

Konekcijam, nevronske mreže in dinamični sistemi

- Konekcionistični pristop (nevronske mreže, paralelno distribuirano procesiranje) so navdihnili spoznanja nevroznanosti o možganih
- modeli vključujejo pomembne značilnosti možganske arhitekture
- zajemajo delovanje nevrona, nevronske mreže in možganov na bolj abstraktni ravni

- Osnovna značilnost konekcionističnega modela je, da je sestavljen iz preprostih, neinteligentnih enot (idealizirani nevroni), ki so medsebojno povezane.
- Vsaka enota ima določeno aktivacijsko vrednost, ki jo preko vezi, ki so različno močne, posreduje drugim enotam in s tem pripomore k povečanju ali zmanjšanju vrednosti teh drugih enot.
- Cel proces se odvija vzporedno in ne potrebuje nobenega osrednjega dela za nadzor.

Zgodovina

- McCulloch in Pitts (1943)
- Rosenblatt (1958) - dvonivojske mreže (Perceptron), ki pa so bile še dokaj omejene v nalogah, ki so jih lahko izračunale (npr. računanje funkcije isključujoči ali je bila pretrd oreh).
- Z odkritjem učnih algoritmov za večnivojske mreže v 80-tih letih (PDP) so bile odpravljene nekatere omejitve in s tem odprta pot hitrejšemu razvoju.
- Proučevanje mrež v vmesnem obdobju ni zamrlo, čeprav je bilo v senci klasične umetne inteligence. Npr. autoasociativne mreže Kohonena in Hopfieldov vnos idej s področja fizike, predvsem statistične mehanike.

Značilnosti

- umetna nevronska mreža ne deluje po vnaprej napisanem programu za manipuliranje s shranjenimi simbolnimi izrazi
- program je prisoten le kot učni algoritem, po katerem se mreža prilagodi na okolje oziroma se na podlagi učnih primerov nauči določene naloge

Osnovne operacije

- izračun nove aktivacijske vrednosti za posamezno enoto kot funkcije obtežene vsote vhodnih podatkov,
- izračun izhodne vrednosti iz enote,
- pravilo za popravljanje vrednosti uteži (učno pravilo).

- Moč vezi predstavljajo uteži w_{ji} , tako da pozitivne uteži predstavljajo vzbujevalne povezave, negativne uteži zaviralne povezave, če pa je utež enaka 0, povezave ni.
- Aktivacijska funkcija $A(a_i)$ preslika vhodne mrežne podatke v aktivacijsko stanje enote a_i , ki ga nato preslika z izhodno funkcijo $F(a_i)$ v izhodni podatek o_{ji} in ga posreduje drugim enotam.

