

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni nastavnički studij matematike i informatike

Sanja Topuzović

Neuronske mreže za predviđanje cijena nekretnina

Diplomski rad

Osijek, 2011.

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni nastavnički studij matematike i informatike

Sanja Topuzović

Neuronske mreže za predviđanje cijena nekretnina

Diplomski rad

Voditelj/ica: Prof. dr. sc. Marijana Zekić-Sušac

Osijek, 2011.

Sadržaj

Uvod	2
1. Svrha i ciljevi istraživanja	3
2. Teorijski koncept umjetne inteligencije i neuronskih mreža	4
2.1. Umjetna inteligencija	4
2.2. Razvoj i pojam umjetne inteligencije	5
2.3. Tehnike umjetne inteligencije	6
2.4. Područja primjene umjetne inteligencije	6
3. Prethodna istraživanja	8
4. Metodologija neuronskih mreža	11
4.1. Pojam i povijesni razvoj neuronskih mreža	11
4.2. Princip rada neuronske mreže	14
4.3. Vrste neuronskih mreža	16
4.4. Prednosti i nedostaci neuronskih mreža	23
4.5. Arhitektura i način rada neuronske mreže višeslojni perceptron	24
4.6. Računanje u mreži	27
5. Opis podataka	32
5.1. Ulazne i izlazne varijable	32
5.2. Izrada neuronske mreže	34
6. Rezultati	35
6.1. Rezultati u fazi učenja i testiranja mreže	35
6.2. Analiza uspješnosti neuronske mreže	39
6.3. Analiza osjetljivosti	40
7. Zaključak	42
Literatura	44
Sažetak	46
Summary	47
Životopis	48

Uvod

Neuronske mreže su relativno novi koncept u analizi podataka, a pod pojmom neuronske mreže podrazumjevaju se umjetne neuronske mreže.

Razvoj neuronskih mreža ne seže daleko u prošlost, ali metode izgrađene na njima nalaze široku primjenu u tehničkim znanostima, ekonomiji i u društvenim znanostima. Primjena u ekonomiji uglavnom je vezana za predviđanje budućih vrijednosti koje se promatraju, a mogu se odnositi npr. na predviđanje ponude i potražnje, za predviđanje cijena akcija na određeni dan, predviđanje cijena nekretnina (ovaj slučaj obuhvaćen je konkretnim primjerom u ovom radu), i sl. Što točnije predviđanje od važnog je značaja za poslovanje svih subjekata. Uspješnost neuronskih mreža osim za potrebe predviđanja značajna je upotreba u svrhu klasifikacije.

Neuronske mreže predstavljaju jednu od metoda umjetne inteligencije, a koriste se matematičkom formom i strukturom ljudskoga mozga kako bi razvile strategiju procesiranja podataka. U radu će biti opisani osnovni pojmovi i principi metode neuronskih mreža (od povijesnog razvoja, biološke osnove do same strukture).

Dan je i opis prethodnih istraživanja u tom području, te je kreiran model za predviđanje cijena nekretnina s pomoću neuronskih mreža. Opisani su rezultati i zaključak sa smjernicama za daljnja istraživanja.

1. Svrha i ciljevi istraživanja

Svrha ovog rada je popularizirati metode umjetne inteligencije, posebice neuronskih mreža kao prediktivne metode. Cilj za ostvarenje svrhe je kreirati model temeljen na metodi neuronskih mreža koji će omogućiti predviđanje cijena nekretnina na području Bostona, analizirati točnost modela u smislu greške, te značajnost ulaznih varijabli kako bi se ukazalo na važne prediktore za određivanje cijena nekretnina.

Cilj će biti ostvaren: primjenom metode na uzorku od 506 promatranja testiranjem algoritma višeslojni perceptron (eng. multilayer perceptron, MLP). Kreirano je 25 arhitektura neuronskih mreža. Na temelju 13 ulaznih varijabli dobiveni su rezultati predviđanja cijena nekretnina na području Bostona, od kojih smo izabrali kao najuspješniji model s najmanjom MSE greškom pomoću koje će se vršiti predviđanje.

Nakon provedenog istraživanja dobivena je najuspješnija arhitektura sa 17 neurona u skrivenom sloju koja producira MSE grešku 0.002019.

Analiza osjetljivosti, provedena nakon treniranja, ukazuje na jačinu utjecaja pojedine varijable na cijenu nekretnina, čime je moguće donijeti zaključak o tome koja varijabla ima najveći utjecaj na cijenu nekretnine, što može biti korisna informacija donositeljima odluke u području primjene modela.

2. Teorijski koncept umjetne inteligencije i neuronskih mreža

Do danas nema opće prihvaćene definicije što je to "umjetna inteligencija", vjerojatno, zato što ni sam pojam "inteligencije" nije jednoznačno definiran. Postoji nekoliko definicija umjetne inteligencije (UI) (eng. Artificial intelligence):

Prema McCarty-u [19] umjetna inteligencija (UI) je znanost i praksa kreiranja inteligentnih strojeva, te posebno inteligentnih računalnih programa, dok je najčešća definicija inteligencije - da je to sposobnost snalaženja jedinke u novim situacijama. [19]

2.1. Umjetna inteligencija

Kroz povijest ljude privlači istraživanje inteligencije, te mogućnost konstruiranja, "inteligentnih strojeva" koji bi samostalno radili te obavljali poslove umjesto nas (ljudi). UI ima temelje u nekoliko znanstvenih disciplina kao što su psihologija, filozofija, matematika i druge, koje su utjecale na njezin razvoj.

Iako je UI novija grana znanosti, mlado područje računarstva, utjecaji na nju sežu daleko u prošlost - još u antičku filozofiju, kada je Aristotel prvi pokušao kodirati "pravilno razmišljanje". Tako su se razmišljanja iz 4.st.pr.Kr. tijekom povijesti nadograđivala i proširivala pa imamo danas stručnjake koji su od razmišljanja o umu došli do kreiranja živog uma tj. umjetne inteligencije (eng. Artificial intelligence, AI). 1943. Warren McCulloch i Walter Pitts napisali su prvi rad koji govori o UI. Također su predložili model umjetnog neurona, kao što su i pokazali da svaka kontinuirana funkcija može biti aproksimirana mrežom povezanih neurona, te da prikladno definirane mreže mogu na taj način učiti na temelju prošlih podataka. Konkretni pomak donosi 1950. Alan Turing koji je izložio kompletnu viziju UI u članku "Mind"; u časopisu "Computing Machinery and Intelligence". [14]

On daje ideju konstrukcije inteligentnog računala programiranjem, a ne strojnom arhitekturom. Turingov stroj privlači pažnju zbog funkcionalnog ustroja određenog s tri parametra: ulazom, stanjem i izlazom, a to su parametri kojima bismo mogli odrediti i funkcioniranje ljudskog organizma: vanjski podražaji, unutrašnja stanja i tjelesno ponašanje.

Turing je predložio test kojim se nastoji provjeriti je li neki stroj inteligentan. Turingov test (1950.) potiče raspravu hoćemo li ikada moći reći da je stroj inteligentan, odnosno da može razmišljati. Test se sastoji od tri sudionika, a to su sudac-osoba koja komunicira s nekim ili nečim u dvije susjedne sobe putem pisanih poruka ili ekrana. U jednoj sobi je čovjek, a u drugoj računalo. Postavljanjem pitanja sudac-osoba mora utvrditi

u kojoj je sobi čovjek, a u kojoj računalo. Ako sudac-osoba nakon testa ne može reći u kojoj je sobi računalo, a u kojoj je čovjek, onda se računalo pripisuje inteligencija - smatra se inteligentnim. Ograničenje Turingovog testa je što se inteligencija očituje u odnosu na kontekst tj. ovisi o sredini sudionika, njihovom raspoloženju, subjektivnosti suca itd.

Najpoznatija kritika testa je eksperiment nazvan Kineska soba, a osmislio ga je John Searle. Metafora kineske sobe govori o mogućnosti davanja točnih odgovora bez razumijevanja ako su računalo na raspolaganju pravila i upute - argument protiv tvrdnje "program je sve što je potrebno za inteligenciju". 1956. na Dartmouthu, John McCarthy, Minsky i suradnici organiziraju dvomjesečnu radionicu kako bi proučavali neuronske mreže i UI. Značaj te radionice je taj da je na njoj umjetna inteligencija proglašena znanstvenom disciplinom, te McCarthyjev novi naziv za to polje: **umjetna inteligencija**. [1], [3], [14]

2.2. Razvoj i pojam umjetne inteligencije

UI se fokusirala na osnove inteligencije što je učenje, zaključivanje, rješavanje problema, percepciju i razumijevanje jezika. Razvoj UI možemo podijeliti na razdoblja: [14] (1943 - 1956) Faza inkubacije

(1952 - 1969) Faza ranog entuzijazma i velikih očekivanja

(1969 - 1979) Faza sustava temeljenih na znanju

(1980 - do danas) Faza kada UI postaje industrija

UI je znanost koja se bavi proučavanjem kako računalo učiniti sposobnim da obavlja neke poslove koje ljudi u ovom trenutku obavljaju bolje. Pripada u područje računalnih znanosti koji ima: [14],[19]

- akademski cilj - (proučavanje i razumijevanje inteligencije)
- praktični cilj - (bolje iskorištavanje računala).

Prema D.W. Patersonu iz 1990. [12] "Umjetna inteligencija je grana računarske znanosti koja se bavi proučavanjem računarskih sustava koji pokazuju neki oblik inteligencije. To su sustavi koji mogu učiti nove koncepte, sustavi koji mogu zaključivati i donositi uporabne zaključke o svijetu, koji ih okružuje, sustavi koji mogu razumijeti prirodni jezik ili spoznati i tumačiti složene vizualne scene, sustavi koji mogu obavljati i druge vrste vještina koje zahtijevaju čovjekovu vrstu inteligencije."

2.3. Tehnike umjetne inteligencije

Područja UI (prema Association of Computing Machinery):[1],[3]

- **Ekspertni sustavi** - su programski produkti namijenjeni rješavanju složenih problema u uskoj *domeni primjene*. Temeljni sadržaj ekspertnih sustava je: predstavljanje i obrada znanja.
- **Formalizmi i metode prikaza znanja** - prikaz znanja se odnosi na prevođenje ili kodiranje znanja u format pogodan za rad računala, postoje brojne tehnike za prikaz znanja, a koju ćemo u konkretnom slučaju upotrijebiti ovisi o promatranom području znanja.
- **Strojno učenje** - odnosi se na izgradnju računalnih sustava koji automatski svoje performanse poboljšavaju kroz iskustvo.
- **Razumijevanje i obrada prirodnih i umjetnih jezika** - prevođenje
- **Rješavanje problema i metode pretraživanja** - metode pretraživanja u dubinu i u širinu, a metode se razlikuju po načinu na koji se obilaze čvorovi.
- **Robotika** - je znanstveno-tehnička disciplina, čiji je cilj izrada i unapređivanje robota, automatiziranih strojeva kojima upravljaju elektronička računala, pomoću upravljačkog programa i informacija primljenih preko elektroničkih osjetila - **senzora**.
- **Računalni vid, raspoznavanje uzorka i analiza scene** - područje umjetne inteligencije koje se bavi prepoznavanjem dvodimenzionalnih i/ili trodimenzionalnih predmeta na primjer, ljudskog lica. Bez razvijenog računalnog vida robot se ne može snalaziti u prostoru, što znači da može biti potencijalno opasan u slučaju ljudske prisutnosti u istom području.
- **Umjetne neuronske mreže, genetski algoritmi i fuzzy logika** - približno izračunavanje (eng. soft computing) *Umjetne neuronske mreže* - stvaranje modela postupnim učenjem na prethodnim podacima *Genetski algoritmi* - traženje najprikladnijeg rješenja prema uzoru na evolucijske procese. *Neizrazita logika* (eng. *fuzzy logic*) - pripadnost nekom skupu nije jednostavno izražena s 0 i 1 nego s vjerojatnošću koja može biti iz intervala (0,1).

