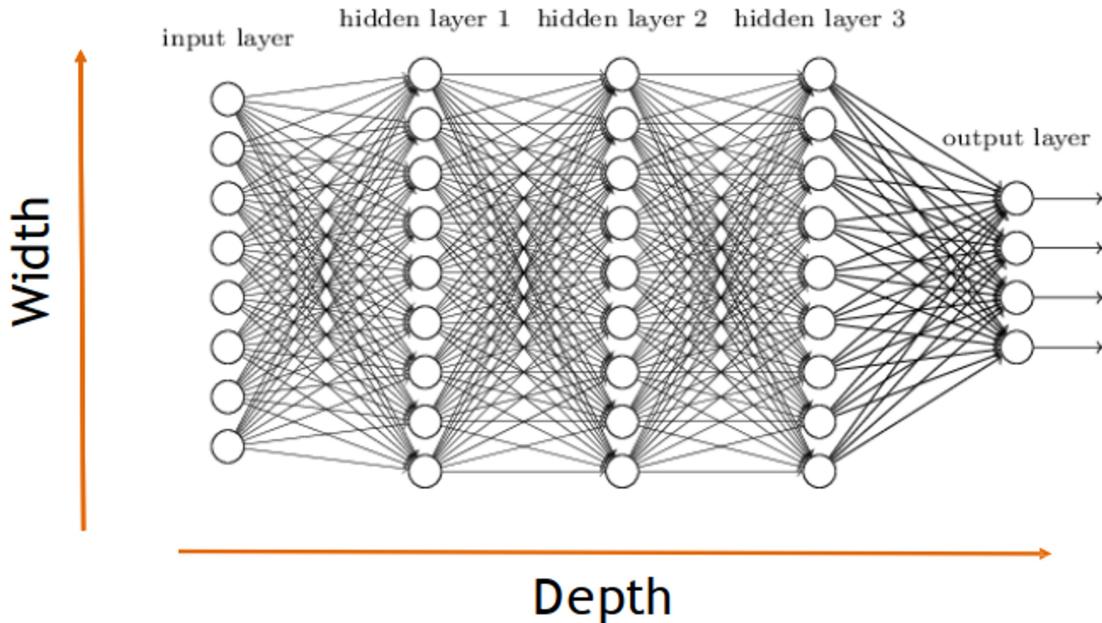
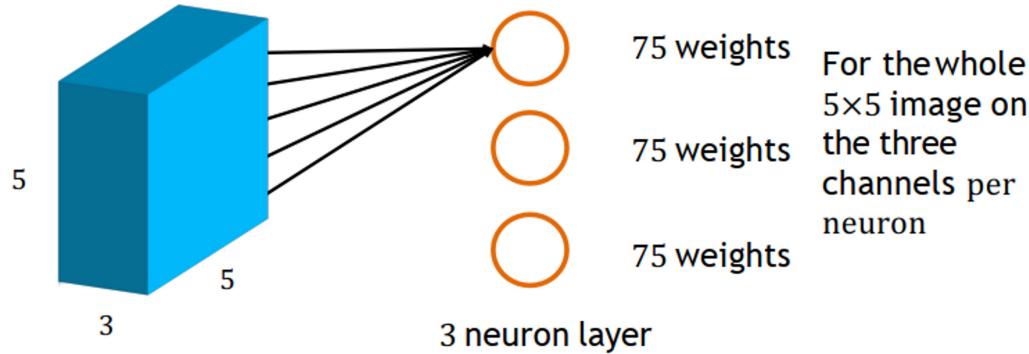


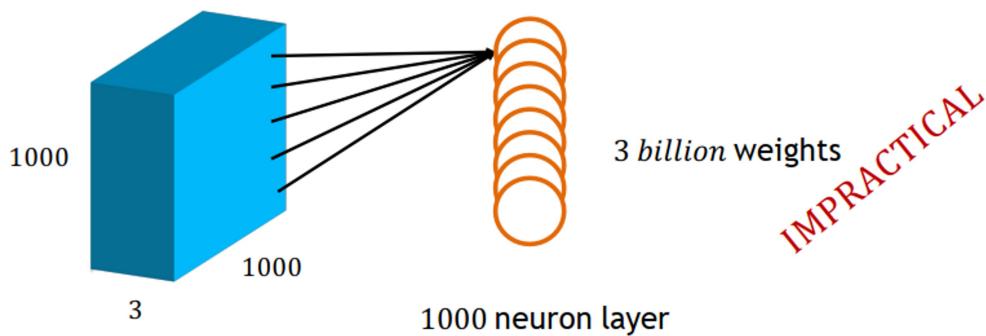
# Konvolucione mreze



- na slici je prikazana **potpuno povezana** (FC - *fully connected*) neuralna mreza - svaki neuron u jednom sloju je povezan sa svakim neuronom u sledecem sloju
- problem sa ovakvima mrezama:
  - procesiranje male slike pomocu FC mreze:



