Duboko učenje. Konvolucijske neuronske mreže.

Sadržaj

- Ponavljanje:
  - Problem klasifikacije
  - Logistička regresija kao najjednostavniji klasifikator
  - Metrike za evaluaciju klasifikatora
  - Potpuno povezana neuronska mreža
- Konvolucijske neuronske mreže
- Učenje konvolucijske neuronske mreže
- Popularne strukture dubokih neuronskih mreža

Problem klasifikacije

Binary Classification

Spam
Not Spam

Cancer
Not Cancer

Positive Sentiment
Negative Sentiment

Fraud
Not Fraud

Multi-Class Classification

Cat
Dog
Fox
Tiger
Lion

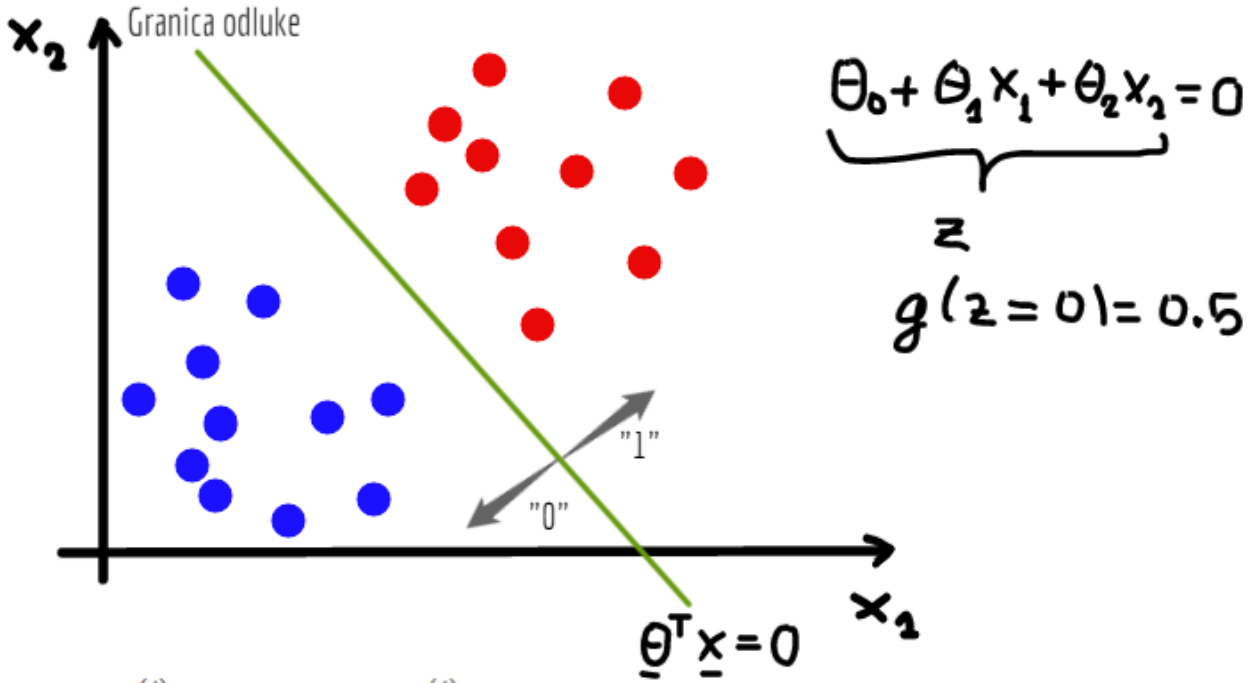
0 5
1 6
2 7
3 8
4 9

Person A
Person B
Person C
.
.
.

Logistička regresija

$$h_\theta(\mathbf{x}) = g(\mathbf{\Theta}^T\mathbf{x}) \quad \text{gdje je} \quad g(z) = \frac{1}{1+e^{-z}}$$

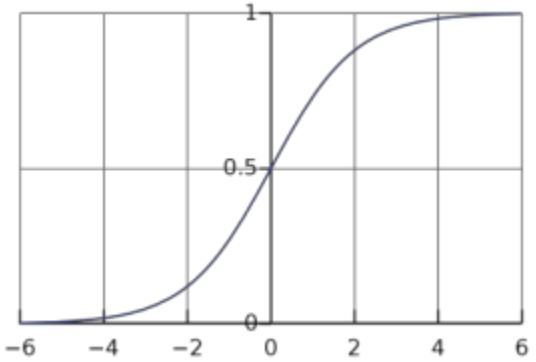
Binarna klasifikacija



$\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

z

$g(z = 0) = 0.5$



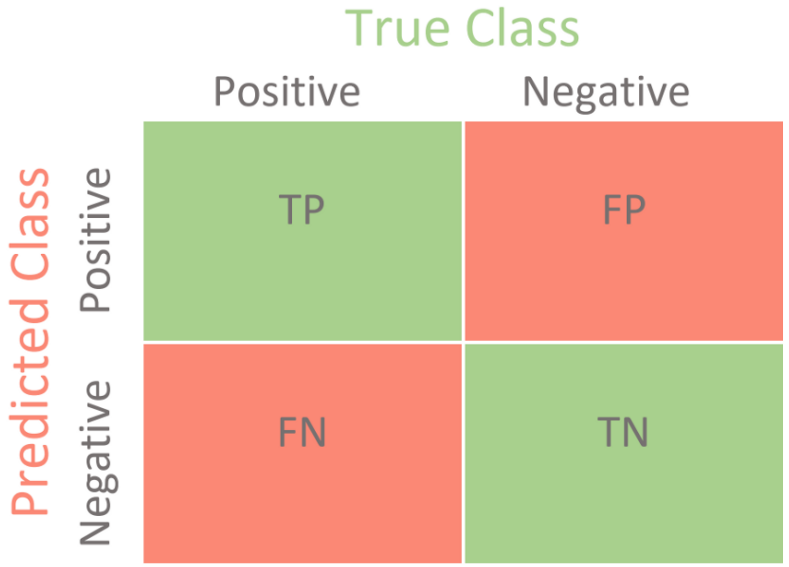
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))]$$

Izvor

Vrednovanje modela

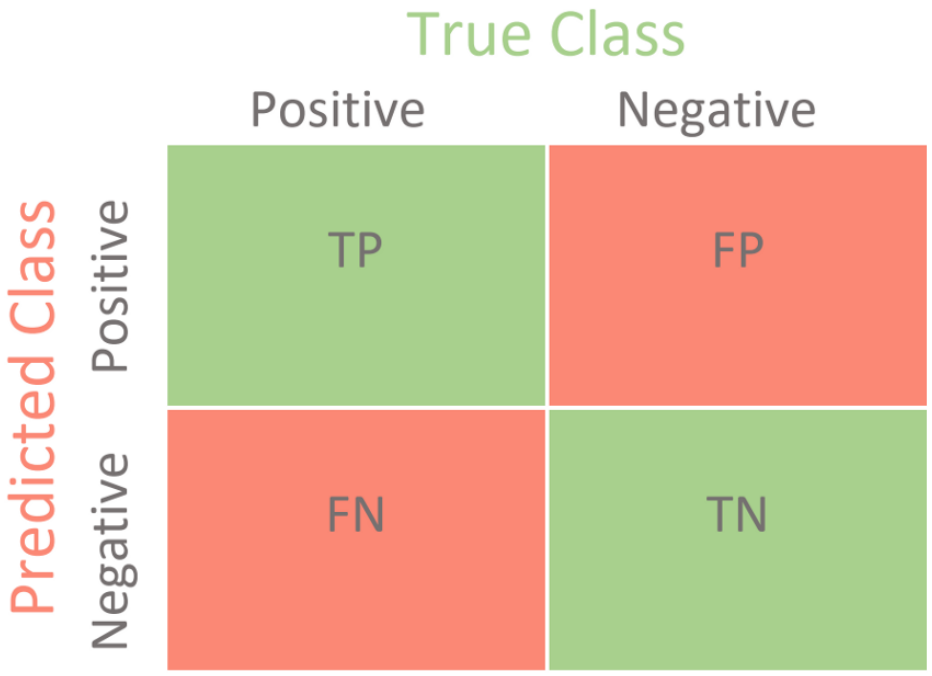
- Metrike za klasifikaciju omogućuju nam evaluaciju (procjenu) efikasnosti ML modela
  - Potrebno je imati mjeru koja pokazuje koliko dobro se predikcija dobivena modelom poklapa sa stvarnim (opaženim) podacima

- **Matrica zabune:**



Vrednovanje modela

- Precision (Positive Predictive Value) - **Preciznost**



- Kada algoritam predvidi pozitivnu klasu, koliko često je to točno?

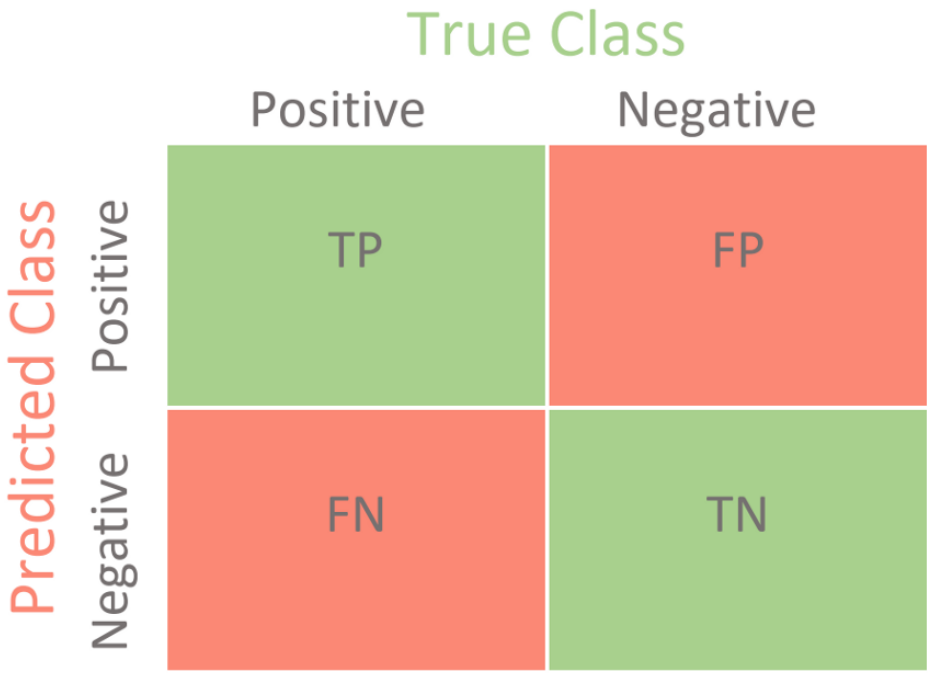
$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- Pogodna mjera kada je „cijena” FP visoka – tj. hoćemo imati što manje FP

- Primjer: spam filtar – pozitivni primjer je spam email
- FN (spam će se naći u inboxu) je prihvatljivije nego FP (non-spam će se filtritrati prema spam filtru)

Vrednovanje modela

- Recall (Sensitivity, True Positive Rate) - **Odziv**



- Koliki dio stvarno pozitivnih primjera je klasifikator uspješno identificirao?

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- Pogodna mjera kada je „cijena” FN visoka – tj. hoćemo imati što manje FN

- Primjer: **detektor prevara** kod transakcije - pozitivni primjer je „prevara”
- FP (normalne transakcije označene su kao moguće prevare) je prihvatljivije nego FN (prevare nisu detektirane)

Vrednovanje modela

- Accuracy - Točnost
  - Ne bi ju trebalo koristiti na neuravnoteženim skupovima

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

		Predicted/Classified: Negative	Predicted/Classified: Positive
Actual	Negative	998	0
	Positive	1	1

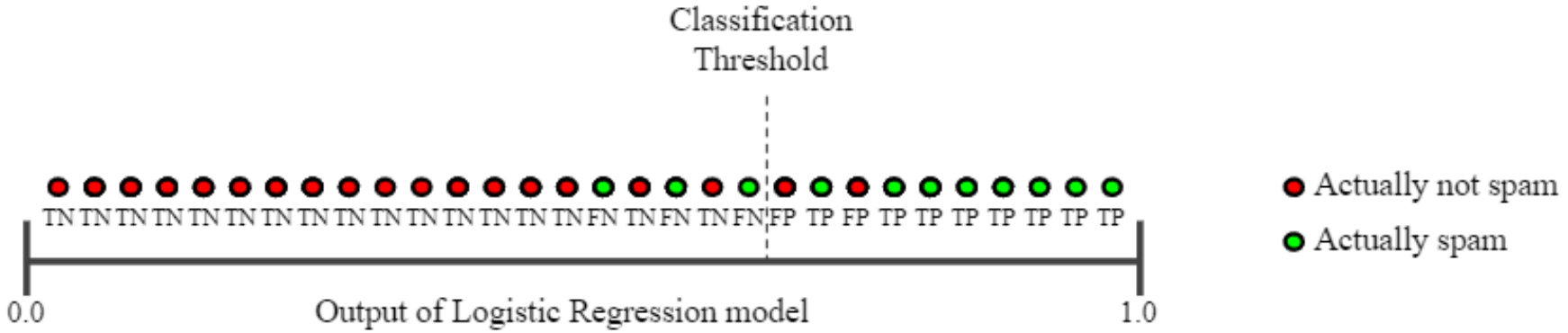
Kolika je točnost? 99.9%

- F1 score – harmonijska sredina precision i recall
  - Kada je potreban balans između precision i recall

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Vrednovanje modela

- vrijednosti *precision i recall* ovisi o korištenom pragu klasifikatora
- Primjer prikazuje 30 predikcija email klasifikatora
  - Desno od granice klasifikator predviđa kao spam; lijevo not spam (regularni email)

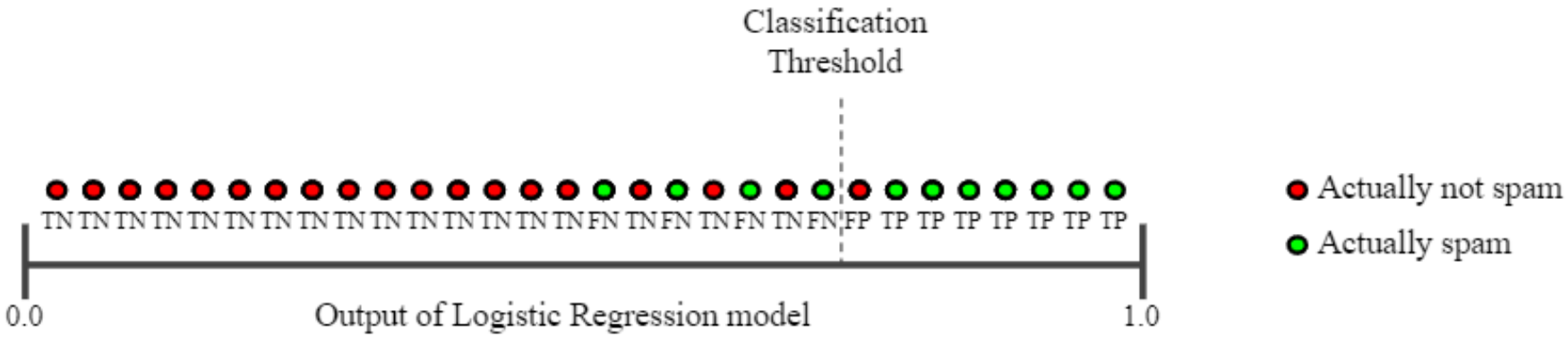


$$\text{Precision} = \frac{TP}{TP + FP} = \frac{8}{8+2} = 0.8$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{8}{8+3} = 0.73$$

Vrednovanje modela

- Ako se poveća prag klasifikatora:



$$\text{Precision} = \frac{TP}{TP + FP} = \frac{7}{7+1} = 0.88$$

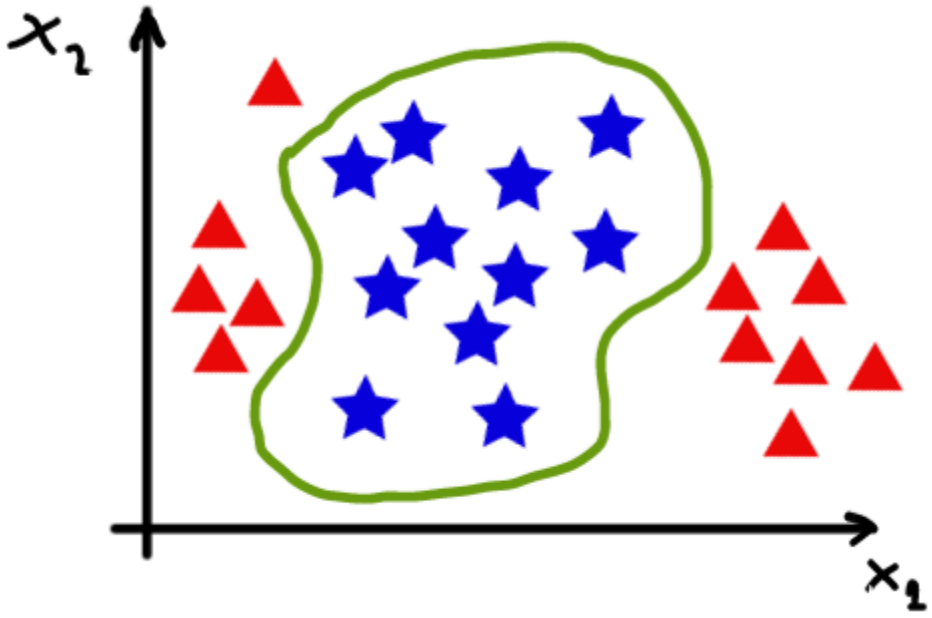
$$\text{Recall} = \frac{TP}{TP + FN} = \frac{7}{7+4} = 0.64$$

Preciznost se povećava s povećanjem praga, odziv se smanjuje!

