UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

DIPLOMSKO DELO

UPORABA NEVRONSKIH MREŽ ZA NAPOVEDOVANJE TEČAJEV VREDNOSTNIH PAPIRJEV

Ljubljana, Junij 2003 BORUT LUKIČ

IZJAVA

Študent Borut Lukič izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom dr. Mojca Indihar-Štemberger in dovolim objavo diplomskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne ____________ Podpis ______________

KAZALO

1. Uvod

Svet, v katerem živimo, se neprenehoma spreminja, te spremembe pa imajo na nas lahko pozitivne ali negativne vplive. Da bi se čimbolj izognili tistim negativnim, so ljudje že v preteklosti poskušali napovedati spremembe, še preden so se pripetile. Mogoče je eden najbolj znanih »napovedovalcev prihodnosti« vsem poznani Nostradamus. V dandanašnjem življenju se pogosto srečujemo z različnimi poskusi napovedovanja prihodnjega in se na nekatere napovedi zelo zanašamo. Ena takih najpogostejših napovedi, s katerimi se srečujemo, je vremenska napoved.

Tudi za poslovni svet veljajo enaka pravila. Poslovno okolje se hitro spreminja, kar ima za gospodarske subjekte ugodne, pa tudi neugodne posledice. Za njihovo preživetje je zato ključnega pomena, da so na prihajajoče spremembe dobro pripravljeni in se na njih odzivajo. V zadnjih letih so se močno razbohotile tehnike, ki temeljijo na kompleksnih manipulacijah s podatki. Nove tehnologije in nova področja poslovanja se hitro razvijajo. Konkurenca se na vseh področjih hitro povečuje. Mednarodna trgovina se razvija prav v vseh panogah. Vse to je prispevalo k temu, da je današnje poslovno okolje bolj kompleksno in konkurenčno kot kadarkoli prej. Organizacije, ki se niso sposobne hitro prilagajati vsem tem spremembam, so obsojene na propad.

Razvoj računalniške tehnologije je močno olajšal delo današnjim organizacijam. Kompleksnost poslovnega okolja proizvaja velikanske količine podatkov, ki jih človek ni več sposoben pregledovati. Tukaj nastopijo računalniško podprti sistemi, ki so sposobni manipulirati z ogromnimi količinami podatkov in opozarjati na majhne detajle, ki utegnejo biti za določeno podjetje ključnega pomena.

Razvoj statističnih analiz podatkov je kmalu pripeljal do prvih tehnik za napovedovanje pojavov. Z računalniško podprtimi statističnimi sistemi je mogoče ob upoštevanju številnih dejavnikov dokaj natančno predvideti vrednost nekega neznanega pojava. Velikokrat pa se pri poskusu napovedovanja določenega pojava v prihodnosti srečujemo s problemom nepoznavanja prihodnjih vrednosti dejavnikov, ki na ta pojav vplivajo. V tem primeru s statističnimi metodami proučujemo dinamiko sprememb v pojavu in poskušamo na podlagi preteklih vrednosti napovedati vrednosti v prihodnje. Prav na področju napovedovanja časovno odvisnih pojavov pa se vedno bolj uveljavljajo tehnike napovedovanja z uporabo nevronskih mrež.

V tej nalogi bom v drugem poglavju predstavil modele, ki se najpogosteje uporabljajo za napovedovanje v ekonomskem svetu. V tretjem poglavju bo natančnejša razlaga nevronskih mrež in opis njihovega delovanja in uporabe pri napovedovanju. V četrtem poglavju bom obravnaval vrednostne papirje in njihove značilnosti. V petem poglavju pa bom predstavil še praktično uporabo nevronskih mrež pri napovedovanju cen vrednostnih papirjev. Za to bom uporabil nevronske mreže, ki jih bom gradil in učil z uporabo programske opreme Matlab. Programi, ki jih bom napisal v ta namen, bodo priloženi v prilogi. S temi mrežami bom poskušal napovedovati cene nekaj vrednostnih papirjev za obdobja, za katera sicer že poznamo dejanske vrednosti. Tako bom poskušal ugotoviti, ali so takšna orodja koristna tudi za uporabo v poslovnem svetu, bolj natančno za uporabo pri borznem poslovanju.

2. Napovedovanje

V slovarju slovenskega knjižnega jezika pod besedo napovedovanje najdemo: delati, da postane prihod koga, nastop česa vnaprej znan.

Ljudje so že od nekdaj poskušali napovedovati prihodnost. V daljni preteklosti so to funkcijo opravljali različni jasnovidci, danes pa si pomagamo z bolj znastvenimi metodami. Večina teh je bila razvitih že v 19. stoletju. Z razvojem računalniške tehnologije pa so v zadnjih desetletjih razvile tudi novejše in bolj kompleksne metode napovedovanja. Z vedno večjo uporabo osebnih računalnikov so se razvili kompleksni sistemi za napovedovanje, katerih uporaba in poznavanje je dandanes bolj pravilo kot izjema.

Razvoj novih tehnik za napovedovanje pa še zdaleč ni končan, ampak se neprestano razvija in prilagaja vedno močnejšim orodjem, ki jih pri tem uporabljajo. Največ pozornosti pri razvoju novih tehnik napovedovanja gre seveda napaki napovedi, ki je neizogibna in venomer prisotna, saj je nerealno pričakovati, da bi lahko napovedovali prihodnost z absolutno natančnostjo. Zato je glavni cilj, ki ga zasleduje razvoj, manjšanje te napake.

Zakaj bi se trudili z razvijanjem kompleksnih sistemov za napovedovanje, če pa že vnaprej vemo, da ne bodo nikdar znali natančno napovedati prihodnosti? Odgovor na to vprašanje je dokaj enostaven. Vse združbe se nahajajo v zelo negotovem poslovnem okolju, ki se hitro spreminja. Kljub temu so vsi primorani v sprejemanje odločitev glede prihodnosti. Odločitve, ki temeljijo na znanju in so podprte z prepričljivi argumenti, so dosti bolj zaželene, kot pa preprosto ugibanje. Seveda ne moremo trditi, da so odločitve sprejete zgolj na podlagi intuicije slabe. Ravno nasprotno so se že mnogokrat izkazale kot najboljše.

Modeli za napovedovanje, ki jih bom predstavil, so mišljeni zgolj kot podpora odločitvam podjetnikov, ki se v praksi bolj naslanjajo na izkušnje. V modelih pa dobijo ustrezno podporo oz. potrditev za svoje odločitve.

2.1. Tehnike napovedovanja

Katere vrste, oz. bolje rečeno tehnike napovedovanja so nam na voljo, ko smo prisiljeni v sprejemanje odločitev o naši prihodnosti? Prva delitev, ki bi jo lahko naredili, je delitev po času, za katerega napovedujemo. Tako imamo dolgoročne in kratkoročne napovedi. Slednje se nanašajo bolj na bližnjo prihodnost in se več uporabljajo pri opreativnih odločitvah, medtem ko se prve nanašajo na dolgo obdobje in so bolj uporabne pri strateških odločitvah. Napovedi bi lahko delili tudi glede na nivo podrobnosti, ki jih vključujejo. Makronapovedi so bolj splošne, mikronapovedi pa se nanašajo na zelo ozka področja (Hanke, Reitsch, 1995, str. 4).

Najbolj pogosta delitev napovedi pa je na kvalitativne in kvantitativne napovedi. Pri popolnoma kvalitativnih napovedih gre zgolj za presojo napovedovalca, ki je rezultat izkušenj, znanja in intuicije. Je nekakšna mentalna manipulacija s podatki, ki so v danem trenutku na voljo napovedovalcu. Pri popolnoma kvantitativnih napovedih je osebna presoja nepomembna. Gre izključno za matematično manipulacijo z danimi podatki. V praksi se je pokazalo, da dobimo najboljše napovedi, kadar združimo oba načina napovedovanja (Hanke, Reitsch, 1995, str. 5).

Kvantitativne tehnike nadalje delimo na statistične in deterministične. Prve se osredotočajo na vzorce, spremembe teh vzorcev in odstopanja zaradi neznanih vplivov. Pri statističnih tehnikah zasledimo dva različna pristopa. Prvi vrednosti podatkov razdeli na komponente, ki sestavljajo samo vrednost. To so trend, cikličnost, sezonska nihanja in slučajnostna komponenta. Drugi, ravno nasprotno, ne deli vrednosti podatka na različne komponente. V zvezi s tem drugim pristopom največkrat povezujemo ekonometrične modele za obravnavo časovnih vrst, na primer Box-Jenkins metode. Deterministične tehnike napovedovanja pa poskušajo iskati povezave med spremenljivko, ki jo napovedujemo, in ostalimi »bolj« znanimi spremenljivkami v okolju. Najbolj uporabne deterministične tehnike napovedovanja so: regresija, vodilne vrste in ekonometrični modeli (Hanke, Reitsch, 1995, str. 97).

Pri izbiranju ustrezne tehnike napovedovanja si pomagamo z odgovori na naslednja vprašanja. Kako natančno napoved potrebujemo? Za koliko časa vnaprej želimo napovedovati? Koliko se pri naši napovedi lahko zanašamo na osebno presojo?

Rezultati naše napovedi nam morajo dati tisto, kar potrebujemo za sprejemanje odločitev. Vselej se moramo zavedati, da bistvo napovedovanja ni v izvajanju kompleksnih matematičnih operacij na velikanskih količinah podatkov, temveč v pridobivanju ustreznih informacij, ki so nam v pomoč pri sprejemanju odločitev glede prihodnosti. Pri vsem pa moramo paziti tudi na to, da so koristi, ki nam jih prinesejo naše napovedi, večje kot stroški, nastali zaradi napovedovanja.

2.2. Proces napovedovanja

Pri večini tehnik za napovedovanje gre za pridobivanje izkušenj z uporabo podatkov iz preteklosti in na podlagi pridobljenega znanja predvidevanje prihodnosti. Torej se naslanjamo na dejstvo, da bodo dejavniki, ki so vplivali na preučevane vrednosti v preteklosti, v enaki meri vplivali tudi na podatke v prihodnosti. Izjema so tu le dejavniki, za katere eksplicitno vemo, kako bodo vplivali v prihodnje in to vključimo tudi v sam model napovedovanja.

Glede na dejstvo, da napovedovanje temelji na podatkih iz preteklosti, lahko razdelimo proces napovedovanja na štiri faze (Hanke, Reitsch, 1995, str. 6):

1. zbiranje podatkov,
2. zmanjševanje količine podatkov, oz. reduciranje podatkov,
3. izgradnja modela,
4. napovedovanje.

V prvi fazi gre za zbiranje relevantnih podatkov in preverjanje njihove točnosti. Večinoma je to eden najzahtevnejših korakov v celotnem procesu napovedovanja, saj nam velikokrat ni na voljo zaželena količina podatkov ali pa so ti premalo natančni ali zajeti premalo podrobno za naše potrebe.

V drugi fazi gre za izločanje nepomembnih podatkov. Podatkov imamo lahko enostavno preveč, nekateri podatki pa so lahko za naše potrebe popolnoma nepomembni ali neuporabni. Pogosto se srečujemo tudi s podatki, ki so za nas uporabni le v določenem časovnem intervalu.

V tretji fazi razvrstimo podatke v model za napovedovanje, ki je izbran glede na potrebe napovedi z glavnim ciljem minimizirati napako napovedi. Bolj kot je model enostaven, lažje je razumljiv in bolj verjetno je, da se bomo pri svojih odločitvah zanašali na napovedi, ki nam jih nudi. Zato je treba doseči ravnotežje med kompleksnim modelom, ki nam zagotavlja natačnejšo napoved, in preprostim modelom, ki nam nudi zgolj boljše razumevanje dobljenih rezultatov.

V četrti fazi dejansko uporabimo model, ki smo ga dobili z izvedbo prvih treh faz za napovedovanje vrednosti željenega podatka v prihodnosti. Velikokrat v tej fazi preverimo natančnost napovedi modela, tako da napovedujemo vrednosti za časovna obdobja, za katera že poznamo rezultate. Napake nato analiziramo in pogosto sledi popravljanje modela napovedovanja v želji minimizirati te napake.

Zelo pomembno je, da v proces napovedovanja vključimo svoje managerske sposobnosti in zdravo pamet. Model napovedovanja mora služiti zgolj kot orodje, ki se uporablja pri sprejemanju odločitev, in ne kot dejanski samostojni odločevalec. Na žalost se v praksi velikokrat preveč zanašamo na z modeli dobljene napovedi, ki jih v celoti niti ne razumemo.

Uporabnost in koristnost tehnik napovedovanja lahko povečamo s tem, da uporabimo bolj realen pristop k napovedovanju. Na samo napovedovanje ne smemo gledati kot na prerokbo, ampak kot na najboljši način zaznavanja in uporabe vzorcev in povezav z namenom napovedovanja. Pri takem pristopu se moramo ves čas zavedati, da so napake pri napovedovanju neizogibne, zato moramo razloge, ki povzročajo ta odstopanja napovedi od realnosti, podrobno raziskovati, da bi v prihodnje te napake čim bolj zmanjšali.

2.3. Časovne vrste

Podatke pogosto zajemamo preko daljšega časovna obdobja. Podatkom, ki so zajeti po natančno določenem časovnem intervalu, pravimo časovna vrsta. Da bi pojasnili gibanje časovne vrste, jo moramo natanko analizirati. Klasična dekompozicija je metoda, ki vrednosti v časovni vrsti razdeli na posamezne komponente. To so trend, ciklična komponentna, sezonska komponenta in slučajnostna komponenta (Hanke, Reitsch, 1995, str. 98).

Trend je dolgoročna komponenta, ki nakazuje rast ali padec časovne vrste skozi zelo dolgo obdobje. To rast ali padanje časovne vrste ugotavljamo na različne načine. Najpogosteje se uporablja metoda drsečih sredin in pa linearna aproksimacija trenda z regresijsko premico. Prisotnost trenda v časovni vrsti nam kažejo visoki koeficienti avtokorelacije časovne vrste, ki se z večanjem časovnega zamika počasi manjšajo.

Ciklična komponenta je valovanje vrednosti v časovni vrsti, ki je navadno posledica splošnih gospodarskih razmer. Ciklični valovi se ponavadi pojavljajo v intervalih dveh let in več. Glavni vzrok za ciklično valovanje je ciklično gibanje v gospodarstvu. Prisotnost ciklične komponente v časovni vrsti nam kažejo visoki koeficienti avtokorelacije pri časovnem zamiku, ki je v bližini časovnega intervala cikličnega nihanja.

Sezonska komponenta je gibanje vrednosti, ki se v podobnem vzorcu ponavlja vsako leto, ali pa celo večkrat letno. Prisotnost nam kažejo visoki koeficienti avtokorelacije ob določenih časovnih zamikih. Ta sezonska nihanja vrednosti v časovni vrsti so lahko posledica narave letnih časov ali pa koledarskih značilnosti (prazniki, dopusti).

Slučajnostna komponenta je gibanje vrednosti v časovni vrsti, ki ga dobimo potem, ko odstranimo trend, cikličnost in sezonsko nihanje. V tej komponenti so zajeti vsi nepojasnjeni in nepredvideni vplivi. Velik del te komponente zato predstavljajo čisto naključna nihanja. Za slučajnostno komponento je značilno to, da je izredno težko napovedljiva, saj se njena vrednost v času spreminja iz neznanih vzrokov in ji zato pravimo tudi naključna komponenta. Velikokrat nam slučajnostna komponenta prikriva sezonska in ciklična nihanja, zato jo imenujemo tudi šum.

2.3.1. Avtokorelacija

Podatki v časovni vrsti so mnogokrat v korelaciji sami s seboj, kadar jih zamaknemo za eno ali več obdobij. Temu pojavu pravimo avtokorelacija. Avtokorelacijski koeficient je torej korelacijski koeficient med dvema časovnima vrstama oz. eno časovno vrsto in to isto časovno vrsto, zamaknjeno za določen časovni interval.

Z računanjem avtokorelacijskih koeficientov z različnimi časovnimi zamiki lahko poiščemo vzorce v časovni vrsti kot so trend, ciklična, sezonska in slučajnostna nihanja. Izračunane kazalce z različnimi odmiki nato predstavimo v korelogramu[1], iz katerega lahko razberemo odgovore na naslednja vprašanja glede časovne vrste (Hanke, Reitsch, str. 104):

1. Ali so vrednosti naključne?
2. Ali je prisoten trend?
3. Ali so vrednosti stacionarne?
4. Ali je prisotna sezonska komponenta?

Kadar imamo opravka s popolnoma naključno časovno vrsto, bo korelacija med Y_t in Y_{t-1} blizu 0, kar pomeni, da vrednosti v časovni vrsti niso odvisne ena od druge.

Kadar je v časovni vrsti prisoten trend, bo korelacijski koeficient med Y_t in Y_{t-1} zelo visok. Avtokorelacijski koeficienti bodo tudi za odmike, večje od ene časovne enote, praviloma močno različni od nič za prvih nekaj odmikov, potem pa počasi padejo, ko se odmik povečuje.

Nasprotno je časovna vrsta stacionarna, kadar ji ne moremo določiti trenda. Dejansko stacionarnost pomeni, da se povprečna vrednost podatka v časovni vrsti na dolgi rok ne spreminja. Torej se vse vrednosti v časovni vrsti gibljejo okoli neke znane, stacionarne vrednosti.

