

PRIMJENA VJEŠTAČKIH NEURONSKIH MREŽA U IZGRADNJI SISTEMA ZA PODRŠKU POSLOVNOM ODLUČIVANJU

USAGE OF ARTIFICIAL NEURAL NETWORKS IN DEVELOPMENT OF BUSINESS-RELATED DECISION SUPPORT SYSTEMS

Mr Lazar K. Radovanović, viši asistent
Ekonomski fakultet u Brčkom

Apstrakt. U članku je prikazan pristup izgradnji sistema podrške poslovnom odlučivanju koji se zasniva na primjeni vještačkih neuronskih mreža. Proces izgradnje neuronske mreže prikazan je u šest iterativnih koraka od prikupljanja i pripreme podataka do testiranja i implementacije mreže. U radu su navedena područja primjene sistema neuronskih mreža i najvažniji softverski alati za izgradnju tih sistema.

Ključne riječi: vještačke neuronske mreže, proces učenja neuronske mreže, proces projektovanja neuronske mreže

Abstract. This article presents an approach in construction of business-related decision support system based on application of the artificial neural networks. The process of developing of neural networks is presented through six iterative steps, starting from the gathering and sorting of the information to testing of the network and its implementation. This study also states the areas of application of the system of neural networks and the most important software tools used in construction of these systems.

Key words: artificial neural networks, neural network learning process, neural network development process

UVOD

Vještačke neuronske mreže pripadaju grupi metodologija za rješavanje problema koje nastoje da oponašaju funkcionisanje ljudskog mozga i čiji razvoj i primjena doživljavaju uspon osamdesetih godina prošloga vijeka kada se počinju primjenjivati za rješavanje složenih problema. Ovome su doprinijele ključne činjenice: potreba da se obrada informacija odvija slično funkcionisanju mozga, napredak u razvoju informacione tehnologije i multidisciplinarno proučavanje funkcionisanja bioloških neuronskih mreža, odnosno ljudskog mozga.

Na početku rada upoređuje se način funkcionisanja ljudskog mozga i vještačke neuronske mreže, da bi se, zatim, navela struktura neuronske mreže i objasnila obrada informacija u mreži, mogućnost učenja mreže i izbor odgovarajućeg algoritma učenja.

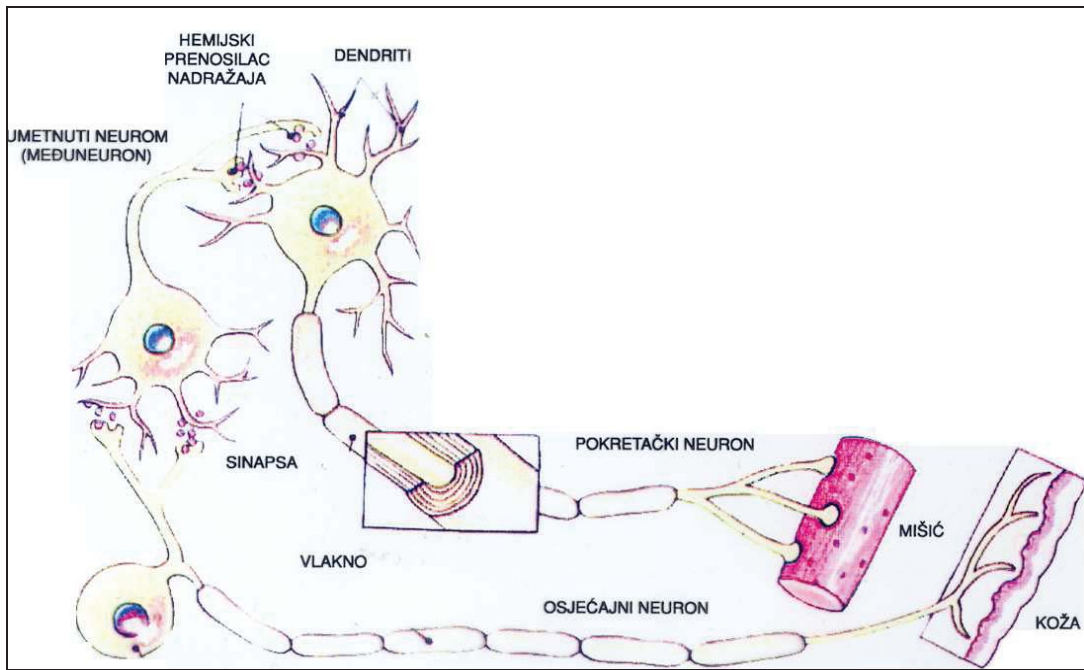
Projektanti sistema zasnovanih na vještačkim neuronskim mrežama koriste se sličnom metodologijom kao i prilikom projektovanja klasičnih informacionih sistema i sistema podrške odlučivanju. Međutim, pojedine faze projektovanja sistema koji se zasnivaju na neuronskim mrežama znatno se razlikuju od faza koje se primjenjuju prilikom projektovanja klasičnih sistema. Pojednostavljen proces projektovanja tih sistema opisan je u radu.

BIOLOŠKE I VJEŠTAČKE NEURONSKE MREŽE

Da bi se objasnilo funkcionisanje vještačke neuronske mreže, potrebno je da se shvati funkcionisanje ljudskog mozga. Procjenjuje se da ljudski mozak ima 50 do 150 milijardi neurona podijeljenih u više od 100 različitih vrsta. Neuroni su podijeljeni u grupe koje se nazivaju mreže. Svaka mreža sadrži nekoliko hiljada međusobno usko povezanih neurona. Prema tome ljudski mozak možemo predstaviti kao kolekciju neuronskih mreža.