Izvor

Proširenje logističke regresije

- matematičkom transformacijom ulaznih veličina moguće je dobiti nelinarnu granicu odluke

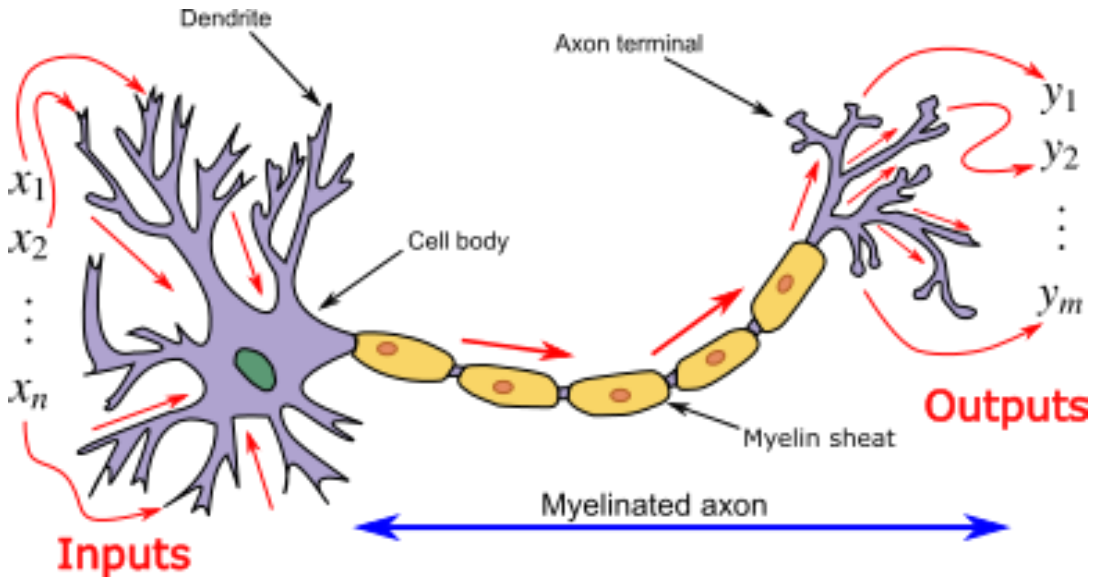


Proširenje logističke regresije

- Ovaj pristup je prihvatljiv ako je mali broj ulaznih veličina x_i (npr. 2-3)
- Za slučaj većeg broja ulaznih veličina, npr.
  - n = 100 → ako se žele uključiti sve kombinacije do kubičnih članova otprilike 170000 *features*
  - Slika 50 x 50 piksela (2500 ulaznih veličina); u slučaju RGB slike je to 7500 ulaznih veličina → ako se žele uključiti sve kombinacije do kubičnih članova otprilike to je nekoliko milijuna parametara
- Zbog velikog broja parametara može doći do overfittinga
- Moguće je uključiti samo neke transformacije ulazne veličine → postavlja se pitanje koliko složenu granicu odluke možemo dobiti na taj način
- Kako bi se ovakav pristup mogao uspješno primijeniti i na probleme s većim brojem ulaznih veličina, potrebno je adaptirati bazne funkcije (transformacije ulaznih veličina) na danim podacima
- Jedan od načina je fiksiranje broja baznih funkcija unaprijed, ali ih učiniti adaptivnim → koristiti parametarski oblik bazne funkcije gdje se parametri adaptiraju tijekom postupka učenja

Umjetne neuronske mreže

- Promatranje ljudskog mozga inspiriralo je koncept umjetne neuronske mreže (engl. *Artifical Neural Networks* – ANN)

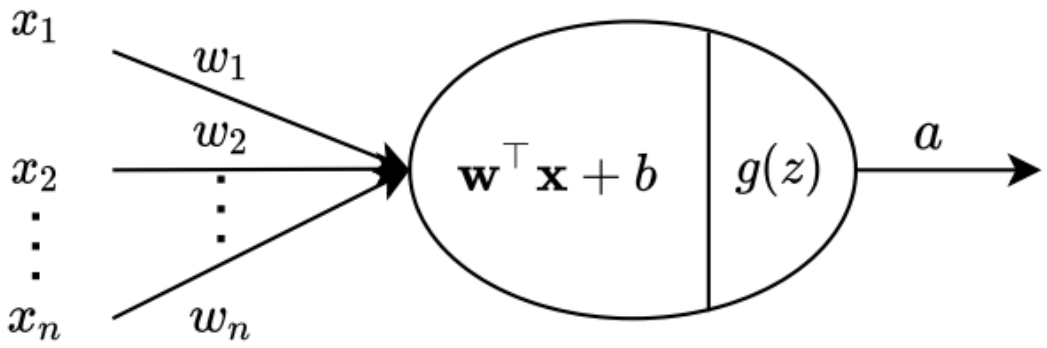


- Dendriti primaju signale drugih neurona
- Akson prenosi impulse do sinaptičkih terminala

Umjetni neuron

- Umjetni neuron prikazan je na slici i zapravo je vrlo sličan modelu logističke regresije - umjesto sigmoidne funkcije može se pretpostaviti bilo koja (derivabilna) funkcija *g*(*z*)
- Uobičajeno se ova funkcija naziva aktivacijska funkcija, a sam izlaz iz neurona *a* aktivacija

- Parametri: $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_n \end{bmatrix}^\top \qquad b$



$$z = \mathbf{w}^\top \mathbf{x} + b$$
$$a = g(z)$$

Aktivacijske funkcije

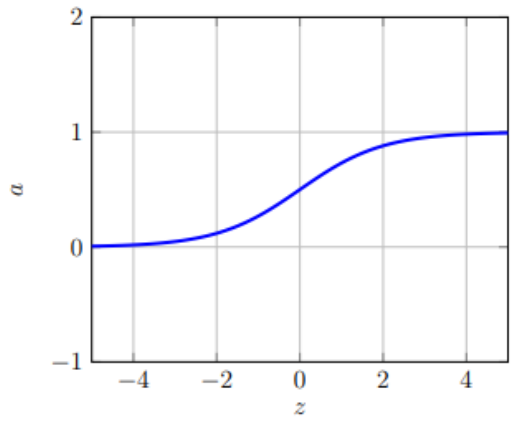
- najčešće korištene aktivacijske funkcije: sigmoidna, tanh, relu i leaky relu

$$a = \frac{1}{1 + e^{z}}$$

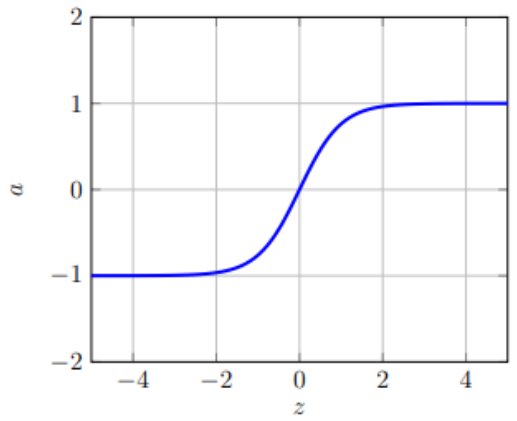
$$a = \frac{(e^{z} - e^{-z})}{(e^{z} + e^{-z})}$$

$$a = \max(0, z)$$

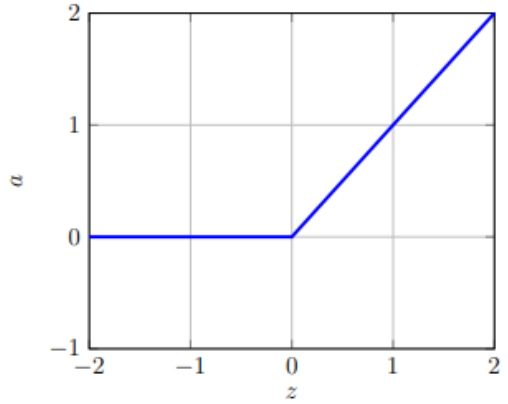
$$a = \max(0.001z, z)$$



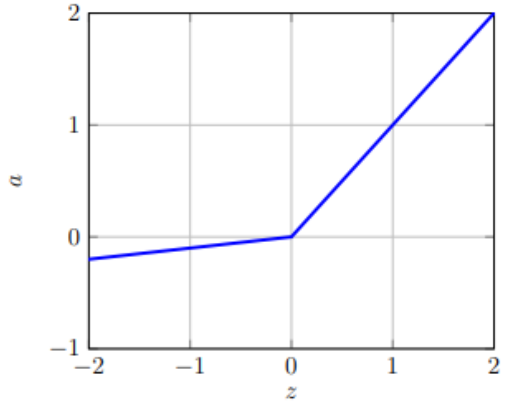
(a) Sigmoidna aktivacijska funkcija



(b) tanh aktivacijska funkcija



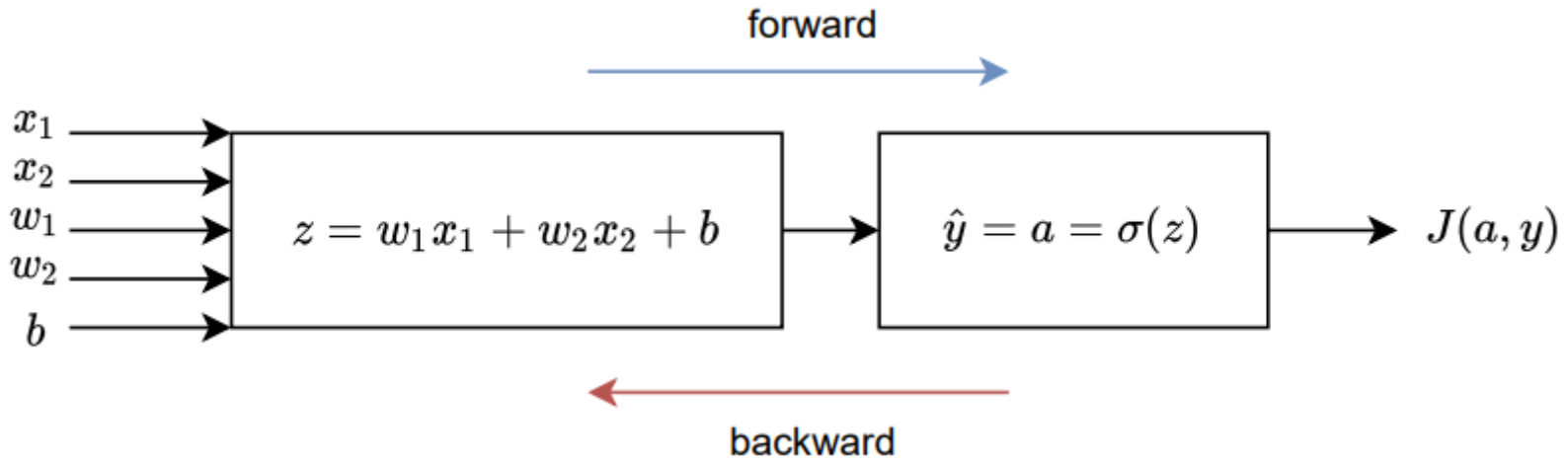
(c) ReLu aktivacijska funkcija



(d) Leaky ReLu aktivacijska funkcija

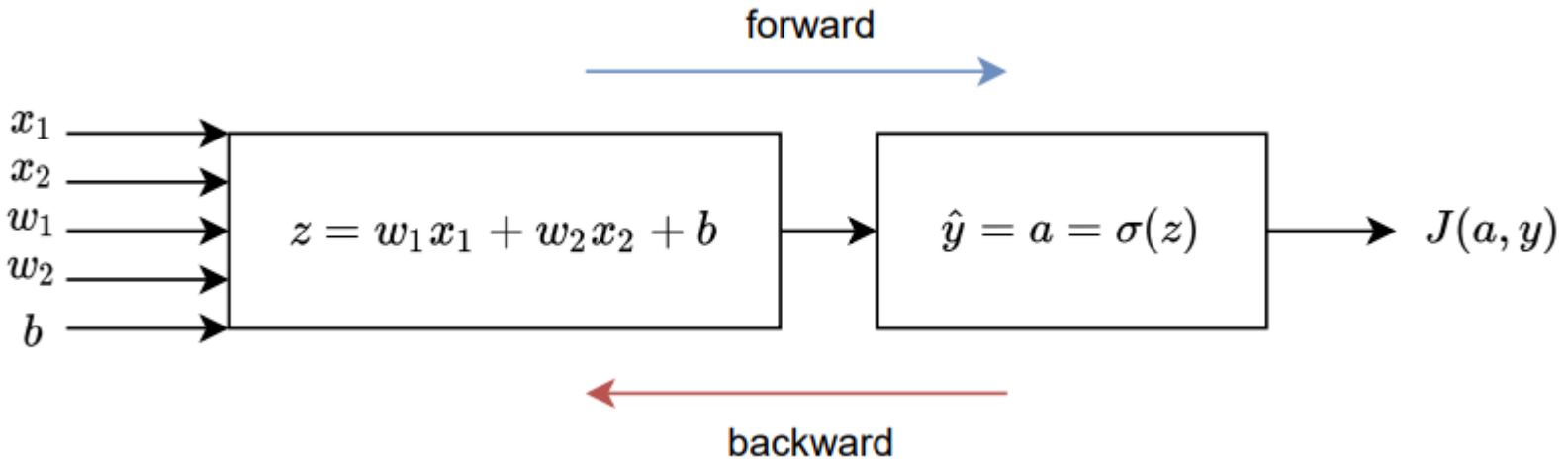
Gradijentna metoda za jedan neuron

- Podešavanje parametara na temelju podataka:



Gradijentna metoda za jedan neuron

- Primjer neurona s dva ulaza, podešavanje parametara na temelju podataka:



$$z = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \mathbf{x} + b = \mathbf{w}^\top \mathbf{x} + b$$

$$\hat{y} = a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$J(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$$