Ko je v časovni vrsti prisoten vzorec sezonskega ali cikličnega nihanja, bodo avtokorelacijski koeficienti na določenih časovnih odmikih zelo visoki. Če se pojav pojavlja periodično vsako leto, bo torej visok korelacijski koeficient izračunan z odmikom časovne vrste za eno leto.

Kdaj pa lahko rečemo, da je koeficient avtokorelacije pri določenem časovnem zamiku dovolj različen od nič? Kako vemo, da je vrednost koeficienta pri določenem zamiku dejansko

[1] Korelogram je graf, ki prikazuje odvisnost avtokorelacijskega koeficienta od velikosti časovnega zamika

posledica enega izmed vzorcev? S tem vprašanjem se je ukvarjalo že mnogo strokovnjakov. Matematično je dokazano, da se porazdelitev vrednosti koeficientov avtokorelacije pri časovni vrsti, ki je popolnoma naključna, približno ujema z krivuljo normalne porazdelitve s standardnim odklonom $1/\sqrt{n}$ (Quenouille, 1949, str. 561).

S tem, ko primerjamo izračunane koeficiente avtokorelacije obravnavane časovne vrste z normalno porazdelitvijo, lahko ugotovimo, ali imamo opravka z naključnimi vrednostmi v časovni vrsti, ali pa je za vrednosti značilen kateri od vzorcev. Če z dovolj veliko verjetnostjo ugotovimo, da za koeficiente avtokorelacije obravnavane časovne vrste ni značilna porazdelitev v območju $0 \pm Z\frac{1}{\sqrt{N}}$, lahko trdimo, da vrednosti v časovni vrsti niso naključne, ampak so v njih pristoni vzorci, ki pa jih lahko napovedujemo, če jih podrobno analiziramo.

2.4. Izbira tehnike napovedovanja

Natančnost napovedovanja je v veliki meri odvisna od izbire ustrezne tehnike napovedovanja glede na naravo pojava, ki ga želimo napovedovati. Zelo pomembno je natančno vedeti kaj želimo napovedovati, za kakšno časovno obdobje, kdo bo uporabnik napovedi, zakaj sploh napovedujemo, koliko podatkov potrebujemo, kakšna je želena točnost napovedi in pa navsezadnje kakšni so predvideni stroški napovedovanja (Hanke, Reitsch, 1995, str. 114).

Podatke, ki so nam na voljo, moramo analizirati in v njih poiskati vzorce, če so ti prisotni. Ugotoviti moramo, ali imamo opravka s stacionarno časovno vrsto, ali pa je v časovni vrsti prisoten trend. Ugotoviti moramo tudi določene ciklične in sezonske vzorce.

2.4.1. Napovedovanje stacionarne časovne vrste

Stacionarno časovno vrsto smo označili kot tisto, katere povprečna vrednost se na dolgi rok ne spreminja. To je pogosto posledica velike stabilnosti dejavnikov, ki vplivajo na obravnavano spremenljivko. Najpreprostejši način za napovedovanje takšne spremenljivke je kar predvidevanje, da se bo tudi v prihodnje gibala v bližini svoje povprečne vrednosti oz. da bo kar zavzemala svojo povprečno vrednost. Pri bolj kompleksnem napovedovanju pa poskušamo upoštevati tudi dejavnike, ki so že v preteklosti vplivali na to, da je vrednost spremenljivke odstopala od svoje povprečne vrednosti (Hanke, Reitsch, 1995, str. 115).

Med tehnike napovedovanja stacionarnih časovnih vrst sodijo metoda izračunavanja povprečne vrednosti, metoda izračunavanja drsečih povprečnih vrednosti, metode eksponentnega glajenja in metode Box-Jenkins (Masters, 1995, str. 181).

Te tehnike uporabljamo, kadar imamo opravka z zelo stabilno časovno vrsto, ki je v že v osnovi dokaj preposta in so dejavniki, ki vplivajo na vrednost spremenljivke, dobro poznani in predvidljivi. Drugi razlog za uporabo takšnih tehnik je, če potrebujemo zelo enostaven model za napovedovanje. Razlog za to je lahko pomankanje podatkov iz preteklosti ali pa recimo potreba po enostavnih in razumljivih rezultatih. Te tehnike uporabljamo tudi v primerih, ko lahko obravnavano časovno vrsto s kakšno transformacijsko funkcijo (npr. logaritemsko) poenostavimo, ali pa že, ko lahko izločimo iz podatkov kak zelo dobro poznan dejavnik in je dobljen rezultat stacionarna časovna vrsta[1].

2.4.2. Napovedovanje časovne vrste s trendom

Za časovno vrsto s trendom je značilno, da je v vrednostih spremenljivke prisotna komponenta, ki skozi dolgo časovno obdobje bodisi raste ali pada. Bolj preprosto bi lahko rekli, da povprečna vrednost spremenljivke v časovni vrsti s trendom na dolgi rok raste ali pada. Trend je prisoten v večini podatkov, ki se uporabljajo v poslovnem svetu. Vzroki za prisotnost trenda so različni. Med najbolj pogostimi v poslovnem okolju najdemo inflacijo, rast bruto proizvoda in rast prebivalsta (Hanke, Reitsch, str. 116).

Za napovedovanje časovnih vrst s trendom uporabljamo: tehnike linearnih drsečih sredin, Brownov model linearno eksponentnega glajenja, Holtov model linearno eksponentnega glajenja, Brownov model kvadratnega eksponentnega glajenja, enostavno regresijo, Gompertzov model, krivulje rasti in različne druge eksponentne modele.

2.4.3. Napovedovanje časovne vrste s sezonsko komponento

V časovni vrsti, v kateri je prisotna sezonska komponenta, se podatki v vsakem letu gibljejo po podobnem vzorcu. Pri napovedovanju časovne vrste s sezonsko komponento beležimo gibanja v preteklih letih in jih nato z metodo prištevanja ali množenja dodajamo osnovnemu trendu časovne vrste. Vzroki, ki povzročajo sezonska nihanja v časovni vrsti, so že v sami naravi letnih časov in vremenskih značilnostih, ali pa v koledarskih posebnostih (prazniki, počitnice) (Hanke, Reitsch, str. 117).

[1] Primer: Z deflacioniranjem podatkov izločimo vpliv inflacije na vrednosti.

Pri napovedovanju časovnih vrst s sezonsko komponento si najpogostoje pomagamo s Census II modelom, Winterjevim exponentnim glajenjem, multiplo regresijo časovnih vrst ali Box-Jenkins metodami (Masters, 1995, str. 181).

2.4.4. Kriteriji izbora tehnike napovedovanja

Časovno obdobje, za katerega napovedujemo, je eden ključnih kriterijev pri izbiri ustrezne tehnike napovedovanja. Za kratki in srednji rok imamo na izbiro veliko različnih tehnik napovedovanja, ko se čas v prihodnost podaljšuje, pa se nam izbira manjša. Za napovedovanje za daljše časovno obdobje so bolj primerni kompleksni ekonometrični modeli, regresijski modeli in pa bolj kvalitativne tehnike napovedovanja.

Pomemben kriterij pri izbiri tehnike napovedovanja je tudi sam namen in uporaba rezultata napovedi. Tu se oziramo predvsem na enostavnost, hitrost in stroške. Pri preprostih odločitvah si zato najpogosteje pomagamo z enostavnimi tehnikami, kot so: eksponentno glajenje, projekcija trenda in regresijski modeli.

Najustreznejša tehnika napovedovanja je torej tista, ki bo pri najmanjši napaki napovedi nudila informacije za naše odločitve pravočasno, katere stroški ne bodo večji od njene koristi in katere rezultati bodo za njihovega uporabnika dovolj razumljivi, da jih bo znal pravilno interpretirati in uporabiti.

2.5. Merjenje napake napovedi

Uporabnost informacij, ki jih dobimo s tehnikami napovedovanja, je pogojena z njihovo natančnostjo. Obstaja več metod, s katerimi preverimo natančnost napovedi, vsem pa je skupno to, da moramo poznati dejanske vrednosti napovedane spremenljivke. Spremljanje spremenljivke v prihodnosti in ugotavljanje napake napovedi je izredno dolgotrajen proces, ki nam da rezultate o natančnosti prepozno, zato največkrat preverimo ustreznost našega sistema napovedovanja kar tako, da napovemo vrednosti za časovna obdobja, za katera že imamo dejanske vrednosti spremenljivke (Hanke, Reitsch, 1995, str. 119).

Glede na to, da imamo pri napovedovanju s kvantitativnimi tehnikami praviloma opravka s časovnimi vrstami, bomo v nadaljevanju za vrednost podatka v časovni vrsti ob določenem času uporabljali simbol Y_t. Razlikovati pa moramo tudi med dejansko vrednostjo in napovedano vrednostjo, zato bomo podatkom, ki so napovedani, dodali nadstrešek in jih zapisovali kot $\hat{Y}_t$.

Sedaj lahko že zapišemo napako napovedi kot $e_t = Y_t - \hat{Y}_t$. Napaka napovedi je torej definiriana kot razlika med dejansko in napovedano vrednostjo podatka.

Le redko pa je naš cilj napovedovati podatek v samo enem časovnem obdobju. Večinoma napovedujemo celotno časovno serijo v prihodnosti, zato moramo enako obravnavati tudi napake, za kar pa obstaja več načinov. Najbolj preprost je izračun povprečne absolutne napake. To dobimo tako, da seštejemo absolutne napake za vsak časovni indeks in dobljeno vsoto delimo z številom časovnih obdobij:

$$PAN = \frac{\sum_{t=1}^{n} \left| Y_t - \hat{Y}_t \right|}{n} \tag{1}$$

Pri napakah moramo posebno pozornost posvečati predvsem tistim večjim, ki imajo lahko tudi mnogo hujše posledice. Zato jih poskušamo poudariti. To naredimo z izračunom povprečne kvadratne napake. To dobimo tako, da seštejemo kvadratne vrednosti napake za vsak časovni indeks in dobljeno vsoto delimo z številom časovnih obdobij:

$$PKN = \frac{\sum_{t=1}^{n} \left(Y_t - \hat{Y}_t \right)^2}{n} \tag{2}$$

Ta metoda je boljša od metode povprečne absolutne napake v tem, da obravnava velike odstope kot izjemno nezaželene, saj se njihova vrednost kvadrira. Napoved, ki nam daje rezultate s srednjo natančnostjo, je velikokrat bolj ustrezna kot takšna, ki da v večini primerov zelo natančne rezultate, vsake toliko pa popolnoma napačne.

Včasih imamo opravka s podatki, katerih vrednost se časovno zelo spreminja, zatorej nam računanje razlike med napovedano in dejansko vrednostjo ne bo dalo realnih rezultatov. Tu imamo v mislih predvsem podatke, katerih vrednost se v času močno poveča ali zmanjša, zaradi česar bodo napake, zajete na delu, kjer so podatki majhni, skoraj neopazne, medtem ko bodo napake, zajete na območju, kjer so vrednosti velike, močno vplivale na rezultat. V takšnih primerih si pomagamo z izračunavanjem povprečne absolutne odstotne napake, ki jo dobimo tako, da seštejemo koeficiente relativne absolutne napake in jih delimo z številom časovnih obdobij:

$$PAON = \frac{\sum_{t=1}^{n} \frac{\left| Y_t - \hat{Y}_t \right|}{Y_t}}{n} \tag{3}$$

Treba pa je tudi ugotoviti, ali so naše napovedi morebiti pristranske. Torej ali so praviloma previsoke oziroma prenizke. Pri tem si pomagamo z povprečno odstotno napako (enačba 4). Kadar je ta blizu ničle, takrat so naše napovedi nepristranske. Če je močno negativna, potem so naše napovedi previsoke oziroma precenjene, v nasprotnem primeru, ko je povprečna odstotna napaka močno pozitivna pa vemo, da so naše napovedi prenizke oziorma podcenjene. Povprečno odstotno napako dobimo tako, da seštejemo koeficiente relativne napake po posameznih časovnih obdobjih in nato dobljeno vsoto delimo z številom časovnih obdobij:

$$PON = \frac{\sum_{t=1}^{n} \frac{\left(Y_t - \hat{Y}_t\right)}{Y_t}}{n} \tag{4}$$

Pri ocenjevanju tehnike napovedovanja si ponavadi pomagamo z več navedenimi kazalci, ki jih primerjamo s kazalci drugih tehnik napovedovanja. Tako lahko ugotovimo, katera izmed tehnik napovedovanja, ki so nam na voljo, je najbolj ustrezna. Žal nam ti kazalci ne izražajo nekih absolutnih kriterijev, po katerih bi lahko ocenjevali uspešnost tehnike napovedovanja. Kadar imamo na voljo le en model za napovedovanje, katerega uporabnost skušamo oceniti, se moramo zato zanašati na kvalitativne metode ocenjevanja učinkovitosti. Pri tem si seveda lahko pomagamo tudi z zgoraj omenjenimi kazalci.

2.6. Kritika tehnik napovedovanja

Najpogosteje se kot kritika omenjenih tehnik napovedovanja pojavlja teorija učinkovitih trgov, ki jo je že leta 1965 razvil E. Fama in se je močno prijela v ekonomskih krogih. Teorija učinkovitih trgov že v svoji šibki obliki predpostavlja, da so v ceni nekega blaga že vključene vse informacije, ki so na voljo v zvezi z ceno tega blaga. V ceno blaga se tudi nemudoma vključijo vse novice, povezane s tem blagom. Ker pa je novice že po definiciji nemogoče napovedovati, je tudi cena blaga nenapovedljiva. Narejenih je bilo veliko študij, ki so poskušale potrditi ali ovreči teorijo učinkovitih trgov. Ker je trgov več, se je tudi teorija učinkovitih trgov različno prijela na posameznih trgih. Trenutno vsaj za trg kapitala velja, da ni učinkovit (Giles, Lawrence, Tsoi, 2001, str. 166).

Če torej trg kapitala ni učinkovit, potem vrednost kapitala ni odraz vseh razpoložljivih informacij. To pa pomeni, da smo v primeru, ko vemo nekaj več kot drugi, v prednosti. Če torej vemo, da je v vrednosti določene kapitalske dobrine napovedljiva komponenta in jo znamo vsaj približno napovedati, imamo konkurenčno prednost. To prednost lahko izkoriščamo toliko časa kot ga drugi potrebujejo, da pridejo do istih informacij.

3. Nevronske mreže

Nevronske mreže so nastale kot plod dolgoletnih raziskav na področju umetne inteligence. Cilj raziskovalcev je bil dati računalniku sposobnost učenja iz izkušenj in na podlagi pridobljenega znanja sposobnost sprejemati oziroma predlagati odločitve. V letih 1960 – 1980 so razvijali kompleksne sisteme umetne inteligence imenovane Expert Systems, ki so temeljili na modelu miselnih procesov visoke stopnje, bolj natančno na predpostavki, da je miselni proces manipulacija z določenimi podatki. Kmalu se je izkazalo, da takšni sistemi niso prav uspešno posnemali »naravne« inteligence. Glavni razlog za to je bil predvsem v tem, da je bilo v osnovi delovanje teh sistemov močno različno od delovanja možganov. Ugotovili so, da lahko umetno inteligenco, ki se bo obnašala kot naravna, naredijo le tako, da prenesejo način delovanja možganov v računalnik (Statsoft, 2002, Neural Networks).

3.1. Biološka osnova

Možgani so izjemno kompleksen, nelinearen, paralelni sistem, ki ga v osnovi sestavlja izredno veliko število nevronov (okrog 10 miljard), ki so med seboj povezani v nekakšno omrežje (posamezni nevron ima lahko tudi po 1000 in več povezav). Vsak nevron ima razvejan vhod (dendrite), telo in pa razvejan izhod (aksone). Aksoni enega nevrona so preko sinapse povezani z dendriti drugega nevrona. Ko se nevron aktivira, sproži elektrokemični signal skozi svoje aksone. Ta signal potem prečka sinapso in preko dendritov vstopi v drug nevron, ki se na podlagi jakosti prejetega signala (če je le-ta dovolj močan) lahko aktivira. Jakost signala je odvisna od izgube moči pri prečkanju sinapse. Ta izguba je odvisna od prepustnosti sinapse. Prepustnost se lahko spreminja, s čimer se spreminja tudi delovanje celotne mreže nevronov. Tudi same povezave med nevroni se lahko spreminjajo s tem, ko nastajajo nove povezave ali pa nekatere obstoječe izginjajo. Tako se spreminja delovanje celotnega sistema, s čimer označujemo pojem učenja. Možgani si z ustvarjanjem novih povezav med nevroni ustvarjajo nekakšne asociativne vzorce, ki so jih v bodoče sposobni prepoznavati in se na njih različno odzivati.

Takšen skupek navidez izredno enostavnih delcev, ki pa nastopajo v zelo velikem številu in v kompleksnem omrežju, je zmožen izvajati zelo kompleksne operacije. Treba je omeniti, da so pri delovanju človeških možganov prisotni še različni drugi dejavniki, ki pa za predmetno nalogo niso relevantni in se zato z njimi ne bomo ukvarjali.

