NEVRONSKE MREŽE

INFORMATICA 2/89

Keywords: neutral networks, brain, artificial intelligence, survey

Igor Kononenko
Fakulteta za elektrotehniko
in računalništvo, Ljubljana

V zadnjem času se vse več raziskovalcev ukvarja z načrtovanjem in uporabo nevronskih mrež. Nevronska mreža je sestavljena iz velikega števila preprostih elementov 'nevronov', ki delujejo paralelno in asinhrono in komunicirajo preko vezi, ki jim pravimo sinapse. Vsaka sinapsa ima pridruženo realno vrednost (utež), kar predstavlja porazdeljen pomnilnik nevronske mreže. Vsak nevron generira svoj izhod iz obteženega vhoda, ki je lokalno dosegljiv. Lepe lastnosti nevronskih mrež so (za vsako lastnost lahko najdemo izjemo in kolikor je avtorju znano ni nobene nevronske mreže, ki bi imela vse te lastnosti): podobnost biološkim sistemom, visok paralelizem in asinhrono izvajanje, večsmerno izvajanje, prilagodljivost v realnem času, robustnost glede na okvare in glede na manjkajoče podatke, sposobnost učenja in avtomatska generalizacija, mreža ne potrebuje programske opreme in eksplicitne konfiguracije, ni razlike med podatki in adreso, področje ima dobro matematično podlago. Glavna pomanjkljivost nevronskih mrež je nezmožnost razložiti svojo odločitev.

NEURAL NETWORKS

In resent years a lot of researches pay attention to the design and application of neural networks. A neural network is constructed from a large number of simple elements, 'neurons', which operate independently and communicate to each other via connections called 'synapses'. A certain weight is associated to each synapse representing the network's distributed memory. Each neuron combines its weighted inputs into its output. All information that a neuron needs for operation is locally available. A good introduction into the field of neural networks are (Rumelhart & McClelland 86), (McClelland et al. 86) and (Kohonen 84). The following are attractive features of neural networks (for each feature there can be found exceptions and as far as the author knows there is no neural network with all these features): biological similarity, high scale parallelism with asynchronous update, real time adaptiveness, multidirectionality, roboustness with respect to damage of a part of a network and to missing data, learning capabilities and automatic generalization, network needs no software, no algorithms, no explicit configuration, there is no distinction between an address and data, good theoretical background. The main weakness of neural networks is the lack of the ability to explain their decisions.

Intelligence emerges from the interaction of large numbers of simple processing units.

(Rumelhart & McClelland 1986, Predgovor)

1. UVOD

Oglejmo si zgled matričnega računa, ki ga je zmožna opraviti nevronska mreža na sliki 1.

Vsak nevron ima dve različni stanji. Za naš primer je stanje vsakega nevrona lahko 1 ali -1. Naloga nevronske mreže je naučiti se razpoznavati naslednja dva vzorca:

$$X1 = (1,1,1)^T \quad \text{in} \quad X2 = (1,-1,-1)^T$$

Oglejmo si najprej, kako to zmore matrični račun. Sestavimo matriko kot vsoto matrik, ki jih dobimo kot produkt vektorja s transponiranim vektorjem samim:

$$M = X1*X1^T + X2*X2^T = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 2 & 2 \end{bmatrix}$$

Definirajmo še odločitveno funkcijo (pragovni element):

$$f(X) = \begin{cases} 1, & X > 0 \\ 0, & X = 0 \\ -1, & X < 0 \end{cases}$$

Če posplošimo odločitveno funkcijo tudi na vektorje, lahko zapišemo naslednje:

$$f(M*X1) = f((2,4,4)^T) = (1,1,1)^T = X1$$

$$f(M*X2) = f((2,-4,-4)^T) = (1,-1,-1)^T = X2$$

Vidimo, da produkt matrike z danim vektorjem obdrži približno smer danega vektorja, ki jo z odločitveno funkcijo še popravimo in dobimo isti vektor.

Poskusimo uporabiti isti postopek, ko prvotne vektorje poznamo le delno. Z 0 označimo neznano vrednost komponente:

$$X1' = (1,1,0)^T \quad \text{in} \quad X2' = (1,0,-1)^T$$

$$f(M*X1') = f((2,2,2)^T) = (1,1,1)^T = X1$$

$$f(M*X2') = f((2,-2,-2)^T) = (1,-1,-1)^T = X2$$

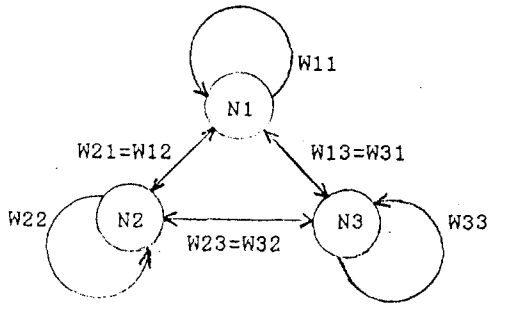
Produkt matrike z delno poznanim vektorjem nadomesti manjkajočo vrednost. Poskusimo sedaj še z napačnimi vrednostmi:

$$X1'' = (1,1,-1)^T = X2''$$

$$f(M*X1'') = f((2,0,0)^T) = (1,0,0)^T$$

V tem primeru je odgovor neodločen. Dejansko je v našem primeru nemogoče ugotoviti, ali je napačna druga ali tretja komponenta.

Dosedanje razmišljanje ne velja vedno oziroma velja pod določenimi pogoji (npr. zahteva se ortogonalnost vektorjev), vendar to presega okvir tega članka. Zanimivo je, da ima vsak shranjeni vzorec shranjen tudi komplement (npr. če shranimo vektor (1,-1,-1), se avtomatsko shrani tudi vektor (-1,1,1)).



Slika 1: Primer nevronske mreže

Oglejmo si, kako lahko gornji račun realiziramo z nevronsko mrežo. Na sliki 1 je prikazana nevronska mreža, sestavljena iz treh nevronov, tako da je vsak povezan z vsakim. Vezi med nevroni ustrezajo elementom matrike. Vezi so dvosmerne, zato je matrika simetrična. Diagonalni elementi matrike ustrezajo vezem nevronov samih s seboj. Vsaka vez ima pridruženo realno vrednost (pozitivno ali negativno), ki ustreza povezavi med nevronoma, ki ju vez povezuje. Tem vrednostim pravimo tudi uteži.

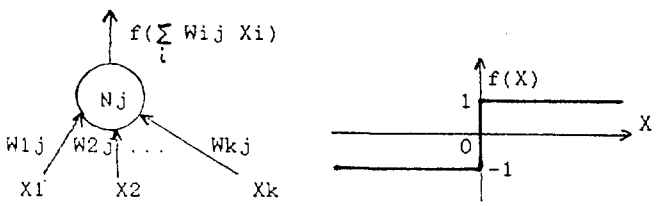
Pred učenjem so vse uteži enake 0. Učenje poteka tako, da se vsakemu nevronu vsili vrednost (stanje) ustrezne komponente iz vhodnega vektorja. Če imata nevrona, ki ju vez povezuje, enake vrednosti, se utež vezi poveča za 1, sicer se zmanjša za 1 (to je posplošeno Hebbovo pravilo učenja, opisano v razdelku 4.3.1). Prej opisan postopek ponovimo za vsak

vhodni vzorec. Ko je mreža naučena, se uteži več ne spreminjajo.

Preverjanje novega primera poteka tako, da se vrednosti ustreznih komponent (delnega) vzorca vsilijo nevronom in vsak nevron paralelno izračuna svoje stanje, ki je hkrati tudi njegov novi izhod, po pravilu:

$$Yj = f(\sum_i Wij\ Xi)$$

kjer je Xj stanje j-tega nevrona, Wij utež sinapse med i-tim in j-tim nevronom in Yi novo stanje i-tega nevrona. Vsak nevron je procesni element, ki zna izračunati obteženo vsoto vhodov in z odločitveno funkcijo preslikati dobljeno vsoto v eno od dveh izhodnih vrednosti (slika 2). Obteženi vsoti pravimo funkcija aktivacije in odločitveni funkciji f pravimo izhodna funkcija. V razdelku 4.4 bomo videli, da lahko izhodna funkcija vrne tudi več kot 2 različni izhodni vrednosti.



