

Neuronske mreže: Duboke neuronske mreže i duboko učenje

Prof. dr. sc. Sven Lončarić
Doc. dr. sc. Marko Subašić

Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu

http://www.fer.hr/predmet/neumre_b

Zašto VEĆE mreže?

- Veći kapacitet mreže može se postići povećanjem broja neurona
 - Tako je i u živom svijetu
 - Jednostavni organizmi nekoliko stotina neurona u mozgu
 - Čovjek 86.000.000.000 neurona
 - Slon 257.000.000.000 neurona – nije sve u broju neurona
- Bitna je i arhitektura mreže
 - Istraživanje arhitektura NM je i dalje vrlo aktualno (uz algoritme učenja)

Zašto “duboke” mreže?

- Može se pokazati da je efikasnije ako se povećanje ostvari korištenjem više slojeva sa manje neurona nasuprot manje slojeva sa više neurona
- Ljudski mozak ima sličnu strukturu
 - Ne može se govoriti o feed forward slojevima jer postoje i povratne veze
- Ljudski kognitivni procesi često su hijerarhijski organizirani i "duboki"

Je li to nova ideja?

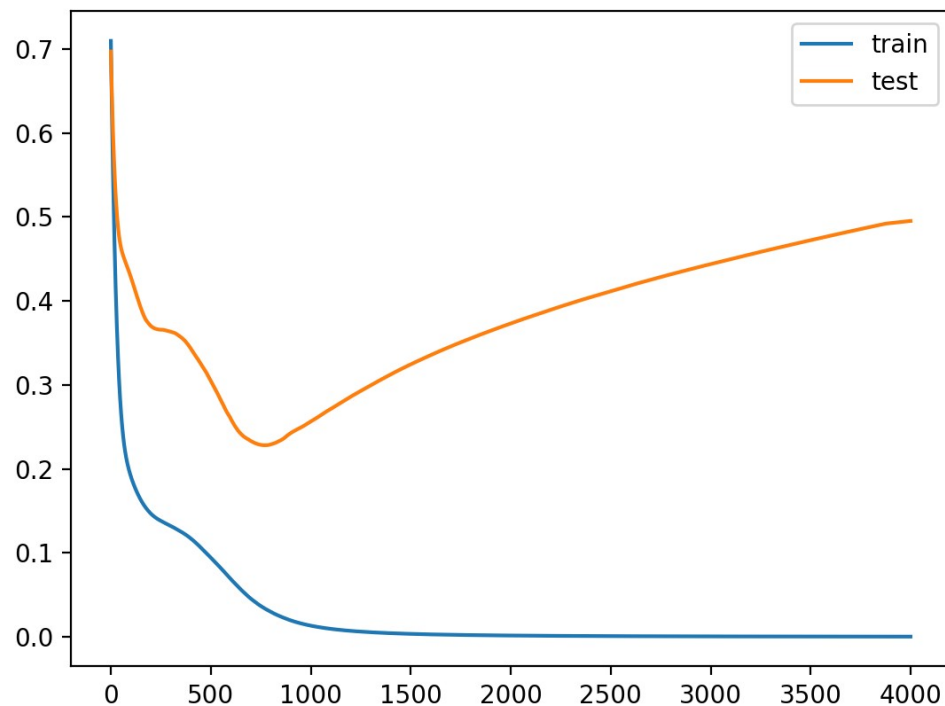
- Višeslojne mreže postojale su i ranije
 - MLP
- Ideja dubokih mreža seže u 1980-te godine
 - Treniranje je bilo presporo
 - U međuvremenu su identificirani razlozi
 - Dio razloga otklonio je napredak hardvera
 - Dio razloga otklonili su novi algoritmi

Je li to nova ideja?

- Koristilo se uglavnom učenje pod nadzorom i povratna propagacija pogreške
 - Kao i danas
- Pokazalo se da su mreže sa većim brojem skrivenih slojeva imale lošije performanse i kod treniranja i kod testiranja
 - Suprotno od očekivanog
 - Koji je razlog?

Stari problemi I

- Pretreniranje – mreža opisuje slučajnu pogrešku ili šum umjesto bitnih odnosa ulaznih podataka



Stari problemi I

- Slaba generalizacija
 - Neće dobro raditi na novim podacima
- Mreža je prekompleksna za dani skup za treniranje
 - Veća kompleksnost mreže potencijalno vodi do veće točnosti
 - Traži se kompromis između kompleksnosti mreže (točnosti) i pretreniranja

Stari problemi I

- Osmišljeni su i neki trikovi za izbjegavanje pretreniranja
 - 1)Early stopping
 - Nije poželjno
 - 2)Smanjenje kompleksnosti mreže
 - Nije poželjno
 - 3)Povećanje količine ulaznih podataka
 - Nije uvijek moguće

Stari problemi I

4) Augmentacija skupa za treniranje

- Umjetno povećanje varijabilnosti u ulaznom skupu
- Dodavanje šuma

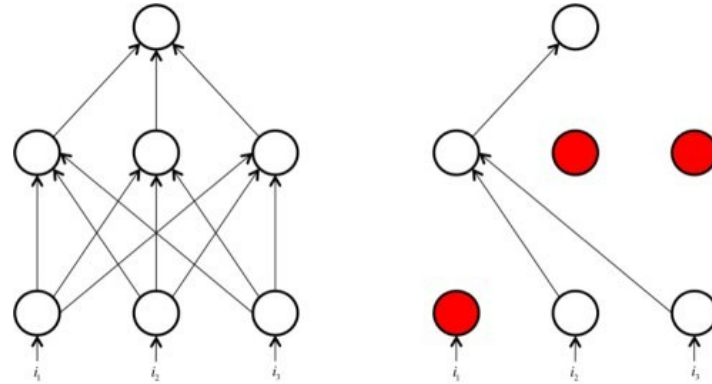
5) Regularizacija

- Dodatni član u funkciji cilja
- Efektivno smanjuje kompleksnost mreže – nadamo se na dobar način
- Npr. L2 regularizacija – kažnjava velike težine
 - Intuitivno – potiče se mreža da podjednako koristi sve ulaze
 - Težine teže prema nuli

Stari problemi I

6) Dropout

- Gašenje pojedinih neurona
- Dodavanje šuma



7) Odabir druge arhitekture

- Koristiti tuđa iskustva i intuiciju
- Pretjerivanje s trikovima obično vodi do loših performansi mreže
 - Koliko je dovoljno?

Stari problemi II

- Backpropagation se temelji na gradijentima
 - oni su ključni za treniranje
 - Propagiranje gradijenata u početne slojeve

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

$$\delta_j(n) = \varphi_j'(v_j(n)) e_j(n)$$

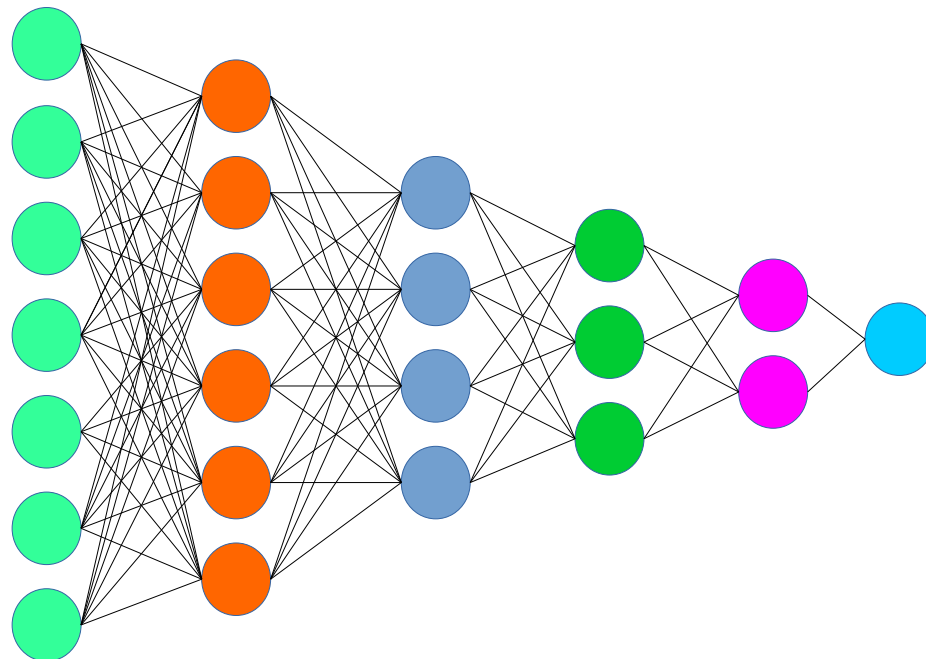
$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

Stari problemi II

- Što želimo na kraju treniranja?
 - Da gradijenti za svaki parametar mreže padnu na 0
 - Tada je mreža kovergirala i daljnja mijenjanja parametara mreže ne minimiziraju funkciju cijene
 - Želimo stati kada je mreža naučila što je trebala – kraj treniranja
- Što ne želimo na početku (ili sredini) treniranja?
 - Da gradijenti za svaki parametar mreže padnu na 0
 - Efektivno zaustavlja treniranje
 - Da gradijenti za svaki parametar mreže bude jako velik
 - Nestabilnost