2.4. Područja primjene umjetne inteligencije

Neka područja primjene UI su [14]:

- **Igranje igara** : Deep Blue (IBM) prvi računarski program koji je pobijedio svjetskog prvaka Garrya Kasparova

- **Dijagnoza** : program za medicinsku dijagnozu baziranu na vjerodostojnoj analizi (bio sposoban opisati slučaj)
- **Robotika** : kirurzi koriste asistenta robota u mikrooperacijama
- **Razumijevanje jezika i rješavanje problema** : program koji rješava križaljke bolje od ljudi, velika baza prijašnjih križaljki, rječnika, ELIZA programa za komunikaciju između čovjeka i računala
- **Samoupravno planiranje i raspored** : NASA program udaljenog agenta za svemirske brodove
- **Samoupravna kontrola** : program treniran za upravljanje autom duž ceste, smješten u kombiju i korišten za navigaciju.

Kako je znanost o neuronskim mrežama još u fazi intenzivnog razvoja svakodnevno se razvijaju i usavršavaju novi algoritmi i arhitekture NM, te se tako proširuju i mogućnosti primjene. Nova područja UI stalno se razvijaju i usavršavaju.

3. Prethodna istraživanja

Najviše istraživanja koja koriste NM provodilo se u području predviđanja poslovnih poteškoća. Metode i modeli iz različitih znanstvenih područja i grana apliciraju se kako bi svi zainteresirani za to područje dobili informaciju o tome kreće li se određeno trgovačko društvo prema stečaju, te kakvi su izgledi za njegova buduća poslovanja.

U nastavku će biti opisani neki radovi koji su pokazali uspješnost NM u rješavanju različitih poslovnih problema.

Zhang G. i dr. [20], u svom radu predstavljaju opći okvir za razumijevanje uloge neuronskih mreža (ANNs) u stečajnom predviđanju. Autori daju i sveobuhvatan pregled neuronskih mreža u ovom području i ilustrirati vezu između neuronskih mreža i tradicionalnog Bayesova teorije klasifikacije. Metoda unakrsne validacije (eng. cross-validation) se koristi za ispitivanje među-varijacija neuronskih mreža za predviđanje stečaja. Na temelju odgovarajućih uzoraka od 220 firmi, predloženi nalazi pokazuju da su neuronske mreže značajno preciznije od logističkih regresijskih modela u predviđanju, kao i klasifikacijskoj stopi procjene.

Zekić-Sušac i dr. [18], u svom radu kreiraju model za predviđanje uspješnosti studenata s pomoću neuronskih mreža i klasifikacijskih stabala odlučivanja, te analizom čimbenika koji utječu na uspješnost studenata. Kreiran je model koji na temelju demografskih podataka o studentima, te podacima o njihovom ponašanju i stavovima prema učenju nastoji klasificirati studenta u jednu od dviju kategorija uspješnosti. Uspješnost je mjerena prosjekom ocjena na studiju. Trenirano je i testirano više različitih arhitektura neuronskih mreža, čiji je najbolji model dobiven s pomoću višeslojne perceptron mreže. Stabla odlučivanja dala su znatno veću točnost klasifikacije od neuronskih mreža, te ih se predlaže koristiti kao točniju metodu na promatranom skupu podataka. Analiza osjetljivosti izlaznih varijabli na ulazne provedena kod neuronskih mreža upućuje da su kolokviranje, prisustvo na vježbama, važnost ocjene za studenta, te stipendije među najznačajnijim čimbenicima uspješnosti studenta. Stabla odlučivanja izlučila su vrijeme provedeno u učenju, prisustvo na vježbama, te vrstu materijala iz kojih se uči kao najznačajnije varijable. U budućim istraživanjima, uz proširenje broja ulaznih varijabli i povećanje uzorka, te proširenje metodologije drugih tehnikama umjetne inteligencije i statističkim metodama, moguće bi bilo kreirati uspješniji model koji bi bio osnova za izgradnju sustava za potporu odlučivanju u visokom obrazovanju.

Ravi V. i dr. [13], u svom su radu istraživali popularno područje predviđanja poslovnih poteškoća i u krajnjemu, stečaja. Ovaj članak prezentira ključne točke u

razvoju ovog područja ekonomije, ističući određene relevantnije autore i njihova istraživanja, s posebnim osvrtom na Republiku Hrvatsku i domaće radove. Naglasak je stavljen na mješoviti logitmodel i neuronske mreže kojih se veća primjena u Hrvatskoj tek očekuje. Područje istraživanja poslovnih poteškoća bogato je različitim metodama i modelima, no za njihovu širu primjenu u Hrvatskoj postoje određene prepreke. Prije svega, to je educiranost istraživača na području ekonometrije, zatim je tu nepristupačnost podataka - dobiti veći broj financijskih izvješća za veći raspon godina uz razuman trošak često je neizvedivo. Konačno, kada postoji i educiranost istraživanja i kada se izvješća napokon pribave, uočava se kako su ulazni podaci najčešće nevjerodostojni, pa otvorenje stečaja nije obvezno posljedica lošeg poslovanja. Zbog toga se ovo područje Hrvatskoj još uvijek pretežno temelji na analizi kvalitativnih varijabla kao što su kompetencije menadžmenta, utjecaj i značenje poslovanja na lokalnu zajednicu, snaga političkih utjecaja, i drugih.

Tonković Z. i dr. [15], u svom radu kreiraju model predviđanja potrošnje prirodnog plina na regionalnoj razini koristeći neuronske mreže, te analiziraju rezultate s ciljem unapređivanja točnosti predviđanja u budućim istraživanjima. Izlazna varijabla sastojala se od potrošnje prirodnog plina sljedećeg dana u satnim intervalima, dok je ulazni varijabli bilo 43 a u njih je spadala potrošnja prethodnog dana, te dodatne egzogene varijable. Nakon procedure selekcije značajnih varijabli, testirana su dva algoritma neuronskih mreža: višeslojni perceptron i mreža s radijalnom funkcijom koristeći različite aktivacijske funkcije. Skup podataka (ukupno 454 dana) sastojao se od stvarnih povijesnih podataka jednog hrvatskog distributera plina. Na temelju srednje apsolutne postotne greške dobivene na testnom uzorku izabran je najbolji model neuronske mreže. Rezultat najbolje mreže daje MLP algoritam (greška 9,36%), s tanges-hiperbolnom aktivacijskom funkcijom koja daje najmanju MAPE 10% dok je MSE greška 0,0865. Mreža je trenirana s maksimalnim bojem skrivenih neurona. Iznesene su određene smjernice koje mogu biti korisne za istraživače i praktičare u ovom području.

Vukomanović M. i dr. [16], u svom radu analiziraju primjenu neuronskih mreža prilikom procjene cijene montažne gradnje na više od 30 objekata. Identificirano je 17 varijabli pomoću kojih se može predvidjeti konačna cijena: 83,8 % predviđenih vrijednosti nalazilo se u granicama odstupanja do 5 % , a 12,9 % u granicama od 5-10 % od stvarnih vrijednosti. Model je verificiran na 28 i validiran na 3 nova objekta, s prosječnim odstupanjem od 4,6 % od stvarnih vrijednosti. Članak predstavlja model pomoću kojeg građevinska poduzeća, još u fazi nudičenja, mogu efektivno predviđati konačnu cijenu motažne gradnje. U zaključku su dane smjernice primjene modela u praksi i područja daljnjeg istraživanja.

Iz navedenog pregleda prethodnih istraživanja može se zaključiti da u većim slučajevima

neuronske mreže pokazuju veću ili jednaku točnost od statističkih modela. Izbor algoritma ili mreže ovisi o podacima, ali većina istraživanja koristi MLP algoritam, stabla odlučivanja, genetske algoritme ili neke druge inteligentne metode.

4. Metodologija neuronskih mreža

Neuronske mreže (NM) su metoda umjetne inteligencije razvijene prema modelu mreže bioloških neurona. To su programi ili hardverski sklopovi koji iterativnim postupkom iz prošlih podataka nastoje pronaći vezu između ulaznih i izlaznih varijabli modela kako bi se za nove ulazne varijable dobila vrijednost izlaza, tj. uče na primjerima.[19]

4.1. Pojam i povijesni razvoj neuronskih mreža

Umjetnoj inteligenciji cilj je stvaranje umjetnog ljudskog mozga, u modelu mozga u kojem se podaci uz pomoć brojnih procesnih elemenata obrađuju paralelno. Područje računarstva koje proučava takvu obradu informacija zovemo neuro-računarstvo, koje se pojavljuje kao alternativa van Neumann-ovim računalima (jer omogućavanju rješavanje nekih problema koji se drugim tehnologijama ne mogu riješiti) nastoje simulirati ili ostvariti paralelnu obradu informacija koju koristi ljudski mozak dok rješava probleme, razmišlja i sjeća se.

To potiče i istraživanja na području arhitekture računala jer se konvencionalna van Neumann-ova arhitektura temelji na sekvencijalnoj obradi podataka koja nema puno zajedničkog s načinom funkcioniranja i strukturom ljudskog mozga. No, van Neumannova računala ipak se mogu koristiti pri implementaciji umjetne NM, odbacujući onda formalizam rješavanja problema putem algoritma, odnosno manipulacije simbolima po definiranim pravilima.[9]

Karakteristike koje razlikuju te arhitekture dane su u Tablici 1.1. :

Von Neumannovo računalo	Neuro - računalo
računalu se unaprijed detaljno mora opisati algoritam u točnom slijedu koraka (program)	neuronska mreža uči samostalno ili s učiteljem
podaci moraju biti precizni - nejasni ili neizraziti podaci se ne obrađuju adekvatno	podaci ne moraju biti precizni (gotovo uvijek su neprecizni)
arhitektura je osjetljiva - kod uništenja nekoliko memorijskih ćelija računalo ne funkcionira	obrađa i rezultat ne mora puno ovisiti o pojedinačnom elementu mreže
postoji eksplicitna veza između semantičkih objekata (varijabla, brojeva, zapisa u bazi) i sklopovlja računala preko pokazivača na memoriju	znanje je implicitno pohranjeno ali ga je teško interpretirati

Tablica 1.1. *Razlike u arhitekturi von Neumannova i neuro-računala* [2]

Za razvoj NM nekoliko događaja je od presudnog značaja [12], [14]:

1943.g. - Warren McCullock i Walter Pitts objavili su članak "*Logički račun ideja svojstvenih nervnoj aktivnosti*" često citiran u literaturi, a kojeg su utvrdili na tri izvora, osnovnom znanju iz filozofije i psihologije, funkcioniranju neurona u mozgu, formalnoj i propozicijskoj logici, te Turingovoj teoriji računanja. Predložili su model umjetnog neurona gdje je neuron osnova i može imati jedno od dva stanja, pobuđujuće ili umirujuće, i da njihova aktivnost ovisi o nekom pragu vrijednosti. Kao i da bilo koja računalna funkcija može biti prikazana pomoću mreže neurona. Sugeriraju i da prikladno definirane mreže mogu učiti, te time postavljaju temelje za razvoj neuronskih mreža.

1949.g. - Donald Hebb objavljuje knjigu "*Organizacija ponašanja*" u kojoj predlaže određen zakon učenja za sinapse, odnosno demonstrira pravilo za mjenjanje jačine veza između neurona, koje je danas poznato kao Hebb-ovo pravilo.

1951.g. - Marvin Minsky i Dean Edmonds konstruiraju prvo neuroračunalo pod imenom *SNARC*, koji je koristio vakumske cijevi za simulaciju neuronske mreže od 40 neurona.

