$

Za poljubni k -ti skriti sloj, kjer je $1 \leq k \leq m$, tako zapišemo:

$$H_{ki}(l) = f(A_{ki}(l)),$$

kjer je

$$A_{ki}(l) = \sum_{j=1}^{N_{k-1}} W_{ij}^{(k)} H_{k-1,j}(l).$$

V zgornji enačbi $H_{k-1,j}$ označuje že izračunane vrednosti nevronov v predhodnem sloju.

V izhodnem sloju dobimo vrednosti, ki nas najbolj zanimajo. Označimo:

$$Y_i(l) = f(A_i(l)) \quad \text{in} \quad A_i(l) = \sum_{j=1}^{N_m} W_{ij}^{(m+1)} H_{m,j}(l).$$

Tako smo dobili izhodne vrednosti, ki jih je izračunala naša mreža. Vrednosti primerjamo s testimi, ki bi si jih želeli dobiti za dani l -ti učni primer. Za posamezen nevron napako ozančimo sledeče:

$$e_i(l) = d_i(l) - Y_i(l),$$

lahko pa enačbo zapišemo tudi v vektoski obliki:

$$e(l) = d(l) - Y(l).$$

Napako celotne nevronske mreže za l -ti učni primer pa definiramo sledeče:

$$(1) \quad E(l) = \frac{1}{2} \sum_{i=1}^{N_Y} e_i^2(l)$$

Uteži popravimo v smeri negativnega odvoda napake E , saj želimo napako minimalizirati. Tu gre za gradientno iskanje. Nove uteži izračunamo po sledeči formuli:

$$(2) \quad W^{NOV} = W - \eta \frac{dE}{dW},$$

kjer η označuje stopnjo učenja. Pri izračunu zgornjega odvoda si bomo pomagali s verižnim pravilom za odvajanje. Odvod enostavno izračunamo za izhodni sloj. Problem se pojavi pri popravljanju uteži v skritih slojih, saj uteži niso direktno povezane z napako. Kot bomo videli v nadaljevanju, je tudi tu rešitev verižno pravilo.

Najprej razpišimo formulo (1), kjer upoštevamo formuli za e_i in Y_i :

$$\begin{aligned} E(l) &= \frac{1}{2} \sum_{i=1}^{N_Y} e_i^2(l) = \frac{1}{2} \sum_{i=1}^{N_Y} (d_i(l) - Y_i(l))^2(l) \\ &= \frac{1}{2} \sum_{i=1}^{N_Y} (d_i(l) - f(\sum_{j=1}^{N_m} W_{ij}^{(m+1)} H_{m,j}(l)))^2(l) \end{aligned}$$

Sedaj za izračun odvoda uporabimo verižno pravilo, formulo razpišemo na posamezno utež:

$$(3) \quad \frac{dE(l)}{dW_{ij}} =$$

3.4. Lastnosti vzvratnega razširjanja napake. Kljub temu, da imajo nevronske mreže veliko pozitivnih lastnosti, ima pravilo vzvratnega razširjanja napake tudi nekaj pomanjkljivosti.

4. FUNKCIJSKI PROGRAMSKI JEZIK OCAML

5. PRAKTIČNI DEL

6. ZAKLJUČEK

SLOVAR STROKOVNIH IZRAZOV

LITERATURA

[1]