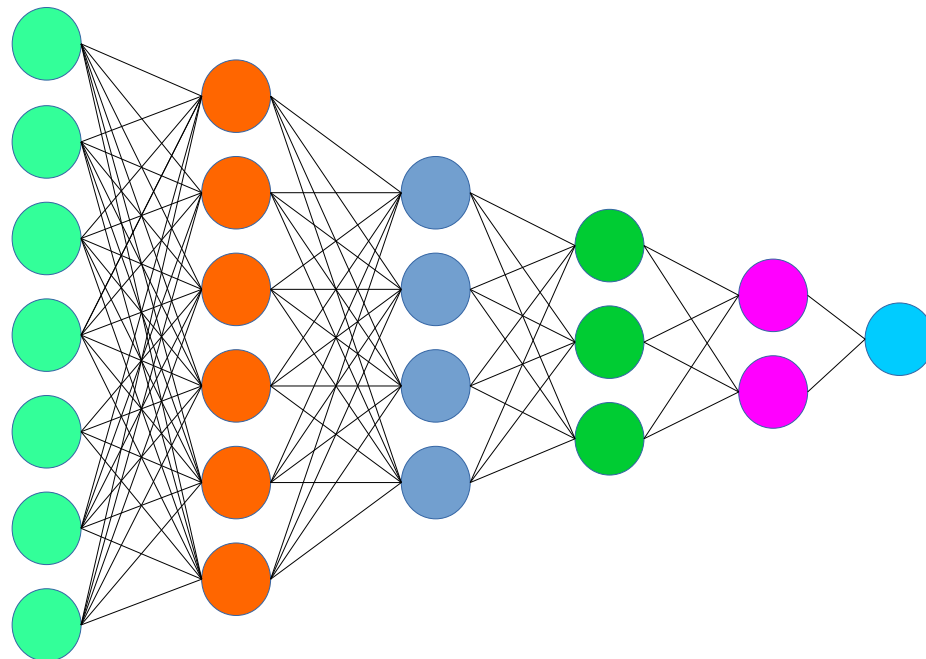
Stari problemi II

- Išcezavajući gradijenti – s povećanjem broja slojeva, povratna propagacija pogreške postaje sve lošija u prosljeđivanju informacija nižim slojevima
- Dublji slojevi blizu ulaznog teško uče
 - Ne propagiraju se gradijenti u početne slojeve
- Problem je značajan kada se koristi slučajna inicijalizacija
- Ne mora nužno značiti kraj treniranja nego se treniranje jako produžuje



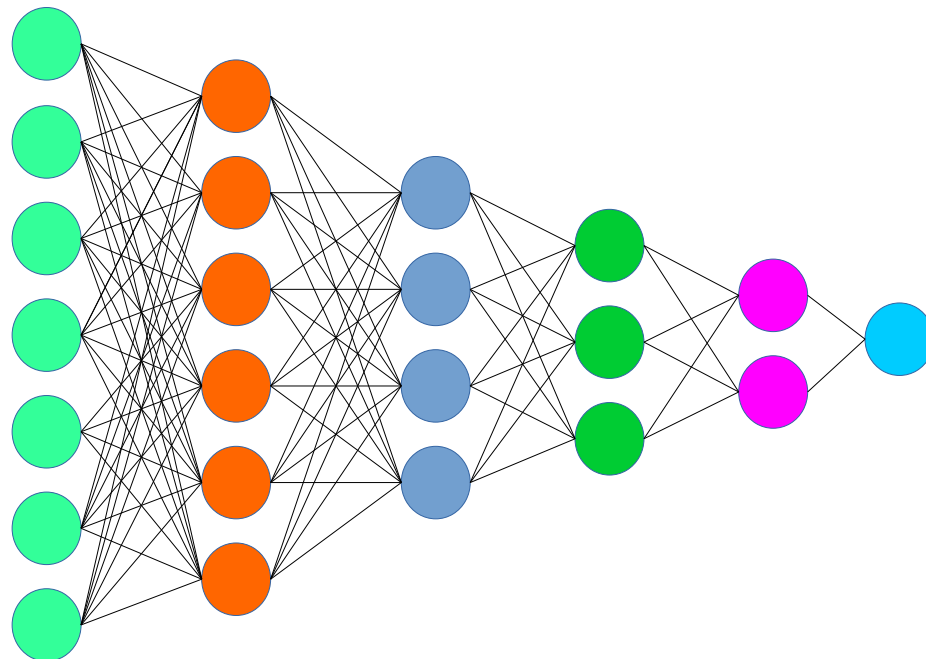
Stari problemi II

- Eksplodirajući gradijenti – korekcija je prevelika pa ne dovodi do smanjenja greške
- Sličan efekt kao i kod iščezavajućih gradijenta
- Bitno nam je da početni slojevi dobro rade



Stari problemi II

- Moguće je da neki od zadnjih skrivenih slojeva duboke mreže ima dovoljno kapaciteta za modeliranje zadanog problema
- Prethodni slojevi tada nisu potrebni i dodaju šum ulaznim podacima
 - Duboka mreža funkcioniра slično kao plitka mreža

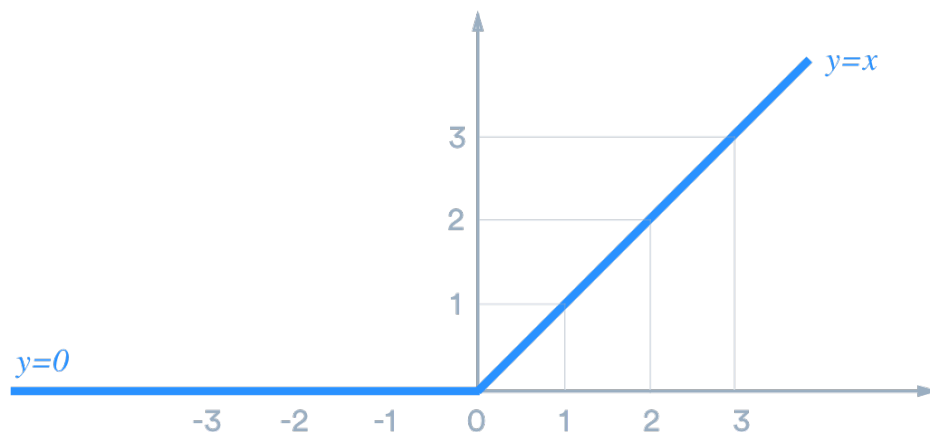


Stari problemi II

- Trikovi osmišljeni za izbjegavanje iščezavajućih gradijenata:
 - Bolja random inicijalizacija parametara mreže
 - Ciljano izbjegavanje malih gradijenata
 - Korištenje predtreniranih slojeva
 - Predtreniranje na drugom problemu
 - Npr. učenjem bez nadzora (autoenkoderi)

Stari problemi II

- Treniranje sloj po sloj
- Drugačije aktivacijske funkcije
 - ReLU



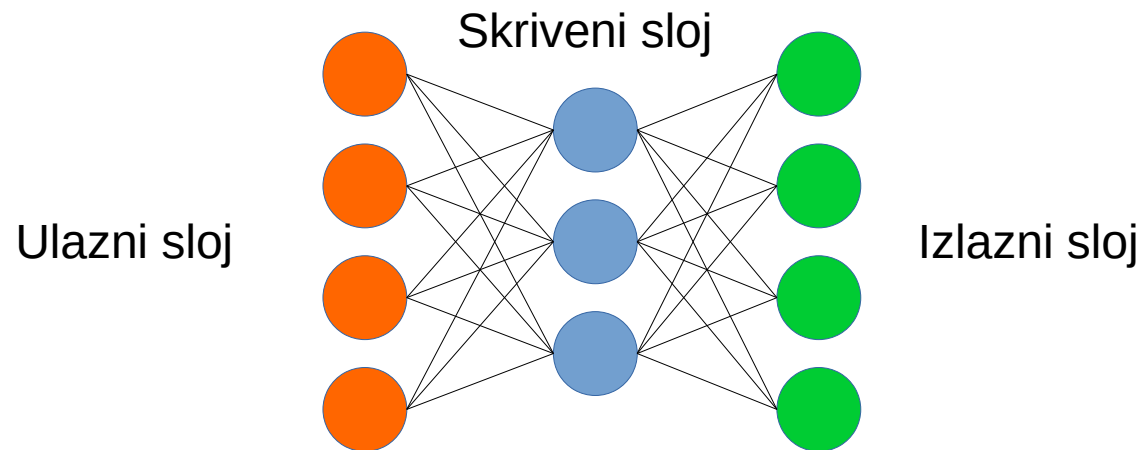
- "Izvor" gradijenata za treniranje bliže ulaznom sloju
 - Npr. kod autoenkodera

Što je još pomoglo

- Jeftina i dostupna računalna snaga iz GPU
- Sve češće situacije u kojima je potrebno efikasno analizirati velike količine sirovih podataka

Autoenkoder

- Feedforward mreža kojom se želi naučiti komprimirati (enkodirati) ulazne vektore



- Informaciju prisutnu u ulaznom vektoru predstaviti pomoću manjeg broja elemenata (čvorova skrivenog sloja) iz kojih je moguća savršena rekonstrukcija svih elemenata ulaznog vektora
- Potrebno ju naučiti kodirati vektore iz skupa za treniranje

Autoenkoder

- Mreža treba naučiti unutarnju strukturu podataka i bitne značajke – bit podataka na ulazu
- Skriveni sloj se naziva detektor značajki
- Broj neurona u skrivenom sloju je manji od broja neurona u ulaznom sloju
- Mreža je prisiljena pronaći samo bitne značajke kako bi postigla redukciju dimenzionalnosti
- Dobar pristup za postizanje generalizacije

Autoenkoder

- Treniranje pod nadzorom – npr. Backpropagation
- Ne radi se o "pravom" nadzoru
 - Ne uči se nepoznata funkcija
 - Provjerava se da li je na izlazu isto što i na ulazu
- U literaturi se takvo treniranje smatra nenadziranim

Autoenkoder

- Postoje brojne varijacije osnovnog autoenkodera
- Fokusirane su na centralni skriven sloj i ekstrakciju bitnih značajki u njemu
 - Mehanizmi regularizacije
 - Sparsitiy, robusnost na šum, nedostajući ulazi, ograničenja na derivacije
 - Veličina sloja je tada manje bitna