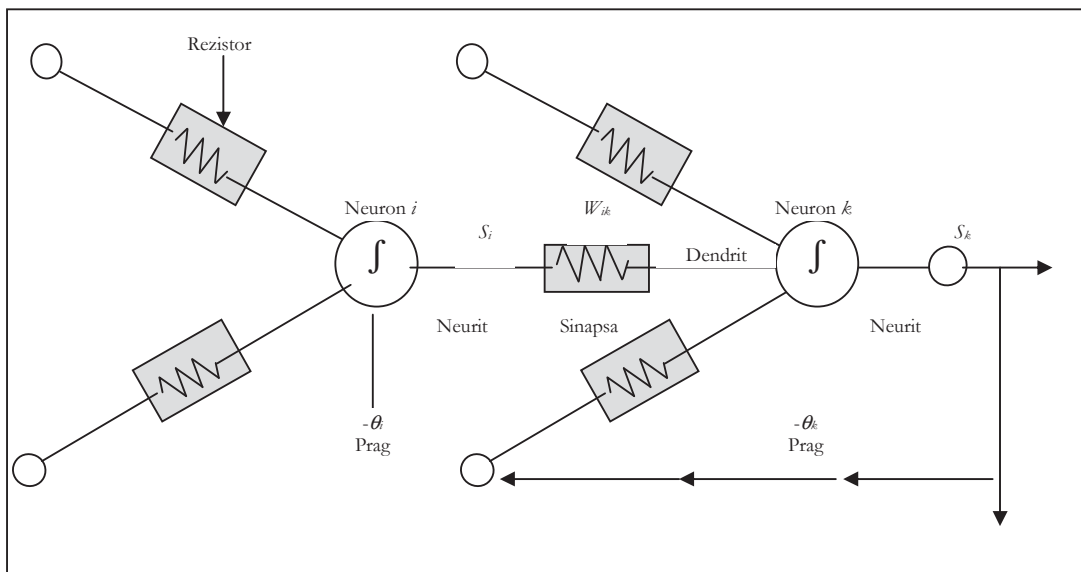
Sposobnost učenja i reagovanja na promjene u okruženju zahtijeva inteligenciju. Mozak i centralni nervni sistem kontrolišu razmišljanje i inteligentno ponašanje. Čovjek koji preživi oštećenje mozga ima otežano učenje i reagovanje na promjene okruženja. Isto tako neoštećeni dijelovi mozga često mogu da kompenzuju oštećene dijelove novim učenjem. Na slici 1. prikazani su biološki neuroni, a na slici 2. vještačka neuronska mreža sa dva neurona.

Slika 1. Biološki neuroni



Izvor: (prilagođeno) Medicinska enciklopedija

Slika 2. Vještačka neuronska mreža sa dva neurona



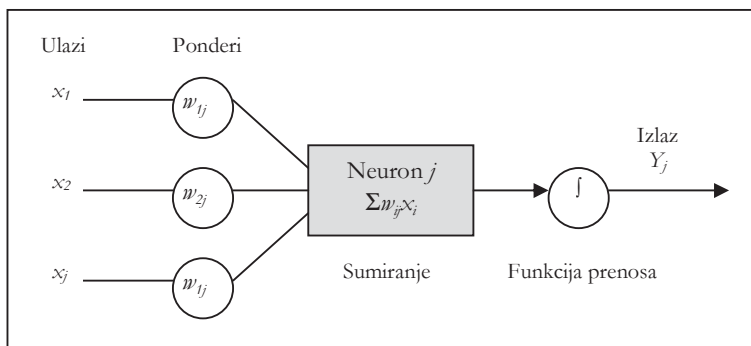
Izvor: Laudon, K. C., Laudon, J. P., 2002, str. 392.

U vještačkoj neuronskoj mreži biološki neuroni postaju elementi obrade, aksoni (neuriti) i dendriti su niti koje prenose signale, a sinapsa postaje primalac varijable koji nosi impulse (signale) koji predstavljaju podatke. Signali mogu da se prenose neizmijenjeni ili se mijenjaju pomoću sinapse. Sinapsa ima mogućnost da poveća ili smanji jačinu veze između neurona i izaziva pobuđivanje ili blokiranje sljedećeg neurona.

Komponente vještačke neuronske mreže

Neuronska mreža sastavljena je od komponenti obrade koji su organizovani na različite načine u obliku mrežne strukture. Osnovna procesna jedinica jeste neuron. Nekoliko neurona organizovano je u mrežu. Postoji više načina organizovanja neurona u vještačkoj neuronskoj mreži što se može shvatiti kao njihova topologija. Popularni pristup, *feedforward-backpropagation*¹ paradigma, odnosno višeslojni perceptron², dopušta svim neuronima da povezuju izlaz u jednom sloju sa ulazom sljedećeg sloja, ali ne dopušta bilo kakvo povratno povezivanje. To je paradigma koja ima najširu primjenu.³ Slika 3. prikazuje da vještački signali mogu da budu promijenjeni težinskim faktorima (ponderima) na način sličan fizičkim promjenama koje se pojavljuju u sinapsama.

Slika 3. Obrada informacije u vještačkom neuronu



Izvor: Turban, E. et. al., 2005, str. 664.

Tabela 1. Odnos biološkog i vještačkog neurona

<i>Biološki neuron</i>	<i>Vještački neuron</i>
Soma (tijelo neurona)	Čvor
Dendriti	Ulaz (input)
Akson (neurit)	Izlaz (output)
Sinapsa	Ponder (težinski faktor)
Mala brzina	Velika brzina
Mnogo neurona (10^9)	Nekoliko neurona (desetak do stotinu hiljada)

Izvor: Medsker, L., Liebowitz, J., 1994, str. 163.