Slika 2: Funkcija, ki jo izračunava posamezni nevron

V našem primeru so vsi nevroni paralelno in sinhrono naenkrat izračunali svoj izhod. Ponovitev iste operacije nebi več spremenila nobenega stanja nevronov. Pravimo, da je izvajanje nevronske mreže prišlo v fiksno točko (ekvilibrium). V splošnem nevronska mreža iteracije ponavlja toliko časa, dokler se stanja nevronov ne ustalijo. Če se to zgodi v končnem času, pravimo da mreža konvergira. Izvajanje nevronske mreže je lahko tudi asinhrono, tako da samo en nevron spremeni svoje stanje naenkrat.

Če je utež na sinapsi pozitivna, pravimo, da je sinapsa vzbujevalna. Če je utež negativna, je sinapsa zaviralna. Manjkajoči sinapsi ustreza utež 0. Ponavadi so uteži na sinapsah predstavljene z matriko.

Dokazano je, da če je matrika simetrična, potem asinhroni model vedno konvergira (Kosko 1988), sinhroni pa ne vedno (McEliece in sod. 1987). Za asinhroni model se zahteva, da samo en nevron naenkrat spremeni svoje stanje. Obstajajo tudi modeli, ki uporabljajo asimetrične matrike, za katere je empirično pokazana konvergenca (Wong 1988).

Če uporabljamo nevronsko mrežo kot asociativni pomnilnik, potem ponavadi mreža konvergira k fiksni točki, ki ustreza najbližjemu (najbolj podobnemu) shranjenemu vzorcu danega vhodnega vzorca. Fiksna točka pa ni nujno eden od shranjenih vektorjev in ni nujno shranjeni vektor, ki je najbolj podoben vhodnemu vektorju (McEliece in sod. 1987). Fiksne točke so linearno neodvisni vektorji, katerih podmnožica so tudi lastni vektorji matrike.

2. PORAZDELJEN POMNILNIK

Osnovni princip nevronske mreže je porazdeljen (distribuiran) pomnilnik. Pomnilniške celice so v povezavah med nevroni. Vsak nevron ima dostop do pomnilnika na povezavah z drugimi nevroni (ni nujno, da je vsak nevron povezan z vsakim). To lahko interpretiramo kot veliko število omejitev, ki se hkrati upoštevajo pri generiranju obnašanja. Porazdeljen pomnilnik ne vrne vedno eksaktnega podatka, ker je podatek shranjen v množici povezav z drugimi podatki.

Lastnosti porazdeljene reprezentacije so:
- avtomatska generalizacija: vsaka pomnilniška celica predstavlja povezanost med aktivnostima dveh nevronov (povprečno glede na vse situacije, v katerih se je mreža do sedaj nahajala oziroma glede na vse do sedaj prikazane učne primere). To znanje se lahko uporablja za aproksimacijo manjkajočih vrednosti ali za popravljanje napačnih vrednosti. Tako znanje je t.i. površinsko znanje. Mreža ne pozna globokega znanja, ki vsebuje znane relacije in zakone, ki veljajo v dani problemski domeni.
- prilagodljivost na spreminjajoče se okolje: s spreminjanjem okolja se mreža inkrementalno uči in s tem prilagaja na okolje; pozabljanje že naučenega lahko kontroliramo z določenimi parametri
- čim večja je distribuiranost, tem bolj natančno lahko shranimo iste podatke z enakim številom nevronov (Hinton in sod. 1986).

Distribuiranost povečamo z drugačnim kodiranjem, npr. namesto da en nevron predstavlja eno komponento lahko več nevronov zakodira vrednosti več komponent vhodnih vektorjev. Na ta način je ena komponenta shranjena v povezavah z več ostalimi komponentami, kar omogoča bolj natančno rekonstrukcijo komponente.
- konstruktivni karakter: vsak procesni element (nevron) predstavlja mikro-atribut, ki hkrati kombinira več osnovnih atributov; vsak mikro-atribut opisuje dano domeno iz svojega zornega kota, ki pa ga je težko interpretirati
- ker je vsak podatek v nevronski mreži predstavljen z mnogo elementi in vsak nevron vključen v reprezentacijo več podatkov, odstranitev enega nevrona povzroči delni padec v natančnosti za več podatkov in ne popolno izgubo enega podatka (Hinton in sod. 1986)
- spontano spominjanje pozabljenih podatkov brez ponovnega učenja istih podatkov med ponovnim učenjem drugih podatkov (Hinton in sod. 1986, Hinton & Sejnowski 1986).

Ko je mreža naučena (uteži na vezeh fiksirane), deluje tako, da nevroni dobijo začetne vrednosti (vhodni vzorec) in zatem neodvisno drug od drugega spreminjajo svoja stanja glede na vhodne obtežene signale. Delovanje je paralelno in asinhrono. Delovanje mreže se ustavi, ko ni več nobene spremembe v stanjih nevronov. Pravimo, da je nevronska mreža prišla v stabilno (fiksno) točko.

Za razliko od lokalne reprezentacije v računalnikih, kjer podatek lokalno adresiramo, pri porazdeljeni reprezentaciji v nevronski mreži podatek adresiramo s podatkom samim (to velja za auto-asociativni pomnilnik, glej razdelek 4.2.1). Če je adresa točna, dobimo identičen podatek, sicer dobimo podatek, ki je po določenih kriterijih podoben adresi oziroma vhodnemu podatku. Temu pravimo tudi vsebinsko naslavljanje. V nevronski mreži torej ni razlike med podatkom in njegovo adreso.

Za reprezentacijo popolnoma različnih podatkov ali podatkov na različnih nivojih abstrakcije zahteva porazdeljena reprezentacija različne module (podatek je lokalno shranjen z globalnega gledišča, vendar globalno shranjen z lokalnega gledišča).

3. LASTNOSTI NEVROŃSKIH MREŽ

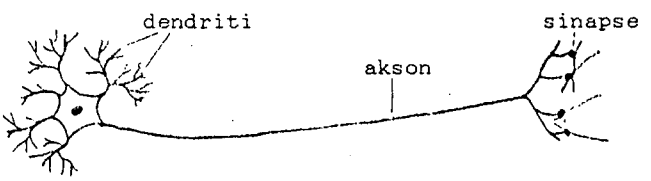
Lepe lastnosti nevronskih mrež so biološka podobnost, visok paralelizem in asinhrono izvajanje, večsmerno izvajanje, prilagodljivost v realnem času, robustnost glede na okvare in glede na manjkajoče podatke, sposobnost učenja in avtomatska generalizacija, mreža ne potrebuje programske opreme in eksplicitne konfiguracije, ni razlike med podatki in adreso, področje ima dobro matematično podlago. Glavna pomankljivost nevronskih mrež je nezmožnost razložiti svojo odločitev. Za vsako lastnost lahko najdemo izjemo, in kolikor je avtorju znano, ne obstaja nevronska mreža, ki bi imela vse naštete lastnosti.

3.1 Biološka podobnost

Pristop do reševanja problemov z nevronskimi mrežami skuša oponašati delovanje človeških možganov in hkrati doseči večjo učinkovitost pri reševanju kompleksnih problemov. Nevronske mreže so abstrakcija in poenostavitev možganskih celic. Nevroni v različnih delih možganov se med seboj močno razlikujejo po obliki, vendar je njihova osnovna struktura enaka.

Primer možganskega nevrona je shematično prikazan na sliki 3. Akson prenaša dražljaje iz celičnega telesa, dendriti pa vodijo vhodne dražljaje v celico. Akson se na koncu razveji in se preko povezav, imenovanih sinapse, povezuje z dendriti ostalih nevronov (Russell 1979).

Aproksimacija z modeli nevronskih mrež kljub poenostavitvam zadostuje za učinkovito modeliranje in analizo. Te raziskave vodijo tudi do boljšega razumevanja procesov v možganih. Več o tem je napisano v razdelku 6.



Slika 3: Shematični prikaz možganskega nevrona

3.2 Paralelizem

Visok paralelizem v nevronskih mrežah je posledica dejstva, da vsak nevron deluje neodvisno od ostalih. To omogoča izredno hitro izvajanje v celoti. Zato so prav nevronske mreže zmožne prilagajanja v realnem času. Večina današnjih aplikacij je simuliranih s klasičnimi računalniki. Deluje pa že nekaj prototipov bolj ali manj specializirane strojne opreme. Implementacijo olajšuje dejstvo, da lahko nevroni delujejo asinhrono.