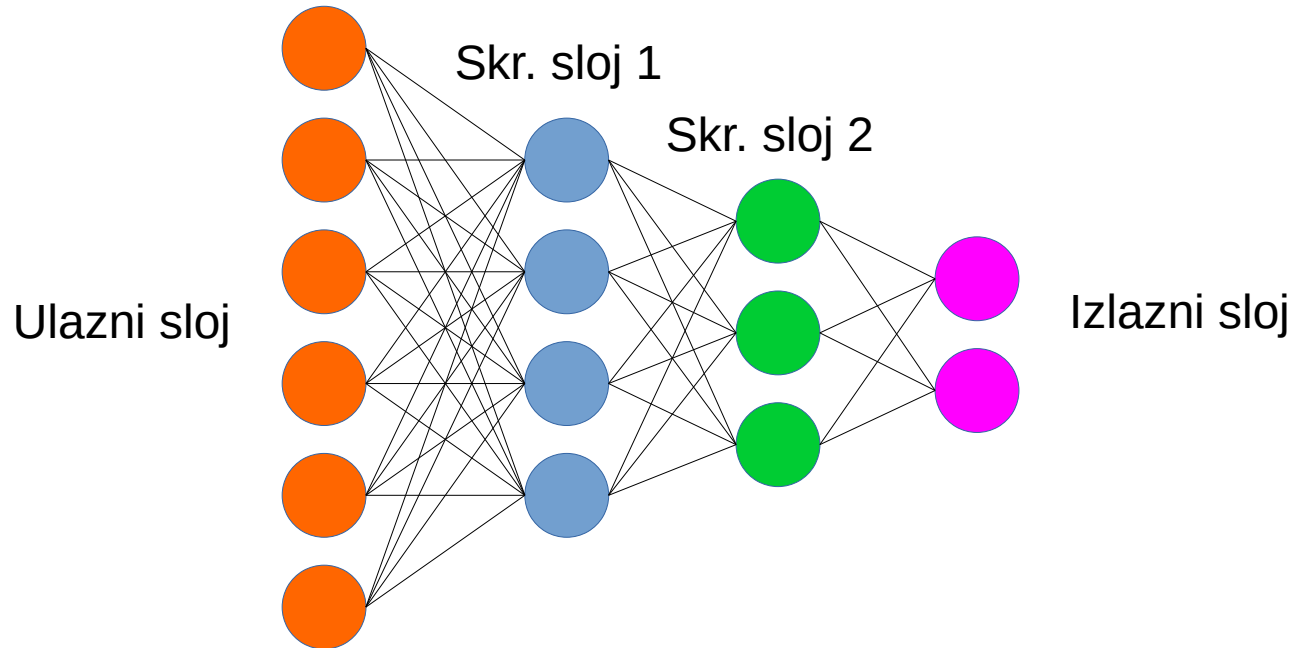
Autoenkoder

- Varijante:
 - Denoising autoenkoder
 - Dodaje se umjetni šum u ulazne podatke
 - Autoenkoder mora naučiti uklanjati šum
 - Contractive autoenkoder
 - Izbjegavanje promjena u skrivenom sloju za male promjene ulaza (mala promjena = šum ?)
 - Dodatni član funkcije cijene temeljen na derivaciji
 - Odnosi se na centralni skriveni sloj
 - Dobro za iščezavajući gradijent

Duboke mreže

- Autoenkoderi mogu funkcionirati kao detektori značajki
- Detektirane značajke su "skrivenne" u skrivenom sloju – nisu direktno upotrebljive
- Spomenute mreže se mogu nastavljati jedna na drugu
 - Pohlepno treniranje sloj po sloj
 - Izostaje problem isčezavajućih gradijenata i pretreniranja

Niz autoenkodera



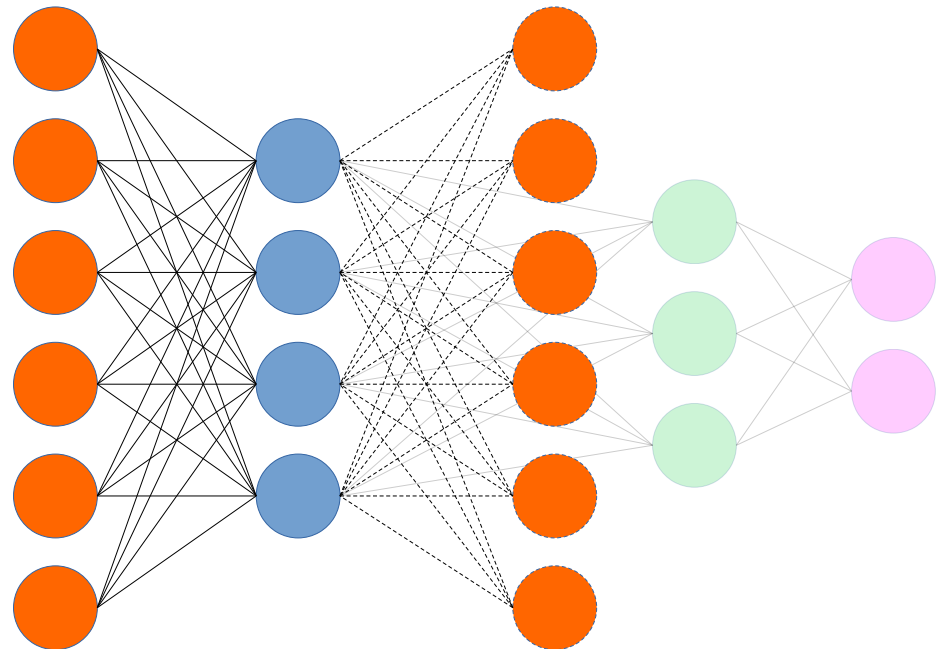
- Ulaz u slijedeći autoenkoder je izlaz iz prethodnog

Niz autoenkodera

- Treniranje

1) Treniranje prvog autoenkodera

- S privremenim izlaznim slojem
- Koristeći sve podatke za treniranje
- Backpropagation algoritam

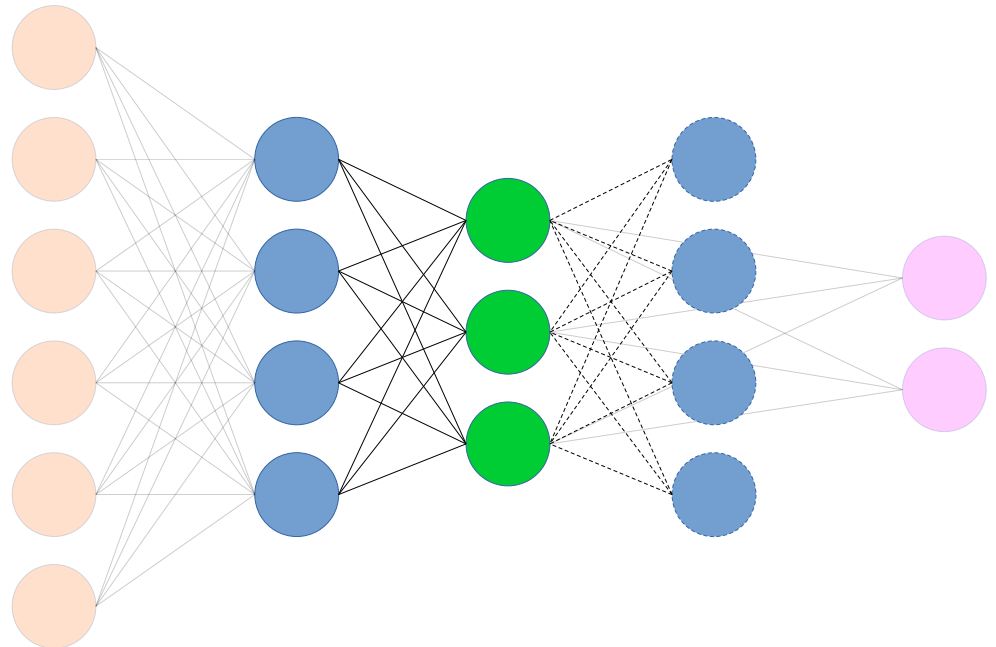


Niz autoenkodera

- Treniranje

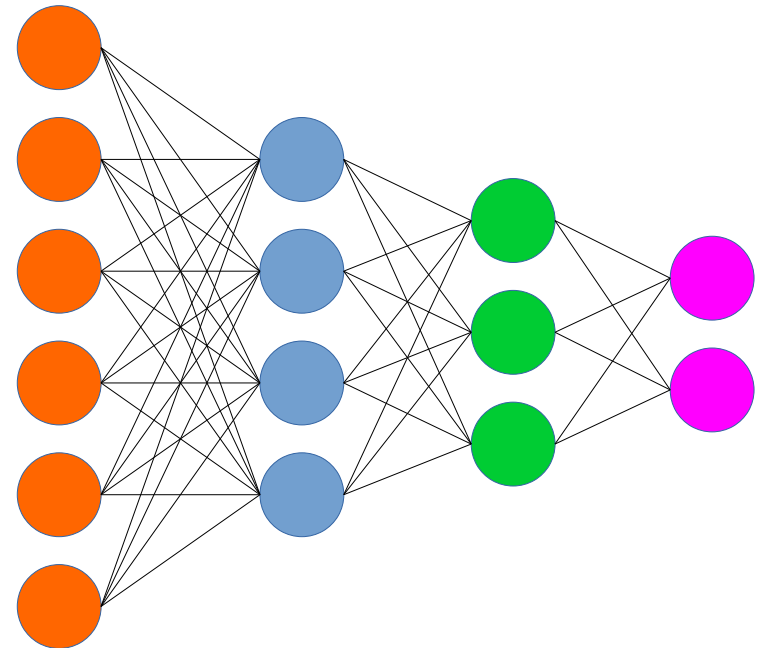
2) Treniranje drugog autoenkodera

- S privremenim izlaznim slojem
- Koristeći sve podatke za treniranje
- Ulaz su izlazi prethodnog istreniranog autoenkodera
- Backpropagation algoritam