1956.g. - Na Dartmouth Summer Conference predstavili su Rochester i suradnici prvu simulaciju Hebb-ovog modela koji je preteča modela neuronskih mreža; generaliziraju model koji uključuje umirujuće stanje da pobuđeni neuroni umiruju ostale.

1958.g. - Frank Rosenblatt razvio prvu NM "*Perceptron*" koja je dvoslojna i potpuno povezana; proučavaju različite načine učenja pa uočavaju da učenjem u dva sloja nije moguće riješiti probleme klasifikacije koji nisu linearno djeljivi (npr. XOR). Frank Rosenblatt i Charles Wightmann sa svojim suradnicima uspjeli su razviti računalo *MARK1*, koji predstavlja prvo neuroračunalo.

1960.g. - Bernard Widrow poboljšava Hebb-ovu metodu učenja, te razvija praktičnu uporabu umjetnih NM. Zajedno sa svojim studentima (najpoznatiji Ted Hoff) razradio je novi jednostavni tip "neurona" kojeg naziva *ADALINE* (ADAPtive Linear Neuron) i odgovarajući tip učenja.

1969.g. - Minsky i Papert objavljuju knjigu "*Perceptrons*" u kojoj se kritički osvrću na Perceptron, dajući matematički dokaz da NM "Perceptron" ne može riješiti XOR problem, ni dodavanjem više slojeva neurona. Rezultat toga je prestanak financiranja u istraživanju NM.

1967.-1982.g. - Pojavljuju se istraživači koji imaju utjecaj na razvoj u ovom području provodeći razna istraživanja. Teuvo Kohonen Anderson, Grossberg i dr., a služe se Hebbovim modelom korelativnog učenja. U ovom razdoblju pojavio se i algoritam "širenja unatrag" (backpropagation) jedan od najvažnijih događaja u istraživanju NM. Algoritam je otkriven barem od četiri različite grupe znanstvenika (Bryson i Ho 1969.) Paul Werbos (1974.) prva verzija mreže "širenja unatrag" tkz. višeslojna perceptron mreža, MLP: uvodi učenje u skrivenom sloju i tako prevladava nedostatak perceptora. Početkom 80-tih američka vojska počinje ulagati u NM pa istraživanja opet počinju.

1986.g. - Rumelhart, Hinton i Williams usavršavaju algoritam "širenja unatrag", što vraća ugled NM jer omogućava aproksimiranje gotovo svih funkcija i rješavanje praktičnih problema; ova godina također se smatra godinom povratka NM, opet raste interes za ovo područje.

Do danas znanstveni interes za NM ne prestaje rasti, tako da su razvijeni brojni algoritmi za NM, koji pomoću različitih pravila učenja, ulaznih i izlaznih funkcija rješavaju različite probleme. NM postale su nezaobilazan koncept pri razvoju inteligentnih sustava. Unatoč svojoj povijesti NM su još u ranoj fazi razvoja, a danas nalaze primjenu u raznim područjima.

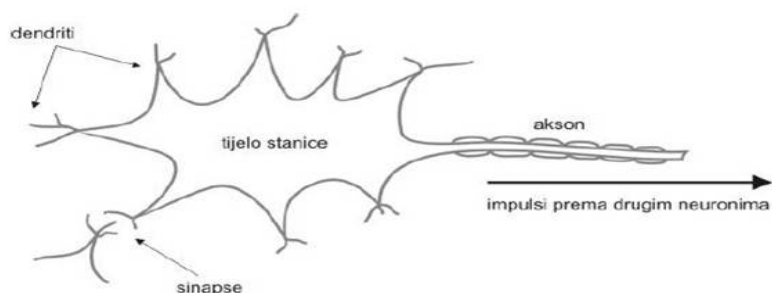
4.2. Princip rada neuronske mreže

Ljudski mozak je primjer biološke NM, a on se sastoji od velikog broja međusobno povezanih živčanih stanica ili neurona. Biološki neuron sastoji se od tijela ili soma, aksona i mnoštva dentrita. Neuroni putem dentrita primaju informacije od drugih dentrita, a preko aksona predaju signale koji proizvodi tijelo (soma) neurona. Odnos neurona i jačina njihovih međusobnih veza oblikuju djelovanje biološke NM. Jačine veza između bioloških neurona mijenjaju se pa tako živa bića uče. Rad neurona u mreži je paralelan i raspodjeljen, što je jako važno svojstvo koje umjetne NM nastoje postići. Struktura biološkog neurona prikazana je na *Slici 2.1*. Umjetna NM razvijena je po modelu biološke NM, sastavljena od jedinica za obradu podataka (vanjskih) koje stoje u određenoj interakciji tako da grade funkcionalnu cjelinu. *Slika 2.2*.

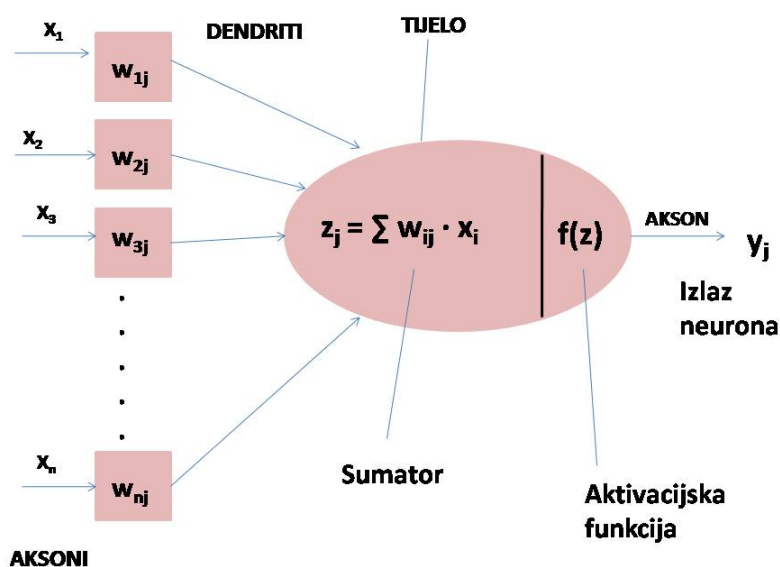
Tijelo biološkog neurona zamjenjuje sumator, a akson umjetnog neurona je izlaz iz sumatora, dok aktivacijske funkcije imaju ulogu osjetljivosti bioloških neurona. Neuroni su povezani u mrežu tako da izlaz svakog neurona predstavlja ulaz u jedan ili više neurona. Veze između neurona možemo grupirati po smjeru (jednosmjerne ili dvosmjerne) i intenzitetu (pobuđujuće ili smirujuće) veze.

Neuroni u mreži organizirani su u slojeve, a postoje tri osnovna tipa slojeva koji

moгу biti ulazni, skriveni i izlazni. Ulazni sloj prima ulazne podatke iz okoline te ih šalje u skrivene slojeve, a tamo se informacije obrađuju te se šalju u neurone izlaznog sloja. Informacije potom putuju unazad kroz mrežu, a vrijednosti težina veza između neurona prilagođavaju se prema željenom izlazu. Proces se ponavlja u mreži onoliko iteracija koliko je potrebno za dobivanje izlaza koji je najbliže željenom (stvarnom) izlazu, koji se na kraju predstavlja korisniku. Snaga veze između dva neurona predstavlja težinu veze, a proces kojim pronalazimo vrijednosti težina veza (snaga veza) među neuronima je učenje (notacija: signal x_j na ulazu j neurona κ ima težinu $w_{\kappa,j}$). Analogija između umjetnog (biološkog) modela nije toliko čvrsta jer postoje karakteristike umjetnih NM koji se ne slažu s onim biološkog sustava, i obratno. [14], [17], [19]



Slika 2.1. *Struktura biološkog neurona* [2]



Slika 2.2. *Struktura umjetnog neurona* [2], modificirano

4.3. Vrste neuronskih mreža

Korištenje NM u različitim znanstvenim područjima rezultiralo je velikim brojem različitih vrsta NM. Proces dizajniranja NM sastoji se od četiri faze:

1. raspoređivanje neurona u različite slojeve
2. određivanje tipa veze između neurona
3. određivanje načina na koji neuroni primaju ulaze i proizvode izlaze
4. određivanje pravila učenja za prilagođavanje težina veza.

Ovisno o formulama koje se koriste za učenje, ulazne i izlazne funkcije, postoje različiti algoritmi NM, a unutar svakog algoritma moguće su promjene u strukturi mreže i izbora parametra učenja pa postoji široki spektar arhitektura NM koje su rezultat dizajniranja NM.

Međusobno se razlikuju prema kriterijima [17]:

- broj slojeva
- tip veze između neurona
- veza između ulaznih i izlaznih podataka
- ulazne i prijenosne funkcije
- tip učenja
- sigurnost ispaljivanja
- vremenske karakteristike
- vrijeme učenja.

Od navedenih karakteristika od osobitog značaja za razlikovanje algoritma je **pravilo učenja** - na koji se način podešavaju težine u mreži, no krenimo redom.

Arhitekture NM prema broju slojeva mogu biti:

- ***dvoslojne*** - samo s ulaznim i izlaznim slojevima i
- ***višeslojne*** - s jednim ili više dodatnih slojeva, a broj skrivenih slojeva treba eksperimentalno odrediti, a često je potrebno više od jednog skrivenog sloja za aproksimaciju složenih nelinearnih funkcija.

Veze između neurona u mreži mogu biti:

1. *inter-slojne veze* između neurona u različitim slojevima, a mogu biti potpuno povezane, djelomično povezane, dvosmjerne, hijerarhijske, unaprijed i rezonantne; primjeri takvih mreža su ADALINE, Perceptron, "širenje unatrag" itd.
2. *intra-slojne veze* između neurona u istom sloju, a mogu biti ponavljajuće i on-center/off-surround; primjeri takvih mreža su ART1, ART2, ART3, ponavljajuća mreža "širenje unatrag", Hopfield-ova mreža.

Veze između ulaznih i izlaznih podataka mogu biti:

- ***autoasocijativne*** - (ulazni vektor je istog sastava kao i izlazni)
- ***heteroasocijativne*** - (izlazni vektor se razlikuje od ulaznog vektora)

Vrste ulaznih i prijenosnih funkcija koje se koriste kod NM su:

1. Ulazne (sumacijske) funkcije zovemo funkcije prema kojima se računaju vrijednosti ulaza koje prima neki neuron iz prethodnog sloja, a računa se, kada neuron i prima ulaz od neurona j prema:

$$input_i = \sum_{j=1}^n (w_{jn} \times output_j) \quad (4.1)$$

gdje se $input_i$ nekog neurona i računa kao suma svih vaganih ulaza koji pristižu u taj neuron, n broj neurona u sloju koji šalju svoj izlaz primljen od neurona i , w_{jn} težina veze između neurona i i neurona j , a $output_j$ predstavlja izlaz poslan od strane neurona j prema neuron i . Uvjet aktivacije neurona je

$$\sum_{j=1}^n (w_{jn} \times output_j) \geq T_i \quad (4.2)$$

T je "prag" (eng. threshold) čija je vrijednost dogovorena. Osim ovog standardnog mrežnog ulaza (4.1) postoje još dva dodatna tipa ulaza u nekoj mreži:

- *extinput_i* - ulaz koji neurona i prima iz vanjske okoline
- *bias_i* - vrijednost pojačanja upotrebljavana za kontrolu aktivacije neurona u mrežama.