Komponente za obradu vještačke neuronske mreže su vještački neuroni. Svaki od neurona prima inpute, obrađuje ih i šalje jedan izlaz (slika 3). Ulaz (input) može da bude neobrađen ulazni podatak ili izlaz ostalih komponenti obrade. Izlaz (output) može da bude finalni rezultat (npr. 1 što znači DA i 0 što znači NE) ili ulazni element za druge neurone.

Vještačka neuronska mreža sastavljena je od skupa neurona koji su grupisani u slojeve. Karakteristična struktura sa jednim neuronom (a) i nekoliko neurona (b) prikazana je na slici 4. Jednostavna mreža sastoji se od dva sloja neurona (ulazni i izlazni). Između ulaznog i izlaznog sloja može da bude smješteno nekoliko međuslojeva koji se nazivaju prikriveni slojevi. Prilikom obrade informacija većina elemenata obrade vrše svoja izračunavanja u isto vrijeme. Ovakvo paralelno procesiranje slično je načinu na koji funkcioniše mozak i razlikuje se od serijskog procesiranja konvencionalnog izračunavanja uz pomoć računara.⁴

¹ Češće se naziva samo *backpropagation*.

² Perceptron - ranija struktura neuronske mreže nije se koristila skrivenim slojem već jednim neuronom koji se naziva perceptron. Višeslojni perceptron podrazumijeva korišćenje i skrivenim slojem.

³ Haykin, S. S., 1999, str. 21.

⁴ Turban, E. et. al., 2005, str. 665.

Obrada informacija u vještačkoj neuronskoj mreži

Nakon određivanja strukture mreže može se pristupiti obradi informacija u mreži. Osnovni pojmovi vezani sa obradu su:⁵ inputi (ulazne veličine), outputi (izlazne veličine), ponderi ili težinski faktori povezivanja, funkcija sumiranja i transformaciona (transferna ili prenosna) funkcija.

Inputi. Svaki input odgovara jednom obilježju. Na primjer, ako je problem da se donese odluka o odobravanju ili neodobravanju kredita, neki od obilježja mogli bi da budu visina prihoda aplikanta, starost i posjedovanje kuće. Numeričke vrijednosti obilježja koje mogu da budu različite vrste podataka kao što su tekst, slike i govor predstavlja ulaz u mrežu. Za pretvaranje simboličkih podataka u odgovarajuće inpute i za mjerenje podataka prije obrade mogu se vršiti različite pripreme.

Izlazi. Izlazi iz mreže sadrže rješenje nekog problema. U slučaju zahtjeva za kredit, na primjer, izlaz može da bude *da* ili *ne*. Vještačka neuronska mreža pridružuje izlazu odgovarajuću numeričku vrijednost: 1 (*da*) ili 0 (*ne*). Zadatak mreže jeste izračunavanje vrijednosti outputa. Ako izlazna vrijednost nije tačno 1 ili 0 neophodna je naknadna obrada.

Ponderi (težinski faktori povezivanja). Ključni elementi u vještačkoj neuronskoj mreži jesu težinski faktori ili ponderi, pomoću kojih se iskazuje relativna jačina (ili matematička vrijednost) ulaznih podataka ili više veza koje prenose podatke iz sloja u sloj. Ponderi pokazuju relativnu važnost svakog ulaza za element obrade i za krajnji izlaz. Ponderi su odlučujući za memorisanje naučenih struktura informacija koje se odvija njihovim ponavljanjem izravnavanja u toku treninga mreže.

Funkcija sumiranja. Funkcija sumiranja izračunava ponderisane sume svih ulaznih elementa unošenjem svakog elementa obrade. Funkcija sumiranja množi svaku vrijednost ulaza s njegovim ponderom i ukupne vrijednosti sabira u izlaznu sumu Y . Formula za n ulaza u jednom elementu obrade (slika 4a) je:

$$Y = \sum_{i=1}^n X_i W_i$$

Za j -ti neuron od nekoliko neurona obrade u jednom sloju (slika 4b) formula je:

$$Y_j = \sum_{i=1}^n X_i W_i$$

Transformaciona ili transfer funkcija (funkcija prenosa). Funkcija sumiranja izračunava unutrašnju stimulaciju ili nivo promjenljivosti neurona. Zavisno od toga nivoa neuron može ili ne može da proizvede izlaz. Veza između unutrašnjeg nivoa promjenljivosti i izlaza može da bude linearna ili nelinearna. Veza se iskazuje jednom od nekoliko vrsta transformacionih (transfernih) funkcija. Popularna i prikladna sigmoidna funkcija (funkcija logičke promjene) jeste nelinearna transferna funkcija:

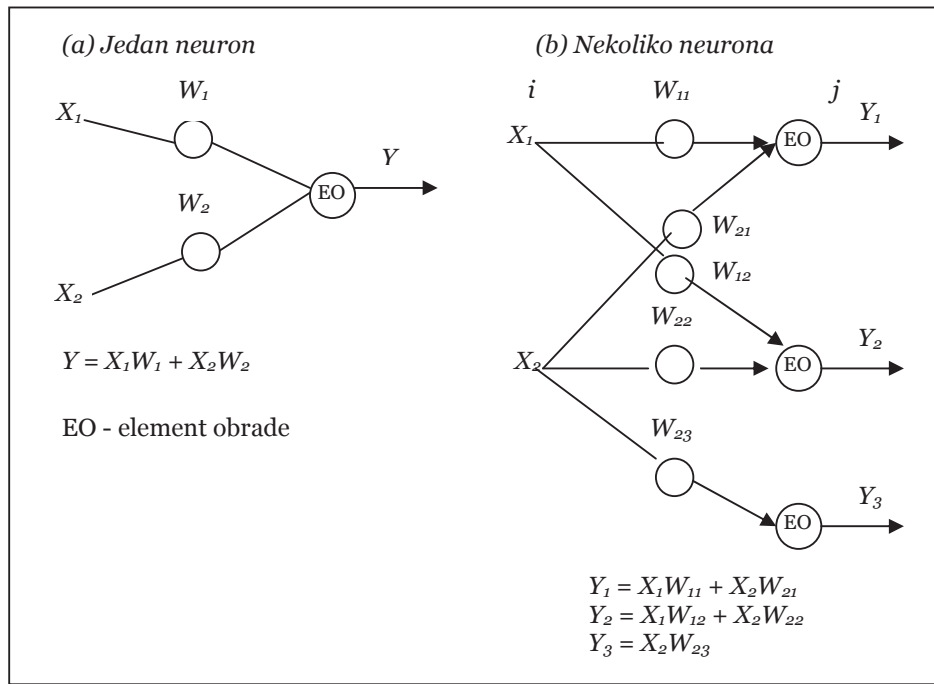
$$Y_T = 1/(1 + e^{-Y})$$

Transformacija modifikuje izlazne nivoe u prihvatljive vrijednosti (obično između 0 i 1), a vrši se prije nego što izlaz dostigne sljedeći nivo. Bez takve transformacije vrijednosti izlaza postajale bi veoma velike, posebno kada postoji nekoliko slojeva neurona. Ponekada, umjesto transformacione funkcije, koristi se *threshold*⁶ vrijednost koja za sve vrijednosti manje ili jednake 0,5 daje vrijednost 0, a za vrijednosti veće od 0,5 daje 1. Transformacija se može događati na izlazu svakog procesnog elementa ili se može izvršavati na krajnjem izlaznom čvoru.

⁵ Isto, str. 665-667.

⁶ Engleska riječ *threshold* ima više značenja: početak, prag, ulaz. U ovom slučaju najviše odgovara riječ prag, međutim ponekada se zadržava i originalna riječ *threshold*.

Slika 4. Funkcija sumiranja za jedan (a) i za nekoliko neurona (b)



Izvor: (prilagođeno) Turban, E. et. al., 2005, str. 667.

Proces učenja u vještačkoj neuronskoj mreži

Jedno od najvažnijih obilježja vještačke neuronske mreže jeste njena mogućnost da uči. Preciznije rečeno, u terminologiji neuronskih mreža, učenje se definiše kao promjena vrijednosti težinskih faktora radi postepenog dobijanja informacija koje kasnije mogu da budu dostupne za pretraživanje. Razvijeno je nekoliko algoritama učenja koji su klasifikovani u sljedeće vrste:⁷

- učenje sa nadgledanjem i učenje bez nadgledanja,
- strukturalno i privremeno učenje,
- indirektno i direktno učenje.

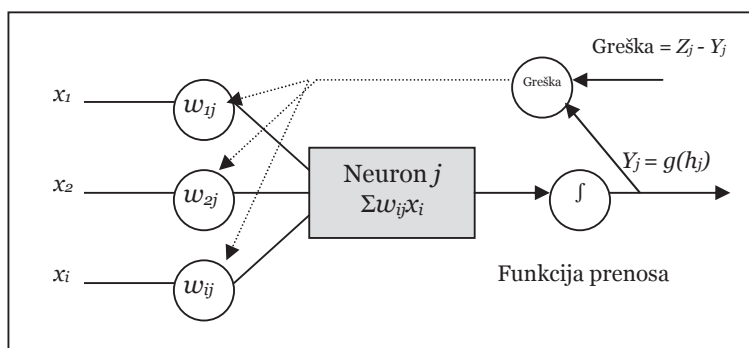
Mreža uči pomoću algoritma sa nadgledanjem ako su za pronalaženje težinskih faktora potrebni i ulazni i željeni izlazni vektori. Ta dva vektora u literaturi su poznati kao parovi obuke (učenja). Obično obuka neuronske mreže zahtijeva korišćenje i testiranje nekoliko takvih parova vektora. Za vrijeme procesa učenja primjenjuje se ulazni vektor, a izračunava izlaz mreže. Izlaz se poredi sa izlaznim vektorom, pa se izračunava razlika, odnosno greška (delta - δ) (slika 5). Zatim se, u skladu s algoritmom, težinski faktori mijenjaju da bi se smanjila greška. Cijeli proces se ponavlja dok izlaz iz mreže ne zadovolji željene specifikacije.⁸ Ovaj proces, poznat pod nazivom *backpropagation* (skraćena za *back error propagation*) je algoritam učenja uz nadgledanje koji se, zbog jednostavne implementacije, u sistemima koji se zasnivaju na vještačkim neuronskim mrežama, najviše koristi.

Strukturalni algoritam učenja daje odgovor iz neuronske mreže koji ne zavisi od prethodnih ulaza i izlaza, za razliku od privremenog algoritma učenja čijom primjenom odgovor mreže zavisi od prethodnih ulaza i izlaza.

⁷ Lin, C. T., Lee, C. S. G., 1996. (Navedeno iz: Matsatsinis, N. F., Siskos, Y., 2003, str. 262-263.)

⁸ Matsatsinis, N. F., Siskos, Y., 2003, str. 262.

Slika 5. *Backpropagation* algoritam



Izvor: (prilagođeno) Turban, E. et. al., 2005, str. 673.