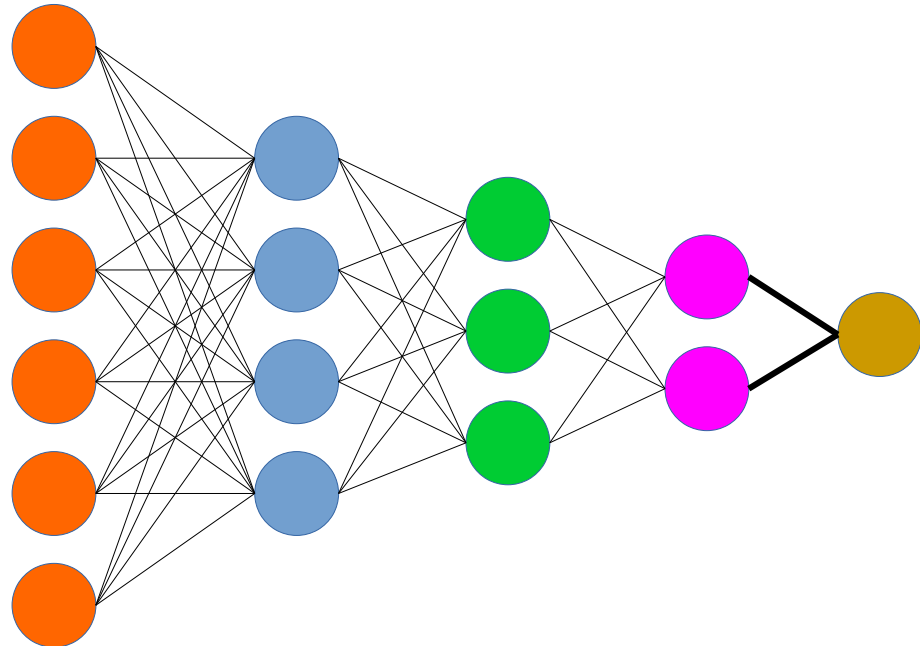
Niz autoenkodera

- Treniranje
 - 3) Ponoviti korak 2) za sve preostale skrivene slojeve / autoenkodere



Niz autoenkodera

- Treniranje
 - 4) Prema potrebi dodaj jedan ili više završnih slojeva

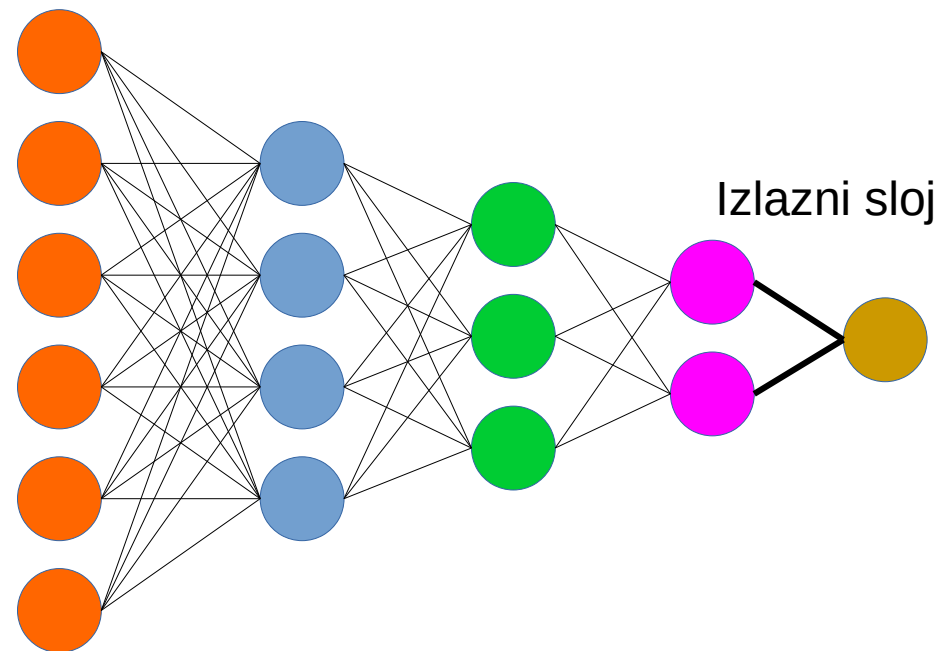


Niz autoenkodera

- Treniranje

5) Treniraj konačni MLP backpropagation algoritmom

- Ovaj korak uključuje modifikaciju svih težina u mreži
- Sve težine dobivene u prethodnim koracima postaju inicijalne težine za konačni MLP
- Nužan korak da bi se značajke naučene u prethodnim koracima povezale sa željenim izlazom mreže
- *Fine-tuning*
- Koraci 1-3 su "predtreniranje"

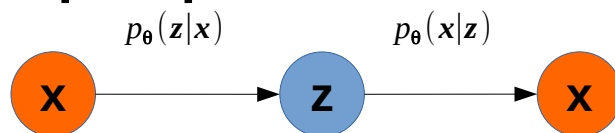


Niz autoenkodera

- Iskorištava se poznata činjenica (problem) da konačan rezultat backpropagation treniranja ovisi o inicijalizaciji
- Prvi sloj uči značajke niske razine
 - Npr. rubove u slici
- Drugi sloj uči složenije značajke
 - Npr. kombinacije (uzorke) nižih značajki
- Viši slojevi – više značajke

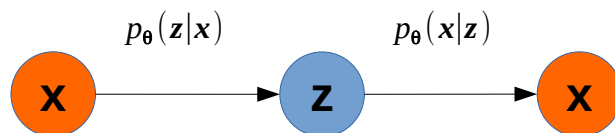
Varijacijski autoenkoder

- Kompleksnija ideja
- Cilj je naučiti distribuciju podataka iz skupa za treniranje $p(\mathbf{x})$
- Nakon toga se mogu generirati uzorci prema toj distribuciji
- Može biti dubok
- Iako je mreža stohastička, za treniranje se koristi back-prop



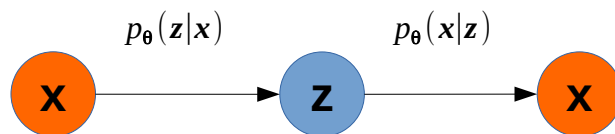
Varijacijski autoenkoder

- Skriveni sloj čini slučajni vektor zadane distribucije u koje se mapiraju uzorci za treniranje
- Iz skrivenog sloja se određuju distribucije izlaznih elemenata na temelju kojih se generiraju izlazni uzorci
- Formalno, cilj je maksimizirati vjerojatnosti uzoraka za treniranje



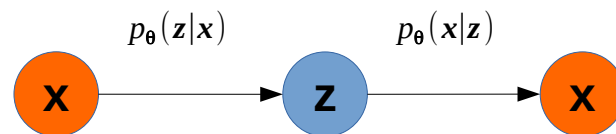
Varijacijski autoenkoder

- Funkcija cilja ima regularizacijski član koji se odnosi na skriveni sloj
 - Slično kao i contractive autoenkoder
 - Omogućava efikasnije treniranje i izbjegavanje iščezavajućeg gradijenta
 - Minimizacija KL divergencije
- Skriveni sloj je stohastički
 - Dodavanje šuma u mrežu
 - Slično kao i denoising autoenkoder



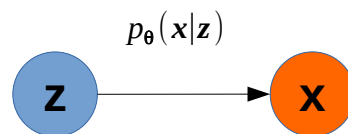
Varijacijski autoenkoder

- Algoritam nema hiperparametara
- Iako je mreža stohastička, uspješno se koristi back-prop



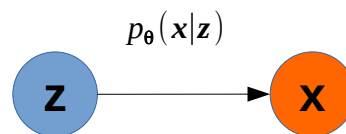
Varijacijski autoenkoder

- Takav sustav omogućuje generiranje novih, nikad viđenih uzoraka
 - Generativni model
 - Dekoderski dio kojem se na ulaz dovede slučajni uzorak iz zadane distribucije



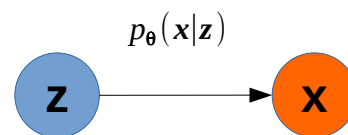
Varijacijski autoenkoder

- Primjeri



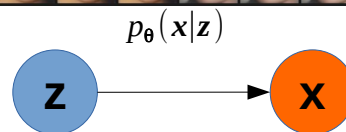
Varijacijski autoenkoder

- Primjeri



Varijacijski autoenkoder

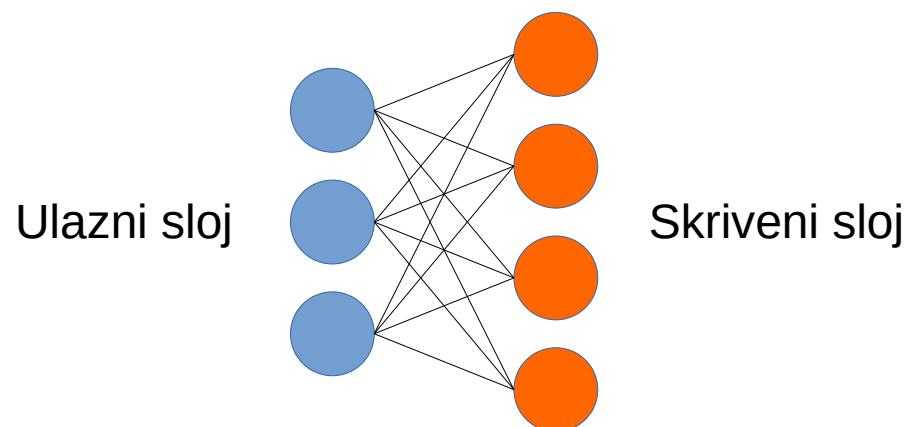
- Pojedini elementi iz skrivenog sloja mogu predstavljati pojedinu karakteristiku izlaza
- Varijacijom tog parametra modulira se prisutnost te karakteristike



Ograničeni Boltzmannov stroj

- Restricted Boltzmann Machine (RBM)
- Neuronska mreža koja može naučiti raspodjelu vjerojatnosti u skupu za učenje
- Mreža je stohastička!