Ulazne vrijednosti mogu se normalizirati na neki interval (obično $[0, 1]$ ili $[-1, 1]$) kako bi se izbjegao utjecaj ulaza s visokim vrijednostima.[11], [14], [17], [19]

2. Izlance (prijenosne) funkcije

Nakon što primi ulaz neuron šalje svoj izlaz drugim neuronima s kojima je povezan (uobičajeno neuronima u sljedećem sloju),

$$output_i = f(input_i - T) \quad (4.3)$$

Neuron izračunava težinski zbroj ulaznih vrijednosti, koristeći dogovorenu vrijednost "praga" T , kroz funkcijski izlaz. Neke od prijenosnih funkcija su:

– **linearna funkcija**

$$output_i = g \cdot input_i \quad (4.4)$$

– **linearna funkcija s pragom**

$$output_i = \begin{cases} 0, & \text{ako je izlaz } input_i \leq T; \\ input_i - T, & \text{ako je izlaz } input_i > T. \end{cases} \quad (4.5)$$

neuron ima vrijednost različitu od nule samo ako njegov ulaz dostigne vrijednost praga T , $T \in \mathbb{R}$.

– **funkcija koraka (step funkcija)**

$$output_i = \begin{cases} 0, & \text{ako je izlaz } input_i \leq T; \\ 1, & \text{ako je izlaz } input_i > T. \end{cases} \quad (4.6)$$

gdje je $T \in \mathbb{R}$.

– **signum funkcija**

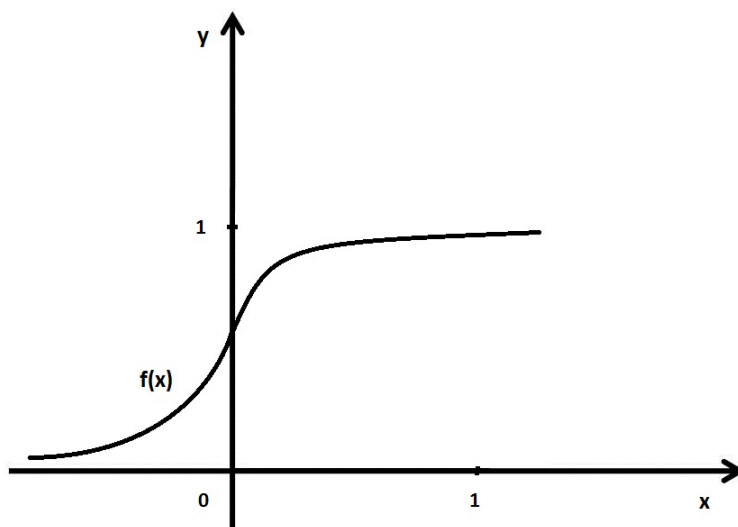
$$output_i = \begin{cases} 1, & \text{ako je } input_i > 0; \\ 0, & \text{ako je } input_i = 0; \\ -1, & \text{ako je } input_i < 0;. \end{cases} \quad (4.7)$$

Upotrebljavla se u prvoj NM Perceptronu, a poseban je oblik step funkcije kada je prag $T = 0$.

– **sigmoidna (logistička) funkcija** - najčešće upotrebljavana prijenosna funkcija u NM, a formula glasi:

$$output_i = \frac{1}{1 + e^{g \cdot input_i}} \quad (4.8)$$

gdje je g doprinos funkcije, a računa se kao $g = \frac{1}{T}$ Doprinos g određuje nagib funkcije oko nule, što rezultira kontinuiranim vrijednostima u intervalu $[0,1]$;



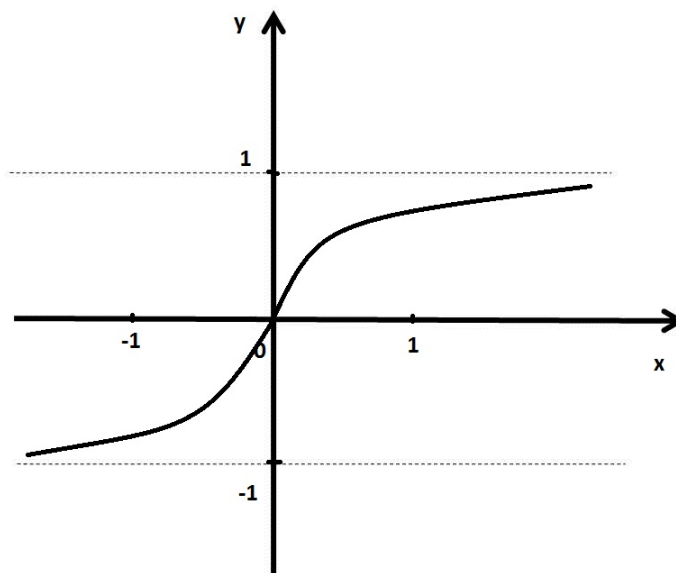
Slika 2.3. Graf sigmoidne funkcije

- **tangens-hiperbolna funkcija** (poseban oblik sigmoidne funkcije)

$$\text{output}_i = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (4.9)$$

gdje je $u = g \cdot \text{input}_i$

Interval vrijednosti je ovdje $[-1,1]$, što ima za mogućnost mapiranja vrijednosti u pozitivna kao i u negativna područja pa je upotrebljavana u mnogim eksperimentima.



Slika 2.4. Graf hiperboličko-tangentne funkcije

Najčešće korištene funkcije su sigmoidna i tangens-hiperbolna jer najbolje oponašaju stvarne linearne pojave.

NM razlikuju se i po tipu učenja koji se koristi [6]:

Svakoj pojedinoj vezi između dva neurona dodjeljena je težina veze, a postupak učenja mreže je proces računanja (podešavanja) težina između neurona u mreži. S vremenom one postaju takve da mreža daje točne odgovore, što znači da je znanje NM pohranjeno u težinskim vezama. Dva su glavna tipa učenja u mreži:

1. *nadgledano* (učenje s učiteljem, eng. supervised) skup podataka za treniranje sastoji se od prethodnih slučajeva s poznatim ulaznim i izlaznim vrijednostima, sustav sam podešava svoje parametre na temelju podataka (mreža "širenje unatrag", modularna mreža)
2. *nenadgledano* (učenje bez učitelja, eng. unsuperised), nisu poznate vrijednosti izlaznih podataka, za svaki uzorak je dan niz opisnih značajki, a sustav mora sam otkriti odnose između podataka, takav tip učenja koristi se za prepoznavanje uzoraka i klasteriranje (grupiranje eng. clustering)

Svaka NM prolazi kroz četiri operativne faze, a to su [17]:

1. **faza učenja (treniranja)** - mreža uči na uzorku za treniranje, na prošlim slučajevima, a težine se prilagođavaju u cilju minimiziranja ciljne funkcije, ova faza traje najdulje
2. **faza selekcije (unakrsne validacije)** - mreža nastoji optimizirati duljinu treniranja, broj skrivenih neurona i parametara - stopu učenja, najbolja dobivena mreža se pohranjuje i testira u sljedećoj fazi
3. **faza testiranja** - mreža se testira na uzorku za testiranje, koji do sad nije vidjela i taj se rezultat uzima kao ocjena mreže dok su težine fiksirane, mreža s najboljim test rezultatom se koristi u praksi
4. **operativana faza (faza opoziva)** - NM se primjenjuje na novim slučajevima s nepoznatim rezultatima, težine su fiksirane.

Za prilagođavanje težina veza među neuronima (w_n) upotrebljava se formula koju zovemo pravilo učenja, a četiri su najčešća korištena pravila [4], [17], [19]:

1. Delta pravilo

Pravilo najmanjih srednjih kvadrata minimizira funkciju cilja određivanjem vrijednosti težina, a cilj je minimizirati sumu kvadrata grešaka, a greška je definirana kao razlika između izračunatog i stvarnog (željenog) izlaza nekog neurona za dane ulazne podatke.

Jednadžba za Delta pravilo je:

$$\Delta w_{jn} = \eta \cdot y_{cj} \cdot \varepsilon_i \quad (4.10)$$

η je koeficijent učenja, Δw_{jn} vrijednost prilagođavanja težine veze od neurona j prema neuronu i , a računa se prema:

$$\Delta w_{jn} = w_{jn}^{nova} - w_{jn}^{stara} \quad (4.11)$$

y_{cj} je vrijednost izlaza izračunatog u neuronu j , dok je ε_i sirova greška koja se računa prema formuli :

$$\varepsilon_i = y_{di} - y_{ci} \quad (4.12)$$

a y_{di} predstavlja stvarni (željeni) izlaz koji se upotrebljava za izračunavanje greške.

Dakle, nova težina između neurona računa se tako da se stara težina korigira za vrijednost greške pomnožene s vrijednošću izlaza j i koeficijentom učenja.

Neki nedostaci Delta pravila su:

- problem lokalnog minimuma, koji se pojavljuje kada je najmanja greška funkcije pronađena samo za lokalno područje te je učenje zaustavljeno bez dostizanja globalnog minimuma, i problem pretreniranja jer nije moguće unaprijed odrediti koliko dugo treba učiti mrežu da bi ona mogla naučeno generalizirati na novim podacima.

2. Poopćeno Delta pravilo

Dodavanjem derivacije ulazne funkcije u Delta pravilo dobiveno je poopćeno Delta pravilo pa se prilagođavanje težina računa prema formuli:

$$\Delta w_{jn} = \eta \cdot y_{cj} \cdot \varepsilon_i \cdot f'(I_i) \quad (4.13)$$

a I_i predstavlja ulaz u neuron i , a dobro ga je koristiti kod nelinearnih funkcija.

3. Delta-Bar-Delta pravilo i prošireno Delta-Bar-Delta pravilo

Delta-Bar-Delta pravilo (DBD pravilo, Jacobs 1988.g.) razvijeno u cilju poboljšanja brzine konvergencije kod klasičnog Delta pravila. Nastoje se lokalizirati koeficijenti učenja na način da svaka mreža ima svoju stopu učenja η i mijenja ih kontinuirano kako učenje napreduje. Dinamičko podešavanje težina u DBD pravilu

izvodi se prema Saridisovom heurističkom pristupu.

Stopa učenja neke veze u mreži se povećava ako je predznak težine za tu vezu isti u nekoliko vremenskih koraka, a smanjuje se ako se predznak težina mijenja za određeni broj vremenskih koraka.

Tako jednačba Delta pravila (4.10) se modificira da je stopa učenja različita za svaku vezu:

$$\Delta w_{jn(k)} = \eta_k \cdot y_{cj} \cdot \varepsilon_i \quad (4.14)$$

Povećanje težina izvodi se linearno, a smanjuje geometrijski. Iako ima dosta prednosti u odnosu na Delta pravilo, DBD ima i nedostataka, a to su: ne uključuje momentum u jednačbu učenja, a zbog linearnog povećanja stope učenja mogu se pojaviti veliki skokovi, koji mogu preskočiti važna područja u površini greške, a to se ne može spriječiti sporim geometrijskim smanjivanjem stope.

Prošireno Delta-Bar-Delta pravilo (EDBD) razvijeno je u cilju ispravljanja nedostataka DBD pravila, predložio Minai i Williams (1998), a uvodi momentum α_k koji varira s vremenom i koristi se za sprječavanje saturacije (efekt zasićenja) težina u mreži, a dano je formlom:

$$\Delta w_{jn(k)}^t = \eta_k \cdot y_{cj} \cdot \varepsilon_i + \alpha_k \cdot \Delta w_{jn(k)}^{t-1} \quad (4.15)$$

gdje je α_k momentum veze k u mreži, a t je vremenska točka u kojoj se težine veze k prilagođavaju, a stope učenja i momentum podešavaju se eksponencijalno.

Gore navedena pravila učenja koriste željeni (stvarni) izlaz za izračunavanje greške, uče nadgledano. Ako željeni izlaz nije poznat upotrebljavamo nenadgledano učenje, kao što je Kohonen-ovo pravilo.