Za poravnavanje težinskih faktora, prilikom učenja algoritmom bez nadgledanja, nije potreban izlazni vektor. U ovom algoritmu slični ulazi (inputi) koji proizvode slične izlaze grupišu se u klase. Algoritmi bez nadgledanja obično se koriste za rješavanje problema u kojima rješenja nisu *a priori* poznata (npr. za probleme predviđanja).⁹ Ovakvi algoritmi spadaju u grupu direktnih algoritama koji se, za razliku od indirektnih algoritama, bave novim situacijama u bilo koje vrijeme. Algoritmi ne daju zadovoljavajuća rješenja za složene probleme kakvi mogu da budu problemi poslovnog odlučivanja koji redovno sadrže rizik i neizvjesnost.

PROJEKTOVANJE SISTEMA KOJI SE ZASNIVAJU NA NEURONSKOJ MREŽI

Metodologija projektovanja sistema zasnovanih na vještačkim neuronskim mrežama slična je metodologiji projektovanja klasičnih informacionih sistema i sistema podrške odlučivanju. Međutim, pojedine faze projektovanja su jedinstvene ili imaju drugačiji pristup u odnosu na projektovanje klasičnih sistema. Na osnovu prikaza na slici 6. može se zaključiti da se proces projektovanja odvija u nekoliko koraka. Turban et al¹⁰ navode devet koraka. Pojednostavljeni proces projektovanja neuronske mreže obuhvata: (1) prikupljanje i pripremu podataka, (2) izbor strukture mreže, (3) izbor algoritma učenja mreže, (4) trening mreže, (5) testiranje i (6) implementaciju.

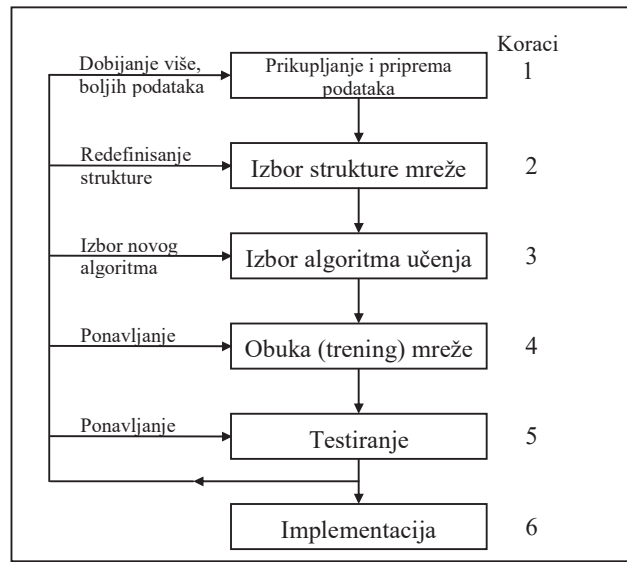
Prikupljanje i priprema podataka. Ovaj korak obuhvata aktivnosti u okviru kojih se prikupljaju i razdvajaju podaci u dva skupa: (1) skup podataka za obuku mreže i (2) skup podataka za testiranje mreže. Slučajevi obuke koriste se za izravnavanje težinskih faktora, a slučajevi za testiranje koriste se za provjeravanje mreže. Podaci koji se koriste za obuku i testiranje moraju da obuhvate sva obilježja koja su pogodna i korisna za rješavanje problema. Sistem može da uči samo onoliko koliko adekvatni podaci to omogućavaju. Prema tome, najvažniji korak za izgradnju kvalitetnog sistema koji se zasniva na vještačkoj neuronskoj mreži jeste prikupljanje i priprema podataka. Mada veći skup podataka produžava trajanje obuke, bolje je koristiti više podataka jer se na taj način poboljšava tačnost obuke i postiže brža konvergencija kvalitetnog skupa težinskih faktora. Za skup podataka srednje veličine obično se za trening na slučajan način bira 80% podataka, a za testiranje 20%, a za mali skup podataka obično se svi podaci koriste i za obuku i za testiranje, a za velike skupove podataka, uzima se uzorak odgovarajuće veličine i sa njim se postupa na sličan način kao za skup srednje veličine.¹¹

⁹ Isto.

¹⁰ Turban, E. et. al., 2005, str. 674-679.

¹¹ Isto, str. 676.

Slika 6. Proces projektovanja neuronske mreže



Izvor: (pojednostavljeno) Turban, E. et. al, 2005, str. 675.

Za objašnjenje ovog koraka može da posluži primjer izgradnje sistema u banci koji se zasniva na neuronskoj mreži. Sistem služi za procjenu finansijskih podataka klijenta da bi se odgovorilo na pitanje da li je klijent ispunio uslove za stečaj. Banka ili stečajni upravnik prvo treba da identifikuje koji finansijski podaci mogu da se koriste kao inputi i na koji način da se pribave takvi podaci. Sljedećih pet obilježja mogu da budu korisni ulazni elementi za mrežu: (1) obrtni kapital/ukupna aktiva, (2) zadržane zarade/ukupna aktiva, (3) zarade prije oporezivanja i ukamaćivanja/ukupna imovina, (4) tržišna vrijednost akcija/ukupne obaveze i (5) prodaja/ukupna prodaja. Izlaz je binarna varijabla stečaj (bankrot) ili ne (1 ili 0).

Izbor strukture mreže. Ovaj korak u projektovanju sistema sa vještačkom neuronskom mrežom obuhvata izbor topologije mreže i određivanje: (1) ulaznih čvorova (neurona), (2) izlaznih čvorova, (3) broj skrivenih slojeva i (4) broj skrivenih čvorova. Osim u posebnim situacijama u poslovnim aplikacijama, pa prema tome i u poslovnom odlučivanju, češće se koristi višeslojna topologija sa spregom unaprijed.