- 5x5 je broj piksela, a 3 nivoa (kanala) predstavljaju RGB komponente boja
- svaki neuron ce imati  $5 \times 5 \times 3$  tezina
- ukoliko se posmatra neka realna velicina slike ( $1000 \times 1000$ ) za **samo jedan sloj** koji ima 1000 neurona potreban je **ogroman broj tezina** (3 milijarde)



- resenje: deljenje pojacanja, koriscenje istih pojacanja za razlicite delove slike - konvolucija

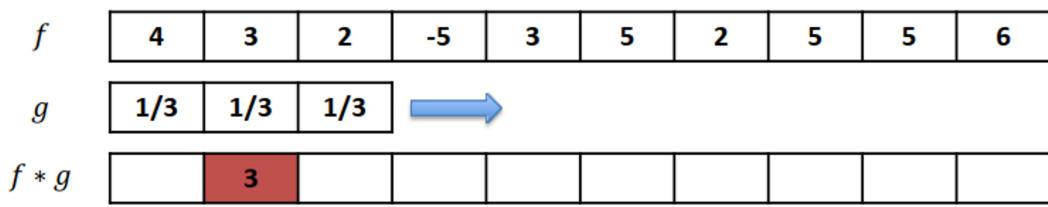
## Konvolucija

**Konvolucija** signala u opstem slučaju predstavlja matematičku operaciju za kombinovanje dva signala kako bi se dobio treci (jedan signal se propusti kroz drugi, pa se nadje integral toga).

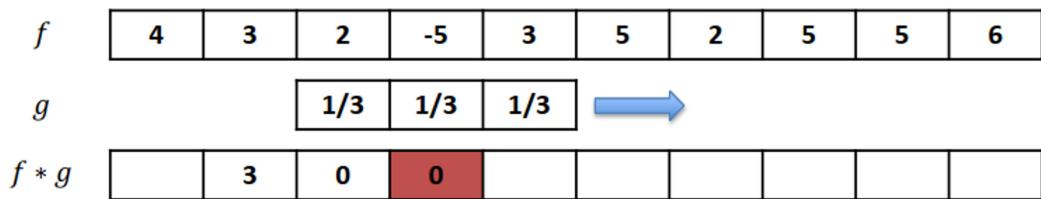
Konvolucija je diskretna ako se primenjuje na 2 diskretna signala.

Primer:

- g funkcija je **slide filter** (kernel), koji se primenjuje na ulazni signal f tako da se filter pomera, tj. klizi kroz ulazni prostor

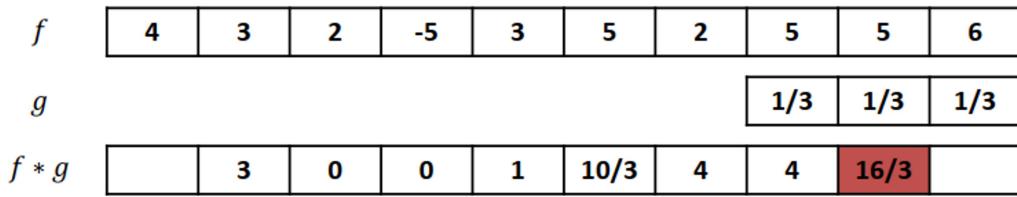


$$4 \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} = 3$$



$$2 \cdot \frac{1}{3} + (-5) \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} = 0$$

- ovo množenje i sabiranje dobijenih signala se ponavlja do kraja, kada se dobija ovakav signal:



$$5 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 6 \cdot \frac{1}{3} = \frac{16}{3}$$

Konvolucija u neuralnim mrežama odnosi se na matematičku operaciju **za primenu filtera** na sliku ili skup podataka i igra ključnu ulogu u **konvolucionim neuralnim mrežama (CNN)**.