Primer manj specializirane strojne opreme, ki je primerna za implementacijo nevronskih mrež je t.i. povezovalni računalnik (connection machine). Blelloch in Rosenberg (1987) sta primerjala hitrosti izvajanja posplošenega delta pravila (glej razdelek 4.3.2) na različnih računalnikih vključno s povezovalnim računalnikom, ki je sestavljen iz 65536 enobitnih procesorjev. Zaradi paralelnega izvajanja je bilo izvajanje na povezovalnem računalniku dvakrat hitrejše kot na računalniku Cray-2, ki je nevronsko mrežo simuliral zaporedno.

Graf je s sodelavci (1988) razvil čip za implementacijo algoritmov nevronskih mrež. Uporablja se lahko kot asociativni pomnilnik ali za klasifikacijo vzorcev. Čas izvajanja v čipu (zaporedje iteracij do fiksne točke) je zanemarljiv (cca 1 urin cikel) glede na potreben čas za vhod/izhod (25-50 urinih ciklov).

3.3 Večsmerno izvajanje

Pri nevronski mreži, kjer je vsak nevron povezan z vsakim, je vsak nevron hkrati vhoden in izhoden. Nevronska mreža v tem primeru ne dela razlike med vhodom in izhodom. Poljubna podmnožica nevronov je lahko vhodna. Delovanje mreže bo pri danem vhodu aproksimiralo neznane vrednosti, ki bodo v tem primeru izhod. Primer take mreže je na sliki 1.

3.4 Robustnost

Nevronska mreža je robustna glede na okvaro posameznih nevronov in sinaps. Robustnost je posledica porazdeljene reprezentacije. Ker je vsak nevron vključen v reprezentacijo več podatkovnih elementov in ker je vsak podatkovni element predstavljen z množico nevronov in sinaps, z odstranitvijo enega nevrona ali ene sinapse ne izgubimo nobenega podatka v celoti. Taka okvara povzroči manjšo napako pri večjem številu shranjenih podatkov. Natančnost nevronske mreže pada sorazmerno s številom uničenih nevronov (kar ustreza delovanju možganov - glej razdelek 6).

Nevronska mreža je robustna tudi glede na pomankljive vhodne podatke. To izvira iz načina delovanja nevronske mreže. V mreži, kjer je vsak nevron povezan z vsakim, se bodo manjkajoči vhodni podatki aproksimirali že v prvi iteraciji iz razpoložljivih vhodnih podatkov in naučenega znanja. Primer take aproksimacije je podan v razdelku 1. Seveda bo aproksimacija tem slabša, čim pomankljivejši so vhodni podatki.

3.5 Učenje

Učenje v nevronskih mrežah poteka s spontanim spreminjanjem uteži na sinapsah. Ena pomnilniška celica (sinapsa) predstavlja povezanost med aktivnostima dveh nevronov, ki ju sinapsa povezuje. Ker se utež spreminja glede na spreminjanje vhodnih podatkov (okolja, učnih primerov), utež predstavlja povezanost glede na vse situacije, v katerih se je mreža do sedaj nahajala oziroma glede na vse do sedaj prezentirane učne primere. Ko je mreža naučena, je vsak nevron sposoben napovedati svoje stanje v odvisnosti od danih stanj z njim povezanih nevronov.

3.6 Relacija med strojno in programsko opremo

V nevronski mreži ni programske opreme v klasičnem pomenu te besede. Edini algoritem je algoritem, po katerem deluje individualni nevron. Vse ostalo delovanje je spontano. Mreža teži k stabilni točki tako, da vsi nevroni paralelno in asinhrono delujejo. Besedi računalnik ali izpeljevalnik (inference machine) nista več primerni za opis delovanja takega stroja. Bolj primerno ime bi bilo npr. relaksator.

Strojna oprema je v nevronski mreži lahko fiksna, lahko pa se tudi spreminja (sinapse lahko propadajo ali se pojavijo nove). Za

določeno izvajanje konfiguracija ni enolično določena. Za določene statistične lastnosti ni nujno, da je vsak nevron povezan z vsakim. Zadostuje, da je število povezav med nevroni mnogo večje od števila nevronov (Kohonen 1984). Konfiguracija sama pa je lahko naključna, kar je analogno z možgani (glej razdelek 6).

3.7 Matematična podlaga

Področje ima dobre teoretične temelje v linearni algebri (Jordan 1986, Kohonen 1984). Pojmi, kot so linearna transformacija, inverzna transformacija, linearnost, ortogonalnost vektorjev, lastna vrednost, lastni vektor, fiksna točka in konvergenca, so formalno definirani in dobro obdelani. Vse več je teoretičnih prispevkov, vezanih na nevronske mreže (Guez in sod. 1988, Kosko 1988, McEliece in sod. 1987, Smolensky 1986, Williams 1986, Wong 1988).

3.8 Razlaga odločitve

Največja slabost nevronskih mrež je težavnost razložiti svojo odločitev. Z gledišča raziskovalcev umetne inteligence sistem, ki svoje odločitve ne more razložiti, ni boljši od katerega koli statističnega sistema za razpoznavanje vzorcev, ki lahko dobro deluje; vendar interpretacija njegovih rezultatov ni možna brez veliko znanja in izkušenj iz matematike in statistike (glej razdelek 7).

4. VRSTE NEVRONSKIH MREŽ

Nevronske mreže delimo po naslednjih kriterijih:
- topologija nevronske mreže
- namen nevronske mreže
- pravilo učenja
- funkcija kombiniranja vhodov nevrona v izhod

4.1 Topologija nevronske mreže

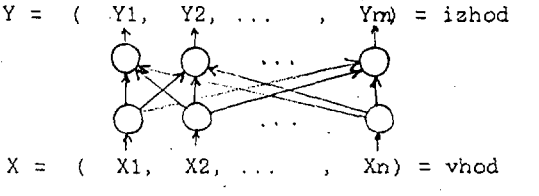
4.1.1 Brez nivojev

Najbolj splošna oblika nevronske mreže je taka, da je vsak nevron hkrati vhoden in izhoden in da je vsak nevron povezan z vsakim nevronom v obeh smereh. Primer je podan na sliki 1.

Nevronska mreža deluje tako, da se na začetku nevronom vsili vrednosti komponent vhodnega vzorca, zatem pa nevroni paralelno sinhrono ali asinhrono spreminjajo svoja stanja in izhod glede na vhod toliko časa, dokler se izhod celotne mreže ne ustali.

4.1.2 Dvonivojske nevronske mreže

Dvonivojska nevronska mreža je sestavljena iz skupine vhodnih in skupine izhodnih nevronov. Vsak vhodni nevron je z enosmerno vezjo povezan z vsakim izhodnim nevronom (glej sliko 4). Izračun take nevronske mreže poteka tako, da se vhodnim nevronom vsilijo vrednosti komponent vhodnega vzorca in zatem v enem koraku vsi izhodni nevroni paralelno izračunajo izhodne vrednosti. Včasih pravijo takim mrežam tudi enonivojske, ker so vhodni nevroni samo senzorji. Primer take mreže je perceptron (Minsky & Papert 1969). Dvonivojske nevronske mreže lahko rešijo samo linearne probleme (npr. 'ekskluzivni ali' ni rešljiv z dvonivojsko mrežo).



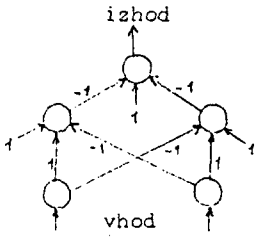
Slika 4: Dvonivojska nevronska mreža

4.1.3 Večnivojske nevronske mreže

Če v dvonivojski nevronski mreži med vhodni in izhodni nivo dodamo še enega ali več skritih nivojev, dobimo večnivojsko nevronsko mrežo. Večnivojske nevronske mreže delujejo podobno kot dvonivojske, le da je število korakov enako številu skritih nivojev plus ena. Z večnivojskimi mrežami lahko rešimo tudi nelinearne probleme (glej 4.3.2).