3.2. Osnove nevronske mreže

»Umetni možgani« ali kot jim strokovno pravimo »nevronske mreže« so ravno tako kot pravi možgani zgrajeni iz številnih nevronov. Lastnosti teh nevronov so zelo podobne lastnostim pravih organskih možganskih celic, tako so le-ti sposobni:

- sprejemati poljubno število vhodnih signalov,
- vhodne signale ustrezno obravnavati in po potrebi spremeniti,
- oddati signale enemu ali večim nevronom.

Dejansko je nevron v računalniškem svetu definiran kot matematična funkcija s poljubnim številom parametrov, ki so primerno uravnoteženi s posameznimi utežmi ter konstantnim odmikom, ki se doda izhodni vrednosti (Dermuth,Beale, 1997, str. 44):

$$a = f\left(\sum_{i=1}^{n} w_i p_i + b\right) \tag{5}$$

kjer je:

- i število vhodov
- p_i signal i-tega vhoda
- w_i utež i-tega vhoda
- b konstantni odmik nevrona
- f transformacijska funkcija
- a izhodni signal

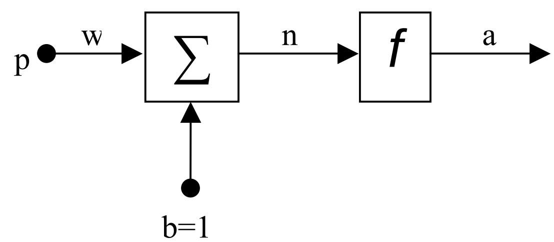
Bistvo takšnega nevrona je v tem, da lahko uteži in vrednost odmika poljubno spreminjamo, s čimer spreminjamo delovanje nevrona. Vrednosti popravljamo toliko časa, da nevron zavzame želen način delovanja, torej da nam pri določenih vhodnih signalih da želeni izhodni signal. Postopku prilagajanja vrednosti uteži in odmikov s primerjanjem izhodnih signalov z želenimi glede na vhodne, pravimo učenje nevrona. Tako naučeni nevron se je potem zmožen odzivati na različne vhodne signale na natanko določen način. Ko imamo opravka z več nevroni, povezanimi v mrežo, pa s prilagajenjem uteži v posameznih nevronih spreminjamo delovanje takšne mreže. Temu postopku potem pravimo učenje mreže.

Poznamo različne tehnike učenja mreže, ki določajo predvsem, katere nevrone bomo spreminjali in za koliko jim bomo popravljali uteži in odmike. Vsem pa je skupno, da učenje mreže poteka na podlagi učne množice. Učna množica je množica podatkov, za katere imamo tudi želene, že znane izhodne vrednosti. Mrežo učimo toliko časa, dokler nam za določeno učno množico ne generira rezultatov, ki so dovolj blizu tistim pravim. Delovanje naučene mreže nato preverjamo še z validacijsko množico (za katero imamo tudi že rezultate) in tako preverimo, kako se mreža odziva na nove vhode, ki jih do sedaj še ni spoznala.

3.2.1. Preprost nevron z enim vhodom

Za lažjo predstavo o tem, kako deluje »umetni« nevron, si oglejmo preprost nevron z enim samim vhodom (slika 1). Delovanje takšnega nevrona se prične z vhodnim signalom (p). Ta vhodni signal se pomnoži z utežjo (w). Vrednost te uteži se lahko spreminja (npr. z učenjem), s čimer se spreminjajo lastnosti nevrona. Dobljeni produkt se spremeni še za konstantni odmik (b). Dobljeni rezultat (n) predstavlja vhodni parameter v transformacijsko funkcijo (f), ki ga spremeni v izhodni signal (a).

Slika 1: Preprost nevron z enim vhodom in konstantnim odmikom b=1



Vir: Dermuth,Beale, 1997, str 44.

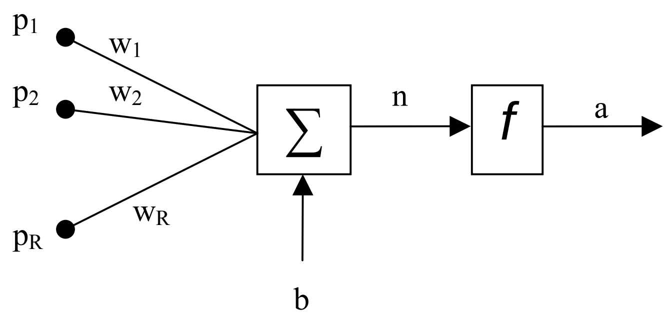
Uporabnost takšnega nevrona je dejansko zelo omejena, saj je enaka preprosti matematični funkciji, ki je kar transformacijska funkcija nevrona, katere vhod se spremeni preko navadne linearne funkcije.

3.2.2. Nevron z več vhodi

V praksi se pogosteje uporabljajo nevroni z več vhodi (slika 2). Vsak vhodni signal (p_i) ima svojo lastno utež (w_i), kar lahko zapišemo kot vektor uteži z dimenzijo, ki je enaka številu vseh vhodov. Skupni seštevek produktov med (R) vhodnimi signali in njihovimi utežmi predstavlja dejanski vhod v nevron, z dodanim konstantnim odmikom pa vhodni parameter (n) za transformacijsko funkcijo (f), katere rezultat je izhodni signal (a). Matematično lahko to zapišemo kot funkcijo:

$$f(\sum p_i w_i + b) = a \qquad (6)$$

Slika 2: Nevron z R vhodi



Vir: Dermuth, Beale, 1997, str. 47.

Vrednosti posameznih uteži in konstantnega odmika lahko poljubno spreminjamo, da dosežemo želeno obnašanje nevrona. Nevron učimo tako, da mu predstavimo vektorje vhodnih signalov. Dobljen izhodni signal primerjamo z želenim in ugotavljamo razlike. Uteži in konstantni odmik nato popravimo v skladu z tehniko učenja, ki jo izvajamo.

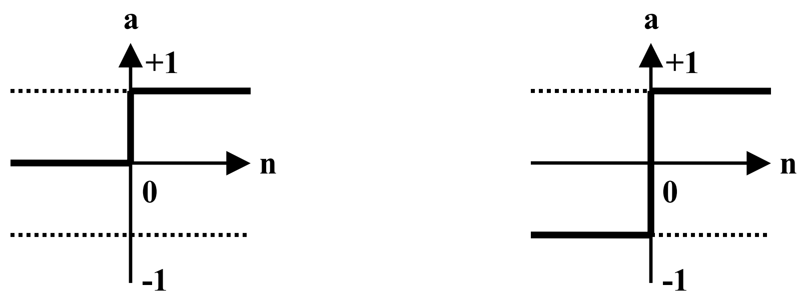
3.2.3. Transformacijska funkcija

Najpomembnejši element nevrona je njegova transformacijska funkcija, ki nekako simulira aktivacijo naravnega nevrona. Prav transformacijska funkcija zato določa obnašanje in kompleksnost posameznega nevrona in uravnava izhodne vrednosti. Glede na potrebne lastnosti nevrona uporabljamo različne transformacijske funkcije.

Stopničasta funkcija
Je ena najpreprostejših in zelo pogosto uporabljenih transformacijskih funkcij. Izhodni signal takšne funkcije je 0, kadar je vhodni signal (n) manjši od 0, ali 1, kadar je vhodni signal večji ali enak 0. Takšno transformacijsko funkcijo uporabljamo pri sistemih za odločanje, kadar želimo imeti izhod omejen le na dve vrednosti, ki največkrat pomenita enostavni da in ne.

Simetrično stopničasta funkcija
Je zelo podobna stopničasti funkciji. Izhodni signal omeji na 1, kadar je vrednost vhodnega parametra večja ali enaka 0, oziroma -1, kadar je vhodni parameter manjši od 0. Uporablja se v podobnih primerih kot navadna stopničasta funkcija, le da je njen vhodni razpon primeren tudi za procesiranje negativnih vrednosti.

Slika 3: Transformacijski funkciji, levo stopničasta, desno simetrično stopničasta



Vir: Dermuth, Beale, 1997, str. 416.

Linerana funkcija

Vhodni signal (n) linearno preslika v izhodni signal (a). To je ena najpogosteje uporabljenih transformacijskih funkcij v izhodni plasti mreže, kadar želimo imeti izhode, ki lahko zavzemajo poljubne vrednosti. Torej jo uporabljamo, kadar želimo za izhod neomejene vrednosti.

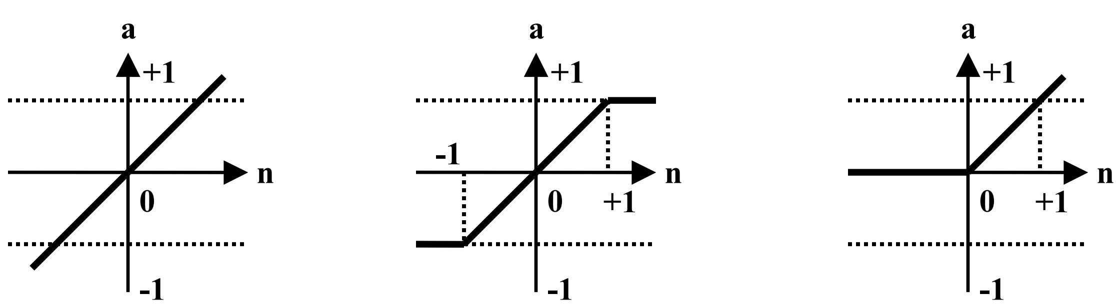
Omejeno linearna funkcija

Linearno preslika vhodni signal (n) v izhodni signal (a), če se ta giblje med -1 in 1. Vrednosti manjše od -1 omeji na to vrednost. Enako velja tudi za vrednosti večje od 1. Bistvo te funkcije je, da omeji izhod na interval vrednosti med -1 in 1. Znotraj tega intervala se izhodne vrednosti preslikajo linearno. Uporabljamo jo, kadar želimo omejiti sicer linearen izhod na določen interval.

Pozitivno (negativno) linearna funkcija

Je linearna za pozitivne (negativne) vrednosti, drugače pa omeji izhod na 0. Podobno kot omejena linerana funkcija ta omeji izhodni signal na določen interval, vrednosti ki so izven intervala pa zavzamejo vrednost, ki predstavlja eno od mej intervala.

Slika 4: Transformacijske funkcije, od leve proti desni: linearna, omejeno linearna, pozitivno linearna



Vir: Dermuth, Beale, 1997, str. 417-418.

Radialana funkcija
Preslika vhodne vrednosti v izhodne po formuli normalne porazdelitve:

$$a(n) = e^{-n^2} \tag{7}$$

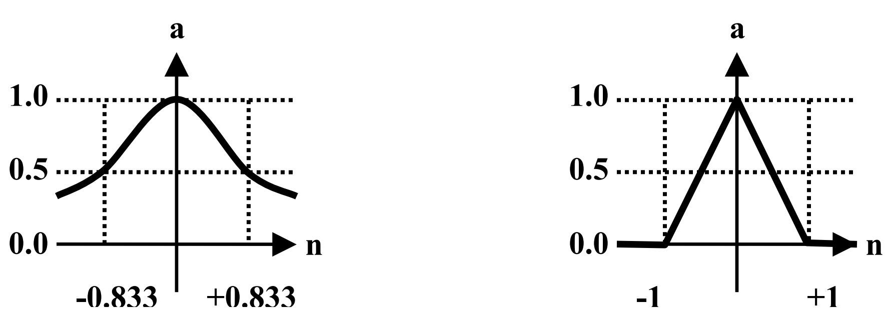
Vhod (n) je pri tej funkciji namesto produkta med utežjo in vhodom praviloma razdalja med utežjo in vhodnim signalom. Izhod iz takšnega nevrona nam torej pokaže, v kolikšni meri je vhodni signal poznan nevronu oziroma koliko se vhodni signal ujema z signalom, s katerim smo nevron učili. Takšni nevroni se pogosto uporabljajo za glajenje kompleksnih funkcij in pa prepoznavanje vzorcev.

Trikotna funkcija
Linearno porazdeli vhodne vrednosti med -1 in 0, ter obratno sorazmerno vhodne vrednosti med 0 in 1. Trikotna funkcija je definirina po naslednji enačbi:

$$a(n) = \begin{cases} 1 - n; 0 < n < 1 \\ 1 + n; \text{-} 1 < n < 0 \end{cases} \tag{8}$$

Takšno transformacijsko funkcijo uporabljamo v podobne namene kot radialno, le da se pri trikotni ne oziramo na normalno porazdelitev, temveč razliko med vhodnimi signali in določenimi utežmi prikažemo popolnoma linearno.

Slika 5: Transformacijski funkciji: levo radialna, desno trikotna



Vir: Dermuth, Beale, 1997, str. 419.

Sigmoidna funkcija

Zelo pogosto uporabljena funkcija, ki preslika vhodne vrednosti med -∞ in +∞ v izhodne, s tem da jih zgladi na vrednosti med 0 in 1. Funkcija je zvezna in tudi zvezno odvedljiva, in sicer definirana z naslednjo enačbo:

$$a(n) = \frac{1}{1+e^{-n}} \tag{9}$$

Prednost te funkcije je, da nam zna preslikati poljubno vhodno vrednost na interval med 0 in 1 tako, da je vsaka od vhodnih vrednosti v izhodnem signalu predstavljena s svojo lastno vrednostjo. Drugače rečeno, vsaka izhodna vrednost je slika natanko ene vhodne vrednosti. Prav ta lastnost jo naredi za idealno funkcijo pri aproksimaciji želenega izhodnega signala. Druga pomembna lastnost te funkcije je, da je mnogo bolj natančna pri vhodnih vrednostih, ki so okrog sredine intervala, vrednosti, ki so na robovih vhodnega signala, pa obravnava bolj površno. Zato je že sama po sebi zmožna omejiti pretiran vpliv skrajnih vhodnih vrednost na izhod.

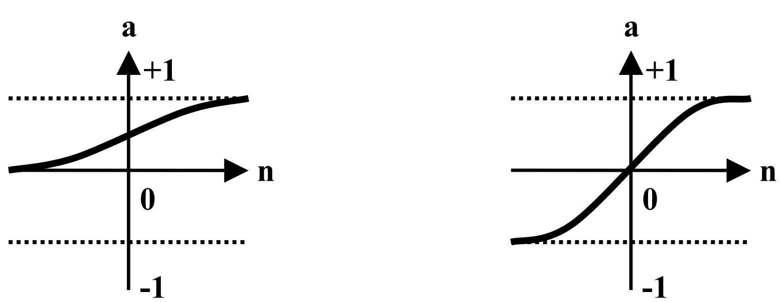
Hiperbolično tangenta funkcija

Podobno kot sigmoidna zgladi vse vhodne vrednosti, vendar v interval med 1 in -1. Definirina je z naslednjo enačbo:

$$a(n) = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}. \tag{10}$$

Prednosti in uporaba te funkcije so identični tistim, ki jih ima sigmoidna transformacijska funkcija, le da nam ta najbolj pogoste vhodne signale pretvori v izhodne okrog vrednosti 0.

Slika 6: Transformacijski funkciji: levo sigmoidna, desno hiperbolično tangentna



Vir: Dermuth, Beale, 1997, str. 416

3.2.4. Enoplastna nevronska mreža

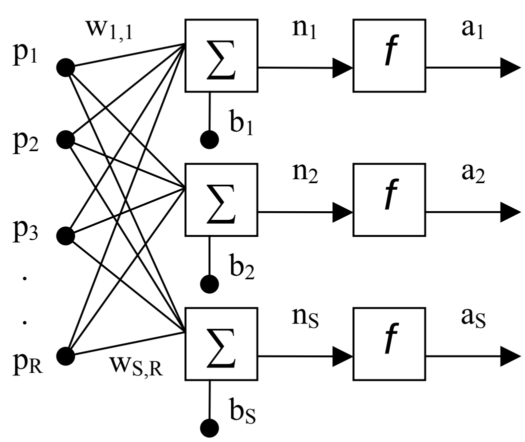
Enoplastno nevronsko mrežo dobimo, če vhodne signale povežemo na večje število nevronov. Na sliki 8 je enoplastna nevronska mreža z R vhodi in S nevroni. Takšni mreži strokovno pravimo perceptron. Vsak nevron ima za vsak vhodni signal določene lastne uteži. Torej se v danem primeru srečamo z matriko uteži W (slika 7) dimenzij $S \times R$, v kateri so vrednostno zapisane povezave vhodov z nevroni. Vsak nevron v mreži ima lahko drugačno transformacijsko funkcijo.

Slika 7: Matrika vhododov v mrežo; R vhodov za S nevronov

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \Lambda & w_{1,R} \\ w_{2,1} & w_{2,2} & \Lambda & w_{2,R} \\ M & M & \Lambda & M \\ w_{S,1} & w_{S,2} & \Lambda & w_{S,R} \end{bmatrix}$$

Vir: Dermuth, Beale, 1997 str. 50.

Slika 8: Enoplastni perceptron z S nevroni in R vhodi.