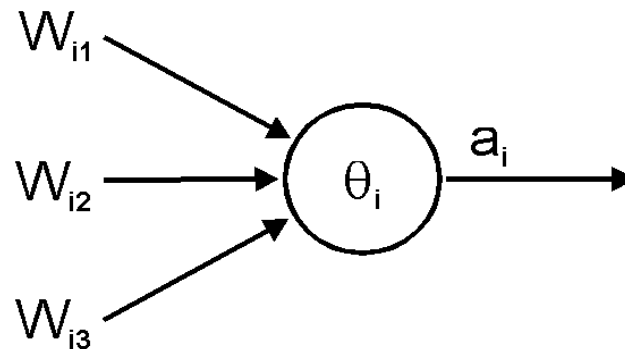


Diagram of a McCulloch-Pitts neuron

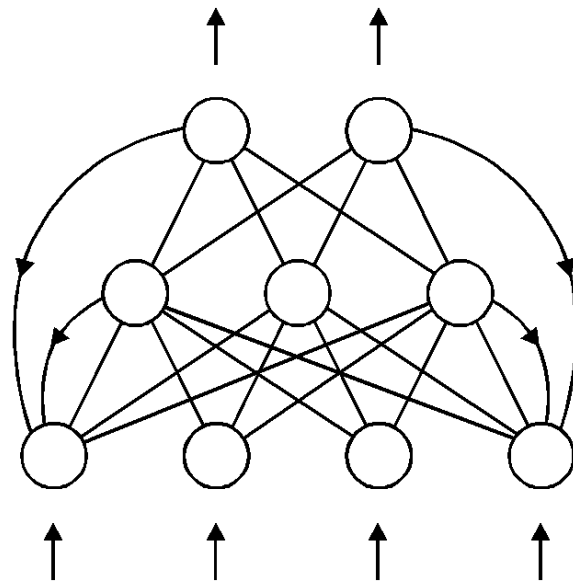
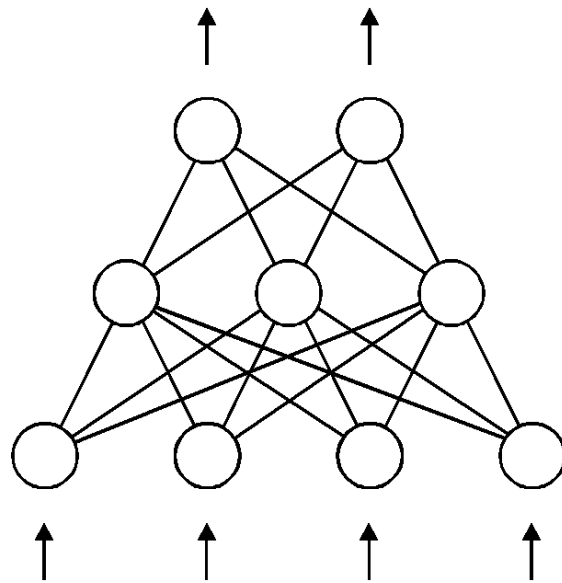
$$a_i(t+1) = F(\sum w_{ji} o_j(t) - \theta_i)$$

$$F(x) = 1, \text{ če } x \geq 0$$

$$F(x) = 0, \text{ če } x < 0$$

- Konekcionistični modeli (mreže) imajo lahko različne arhitekture.
- Razlikujemo jih:
 - glede na topologijo - koliko nivojev imajo (eden, dva ali več) in kako so enote med seboj povezane (enosmerno ali dvosmerno),
 - glede na to, katero funkcijo aktivacije in katero izhodno funkcijo izberemo
 - glede na izbiro učnega pravila.

tronivojski mreži



a b

a) z enosmernimi povezavami (“feedforward”)

b) s povratnimi povezavami (“recurrent”)

Funkcija aktivacije

- Delovanje posamezne enote je določeno z aktivacijsko funkcijo in izhodno funkcijo in od izbire obeh je odvisno vedenje sistema: kako uspešen bo pri razvrščanju značilnosti vhodnih podatkov in kategorizaciji, ali bo dosegel globalno optimalno stanje, ali pa se bo zaustavil na lokalnem ekstremu.
- Aktivacijska funkcija določa, kako se bodo vhodni podatki preslikali v aktivacijsko stanje enote
 - $A(a_j) = \sum w_{ji} * a_j + \theta_i$
 - kvadratna $A(a_j) = \sum w_{ji} * a_j^2 + \theta_i$
 - sferična $A(a_j) = \sum (a_j - w_{ji})^2 - \theta_i$ utež vezi med i-to in j-to enoto, a_j stanje j-te enote in θ_i prag i-te enote

izhodna funkcija

- preslika aktivacijo enote v izhod enote
 - deterministična binarna
 - deterministična zvezna
 - stohastična
- Pri deterministični funkciji je izhod iz enote enolično določen z vhomom, pri stohastični 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. Primer take funkcije je $f(x) = 1/(1+e^a)$.
- Stohastične funkcije, kjer funkcije vrnejo verjetnost za posamezne vrednosti. Enota potem izbere eno od vrednosti z dano verjetnostjo. Proces konvergence (približevanje stabilnemu stanju) je zato počasnejši, vendar na ta način sistem z večjo verjetnostjo poišče globalni optimum.