Primer tronivojske mreže, ki izračunava 'ekskluzivni ali', je na sliki 5. Dva nevrona sta vhodna, dva skrita in en izhodni. Skrita nevrona in izhodni nevron imajo tudi poseben konstanten vhod z utežjo 1. Mreža je simetrična

glede na vhod. Če sta oba vhodna nevrona v enakem stanju (oba 1 ali oba -1), potem sta oba skrita nevrona aktivna (imata izhod 1) in izhodni nevron postane pasiven (izhodna vrednost je -1). Če sta vhoda različna (eden 1 in drugi -1), potem sta tudi stanji skritih nevronov različni in je izhodni nevron aktiven (izhodna vrednost je 1).



vhod		izhod
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Slika 5: Tronivojska nevronska mreža, ki izračunava ekskluzivni ali.

4.1.4 Dvosmerni asociativni pomnilnik

Dvosmerni asociativni pomnilnik (Kosko 1987,1988) je sestavljen iz dveh nivojev nevronov, ki sta povezani med seboj z dvosmernimi vezmi. Izračunavanje poteka tako, da nevronom ustreznega nivoja vsilimo vrednosti komponent vhodnega vektorja. Zatem nevroni iz drugega nivoja izračunajo svoje izhode, zatem nevroni iz prvega nivoja izračunajo svoje izhode in tako iterativno naprej, dokler se izhodi obeh nivojev ne ustalijo. Delovanje nevronov je lahko sihrono ali asihrono. Že samo ime pove, da se taka mreža uporablja kot asociativni pomnilnik, ki na osnovi vhodnega vzorca izračuna ustrezen vzorec, ki je bil shranjen kot par danemu vhodnemu vzorcu. Dokazano je, da pri asinhronemu izvajanju dvosmerni asociativni pomnilnik konvergira pri poljubno izbranih utežeh na vezeh med nevroni (Kosko 1988).

4.2 Namen nevronske mreže

4.2.1 Avto-asociativni pomnilnik

Avto-asociativni pomnilnik ponavadi deluje tako, da so vsi nevroni hkrati vhodni in izhodni. Na vhodu damo vzorec ali del vzorca (pomankljiv vzorec). Mreža nato iterativno generira (aproksimira) manjkajoči del vzorca in popravlja napačne dele (šum) v vhodnem vzorcu. Ko se izvajanje ustali (nadaljnje iteracije ne spremenijo več vzorca), dobimo na izhodu celoten, dopolnjen in popravljen vzorec. Če je vhodni vzorec preveč pomankljiv ali napačen, se seveda lahko zgodi, da je dobljeni vzorec napačen. McEliece s sod. (1987) je dokazal, da z N nevroni lahko eksaktno shranimo N/(4*log N) vektorjev (N dimenzionalnih). Wong (1988) je empirično pokazal, da mreža z N nevroni lahko shrani N vektorjev.

4.2.2 Hetero-asociativni pomnilnik

Hetero-asociativni pomnilnik je varianta avto-asociativnega pomnilnika, pri kateri je vzorec sestavljen iz več podvzorcev. Na vhodu damo enega ali več podvzorcev, na izhodu pa dobimo preostali del vzorca.

4.2.3 Časovni asociativni pomnilnik

Časovni asociativni pomnilnik deluje kot hetero-asociativni pomnilnik, le da je na vhodu vedno vzorec iz nekega časovnega intervala, izhodni vzorec pa je napoved vzorca za naslednji časovni interval (Kosko 1988). Torej časovni asociativni pomnilnik deluje kot naivni kvalitativni simulator, ki na osnovi dane situacije generira naslednjo situacijo.

Učenje časovnega asociativnega pomnilnika poteka na osnovi primerov. Torej simulator upošteva samo površinsko znanje brez poznavanja zakonov in relacij, ki veljajo v dani problemski domeni. Zato je napovedovanje aproksimacija z upoštevanjem primerov, ki so bili mreži prej prezentirani, in ne izpeljava iz znanih zakonov iz domene.

Časovni asociativni pomnilnik lahko realiziramo tudi tako, da upošteva kontekst in določeno predznanje, kjer opis dane situacije dopušča več alternativnih nadaljevanj. V tem primeru je vhod razdeljen na podvzorec trenutne situacije in podvzorec konteksta.

4.2.4 Klasifikacija

Klasifikacija je poseben primer hetero-asociativnega pomnilnika, ki ima fiksno določene vhodne nevrone in fiksno število izhodnih nevronov, ki določajo razred. Učenje poteka na osnovi primerov z znanimi vhodnimi podatki in znanimi razredi. Primer, za katerega so znani samo vhodni podatki ali samo del vhodnih podatkov, se klasificira v enega od razredov.

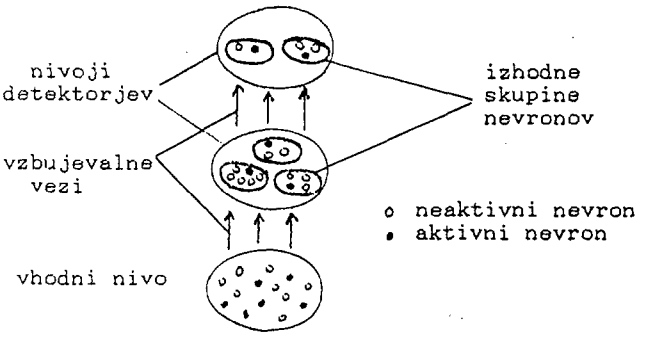
4.2.5 Grupiranje

Tu je problem podan tako, da mora sistem sam znati grupirati vhodne primere v fiksno število razredov. Število primerov je lahko v naprej podano ali pa tudi ne. Spodaj je opisan primer nevronske mreže iz (Rumelhart in Zipser 1986), ki zna grupirati primere v vnaprej določeno število razredov.

Mreža je sestavljena iz vhodnih in izhodnih nevronov. Vsak nevron je lahko aktiven ali pa neaktiven. Vsak vhodni nevron je povezan z vzbujevalnimi vezmi z vsakim izhodnim nevronom. Število izhodnih nevronov določa število končnih razredov. Vsak izhodni nevron je povezan z zaviralnimi vezmi z vsemi ostalimi izhodnimi nevroni, tako da je lahko naenkrat aktiven samo en izhodni nevron.

Pred učenjem ima vsak izhodni nevron na vezi z vsemi vhodnimi nevroni enako utež in sicer tako, da je vsota vseh uteži enaka 1. Med učnjem izhodni nevron premika uteži od neaktivnih vhodov k aktivnim tako, da vzdržuje vsoto uteži enako 1. Najbolj spreminja svoje uteži tisti izhodni nevron, ki postane aktiven (zmaga). Zato pravimo takemu učenju tudi tekmovalno učenje (competitive learning). Sčasoma se mreža nauči, da določen izhodni nevron zmaga vedno pri vhodnih vzorcih, ki so si po določenih lastnostih podobni.

Če povežemo več izhodnih skupin nevronov vzporedno brez medsebojnih povezav, lahko hkrati dobimo več različnih grupiranj. Vsaka izhodna skupina nevronov se nauči grupirati vhodne primere glede na določene lastnosti. Lahko rečemo, da se vsak nevron nauči detektirati določen nov atribut vhodnega vzorca. Zato takemu učenju pravimo tudi odkrivanje atributov (feature discovery). En nivo detektorjev atributov lahko serijsko povežemo v več nivojev (glej sliko 6). Vsak naslednji nivo bo dobil vhodne primere, opisane z nekimi novimi atributi.



Slika 6: Nevronska mreža za grupiranje

4.2.6 Samoorganizacija in sortiranje

Včasih je za učinkovito predstavitev informacije potrebno nevrone urediti tako, da vsak odgovarja na določeno vrednost vhodnega parametra (npr. frekvenca zvoka). Z ustrezno topologijo nevronov dosežemo, da se nevroni sami uredijo tako, da ustrezno ležeči nevroni odgovarjajo na ustrezne vhodne signale. Primer take mreže je dvonivojska mreža, kjer je vsak izhodni nevron povezan še z bližnjimi izhodnimi nevroni s povratnimi vezmi. Z najbližjimi nevroni je povezan z vzbujevalnimi vezmi in z manj bližjimi z zaviralnimi vezmi (Kohonen 1984). Vsak nevron sprejema na vhodu isti signal, ki ustreza realni vrednosti danega vhodnega parametra. Vsak nevron ima spremenljivo utež za vhodni signal, ki jo primerja z vhodnim signalom. Čim manjša je razlika uteži in vhodnega signala, tem močnejši bo nevronov odgovor.