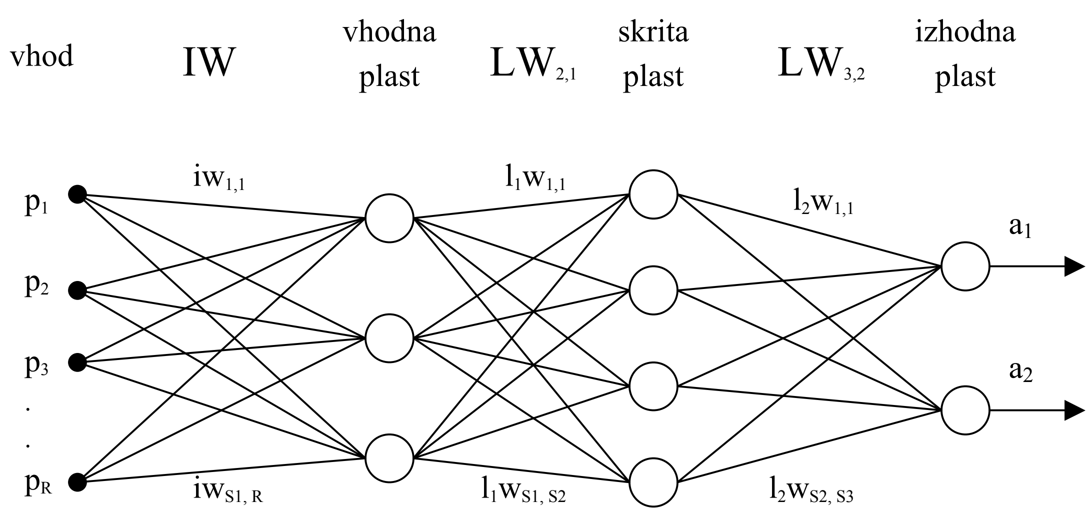
Vir: Dermuth, Beale, 1997 str. 50.

Takšna mreža nam vhodni vektor dolžine R preslika v izhodni vektor dolžine S. Preslikavo določajo transformacijske funkcije posameznih nevronov, njihovi konstantni odmiki, ter matrika vhodnih signalov W (na sliki 8 prikazana kot R vhodnih signalov s po S povezavami). Posamezne uteži in konstantne odmike najpogosteje nastavimo tokom učenja mreže. Ker nevroni v takšni mreži niso medsebojno povezani, je postopek učenja takšne mreže identičen učenju samostojnega nevrona z več vhodi.

3.2.5. Večplastna nevronska mreža

Večplastno nevronsko mrežo oziroma večplastni perceptron dobimo, ko sestavimo skupaj več enoplastnih nevronskih mrež tako, da so izhodi ene plasti vhodi v drugo plast. Plast, ki sprejema vhodne signal od zunaj, se imenuje vhodna plast. Zadnja plast, ki daje izhodne signale, pa izhodna plast. Vse ostale, vmesne plasti, imenujemo skrite plasti. Vsaka plast ima lahko poljubno število nevronov. Vsaka plast ima tudi svojo lastno matriko uteži vhodov (W). Pri večplastni mreži posebej ločimo matriko uteži vhodne plasti (IW). Vhodni signali v takšno mrežo vstopijo prek vhodne plasti, kjer se prek ustreznih uteži prenesejo na nevrone. Nevroni vhodne plasti te signale transformirajo s svojo transformacijsko funkcijo in konstantim odmikom, ter jih pošljejo v naslednjo plast. Med plastema se jakosti signalov zopet ustrezno popravijo s pomočjo uteži, preden pridejo do nevronov v tej plasti. Tako signal potuje od vhodne do izhodne plasti. Nevroni izhodne plasti nam dajo končni rezultat. Zaradi smeri signala, ki potuje izključno v smeri od vhodne do izhodne plasti, pravimo takšnim mrežam tudi enosmerne nevronske mreže.

Slika 9: Večplastna nevronska mreža: troplastni perceptron



Vir: Dermuth, Beale, 1997 str. 50.

Na sliki 9 je prikazan troplastni perceptron, ki vhodni vektor dolžine R preslika na tri nevrone v vhodne plasti preko matrike vhodnih uteži (IW). Izhodi iz nevronov vhodne plasti so preko uteži (v matriki $LW_{2,1}$) povezani na štiri nevrone v skriti plasti. Od tam se signal zopet prek uteži prenese na dva nevrona v izhodni plasti. Končni rezultat mreže sta dva izhoda (a_1, a_2). Želeno delovanje takšne mreže dobimo z nastavljanjem uteži v vhodni in vseh ostalih matrikah vhodov med plastmi ter z nastavljanjem konstantnih odmikov posameznih nevronov v mreži. Za nastavljanje teh se najpogostoje uporablja postopek, ki mu pravimo vzvratno učenje. Pri vzvratnem učenju primerjamo dobljeni izhod ob danih vhodih z želenim. Potem najprej popravljamo vrednosti nevronov v izhodni plasti, s tem, da uteži izhodov nevronov prejšne mreže popravljamo tako, da povečujemo uteži signalom, ki so bližje želenemu izhodnemu, ter zmanjšujemo tistim, ki se od želenega rezultata mončno razlikujejo.

3.3. Vrste nevronskih mrež

Glede na število nevronskih plasti, števila nevronov v posamezni plasti, karakteristike povezav in transformacijskih funkcij, ločimo različne tipe nevronskih mrež. Najbolj znane oblike so: enostavni perceptron, večplastni perceptron, radialno zasnovane mreže, samo-organizirajoče se mreže in povratne mreže.

3.3.1. Perceptron

Perceptron je najenostavnejši tip nevronske mreže. Sestavlja ga ena plast nevronov, ki imajo praviloma stopničasto transformacijsko funkcijo (slika 8, na str. 19). Izhodi iz te mreže so torej omejeni na vrednost 0 ali 1. Takšne nevronske mreže se uporabljajo le za zelo enostavne naloge razvrščanja vzorcev. Zaradi njihove enostavnosti se mnogokrat uporabljajo za študijske namene, saj je njihovo delovanje lahko razumljivo, zaradi česar je tudi interpretacija rezultatov enostavna.

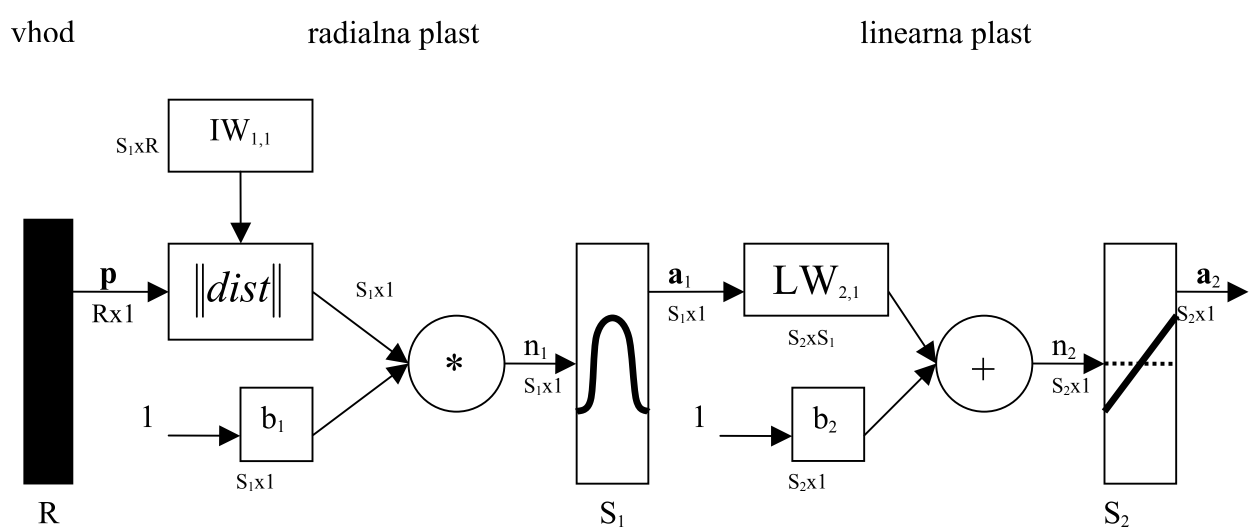
3.3.2. Večplastni perceptron

Večplastni perceptron oziroma MLP (ang. multi layer perceptron) je ena najpogosteje uporabljanih nevronskih mrež. Kot pove že samo ime, je ta mreža sestavljena iz več plasti (slika 9). Pri MLP mrežah uporabljamo tudi nevrone z bolj kompleksnimi transformacijskimi funkcijami. Večplastni perceptron je tipični predstavnik enosmernih mrež, pri katerih se signal širi enosmerno od vhoda proti izhodu. Navadno imajo takšne mreže eno skrito plast, število nevronov v tej plasti pa je približno pol manjše od števila vhodov v vhodno plast. Z večanjem števila plasti in nevronov v posamezni plasti lahko postanejo takšne mreže zelo kompleksne in sposobne reševati tudi bolj komplicirane probleme. Uporabljamo jih za napovedovanje časovnih vrst in aproksimacijo funkcij, tako linearnih kot nelinearnih.

3.3.3. Radialno zasnovane mreže

Radialno zasnovane mreže se od ostalih razlikujejo že po samih nevronih. Pri teh mrežah so v uporabi izključno nevroni z radialno transformacijsko funkcijo, katere vhod pa je absolutna razlika med utežjo in vhodnim parametrom, pomnožena s konstantnim odmikom nevrona (slika 10). Narava takšnega nevrona je, da je izhod močnejši, bolj kot je vhodni signal blizu vrednosti uteži. Takšne mreže imajo dve ključni plasti: skrito (radialno) plast, ki jo sestavljajo nevroni z radialno transformacijsko funkcijo, ter izhodno (linearno) plast, katere nevroni imajo linearne transformacijske funkcije. Število nevronov v skriti plasti je odvisno od želene stopnje natančnosti, s katero naj mreža prepoznava vhodne signale. V izhodni plasti pa se večinoma nahaja en sam nevron.

Slika 10: Radialno zasnovana nevronska mreža.



Vir: Dermuth, Beale, 1997, str. 198.

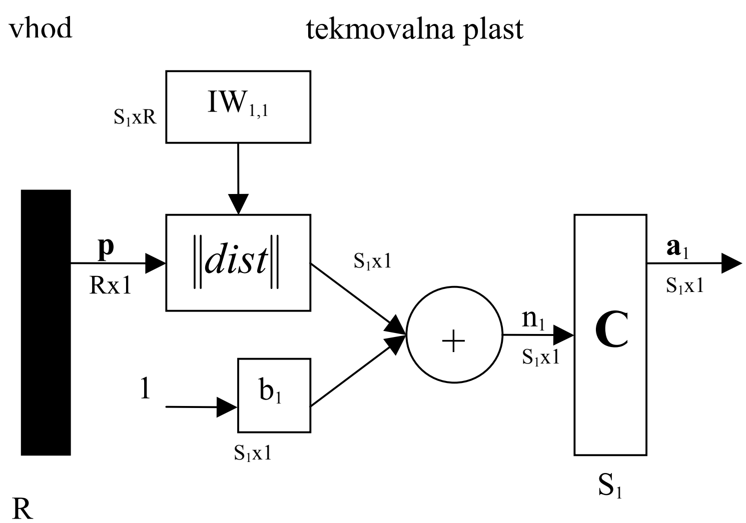
Da bi razumeli, kako deluje takšna mreža, moramo slediti vhodu p skozi mrežo do izhoda a_2. Ko mreži damo vhodni signal, bo vsak radialni nevron nanj reagiral glede na to, kako blizu je ta vhodni signal njegovi uteži. Torej bodo signali, ki se močno razlikujejo od vhodnih uteži, imeli vrednosti blizu nič in bodo bolj opazni v plasti z linearnimi nevroni. Nasprotno bodo signali, ki so po vrednosti zelo blizu uteži nevronov, zelo opazni po linearni transformaciji. Iz danega lahko sklepamo, da je takšna mreža izredno učinkovita pri prepoznavanju določenih vzorcev. Tudi dejansko se takšne mreže pogosto uporabljajo v sistemih za prepoznavanje vzorcev (govora, slike). Takšne mreže so tudi zelo uspešne pri prepoznavanju vzorcev v časovnih vrstah, zaradi česar jih uporabljamo tudi za napovedovanje časovnih vrst.

3.3.4. Samo-organizirajoče se mreže

Samo-organizirajoče se mreže so zagotovo eno od najbolj zanimih področij nevronskih mrež. To so mreže, ki so se same sposobne naučiti ugotavljati zakonitosti in korelacije vhodnih signalov. Če se ostale mreže naučijo prepoznavati vzorce v vhodnih signalih, se te mreže naučijo to tako, da na podobne vhode reagirjo nevroni, ki so v medsebojni bližini. Nevroni so v tej mreži porazdeljeni po tekmovalni plasti, in sicer tako, da so skoncentrirani na območjih, na katerih se pojavlja večina vhodnih signalov.

Nevron v takšni mreži primerja vhodni signal (p) z utežjo (w) in izračuna razliko. Ta je vedno negativna absolutna vrednost, od katere se odšteje še konstanti odmik. Večja kot je razlika med vhodnim signalom in utežjo, bolj negativna je vrednost vhodnega signala (n) v kompetitivno transformacijsko funkcijo plasti. Kompetitivna transformacijska funkcije izbere najmanj negativno vrednost in jo spremeni v 1. Izhodna vrednost ostalih nevronov se spremeni v 0. Torej izhodni vektor (a) sestavljajo ničle, razen na mestu zmagovalnega nevrona, kjer se pojavlja vrednost 1. Prednost tega je v tem, da natančno vemo, kateri nevron se je najbolje odzval na vhodni signal. Drugače povedano: vemo, kako je mreža kategorizirala vhodni signal. Zato te mreže uporabljamo predvsem za razvrščanje vhodnih signalov po določenih kriterijih (npr. pri podatkovnih bazah).

Slika 11: Samo-organizirajoča se mreža



Vir: Dermuth, Beale, 1997 str. 217.

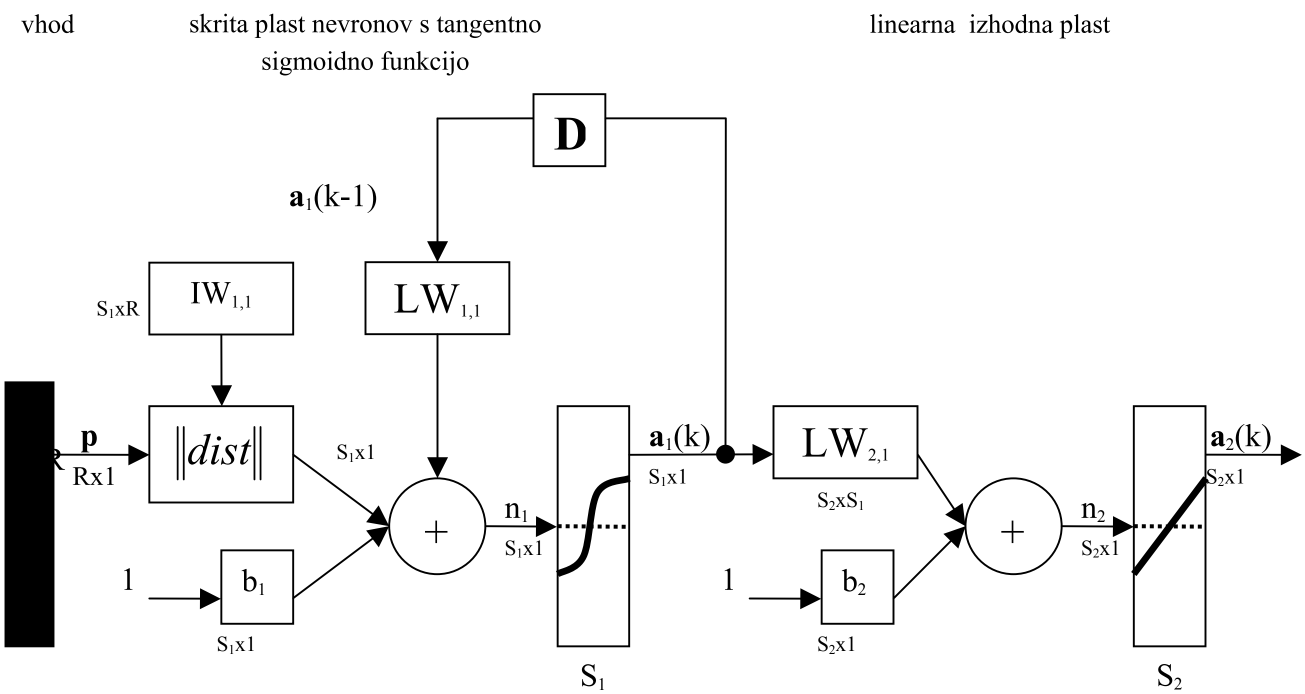
3.3.5. Povratne mreže

Povratne mreže (ang. recurrent networks) predstavljajo eno izmed najpomembnejših vej v nevronskih mrežah. Od navadnih večplastnih perceptronov se razlikujejo po tem, da imajo vsaj eno povratno povezavo. Izvor povratne povezave je lahko v katerikoli skriti plasti ali pa celo v izhodni plasti. Prav ta povratna povezava oziroma zanka izboljšuje sposobnost delovanja in učenja mreže. V povratni povezavi je prisotna tudi zakasnitev, kar povzroča še bolj nelinearno obnašanje mreže. Najbolj znana tipa mrež s povratno zanko sta Elmanova in Hopfieldova mreža.