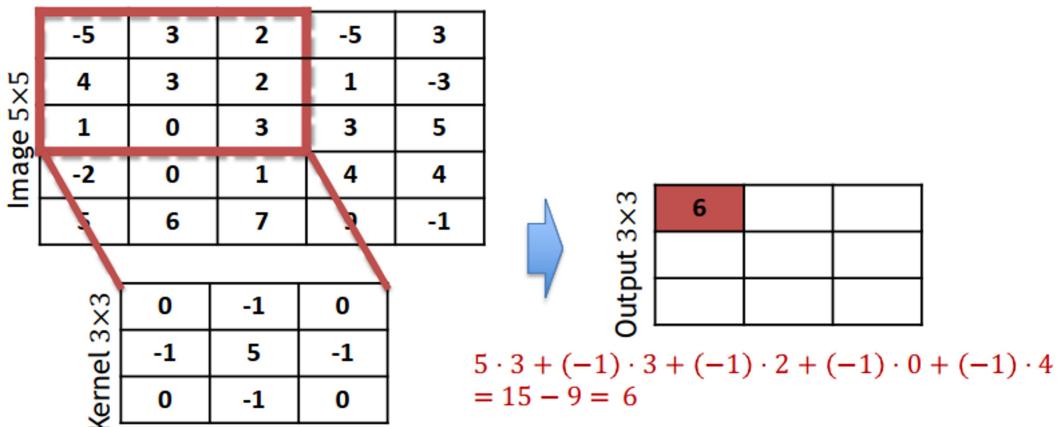


Na slici levo prikazana je FC neuralna mreža, a na slici desno konvolucionala. Konvolucionne neuralne mreže se zasnivaju na konvoluciji sa filtrima koji su **manjih dimenzija od ulaznih podataka**, pa neuroni u prethodnom sloju utiču na mnogo manji deo sledećeg sloja nego u FC mrezi.

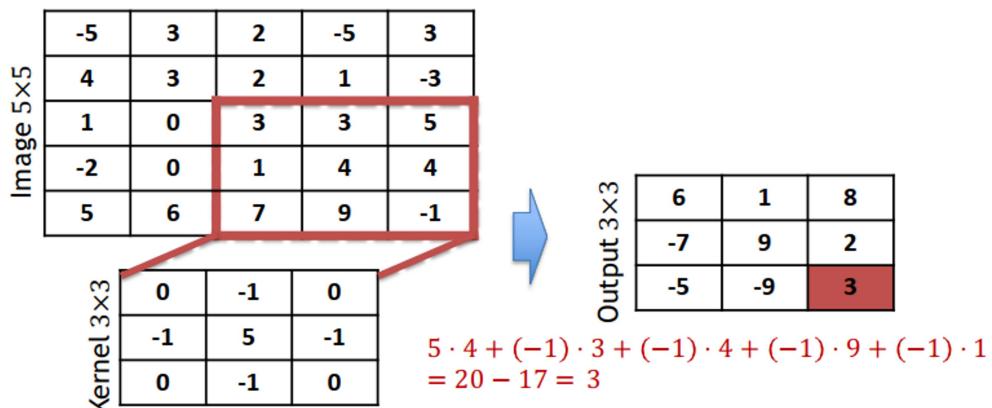
Konvolucione mreze u *computer vision*-u:

- **klasifikacija** - koji je objekat (pas ili macka)
- **lokalizacija** - gde se tacno na slici nalazi taj objekat (ivice oko objekata)
- **detekcija** - kada ima vise razlicitih objekata da se pronadje gde je koji
- **segmentacija** - da se objekat u potpunosti uokviri (kao da se *crop*-uje sa slike)