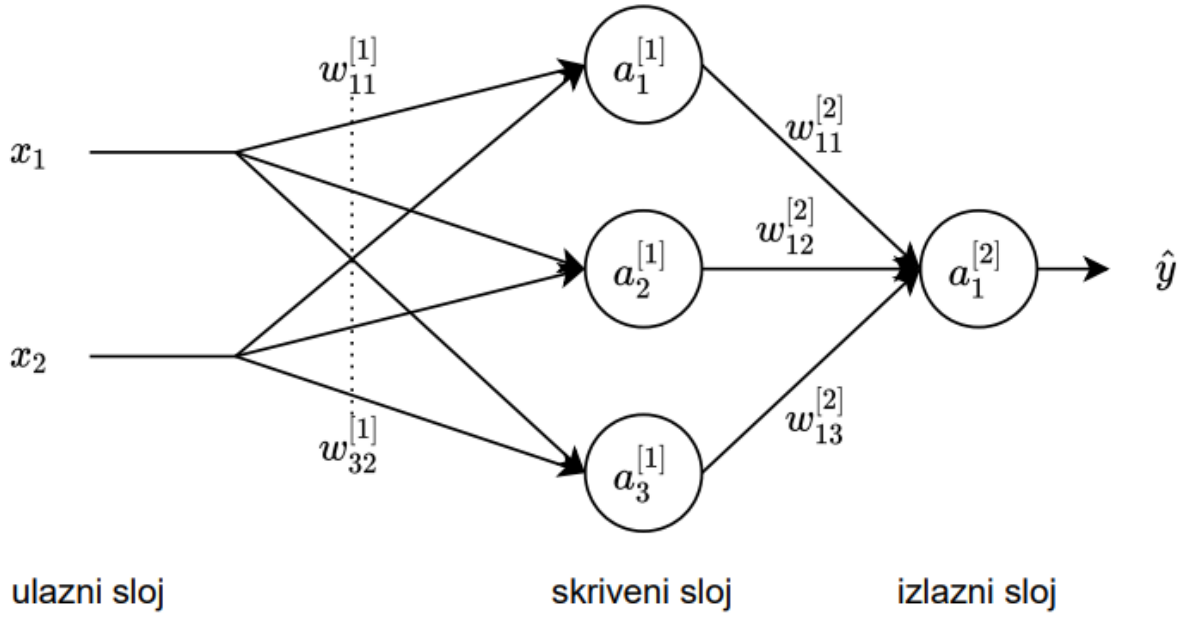
$$dw_1 = \frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_1} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial w_1} = dz x_1 = (a - y) x_1$$

$$dw_2 = \frac{\partial J}{\partial w2} = \frac{\partial J}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_2} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial w_2} = dz x_2 = (a - y) x_2$$

$$db = \frac{\partial J}{\partial b} = \frac{\partial J}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial b} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial b} = dz 1 = (a - y)$$

Višeslojne neuronske mreže

- Primjer mreže jednim skrivenim slojem i jednim izlaznim slojem



- Unaprijedna potpuno povezana mreža
- podešavanje parametara – backpropagation algoritam

Yes you should understand backprop

Duboko učenje u praksi

- Duboko učenje koristi se u rješavanju mnogih problema:
- Obrada slike i računalni vid (detekcija objekata, background removal, colorization of images, …)
- Video description generation
- Action and activity recognition, human pose estimation
- Robotika
- AI u računalnim igrama
- Obrada prirodnog jezika
- Posebno kod autonomne vožnje:
  - Detekcija objekata
  - Segmentacija scene
  - Upravljanje vozilom
  - Praćenje stanja vozača
  - …

Izvor

Računalni vid kao osnova autonomne vožnje

- Računalni vid je izrazito zahtijevan
- Jedan od osnovnih problema u računalnom vidu je klasifikacija slika (pridjeljivanje odgovarajuće klase ulaznoj slici)
- Računalo „vidi“ matricu koja je popunjena brojevima [0,255] (postoji „semantički gap“)

Izvor

Uz pretpostavku da imamo set diskretnih labela {biciklist, automobil, pješak, znak, …}

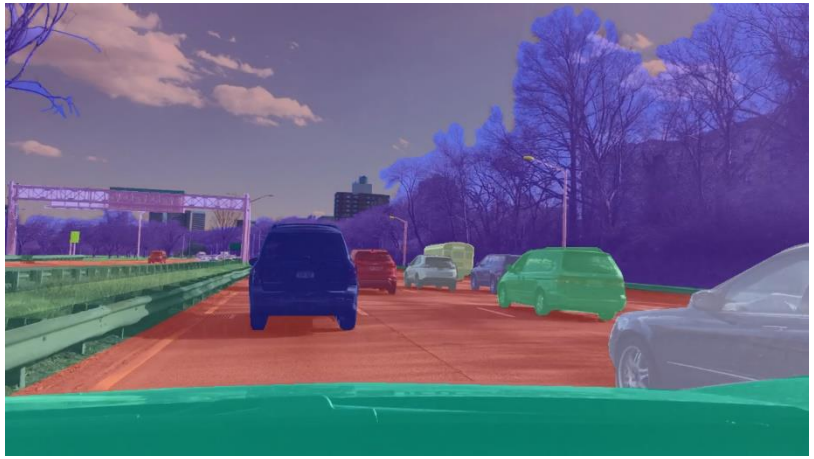
automobil

16	13	10	15	0
16	16	16	10	0
15	12	0	0	0
0	5	9	15	0
16	3	12	11	0
0	8	14	16	12
15	3	8	13	0
13	0	3	15	1
11	15	16	16	0
1	0	9	15	0
13	15	1	0	0
0	0	11	16	4

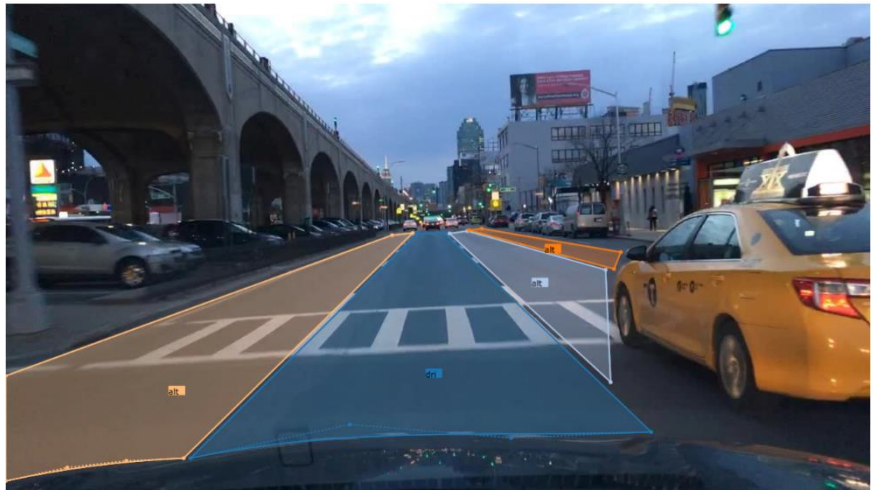
Računalni vid kao osnova autonomne vožnje

- Tipične primjene u autonomnoj vožnji

Instance segmentation

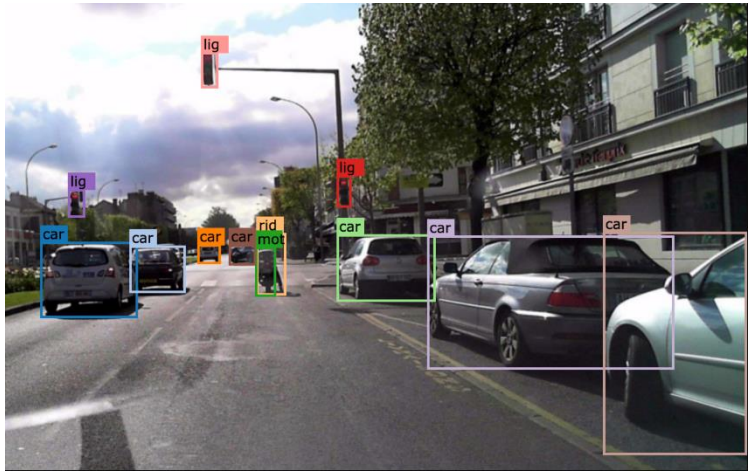


Driveable area



Lane markings detection

Road object detection



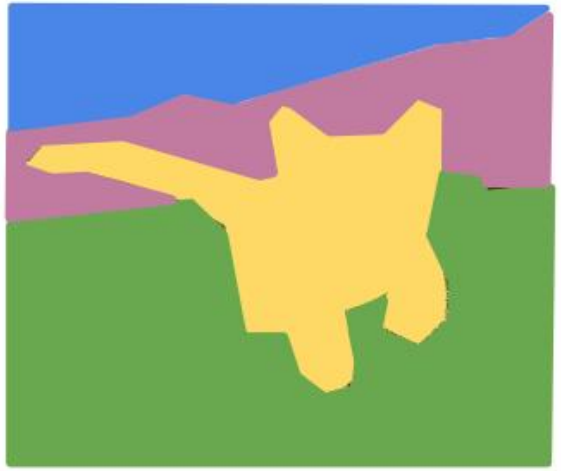
Izvor

Izazovi prilikom klasifikacije slika pomoću računalnog vida

- Koji su izazovi kod klasifikacije slika?
- Varijacije u točki snimanja – kako se pomiče kamera, mijenjaju se i vrijednosti piksela
- Varijacije u osvjetljenju i skaliranju
- Deformacije objekata
- Okluzija
- Background clutter
- Intraclass variation
- Velik broj klasa

Zadaće u računalnom vidu

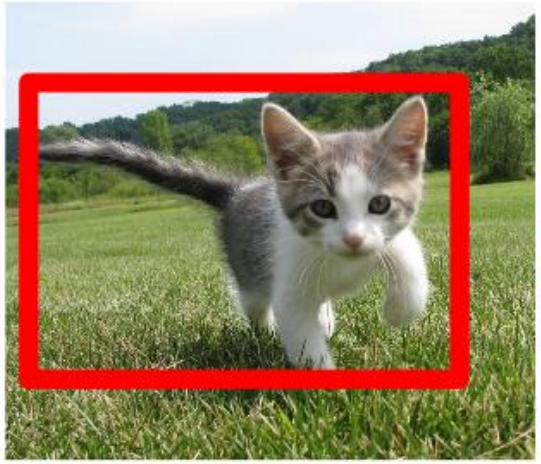
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

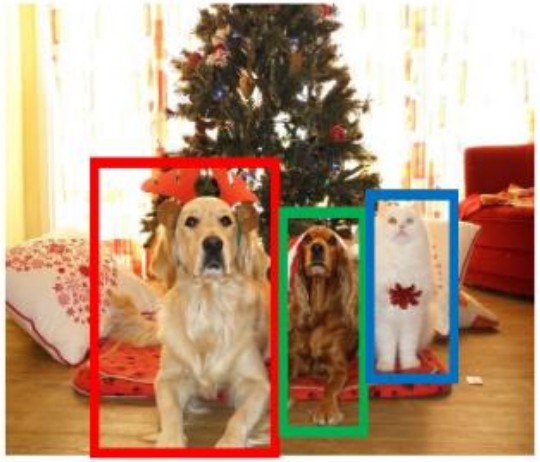
Classification + Localization



CAT

Single Object

Object Detection



DOG, DOG, CAT

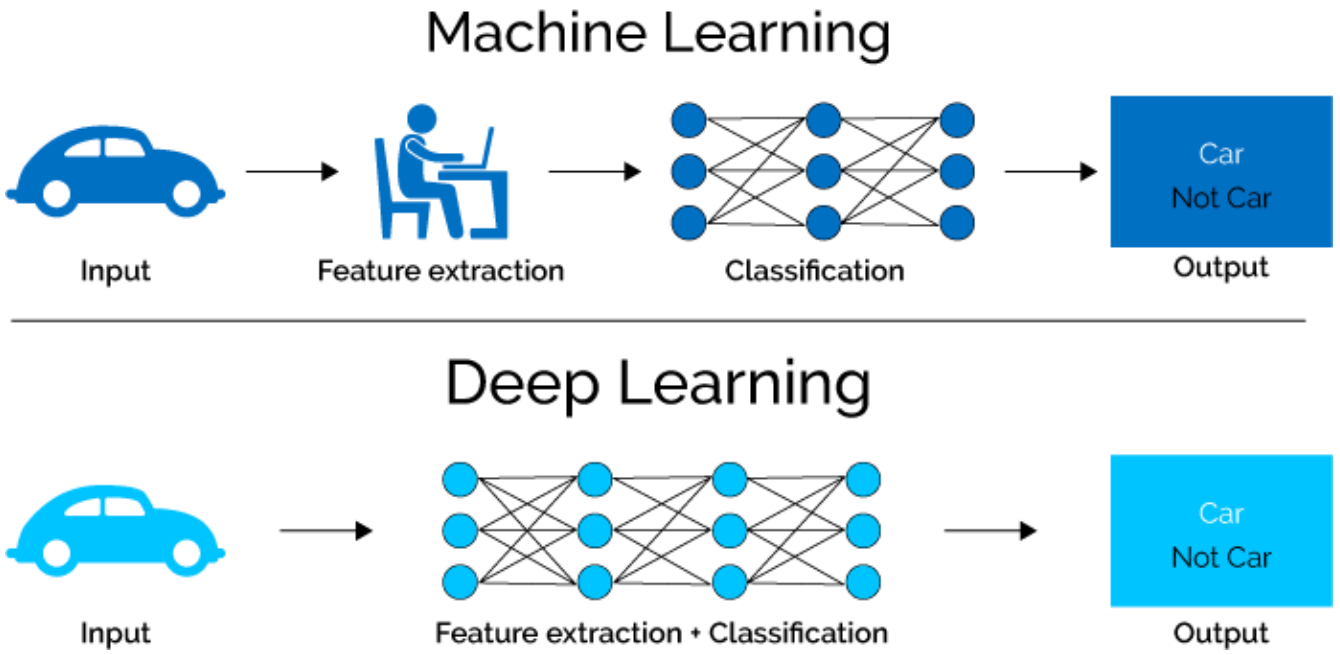
Instance Segmentation

DOG, DOG, CAT

Multiple Object

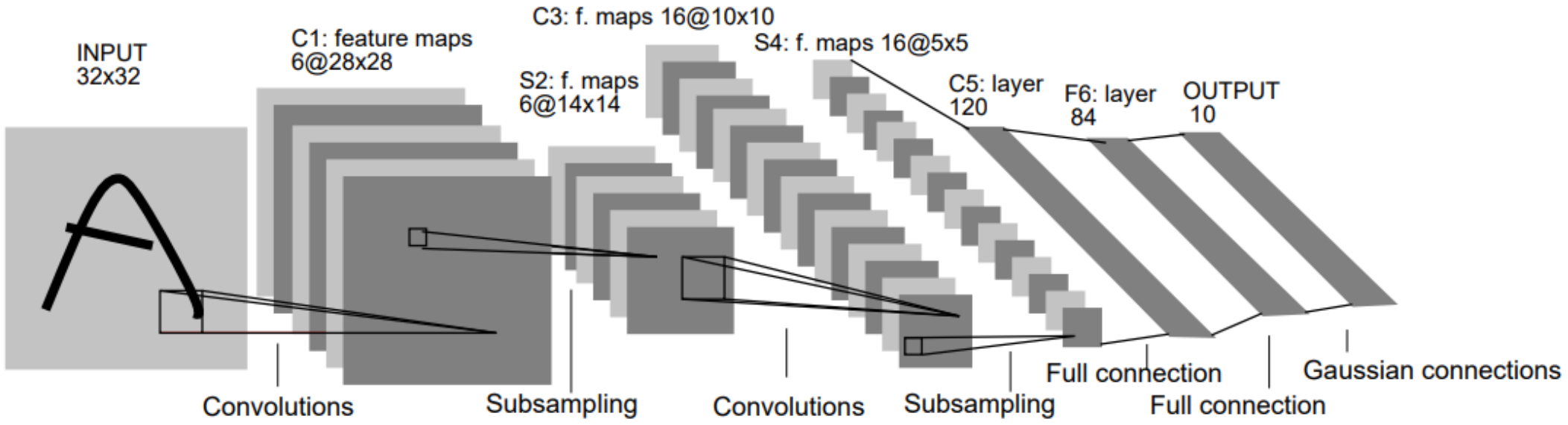
Tradicionalni računalni vid i duboko učenje

- „Tradicionalni“ računalni vid zasniva se na ručnom izdvajanju značajki te učenju klasifikatora na temelju podataka
- Moderni pristup temelji se na „end-to-end learning“ – značajke se također „uče“ iz dostupnih podataka