4. Kohonen-ovo pravilo

Kohonen-ova mreža ne uči na poznatim izlazima, a težine se prilagođavaju koristeći ulaz u neuron i :

$$\Delta w_{ji} = \eta(\text{extinput}_i - w_{ji}) \quad (4.16)$$

dok je $extinput_i$ ulaz koji neuron i prima iz vanjske okoline, pravilo se upotrebljava u Kohonen-ovoj samoorganizirajućoj mreži.

Prema sigurnosti ispaljivanja, NM se mogu podijeliti na:

- a) *Determinističke mreže* - neuron dostigne određenu razinu aktivacije, šalje impulse drugim neuronima ("ispaljivanje")
- b) *Stohastičke mreže* - ispaljivanje nije sigurno, odvija se prema probabilističkoj distribuciji

NM prema načinu prostiranja signala kroz mrežu [4],[11]:

- 1) **Statičke mreže** (eng. feedforward) - primaju ulaze u jednom prolazu, viši slojevi ne vraćaju informacije u niže slojeve
- 2) **Dinamičke mreže** (eng. feedback) - primaju ulaze u vremenskim intervalima, viši slojevi vraćaju informacije unazad u niže slojeve

NM prema vremenu učenja:

- **skupno (batch) učenje** - mreža uči samo u fazi učenja, dok su u ostalim fazama težine fiksirane
- **sekvencijalno (on-line) učenje** - mreža prilagođava svoje težine i u fazi opoziva

4.4. Prednosti i nedostaci neuronskih mreža

Neuronske mreže su vrlo popularan model obrade podataka jer posjeduju neke sposobnosti koje podsjećaju na ljudske. Najčešće su realizirane u obliku računalnog programa kojim se na računalu simulira njihov način rada. Pa bismo mogli reći da su umjetne neuronske mreže računalni programi koji mogu prepoznati uzorke u skupu podataka, te napraviti model za te podatke. Rezultati mnogih istraživanja pokazuju da neuronske mreže rješavaju gotovo sve probleme bolje nego tradicionalne metode i statističke metode. Razlog boljih rezultata NM u odnosu na rezultate statističkih metoda je u njihovoj mogućnosti da analiziraju nedostatke u podacima, podatke sa smetnjama, te sposobnost da uče na prošlim podacima. Također, imaju sposobnost otkrivanja složenih skrivenih veza između ulaznih i izlaznih podataka u iterativnim postupcima.

Neuronske mreže odlično rješavaju probleme predviđanja i klasifikacije (općenito sve probleme kod kojih postoji odnos između ulaznih i izlaznih varijabli, bez obzira na složenost veze), a danas se primjenjuju u mnogim segmentima života poput medicine, fizike, bankarstva itd., a najčešće su zadaci: predviđanje (kretanje dionica), obrada slike, obrada govora, raspoznavanje uzorka, problemi optimizacije i dr.

Neke osobitosti NM naspram konvencionalnih (simboličkih) načina obrade podataka su:

- uspješne u procjeni nelinearnih odnosa uzoraka
- mogu raditi s nejasnim ili manjkavim podacima tipičnim za podatke iz različitih senzora (kamera, mikrofona) i u njima raspoznavati uzorke
- robusni su na pogreške u podacima
- stvaraju vlastite odnose između podataka
- mogu raditi s velikim brojem varijabli
- formiraju znanje učeći iz iskustva (tj. primjera)

No, NM imaju i nedostatke:

- uspješno učenje zahtijeva velik broj podataka
- ne mogu davati odgovore izvan raspona vrijednosti (primjera) iz kojih uče
- generalizacija naučenih primjera uspješna je samo kod relativno "neprekidnih" pojava.

NM istraživane su s dva različita pristupa otkako su se pojavile u znanosti [17]:

- prvi, biološki pristup, istražuje NM kao pojednostavljene simulacije ljudskog mozga i upotrebljava ih za testiranje hipoteza u funkcioniranju ljudskog mozga
- drugi pristup, tretira NM kao tehnološke sustave za složenu obradu informacija.

4.5. Arhitektura i način rada neuronske mreže višeslojni perceptron

Statičke NM su najčešće korištene, jer su neuroni organizirani na tkz. unaprijed način, a to znači da mogu biti međusobno povezani bez formiranja povratnih veza, bez dinamičkih članova, što ih čini strukturno stabilnima.

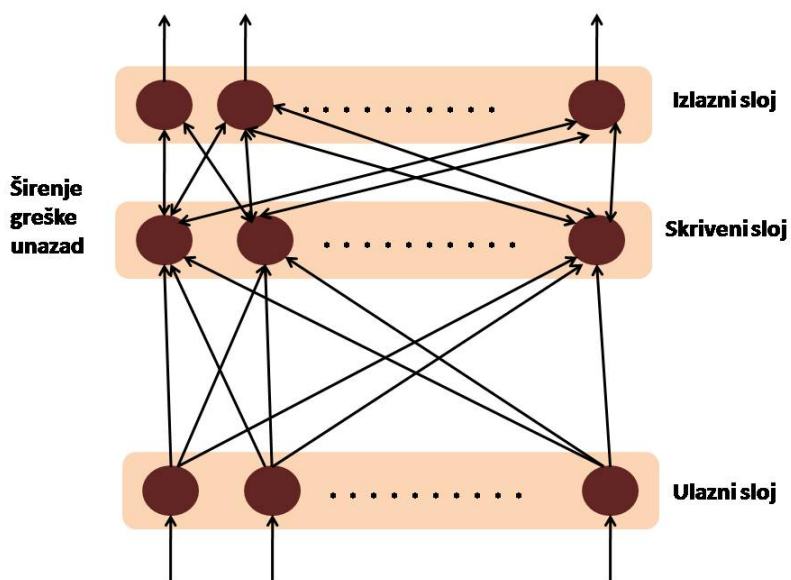
Nadalje opisujemo mrežu višeslojni perceptron korišten u radu. Višeslojni perceptron (MLP) jedna je od najraširenijih implementarnih topologija NM. Općenito, za klasifikaciju sa statičnim uzorkom, MLP sa dva skrivena sloja je univerzalni uzorak klasifikacije.

Riječ je o najpopularnijoj arhitekturi mreže, gdje se učenje odvija pod nadzorom pomoću algoritma "širenje unatrag", te je najviše zaslužan za "povratak" NM i učinio ih široko upotrebljavanom i popularnom metodom u različitim područjima. Kreator algoritma "širenje unatrag" bio je Paul Werbos 1974., a usavršavaju ga Rumelhart, Hinton i Williams 1986. (prva NM s jednim ili više skrivenih slojeva).

Višeslojni perceptron sastoji se od skupa neurona koji su međusobno povezani i čine strukturu mreže od jednog ulaznog sloja, jedan izlazni sloj i jedan ili više skrivenih slojeva, što je prikazano na *Slici 2.5*.

Tok podataka kroz mrežu može se opisati kroz nekoliko koraka [17], [19]:

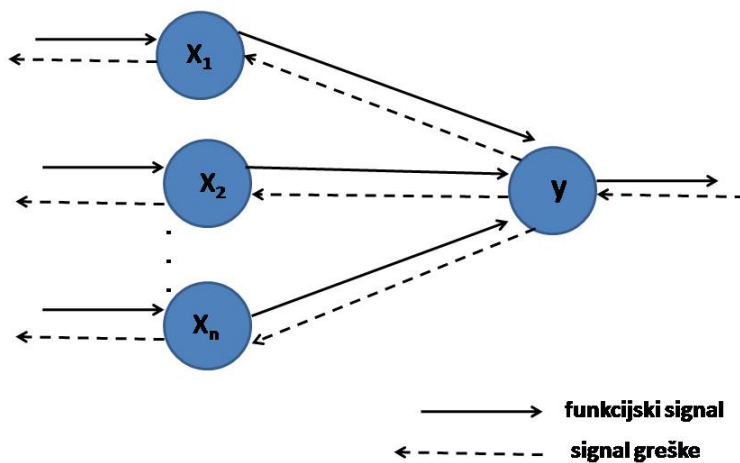
1. od ulaznog sloja prema skrivenom sloju: ulazni sloj učitava podatke iz ulaznog vektora X , i šalje ih u prvi skriveni sloj;
2. u skrivenom sloju: neuroni u skrivenom sloju primaju vagani ulaz i prenose ga u naredni skriveni ili u izlazni sloj koristeći prijenosnu funkciju;
3. kako informacije putuju kroz mrežu, za svaku jedinicu obrade računaju se sumirani ulazi i izlazi;
4. u izlaznom sloju: za svaki neuron računa se skalirana lokalna greška koja se upotrebljava u određivanju povećanja ili smanjenja težina;
5. "širenje unatrag" od izlaznog sloja do skrivenih slojeva: počevši od sloja neposredno ispod izlaznog sve do prvog skrivenog sloja, za svaki sloj unazad, računa se skalirana lokalna greška, te povećanje ili smanjenje težina i težine se podešavaju.



Slika 2.5. Arhitektura mreže višeslojni perceptron (MLP) [19], modificirano

U mreži su također prisutne i dvije vrste signala, što vidimo na *Slici 2.6*:

1. **ulazni signal** (funkcijski signal) - pobuda koja počinje na ulaznom dijelu mreže, širi se prema unaprijed neuron po neuron, a javlja se na izlazu mreže kao izlazni signal;
2. **signal greške** - javlja se kod izlaznog neurona mreže i širi se unatrag, sloj po sloj, kroz mrežu, a jednak je razlici između željenog i stvarnog izlaza



Slika 2.6. Vrste signala u MLP-u [10], modificirano

4.6. Računanje u mreži

Svaki neuron skrivenog sloja prima vagani ulaz iz ulaznog sloja kad ulazni sloj šalje podatke u prvi skriveni sloj, dok su početne težine postavljene slučajno, (često u interval od $[-0.1$ do $0.1]$), a računaju se prema formuli (4.1), odnosno ona sada glasi:

$$I_j^{[s]} = \sum w_{jn}^{[s]} \cdot x_i^{[s]} \quad (4.17)$$

gdje je $I_j^{[s]}$ ulaz u neuron j u sloju s , a $w_{jn}^{[s]}$ težina veze neurona j prema neuron i u sloju $(s-1)$.

Neuroni u skrivenom sloju svoje ulaze prenose prema formuli neke prijenosne funkcije, a formula glasi:

$$x_j^{[s]} = f\left(\sum w_{jn}^{[s]} \cdot x_i^{[s-1]}\right) = f(I_j^{[s]}) \quad (4.18)$$

Gdje je f neka prijenosna funkcija (npr. sigmoidna ili tangens-hiperbolna), a $w_{jn}^{[s]}$ izlaz neurona j u sloj s .

S obzirom da su formule tih funkcija dane prije u odjeljku (4.3) nećemo ih ovdje navoditi.

U slučaju da postoji više od jednog skrivenog sloja, prijenosna funkcija upotrebljava se kroz sve skrivene slojeve sve dok se ne dostigne izlazni sloj. A tada se izlaz mreže uspoređuje s željenim (stvarnim) izlazom i određuje se globalna greška E , prema formuli:

$$E = \frac{1}{2} \sum_k (d_k - x_k)^2 \quad (4.19)$$

x_k izlaz mreže, d_k željeni izlaz, a k indeks izlazne komponente, odnosno broj izlaznih neurona.

Izlazni neuron ima vlastitu grešku e oblika $(d_k - x_k)$, ali što se propagira kroz mrežu je skalirana greška u obliku gradijent komponente:

$$e_k^x = -\frac{\partial E}{\partial I_k^{(x)}} = -\frac{\partial E}{\partial x_k} \cdot \frac{\partial x_k}{\partial I_k^{(x)}} = (d_k - x_k) f(I_k)' \quad (4.20)$$

Za cilj procesa učenja u mreži MLP ima minimizaciju globalne greške E šireći je unazad kroz mrežu sve do ulaznog sloja (signal greške *Slika 2.4*).