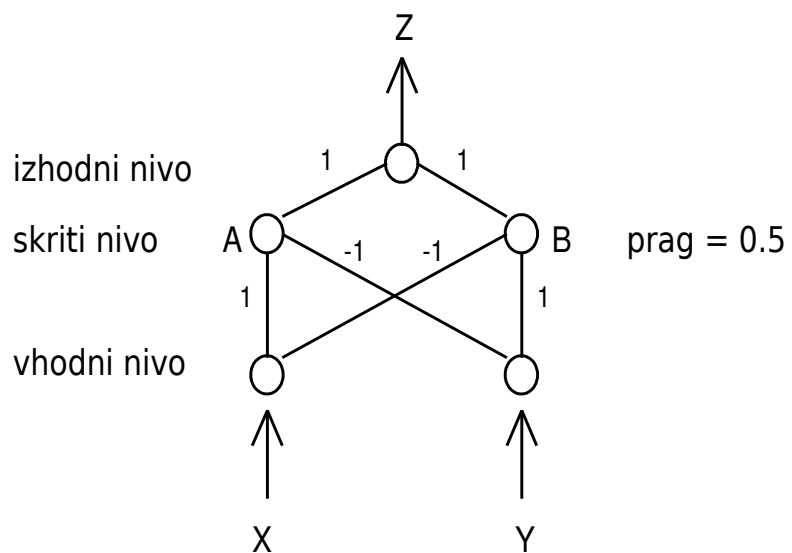
Učna pravila

- S pomočjo učnega pravila sistem postopno spreminja moč povezav. Proces učenja lahko poteka nenadzorovano ali pa nadzorovano.
- Nenadzorovano
 - algoritmi učenja, ki temeljijo na pravilu, ki ga je za učenje že leta 1949 predlagal nevrofiziolog Hebb, ko je opazoval spreminjanje jakosti sinaps med dvema živčnima celicama. Po tem pravilu se med dvema enotama, ki sta istočasno aktivni, poveča moč povezave, v vseh drugih primerih pa se zmanjša. Na ta način se sistem samoorganizira.
 - Tekmovalno pravilo, s pomočjo katerega mreža razvršča vhodne podatke (podobne vhodne podatke uvrsti v isto kategorijo - aktivira isto izhodno enoto). Pri tem pravilu se zahteva, da je aktivna samo ena izhodna enota (ali ena sama enota iz skupine). Najprej so vse enote povezane samo z zaviralnimi vezmi, ko pa ena enota postane aktivna, z zaviralnimi vezmi prepreči drugim enotam, da bi postale aktivne. Zmagovalna enota se uči tako, da poveča uteži vezem, ki so ji pomagale k “zmagi”, in zmanjša uteži ostalim vezem.

Nadzorovano učenje

- algoritmi učenja, kjer primerjamo izhodne podatke z znanimi pravilnimi odgovori - učenje z učiteljem
- Najbolj razširjena so delta pravila, pri katerih učenje poteka z znanimi pari vhodnih in izhodnih vzorcev (učni primeri). Po delta pravilu se v vsakem koraku izračuna izhod pri danem vhodu in razlika med dejanskim izhodom in želenim izhodom. Uteži povezav se potem zmanjšajo ali povečajo sorazmerno razliki. Preprosto pravilo omogoča učenje v dvonivojskih mrežah, vendar je uspešno le, če je funkcija, ki se jo mora naučiti, linearna (ne more rešiti nelinearnih problemov, kot je npr. "funkcija izključujoči ali"). Posplošeno delta pravilo (backpropagation) omogoča učenje tudi v večnivojskih mrežah, tako da za vsak nivo izračuna željene vrednosti in razliko med željeno in dejansko vrednostjo. Tako lahko mreža reši tudi nelinearne probleme, vendar pa je potrebno postopek zelo velikokrat ponoviti.

trinivojska mreža za “izključujoči ali”



X	Y	vhod v A	vhod v B	A	B	Z
0	0	$0x1+0x(-1)=0$	$0x(-1)+0x1=0$	0	0	$0x1+0x1=0$
1	0	$1x1+0x(-1)=1$	$1x(-1)+0x1=-1$	1	0	$1x1+0x1=1$
0	1	$0x1+1x(-1)=-1$	$0x(-1)+1x1=1$	0	1	$0x1+1x1=1$
1	1	$1x1+1x(-1)=0$	$1x(-1)+1x1=0$	0	0	$0x1+0x1=0$