Dizajniranje ulaznih neurona (čvorova) mora da se zasniva na obilježjima skupa podataka. U primjeru za predviđanje stečaja može da se izabere troslojna struktura koja obuhvata jedan ulazni, jedan izlazni i jedan skriveni sloj (slika 7). Ulazni sloj sadrži pet čvorova od kojih svaki predstavlja varijablu, a izlazni sloj sadrži jedan čvor sa 0 za stečaj ili 1 za ozdravljenje (sanaciju). Određivanje broja skrivenih čvorova može da bude sumnjivo. Bilo je preporučeno nekoliko heuristika, međutim nesumnjivo je da nijedna od njih nije najbolja. Karakterističan je pristup da se broj skrivenih čvorova odredi kao prosječan broj ulaznih i izlaznih čvorova.¹² U našem primjeru broj skrivenih čvorova dobija se izračunavanjem prosjeka na osnovu izraza $(5 + 1)/2 = 3$.

Izbor algoritma učenja mreže. Nakon definisanja strukture mreže pronalazi se odgovarajući algoritam učenja kako bi se identifikovao skup težinskih faktora koji će najbolje da obuhvate podatke učenja i imati najbolju tačnost predviđanja. Za izabranu topologiju sa spregom unaprijed, kao u prethodnom koraku, obično se koristi algoritam *backpropagation* (slika 5). Na tržištu postoji više komercijalnih paketa za treniranje mreže tako da korisnik ne mora sam da implementira algoritam učenja. Probne verzije takvih paketa kao što su *NeuralWare*¹³, *BrainMaker – California Scientific BrainMaker neural network software*¹⁴, *Megaputer – Neural network software* i alat za analizu podataka¹⁵, *EasyNN-plus*¹⁶ i drugi paketi za neuronske mreže mogu se naći na Web-u.

¹² Isto, str. 677

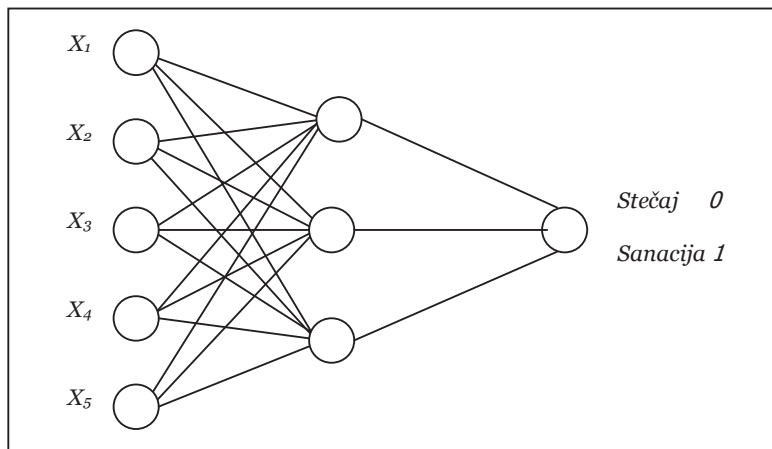
¹³ www.neuralware.com

¹⁴ www.calsci.com

¹⁵ www.megaputer

¹⁶ www.easynn.com

Slika 7. Struktura neuronske mreže za predviđanje stečaja



Trening mreže. Trening (učenje ili obuka) mreže je iterativni proces koji počinje postavljanjem na slučajan način odabranih težinskih faktora i postepenim povećavanjem prilagodavanja modela mreže poznatom skupu podataka. Iteracija se nastavlja sve dok suma greške konvergira naniže prema unaprijed postavljenom prihvatljivom nivou. Za kontrolu brzine postizanja rješenja, prilikom primjene *backpropagation* algoritma, postavljaju se dva parametra: tempo ili stopa učenja (*learning rate*) i momenat (*momentum*).¹⁷ Stopa učenja obično je vrijednost iz intervala [0, 1]. Ukoliko je vrijednost ovog koeficijenta veća mreža uči brže i obrnuto. Ako je vrijednost ovog parametra velika, smanjuje se mogućnost generalizacije sistema što predstavlja značajnu karakteristiku neuronske mreže. Generalizacija omogućava neuronskoj mreži da donosi kvalitetne odluke i iz netipičnih i nekompletnih podataka.¹⁸ Neki softverski paketi u svojim heuristikama učenja imaju sopstvene parametre za povećanje brzine procesa učenja.

*EasyNN-plus*¹⁹ postavlja stopu učenja unaprijed na 0,6 u momentu kreiranja neuronske mreže, ali se ona može promijeniti na bilo koju vrijednost od 0,1 do 10. Vrlo male vrijednosti dovode do sporog učenja, a vrijednosti iznad 1,5 često dovode do pogrešnog učenja i oscilacija. Isto tako, momenat se postavlja unaprijed na 0,8, a može da se promijeni na bilo koju vrijednost između 0 i 0,9. *EasyNN-plus* nudi mogućnost da se tempo učenja i momenat automatski odrede pokretanjem nekoliko ciklusa učenja sa različitim vrijednostima stopa učenja i momenta kao i mogućnost da se automatski smanji stopa učenja i momenat za vrijeme treninga ako se pojave oscilacije ili pogrešno učenje.