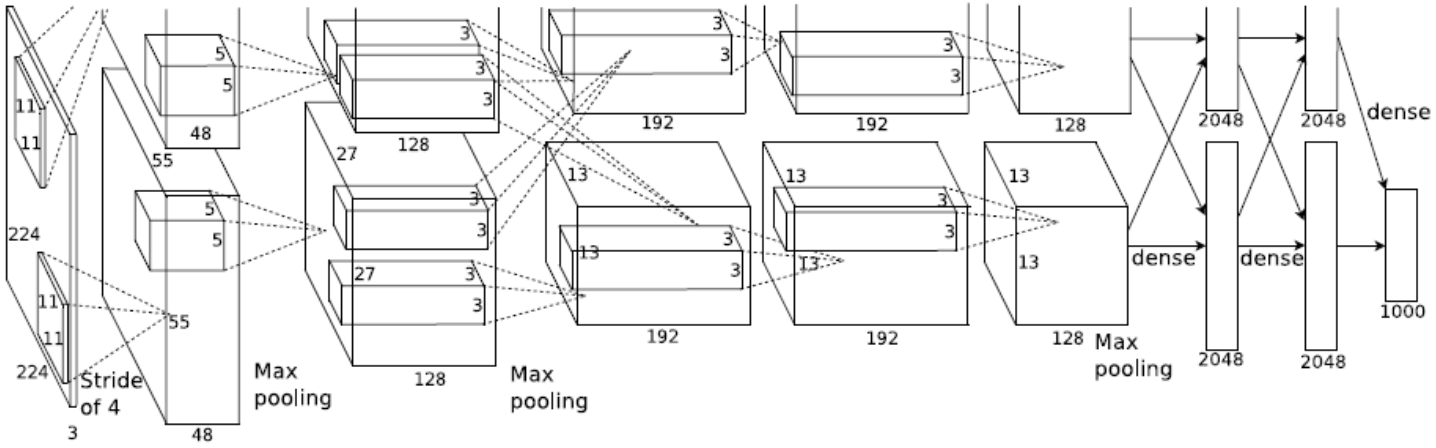
Prvo pojavljivanje konvolucijske neuronske mreže

- Znanstveni rad: LeCun, Bottou, Bengio, Haffner. Gradient-based learning applied to document recognition, 1998.
- Predložena mreža naziva LeNet-5
- Konvolucijska neuronska mreža koja je naučena pomoću bakcpropagation algoritma; prepoznavanje rukom pisanih brojeva
- Inspiracija otkriće lokalno osjetljivih i orijentacijski osjetljivih neurona kod vizualnog sustava mačke (Hubel and Wiesel) i NeoCognitron (Fukushima)
- Dugo vremena ovakva mreža nije korištena za složenije i veće slike već je korišten „klasični computer vision”



Značajniji rezultat - AlexNet

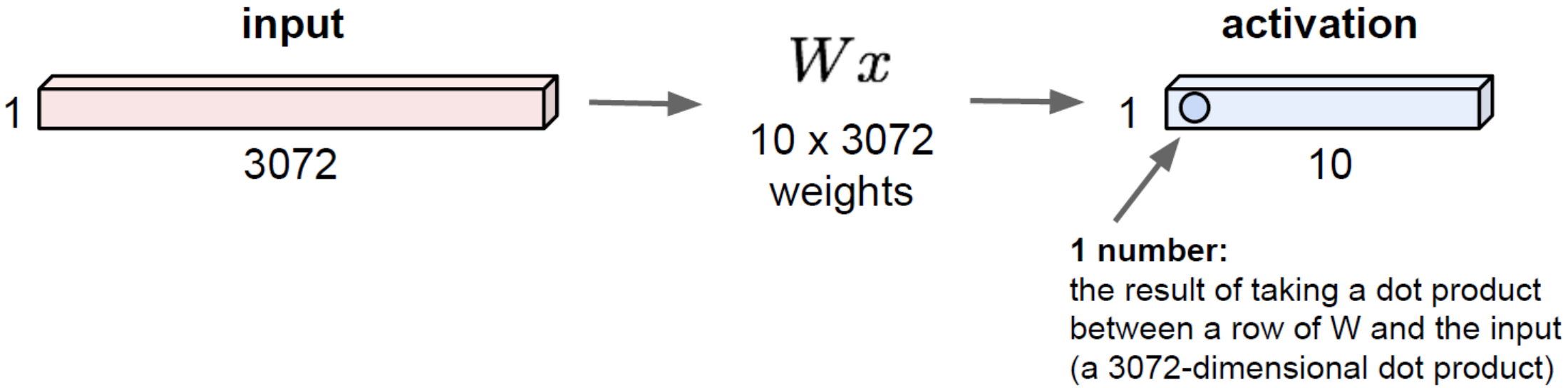
- Abdel-rahman Mohamed, George Dahl, Geoffrey Hinton. Acoustic Modeling using Deep Belief Networks, 2010.
- George Dahl, Dong Yu, Li Deng, Alex Acero . Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition, 2012.
- Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks, 2012.



„We trained the network for roughly 90 cycles through the training set of 1.2 million images, which took five to six days on two NVIDIA GTX 580 3GB GPUs”

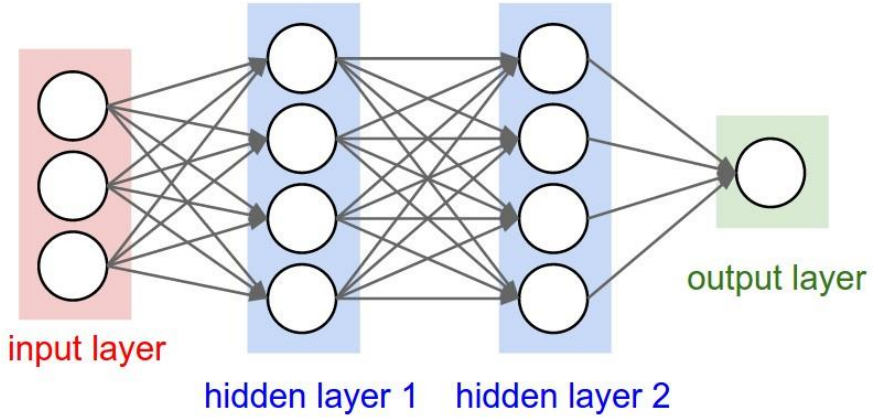
Podsjetnik – potpuno povezana mreža/sloj

- Primjenom potpuno povezanog sloja (engl. fully connected - FC) gubi se prostorna struktura koja je prisutna u slici (sliku razvučemo u vektor i množimo s matricom težina)
- Primjenom ovakvog sloja na ulaznu sliku broj parametara vrlo brzo raste s povećanjem rezolucije slike i/ili broja neurona → nepraktično

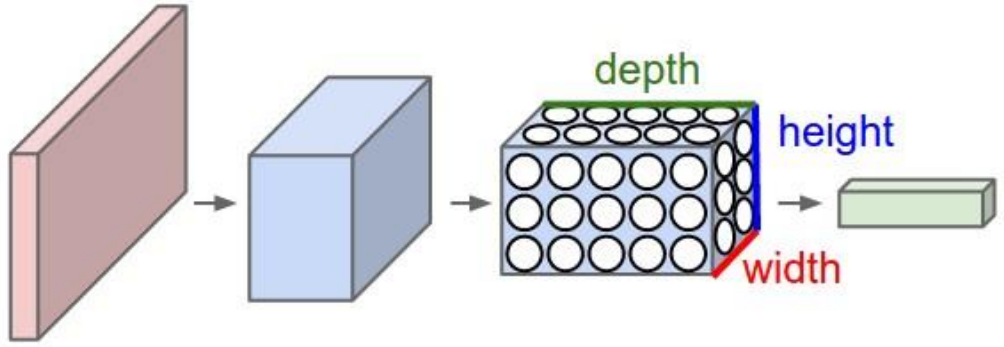


Konvolucijske neuronske mreže

- Konvolucijska neuronska mreža sastoji se od niza slojeva (konvolucijski slojevi, pooling slojevi, potpuno povezani slojevi)
- Broj parametara ovisi o strukturi (uobičajeno ~10M)
- Zanimljivo, prisutne su samo dvije matematičke operacije:
  - Skalarni produkt
  - Max funkcija
- Efikasno posloženi slojevi → uzmaju u obzir da je ulazni podatak slika; neuroni su posloženi u tri dimenzije (width, height, depth)



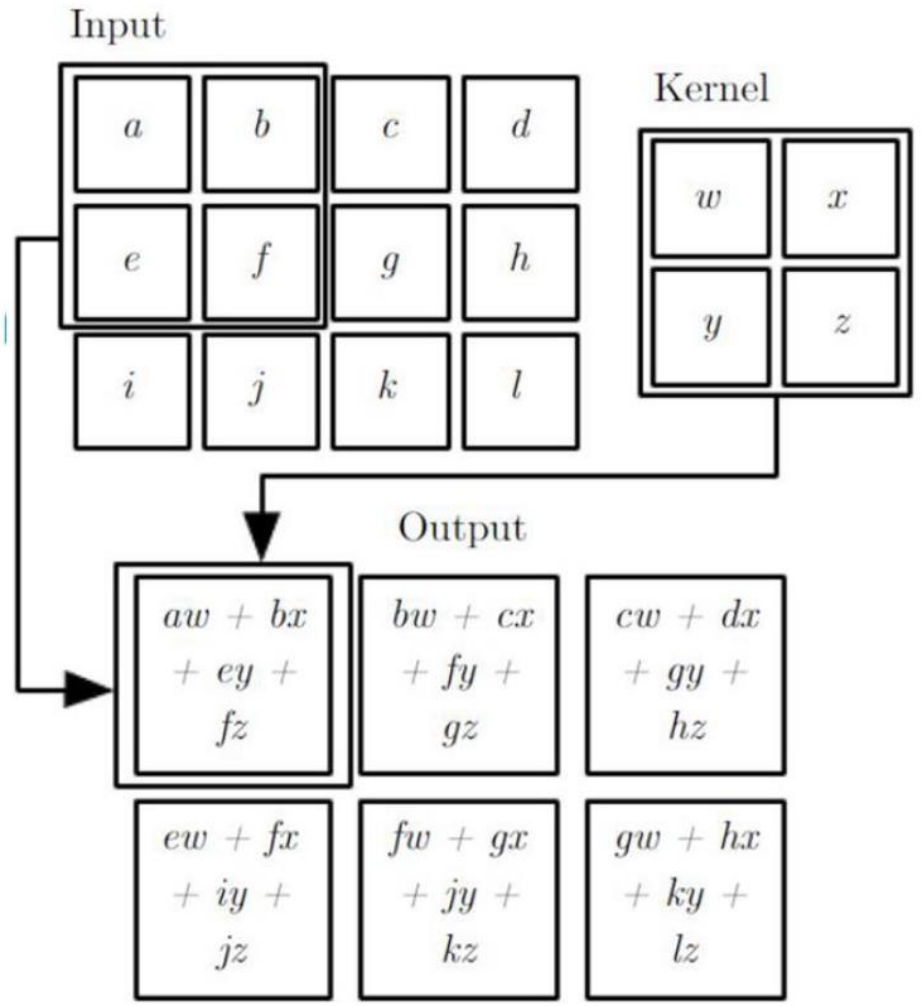
Klasična troslojna neuronska mreža



Konvolucijska neuronska mreža

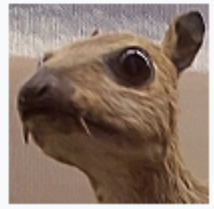
2D konvolucija

2D Operation



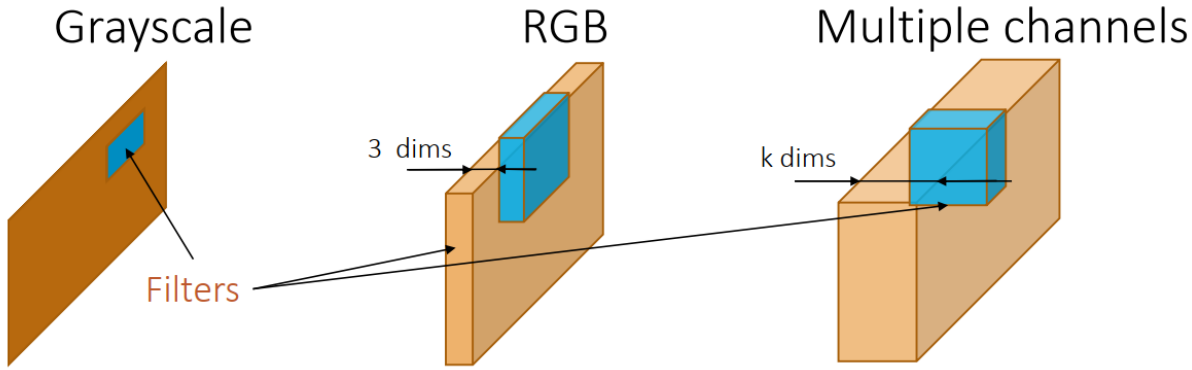
Primjeri poznatih kernela

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

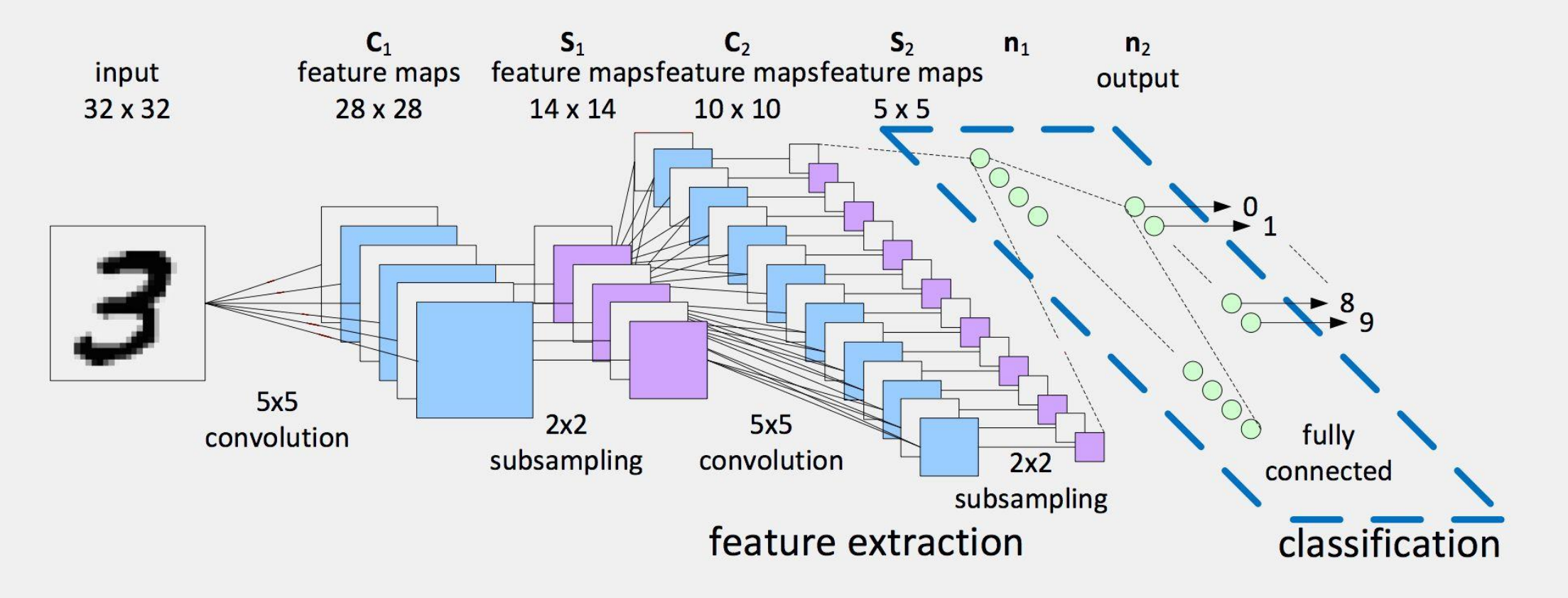
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5 × 5 (approximation)	$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Unsharp masking 5 × 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$\frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

Konvolucijske neuronske mreže, motivacija

- Obrada slike i računalni vid imaju mnogo “handcrafted filters”
  - Canny, Sobel, Gaussian blur, smoothing, lowlevel segmentation, morphological filters, Gabor filters...
- Postavlja se pitanje jesu li ovi filtri optimalni za dani problem klasifikacije?
- Mogu li se naučiti optimalni filtri iz dostupnih podataka?
- Naime objekti se mogu pojaviti na različitim mjestima, pod različitim kutem i sl.
- Ako su slike 2-D, parametri filtra također trebaju biti organizirani u 2-D
  - Na taj način mogu se naučiti lokalne korelacije između ulaznih veličina (piksela)
  - Moguće je iskoristiti prostornu prirodu slike
- Pretpostavka: različite statistike nisu ovisne o položaju na slici (slike su stacionarne)
- Isti filtar bi trebao raditi jednako na svakom dijelu slike