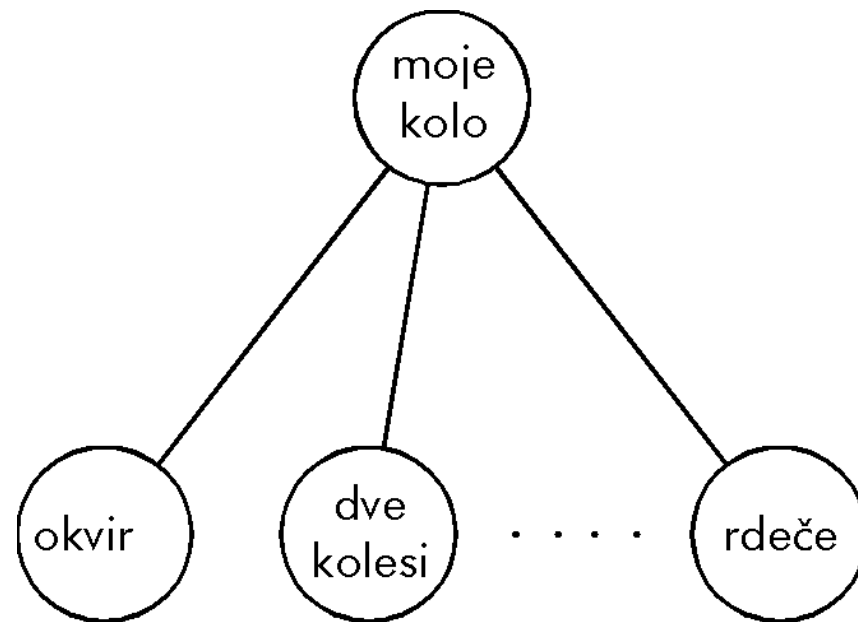
- Kadar je mreža v fazi učenja, obdeluje učne primere in po izbranem učnem algoritmu popravlja uteži povezav. Učenje se zaključi, ko se uteži ob ponovni predstavitvi učnega primera ne spreminjajo več (oz. so razlike v vrednostih pod določeno mejo).
- V interaktivnih mrežah (npr. Hopfildove mreže, Boltzmanov stroj), kjer so enote dvosmerno povezane ena z drugo, se aktivacijske vrednosti dinamično spreminjajo skozi večje število ciklov: mreža se samoorganizira – **dinamični sistemi**

- V konekcionističnem modelu vsaka enota deluje le v svojem lokalnem okolju in dobi informacije samo od enot, s katerimi je povezana. Informacija, ki jo enota posreduje, je signal, izražen s številsko vrednostjo in se nanaša na aktivacijsko stanje enot. Informacije o sosedah so edino, kar posamezna enota “ve”. Nobena enota ne “ve”, kaj dela sistem v celoti.
- Ker se uteži spreminjajo glede na spreminjanje vhodnih podatkov (učnih primerov, okolja), predstavlja utež povezanost glede na vse pretekle situacije. Znanje sistema je tako porazdeljeno v utežeh povezav med enotami.
- Pomnilnik, ki je pri klasičnih računalnikih poseben del, je v konekcionističnih sistemih del procesa in je porazdeljen v povezavah med enotami. Čeprav vsa pravila delujejo lokalno, pa v mrežah prihaja do globalnega sodelovanja. To se pojavi spontano, takrat, ko vse sodelujoče enote dosežejo medsebojno zadovoljivo stanje.
- Glavna značilnost dinamičnih mrež - prehod od lokalnih pravil do globalne usklajenosti in do novih globalnih lastnosti.