Kroz proces mjenjanja težina, svaka veza u mreži se korigira kako bi se postigla manja globalna greška. Proces povećanja ili smanjenja težina (učenje) izvodi se korištenjem pravila gradijentnog opadanja:

$$\Delta w_j^{[s]} = -\eta \cdot \frac{\partial E}{\partial w_j^{[s]}} \quad (4.21)$$

η koeficijent učenja, a za računanje parcijalne derivacije jednadžbe (4.21) koristi se (4.20) što daje:

$$\frac{\partial E}{\partial w_j^{[s]}} = \frac{\partial E}{\partial I_j^{[s]}} \cdot \frac{\partial I_j^{[s]}}{\partial w_j^{[s]}} = -e_j^{[s]} \cdot x_i^{[s-1]} \quad (4.22)$$

što onda za podešavanje težina (učenje) daje, (kada rezultat uvrstimo u (4.21)):

$$\Delta w_j^{[s]} = \eta \cdot e_j^{[s]} \cdot x_i^{[s-1]} \quad (4.23)$$

,a to vodi do problema određivanja stope učenja η .

Koeficijent učenja η trebalo bi održavati na niskoj vrijednosti jer on određuje područje u kojem je površina greške lokalno linearna, a to je u sukobu s činjenicom da vrlo mali koeficijent učenja znači vrlo sporo učenje. U svrhu ubrzanja procesa učenja moguće je odabrati prevelik parametar η , što rezultira time da mreža postaje nestabilna (oscilira).

Rješenje problema sporog učenja temelji se na modificiranom delta pravilu (4.15) gdje uvodimo momentum α , koji ubrzava učenje kada je koeficijent učenja nizak. Razlike prethodnih težina u vremenu $(t-1)$ dodane su jednadžbi (4.11), tako da je trenutno podešavanje težina:

$$\Delta w_j^{t[s]} = \eta \cdot e^{[s]} \cdot x^{[s-1]} + \alpha \Delta w_j^{(t-1)[s]} \quad (4.24)$$

Također se učenje može ubrzati i na način da se težine ne podešavaju za svaki vektor treniranja, nego kumulativno, broj vektora treniranja nakon kojeg se težine podešavaju zove se **epoha**. Epoha koja nije velika može unaprijediti brzinu konvergiranja, dok velika može računanje greške učiniti puno složenijim i tako umanjiti prednosti.

Dok se greška ne smanji, težine i pragovi mreže ne stabiliziraju, učenje se odvija (iz epohe u epohu). Dva su glavna pristupa učenju (treniranju) :

- *on-line-treniranje* - korekcija težina izvodi se nakon svakog prezentiranog uzorka za učenje, jednostavnost izvedbe
- *batch treniranje* - prezentiraju se mreži svi parovi uzoraka iz skupa za treniranje, računa se pogreška za sve parove, podešavanje težina izvodi se jednom za sve uzorke.

Neki od najpoznatijih problema mreže "širenje unatrag" su problem sporog učenja koji se rješava unapređivanjem pravila učenja i drugih parametara, zatim problem

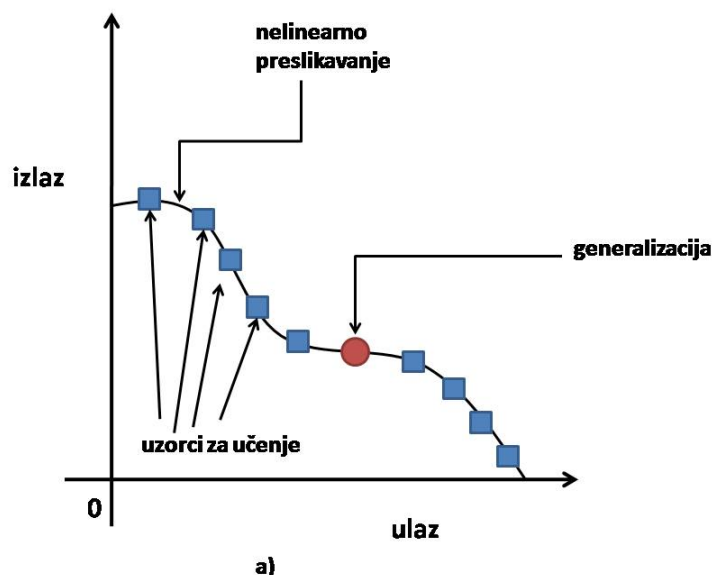
zasićenja neurona, problem lokalnog minimum i pretreniranje (overtraining), koje je također moguće unaprijediti.

Problem zasićenja neurona je prestanak učenja nekih neurona, ako njihove ulazne težine postanu velike. Ta situacija može se spriječiti dodavanjem F' offset parametra u derivaciju prijenosne funkcije. Također, vjerojatnost zasićenja je manja pri korištenju manjeg broja skrivenih neurona (koristi se minimalni neophodni broj skrivenih neurona.) [10], [17]

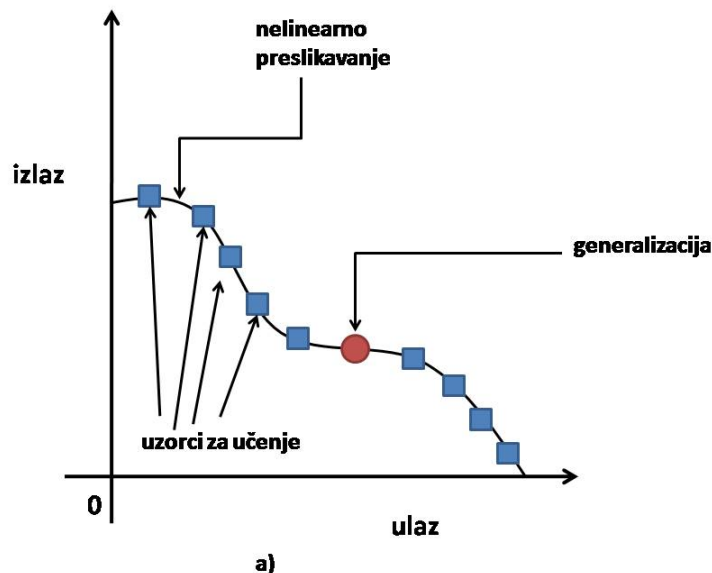
Problem lokalnog minimum nazivamo kada učenje upadne u lokalni minimum, a greška se minimizira samo lokalno, što se događa zbog načina na koji se greška propagira kroz mrežu ("gradient descent" optimizacija). Neka od rješenja navedenog problema su deterministička i koriste jednadžbe drugog reda za izračunavanje greške, a druga su stohastička oslanjajući se radije na slučajne brojeve, nego na jednadžbe.

Kažemo da mreža dobro generalizira kad daje dobre izlaze za nepoznate mreže (nisu sadržane u skupu za učenje mreže). Dobra generalizacija je interpolacija raspoloživih podataka s jednostavnom krivuljom, onom koja je najviše glatka.

NM ako dobro generalizira će dati točno ulazno-izlazno preslikavanje pa i u slučaju kad je ulaz različit od uzorka za učenje (treniranje), a kad mreža nauči previše ulazno - izlaznih primjera, neki uzorci za učenje budu zapamćeni. *Slika 2.7.* prikazuje dobru generalizaciju, a *Slika 2.8.* prikazuje lošu odnosno pretreniranu generalizaciju.



Slika 2.7. Generalizacija a) dobra [10], modificirano



Slika 2.8. *Generalizacija b) loša - pretrenirana* [10], modificirano

Pretreniranje (eng. overtraining) se dogodilo kad mreža gubi mogućnost generaliziranja između ulazno-izlanih primjera tj. neki uzorci za učenje su zapamćeni, što dovodi do problema pretreniranja. Pretreniranje je univerzalan problem za sve tipove algoritma NM, a uzrok jednog još neodgovorenog pitanja: kako dugo učiti na mreži?

Ne postoje točni uvjeti za prestanak učenja, ali postoje neki kriteriji. Pronalaženju kriterija za zaustavljanje pristupa se na više načina, a neki od njih su: unakrsnim ocjenjivanjem upotrebljavajući uzorak za potvrđivanje da bi se odredio trenutak u kojem prestaje učenje, dodavanjem pojačanja (bias-a) i slučajne greške u procjenu parametra, bootstrapping i dr. MLP ima široku primjenu i nemoguće je izbrojiti sva područja gdje se uspješno primjenjuje, a namjenjen je problemima s kontinuiranim vrijednostima ulaza i izlaza (više za predviđanje nego za klasifikaciju.) [10], [17]

Za problem predviđanja, MLP je primjenjiv na problem s nelinearnim zavisnostima koji koriste naredne vrste podataka: podaci sa smetnjama (impresivni rezultati na podacima u području obrade signala, za filtriranje i uklanjanje smetnji iz vremenskih serija signala), kaotične vremenske serije, financijske vremenske serije i problem multiple regresije, tj. može se primjeniti na problem, ali s oprezom na njene nedostatke.

Mreža MLP ne preporučuje se za upotrebu na nestacionarnim podacima ili za slučajeve kad podaci skrivaju u sebi više različitih problema. Za takve probleme rješenje se može pronaći u upotrebi nekoliko NM, a svaka bi rješavala zasebno pojedini problem, ili u izboru nekog drugog algoritma.

U radu je korišteno unakrsno ocjenjivanje upotrebljavajući uzorak za selekciju (treniranje se nastavlja sve dok greška u uzorku za selekciju napreduje, iterativni proces poznat kao "pohrani najbolju" alterativno trenira, testira mrežu i pohranjuje bolju sve dok rezultat ne napreduje u n-iteracija, izabrana najbolja mreža se testira na novom test uzorku u cilju određivanja sposobnosti generaliziranja mreže).

5. Opis podataka

Problem ovog istraživanja je praćenje ovisnosti varijabli koje utječu na cijenu kuća i njihovih promjena. Pomoću neuronskih mreža pokušat ćemo predvidjeti kretanje cijena kuća u predgrađu Bostona na tržištu nekretnina.

Istraživanje se temelji na podacima prikupljenim na StatLib baze podataka sa sveučilišta Carnegie Mellon University [7].

5.1. Ulazne i izlazne varijable

Podaci korišteni u ovom radu preuzeti su s web stranice Centra za strojno učenje i inteligentne sustave na University of California Irwine:

<http://archive.ics.uci.edu/ml/datasets/Housing>

Dani skup podataka sastoji se od 506 različitih slučajeva. U svakom slučaju na temelju trinaest ulaznih varijabli pokušavamo predvidjeti jednu izlaznu varijablu, cijenu kuće na tržištu nekretnina u predgrađu Bostona.

Varijable korištene u istraživanju su sljedeće:

1. Stopa kriminala po glavi stanovnika - kontinuirana varijabla sa rasponom vrijednosti 0,00632-9,96654
2. Razmjer stambenog prostora većih od 25000 kvadratnih stopa - kontinuirana varijabla sa rasponom vrijednosti 0-100
3. Razmjer prostora koji nije namijenjen poslovnim objektima po općinama - kontinuirana varijabla sa rasponom vrijednosti 0,46-27,74
4. Rijeka - pomoćna binarna varijabla koja pokazuje je li kuća u blizini rijeke - kategorijalna varijabla sa rasponom vrijednosti 0 ili 1
5. Postotak nitričnog-oksida u zraku - kontinuirana varijabla sa rasponom vrijednosti 0,385-0,871
6. Prosječan broj soba po domaćinstvu (stanu) - kontinuirana varijabla sa rasponom vrijednosti 3,561-8,78
7. Starost naselja - kontinuirana varijabla sa rasponom vrijednosti 2,9-100
8. Udaljenost od centra - kontinuirana varijabla sa rasponom vrijednosti 1,1296-12,1265
9. Indeks pristupačnosti - kontinuirana varijabla sa rasponom vrijednosti 1-24

10. Stopa poreza - kontinuirana varijabla sa rasponom vrijednosti 187-711
11. Odnos broja učenika-učitelja - kontinuirana varijabla sa rasponom vrijednosti 12,6-22
12. Udio obojenog stanovništva - kontinuirana varijabla sa rasponom vrijednosti 0,32-396,9
13. Postotak siromašnog stanovništva (udio nižes staleža) - kontinuirana varijabla sa rasponom vrijednosti 1,73-34,41
14. Srednja vrijednost nekretnina izražena u tisućama dolara - kontinuirana varijabla sa rasponom vrijednosti 6,3-50

Ulazne varijable su izražene u odnosima (razmjerima), postocima, indeksima dok je izlazna varijabla izražena kao srednja vrijednost cijene kuća u tisućama dolara.