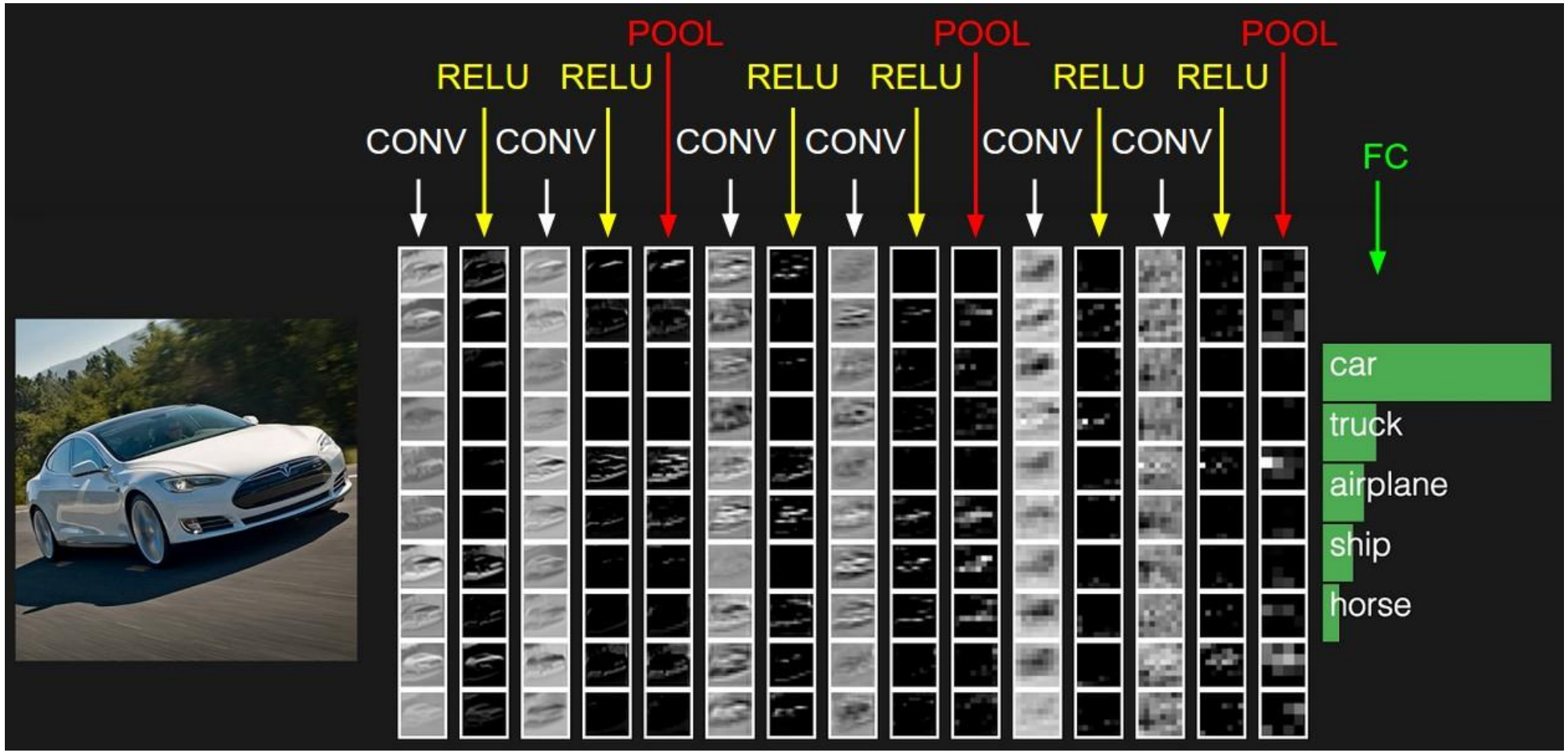
Konvolucijska neuronska mreža



Izvor

Konvolucijska neuronska mreža

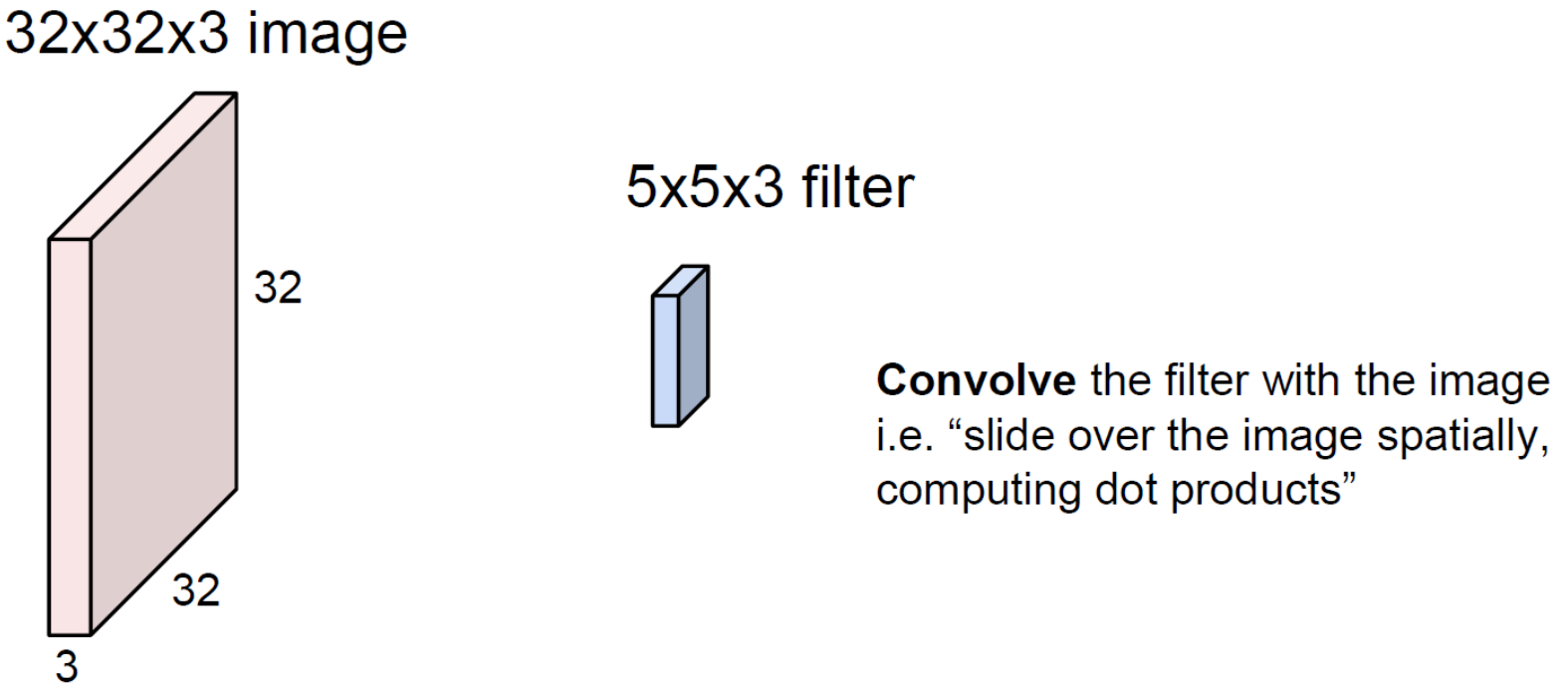
- Primjer arhitekture konvolucijske neuronske mreže i procesa inferencije



Izvor

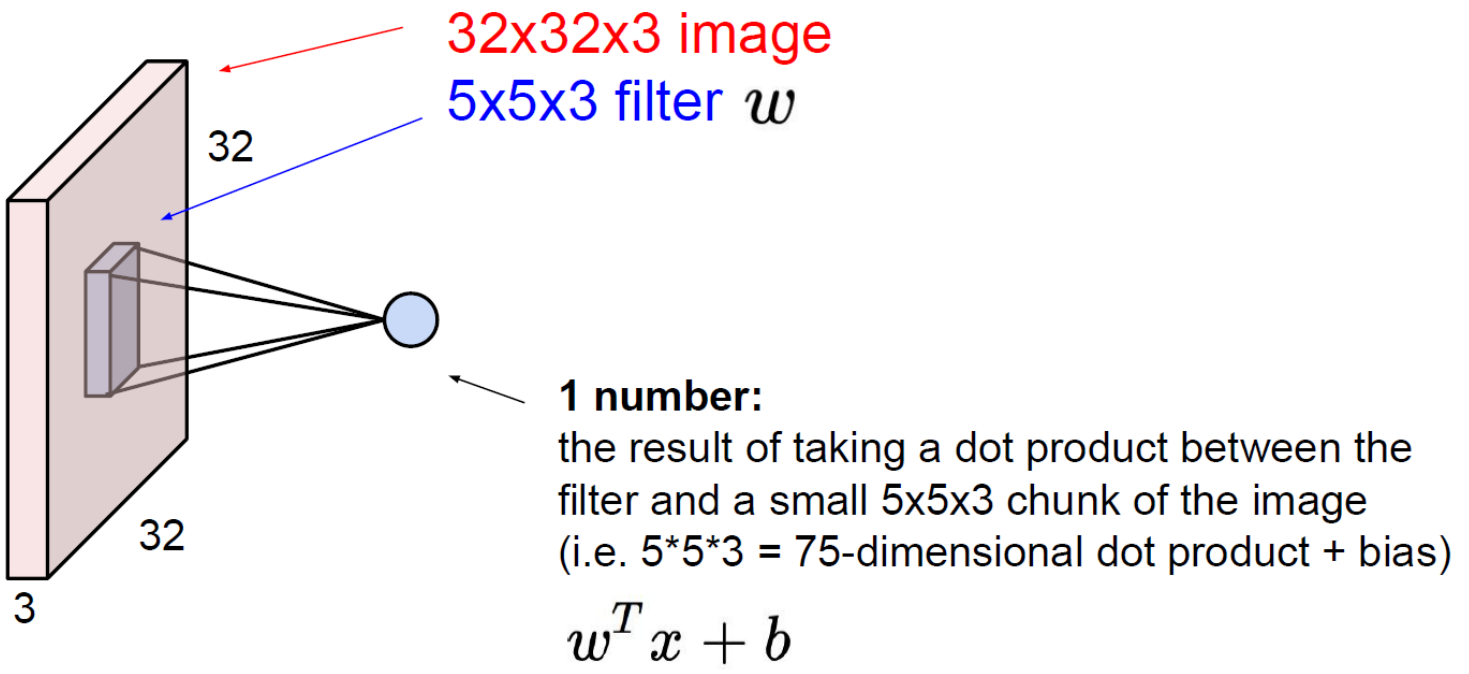
Konvolucijski sloj

- Osnova konvolucijskih neuronskih mreža je konvolucijski sloj
- Primjenom konvolucijskog sloja zadržava se prostorna struktura ulaznih podataka
- Svaki neuron povezan je samo s malim područjem u ulaznoj slici (receptivno polje) ili u prethodnom sloju



Konvolucijski sloj

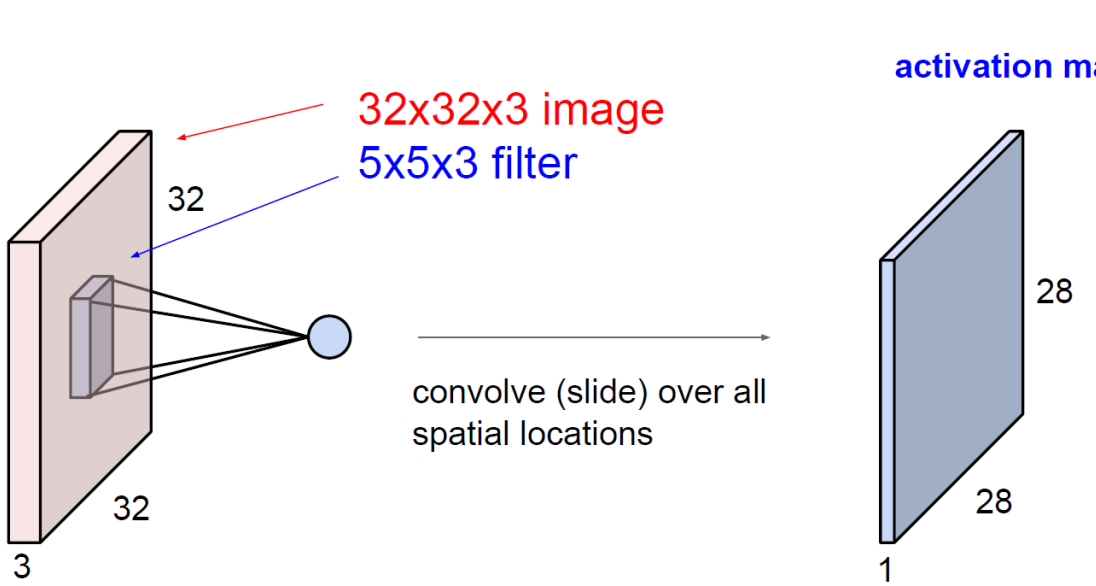
- Primjena filtra na određenom dijelu slike rezultira u skalarnoj vrijednosti (skalarni produkt dijela slike i filtra)
- Filtar pokriva malo prostorno područje, ali se prostire preko sva tri kanala ulazne slike)
- Težine (w) filtra i bias (b) predstavljaju mogu se podešavati (to su parametri mreže; mijenjaju se tijekom učenja)

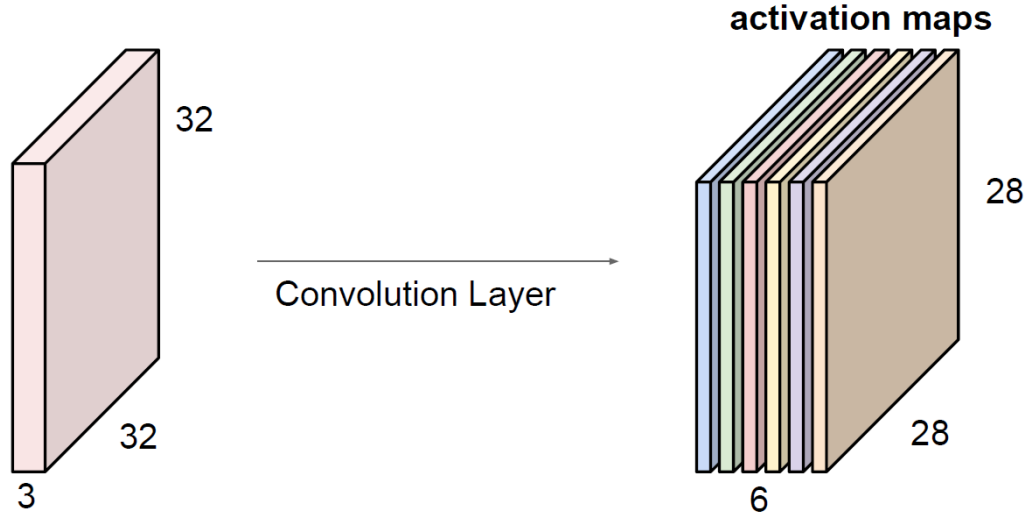


Aktivacijska mapa

- Primjenom istog filtra na različitim pozicijama dobiva se aktivacijska mapa → dvodimenzionalna matrica koja sadrži odziv filtra na pojedinom dijelu ulazne slike
- Svi neuroni iste aktivacijske mape imaju zajedničke parametre (znatno manje parametara nego kod FC sloja)
- Moguće je primijeniti više filtara → dobivaju se zasebne aktivacijske mape koje se „slažu“ jedna pored druge (dobivamo drugačiju reporezentaciju ulazne slike, tzv. 28x28x6 volume)