Ograničeni Boltzmannov stroj



- Jedan ulazni i jedan skriveni sloj
- Povezivanje neurona u istom sloju nije dozvoljeno kao kod Boltzmanovog stroja – ograničenje
- Povezanost između slojeva je potpuna
- Veze između neurona su dvosmjerne i simetrične

Ograničeni Boltzmannov stroj

- Binarni neuroni
- Način rada:

1) Odredi aktivacijsku energiju

$$a_i = \sum_j w_{ij} u_j$$

2) Postavi s_i u 1 s vjerojatnosti

$$p_{1i} = \frac{1}{1 + e^{-a_i}}$$

odnosno 0 s vjerojatnosti

$$p_{0i} = 1 - p_{1i}$$

Contrastive divergence

- Pozitivna faza
 - Postavi uzorak na ulaz u
 - Odredi odziv skrivenog sloja s
- Negativna faza
 - Odredi odziv ulaznog sloja u' s obzirom na odziv skrivenog sloja s
 - Odredi odziv skrivenog sloja s' s obzirom na ulaz u'
- Korekcija težina

$$\Delta w_{ij} = \eta (u_i s_j^T - u'_i s'^T_j)$$

Contrastive divergence

- Korekcija težina

$$\Delta w = \eta (u_i s_j^T - u'_i s'^T_j)$$

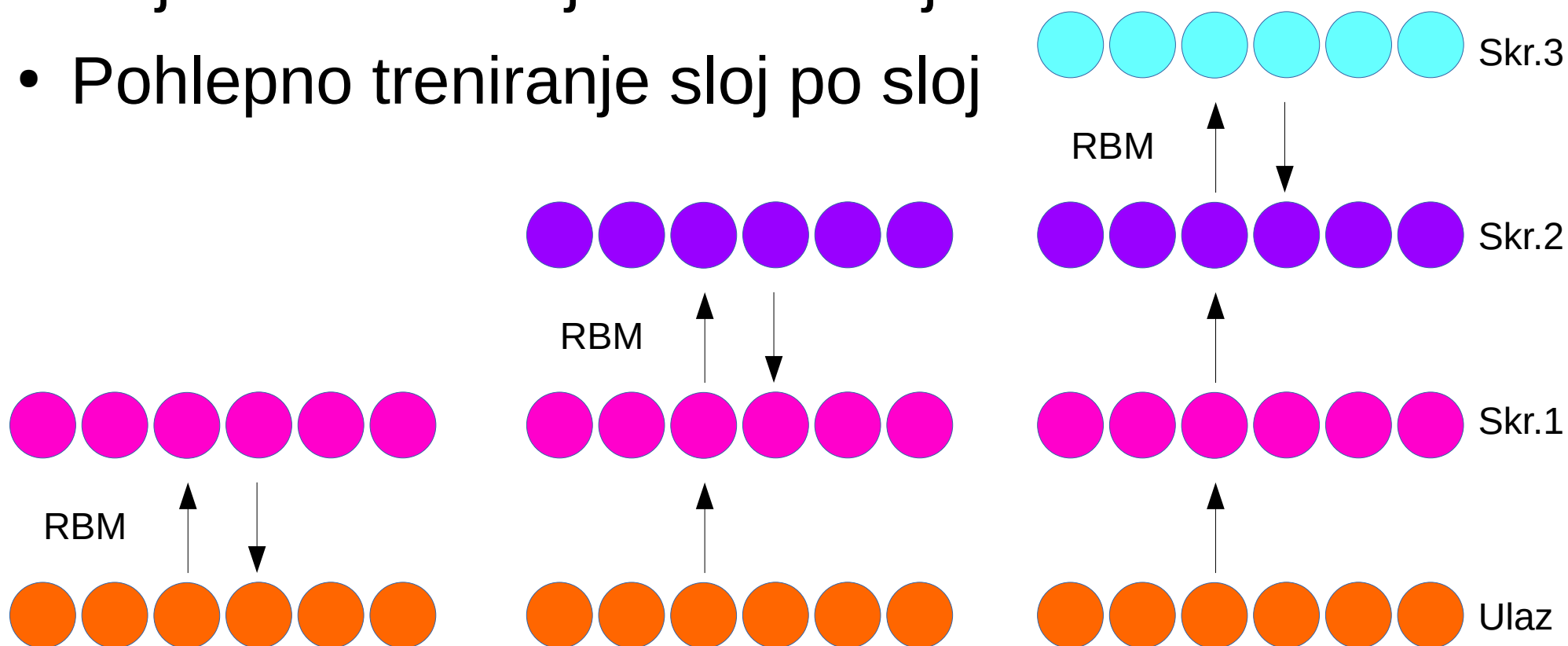
- U pozitivnoj fazi mreža određuje svoju reprezentaciju (s) ulaznog podatka (u)
- U negativnoj fazi mreža određuje rekonstrukciju ulaznog podatka (u') na temelju svoje reprezentacije (s)
- Cilj je postići sličnost između u i u'
- Mreža se korigira dok ne postigne cilj

Duboke mreže

- RBMs mogu funkcionirati kao detektori značajki
- Detektirane značajke su "skrivenene" u skrivenom sloju – nisu direktno upotrebljive
- Spomenute mreže se mogu nastavljati jedna na drugu
 - Pohlepno treniranje sloj po sloj
 - Izostaje problem isčezavajućih gradijenata i pretreniranja

Deep Belief Networks

- Nizanje Boltzmanovih strojeva
- Skriveni sloj prethodnog RBM-a postaje ulazni sloj za RBM u slijedećem sloju
- Pohlepno treniranje sloj po sloj