Od 506 različitih slučajeva koliko ih se nalazi u bazi, u nekima od njih bilo je nedostajućih vrijednosti te su ti slučajevi izostavljeni iz uzorka, tako da je konačni uzorak 440 slučajeva.

Uzorak se sastoji od 440 slučajeva, a oni su podjeljeni na tri dijela. Prvi i najveći dio služi mreži za treniranje čak 70% (308 slučajeva), 10% ili 15% u unakrsnoj validaciji, te 20% ili 15% za testiranje.

Tijekom testiranja različitih arhitektura neuronskih mreža na ovim podacima, uzorak je bio podijeljen na dva različita načina koje prikazuje sljedeća tablica:

Podjela uzorka	Broj opažanja u uzorku za treniranje	Broj opažanja u uzorku za testiranje	Broj opažanja u uzorku za validaciju	Broj mreža testiranih na tom uzorku
70% , 10% , 20%	308	44	88	18
70% , 15% , 15%	308	66	66	7

Tablica 5.1. Podjela uzorka

5.2. Izrada neuronske mreže

Tijekom treniranja mreže računaju se i postavljaju inicijalne težine na međuovisne ulazne varijable ovisno o njihovom utjecaju na konačnu izlaznu varijablu. Unakrsna validacija poboljšava težine te pronalazi optimalne duljine učenja korištenja i/ili strukture mreže pri čemu mreža u iterativnom postupku uči na uzorku za treniranje koristeći različite parametre, kao npr. različiti algoritmi učenja, a svaka kombinacija se testira na uzorku za unakrsnu validaciju.

Cilj je pronaći mrežu s najmanjom duljinom učenja i strukturu mreže koja daje najbolji rezultat na uzorku za validaciju. Na kraju se tako dobivena mreža testira na uzorku za testiranje, a dobiveni rezultat se nakon faze testiranja uzima kao konačno mjerilo uspješnosti mreže.

Mali koeficijent učenja znači vrlo sporo učenje mreže, a momentum je koeficijent koji ubrzava učenje kada je koeficijent učenja nizak.

Epoha predstavlja broj slučajeva nakon kojih se vrši podešavanje težina u mreži tijekom učenja, ovisi o veličini uzorka, za veće uzorke moguće je odrediti veće epohe kako bi se povećala brzina rada mreže, a za manje uzorke bolje je da epoha bude što manja (epoha 10 znači da će mreža podešavati težine nakon što svakih 10 slučajeva prođe kroz mrežu).

Epoha koja je jako velika može unaprijediti brzinu konvergiranja, a isto tako može računanje greške učiniti mnogo složenijim i na taj način umanjiti prednosti.

U fazi testiranja računa se greška mreže na podacima na kojima nije vršeno podešavanje težina. Greška se utvrđuje tako da se pomoću ulaznih varijabli preko postavljene mreže računa izlaz te se uspoređuje sa izlazom koji se nalazi u podacima.

6. Rezultati

6.1. Rezultati u fazi učenja i testiranja mreže

Kod predviđanja cijena nekretnina neuronskim mrežama istrenirano je 25 različitih arhitektura neuronskih mreža. Arhitekture se razlikuju po broju skrivenih neurona, aktivacijskim funkcijama u skrivenom, odnosno izlaznom sloju, funkciji greške ili raspodjeli uzorka.

Praćeni su dobiveni rezultati te je na kraju dobivena mreža koja daje najbolji rezultat. Nakon testiranja 25 NM izabrana je mreža s najučinkovitijom arhitekturom.

Br.	Podjela uzorka	Struktura mreže	Funkcija greške	Aktivacijska funkcija u skrivenom sloju	Aktivacijska funkcija u izlaznom sloju	TRAIN rezultati	TEST rezultati
1	70% 10% 20%	MLP 14-12-1	SOS	logistic	identity	0,003988	0,006224
2	70% 10% 20%	MLP 14-12-1	SOS	logistic	identity	0,003804	0,005447
3	70% 10% 20%	MLP 14-10-1	SOS	logistic	identity	0,004312	0,005055
4	70% 10% 20%	MLP 14-11-1	SOS	logistic	identity	0,003994	0,006756
5	70% 10% 20%	MLP 14-10-1	SOS	logistic	identity	0,004405	0,005454
6	70% 10% 20%	MLP 14-4-1	SOS	Tanh	identity	0,004529	0,005224

Br.	Podjela uzorka	Struktura mreže	Funkcija greške	Aktivacijska funkcija u skrivenom sloju	Aktivacijska funkcija u izlaznom sloju	TRAIN rezultati	TEST rezultati
7	70% 10% 20%	MLP 14-6-1	SOS	Tanh	identity	0,004059	0,007819
8	70% 10% 20%	MLP 14-27-1	SOS	Tanh	identity	0,004315	0,006717
9	70% 10% 20%	MLP 14-17-1	SOS	Tanh	identity	0,003774	0,006526
10	70% 10% 20%	MLP 14-11-1	SOS	Tanh	identity	0,003855	0,005635
11	70% 10% 20%	MLP 14-28-1	SOS	logistic	identity	0,004584	0,004926
12	70% 10% 20%	MLP 14-14-1	SOS	Tanh	identity	0,004206	0,006525
13	70% 10% 20%	RBF 14-25-1	SOS	Gaussian	identity	0,010200	0,007482
14	70% 10% 20%	RBF 14-19-1	SOS	Gaussian	identity	0,009784	0,006506
15	70% 10% 20%	RBF 14-8-1	SOS	Gaussian	identity	0,011336	0,007986

Br.	Podjela uzorka	Struktura mreže	Funkcija greške	Aktivacijska funkcija u skrivenom sloju	Aktivacijska funkcija u izlaznom sloju	TRAIN rezultati	TEST rezultati
16	70% 10% 20%	MLP 14-20-1	SOS	Exponential	identity	0,004254	0,005168
17	70% 15% 15%	MLP 14-20-1	SOS	Logistic	identity	0,001224	0,002357
18	70% 15% 15%	MLP 14-25-1	SOS	Logistic	identity	0,001293	0,002429
19	70% 15% 15%	MLP 14-22-1	SOS	Logistic	identity	0,001301	0,002134
20	70% 15% 15%	MLP 14-17-1	SOS	Logistic	identity	0,001265	0,002019

Tablica 7.1. *Rezultati*

U sljedećoj tablici se nalazi se mreža koja je dala najbolji rezultat jer ima najmanju grešku (MSE).

Struktura mreže	MLP 14-17-1
Koeficijent korelacije na uzorku za treniranje	0,970059
Koeficijent korelacije na uzorku za testiranje	0,957910
Koeficijent korelacije na uzorku za validaciju	0,930620
Greška na uzorku za treniranje	0,001265
Greška na uzorku za testiranje	0,002000
Greška na uzorku za validaciju	0,002019
Algoritam treniranja	BFGS 55
Funkcija greške	SOS
Aktivacijska funkcija u skrivenom sloju	Logistic
Aktivacijska funkcija u izlaznom sloju	Identity

Tablica 7.2. Sažetak rezultata najuspješnijeg modela

Koeficijent korelacije je izračunat po formuli Pearson'ov koeficijent korelacije koji opisuje smjer i jačinu veze između varijabli, a u ovom slučaju opisuje jačinu veze između stvarnog i predviđenog izlaza.

S obzirom da su dobiveni koeficijenti korelacije preko 0,9 konkretno, na uzorku za validaciju koeficijent iznosi 0,9306 to znači da je veza između stvarnih i predviđenih izlaznih vrijednosti jaka.

Najuspješnija neuronska mreža je algoritam mreže "širenje unatrag" (MLP), mreža sa strukturom 14-17-1: 14 neurona u ulaznom sloju, 17 u skrivenom sloju i 1 u izlaznom sloju, jer ima najmanju grešku na uzorku za validaciju. Algoritam treniranja je BFGS 55 - mreža "širenja unatrag" (eng. backpropagation), a funkcija greške je SOS - suma kvadrata (eng. sum of squares). Aktivacijska funkcija u skrivenom sloju je logistička, a u izlaznom sloju ne koristi se aktivacijska funkcija.

6.2. Analiza uspješnosti neuronske mreže

Cilj treniranja neuronske mreže kod problema predviđanja je dobiti što manju grešku (najčešće MSE ili NMSE grešku).

Poželjno bi bilo kad bi uspjeli dobiti grešku ispod 0.01, ali i veće greške su prihvatljive ovisno o problemu, u našem primjeru greška iznosi 0,002019.

Radi usporedbe rezultata različitih modela, bolje je izračunati normaliziranu MSE grešku (NMSE), što znači da je prosječno odstupanje izračunatog od ciljanog (stvarnog) izlaza 0,2%, a izračuna se tako da se vrijednosti varijabli najprije svedu na interval [0,1] kako greška ne bi ovisila o rasponu vrijednosti izlaznih varijabli.

Najuspješnija mreža predviđa cijene nekretnina u predgrađu Bostona s prosječnim odstupanjem od 0,2%. Kretanje stvarne cijene (izlaza) i cijene koja je mreža predviđala možemo vidjeti na sljedećem grafu:



Slika 6.2. Rezultati predviđanja neuronske mreže

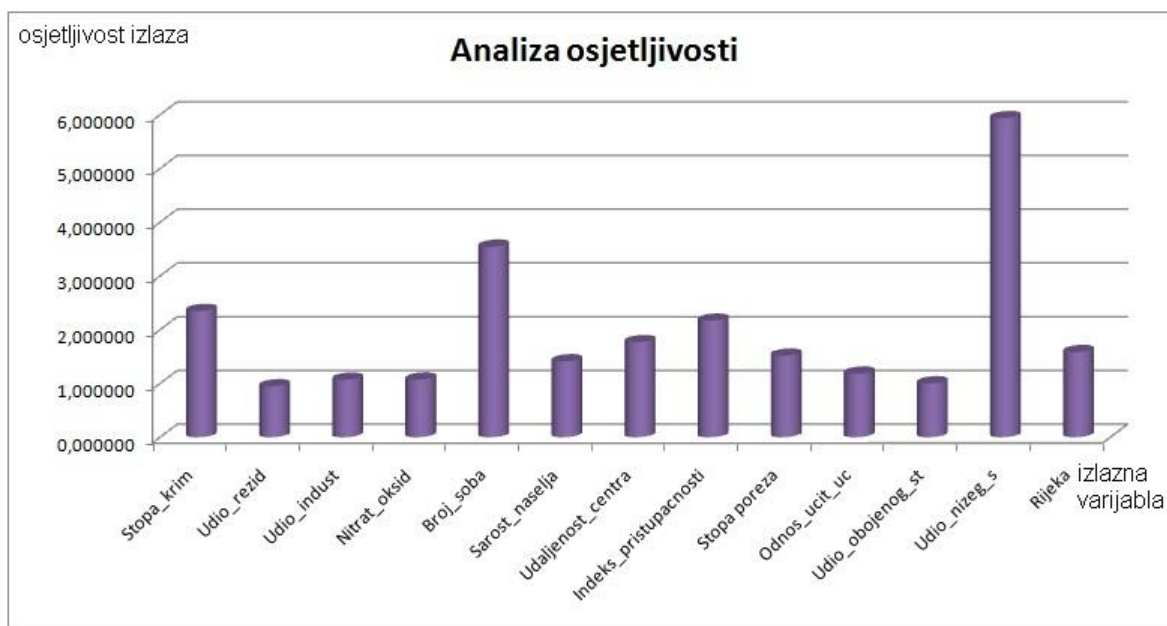
Na slici 6.2. prikazane su krivulje srednje vrijednosti. Iz grafikona se može vidjeti da krivulja predviđanja prati krivulju stvarne vrijednosti, ali da postoje odstupanja.