6 (5x5) konvolucijskih filtara
Ukupno parametara (5*5*3*6 + 6) = 456

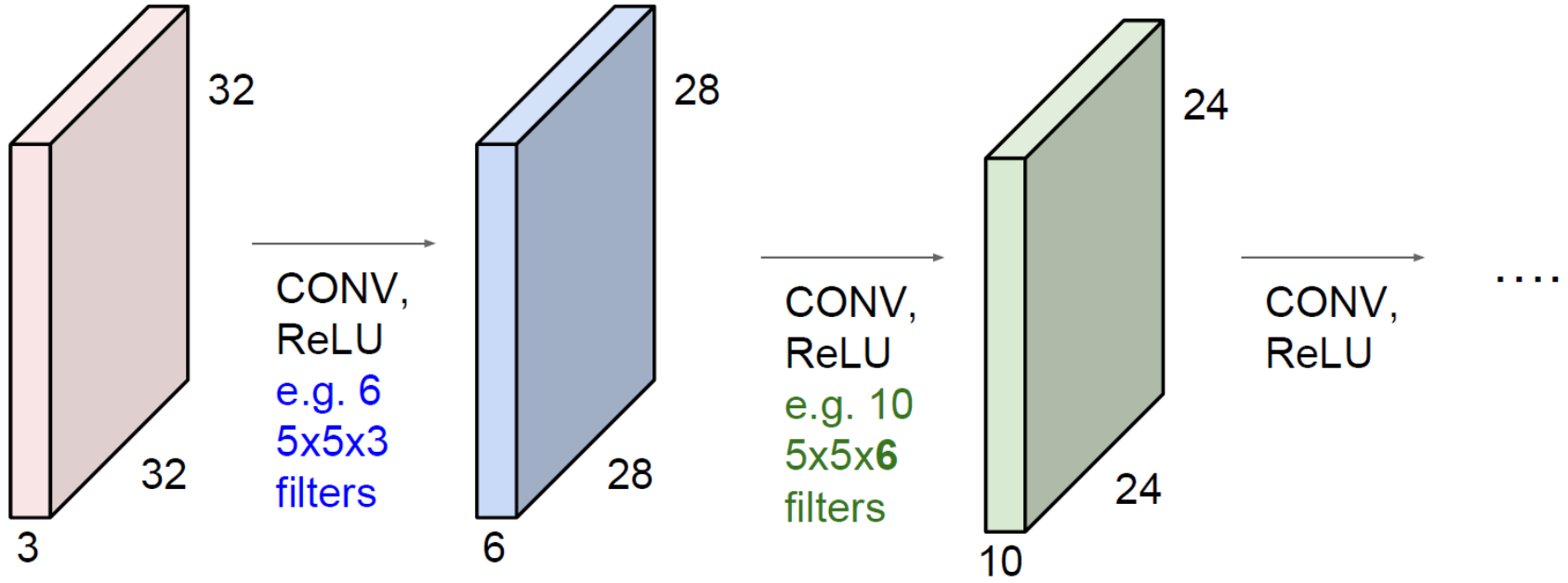




U slučaju potpuno povezanog sloja broj parametara bio bi jednak: 32*32*3*28*28*6 =14450688 (14.5M)

Konvolucijska mreža

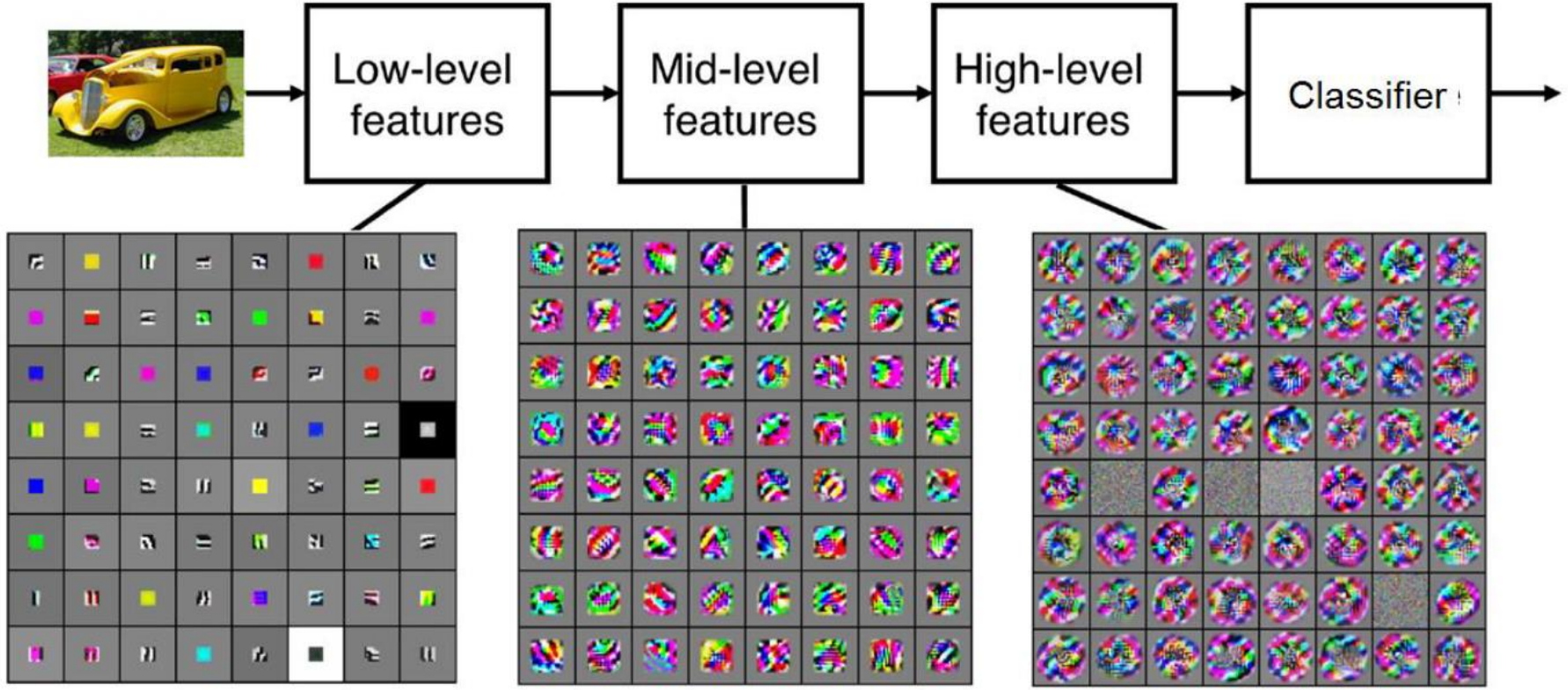
- CNN temelji se na sekvenci takvih konvolucijskih slojeva između kojih se nalaze aktivacijske funkcije (najčešće ReLU)
- Svaki neuron povezan je s lokalnim područjem ulaznog volumena (prostorni opseg se često naziva receptivno polje, a odgovara veličini filtra)



Što se postiže sekvencom konvolucijskih slojeva?

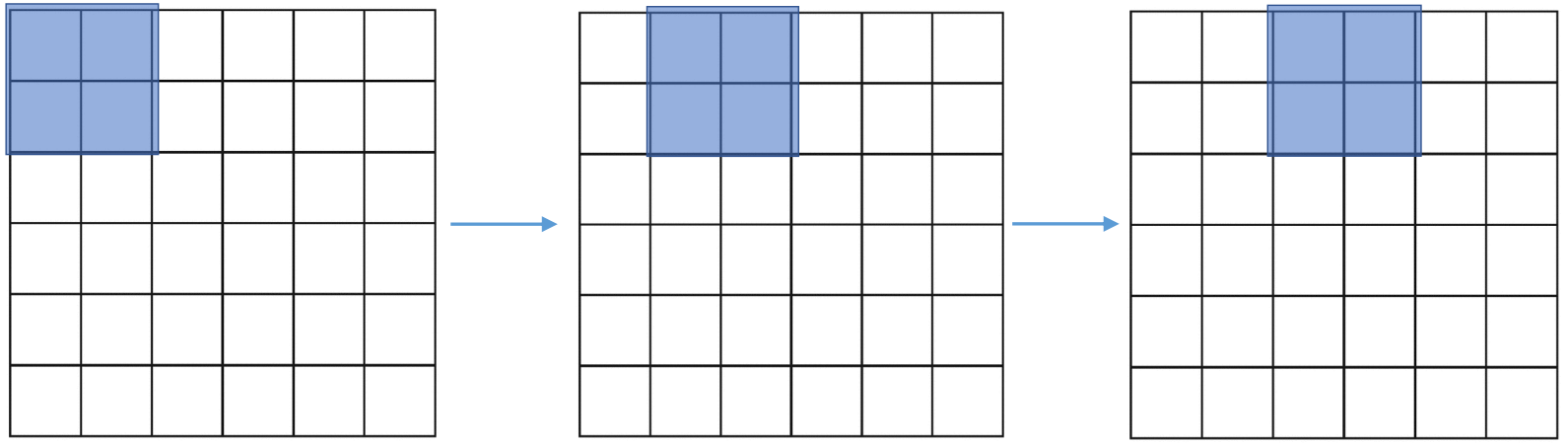
Features

- Prior to the widespread adoption of CNNs, most pattern recognition tasks were using an initial stage of hand-crafted feature extraction followed by a classifier.
- The breakthrough of CNNs is that features are learned automatically from training examples .



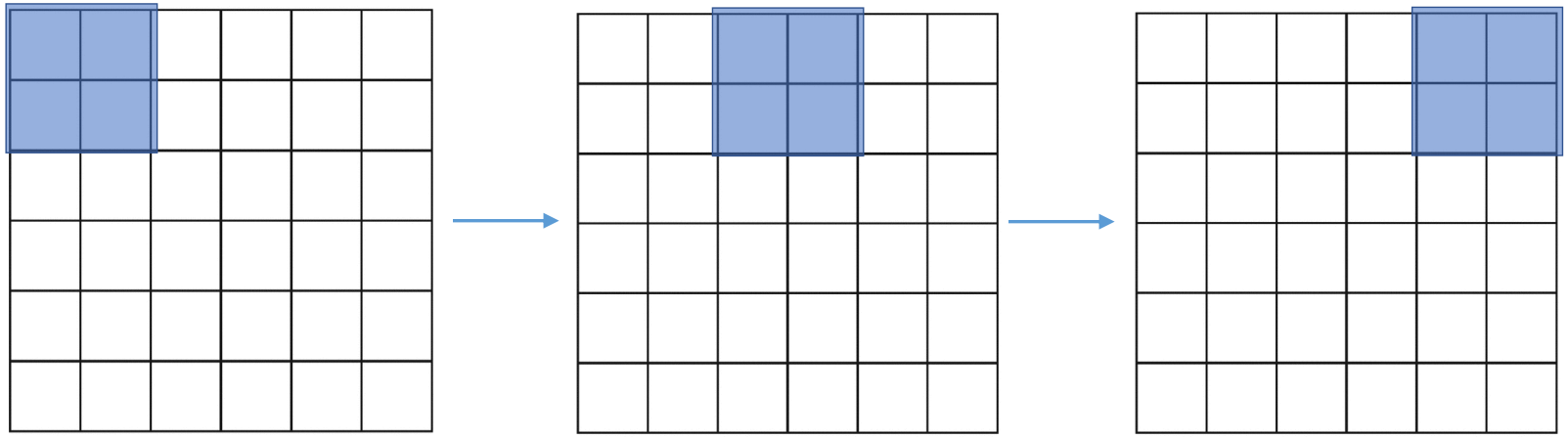
Konvolucijski sloj u detalje

- Stride → definira za koliko se filtar pomiče po aktivacijskoj mapi u svakom koraku



Filtar 2x2
Stride 1

Kolika je veličina rezultantne aktivacijske mape?

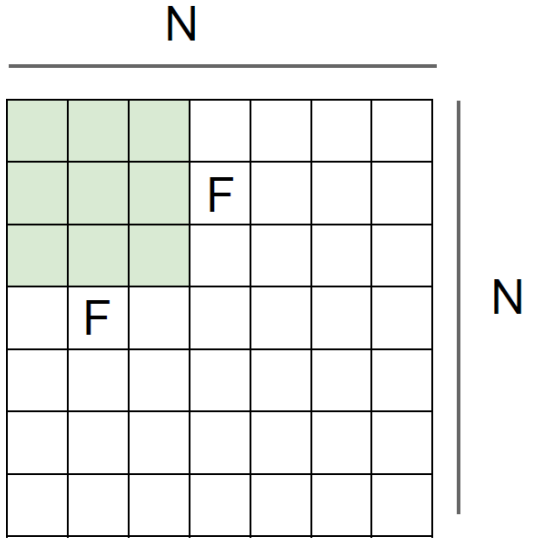


Filtar 2x2
Stride 2

Kolika je veličina rezultantne aktivacijske mape?

Konvolucijski sloj u detalje

- Uobičajeno je veličina filtera jednaka po obje koordinate i iznosi F
- Uz pretpostavku da ulazni volumen dimenzije NxN, veličina izlazne mape jednaka je (N-F) /stride + 1
- Ograničen broj kombinacija (stride, veličina filtera i veličina ulazne slike) jer veličina izlazne mape mora biti cijeli broj



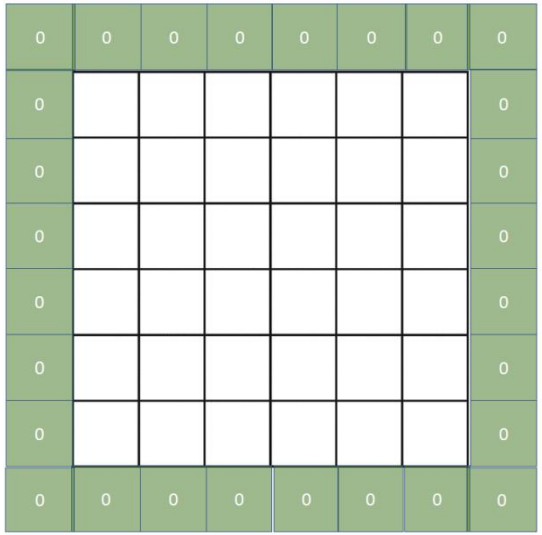
Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33

Zero padding

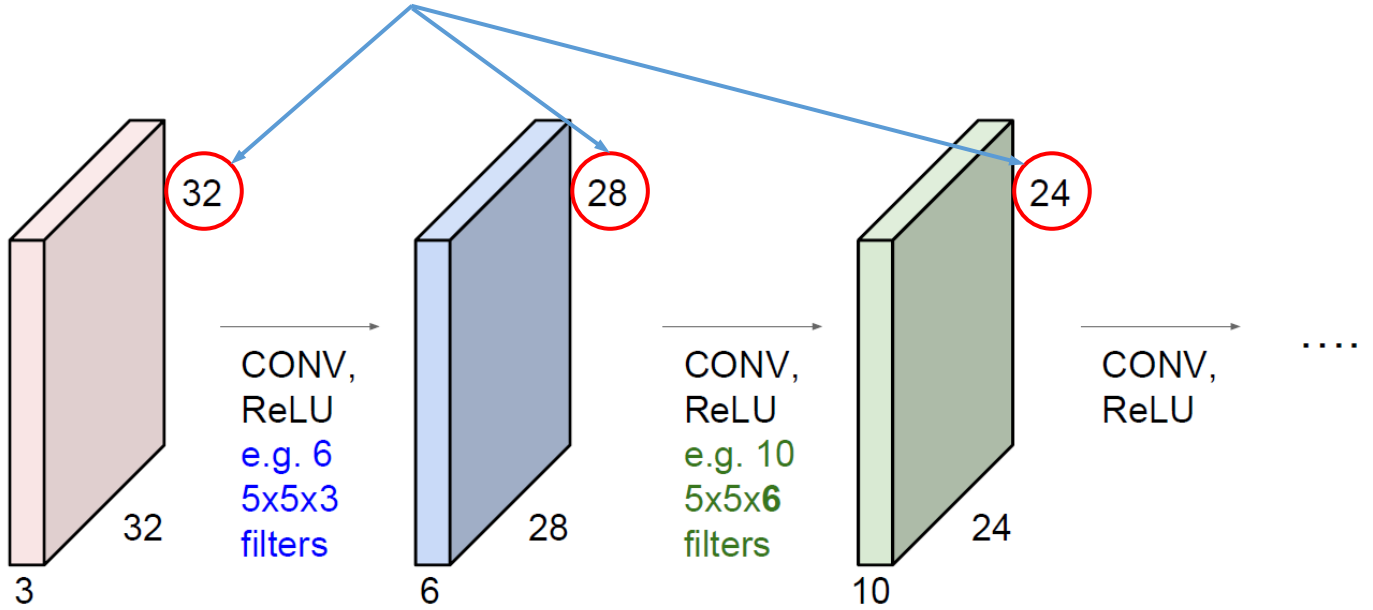
- Obično se dodaje rub koji sadrži nule → na taj način se primjenom konvolucijskih slojeva ne gubi na prostornoj veličini
- Također, dodaje se u slučajevima kada filter ne prolazi simetrično cijeli ulazni volumen