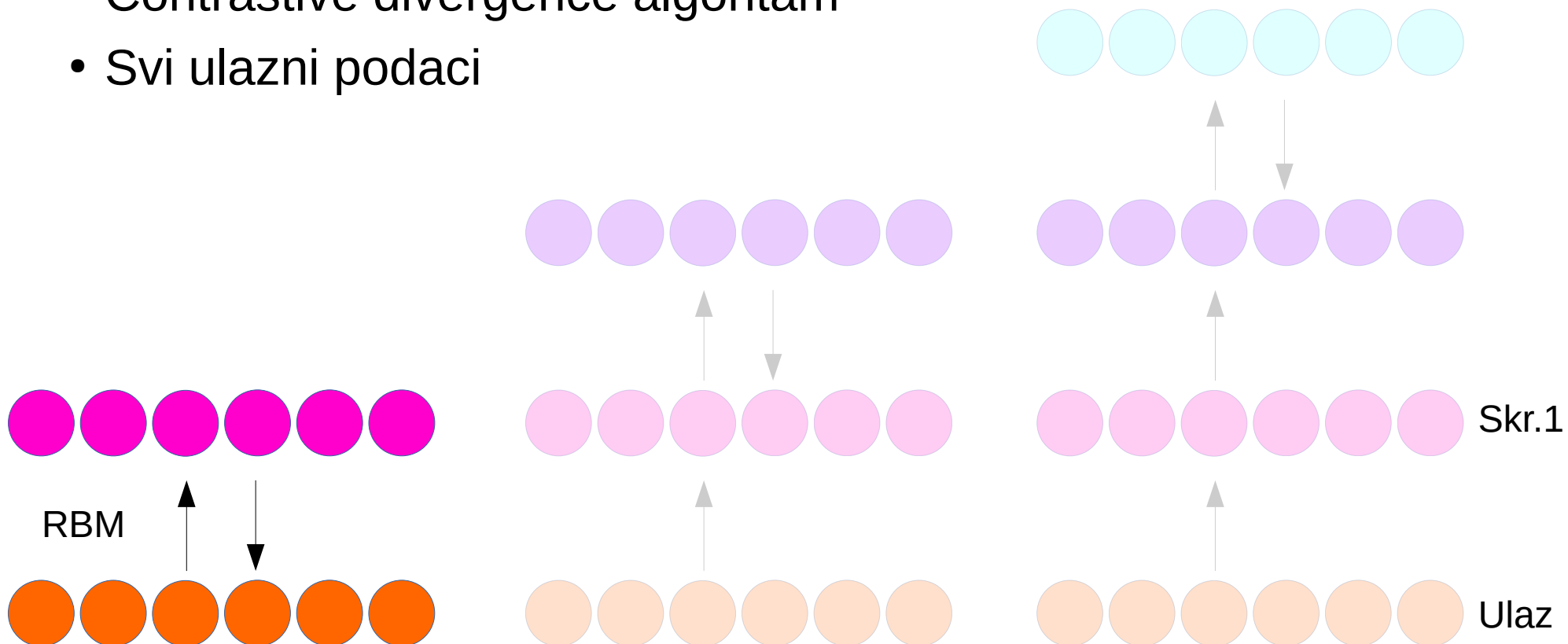


Deep Belief Networks

- Treniranje

1) Treniraj prvi RBM

- Contrastive divergence algoritam
- Svi ulazni podaci

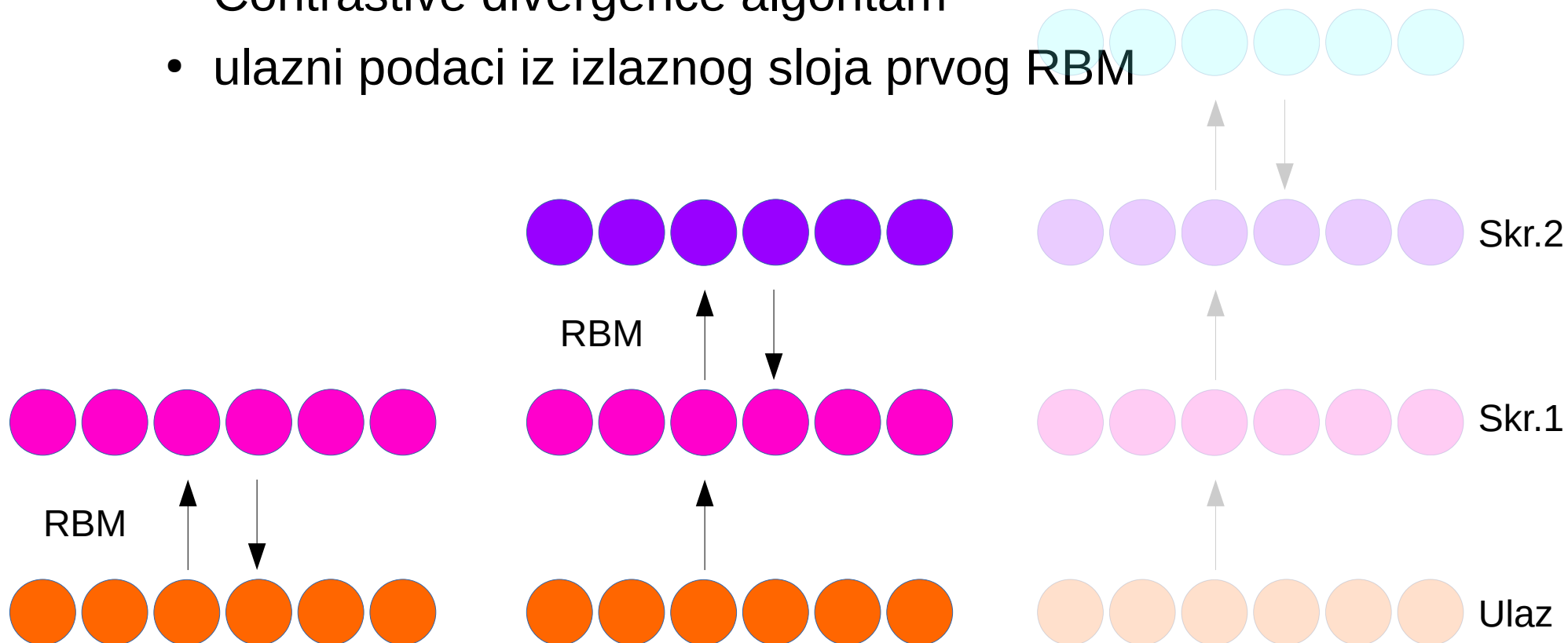


Deep Belief Networks

- Treniranje

2) Treniraj drugi RBM

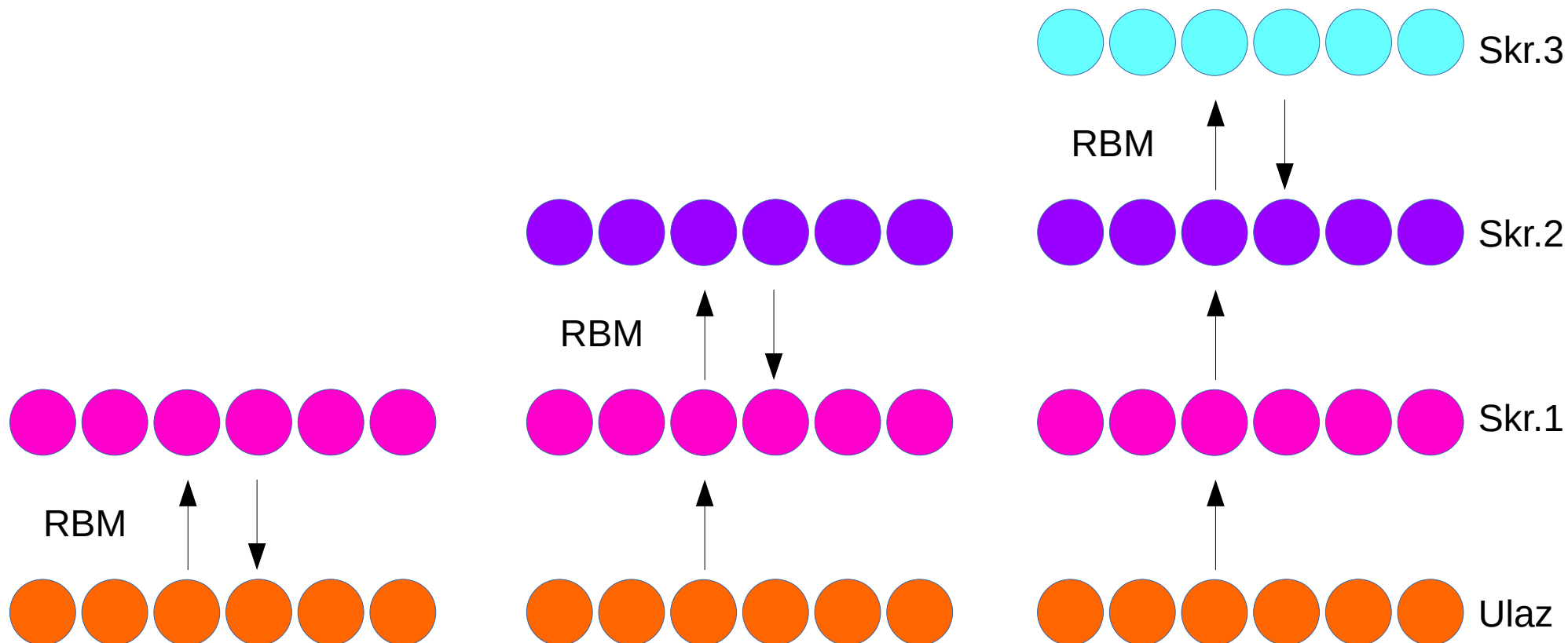
- Contrastive divergence algoritam
- ulazni podaci iz izlaznog sloja prvog RBM



Deep Belief Networks

- Treniranje

3) Ponovi postupak za sve preostale RBM-ove

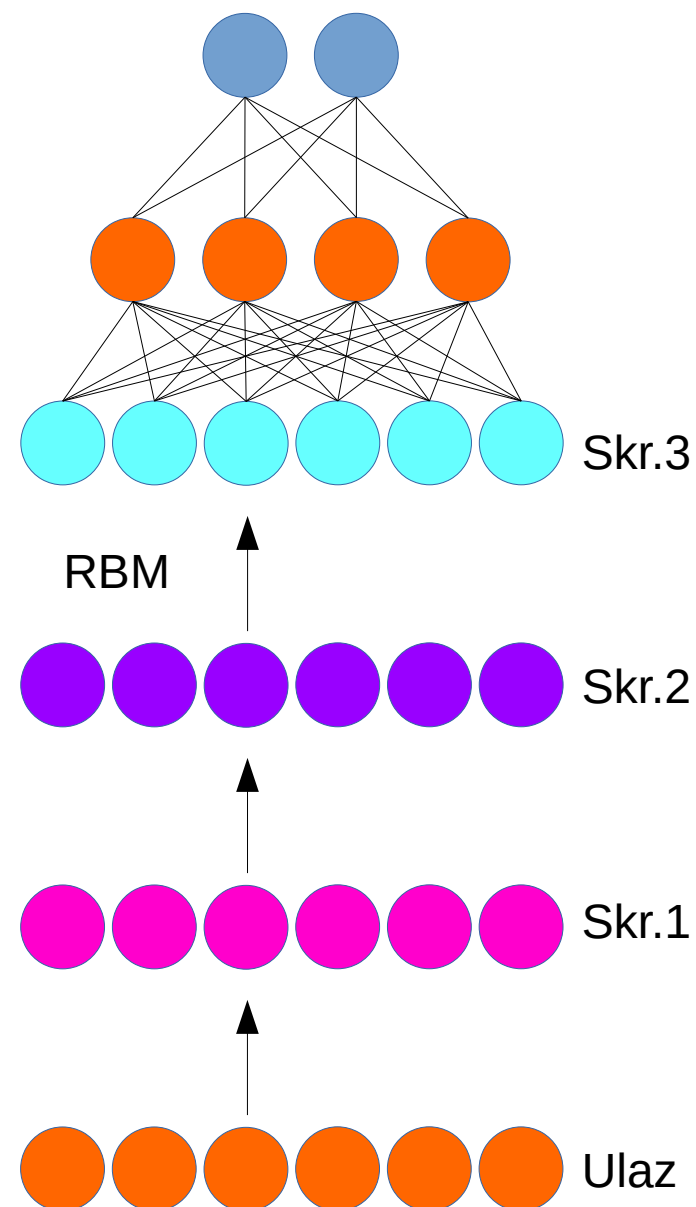


Deep Belief Networks

- Treniranje

4) Postavi MLP na zadnji RBM u nizu

- Treniranje backpropagation algoritmom
- Uspostavlja vezu naučenih značajki s željenim izlazima mreže



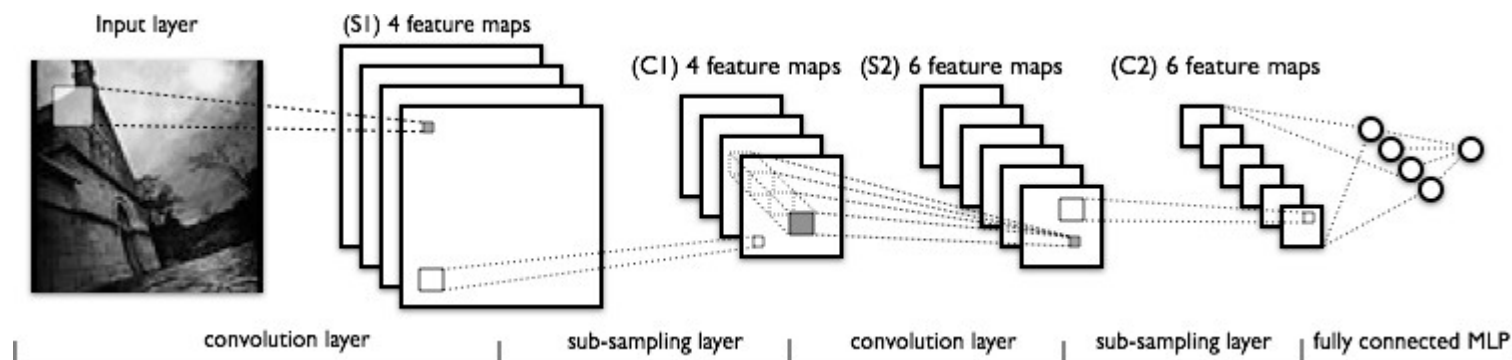
Deep Belief Networks

- Kroz slojeve se uče značajke od značajki

Pohlepno učenje

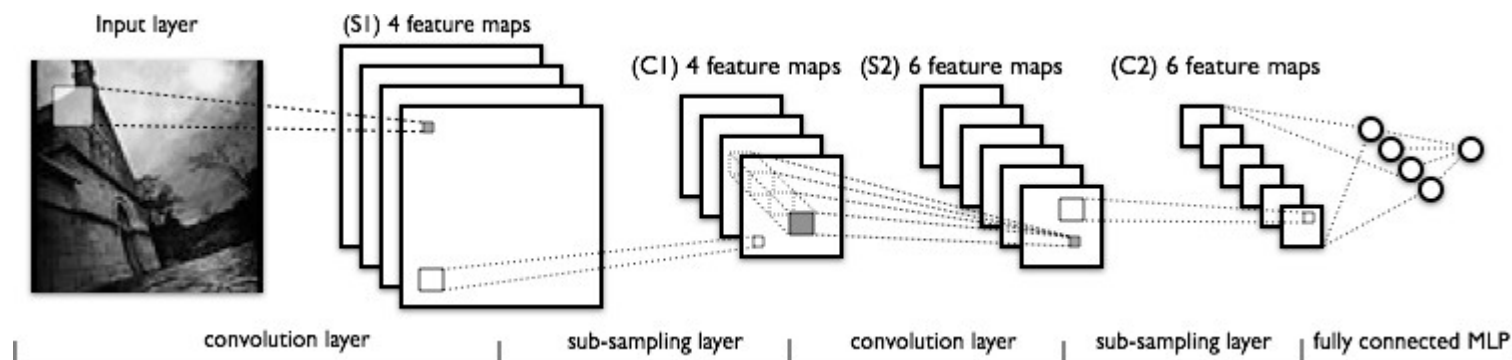
- Sloj po sloj
- Može biti s nadzorom ili **bez nadzora**
- Bez nadzora -> iskorištavanje neoznačenih podataka
 - Dostupniji su
 - Kombiniranje veće količine neoznačenih podataka za inicijalizaciju s manjom količinom označenih podataka za fine-tuning
- Bolji pronalazak globalnog minimuma
 - Zahvaljujući dobroj inicijalizaciji