6.3. Analiza osjetljivosti

Analizom osjetljivosti moguće je vidjeti kakav utjecaj ulazne varijable imaju na izlaznu varijablu te u kojoj mjeri utječu na veličinu izlaza.

Analiza osjetljivosti izlazne varijable na pojedinu ulaznu varijablu, provodi se na način da se jedna ulazna varijabla promjeni za određenu vrijednost u intervalu $\pm 5\%$ (ili dr.) i promatra se promjena izlazne varijable, pod uvjetom da vrijednosti ostalih ulaznih varijabli ostanu nepromijenjene.

Na sljedećem grafu je prikazana osjetljivost izlazne varijable na svaku pojedinu ulaznu varijablu:



Slika 6.3. Rezultati analize osjetljivosti modela neuronske mreže

Vrijednost na osi y grafikona prikazuju kolika je promjena izlaznih varijabli ako se pojedina ulazna varijabla promijeni za jednu jedinicu.

Cijena nekretnina je najosjetljivija na varijablu postotka siromašnog stanovništva u susjedstvu, veliki utjecaj na cijenu imale su broj soba, stopa kriminala, pristupačnost centru, dok je najmanji utjecaj na cijenu imala varijabla razmjera stambenog prostora.

7. Zaključak

U ovom radu kreirana je neuronska mreža koja na osnovi ulaznih varijabli predviđa cijene nekretnina u predgrađu Bostona korištenjem neuronskih mreža. Na temelju trinaest ulaznih varijabli koje opisuju uvjete u kojima se nekretnina nalazi predviđa se njihova cijena. Korišten je algoritam neuronske mreže višeslojni perceptron, a rezultati najuspješnije neuronske mreže pokazuju da mreža ostvaruje veliku točnost u predviđanju cijene.

Neuronske mreže i umjetna inteligencija uopće mogu pridonijeti boljoj i kvalitetnijoj obradi podataka. Danas je još uvijek praksa da se za analizu podataka koriste regresivne statističke metode, ali kao što je pokazano u prethodnim istraživanjima u većini slučajeva mogu puno brže i kvalitetnije analizirati podatke i na kraju doći do točnijeg rezultata. S obzirom na dosadašnju uspješnost neuronskih mreža za očekivati je da će se u budućnosti sve više koristiti te da će statističke metode izgubiti prednost u problemu predviđanja.

Želimo li predvidjeti kamatnu stopu, cijenu nekretnina, cijenu energenata ili nešto drugo vrlo rijetko ćemo promatrati podatke u linearnom odnosu. Upravo ju tu najveća snaga neuronskih mreža, a ona leži u robusnim algoritmima koji omogućavaju obradu ne samo linearnih, već i nelinearnih podataka. U fazi učenja mreža računa vagoni (ponderani) odnos između ulaznih i izlaznih (ciljanih) varijabli, te na taj način uspostavlja nelinearan odnos između njih. Takav način promatranja podataka omogućava nam uočavanja nekih novih zakonitosti u promatranim podacima.

Promatrana neuronska mreža predviđa cijenu nekretnina u okolici Bostona s prosječnim odstupanjem od 0,2%. To znači da cijena koju predviđa neuronska mreža na osnovu ulaznih varijabli može maksimalno odstupati 0,2% od stvarne cijene. Brojni znanstvenici i dalje istražuju kako povećati pouzdanost i smanjiti grešku te razvijaju nove algoritme i arhitekture neuronskih mreža. Značajno poboljšanje mreža je doživjela upotrebom algoritma "širenje unatrag" koji je znatno povećao pouzdanost mreža.

Mogućnosti su primjene neuronskih mreža velike. Pokazana je i potvrđena primjenjivost na testiranom primjeru cijena nekretnina na području Bostona te na još nekoliko. Uskoro možemo očekivati veći broj mreža koje se uspješno primjenjuju na području ekonomije, ali i drugim područjima.

Nažalost još ne postoji standardiziran pristup kako neki problem riješiti pomoću neuronske mreže, ali to je zbog toga što svakodnevno nastaju novi algoritmi i arhitekture, a znanost je još u fazi intenzivnog razvoja. Svaki put kad želimo ispitati neki

problem moramo prvo napraviti i isprobati brojne neuronske mreže dok ne dobijemo najkvalitetniju s najvećom pouzdanošću. Kada znanstvenici standardiziraju pristup rješavanju problema na način da se točno definira koja arhitektura, koja pravila učenja, koliko skrivenih slojeva, koliko neurona u skrivenim slojevima te koje prijenosne funkcije koristiti u rješavanju pojedinog problema smatram da će tek tada neuronske mreže postati alat opće primjene u svakom području.

Daljnijim prikupljanjem podataka o kretanju cijena nekretnina, promatrajući veći broj ulaznih varijabli i uz daljnji razvoj još robusnijih algoritama bit ćemo u stanju postaviti model koji će još uspješnije opisati odnos pojedinih varijabli i njihov utjecaj na cijenu te ćemo u budućnosti biti u mogućnosti postaviti model koji će skoro sa zanemarivom greškom predviđati cijenu nekretnina.

U cilju poboljšanja točnosti promatranog modela, bilo bi potrebno revidirati ulazne varijable i povećati njihov broj, testirati druge inteligentne i statističke metode kako bi se usporedila njihova točnost na podacima, te ispitati mogućnost integrativnog pristupa kombiniranjem više metoda za dobivanje učinkovitog prediktivnog modela pa u tu svrhu možemo koristiti MLP algoritam, stabla odlučivanja, genetske algoritme, neizrastitu logiku ili neke druge inteligentne metode.

Literatura

- [1] B. Dalbelo Bašić, Umjetna inteligencija, Fakultet elektrotehnike i računarstva, Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave, http://www.fer.hr/_download/repository/UI_1_uvod.pdf 09.03.2009.
- [2] B. B. Dalbelo Bašić, M. Čupić, Jan Šnajder, Umjetne neuronske mreže, Fakultet elektrotehnike i računarstva, Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave, http://www.fer.hr/_download/repository/UI_14_umjetne_neuronske_mreze.pdf 09.03.2009.
- [3] L. Budin, B. Dalbelo Bašić, *Inteligentni sustavi*, Zagreb 2001.
- [4] J. P. Bigns, *Data Mining with Neural Networks*
- [5] Nikunj Chauhan, V. Ravi, D. Karthik Chandra, *Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks*, Expert System with Applications 36 (2009), 7659-7665
- [6] V. Čerić, M. Varga, *Informacijska tehnologija u poslovanju*, Element, Zagreb, 2004.
- [7] Harrison, D. and Rubinfeld, D.L. Hedonic prices and the demand for clean air, *J. Environ. Economics & Management*, vol.5, 81-102, 1978., Boston Housing Data - baza podataka korištena u praktičnom dijelu rada, <http://archive.ics.uci.edu/ml/datasets/Housing>
- [8] Jasna Horvat, *Statistika pomoću SPSS/PC+*, Osijek 1995.
- [9] V. Ilić, Neuronske mreže, <http://solair.eunet.rs/ilicv/neuro.html> 09.03.2009.
- [10] Kristijan Kraupner, *Uporaba višeslojnog perceptrona za raspoznavanje brojčano-slovčanih znakova na registarskim tablicama*, Fakultet elektrotehnike i računarstva, <http://www.zemris.fer.hr/projects/LicensePlates/english/Diplomski/kraupner-diplom> 09.03.2009.
- [11] S. Lončarić, Višeslojni perceptron, Fakultet elektrotehnike i računarstva, Zagreb <http://nm.zesoi.fer.hr/predavanja/pdf/neuro04.pdf> 9.03.2009.
- [12] D.W.Patterson, *Artificial Neural Networks*, Prentice Hall, 1995.

- [13] V. Ravi, C. Pramodh, *Threshold accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks*, Applied Soft Computing 8 (2008), 1539-1548
- [14] S. Russel, P. Norving, *Artificial Intelligence, a modern approach*, Prentice Hall series in Artificial Intelligence, 2003.
- [15] Z. Tonković, M. Zekić-Sušac, M. Somolanji, *Predicting natural gas consumption by neural networks*, Technical Gazette 16, 3(2009), 51-61
- [16] M. Vukomanović, M. Kararić, *Model for cost prediction of prefabricated housing*, Technical Gazette 16, 3(2009), 39-43
- [17] Mirjana Zekić-Sušac, *Neuronske mreže u predviđanju profitabilnosti ulaganja*, doktorska disertacija, Sveučilište u Zagrebu, Fakultet organizacije i informatike Varaždin 2000.
- [18] M. Zekić-Sušac, Anita Frajman-Jakšić, Nataša Drvenkar, *Neuronske mreže i stabla odlučivanja za predviđanje uspješnosti studiranja*, UDK 007:378.14(497.5)
- [19] Mirjana Zekić-Sušac, *Osnove umjetne inteligencije, predavanja*, Odjel za matematiku, Osijek, 2009.
- [20] Guoqiang Zhang, Michael Y. Hu, B. Eddy Patuwo, Daniel C. Indro, *Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis*, European Journal of Operational Research 116 (1999), 16-32
- [21] Š.Znam, L.Bukovský, M.Hejný, J.Hvorecký, B. Riečan, *Pogled u povijest matematike*, Tehnička knjiga, Zagreb

Sažetak

Sažetak. U ovom radu kreiran je model neuronske mreže koji na osnovu ulaznih varijabli predviđa cijene nekretnina u predgrađu Bostona korištenjem neuronskih mreža. Na temelju trinaest ulaznih varijabli, koje opisuju svojstva nekretnine i uvjete u kojima se nekretnina nalazi, predviđa se njihova cijena. Korišten je algoritam neuronske mreže višeslojni perceptron, a rezultati najuspješnije neuronske mreže pokazuju da je mreža prilikom učenja u mogućnosti aproksimirati funkciju između ulaznih i izlaznih varijabli te u predviđanju ostvariti veliku točnost. Analiza osjetljivosti ukazala je na značaj pojedinih ulaznih varijabli na izlazne varijable.

Ključne riječi: umjetna inteligencija, neuronske mreže, predviđanje, višeslojni perceptron, algoritam "širenje unatrag"

Summary

Abstract. In this work, a neural network model for predicting prices of real estate in Boston's suburb is created. On the basis of input variables, can predict the price of real estate in Boston's suburb. The price is being predicted on the basis of 13 input variables which describe the characteristics condition of a real estate.

The multilayer perceptron neural network algorithm, has been used and the results achieved by the best neural network indicates that it is able of approximate a non linear function between the output and input variables and achieves a high level of accuracy in prediction. The sensitivity analysis has shown that some input variables are more significant for the output than the others.

Key words: artificial intelligence, neural networks, prediction, multilayer perceptron, backpropagation algorithm

Životopis

Rođena sam 16. lipnja 1985. godine u Osijeku. Osnovnu školu "Vladimir Nazor" pohađala sam u Đakovu, nakon koje sam upisala opću gimnaziju također u Đakovu. 2004. godine upisala sam se na Odjel za matematiku u Osijeku.