Zero pad the border



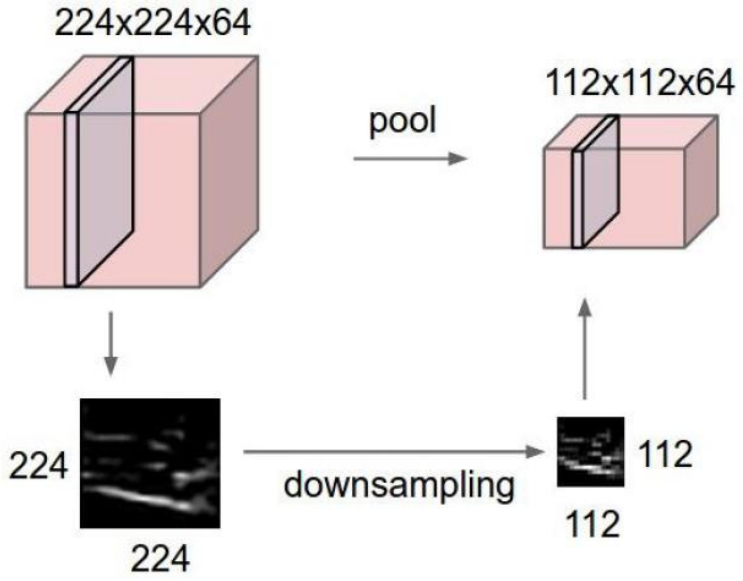
U praksi:
F = 3 i zero pad jednak 1
F = 5 i zero pad jednak 2
F =7 i zero pad jednak 3

Podsjetnik: sekvencijalnom se primjenom konvolucijskih filtara gubi prostorna veličina



Sloj sažimanja

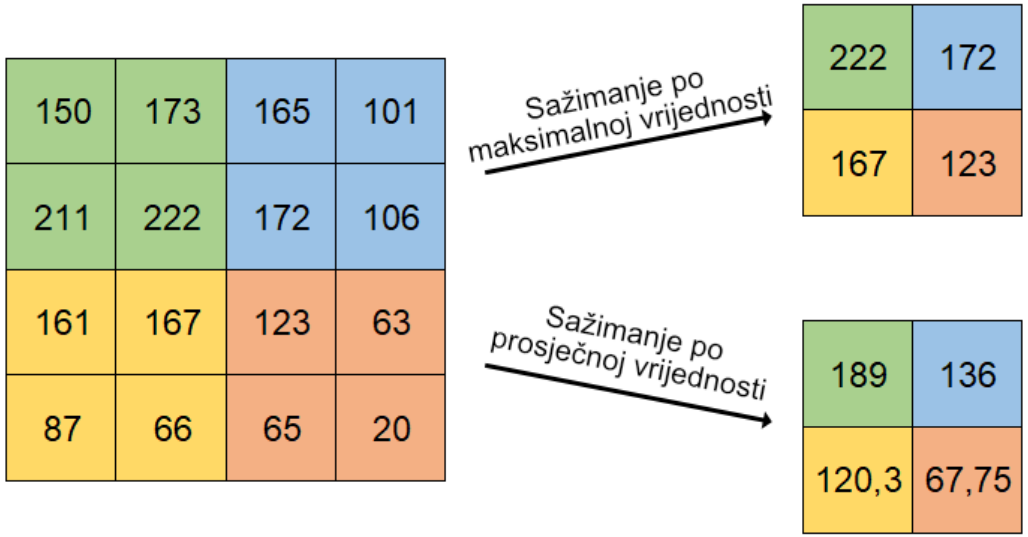
- Kako bi se reprezentacija slike držala što manjom, uvode se slojevi sažimanja (engl. pooling layers)
- Ovo je operator koji se primjenjuje na svaku aktivacijsku mapu zasebno
- Praktički ovime se provodi downsampling (smanjivanje rezolucije) svake pojedine aktivacijske mape



„Pooling in convolution network is designed to filter noisy activations in a lower layer by abstracting activations in a receptive field with a single representative value.”

Sloj sažimanja

- Sloj sažimanja uzima predefiniranu veličinu podmatrice i koristi jednu od operacija kako bi sažeo podatke
- Najčešći oblik sloja sažimanja je po maksimalnoj vrijednosti (engl. max pool layer)
- Ovaj sloj nema parametre koji se određuju postupkom učenja



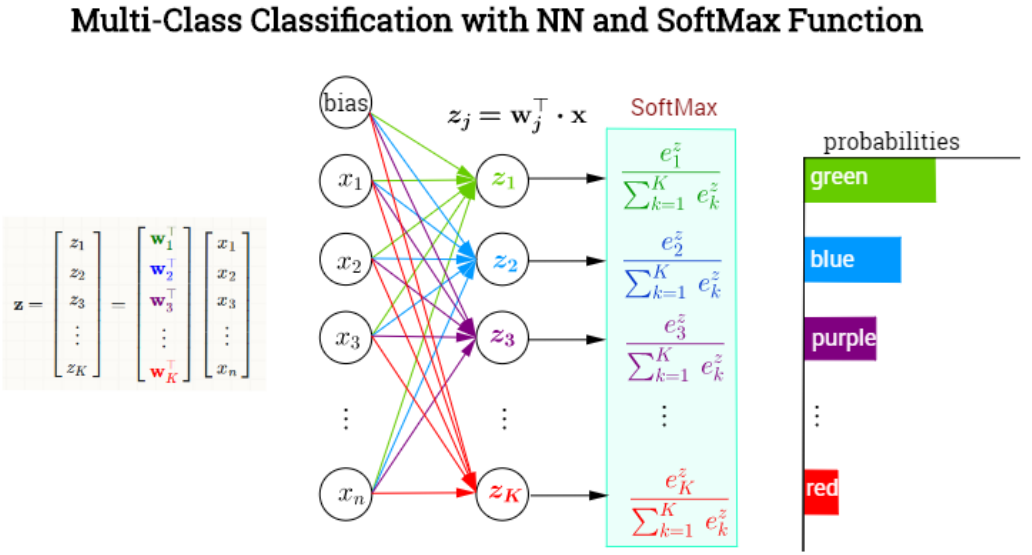
Prikaz operacije sažimanja s veličinom podmatrice 2×2 i pomakom 2

Softmax sloj

- Softmax sloj zapravo je potpuno povezani sloj sa softmax aktivacijskom funkcijom i koristi se kao izlazni sloj kod mreže kojoj je namjena višeklasna (engl. multiclass) klasifikacija
- Broj izlaznih neurona neke mreže koja obavlja višeklasnu klasifikaciju, tj. broj neurona softmax sloja, jednak je broju klasa iz skupa podataka na kojemu se mreža uči
- Softmax sloj kao izlazni sloj neke mreže određuje vjerojatnosti da objekt koji se klasificira pripada pojedinoj klasi, a ukupan zbroj vjerojatnosti uvijek je jednak jedan
- Npr. mreža s tri izlazna neurona (K=3):

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

$$\begin{bmatrix} 1.2 \\ 0.9 \\ 0.4 \end{bmatrix} \rightarrow \boxed{\text{Softmax}} \rightarrow \begin{bmatrix} 0.46 \\ 0.34 \\ 0.20 \end{bmatrix}$$

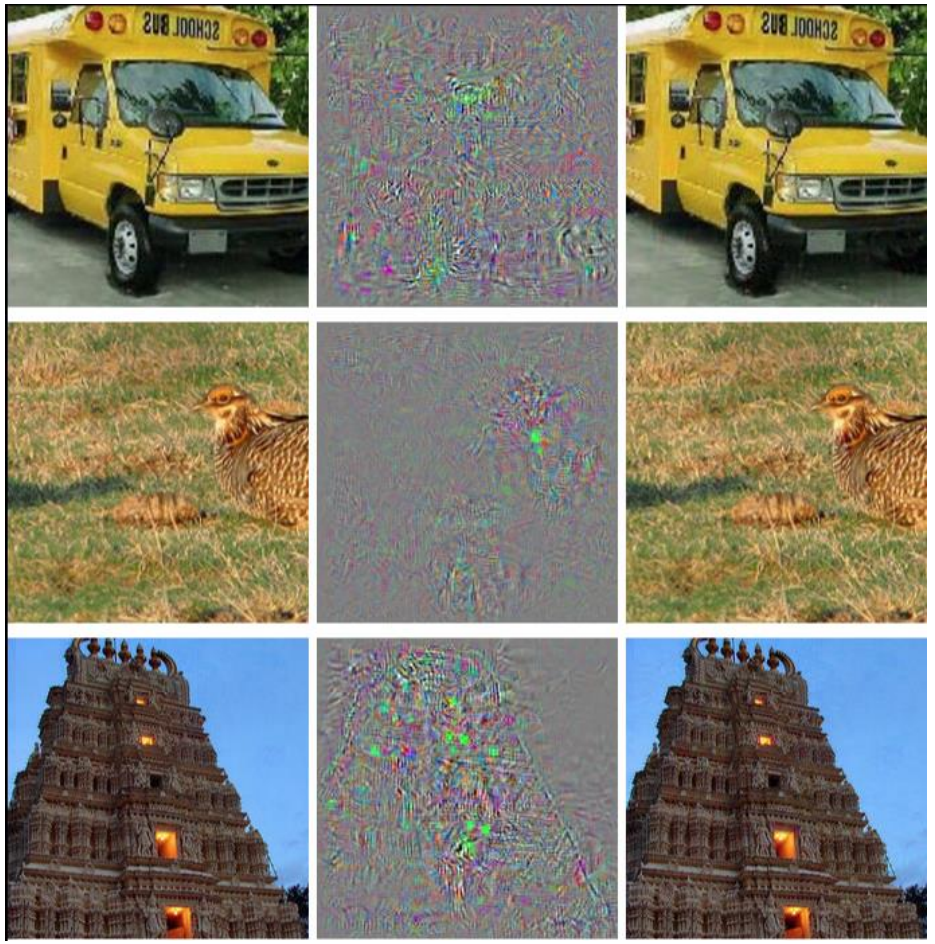


Zadatak

- Otvorite CNN_zadatak.pdf
- Odredite dimenzije svakog dijela mreže (označeno s ?)

Nedostatci CNN

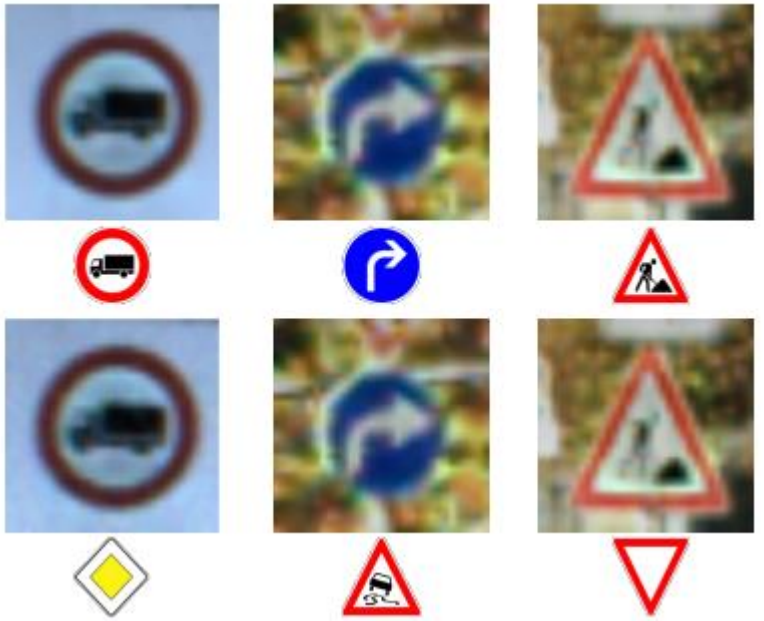
- Lako ih je „prevariti”
- Adversarial example –> „optička iluzija za ML algoritam”
- Lijeva slika je originalna slika → CNN pravilno klasificira
- Desna slika je lijeva slika + precizna distorzija → CNN pogrešno klasificira
- I ljudi imaju ovaj problem, ali ne u mjeri koja je prisutna kod CNN



Izvor: Intriguing properties of neural networks

„Neprijateljski” primjeri u autonomnoj vožnji

- Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, Ananthram Swami. Practical Black-Box Attacks against Machine Learning 2017.
- multiclass DNN classifier

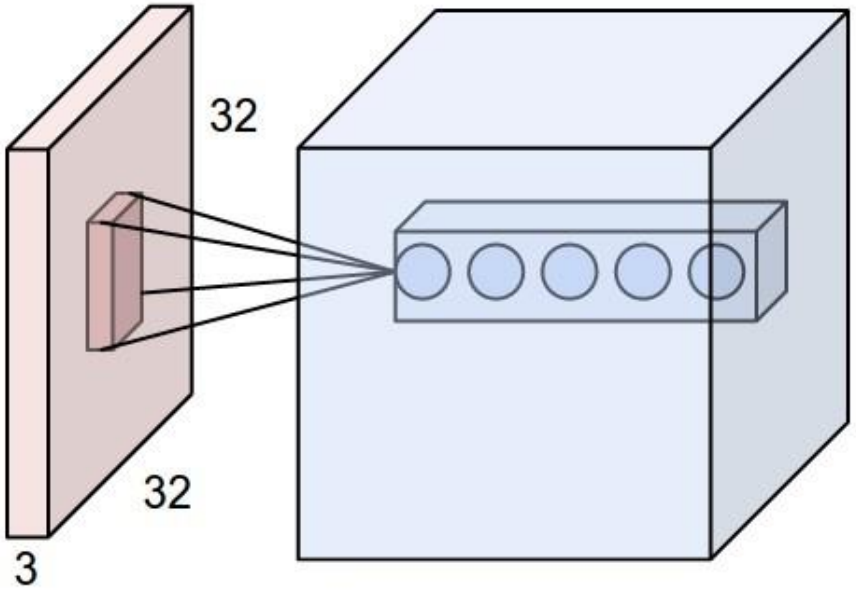


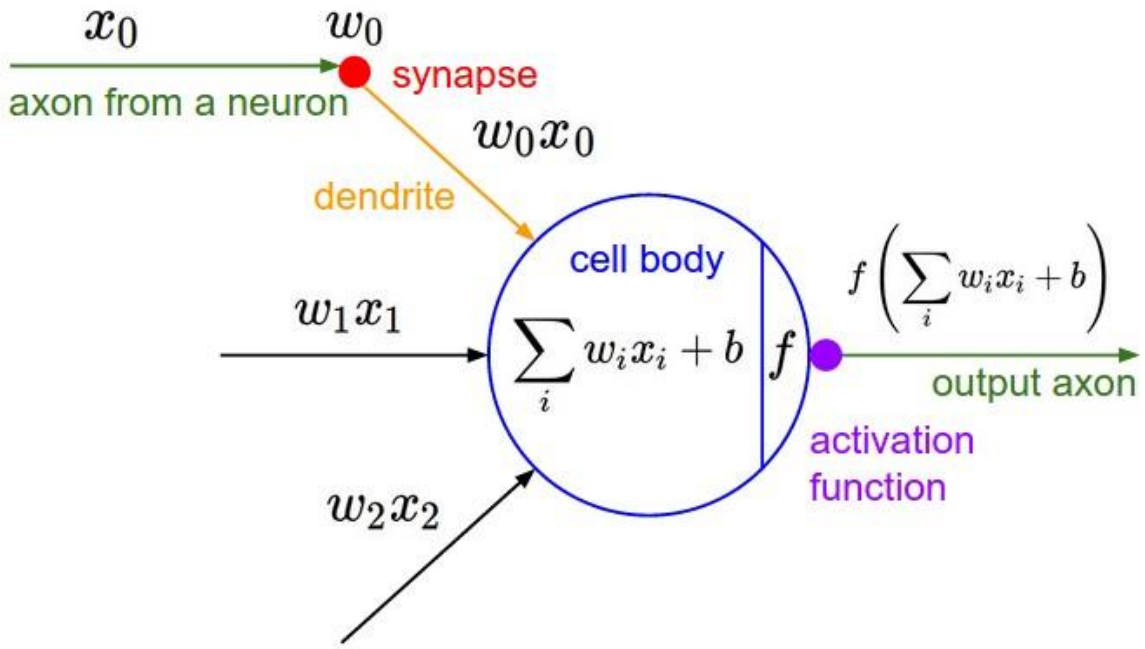
„It is thus conceivable that physical adversarial traffic signs could be generated by maliciously modifying the sign itself, e.g., with stickers or paint.”