Učenje poteka tako, da se spreminjajo samo vhodne uteži zmagovalnih nevronov, to je nevronov, ki pri danem vhodu postanejo aktivni. Vhodne uteži se spreminjajo v smeri vhodnega signala. Sčasoma bodo v nevronski mreži vhodne uteži urejene, torej bo lega nevrona, ki reagira na signal dane velikosti, odgovarjala velikosti vhodnega signala. Kohonen (1984) je pokazal konvergenco tega procesa pod določenimi pogoji.

V gornjem učenju Kohonen navaja dva procesa. Eden je ta, da uteži skušajo aproksimirati trenutni signal. Drugi pa je ta, da lokalne interakcije med nevroni skušajo ohranjati kontinuiteto. Analogijo lahko najdemo v algoritmu sortiranja po metodi mehurčkov. Dva sosednja elementa se podpirata, če sta v pravilnem vrstnem redu. To ustreza drugemu principu. Če pa nista v pravilnem vrstnem redu, prvi princip prevlada in vrednosti sosednjih elementov se zamenjata.

4.3 Pravilo učenja

4.3.1 Hebbovo pravilo

Osnovno pravilo, ki ga v različnih variantah uporabljajo nevronske mreže pri učenju, je definiral že Hebb (1949). Pravilo pravi, da se vez med dvema aktivnima nevronoma ojača (utež se poveča). To pravilo zadostuje, da si mreža zapomni frekvenco (in s tem verjetnost), s katero sta dva sosednja nevrona hkrati aktivna. Biološka verjetnost tega pravila (verjetnost, da se po takem ali podobnem pravilu učijo možgani) je v uporabi informacije, ki je lokalno dosegljiva vsakemu nevronu.

Posplošeno Hebbovo pravilo (Rumelhart in sod. 1986) pravi, da se vez med dvem nevronoma ojača, če sta oba hkrati aktivna ali če sta oba hkrati pasivna. Na ta način si mreža zapomni povezanost med aktivnostima dveh povezanih nevronov.

4.3.2 Pravilo delta

Preprosto pravilo, ki omogoča učenje v dvonivojski mreži, sta definirala že Minsky in Papert (1969). V dvonivojski mreži je prvi nivo nevronov vhoden in drugi izhoden. Vsak vhodni nevron je povezan z vsakom izhodnim nevronom. Pred začetkom učenja so uteži izbrane naključno. Učenje poteka z znanimi pari vhodnih in izhodnih vzorcev. Vhodni nevroni dobijo vhodni vzorec. Zatem se v enem koraku izračuna izhod pri danem vhodu. Izračuna se razlika med dejanskim izhodom in željenim izhodom. Zatem se uteži povezav med vhodnimi in izhodnimi nevroni zmanjšajo ali povečajo sorazmerno razliki med dejanskim in željenim izhodom.

Z iterativnim prikazovanjem znanih parov vhodnih in izhodnih vzorcev dosežemo, da se mreža nauči pravilno odgovarjati na vse vhode. Minsky in Papert (1969) sta dokazala, da ta algoritem konvergira k fiksni točki, če je funkcija, ki se jo mora mreža naučiti, linearna. Dejansko dvonivojska mreža ne zmore rešiti nelinearnih problemov (npr. ekskluzivni ali).

Rumelhart je s sodelavci (1986) razvil posplošeno pravilo delta. Le ta omogoča učenje mreže, sestavljene iz poljubnega števila nivojev. Večnivojska mreža lahko reši tudi nelinearne probleme (npr. ekskluzivni ali, glej sliko 5).

Osnovni princip posplošenega pravila delta je isti, kot pri navadnem pravilu delta. Na začetku so uteži naključne. Na vhodu mreža dobi vhodni vzorec in s propagiranjem po nivojih do izhodnega nivoja izračuna izhod. Zatem se izračuna razlika med dejanskim in željenim izhodom. Najprej se spremenijo uteži med zadnjim in predzadnjim nivojem kot pri osnovnemu pravilu delta. Zatem se izračunajo želene vrednosti nevronov na predzadnjem nivoju (to je ključni korak algoritma). Izračuna se razlika med želenim in dejanskimi vrednostmi nevronov na predzadnjem nivoju in rekurzivno se nadaljuje spreminjanje uteži vse do vhodnega nivoja nevronov.

Za posplošeno pravilo delta velja, da ne konvergira vedno (lahko obtiči v lokalnem minimumu). Razen tega zahteva zelo veliko število prehodov preko učnih primerov, t.j. mreži moramo velikokrat pokazati iste učne primere (tipično nekaj sto do nekaj tisoč krat). Tako bi nevronska mreža na sliki 5 potrebovala okoli 500 prehodov preko štirih učnih primerov za rešitev problema XOR (Rumelhart in sod. 1986). Prav tako pravilo delta in posplošeno pravilo delta nimata biološke analogije z možgani (Wassermann & Schwartz 1988).

4.3.3 Tekmovalno pravilo

Pri tekmovalnem pravilu se zahteva, da je v določeni skupini nevronov aktiven vedno natanko eden. To se doseže z ustrezno topologijo, kjer je v skupini vsak nevron povezan z vsakim z zaviralnimi vezmi. Ko en nevron postane aktiven, z zaviralnimi vezmi 'zatre' ostale nevrone, da ne morejo postati aktivni. Pri tem pravilu se uči samo zmagovalni nevron in sicer

tako, da poveča uteži vezem, ki so mu pomagale, da je 'zmagal', in zmanjša uteži ostalim vezem.

Primer tekmovalnega pravila je metoda grupiranja Rumelharta in Zipserja (1986) (glej 4.2.5). Kohonenov primer samoorganizacije, ki tudi vključuje tekmovalno pravilo, je opisan v razdelku 4.2.6.

4.3.4 Pozabljanje

Nekatere metode učenja vključujejo tudi pozabljanje. To dosežejo tako, da prej naučenega ne upoševajo enakovredno z novo naučenim. Na ta način je novo naučeno vedno "najbolj sveže". Včasih pozabljanje realizirajo tako, da vse uteži med nevroni s časom zvezno znižujejo v sorazmerju z njihovimi velikostmi. Ta metoda preprečuje, da bi se uteži preveč povečale (Kohonen 1984).

4.4 Funkcija kombiniranja vhodov nevrona v izhod

Funkcija kombiniranja je sestavljena iz dveh delov (glej sliko 2): funkcija aktivacije (A) in izhodna funkcija (f). Funkcija aktivacije kombinira vhode v vmesno vrednost. Izhodna funkcija, ki je tipično pragovni logični element, preslika rezultat aktivacije v izhod nevrona.

4.4.1 Funkcija aktivacije

Najbolj pogosta funkcija aktivacije je obtežena vsota:

$$A(Xj) = \sum_i Wij * Xi + Cj$$

kjer je Wij utež vezi med i-tim in j-tim nevronom, Xi stanje i-tega nevrona in Cj konstantna aktivacija j-tega nevrona (prag). Če opišemo stanja n nevronov z vektorjem

$$X = (X1, X2, \ldots, Xn)^T$$

in uteži vezi Wij med nevronoma i in j z matriko

$$M = \begin{bmatrix} W11 & W12 & \ldots & W1n \\ W21 & W22 & \ldots & W2n \\ \ldots & & & \ldots \\ Wn1 & Wn2 & & Wnn \end{bmatrix}$$

dobimo stanje Y nevronske mreže po eni iteraciji z

$$Y = M * X$$

Ker matrika M predstavlja linearno transformacijo, je funkcija aktivacije Y = A(X) linearna.

Bolj splošna je t.i. funkcija sigma-pi, katere splošna oblika je

$$sp(X1, \ldots, Xn) = \sum_{Si \in P} Wi * \prod_{k \in Si} Xk ,$$

kjer je P potenčna množica množice indeksov vseh nevronov. Si je množica indeksov nevronov, katerih stanja se med seboj zmnožijo in nato se zmnožek 'obteži' z utežjo Wi.