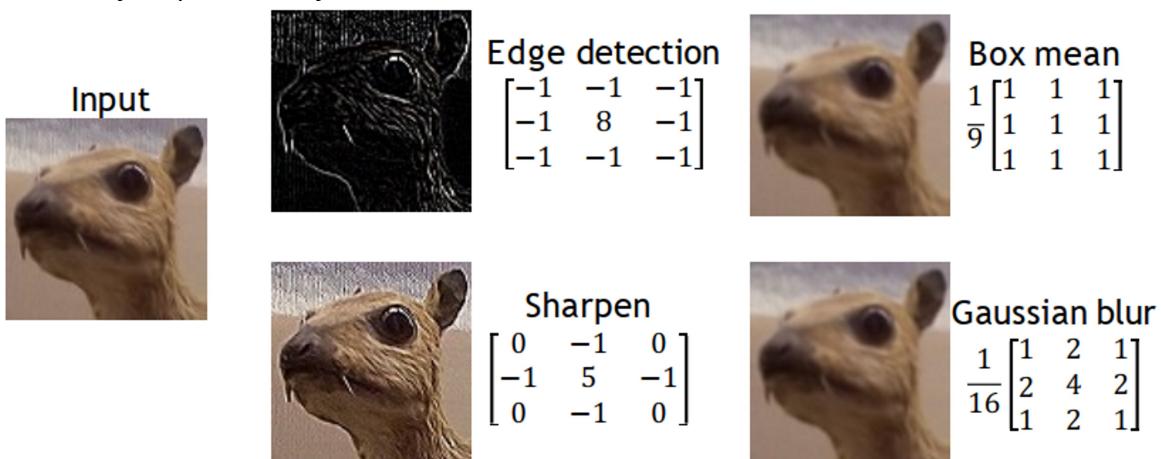
Posmatramo sliku kao matricu piksela, gde svaki piksel ima odredjenu vrednost (jacinu). Na isti nacin se primenjuje filter na sliku, kao u prethodnom primeru za signale f i g, samo sto se sada radi sa matricama, a ne nizovima vrednosti.



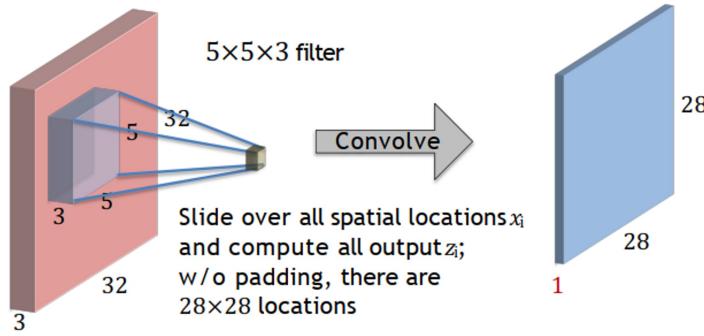
Kernel se primenjuje na sve 3x3 matrice na slici i dobija se krajnji rezultat:



Svaki kernel koji se primeni daje razlicito filtriranu sliku:

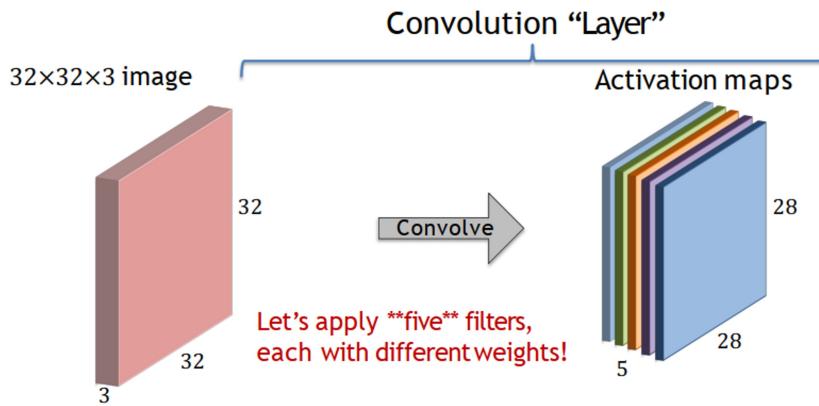


Konvolucija na RGB slikama:



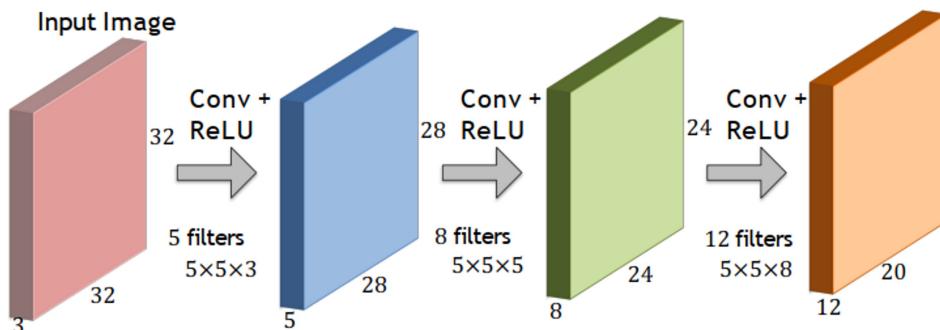
- slika je dimenzija  $32 \times 32 \times 3$ , a filter  $5 \times 5 \times 3$ 
  - 3 je dubina slike (broj kanala - rgb), i ona **mora biti ista i na slici i na filteru**
  - filter dimenzija  $5 \times 5 \times 3$  ima **toliki broj tezina (75) + 1 za bias**
- dobija se slika dimenzija  $28 \times 28 \times 1$  koja se naziva **aktivaciona mapa** (takođe i feature mapa)
  - broj 28 se racuna kao  $32-5+1$ , tj. **sirine slike-sirina filtra+1**, i taj broj označava koliko puta može da se izvrsi "klizanje" filtra na desno (za po 1 mesto - *stride*, koji može da bude i veci od 1, ali samo onoliki da se dobije ceo broj pomeranja)
  - broj 1 u dimenziji označava da su sve boje presle u crno belu sliku