Elmanova nevronska mreža

Elmanova mreža je sestavljena iz dveh plasti. Izhodna plast je z vhodno povezana s povratno povezavo, ki ima zakasnitev D (slika 12), kar omogoča tej mreži, da se uči spoznavati ali generirati časovno variabline, pa tudi prostorske vzorce. Tako se izhodni signal v naslednjem koraku vrača nazaj v mrežo kot vhodni signal, kar mreži omogoča dobro spoznavanje časovne odvisnosti signala. V skriti plasti oziroma povratni plasti vsebuje nevrone s tangentno transformacijsko funkcijo, v izhodni plasti pa nevrone z linearno transformacijsko funkcijo. Ta edinstvena kombinacija ji omogoča aproksimirati katerokoli funkcijo s končnim številom nezveznosti s poljubno natančnostjo. Takšne mreže se najpogosteje uporabljajo za napovedovanje spremenljivk, katerih vrednost je zelo časovno odvisna. Izredno uporabnost so te mreže pokazale pri napovedovanju časovnih vrst na podlagi preteklih podatkov.

Slika 12: Elmanova nevronska mreža s povratno zanko z zakasnitvijo D.



Vir: Dermuth, Beale, 1997 str. 265.

Hopfieldova nevronska mreža

Povratna zanka pri tej mreži povezuje izhod z vhodom. Mreža si preko te povratne povezave zapomni izhodne vrednosti za določene vhodne vrednosti. Prav te »zapomnjene« izhodne vrednosti mreža prikliče, kadar ima opravka s podobnimi vhodnimi vrednostmi. Zakasnitev v povratni zanki tu povzoroča, da imamo vedno shranjen podatek iz prejšnega koraka, ki se nato zopet uporabi kot vhod v trenutni korak. S tem ko si mreža hrani reference za prihodnje podatke, se je sposobna naučiti začasnih vzorcev. Mreža shranjuje določene množice »ravnovesnih« točk, v katerih so zagotovljeni prvotni pogoji. Pričakujemo, da se bo mreža v eni od teh točk tudi ustalila. Slabost te mreže je v tem, da so nekatere ravnovesne točke lažne, oziroma napačne. Število teh moramo zmanjšati na minimalno vrednost že, ko načrtujemo mrežo, število pravilnih želenih ravnovesnih točk pa narediti čim večje. Za načrtovanje takšne mreže, torej ravnovesnih točk, uporabljamo rešitve, ki temeljijo na logiki »fuzzy-logic«. Tako ne postavimo jasne meje med pravilnimi in nepravilnimi rešitvami, ampak se osredotočimo zgolj na rešitve, ki so dovolj ustrezne. Takšne mreže se uporabljajo predvsem za pojasnjevanje oziroma čiščenje vhodnih vrednosti. Bolj preprosto povedano, ta mreža nam za poljubno vhodno vrednost kot izhodno vrednost vrne eno od ravnotežnih vrednosti, na katerih smo mrežo osnovali, ki najbolj odgovarja oziroma je najbližje vhodni.

3.4. Učenje nevronske mreže

Najpomembnejša karakteristika nevronskih mrež je njihova sposobnost učenja. Zaradi različnosti arhitekture različnih tipov mrež so tudi njihovi načini učenja različni. V osnovi ločimo učenje mrež na nenadzorovano in nadzorovano. Pri prvem gre za učenje brez zunanjega posredovanja. Mreža se z mnogokratnim ponavljanjem privaja na določen način obnašanja. Večinoma takšno učenje temelji na razvrščanju vhodnih vektorjev na končno število izhodnih. Pri nadzorovanem učenju je bistveno to, da imamo za določene vhodne vrednosti že dane želene izhodne vrednosti. S spreminjanjem uteži znotraj mreže dosežemo, da nam mreža pri določenih vhodnih vrednosti da izhodne vrednosti, ki so pri želeni natančnosti podobni znanim. Nadzorovano učenje ponavadi poteka tako, da mreži dajemo kot vhode vrednosti, za katere poznamo želene rezultate. Te vhodne vrednosti imenujemo učna množica. Izhode iz mreže nato primerjamo z že znanimi »pravilnimi« rezultati in izračunamo napako. Najpogosteje zato uporabljamo enačbo za izračun vsote kvadratne napake (enačba 2, na str. 10). Če je izračunana napaka večja od želene, popravimo uteži v mreži in postopek ponovimo. Postopek učenja se konča, ko je napaka rezultatov dovolj majhna, torej ko so izhodne vrednosti dovolj natančne. Takšen način učenja se imenuje vzvratno učenje, saj napako pošiljamo nazaj skozi mrežo in tako popravljamo uteži v mreži.

Pri vzvratnem učenju poznamo še različico serijskega vzvratnega učenja, pri katerem najprej skozi mrežo pošljemo celotno učno množico, ter izmerimo celotno napako, šele nato pa ustrezno popravljamo uteži. Serijskemu vzrvratnem učenju pravimo tudi adaptacija mreže.

Poznamo več postopkov, po katerih med učenjem prilagajamo uteži in odmike v nevronih. Pri najbolj enostavnem najprej popravljamo konstantni odmik nevrona, ki ga povečamo ali zmanjšamo za določeno konstantno vrednost, ki ji pravimo tudi hitrost učenja, s čimer rezultat približamo želenemu. Uteži vhodov v nevron popravljamo tako, da vrednost uteži zmanjšamo oziroma povečujemo v skladu z stopnjo učenja. Vrednost uteži povečujemo pri tistih vhodnih signalih, katerih vrednost je najbližje želeni izhodni, in zmanjšujemo tistim, katerih vrednost se od želene najbolj razlikuje.

Postopek učenja nevronske mreže je dokaj kompleksen, zato so zanj značilni določeni problemi, ki se ob tem pojavljajo. Najpogostejši problem je generalizacija. Ta pojav povzroča, da se mreža predobro nauči učno množico in se zna zaradi tega odzivati le na znane vhodne vrednosti. Napake izhodnih so pri mreži znanih vhodnih vrednosti sicer zelo majhne, je pa mreža nesposobna generalizacije vzorcev v vhodnih vrednostih in s tem produciranja ustreznih rezultatov za njej neznane vhodne vrednosti. To pomankljivost pri učenju v glavnem odpravljamo z regularizacijo mreže in ustreznostne funkcije[1].

3.5. Programska oprema za delo z nevronskimi mrežami

Za delo z nevronskimi mrežami obstajajo mnoga različna programska orodja. Nekatera so zelo kompleksna in omogočajo delo z velikanskimi mrežami in so namenjena močnim super-računalnikom. Obstaja tudi mnogo orodij za delo z nevronskimi mrežami, ki delujejo na osebnih računalnikih. Ta se razlikujejo predvsem po izbiri različnih mrež, ki jih podpirajo, in pa namenu uporabe. Glede na lastne potrebe sem se odločil za uporabo programskega orodja Matlab. To orodje sicer v osnovi služi za reševanje kompleksnih matematičnih problemov, vendar podpira mnoge razširitve, med katerimi so tudi nevronske mreže. Tako lahko s tem orodjem ustvarjamo, učimo in uporabljamo različne nevronske mreže. Vse to počnemo preko ukazne vrstice ali pa že vnaprej pripravljenih programov.

Za delo z nevronskimi mrežami z uporabo orodja Matlab je potrebno dobro poznavanje samega orodja. Samo krmiljenje mrež in prilagajanje podatkov je dokaj zamudno in kompleksno opravilo, zaradi česar je takšno orodje zelo verjetno popolnoma neustrezno za praktično uporabo. Je pa po drugi strani zaradi velikanskega izbora različnih mrež in transformacijskih funkcij zelo

[1] Ustreznostna funkcija nam meri ustreznost rezultatov glede na vhode. Največkrat je to funkcija vsote kvadratov napak.

primerno za raziskovalno uporabo. Zelo dobro pri tem orodju je tudi to, da je v osnovi namenjeno matematičnim operacijam, zaradi česar podpira vrsto operacij, ki močno olajšajo prirejanje podatkov za uporabo v nevronskih mrežah.

V prilogi so različni programi napisani v programskem jeziku okolja Matlab. Te programe sem napisal, da sem si olajšal velikokrat ponavljajoča se opravila. Tako sem napisal program, za ugotavljanje in izločanje trenda, za normalizacijo časovne vrste, za učenje nevronske mreže in za simulacijo z uporabo nevronske mreže.

3.6. Uporaba v današnjem svetu

Uporabo nevronskih mrež lahko dandanes zasledimo že na mnogih področjih. Velikokrat so nevronske mreže integriran del kompleksnih sistemov, ki jih uporabljamo vsakodnevno, pa se njihove prisotnosti sploh ne zavedamo. Spletni brskalniki jih uporabljajo za razvrščanje podatkov, meteorologom pomagajo pri napovedovanju vremena, srečamo jih celo v novejših »pametnih« samodejnih menjalnikih v avtomobilih. Vedno več pa se uporabljajo tudi v poslovnem svetu, kot del sistemov za odločanje ali pa v kompleksnih informacijskih sistemih.

Zlasti v sistemih za odločanje so nevronske mreže vedno bolj prisotne, kot alternativa statističnim metodam pri napovedovanju ekonomskih dejavnikov. Tako se pogosto uporabljajo za napovedovanje gibanj deviznih tečajev, borznih indeksov in celo samih cen vrednostnih papirjev.

4. Vrednostni papirji

Pomen vrednostnih papirjev v sodobnem svetu narašča. Vrednostne papirje lahko uporabljamo za različne namene. Rabijo nam lahko kot kreditno sredstvo, kot instrument negotovinskega plačevanja, kot podlaga za koncentracijo kapitala oz. za zbiranje sredstev, kot pospeševalec cirkulacije blaga, kot instrument financiranja, za urejanje količine denarja v obtoku itn. (Puharič, 1995, str. 183).

Obligacijski zakonik (OZ) vrednostni papir opredeljuje kot pisno listino, s katero se izdajatelj zavezuje, da bo izpolnil na njej zapisano obveznost njenemu zakonitemu imetniku. (Obligacijski zakonik, 2001, 212. člen). 213. člen OZ določa bistvene sestavine vrednostnega papirja, in sicer mora vrednostni papir obsegati sledeče:

1. označbo vrste vrednostnega papirja;
2. firmo in sedež oziroma ime in prebivališče izdajatelja vrednostnega papirja;

3. firmo ali ime osebe, na katero se glasi oziroma ki odreja, na koga se papir glasi, ali označbo, da se glasi na prinosnika;
4. natančno določeno obveznost izdajatelja, ki izhaja iz vrednostnega papirja;
5. kraj in datum izdaje vrednostnega papirja, pri tistih, ki so izdani v seriji, pa tudi njegovo serijsko številko;
6. podpis izdajatelja vrednostnega papirja oziroma faksimile podpisa izdajatelja papirja, izdanega v seriji.

Listina, ki ne vsebuje katerekoli izmed bistvenih sestavin, ni vrednostni papir.

Vrednostni papir torej predstavlja terjatev imetnika papirja do izdajatelja. Kakšna je ta terjatev, je podrobno zapisano na samem vrednostnem papirju. Izdajatelj papirja pa je zavezan, da izpolni na papirju zapisano obveznost v odrejenem roku, oziroma roku dospetja vrednostnega papirja. S prenosom lastništva vrednostnega papirja se prenaša tudi terjatev od starega do novega imetnika papirja. (Cigoj, 1978, str. 246). Vrednostni papirji se lahko glasijo na ime, na prinosnika ali po odredbi (OZ, člen 214). Terjatev iz imenskega vrednostnega papirja je zavezanec dolžan izpolniti tistemu, ki je na papirju poimensko naveden, terjatev iz prinosniškega papirja prinosniku, terjatev iz papirja po odredbi (ordrskega papirja) pa po odredbi tistega, na čigar se glasi.

Obveznosti, terjatev in pravic iz vrednostnega papirja ne moremo izvrševati, ne da bi listino tudi imeli. Izhajajoč iz načela inkorporacije vrednostni papirji vsebujejo pravico, ki je utelešena (inkorporirana) v listini. Prav zato pravic iz vrednostnega papirja ne moremo ne ustanoviti in ne uveljavljati in tudi ne prenašati, ne da bi listino pri tem tudi lahko pokazali in tako dokazali obstoj pravnega temelja iz določenega vrednostnega papirja (Puharič, 1995, str. 184).

Poglavitna značilnost vrednostnih papirjev je v tem, da imajo samostojno pravno naravo in da so ločeni od posla, zaradi katerega so bili izdani. Kavzalni posel (torej posel, na podlagi katerega je bil papir izdan) je pri vrednostnih papirjih pomemben le z vidika razmerja med dolžnikom in prvim pridobitnikom vrednostnega papirja. Vsi naslednji pridobitniki vrednostnega papirja pa pravico uveljavljajo iz samega papirja (Puharič, 1995, str. 184).

Po vsebini pravic delimo vrednostne papirje na:

- obligacijskopravne vrednostne papirje, ki vsebujejo določeno obligacijsko pravico oz. upnikovo terjatev za denar ali kakšno drugačno dajatev ali storitev; npr. ček, menica, certifikat, blagajniški zapis
- stvarnopravne, torej tiste, v katerih je utelešena stvarna pravica in predstavljajo stvar ali blago samo. Upravičenec na podlagi takšnega papirja nima le pravice zahtevati izročitve stvari, ki so v papirju navedene, temveč lahko z izročitvijo papirja prenese na novega upravičenca tudi stvarno pravico na blagu oz. blago samo ; npr. skladiščica, nakladnica

- korporacijske, ki vsebujejo krog članskih pravic v določeni pojavni obliki kapitalskih družb; npr. delnica

Zaradi zgoraj opisanih značilnosti je z vrednostnimi papirji mogoče trgovati. Nekateri imajo tudi svojo tržno ceno in lahko z njimi trgujemo kot z blagom. Ponavadi imajo takšni vrednostni papirji daljše roke dospelosti. Trgovanje z njimi se odvija na trgu vrednostnih papirjev.

Trg vrednostnih papirjev je prost ekonomski sistem, na katerega vplivajo zgolj spremembe v ponudbi in povpraševanju. Organiziran trg vrednostnih papirjev predstavlja borza, ki ureja in posreduje podatke o cenah posameznih vrednostnih papirjev, posreduje pri menjavi in tudi nadzira trgovanje ter preprečuje morebitne nepravilnosti. Na borzi kotira več različnih tipov korporacijskih vrednostnih papirjev. V osnovi jih delimo na delnice in obveznice.

4.1. Delnice

Delnica predstavlja del osnovnega kapitala delniške družbe. Torej je to lastnina dela kapitala vloženega v neko podjetje, zaradi česar se imetniku tega vrednostnega papirja pripisujejo tudi pravice, ki izhajajo iz lastnine. Delnice delimo na materializirane in nematerializirane delnice, glede na to v kakšni obliki se pojavljajo. Po vsebini pravic, ki jih ima imetnik delnice, pa jih delimo na navadne (redne) in prednostne (ugodnostne) delnice.

4.1.1. Navadne delnice

Navadne delnice dajejo imetniku naslednje pravice (ZGD, 1998, 177. člen):

- pravico do udeležbe pri upravljanju družbe
- pravico do dela dobička
- pravico do ustreznega dela preostalega premoženja po likvidaciji ali stečaju družbe

Del dobička podjetja se po sklepu vodstva družbe deli med delničarje. Znesek, ki pripada eni delnici, se imenuje dividenda. Prav višina dividende pa je glavni faktor, ki določa ceno delnice na trgu vrednostnih papirjev. Cena delnice je namreč zelo odvisna od sedanje vrednosti pričakovanih dividend.

4.1.2. Prednostne delnice

Prednostne delnice njihovim lastnikom poleg pravic, ki jih zagotavljajo navadne delnice, zagotavljajo še določene ugodnosti, kot je prednost pri izplačilu dividend ali prednost pri delitvi premoženja ob likvidaciji ali stečaju združbe.

4.2. Obveznice

Obveznice so dolžniški vrednostni papirji, ki dajejo imetniku pravico do izplačila glavnice in morebitnih obresti oziroma drugih donosov.

Po vsebini pravic delimo obveznice na navadne in participativne, ki imetniku poleg obresti dajejo tudi pravico do udeležbe pri dobičku izdajatelja obveznice.

Ceno obveznice določa glavnica zapisana na obveznici, obrestna mera in rok dospetja obveznice. Lahko rečemo, da je cena obveznice izražena s sedanjo vrednostjo vseh prihodnjih obresti in glavnice, ki se izplača ob dospetju.

4.3. Naložbe v vrednostne papirje

Nakup vrednostnega papirja ekonomsko lahko enačimo z investicijo. Pri začetni nakupni vrednosti (vrednosti investicije) imamo prek določenega obdobja določene donose. Naložbe v vrednostne papirje zato ocenjujemo po metodah za ocenjevanje investicij. Tveganje je pri posameznih oblikah vrednostnih papirjev različno. Najbolj tvegane so naložbe v delnice, najmanj pa v državne obveznice, saj predstavljajo terjatev do države. Tudi življenjska doba investicije se razlikuje od vrste vrednostnega papirja. Pri delnicah je ta enaka življenski dobi družbe v primeru, da delnice ne prodamo, pri obveznicah pa do dospetja. Pri obveznicah predstavljajo donos anuitetne obresti ter glavnica ob dospetju, pri delnicah pa dividende. Če vrednostne papirje prodamo pred dospetjem nam razlika med prodajno in nabavno ceno daje določen donos ali izgubo.