Williams (1986) je pokazal, da s funkcijo sigma-pi in s pragovnimi elementi lahko realiziramo poljubno monotono boolovo funkcijo. Pri tej funkciji je pomembno to, da je en sam nevron ne more realizirati, ker število podmnožic Si narašča s številom vhodnih nevronov. Vsak nevron ima na razpolago toliko spominskih enot za uteži Wi, kolikor je vhodnih nevronov. Zato pa lahko s kombiniranjem več nevronov realiziramo poljubno monotono funkcijo.

4.4.2 Izhodna funkcija

Izhodna funkcija, ki preslika aktivacijo v izhod nevrona, je lahko deterministična binarna, deterministična zvezna ali stohastična, ki je ponavadi binarna. Pri deterministični funkciji je izhod nevrona enolično določen z vhodom. Pri stohastični funkciji pa je izhod določen samo z določeno verjetnostjo.

Binarna deterministična funkcija je ponavadi pragovni logični element. Izhodne vrednosti so ponavadi 0 in 1 ali pa -1 in 1. Prag je lahko fiksen ali pa se ob učenju spreminja.

Zvezna deterministična funkcija je ponavadi sigmoidna funkcija, ki preslika poljubno realno vrednost na interval od 0 do 1. Izhod nevrona je torej poljubno realno število med 0 in 1. Zvezna funkcija občutno poveča pomnilniško kapaciteto nevronske mreže. Guez s sod. (1988) je pokazal, da ima mreža z N nevroni 3 na N fiksnih točk. Primer take funkcije je

$$f(X) = 1/(1 + e^{-X})$$

Da se izognejo lokalnim rešitvam, uporabljajo v nevronskih mrežah tudi stohastične funkcije kombiniranja. To so funkcije, ki ne dajo ene ali druge vrednosti, vrnejo verjetnost za npr. prvo vrednost. Nevron nato izbere eno od vrednosti z dano verjetnostjo. Proces konvergence je zato počasnejši. Na ta način se lahko izognemo večini manjših lokalnih ekstremov in sistem z veliko verjetnostjo poišče globalni optimum.

V zvezi s stohastičnimi funkcijami je povezana vpeljava temperature sistema po analogiji iz termodinamike. Čim višja je temperatura, tem večja je entropija sistema. Z zniževanjem temperature sistem postaja bolj in bolj deterministično. Pri temperaturi 0 so vse odločitve deterministične. Proces v sistemu začnemo pri visoki temperaturi, ki jo počasi znižujemo. Čim počasneje se temperatura znižuje, tem večja je verjetnost, da bo sistem pristal v globalnem optimumu.
Primer stohastične funkcije je Smolensky-jeva teorija harmonije (1986). Harmonija je obratno sorazmerna Hopfieldovi energiji sistema. Čim večja je harmonija, tem nižja je energija, tem bližje je sistem rešitvi. V globalnem optimumu je harmonija največja. Stohastično funkcijo uporabljata tudi Hinton in Sejnowski (1986) v Boltzmannovih strojih (funkcija sledi Boltzmannovi distribuciji).

5. PRIMERI UPORABE

V zadnjem času je vse več komercialno dosegljivih paketov za razvoj nevronskih mrež (Nestor Development System, Cognitron, NeuralWorks, NetsSet, Axon itd.), prodajajo pa tudi že kartico (ANZA) za IBM-PC kompatibilne računalnike in SUN delovne postaje, ki omogoča nekaj 100 krat hitrejše izvajanje. Paketi omogočajo hitro definicijo nevronske mreže in uporabo nekaterih standardnih modelov nevronskih mrež kot so Hopfieldov, Boltzmannov, Grossbergov, 'back-propagation', itd. Za Nestor Development system navajajo naslednje aplikacije (Nestor Inc. 1988):

- Kontrola kvalitete izdelave trdih diskov, ki v naprej napove možna odstopanja od zahtevane natančnosti, tako da jih pravočasno z ustrezno obdelavo lahko odpravimo. Navajajo 100 % natančnost klasifikacije delnih izdelkov v eno od štirih kategorij.
- Diagnostika na osnovi signala EKG. Navajajo 100 % natančnost ločevanja med normalnim EKG in določeno aritmijo.
- Razpoznavanje ročno zapisanih znakov. Navajajo 97.7 % natančnost, pri tem da lahko sistem izloči znake, ki jih ne more zanesljivo klasificirati.
- Preverjanje podpisov na čekih. Navajajo 4 % podpisov narobe klasificiranih kot napačen podpis, kar daleč presega ljudi strokovnjake.
- Identifikacija objektov z radarjem. Navajajo 100 % natančnost pri določenem problemu v primerjavi z 93% natančnostjo Bayesovega klasifikatorja.

Veliko je aplikacij v računalniškem razpoznavanju govora in računalniškem vidu. Kohonen (1988) navaja od 92 do 97 % natančnost razpoznavanja govora (pisanja po nareku, ne da bi sistem razumel, kaj piše) in 96 do 98 % natančnost razpoznavanja izgovorjenih izoliranih besed iz slovarja 1000 besed. Uporabljena metoda učenja je samoorganiziranje (glej razdelek 4.2.6), kjer se vsak nevron specializira za določen vhodni signal. Računalniški vid bi moral biti invarianten glede na rotacijo, translacijo in velikost opazovanih objektov. Hinton (1981) je pokazal, kako lahko sistem razpozna vhodni vzorec, čeprav ne ve vnaprej, za kakšen vzorec gre in katera transformacija je potrebna. Ideja je v tem, da sistem poskuša paralelno vse transformacije naenkrat in jih primerja z vsemi shranjenimi vzorci naenkrat (kar je naravni princip v nevronskih mrežah). Z iterativnim ponavljanjem se bosta ojačala tista transformacija in tisti vzorec, ki najbolj ustrezata vhodnemu vzorcu.

Fukushima (1988) je s posebno topologijo nevronske mreže dosegel razpoznavanje cifer invariantno glede na velikost in translacijo in delno invariantno glede na deformacijo. Widrow in Winter (1988) predlagata razpoznavanje v dveh korakih z dvema nevronskima mrežama. Prva mreža bi se naučila spreminjati vhodne vzorce v vzorce invariantne glede rotacije, translacije in velikosti. Druga mreža bi se lahko naučila iz vmesnih vzorcev generirati originalne v standardni poziciji, orientaciji in velikosti.

Grabec in Sachse (1988) sta uporabila nevronsko mrežo za analizo in procesiranje signalov

akustične emisije. Pokazala sta, da se z nevronsko mrežo lahko učinkovito reši inverzni problem akustične emisije, t.j. odkrivanje izvora zvoka na netrivialnem problemu, ki je eksaktno praktično nerešljiv. Nevronska mreža se brez uporabe enačb iz elastodinamične teorije iz primerov nauči odkrivati karakteristike izvora akustičnih signalov. Ta aplikacija napoveduje novo generacijo merilnih instrumentov, ki naj bi temeljila na principu nevronskih mrež.

Nevronske mreže lahko hitro najdejo približne rešitve NP-polnih problemov (v linearnem ali celo konstantnem času). Pri tem mora število nevronov ustrezati velikostnemu parametru problema. Hopfiled in Tank (1985) sta demonstrirala hitro reševanje problema trgovskega potnika (pri N danih mestih poiskati najkrajšo pot skozi vsa mesta, tako da nobenega mesta ne obiščemo več kot enkrat).

Barto in sod. (1983) zagovarjajo bolj kompleksne modele nevronov. Samo dva (različna) ustrezno povezana nevrona sta se naučila balansirati palico na premičnem vozičku mnogo bolje kot originalni sistem BOXES, ki sta ga razvila Michie in Chambers (1968). En nevron se je na napakah učil kontrolirati voziček iz danega opisa stanja sistema. Drugi nevron pa se je na napakah naučil iz danega stanja sistema napovedovati napake (odkrival je kritična stanja sistema).

Popolnoma drugo področje uporabe nevronskih mrež je kognitivno modeliranje. Nevronske mreže so dovolj močne za simulacijo določenih kognitivnih procesov, po drugi strani pa dovolj preproste in formalno definirane, da jih je razmeroma preprosto modelirati in analizirati. Hood (1986) uporablja nevronske mreže za raziskovanje učenja nižjih organizmov. Pazzani in Dyer (1987) primerjata človekovo učenje konceptov in učenje nevronskih mrež.