**Konvolucioni sloj** se dobija kada se vise filtera sa razlicitim vrednostima (tj. tezinama, pojacanjima) primeni na jednu sliku:



- primenjeno je 5 filtera, pa se u konvolucionom sloju nalazi 5 aktivacionih mapa dimenzija  $28 \times 28 \times 1$
- **svaki filter hvata razlicitu karakteristiku slike** (horizontalne linije, vertikalne linije, krugove...)

CNN mreža predstavlja konkatenaciju konvolucionih slojeva i aktivacionih funkcija (f-ja ReLU):



- ReLU f-ja je bila osnova za nastajanje dubokih mreža (ranije koriscena sigmoid funkcija nije mogla da se koristi za ovakvo ucenje)
- ReLU određuje koji *feature* na slici nam je bitan (f-ja ima "mrtav" deo koji je za potiskivanje delova koji nisu bitni, i linearni deo za isticanje onih koji jesu)

Dodavanje *padding*-a:

- na ivicama oko cele matrice stavi se jos po jedan dodatni piksel, i to je uglavnom broj 0
- znacaj:
  - cuvanje informacija na krajevima, jer ti pikseli imaju manje susednih piksela u odnosu na druge i onda se njihova vrednost manje koristi
  - sprecava se smanjenje dimenzija, jer se bez padding-a prilikom prolaska kroz filter dimenzije mape znacajno smanjuju
  - invarijantnost na translaciju

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Input ( $N \times N$ ):  $7 \times 7$   
 Filter ( $F \times F$ ):  $3 \times 3$   
 Padding ( $P$ ): 1  
 Stride ( $S$ ): 1  
 Output  $7 \times 7$

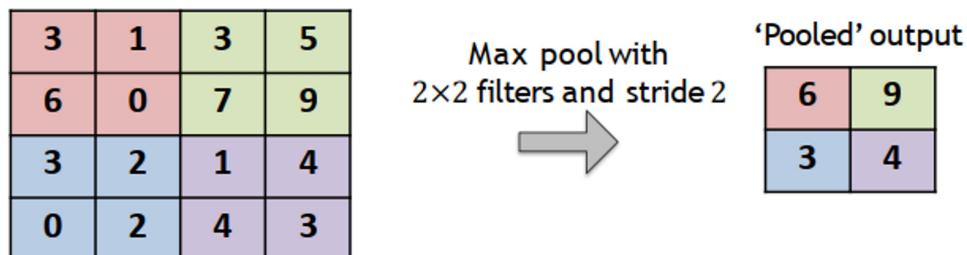
Formula za racunanje dimenzija izlaza:

$$\left( \left\lfloor \frac{N + 2P - F}{S} \right\rfloor + 1 \right) \times \left( \left\lfloor \frac{N + 2P - F}{S} \right\rfloor + 1 \right)$$

Pooling:

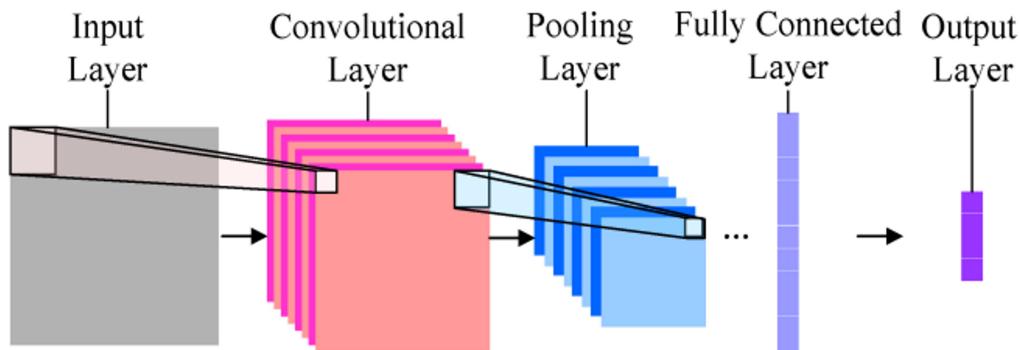
- **pooling** tehinka podrazumeva smanjivanje rezolucije slike (dimenzije) radi sazimanja njenih prostornih dimenzija i smanjivanja broja parametara
- konvolucioni sloj sluzi za *feature extraction* tako sto racuna sve *feature* u matrici, dok *pooling* sloj sluzi za *feature selection* - bira najjacu aktivaciju tako da se izdvaja ono sto je najjasnije prikazano na slici
- **max pooling** - iz regiona  $F \times F$  bira najvecu vrednost i nju upisuje u *pooling* sloj (izlaz):

Single depthslice of input



- **average pooling** - iz regiona  $F \times F$  racuna prosečnu vrednost i nju upisuje u *pooling* sloj

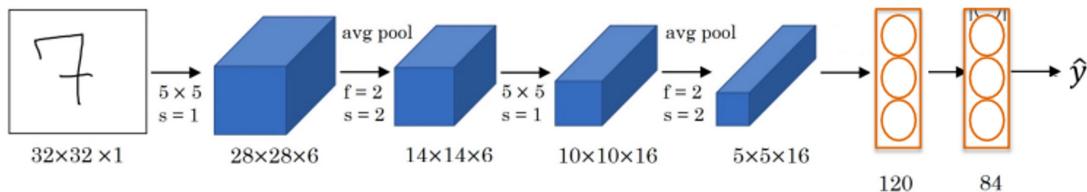
CNN prototip:



## LeNet mreza

- razvio je Yann LeCun i njegovi saradnici 1989. godine, a bila je dizajnirana za **prepoznavanje rukom pisanih znakova**

Arhitektura:



- prima slike velicine 32x32, sto je standardna velicina za rukom pisane znakove
- prvi sloj - CNN - koristi 6 filtera velicine 5x5, pa je izlaz aktivaciona mapa dubine 6
- drugi sloj - pooling - koji koristi **average pooling** (tada se koristio) s regijama velicine 2x2 i korakom (*stride*) 2
- treci sloj - CNN - koristi 16 filtera velicine 5x5
- cetvrti sloj - pooling - isto kao i u drugom sloju
- peti i sesti slojevi - FC - sa 120 i 84 neurona
  - u FC slojevima su se koristile **tanh/sigmoid aktivacije**, koje se sada ne koriste
- 10 izlaznih klasa za cifre

## AlexNet mreza

- slična LeNet mrezi, ali:
  - dosta veca (1000 puta)
  - koristi **max pooling**
  - koristi **ReLU funkciju**
  - koristi **dropout** za sprecavanje preobucavanja
- ulazna slika se smanji i *crop-uje* na velicinu 256x256
- pre obucavanja se vrši augmentacija podataka na sledeće nacine:
  - mirroring
  - random crops
- slojevi: CONV-POOL-CONV-POOL-CONV-CONV-CONV-POOL-FC-FC-FC
  - poslednji FC sloj ima 1000 neurona i koristi **softmax** funkciju aktivacije, dok 2 FC sloja pre njega imaju po 4096 neurona i koriste ReLU
- ReLU je omogućila da mreza uči visestruko brže od one mreze koja koristi tanh funkciju
- **problem nestajuceg gradijenta**
  - kada gradijenti postaju eksponencijalno mali kako se propagiraju unazad kroz mrežu (backpropagation), što može uzrokovati probleme u obucavanju
  - pojednostavljanje neće moći da se pomeri (*update-i* su mnogo mali)
  - desava se kada se koriste tanh/sigmoidne funkcije, zbog male vrednosti njihovog prvog izvoda, za razliku od situacije kada se koristi ReLU funkcija kod koje se ne javlja taj problem
- **problem ekspodirajućeg gradijenta**
  - kada gradijenti postaju izuzetno veliki tokom procesa propagacije unazad
  - javlja se kada su tezine inicijalizovane prevelikim vrednostima ili kada postoji jako veliki broj tezina i parametara