U procesu treninga mreže ponekada su potrebna i određena prilagodavanja podataka kao što su: (1) promjena formata podataka da bi se udovoljilo zahtjevima softvera, (2) normalizacija opsega podataka da bi se omogućilo njihovo poređenje i (3) odstranjivanje sumnjivih podataka.²⁰

Nakon prikupljanja i prilagodavanja podataka i njihovog smještanja u tabele odgovarajućeg softvera za rad sa neuronskim mrežama izvršava se proces učenja koji, zavisno od broja čvorova (neurona) i veličine skupa podataka za učenje, da bi postigao rješenje može da se odvija u nekoliko hiljada do nekoliko miliona iteracija. Prema tome, složeni problemi kakvi mogu da budu i problemi poslovnog odlučivanja, naročito kada se radi o donošenju strateških poslovnih odluka, prilikom primjene odgovarajućeg softvera sa neuronskim mrežama zahtijevaće i veliki broj iteracija, pa je prilikom projektovanja takvih sistema veoma bitno da se korisnici odluče za snažne računare sa velikim kapacitetom memorije i velikom brzinom rada.

Testiranje. Već je navedeno da se prilikom prikupljanja i pripreme podataka formiraju dva skupa podataka, jedan za obuku mreže, a drugi za testiranje. Nakon treninga pristupa se testiranju mreže pomoću kojeg se provjeravaju performanse dobijenog modela mreže mjerenjem njegove sposobnosti da korektno klasifikuje podatke za testiranje. Osnovni pristup testiranju mreže jeste metoda crne kutije koja se primjenjuje tako što

17 Turban, E. et. al., 2005, str. 677.

18 Soldić-Aleksić, J., 2001, str. 149.

19 www.easynn.com

20 Turban, E. et. al., 2005, str. 678.

se porede rezultati testa sa rezultatima iz prošlosti da bi se potvrdilo da li ulazni elementi daju odgovarajuće rezultate. Pogrešni izlazi mogu da se koriste za poređenje rezultata dobijenih po poznatim metodama testiranja sistema. Od mreže se ne očekuje da postiže perfektne rezultate (tzv. zero error²¹ teško je, ako ne i nemoguće, postići) već se samo zahtijeva izvjestan stepen tačnosti. Aplikacija neuronske mreže obično je alternativa drugim, najčešće statističkim i kvantitativnim metodama čiji rezultati mogu da se koriste kao mjerilo za poređenje tačnosti. Ainscough, T. L. i Aronson, J. E.²² istraživali su primjenu modela neuronskih mreža u predviđanju prodaje u trgovini na malo na osnovu skupa ulaznih veličina (cijena, promocija itd.). Poredili su rezultate neuronske mreže sa rezultatima višestruke regresije i poboljšali koeficijent korelacije (R^2) sa 0,5 na 0,7.

Implementacija. Ova faza projektovanja vještačke neuronske mreže vrlo često zahtijeva interfejs sa drugim računarskim informacionim sistemima i obuku korisnika. Za unapređivanje sistema i dugoročni uspjeh preporučuje se kontrola i povratna informacija (*feedback*) prema projektantu. U ovoj fazi važno je, takođe, da se prethodno dobije povjerenje korisnika i menadžmenta u projektovanje da bi se potvrdilo da je sistem prihvaćen i da bi se obezbijedila njegova pravilna upotreba.

PRIMJENA VJEŠTAČKIH NEURONSKIH MREŽA

Vještačke neuronske mreže primjenjuju se u više područja, a naročito su prikladne za rješavanje problema čiji su ulazni elementi alfanumeričkog i numeričkog tipa i u kojima veze između ulaznih elemenata i rezultata nisu linearne ili ulazni podaci ne pripadaju normalnoj distribuciji. U takvim slučajevima opšte statističke metode nisu pouzdane. Pošto se za primjenu vještačke neuronske mreže ne postavljaju nikakvi zahtjevi u vezi sa distribucijom podataka, njihove mogućnosti manje su ograničene od tradicionalnih statističkih metoda u slučaju kada podaci nisu normalno distribuirani.²³

Fadlalla i Lin (2001)²⁴ istraživali su primjenu vještačkih neuronskih mreža u finansijama. Lee et al.²⁵, u svojim istraživanjima, 2002. godine, proučavali su pretraživanje i filtriranje sadržaja na Internetu. Davis et al. (2001)²⁶ koristili su se vještačkom neuronskom mrežom pri istraživanju i predviđanju cijena akcija na kanadskom i američkom finansijskom tržištu. Vještačke neuronske mreže mogu da se primjenjuju i u drugim područjima, pa prema tome i u području poslovnog odlučivanja. Tvrdimo da su vještačke neuronske mreže upravo najprikladnije u tom području, jer donošenje poslovnih odluka zahtijeva mnogo ulaznih parametara, a izlazni rezultat je jedinstven: donošenje ili nedonošenje poslovne odluke.

Na tržištu postoje brojni alati za projektovanje vještačkih neuronskih mreža čije funkcije, u nekim slučajevima, liče na kosture ekspertnih sistema. Alati su snabdjeveni skupom standardnih arhitektura, algoritmima učenja i parametrima neophodnim za manipulisanje podacima. Postoje projektni alati koji mogu da podrže više desetina paradigmi mreže i algoritama učenja. Pored standardnih proizvoda postoje i specijalizovani proizvodi od kojih su neki projektovani u Javi i mogu se pokrenuti direktno sa *Web*-a uz pomoć *Web* pretraživača. Drugi proizvodi, tzv. hibridni proizvodi, projektovani su da bi se povezali s ekspertnim sistemima.²⁷

Korišćenje alata vještačke neuronske mreže ograničeno je njenom konfiguracijom. Međutim, projektanti neuronskih mreža često preferiraju upotrebu programskih jezika opšte namjene kao što je C++ ili radne tabele da bi programirali model i izvršavali izračunavanja u neuronskoj mreži. Osim toga mogu se koristiti i biblioteke rutina vještačkih neuronskih mreža, a za implementaciju samostalnih ili ugrađenih različitih vrsta vještačkih neuronskih mreža *havSoftware*²⁸ obezbjeđuje biblioteku C++ klasa. Nekoliko popularnijih paketa za projektovanje vještačkih neuronskih mreža navedeno je prilikom objašnjavanja izbora algoritama učenja.