6. ANALOGIJA Z MOZGANI

Možgani so vsebinsko naslovljivi. Mentalni procesi uporabljajo možgane v celoti. Ni možno lokalno ločiti funkcije pomnilnika od procesorjev, kot je to možno v klasičnih digitalnih računalnikih. Pomnilnik v možganski skorji je distribuiranega tipa in ne lokalnega. Najbolj pomemben delež zapomnjenja v možganih prispevajo sinaptične povezave (Kohonen 1984).

Hitrost nevronov v možganih je velikostnega reda milisekund. Percepcija, procesiranje jezika, intuitivno razmišljanje, dostop do spomina zahteva v možganih približno desetinko sekunde, torej kakih 100 korakov. Tako je Feldman definiral "100 koračni program" kot omejitev za izvajanje elementarnih operacij programa, ki bi obrazložil mentalne procese. Realizirati tak program je možno seveda samo z uporabo izredno visoke stopnje paralelizma.

Tako v možganih kot v nevronski mreži učinkovitost počasi (zvezno) pada s številom uničenih nevronov (Russell 1979). Ni nobenega ključnega nevrona, katerega okvara bi 'sesula' celoten sistem. V možganski skorji ni nobenega dela, od katerega bi bili odvisni vsi ostali deli (Rumelhart in McClelland 1986a). Prav tako noben del možganov ni nenadomestljiv. Dokazano je npr., da pri otrocih, rojenih samo z eno možgansko poloblo, le ta prevzame funkcije manjkajoče poloble in otroci odrastejo brez okvarjenih funkcij (Russell 1979).

Človeška percepcija je do določene mere invariantna glede na velikost, translacijo in rotacijo (Kohonen 1984). Fizični sistem je sposoben avtomatično generirati reducirano reprezentacijo. Različna področja v možganski skorji so se specializirala za različne senzorske signale in sicer tako, da ohranjajo topologijo prostorskih senzorjev. Ta princip lahko simuliramo z nevronsko mrežo (glej 4.2.6). Kohonen (1984) navaja, da je precej anatomske in psihološke evidence, da obstojajo v možganski skorji povezave po istem principu, kot je opisano v razdelku 4.2.6.

Za učinkovito delovanje nevronske mreže ni potrebno povezati vsak par nevronov med seboj. Za določeno statistično natančnost zadošča, da je število povezav, če so le-te statistično porazdeljene po mreži, mnogo večje od števila nevronov. Če bi možgani z 10^{12} nevroni in s povprečno 10^3 povezavami na nevron delovali po principu preprostega pragovnega elementa, imajo zadostno kapaciteto, da z zadostno natančnostjo shranijo vse, kar človek doživi v 100 letih (Kohonen 1984). Tudi psihologi navajajo, da je precej empirične evidence, da si lahko človek zapomni prav vse, kar se mu pripeti v življenju (Russell 1979).

Geni ne določajo vseh povezav v možganih, določene povezave so omejene zaradi prostorske lege, in očitno je del povezav nakljućen (Rumelhart in McClelland 1986a). V možganih ni niti hardware-a v strogem pomenu besede niti software-a v običajnem pomenu besede (glej razdelek 3.6).

Rekurzija ni domača človekovemu načinu razmišljanja. Rumelhart in McClelland (1986a) navajata stavek "The man the boy the girl hit kissed moved" kot preprost primer rekurzivnega stavka, ki pa je očitno težko razumljiv.

Modeli nevronskih mrež ne sledijo popolnoma analogiji z možgani. Večina nevronov v možganski skorji ima ali vzbujevalne ali zaviralne vezi, kar je v nasprotju z nevronsko mrežo, ki dopušča obe vrste vezi pri enem nevronu. Velikost signalov v možganih je podana s frekvenco impulzov in ne z jakostjo signala samega kot v modelih nevronskih mrež (Kohonen 1984). Današnji modeli nevronskih mrež ne upoštevajo globalne komunikacije, ki v možganih poteka s pomočjo določenih kemičnih snovi, ki se pretakajo po krvi med različnimi predeli v možganih.

7. RELACIJA Z UMETNO INTELIGENCO

Zastarel očitek nevronskim mrežam je ta, da niso zmožne vsega izračunati. Že Minsky in Papert (1969) sta pokazala, da lahko univerzalni računski stroj simuliramo z nevronsko mrežo. Seveda pa ta rezultat nima praktične uporabnosti.

Bolj pereč problem pri nevronskih mrežah je težavnost razložiti svojo odločitev. Kot že rečeno v razdelku 3.8, z gledišča raziskovalcev umetne inteligence tak sistem ni boljši od katerega koli statističnega sistema za razpoznavanje vzorcev, ki lahko dobro deluje, vendar interpretacija njegovih rezultatov ni možna brez veliko znanja in izkušenj na področju matematike in statistike.

Današnje raziskave umetne inteligence so usmerjene v razvoj metod in orodij, ki bi omogočile računalnikom bolj inteligentno obnašanje. Pri tem je temeljna zahteva, da zna sistem svojo odločitev obrazložiti in argumentirati. Inspiracije za razvoj metod pogosto pridejo iz analogije s človekovim načinom razmišljanja, vendar samo na visokem, simboličnem in logičnem nivoju. Tako se raziskovalci ozirajo na to, kaj so možgani zmožni napraviti, ne pa tudi, kako to dejansko napravijo.

Pristop z nevronskimi mrežami prav tako išče analogijo s človeškimi možgani, vendar na nizkem, 'subsimboličnem' nivoju. Pri tem raziskovalcev ne zanima samo, kaj možgani zmorejo, ampak tudi, kako možgani delujejo.

Standardni opis človekovega učenja, ki ga navajajo raziskovalci umetne inteligence, je naslednji (Smolensky 1986). Neizkušeni strokovnjaki pri svojem delu uporabljajo splošna pravila, veljavna v dani domeni. Ker so pravila splošna, je delo z njimi počasno. Sledenje pravilom je zavestno. Sčasoma pa si strokovnjak ob reševanju problemov ustvarja svoja bolj specialna pravila, ki jih lahko uporabi brez poglabljanja v samo teorijo, zato postane njegovo delo hitrejše. Sledenje specialnim pravilom je podzavestno.

Večina prevladujočih računalniških modelov v umetni inteligenci je daleč od biološke podobnosti in analogije. Induktivno učenje (Kononenko 1985), ki temelji na metodah umetne inteligence, generira eksplicitna specialna pravila na osnovi že rešenih problemov, kar zahteva velike računalniške zmogljivosti.

Podobno velja pri nevronski mreži, le da specialna pravila niso eksplicitno shranjena, ampak se dinamično kreirajo po potrebi. Shranjeni vzorci, pridobljeni skozi izkušnje, se lahko kombinirajo dinamično na popolnoma nov način, kar omogoča generiranje pravila, ki ga strokovnjak sam ni nikoli videl, oziroma hitro rešitev problema, s katerim se je strokovnjak prvič srečal. Mreža, ki deluje kot klasifikator, se obnaša tako, kot da pozna pravila. Za izvajanje zadoščajo preproste procesne enote, ki delujejo asinhrono in potrebujejo samo lokalno dosegljivo informacijo.

Lahko bi rekli, da nevronske mreže simulirajo funkcijo desne možganske poloble, ki deluje bolj paralelno in podzavestno (intuitivno), medtem ko metode umetne inteligence simulirajo funkcije leve možganske poloble, katere delovanje je bolj zavestno, logično in zaporedno (čeprav je seveda osnovni princip

obeh polobel enak). Z ustrezno združitvijo obeh metod lahko pričakujemo velik skok v zmožnosti in učinkovitosti računalniške obdelave.

Rumelhart in McClelland (1986a) poudarjata, da ko bomo dovolj razumeli mikro nivo, bomo mogoče hoteli formulirati popolnoma drugačne makro nivojske modele. Navajata namišljeni računalnik, ki bi imel primitivne ukaze kot so: "sprosti se v stanje, ki optimalno interpretira trenutni vhod", "dobi iz pomnilnika reprezentacijo, ki maksimalno ustreza trenutnemu vhodu in dodaj manjkajoče podrobnosti k shranjeni reprezentaciji" ter "konstruiraj dinamično konfiguracijo struktur znanja, ki ustreza dani situaciji z ustrezno opredeljenimi spremenljivkami". Takemu sistemu bi bolje ustrezalo ime 'relaksator' kot računalnik.