Da bi se lahko kar najbolj racionalno odločili za investicije v vrednostje papirje, bi morali poznati obrestne mere in njihove cene v prihodnosti. Pri tem si pomagamo s sistemi za napovedovanje, s katerimi poskušamo napovedati cene vrednostnih papirjev in s tem pretehtati uspešnost investicije.

5. Napovedovanje z nevronsko mrežo

Nevronske mreže so zelo primerne za napovedovanje tečajev vrednostnih papirjev (Pham, Liu, 1995, str. 83). Uporabljajo se tako bolj enostavne mreže, kot so večplastni perceptroni, kakor tudi bolj kompleksne, kot je Elmanova povratna mreža. Za napovedovanje za daljše obdobje pa so bolj v uporabi radialno zasnovane mreže. Kot vsi sistemi za napovedovanje so tudi nevronske mreže zmožne napovedovati zgolj deterministične pojave. Zaradi tega se jih vedno bolj uporablja kot pomoč pri napovedovanju ekonomskih dejavnikov. V tem poglavju bom z nevronsko mrežo poskušal napovedati gibanje tečajev nekaj izbranih vrednostnih papirjev in preverjal ustreznost takšne napovedi.

Uporabljal bom tri različne mreže. Elmanovo povratno mrežo in troplastni perceptron za napovedovanje v časovnem obdobju za dva tedna vnaprej, ter radialno mrežo s katero bom poskušal napovedati dolgoročno gibanje vrednosti tečaja. Vse tri mreže bom gradil s pomočjo programskega orodja Matlab. Delo si bom olajšal s tem, da bom za ponavljajoča se opravila napisal programe v programskem jeziku tega orodja. Ti programi bodo pripeti v prilogi.

5.1. Prilagajanje podatkov

Za vrednostne papirje je značilno, da so to kompleksni dinamični sistemi, ki se obnašajo deterministično kaotično, prisotno pa imajo tudi stohastično komponento (Jamšek, 2000, str. 18).

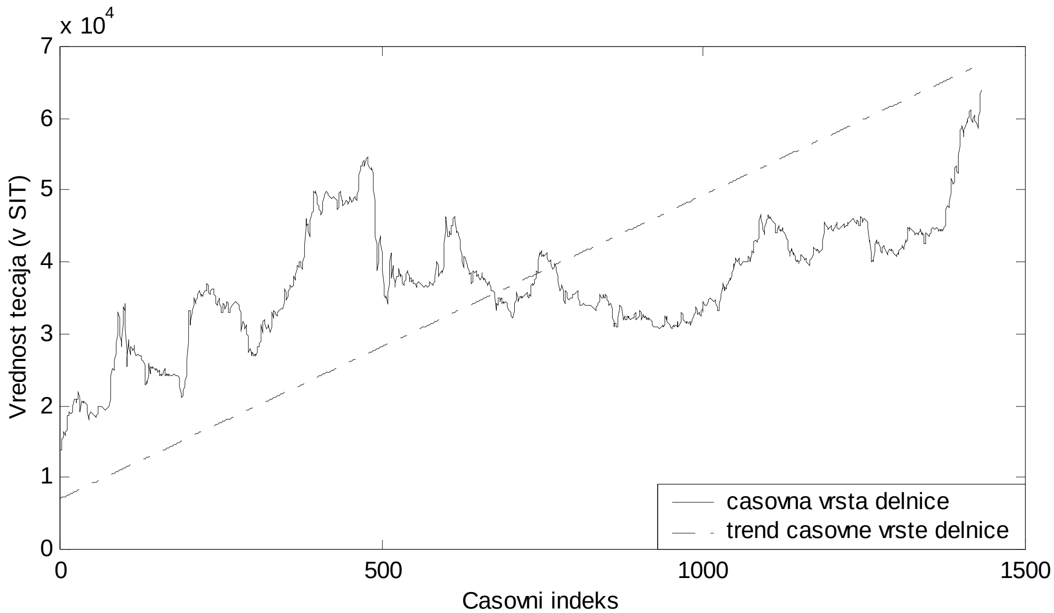
To pomeni, da je cena vrednostnega papirja determinističen pojav in ga je zato možno v neki meri napovedati v prihodnosti z uporabo ustreznih metod. Podatki, s katerimi se bom ukvarjal, predstavljajo cene treh naključno izbranih vrednostnih papirjev, ki kotirajo na slovenski borzi. Izbrani vrednostni papirji so delnici LEKA in SAVA, ter PID TG1N. Cene vrednosti papirjev so predstavljene za posamezni dan v letu vrednostno v SIT.

Pri uporabi nevronskih mrež za napovedovanje moramo biti pozorni na vrednost podatka v časovni vrsti. Glede na to, da uporabljamo v mrežah transformacijske funkcije, ki delujejo bodisi v območju med 0 in 1 ali območju med -1 in 1, moramo vrednosti podatkov v časovni vrsti že na samem začetku ustrezno predelati. Mreže so najbolj učinkovite pri napovedovanju stacionarnih časovnih vrst, saj pri časovnih vrstah s prisotnim trendom lahko prehitro pride do generalizacije. Zato bomo iz časovne vrste odstranili trend. Da bi v mrežah lahko uporabljal nevrone s hiperbolično tangentno funkcijo, pa bom vrednosti časovne vrste transformiral na območje med -1 in 1.

Časovno vrsto bom ustvaril iz vrednosti tečaja vrednostnega papirja po posameznih dnevih. V časovni vrsti ne bom zajel podatka za dneve, ko borza ni delovala, saj so ti podatki povsem nepomembni. Ponavljanje podatka v dnevih, ko na borzi ni bilo trgovanja, bi povzročalo tudi težave pri ugotavljanju zakonitosti časovne vrste. Zajete podatke o tečaju obravnavanega vrednostnega papirja bom uredil v časovno vrsto (slika 14).

V časovni vrsti bom najprej poiskal trend in ga iz podatkov odstranil[1]. Časovno vrsto bom nato transformiral tako, da bodo podatki zavzemali interval med -1 in 1. Nazadnje bom časovno vrsto še centraliziral. To storimo tako, da vsem vrednostim v časovni vrsti prištejemo oziroma odštejemo povprečno vrednost podatka v vrsti, s čimer dosežemo, da bo izračunana povprečna vrednost enaka 0. S tem izločimo vse trendne vplive in dobimo zares stacionarno časovno vrsto.

Slika 14: Časovna vrsta delnice LEKA z vrisanim trendom



Vir: Statistični podatki ljubljanske borze d.d.

5.1.1. Določanje in odstanjevanje trenda

Preden začnemo z računanjem trenda, je pametno preveriti ali je v naši časovni vrsti (slika 14) ta sploh prisoten. To najlažje storimo z izračunom koeficientov avtokorelacije[2], ki jih prikažemo v korelogramu (slika 15). Če so koeficienti avtokorelacije za začetne zamike visoki in s

---

[1] Program, ki sem ga napisal v programskem jeziku orodja Matlab, je v prilogi 2.

[2] Program, ki sem ga napisal za izračun koeficientov avtokorelacije, je v prilogi 1.

povečevanjem časovnega zamika počasi padajo, je v časovni vrsti prisoten trend. V korelogramu bodo visoki koeficienti prikazani z močno pozitivno ali močno negativno vrednostjo, nizki koeficienti pa z vrednostmi blizu nič.

Ko ugotovimo prisotnost trenda, je trend treba natančno določiti. Zato obstaja več metod, ki se pogosto uporabljajo v statistiki. Zaradi majhnega števila podatkov v obravnavani časovni vrsti je naprimernejša metoda za izračun trenda regresijska premica, oz. drugače imenovana metoda za izračun linearnega trenda (Arh, 1996, str. 94). Po tej metodi nam trend predstavlja linearna enačba, ki jo lahko zapišemo kot $T_x = \alpha + \beta x$ (slika 14), pri tem pa predpostavljamo da je x=0 v sredini časovne vrste. Iz tega lahko izpeljemo, da je vrednost α enaka povprečni vrednosti podatka v časovni vrsti $\alpha = \overline{Y}$, kar pa je tudi vrednost trenda v izhodiščni točki x=0. Smerni koeficient trenda β pokaže spremembo vrednosti linearnega trenda T na eno časovno enoto, torej nam pove, za koliko se vrednost linearnega trenda poveča ($\beta > 0$) ali pa zmanjša ($\beta < 0$) v vsaki časovni enoti vrste. Vrednost smernega koeficienta izračunamo po naslednji enačbi:

$$\beta = \frac{\sum Yx}{\sum x^2} \qquad (11)$$

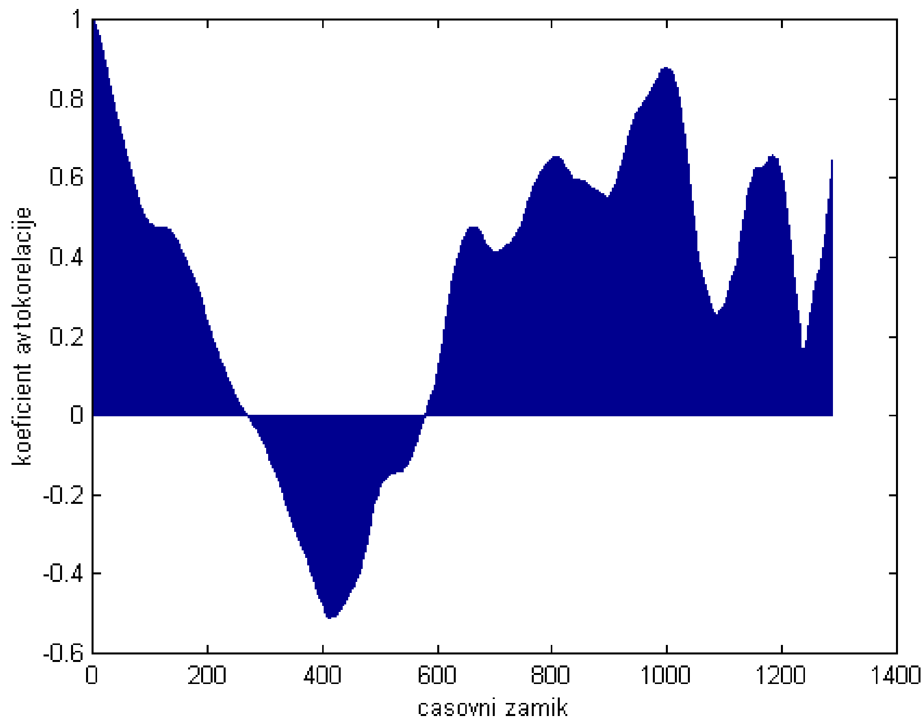
Enačbo linearnega trenda nato transformira tako, da dobimo izhodiščno točko x=0 na začetku časovne vrste t=0. To storimo tako, da v enačbo linearnega trenda z izhodiščno točko v sredini časovne vrste vstavimo namesto x njeno linearno transformacijo $x = t - \bar{t}$. Po enačbi:

$$\alpha' = \alpha - \beta \frac{N+1}{2} \qquad (12)$$

dobimo vrednost parametra α', ki nam kaže vrednost linearnega trenda na začetku časovne vrste.

Trend iz časovne vrste odstranimo tako, da od vsakega podatka v časovni vrsti odštejemo vrednost trenda, izračunano po enačbi regresijske premice, v isti časovni enoti.

Slika 15: Korelogram časovne vrste delnice lekA



Vir: Mathlab.

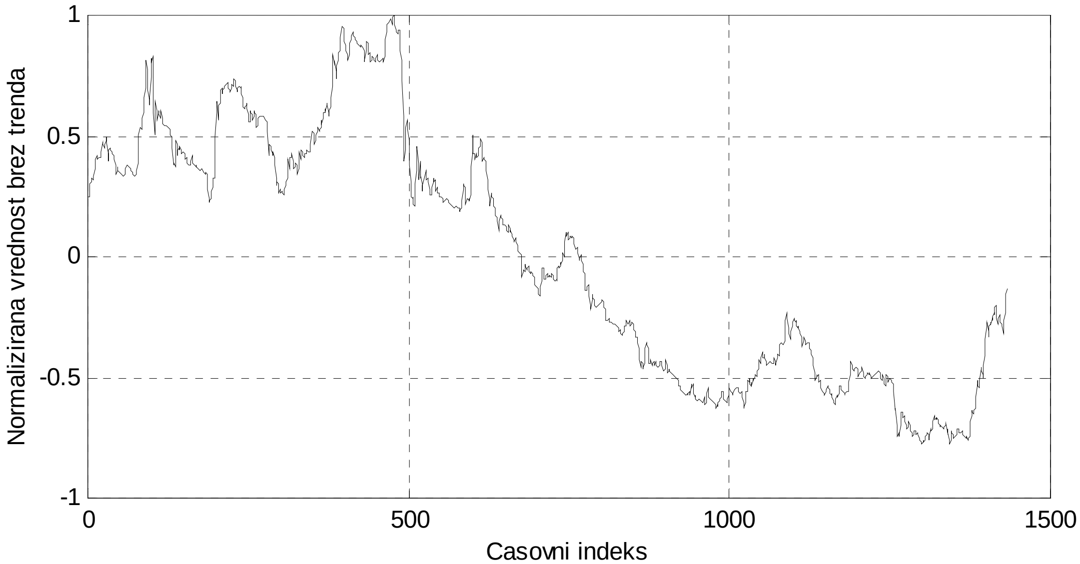
5.1.2. Normalizacija in centralizacija stacionarne časovne vrste

Z normalizacijo časovne vrste transformiramo vse podatke v časovni vrsti tako, da zasedejo vrednosti med -1 in 1. To naredimo tako, da vsak podatek v časovni vrsti delimo z največjo absolutno vrednostjo podatka v časovni vrsti.

Če želimo dobiti zares stacionarno časovno vrsto, katere podatki se bodo gibali okrog vrednosti 0 v intervalu med -1 in 1, moramo našo časovno vrsto še centralizirati. To storimo tako, da izračunamo povprečno vrednost podatka v časovni vrsti. To povprečno vrednost nato odštejemo od vsakega podatka v časovni vrsti[1]. Če je časovna vrsta centralizirana, je povprečna vrednost podatka enaka 0 (slika 16).

[1] Program za centralizacijo, normalizacijo in odstranitev trenda časovne vrste je v prilogi 2.

Slika 16: Časovna vrsta delnice lekA z odstranjenim trendom, ter po normalizaciji in centralizaciji



Vir: Mathlab.

5.1.3. Izbira učne in testne množice

Za preizkus modela za napovedovanje časovnih vrst na osnovi nevronskih mrež bom poskusil napovedati časovne vrste delnic LEKA in SAVA, ter časovno vrsto delnice PID TG1N. Izbrane časovne vrste bom razdelil na učno množico, s katero bom učil nevronske mreže, ter testno množico, s katero bom preverjal ustreznost modela. Razdelitev časovnih vrst, je prikazana v tabeli 1.

Tabela 1: Delitev časovnih vrst na učne in testne množice

Časovna vrsta	Celotno število podatkov v vrsti	Izbrani podatki za v učno množico	Izbrani podatki za v testno množico
LEKA	1432	1 - 1306	1307 – 1426
SAVA	596	1 - 470	471 – 590
TG1N	714	1 - 588	589 - 708

5.2. Način napovedovanja

Poznamo dva osnovna tipa napovedovanja z nevronskimi mrežami. Pri enokoračnem napovedovanju damo mreži kot vhod znane vrednosti, mreža pa nam da kot rezultat napovedano naslednjo vrednost v seriji. Pri večkoračnem napovedovanju po prvem koraku vzamemo rezultat mreže in ga pošljemo nazaj skozi mrežo kot vhod. Takšen postopek ponavljamo poljubno dolgo. Pri enosmernih mrežah uporabljamo kot vhod v mrežo vektor vrednosti za nekaj zaporednih obdobij. Pri večkoračnem napovedovanju moramo torej premakniti vrednosti v vhodnem vektorju navzgor, kot zadnjo pa dodati izhodno vrednost iz mreže. Pri mrežah s povratno zanko pa imamo kot vhod le trenutno vrednost, saj si mreža preko povratne povezave sama zapomni stanja v prejšnih korakih. Enokoračno napovedovanje uporabljamo zgolj za zelo kratkoročno napoved, oziroma napoved ene naslednje vrednosti v časovni vrsti, večkoračno napovedovanje pa za napovedovanje v daljšem časovnem obdobju[1].

Pri napovedovanju časovnih vrst za daljše obdobje se največkrat uporabljajo radialno zasnovane nevronske mreže, ki so sposobne aproksimirati nelinearno funkcijo. Takšne mreže zaradi svoje narave niso sposobne dajati natančnih napovedi za zelo kratki rok, so se pa sposobne naučiti in ugotoviti sezonska in ciklična nihanja in trende časovne vrste.