Konvolucijske mreže



- Varijanta MLP-a
- Temelji se na filtriranju slike - konvoluciji
- Sve težine treniraju se prilagođenim backpropagation algoritmom

Konvolucijske mreže



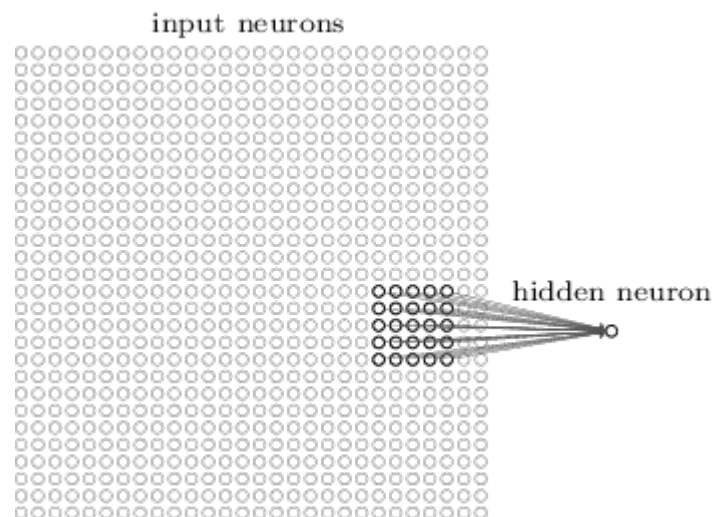
- Mreža je deterministička
- Zadnji dio mreže može biti klasični MLP (bez konvolucije)
 - Uspostavlja vezu između naučenih mapi značajki i željenih izlaza

Konvolucijske mreže

- Pogodna za analizu slika – detekciju objekata
 - Slike su stacionarne
 - Značajka koja se pojavi na jednoj lokaciji u slici, jednako je vjerojatna i na drugim lokacijama
 - Neosjetljivost na translaciju u slici
 - Logično je agregirati informacije o takvim značajkama

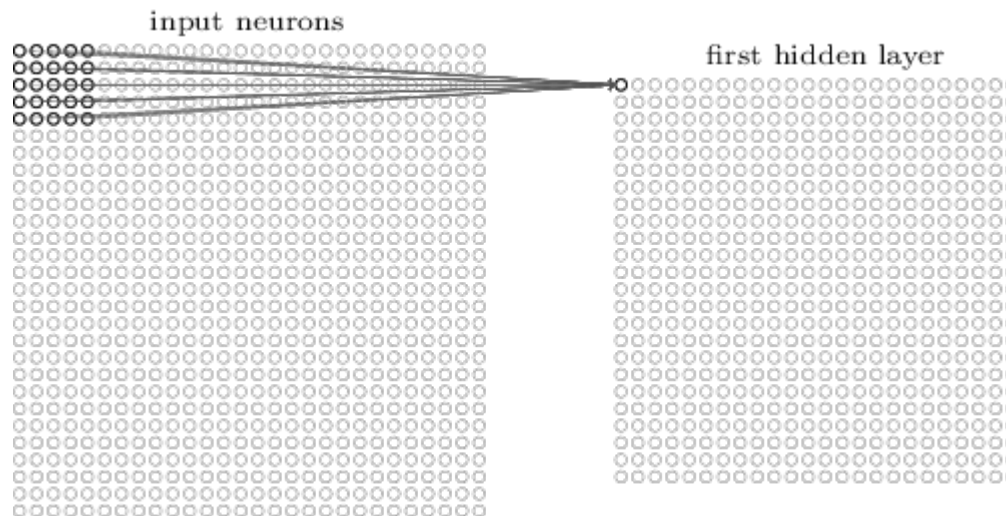
Konvolucijske mreže

- Konvolucijski slojevi
 - Primjenjuje se određeni broj filtera - neurona
 - Omogućuje traženje različitih značajki
 - Svaki filter je zadužen za jednu "značajku"
 - Lokalno receptivno polje



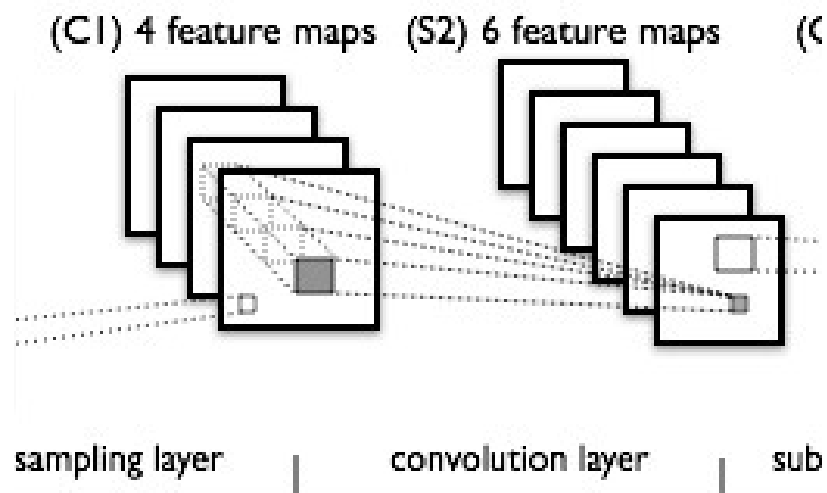
Konvolucijske mreže

- Konvolucijski slojevi
 - Rezultat filtriranja svakim filtrom se naziva mapa značajki (*feature map*)



Konvolucijske mreže

- Konvolucijski slojevi
 - Svaki se filter primjenjuje na svim mapama značajki prethodnog sloja ali uz iste/slične težine
 - Omogućuje pronalazak značajki bez obzira u kojoj mapi značajki se nalaze

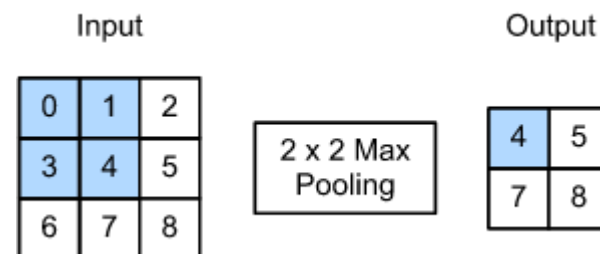
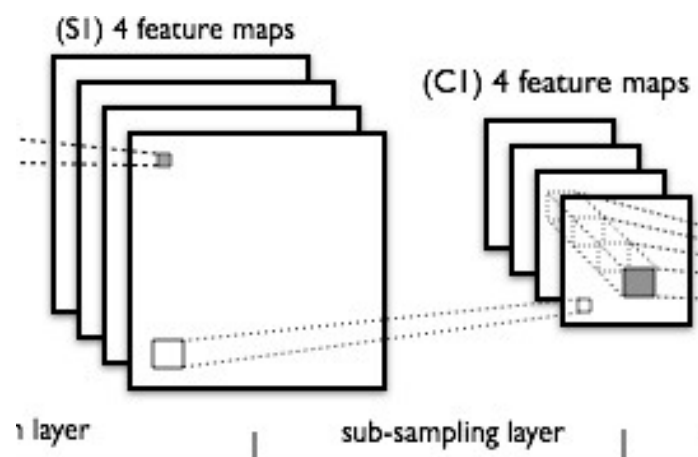


Konvolucijske mreže

- Konvolucijski slojevi
 - Pozicija značajke nije bitna – bitan je međusobni prostorni raspored značajki
 - Neovisnost o translaciji
 - Ograničena robusnost na rotaciju

Konvolucijske mreže

- Podotipkavajući sloj
 - Smanjuje veličinu ulaza
 - Umanjuje utjecaj pomaka značajki u prostoru i drugih distorzija
 - Različite procedure
 - Max pooling
 - Usrednjavanje
 - Stohastičko uzorkovanje
 - Trenirani neuron



Konvolucijske mreže - backprop

- Moguće su i razne druge varijacije i dodatci
- Nužno je samo da funkcionira povratna propagacija pogreške
 - Računanje parcijalnih derivacija izlaza s obzirom na svaki ulaz u sloju
- Kod konvolucijskih slojeva možemo govoriti o dijeljenim težinama koje se onda korigiraju odjednom
 - Prosječna korekcija

$$\Delta w_i^{uk} = \frac{1}{N} \sum_n \Delta w_i^n$$

Konvolucijske mreže - backprop

- Obično se treniranje provodi u mini grupama (mini batch)
 - Provodi se radi što veće paralelizacije i ubrzavanja treniranja
 - Glavno ograničenje je količina GPU memorije

$$\Delta w_i^b = \frac{1}{N} \sum_m \Delta w_i^{uk}(m)$$

Konvolucijske mreže - problem

- Puno konvolucijskih slojeva – duboka mreža
 - Problem iščezavajućeg gradijnta i dalje postoji
 - Problem pretreniranja i dalje postoji