21 Zero error ili zero defect - funkcionisanje sistema, pojedinca ili organizacije bez greške.

22 Ainscough, T. L., Aronson, J. E., 1999. (Navedeno iz: Turban, E. et al., 2005, str. 679.)

23 Turban, E. et al., 2005, str. 679.

24 Fadlalla, A., Lin, C., 2001, str. 122.

25 Lee, P. Y. et al., 2002, str. 48-57.

26 Davis, J. T. et al., 2001, str. 83-96.

27 Turban, E. et al., 2005, str. 678.

28 www.hav.com

Pored ovih postoje i drugi paketi kao što su *NeuroShell Easy* i *SPSS Neural Connection* pomoću kojih mogu da se kreiraju sistemi neuronskih mreža za rješavanje kompleksnih problema poslovnog odlučivanja.

*EasyNN-plus*²⁹ služi za izgradnju neuronskih mreža za marketing, predviđanje prodaje, planiranje, investiranje i dr. *EasyNN-plus* obavlja analizu kompleksnih podataka na osnovu koje se predviđanje, prognoziranje, klasifikacija i projekcija vremenskih serija odvija brzo i jednostavno. Mreža podataka *EasyNN-plusa* kreira se pomoću tekstualnih dokumenata, radnih tabela ili binarnih fajlova. Višeslojne neuronske mreže dobijaju se od mreža podataka sa minimalnom intervencijom korisnika. Za izgradnju neuronskih mreža mogu da se koriste numerici, tekst, slike ili kombinacije različitih tipova podataka. *EasyNN-plus* neuronske mreže mogu da se treniraju, provjeravaju i da im se postavljaju upiti. Dijagrami, grafikoni i input/output podaci koje daje neuronska mreža mogu da se detaljno prikažu i odštampaju. Svi grafikoni, mreže podataka i dijagrami neuronskih mreža ažuriraju se dinamički tako da se može vidjeti kako pojedini elementi mreže funkcionišu. *EasyNN-plus* može da se koristi direktno iz više drugih aplikacija ili sa komandne linije pomoću skript (*script*³⁰) jezika ili snimljenog makroa.

ZAKLJUČAK

Primjena vještačkih neuronskih mreža u izgradnji sistema podrške odlučivanju sve više dobija na značaju. Neuronske mreže naročito su prikladne za probleme koji su karakteristični po velikom broju ulaznih, a malom broju izlaznih veličina i u kojima relacije između ulaznih i izlaznih veličina nisu linearne ili ulazni podaci nisu normalno distribuirani, pa se ne mogu primijeniti statističke metode. Takvi su i problemi poslovnog odlučivanja čije je rješenje odluka (pozitivna ili negativana), a ulazni podaci mogu da budu alfanumerički i/ili numerički.

Prilikom projektovanja sistema zasnovanog na neuronskoj mreži projektanti mogu da se koriste raznim alatima čije je korišćenje ograničeno strukturom mreže, da se koriste programskim jezicima opšte namjene ili radnom tabelom da bi programirali model i izvršavali izračunavanja. Pored toga mogu da se koriste bibliotekom rutina vještačkih neuronskih mreža. Na elektronskom tržištu postoji nekoliko popularnih paketa za projektovanje vještačkih neuronskih mreža pomoću kojih mogu da se kreiraju sistemi neuronskih mreža i za rješavanje kompleksnih problema poslovnog odlučivanja.

LITERATURA:

1. Ainscough, T. L. and Aronson, J. E. 1999. "A Neural Networks Approach for the Analysis of Scanner Data". Journal of Retailing and Consumer Services. Vol. 6.
2. Davis, J. T. et al. 2001. "Predicting Direction Shifts on Canadian-US Exchange Rates with Artificial Neural Networks". International Journal of Intelligent Systems in Accounting, Finance and Management. Vol. 10.
3. Fadlalla, A. and Lin, C. 2001. "An Analysis of the Applications of Neural Networks in Finance". Interface. Vol. 31. No. 4.
4. Haykin, S. S. 1999. Neural Networks: A Comprehensive Foundation, 2nd Ed. Upper Saddle River, NJ: Prentice Hall.
5. Laudon, K. C. and Laudon, J. P. 2002. Management Information Systems. New Jersey: Prentice-Hall, Upper Saddle River.
6. Lee, P. Y. et al. 2002. "Neural Networks for Web Content Filtering". IEEE Intelligent Systems.
7. Lin, C. T. and Lee, C. S. G. 1996. Neural Fuzzy Systems. Prentice Hall.
8. Matsatsinis, N. F. and Siskos, Y. 2003. Intelligent Support Systems for Marketing Decisions. Dordrecht: Kluwer Academic Publishers Group.
9. Medsker, L. and Liebowitz, J. 1994. Design and Development of Expert Systems and Neural Networks. New York: Macmillan.
10. Soldić-Aleksić, J. 2001. Inteligentni sistemi za poslovno odlučivanje. Beograd: Ekonomski fakultet.
11. Turban, E. et. al. 2005. Decision Support Systems and Intelligent Systems. New Jersey: Pearson Prentice Hall.

Web izvori: www.calsci.com www.hav.com www.neuralware.com
 www.easynn.com www.megaputer www.neurosolutions.com

²⁹ www.easynn.com/easynnplus.html

³⁰ script - struktura u obliku frejma (okvira) koja predstavlja stereotipnu sekvencu događaja (Turban, E. et. al., 2005, Glossary, str. 862.)