5.3.1. Napovedovanje z uporabo večplastnega perceptrona

Z večplastnim perceptronom bom poskušal napovedati ceno vrednostnega papirja za obdobje dveh tednov, kar predstavlja deset zaporednih vrednosti v časovni vrsti. Uporabil bom troplastni perceptron, ki ima v vhodni plasti 5 nevronov s hiperbolično tangnetno transformacijsko funkcijo, na katere bom priklopil 5 vhodnih signalov, ki bodo predstavljali 5 zaporednih vrednosti v časovni vrsti[2]. Skrito plast bo sestavljajo šest nevronov, torej eden več kot v vhodni plasti. Tudi nevroni v skriti plasti bodo imeli hiperbolično tangentno transformacijsko funkcijo. V izhodni plasti pa bo le en nevron z linearno transformacijsko funkcijo, saj iz mreže potrebujem le en izhod, ki bo predstavljal naslednjo vrednost v časovni vrsti. Mrežo bom učili v 10000 korakih in vsakič po postopku učenja mreže popravljal uteži. Pri napovedovanju za obdobje dveh tednov bom rezultate mreže za naslednji dan pošiljal nazaj v mrežo kot prvo izmed petih vhodnih vrednosti. Vektor vhodnih vrednosti bom pred tem premaknil za eno vrstico navzdol ter odstranil zadnjo vrednost. Dobljene rezultate bom nato primerjal z resničnimi vrednostmi v istem obdobju (slika 17).

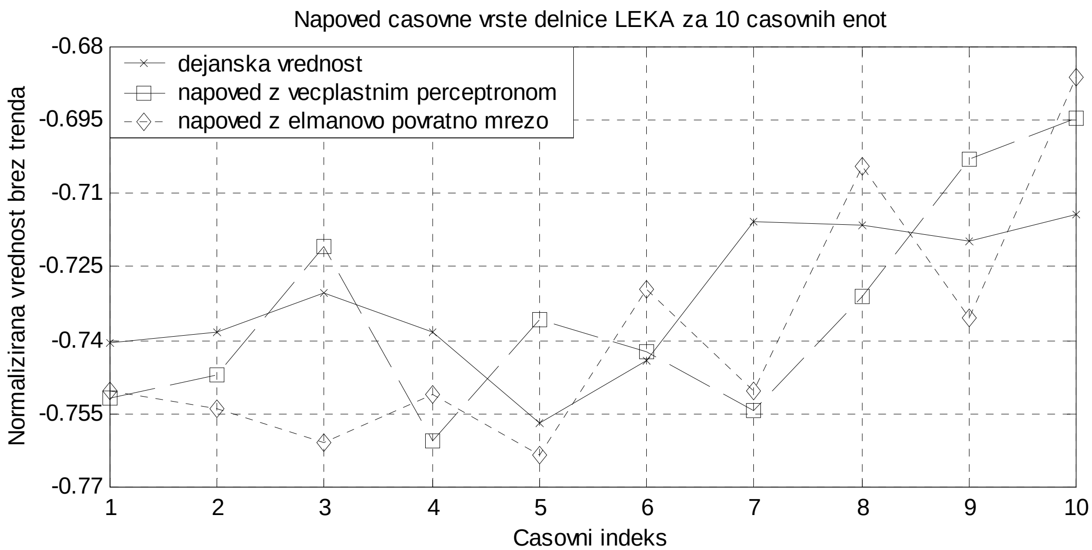
5.3.2. Napovedovanje z uporabo Elmanove povratne mreže

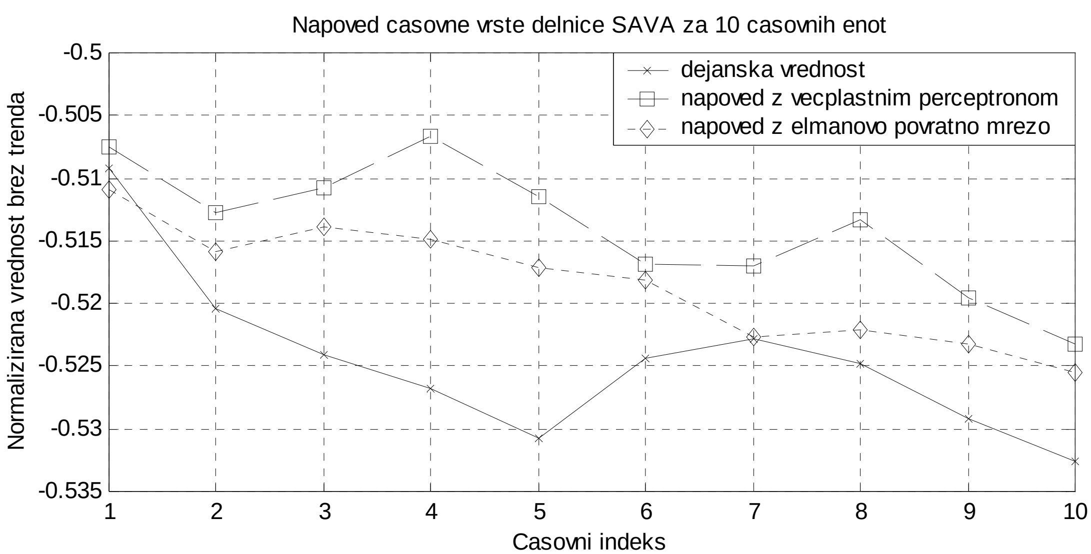
Tudi z elmanovo mrežo bomo poskusil napovedati časovno vrsto za obdobje dveh tednov. Uporabil bom mrežo z petimi vhodnimi vrednosti, ki bodo tako kot pri večplastnem perceptronu predstavljali zaporedne vrednosti časovne vrste. Prek povratne zanke bo mreža tokom učenja

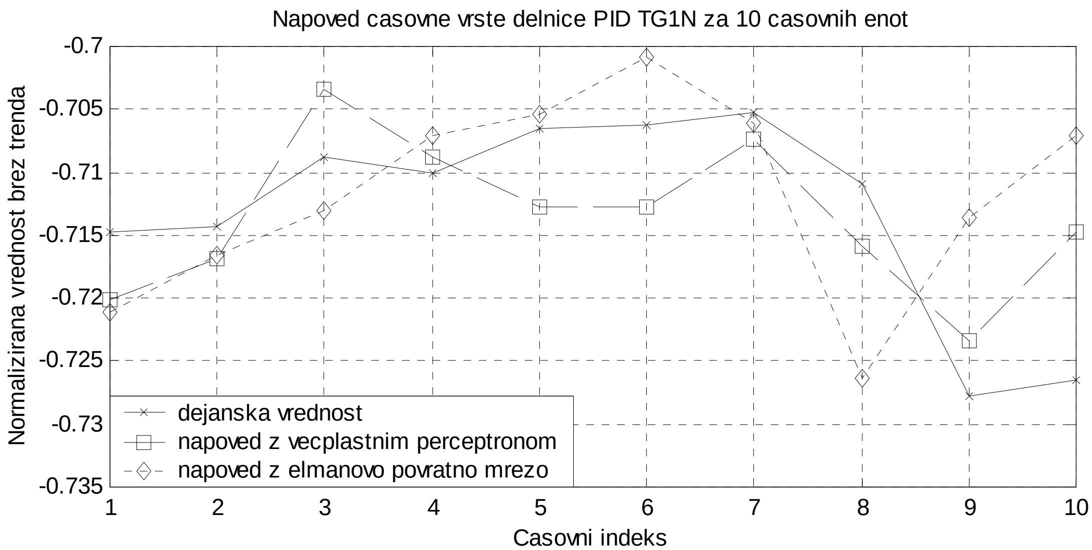
[1] Program v programskem jeziku Matlab, ki sem ga napisal za večkoračno napovedovanje, je v prilogi.
[2] Število pet sem uporabil, ker je to število podatkov, ki predstavlja obdobje enega tedna.

lahko sama ugotavljala napake pri posameznih napovedih. Za razliko od večplastnega perceptrona bo ta mreža sestavljena le iz dveh plasti. V skriti plasti bo 15 nevronov s hiperbolično tangentno transformacijsko funkcijo, v izhodni plasti pa le en nevron z linearno transformacijsko funkcijo. Edina izhodna vrednost bo napoved vrednosti časovne vrste za eno časovno enoto vnaprej. Mrežo bom, tako kot večplastni perceptron, učil v 10000 korakih. Pri napovedovanju bom tako kot pri uporabi perceptrona uporabil večkoračno napovedovanje, tako da bom dobljene rezultate mreže pošiljal nazaj kot vhode (slika 17).

Slika 17: Napoved časovnih vrst za 2 tedna vnaprej





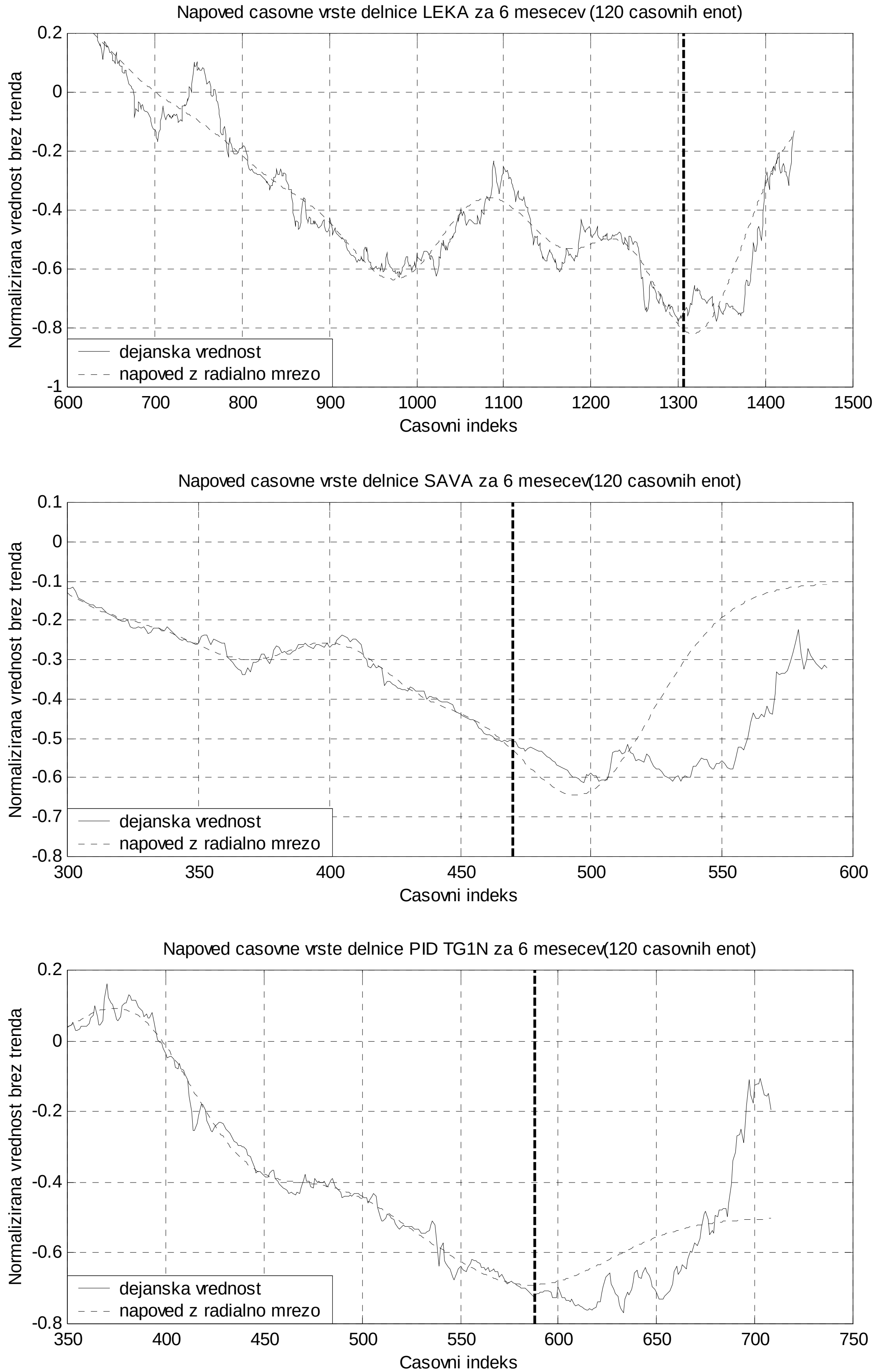


Vir: Matlab.

5.3.3. Napovedovanje z uporabo radialno zasnovane mreže

Radialno zasnovano mrežo bom uporabil za aproksimacijo funkcije časovne vrste. Mreža se bo naučila zakonitosti valovanj v tej funkciji in bo predvidoma sposobna napovedati ta valovanja tudi za prihodnje vrednosti. Tako bom lahko napovedal vrednosti za daljše časovno obdobje in sicer za 120 časovnih enot, kar v tem primeru kaže na čas približno šestih mesecev. Pri učenju mreže moramo biti pozorni, da mreže ne naučimo preveč podrobno, saj potem ne bo sposobna napovedovati vrednosti za v njej neznanih prihodnjih obdobjih. Pri učenju radialno zasnovane mreže vedno začnemo s prazno mrežo, ki vsebuje le vhodno matriko in nič nevronov. Izhod iz takšne mreže je ne glede na vhodno vrednost zmeraj enak nič. Izhod iz mreže primerjamo z želenimi izhodnimi vrednostmi in dodajamo nevrone z radialno funkcijo, katere uteži in odmik se ustrezno prilagodijo. V mrežo dodajamo poljubno število nevronov, ki reagirajo na določene vhodne signale in v izhod vnašajo dodatna radialna nihanja, s čemer rezultat vedno bolj približujejo želenemu. Paziti moramo, da mreže ne naučimo preveč, saj bo, čeprav se bo za znane vrednosti skoraj popolnoma prilegala želenim rezultatom, slabo ravnala z neznanimi vhodnimi signali. Izhod iz učene mreže je aproksimacija funkcije časovne vrste, za katero lahko tudi rečemo, da je zglajena funkcija časovne vrste. Mreži bom kot vhode pošiljal zaporedne časovne indekse in jo učil z izhodnimi signali, ki predstavljajo normalizirane vrednosti časovne vrste. Pri napovedovanju bom mreži pošiljal kot vhode časovne indekse za prihodnja obdobja, ki so mreži še nepoznani (slika 18).

Slika 18: 6 mesečna napoved časovne vrste



Vir: Matlab.

5.3.4. Uspešnost modela napovedovanja

Uspešnost modela napovedovanja bom izmeril tako, da bom izmeril razlike med napovedano in dejansko vrednostjo parametra. Izračunanim normaliziranim vrednostnim, ki ne vsebujejo trenda, bi lahko dodal še trendno vrednost, potem pa vrednosti pretransformiral nazaj v prvotno obliko, da bi dobil dejanske podatke izražene v SIT. To bi bilo treba storiti v primeru, če bi želel napovedi uporabiti za odločanje o nakupu. Ker pa me zanima zgolj učinkovitost modela napovedovanja, pa so dovolj dobre normalizirane vrednosti, ki ne vsebujejo trenda. Glede na to, da bom med seboj primerjal učinkovitost napovedi za različne časovne serije in različne modele, bom učinkovitost izražal z relativno napako. Narava normaliziranih vrednosti pa mi omogoča tudi računanje absolutne napake normalizirane vrednosti, ki pa so za razliko od pravih vrednosti časovne serije med seboj tudi primerljive. Po enačbah za merjenje napak (enačbe na str. 10) sem izračunali kazalce in jih povzel v tabeli 2.

Tabela 2: Izmerjene napake napovedi

Delnica	Napoved za 2 tedna z večplastnim perceptronom		Napoved za 2 tedna z elmanovo povratno mrežo		Napoved za 6 mesecev z radialno mrežo	
	PAON	PON	PAON	PON	PAON	PON
LEKA	0,02227	0,00303	0,02449	0,00889	0,43673	-0,39489
SAVA	0,02056	-0,02056	0,01235	-0,01167	0,20231	-0,12615
TG1N	0,00712	0,00073	0,01015	-0,00200	0,22075	-0,00076

Vir. Mathlab.

Izračunane kazalce analiziramo tako, da jih primerjamo z določeno osnovo. Osnova, ki sem jo vzel za primerjave kazalcev, bodo kar izračunani kazalci za druge časovne vrste in pri uporabi različnih modelov. Torej bom kazalce le primerjal med seboj.

Pri primerjavi sem najprej ugotovil, da so kazalci napak izračunani za šest mesečno napoved dosti višji od tistih za dvo tedensko, kar je tudi pričakovati. Opazil sem tudi, da je natančnost napovedovanja modelov od ene časovne vrste do druge močno različna. To pripisujem dejstvu, da imamo različno dolge časovne serije, ki so različno kompleksne, za napovedovanje vseh pa sem uporabljal mreže z enako arhitekturo. Bolj preseneča dejstvo, da Elmanova mreža s povratno zanko ni bolj učinkovita pri napovedovanju od enosmernega večplastnega perceptrona. Razlog za to je najverjetneje v neustrezni arhitekturi Elmanove mreže glede na dan problem.

Pri napovedovanju za dva tedna vnaprej se je bila povprečna absolutna odstotna napaka napovedi pri različnih modelih in po različnih časovnih vrstah gibala med med 0,7 in 2,4 odstotki, kar lahko smatramo kot dokaj natančno.

Napovedovanje časovne vrste z radialno mrežo za šest mesečno obdobje pa se je izkazalo za dokaj nenatančno, razen v primeru časovne serije TG1N. Verjetno je pri ostalih dveh časovnih serijah prišlo do prevelike generalizacije mreže, ali pa je bilo prisotnih preveč neznanih in nepredvidljivih dejavnikov, ki so vplivali na vrednost podatka.