Konvolucijske mreže - trikovi

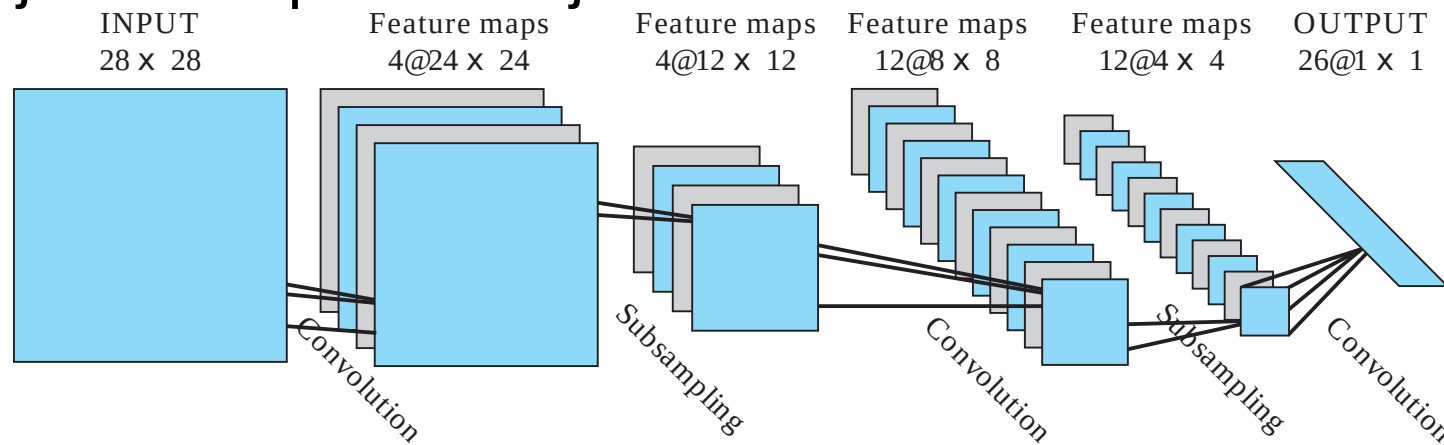
- Jedno rješenje je korištenje predtreniranih slojeva
 - Transfer learning
 - Postoje popularni predtrenirani konvolucijski slojevi
 - Trenirani na velikoj količini slika za neki problem analize slike
 - najčešće klasifikacija slika
 - I oni se po želji mogu trenirati za konkretan problem, ali će zbog iščezavajućeg gradijenta promjena biti mala
 - Fine-tuning

Konvolucijske mreže - trikovi

- Još jedno rješenje je dropout
 - Slučajno isključivanje pojedinih neurona
 - Koristi se samo kod treniranja
 - Tjera mrežu da bude redundantna
 - Uvodi se stohastika u deterministički proces treniranja

Primjer konvolucijske mreže

- Primjena - raspoznavanje slova



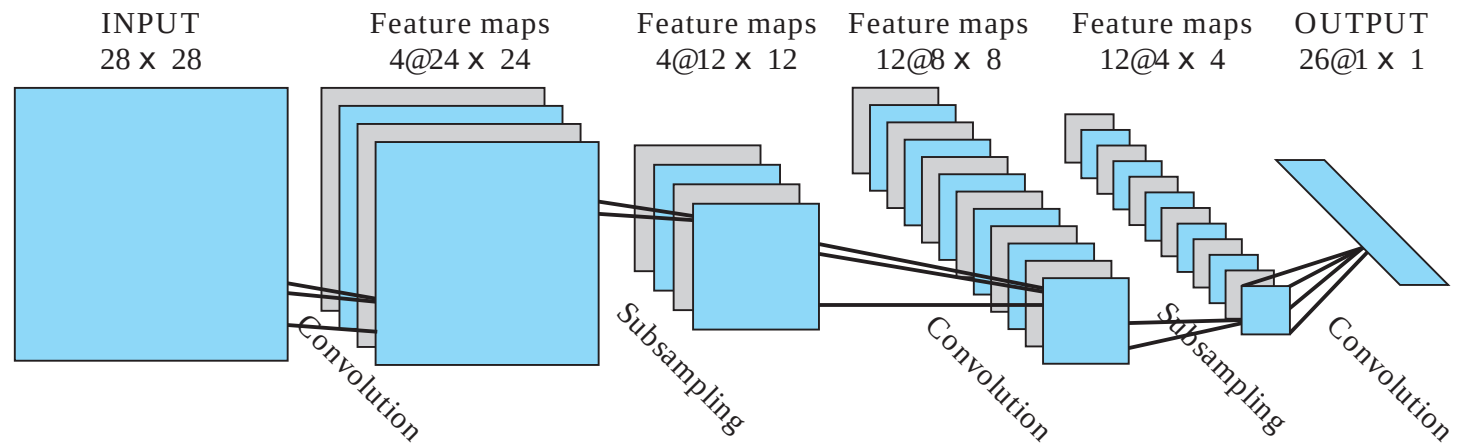
1. skriveni sloj provodi konvoluciju

- Rezultat su četiri mape značajki sa 24x24 neurona
- Svaki neuron ima receptivno polje 5x5

2. skriveni sloj provodi podotipkavanje i lokalno usrednjavanje

- Mape značajki 12x12
- Receptivana polja neurona 2x2
- Svaki neuron ima težinu/e, posmak i nelinearnu aktivacijsku funkciju

Primjer konvolucijske mreže



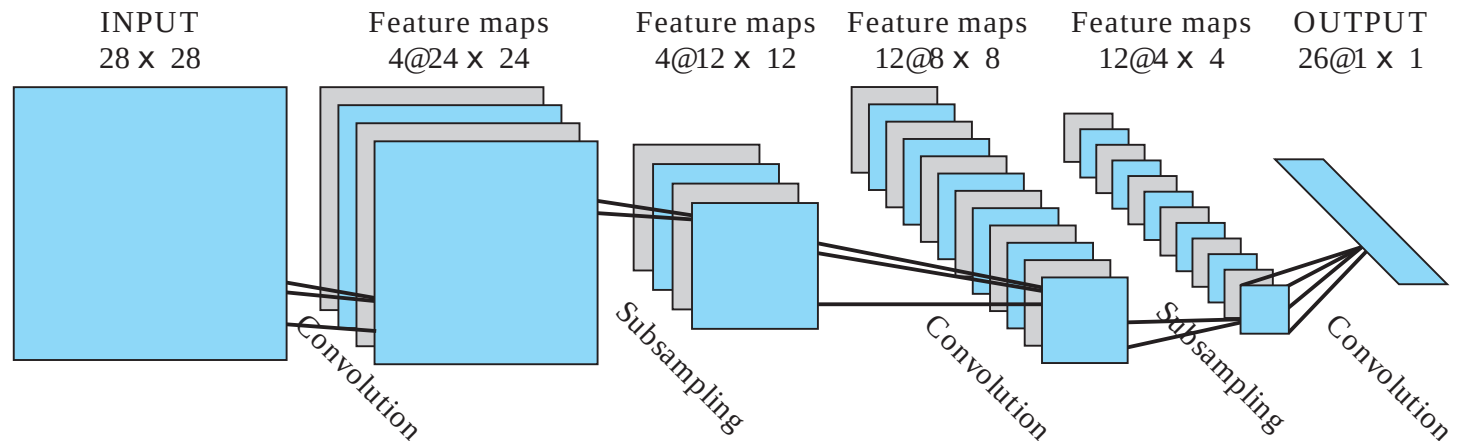
3. skriveni sloj provodi konvoluciju

- Rezultat su četiri mape značajki sa 8×8 neurona
- Svaki neuron težinski je povezan sa svim mapama značajki iz prethodnog sloja

4. skriveni sloj provodi podotipkavanje i lokalno usrednjavanje

- Mape značajki 4×4
- Receptivana polja neurona 2×2

Primjer konvolucijske mreže



5. Izlazni sloj

- ima 26 neurona – svaki za jedno slovo
- Receptivno polje svakog neurona je 4x4 po svim mapama značajki iz prethodnog sloja
- Prostorna rezolucija se smanjuje dok se broj mapa značajki povećava
 - Inspirirano vizualnim korteksom mačke
- Svaki element mape značajki je jedan neuron no svi neuroni u jednom sloju dijele isti skup težina koje se treniraju
- Redukcija broja parametara koji se treniraju
 - Ograničavanje kapaciteta mreže
 - Bolja generalizacija
- Paralelizacija!

Konvolucijske mreže – trikovi

- Korišćenje isključivo konvolucijskih slojeva
 - Zadnja mapa značajki postaje mapa rješenja
 - Konačno rješenje dobijemo usrednjavanjem
 - Average pooling
 - Prednosti
 - Veća robusnost na pomak
 - Ne ovisi o veličini slike

Konvolucijske mreže

- Znatno manji broj parametara u odnosu na MLP
- Broj filtara u konvolucijskom sloju
 - Znatno utječe na vrijeme izvođenja i treniranja
 - Manje filtara u slojevima bližim ulaznom sloju
 - Mape značajki su veće
 - Recept: ujednačit broj računskih operacija među slojevima – umnožak broja piksela/neurona i filtara u sloju konstantan

Konvolucijske mreže - izbori

- Veličina filtara
 - Ovisno o skupu podataka
- Podotipkavanje
 - Skala
 - Procedura (ako se ne trenira)
- Faktori učenja
 - Smanjivati ih sa svakom epohom?
- Predprocesiranje
 - PCA?

Konvolucijske mreže

- Konačni rezultat je MLP čija veličina nije pretjerana
- Lokalna povezanost neurona
 - Izbjegnuta je puna povezanost mreže
- Dizajn mreže je prilagođen problemu koji se nastoji riješiti

Sažetak

- Duboke mreže - Veći broj slojeva sa nelinearnim neuronima (npr. veći od 10)
- Učenje sa ili bez nadzora
- Hijerarhijska struktura: od jednostavnijih ka složenijim značajkama
- Više slojeva – veća razina apstrakcije
- U nekim zadacima dosegle točnost ljudskih eksperata
- Učestala kritika – koristi se kao crna kutija