ZAHVALA

Zahvaljujem se svojim kolegom Marku Bohancu, Marku Coklinu, Andreju Dobnikarju, Matevžu Kovačiču in Tanji Urbančič za natančen pregled rokopisa in za koristne pripombe, ki so precej dopolnile tale prispevek. Zahvaljujem se Andreju Dobnikarju, Bojanu Petku in Igorju Grabcu za skupne napore pri zbiranju literature.

LITERATURA

Barto, A.G., Sutton, R.S. & Anderson C.W. (1983) Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems. IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-13, no. 5, pp. 834-846.

Blelloch, G. & Rosenberg, C.R. (1987) Network Learning on the Connection Machine. Proc. of 10th Internat. Conf. on Artificial Intelligence, Milano, August 23-28, pp. 323-326.

Grabec, I. & Sachse, W. (1988) Application of an Intelligent Signal Processing System to Acoustic Emission Analysis. Report #6428, Dep. of Theoretical and Applied Mechanics, Cornell University, Ithaca, New York.

Graf, H.P., Jackel, L.D & Hubbard, W.E. (1988) VLSI Implementation of a Neural Network Model. IEEE Computer, March 1988, pp.41-49.

Guez, A., Protopopsecu, V. & Barhen, J. (1988) On the Stability, Storage Capacity and Design of Nonlinear Continuous Neural Networks. IEEE Trans. on Systems, Man, and Cybernetics, Vol. 18, No. 1, pp. 80-87.

Fukushima, K. (1988) A neural Network for Visual Pattern Recognition., IEEE Computer, March 1988, pp. 65-75.

Hebb, D.O. (1949) The Organization of Behavior. New York: Wiley.

Hinton, G.E. (1981) A parallel computation that assigns cannonical object based frames of reference. Proc. of 7th. Internat. joint conf. on AI.

Hinton, G.E., McClelland, J.L. & Rumelhart, D.E. (1986) Distributed Representations. in Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Hinton, G.E. & Sejnowski, T.J. (1986) Learning and Relearning in Boltzmann Machines. in Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Hood, G. (1986) Neural Modeling as one Approach to Machine Learning. in Mitchell, T.M., Carbonell, J.G. & Michalski, R.S. (eds.) Machine Learning - A Guide to Current Research. Kluwer Academic Publichers.

Hopfield, J.J. & Tank, D.W. (1985) "Neural" Computation of Decisions in Optimization Problems. Biological Cybernetics, Vol. 52, pp.141-152.

Jordan, M.I. (1986) An Introduction to Linear Algebra in Parallel Distributed Processing. in Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Kohonen, T. (1984) Self-Organization and Associative Memory. Berlin: Springer-Verlag.

Kohonen, T. (1988) The "Neural" Phonetic Typewriter. IEEE Computer, March 1988, pp. 11-22.

Kononenko, I. (1985) Strukturno avtomatsko ucenje. Informatica, vol. 9, no. 4, pp. 44-55.

Kosko, B. (1987) Constructing an Associative Memory. Byte, september 1987, pp. 137-144.

Kosko, B. (1988) Bidirectional Associative Memories. IEEE Trans. on Systems, Man, and Cybernetics, Vol. 18, no. 1, pp.49-50.

McClelland, J.L., Rumelhart, D.E. & PDP Research Group (1986)
Parallel Distributed Processing, Vol. 2: Psychological and Biological Models. Cambridge: MIT Press.

McClelland, J.L., Rumelhart, D.E. & Hinton, G.E. (1986a) The Appeal of Parallel Distributed Processing. In: Rumelhart, D.E. & McClelland, J.L. (eds.), Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

McEliece, R.J., Posner, E.C., Rodemich, E.R. & Venkatesh, S.S. (1987) The Capacity of the Hopfield Assiciative Memory. IEEE Trans. on Information Theory, Vol IT-33, No. 4, pp. 461-482.

Michie, D. & Chambers, R.A. (1968) BOXES: An Experiment in Adaptive Control. in Dale, E. & Michie, D. (eds.) Machine Intelligence 2. Edinburgh: Oliver and Boyd.

Minsky, M. & Papert, S. (1969) Perceptrons. Cambridge, MA: MIT Press.

Nestor Inc. (1988) The Nestor Development System. Reklamni material.

Pazzani, M. & Dyer, M. (1987) A Comparison of Concept Identification in Human Learning and Network Learning with Generalized Delta Rule. Proc. of 10th Internat. Conf. on Artificial Intelligence, Milano, August 23-28, pp.147-150.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986) Learning Internal Representations by Error Propagation. in: Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Rumelhart, D.E. & McClelland, J.L. (eds.) (1986) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Rumelhart, D.E. & McClelland, J.L. (1986a) PDP Models and General Issues in Cognitive Science. in: Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Rumelhart, D.E. & Zipser (1986) Feature Discovery by Competitive Learning. in Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Russell, P. (1979) The Brain Book. Routledge and Kegan Paul, London in Hawthorne, New York.

Smolensky, P. (1986) Information Processing in Dynamical Systems: Foundations of Harmony Theory. in Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Wasserman, P.D. & Schwartz, T. (1988) Neural Networks (Part 2): What are They and Why are Everybody so Interested in them Now. IEEE Expert, Vol. 3, pp.10-15.

Widrow, B. & Winter, R. (1988) Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition. IEEE Computer, March 1988, pp.25-39.

Williams, R.J. (1986) The Logic of Activation Functions. in Rumelhart, D.E. & McClelland, J.L. (eds.) Parallel Distributed Processing, Vol. 1: Foundations. Cambridge: MIT Press.

Wong, A.J.W. (1988) Recognition of General Patterens Using Neural Networks. Biological Cybernetics, Vol. 58, no. 6, pp.361-372.

DODATEK: SLOVARČEK NEKATERIH ANGLEŠKIH STROKOVNIH IZRAZOV

activation function - funkcija aktivacije, ki kombinira vhode nevrona v stanje nevrona

autoassociative memory - avto asociativni pomnilnik:delni vzorec prikliče celotni vzorec

back propagation rule - glej 'generalized delta rule'

competitive learning - tekmovalno učenje; učenje poteka tako, da se uči samo nevron, ki zmaga v tekmovanju - postane aktiven, medtem ko so vsi ostali nevroni iz istega nivoja neaktivni

content addressible memory - vsebinsko naslovljiv pomnilnik (asiciativni pomnilnik); podatek se prikliče, če je podan del podatka

delta rule - pravilo učenja, ki spreminja uteži vezi med nevroni sorazmerno razliki med dejanskim in željenim izhodom

eigen value - lastna vrednost matrike

eigen vector - lastni vektor matrike

energy function - energijska funkcija opisuje stanje v nevronski mreži; lokalni minimum te funkcije ustreza fiksni točki

excitatory connection - vzbujevalna vez med dvema nevronoma: povečanje aktivnosti prvega nevrona poveča aktivnost drugega nevrona

fixed point - stabilna točka (stanje nevronske mreže opisano zvrednostmi za posamezen nevron), ki se z nadaljnimi iteracijami ne spremeni

generalized delta rule - posplošeno pravilo delta (glej 'delta rule'), ki omogoča učenje večnivojskih nevronskih mrež

Hamming distance - število komponent, v katerih se dva vektorja razlikujeta

Hebbian learning rule - pravilo učenja v nevronski mreži, ki ojača povezavo med dvema nevronoma, če sta oba nevrona aktivna

heteroassociative memory - hetero asociativni pomnilnik; vzorec prikliče ustrezno n-terico vzorcev

inhibitory connection - zaviralna vez med dvema nevronoma; povečanje aktivnosti prvega nevrona zmanjša aktivnost drugega nevrona

inner product - skalarni produkt dveh vektorjev

neural network - nevronska mreža

outer product - produkt dveh vektrojev interpretiranih kot matrike, tako da je prvi stolpčni in drugi vrstični vektor; rezultat je korelacijska matrika

output function - izhodna funkcija (tipično pragovni element), ki preslika aktivacijo nevrona (glej activation function) v njegov izhod

radius of attraction - maksimalna razdalja od fiksne točke v asociativnem pomniln., ki še omogoča konvergenco v dano fiksno točko

temporal associative memory - časovni asociativni pomnilnik; na vhodu damo vzorec iz danega časovnega intervala, na izhodu pa dobimo vzorec iz naslednjega časovnega intervala