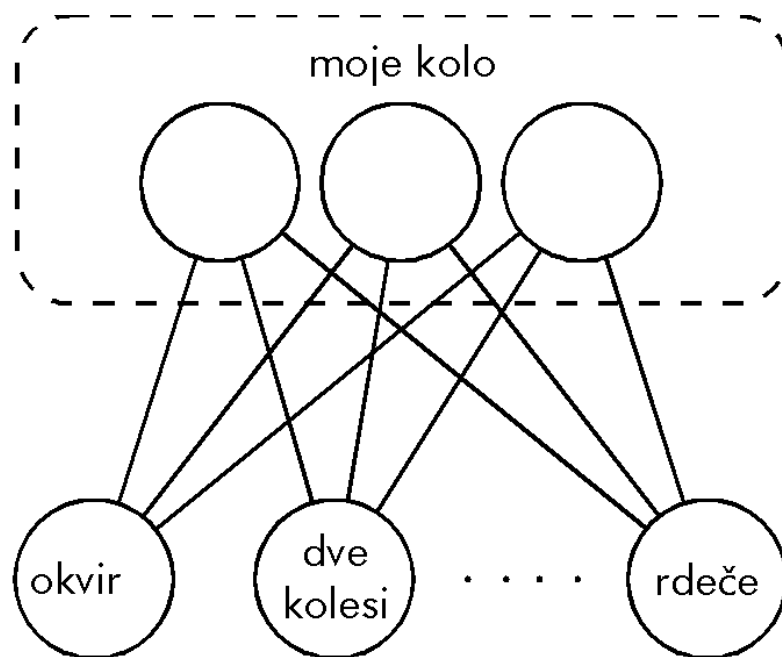
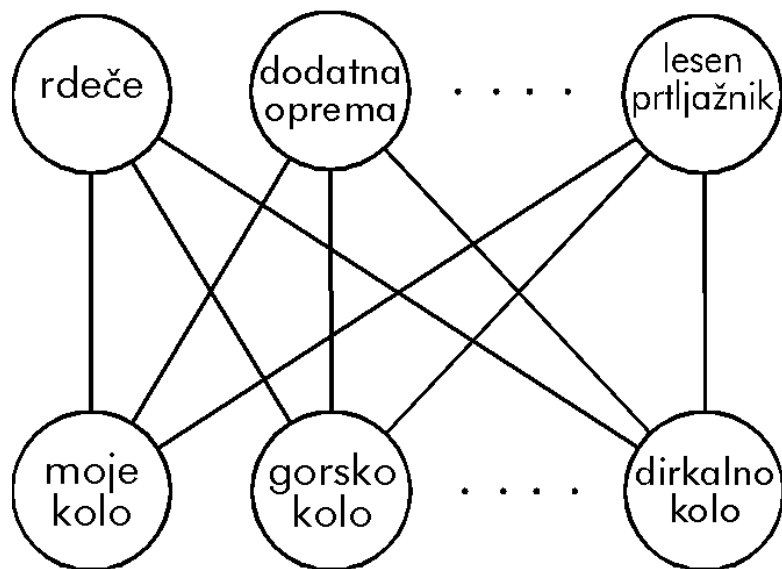
Reprezentacije

- Kadar načrtujemo konekcionistično mrežo kot kognitivni model, moramo na nek način predstaviti pojme, ki so pomembni za to področje.
- lokalne reprezentacije - vsaki enoti pripišemo en pojem)
- porazdeležene reprezentacije - reprezentacija vsakega pojma je porazdeljena po več enotah

Lokalna reprezentacija



Porazdeljene reprezentacije



1. vsaka enota predstavlja nekakšen mikro-atribut reprezentacije - podsimboli. Atributi vhodne reprezentacije so izbrani a priori in jih ne moremo spreminjati.
2. porazdeljena reprezentacija v skritih enotah, ki niso vnaprej določene in vsaka enota odgovarja na enega ali več atributov v vhodnih reprezentacijah. Enote predstavljajo entitete, ki jih le težko opišemo v naravnem jeziku. Šele vzorcu kot celoti lahko pripišemo pomen. Take reprezentacije predstavljajo prilagoditev sistema na zahteve okolja in so neposredna posledica učenja.

- V konekcionističnih modelih s skritimi enotami so reprezentacije posledica učenja. Mreža med procesom učenja sama “odkrije” reprezentacije skozi izbor in kombinacije predhodno definiranih vhodnih vzorcev.
- običajno vprašanje glede reprezentacij postavi na glavo: ”Namesto spraševanja, kaj je reprezentacija za skodelico ali stol, želi konekcionista vedeti, pod kakšnimi pogoji se bo sistem naučil koristno reprezentacijo za skodelico ali stol.”

(Hanson in Burr (1990, str. 478)

Razlogi v prid konekcionističnim modelom

- sistem medsebojno povezanih enot (nevronska mreža)
 - zvezno spreminjanje vrednosti
 - izhod je nelinearna funkcija vhodnih podatkov
 - kompleksnost povezav
- vzporedno delovanje (omogoča hitro izvajanje osnovnih operacij - pravilo 100 korakov)
- plastičnost - povezave se spreminjajo
- nenadzorovano učenje
- vsebinsko naslavljanje spomina (asociativni spomin)
- postopno slabšanje izvedbe ob poškodbah manjšega dela enot (robustnost)
- modeli določenih poškodb možganov ustrezajo vedenju ljudi s takimi poškodbami (npr. disleksija)