6. Sklep

Nevronske mreže se vse bolj uporabljajo v današnjih modelih za napovedovanje. To je predvsem posledica mnogo enostavnejše in hitrejše uporabe v primerjavi z kompleksnimi matematično statističnimi modeli. Mrežo se definira le enkrat, potem pa se jo lahko poljubno uči na določenih intervalih, za kar je treba zgolj imeti podatke za sestavo učne in testne množice. Mreža je sama sposobna ugotavljati sezonska in ciklična nihanja ter jih posnemati. Rezultati, ki nam jih dajejo nevronske mreže, so vsaj primerljivi tistim, ki bi jih dobili po statističnih metodah, in mnogo bolj razumljivi, čeprav je samo delovanje nevronskih mrež zelo težko razumljivo. Je pa zato delo z njimi dokaj enostavno, vsaj dokler imamo opravka z enostavnimi, ne preveč kompleksnimi problemi. Z večanjem kompleksnosti problema, ki ga rešujemo, se povečuje tudi težavnost uporabe nevronske mreže. Ugotovili smo, da je časovna vrsta delnice dokaj kompleksen sistem. Zato je tudi načrtovanje mreže za napovedovanje takšne vrste dokaj kompleksno. Sama uporaba dobljenega modela pa je še zmeraj enostavna. Zato lahko rečemo, da je takšen model primeren za splošno uporabo, saj ne zahteva posebnega znanja.

Natančnost rezultata je močno odvisna od same zasnove nevronske mreže in pa od načina učenja. Ker so pred vsakim učenjem uteži v mreži nastavljene na naključne vrednosti, je uspešnost učenja tudi v primeru nespremenjenih ostalih pogojev različna. Učenje izboljšamo s tem, da v proces učenja vpeljemo validacijsko množico (kot validacijsko smo uporabili kar testno množico), s katero kontroliramo stopnjo naučenosti. Med samim postopkom učenja lahko opazujemo, kako napaka pri simulaciji podatkov iz validacijske množice do neke stopnje pada, potem pa se začne povečevati, kar je indikacija generalizacije. Ob znaku generalizacije se učenje mreže prekine.

Dobljeni rezultati pri napovedovanju izbranih delnic so dali ugodne rezultate, kar potrjuje uporabnost modelov za napovedovanje osnovanih na nevronskih mrežah. Dokazali smo, da se je mreža res sposobna naučiti nekaterih zakonitosti in na podlagi nabranega znanja napovedati vrednosti v prihodnje.

Ugotovil sem tudi, da z večanjem časovnega horizonta napovedi natančnost le-te močno upada, s tem pa se povečuje tudi tveganje pri uporabi te informacije. Očitno pa je, da bi pri preučevanih delnicah izračunane napovedi dale kar koristne informacije za odločanje o nakupu teh vrednostnih papirjev, kar pa je tudi sam namen modela za napovedovanje.

Ne gre prezreti dejstva, da sem razpolagal z dokaj majhno količino podatkov v vsaki časovni seriji, kar je imelo velik vpliv na učenje nevronske mreže. Iz korelograma časovnih vrst je razvidno, da je pri vseh poleg trenda prisotno tudi določeno ciklično nihanje. Podatki pa so bili premalo obsežni, da bi vsebovali celoten cikel. To je zagotovo vplivalo na rezultate napovedi, predvsem na tiste za daljše obdobje, saj mreža ni znala dobro predvidevati cikličnih nihanj.

Drugo dejstvo, ki govori v prid manjši natančnosti napovedi, je to, da sem uporabljal enako arhitekturo mreže, ne glede na velikost časovne serije. Zato je bila v nekaterih primerih mreža najbrž premalo sposobna posnemati vse značilnosti serije, v nekaterih pa je prišlo celo do generalizacije. To je še zlasti opazno pri dolgoročni napovedi.

Dodaten faktor, ki je otežil napovedovanje, pa je dejstvo, da sem analiziral delnice iz slovenskega trga vrednostnih papirjev, ki pa je dokaj neučinkovit. Zaradi majhnega obsega so cene vrednostnih papirjev močno odvisne od trgovanja z njimi. Že malo večji nakupi lahko zelo močno vplivajo na ceno določenega vrednostnega papirja. Zaradi dinamike trgovanja z vrednostnimi papirji, ki je dokaj neenakomerna, se zato v časovnih vrstah slovenskih vrednostnih papirjev pojavljajo še dodatna slučajnostna nihanja. Večji trgi vrednostnih papirjev so dosti manj pod vplivom različnih dejavnikov, zato se tam cene vrednostih papirjev gibljejo bolj predvidljivo in po več ali manj znanih zakonitostih. V takšnem okolju je uporaba sistemov za napovedovanje mnogo bolj učinkovita.

Zmeraj pa se moramo zavedati, da nam model daje samo informacije, ki so »nenatančne« napovedi. Pričakovati moramo, da se dejanske vrednosti v prihodnosti ne bodo gibale natanko po napovedih. Zato se na takšne modele ni popolnoma za zanašati. Zmeraj moramo pri odločitvah uporabiti še lastne izkušnje, mnenja drugih in vse ostale informacije, ki so nam na voljo, pa še niso vključene v ta model. Napovedane vrednosti pa uporabljamo zgolj kot eno od mnogih informacij pri ugotavljanju smotrnosti nakupa nekega vrednostnega papirja.

Končni sklep je, da so sistemi za napovedovanje na osnovi nevronskih mrež učinkoviti pri opravljanju svoje funkcije, ki pa je nuditi dodatne informacije kot podporo za odločanje. Zaradi enostavnosti in hitrosti uporabe se bodo v prihodnosti prav gotovo uveljavili vsaj na določenih področjih.

Literatura

1. Arh Franc: Statistika 1. Obrazci in postopki. 2. izd. Ljubljana : Ekonomska Fakulteta, 1996. 118 str.
2. Batagelj Teja: Nevronske Mreže : Zbornik posvetovanja Dnevi slovenske informatike. Ljubljana : Slovensko društvo Informatika, 2002, str. 481-485.
3. Cigoj Stojan: Komentar obligacijskih razmerij. Ljubljana : Uradni list Republike Slovenije, 1978, str. 245-264.
4. Dermuth Howard, Beale Mark: Neural Network Toolbox. B.k. : The Math Works, 1997. 742 str.
5. Giles C. Lee, Lawrence Steve, Tsoi Ah Chung: Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference. Machine Learning. New Jersey, Volume 44, 2001, str. 161-183.
6. Gurney Kevin: An Introduction to Neural Networks. London : UCL Press 1, 1996.
7. Hanke E. John, Reitsch G. Arthur: Business Forecasting. 5. izd. Upper Saddle River : Prentice Hall, 1995. 581 str.
8. Haykin Simon: Neural Networks. 2. izd. Upper Saddle River : Prentice-Hall, 1999. 842 str.
9. Hunt J. Kenneth, Irwin R. George: Neural Network Engineering in Dynamic Control Systems. 2. izd. Berlin : Springer-Verlag, 1996. 278 str.
10. Jamšek Janez: Predikcija vrednosti delnice z nevronsko mrežo. Diplomska naloga poslovne šole. Ljubljana : Ekonomska fakulteta, 2000.
11. Kumer Slavko: Stohastični modeli in nevronske mreže pri ravnanju z zalogami. Diplomsko delo visoke poslovne šole. Ljubljana : Ekonomska fakulteta, 1999. 55 str.
12. Masters Timothy: Neural, Novel & Hybrid Algorithms for Time Series Prediction. New York : John Wiley & Sons, 1995. 514 str.
13. Pham Duc Truong, Liu Xing: Neural Networks for Identification Prediction and Control. 2. izd. London : Springer-Verlag, 1995. 238 str.
14. Puharič Krešo Gospodarsko pravo z osnovami prava. Ljubljana : Uradni list Republike Slovenije, 1995. 234 str.
15. Quenouille H. Maurice: The Joint Distribution of Serial Correlation Coefficients. Annuals of Mathematical Statistics. B.k., Institute of Mathematical Statistics, Volume 20, 1949. str. 561-671
16. Ramaswamy Srichander: One-step Prediction of Financial Time Series. Working Papers, B.k. : Basle, 1988. 24 str.
17. Sutton S. Richard, Barto G. Andrew: Reinforcement Learning. [URL: http://www-anw.cs.umass.edu/~rich/book/the-book.html], 3.5.2002.

18. Trippi R. Robert, Turban Efraim: Neural Networks in Finance and Investing. 2. izd. Irwin : Irwin Professional Publishing, 1996. 821 str.

Viri

1. Electronic Statistics Textbook: Statsoft. [URL:http://www.statsoft.com/textbook/stathome.html], 27.4.2002.
2. Statistični podatki ljubljanske borze d.d. [URL: http://www.ljse.si/StrSlo/DirektSk/trgUrTec.htm], 17.5.2002.
3. Obligacijski zakonik (OZ). Ljubljana : Uradni list Republike Slovenije, 2001. str. 181
4. Slovar slovenskega knjižnega jezika (druga knjiga). Ljubljana : Akademija znanosti in umetnosti, 1975. str. 967
5. Zakon o gospodarskih družbah (ZGD). Ljubljana : Udarni list Republike Slovenije, 1998

Priloge

1. Program v programskem jeziku orodja MATLAB za izračun koeficientov avtokorelacije

```
% AVTOKORELACIJA – Program za izracun koeficientov avtokorelacije
%                  casovne vrste
%
% Borut Lukic, 29. 5. 2002
%
%**********************************************************************

% Nalozimo podatke iz tekstovne datoteke. Podatki so v vrsticni obliki
% loceni z znakom za novo vrstico
load leka.txt -ascii;

data=leka;        % Nastavimo spremenljivko data na prebrano casovno vrsto

N=length(data);   % Izracunamo dolzino casovne vrste

min=30;           % Najmanjsa dolzina casovne serije za katero se racunamo
                    koeficient avtokorelacije

% Izracunamo koeficiente avtokorelacije za casovne odmike od 1 do N-min
for i = 2:(N-min)
    kf = corrcoef( data(i:N), data(1:(N-i+1)) );
    Yac(i) = kf(1, 2);
end;

% Izris korelograma
bar(Yac);
xlabel('Casovni zamik');
ylabel('Koeficient avtokorelacije');
axis tight;
```

2. Program v programskem jeziku orodja MATLAB za izračun in odstranitev trenda ter normalizacijo in centralizacijo časovne vrste

```
% PRIPRAVI – Program za pripravo casovne vrste za delo z nevronskimi
%            mrezami
%
% Borut Lukic, 30. 5. 2002
%
%**********************************************************************

% Nalozimo podatke iz tekstovne datoteke. Podatki so v vrsticni obliki
% loceni z znakom za novo vrstico
load leka.txt -ascii;

data=leka;              % Nastavimo spremenljivko data na prebrano casovno
                          vrsto

N=length(data);         % Izracunamo dolzino casovne vrste

X = (1:N);              % v spremenljivko X shranimo zaporedne stevilke
                          casovnih indeksov (1, 2, 3, 4, ... , N)

sumY = sum(data);       % izracunamo vsoto vseh podatkov v casovni vrsti

Yavg = sumY/N;          % Izracunamo povprecno vrednost podatka

sumX = sum(X);          % Vsota casovnih indeksov
sumX2 = sum(X.^2);      % Vsota kvadriranih casovnih indeksov

sumYX = sum(X.*data);   % Vsota produktov med indeksi in vrednostmi

% regresijske koeficiente
B = sumYX/sumX2;        % Smerni koeficient trenda
A = Yavg - (B*sumX/N);  % Vrednost trenda v izhodiscu casovne vrste

Ytrend = (A + B.*X);    % casovna serija trendnih vrednosti

% Graficni prikaz casovne serije in trenda
plot(X,data,'-',X,Ytrend,'--');
xlabel('Casovni indeks');
ylabel('Nominalna vrednost');

% Pocakamo na pritisk tipke
pause
```

```
% Nato odstranimo iz casovne serije trend
Ynt = diff([Ytrend;data]);

% Graficni prikaz casovne serije brez trenda
plot(Ynt);
xlabel('Casovni indeks');
ylabel('Nominalna vrednost');

% Pocakamo na pritisk tipke
pause

Ymax = max(abs(Ynt));   % Poiscemo najvecjo vrednost v seriji

Yn = Ynt/Ymax;          % Normaliziramo casovno vrsto

Ymean = sum(Yn)/N;      % Poiscemo povprecno vrednost normalizirane vrste

Ync = diff([Ymean;Yn]); % Iz vrste odstranimo povprecno vrednost in s tem
                          vrsto centraliziramo

plot(Ync);
xlabel('Casovni indeks');
ylabel('Normalizirana in centralizirana vrednost');
```

3. Program v programskem jeziku orodja MATLAB za učenje nevronske mreže

```
% NETTRAIN – Program za ucenje nevronske mreze
%
%
% Borut Lukic, 30. 5. 2002
%
%*********************************************************************

% Nalozimo podatke iz tekstovne datoteke. Podatki so v vrsticni obliki
% loceni z znakom za novo vrstico. Podatki so ze stacionarni, torej
% smo iz njih predhodno odstranili trend in jih normalizirali
load leka_p.txt –ascii;

data = leka_p;

N=length(data);         % Izracunamo dolzino casovne vrste

X = (1:N);              % v spremenljivko X shranimo zaporedne stevilke
                          casovnih indeksov (1, 2, 3, 4, ... , N)

Npl = 120;  % stevilo dni za daljso napoved
Nps = 10;   % stevilo dni za krajso napoved

% Zgradimo nevronsko mrezo :
%  vecplastni perceptron: newff
%  elmanova povratna mreza: newelm
%  radialno zasnovana mreza: newrb

Nin = 5;    % stevilo vhodov v mrezo

% vhodi v mrezo morajo biti ustrezni z spremenljivko Nin
% primer je za: vecplastni perceptron
net = newff([-1 1;-1 1;-1 1;-1 1;-1 1], [5 6 1],
             {'tansig','tansig','purelin'});

% Mreza ima 5 vhodov, ki zavzemajo vrednosti med -1 in 1
% V vhodni plasti ima 5 nevronov z tansig transformacijsko funkcijo
% V skriti plasti ima 6 nevronov z tansig transformacijsko funkcijo
% V izhodni plasti ima 1 nevron z purelin transformacijsko funkcijo

net.trainParam.epochs = 10000;      % stevilo korakov ucenja
net.trainParam.show = 10;           % izpis napake na vsake 10 korakov
net.trainParam.goal = 0.01;         % cilj ucenja
net.performFcn = 'sse';             % sum squared error, funkcija kvadratne
                                    % napake za preverjanje ucinkovitosti
```

```
% Skreiramo ucno in testno mnozico

Nu = N-(Npl+Nin+1);     % dolocimo velikost ucne mnozice

% Ucna mnozica vhodov [Pu] je dimenzij Nin x Nu
Pu = zeros(Nin,Nu);
for i = 1:Nin
 Pu(i, i:Nu) = data(1,1:(Nu-i+1));
end

% Ucna mnozica izhodov [Tu] je le vektor izhodnih vrednosti dolzine Nu
Tu = data(2:Nu+1);

% Treniramo mrezo
net = train(net, Pu, Tu);
```

4. Program v programskem jeziku orodja MATLAB za napovedovanje z nevronsko mrežo

```
% PREDICT – Program za napovedovanje z nevronsko mrezo
%
%
% Borut Lukic, 30. 5. 2002
%
%********************************************************************

% Nalozimo podatke iz tekstovne datoteke. Podatki so v vrsticni obliki
% loceni z znakom za novo vrstico. Podatki so ze stacionarni, torej
% smo iz njih predhodno odstranili trend in jih normalizirali
load leka_p.txt –ascii;

% Nalozimo mrezo, ki smo jo strenirali z uporabo programa za ucenje mreze
% Mreza je spravljena pod simbolom <net>
load trained_net;

% Skupaj z mrezo so v datoteki trained_net spravljene tudi spremenljivke
%   Nu – Dolzina ucne mnozice
%   Nin – Stevilo vhodnih parametrov

Np = 20;                % Dolocimo stevilo korakov napovedovanja
N = length(leka_p);     % ugotovimo kolicino razpolozljivih podatkov

% Sestavimo vektor zacetnih vhodnih parametrov, tako da vzamemo <Nin> zadnjih
% vrednosti ucne mnozice
Pp = Pu(2:Nin, Nu);
Pp(Nin) = Tu(Nu);

for i = 1:Np
 % Vhodni vektor posljemo skozi mrezo in dobimo izhodno vrednost.
 y = sim(net, Pp);
 Tp(i) = y;

 % Vhodni vektor nato pomaknemo navzdol, na zacetek pa vrinemo izhodno
 %  vrednost mreze, ki predstavlja najnovejsi podatek
 Pp = Pp(2:Nin);
 Pp(Nin) = y;
end

% Izrisemo graf na katerem prikazemo dejanske in izracunane vrednosti
OffX = Np/2;
A = leka_p(Nu-OffX+1:Nu+Np);
Alen = length(A);
B = [A(1:OffX) Tp];

plot((1:Alen), A, (1:Alen), B)
```