Pravilo stotih korakov

- Nevroni, katerih osnovna komputacijska hitrost je nekaj milisekund, morajo omogočiti kompleksno vedenje, ki se izvede v nekaj sto milisekundah. To pomeni, da je **CELOTNO KOMPLEKSNO VEDENJE IZVEDENO V MANJ KOT STO ČASOVNIH KORAKIH** (Feldman in Ballard 1982)
- "Pravilo stotih korakov" predstavlja omejitev pri konstrukciji kognitivne teorije. Če je za kompleksno vedenje, npr. prepoznavanje besede, potrebnih nekaj sto milisekund, teorija pa za isto nalogo zahteva več milijonov časovnih korakov, potem taka teorija ni nevrološko verodostojna. Boljša je teorija, ki lahko pojasni izbrani kognitivni pojav tako, da hkrati upošteva omejitev stotih časovnih korakov.
-

Novi teoretski pojmi

- zadovoljevanje mehkih pogojev
- posplošitev
- porazdeljene reprezentacije
- samo-organizacija

- ***Zadovoljevanje mehkih pogojev*** je način, kako mreža brez uporabe pravil poišče odgovor na več različnih pogojev, ki silijo mrežo vsak v svojo smer.
- Primer: odločanje - soočeni z nasprotujočimi si željami, med katerimi moramo izbrati.
 - model lokalistične interaktivne mreže (nelinearni dinamični sistem), kjer je vsaka želja predstavljena kot “mehek pogoj” med eno izmed enot, ki predstavljajo različne aspekte situacije, in eno od enot, ki predstavljajo možne akcije. Ko mreži predstavimo novo situacijo, ji pustimo, da se ustali v stabilnem stanju. To stanje predstavlja najboljšo zadovoljitev pogojev, čeprav ne bodo zadovoljene prav vse želje. Uporaba “mehkih pogojev” konekcionističnemu sistemu omogoča tekmovanje med željami, ne da bi moral natančno določiti pravila, ki odločajo o tekmovanju.

- ***Posploševanje***, je posledica tesne povezanosti učenja in reprezentacij znanja
- Uteži so nosilec dolgotrajnih sprememb v mreži, ki mreži omogočajo procesiranje različnih vhodnih vzorcev tako, kot se je mreža naučila. Znanje mreže je porazdeljeno v povezavah med enotami in shranjeno v utežeh.
- Ugodna posledica tako shranjenega znanja je avtomatsko posploševanje. Model, ki se je naučil določene naloge, bo na nov primer s tega področja odgovoril v skladu z znanjem, ki je implicitno shranjeno v povezavah. Novi vhodni vzorec bo uporabil obstoječe vezi in z njihovo pomočjo sestavil nov odgovor.

- **Robustnost**

- Poškodba nekaj enot ali vezi v splošnem še ne pomeni resnejše ovire za delovanje sistema. To je lastnosti, ki jo lahko najdemo tudi pri človeku, ne pa pri klasičnem modelu, saj tam izguba elementa pomeni izgubo celotne informacije, ki jo je ta element nosil.

- Omenjene značilnosti konekcionističnih sistemov so ugodne predvsem za modeliranje spoznavnih procesov, ki temeljijo na prepoznavanju vzorcev, za učenje s pomočjo primerov, za učenje veščin, torej povsod tam, kjer ne moremo natančno podati pravil in opisati postopka. Posledica tega je, da je namesto reševanja problemov in logičnega sklepanja paradigmatško raziskovanje postalo prepoznavanje vzorcev in učenje.

Razlogi proti

- veliko dejstev o delovanju celice in neurotransmiterjev, ki jih modeli ne zajamejo
- priklic iz spomina je pri človeku hitrejši, kot nakazujejo modeli
- učenje s pravilom “backpropagation”
- enote in vezi v modelu niso prostorsko umeščene
- modeli so izjemno majhni v primerjavi z dejanskimi nevronskimi mrežami.

Konekcijonizem

```
graph TD; A[Konekcijonizem] --- B[preučevanje delovanja možganov  
nevroznanost]; A --- C[preučevanje vedenja psihologija  
(kognitivna raven)]; A --- D[preučevanje teorije komputacije računalništvo];
```

preučevanje
delovanja možganov
nevroznanost

preučevanje
vedenja
psihologija
(kognitivna raven)

preučevanje
teorije komputacije
računalništvo

kognitivna teorija