Učenje neuronske mreže

- Učenje neuronske mreže podrazumijeva različite korake
  - Priprema skupova podataka za učenje, validaciju i testiranje
  - Odabir strukture neuronske mreže (slojevi, broj neurona, aktivacijske funkcije)
  - Inicijalizacija parametara mreže
  - Predobrada podataka
  - Kriterijska funkcija
  - Regularizacija (kriterijska funkcija, dropout, …)
  - Batch normalizacija
  - Podešavanje različitih hiperparametara (stope učenja, mijenjanje strukture mreže, veličine batch-a, parametra regularizacije, …)
  - …

Odabir aktivacijske funkcije



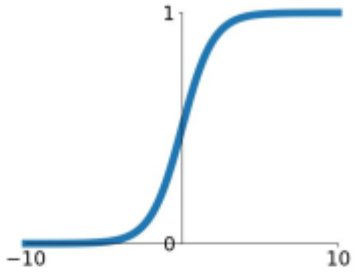


Izvor

Odabir aktivacijske funkcije

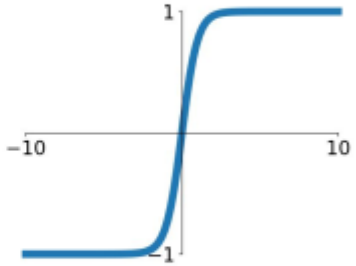
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



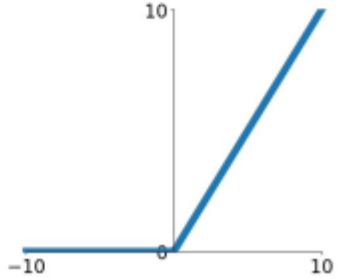
tanh

$$\tanh(x)$$



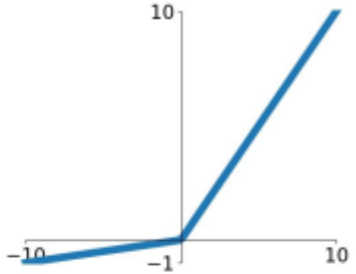
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

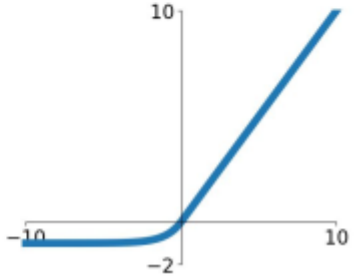


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Odabir aktivacijske funkcije

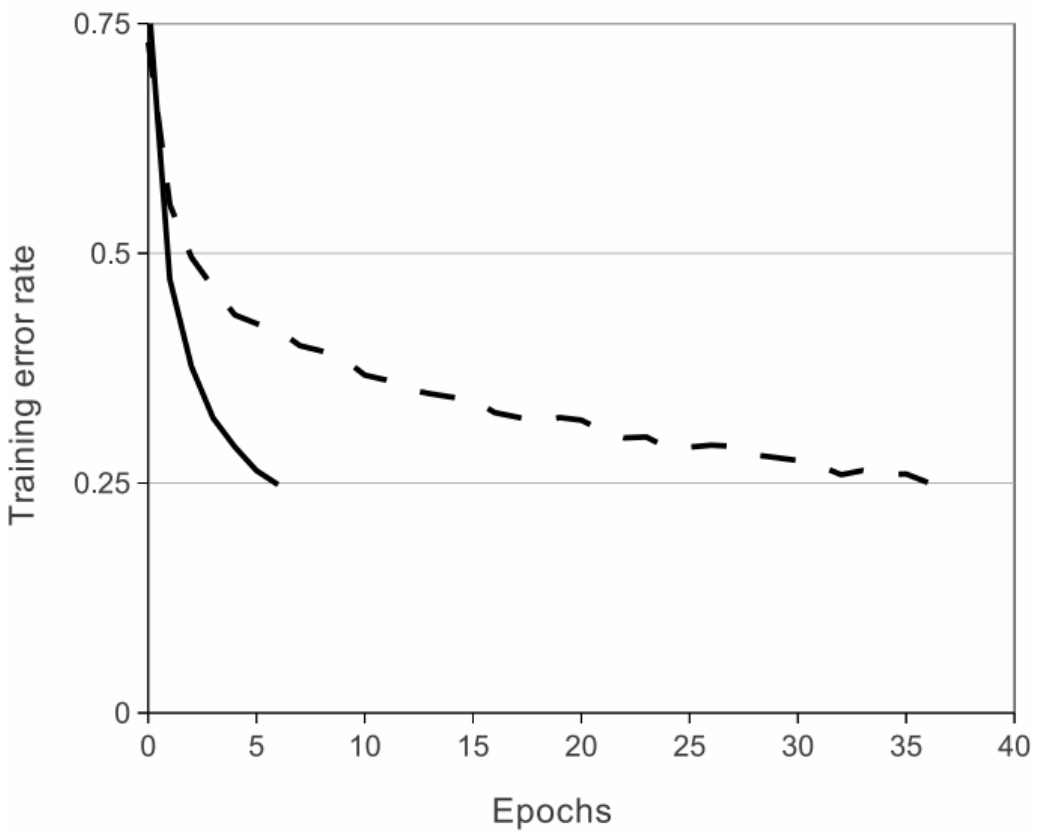
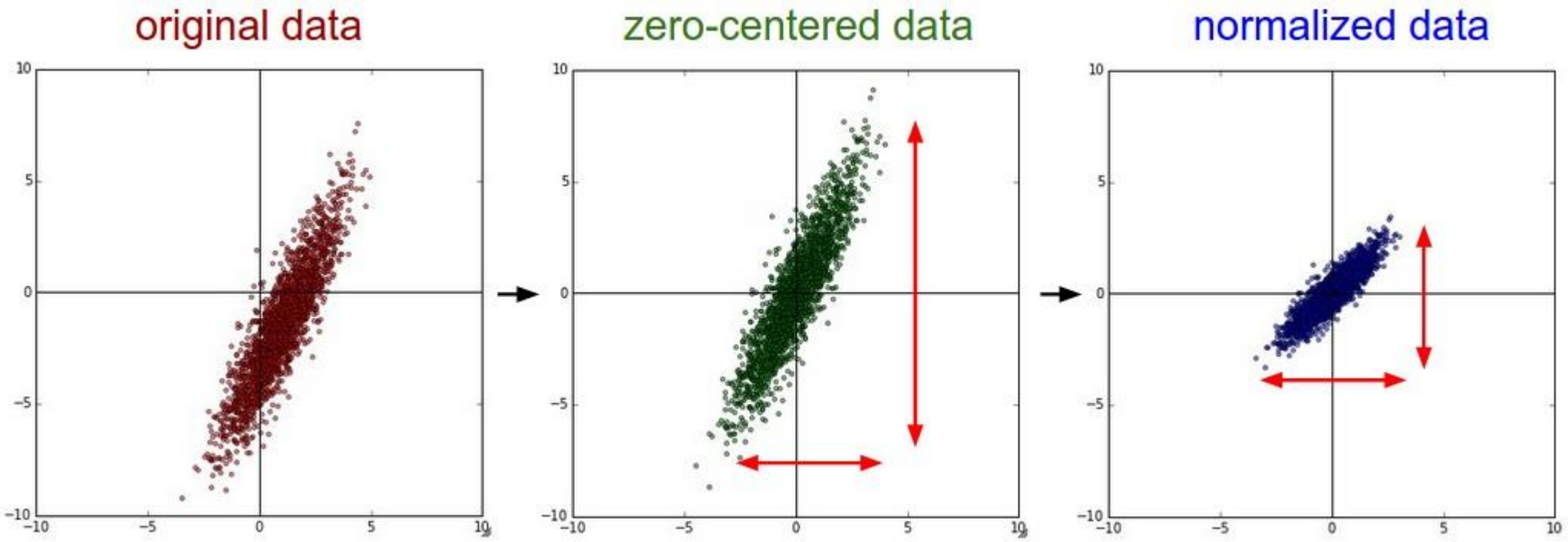


Figure 1: A four-layer convolutional neural network with ReLUs **(solid line)** reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons **(dashed line)**. The learning rates for each network were chosen independently to make training as fast as possible. No regularization of any kind was employed. The magnitude of the effect demonstrated here varies with network architecture, but networks with ReLUs consistently learn several times faster than equivalents with saturating neurons.

Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks, 2012.

Predobrada podataka

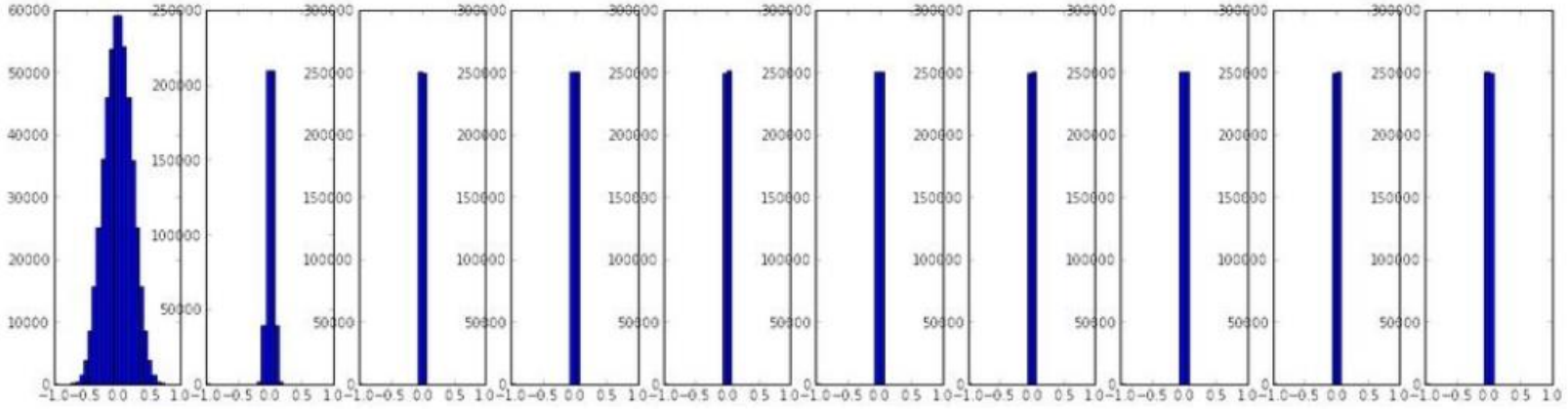


Kod slika samo se centriraju podaci (efikasniji postupak optimizacije parametara)
Ne radi se skaliranje (jer pikseli imaju isti "dimenziju", npr. tri kanala 0-255)

-oduzimanje "srednje slike" (e.g. AlexNet)
(srednja vrijednost svih slika , npr. [32,32,3] polje)
- oduzimanje srednje vrijednosti kanala (e.g. VGGNet)
(srednja vrijednost po svakom kanalu, npr. 3 broja)

Inicijalizacija parametara mreže

- Za plitke mreže uobičajeno je inicijalizirati parametre mreže da imaju srednju vrijednost 0 i neku malu standardnu devijaciju (npr. 0.01)
- Međutim, kod dubokih mreža ovakav način uzrokuje probleme prilikom optimizacije
- Primjer: mreža od 10 slojeva, 500 neurona u svakom sloju, s tanh aktivacijskim funkcijama



Backward pass: ovisi o vrijednostima ulaza za svaki neuron → zbog malih vrijednosti sporo se podešavaju parametri (gradijent je jako mali, engl. *vanishing gradient problem*)

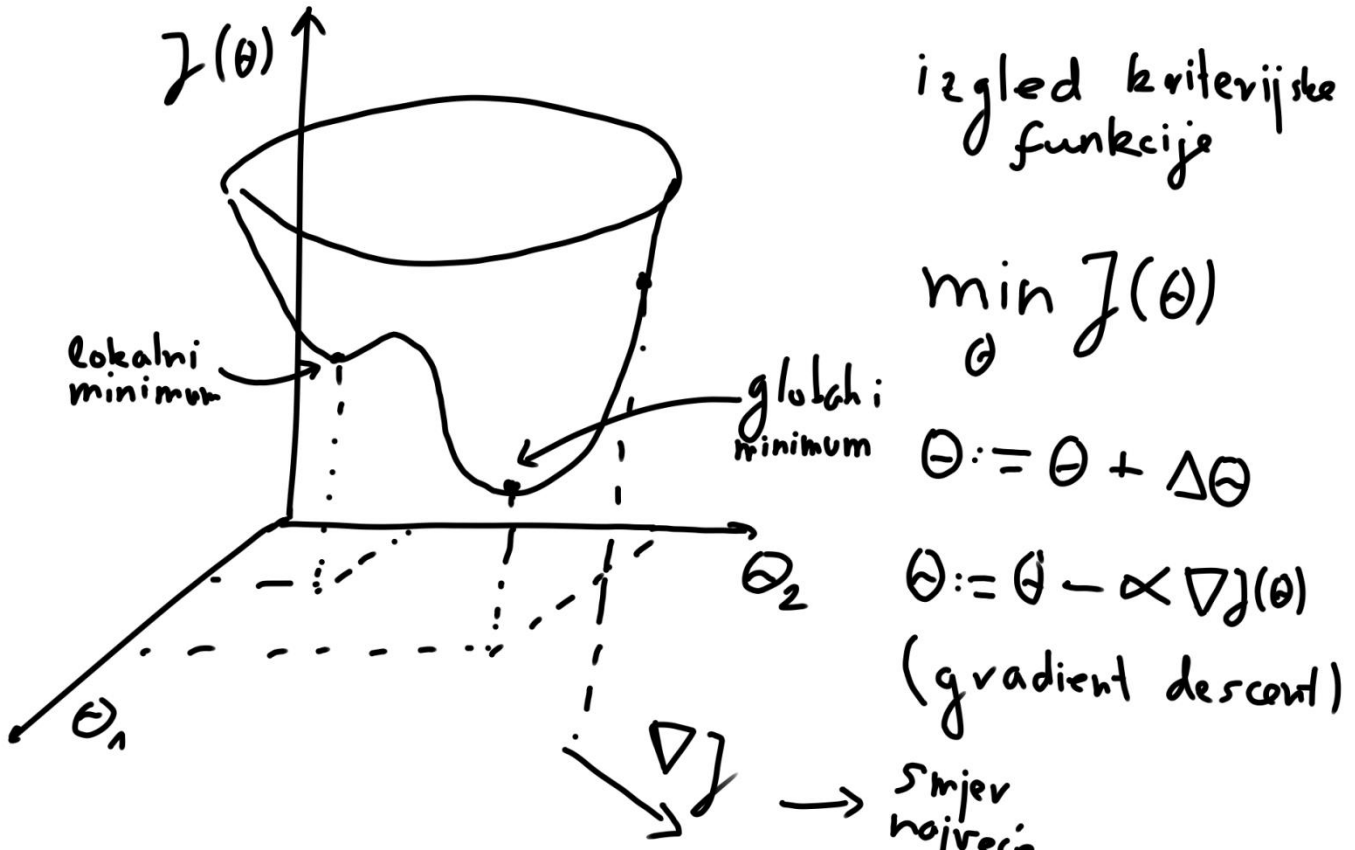
Propustimo nasumični podatak kroz mrežu (forward pass)

Statistika izlaza iz neurona (aktivacija) za svaki sloj

Brzo se smanjuje devijacija izlaza
(teže u 0)

Uzrok: množenje malih brojeva

Učenje mreže



izgled kriterijske funkcije

$$\min_{\theta} J(\theta)$$

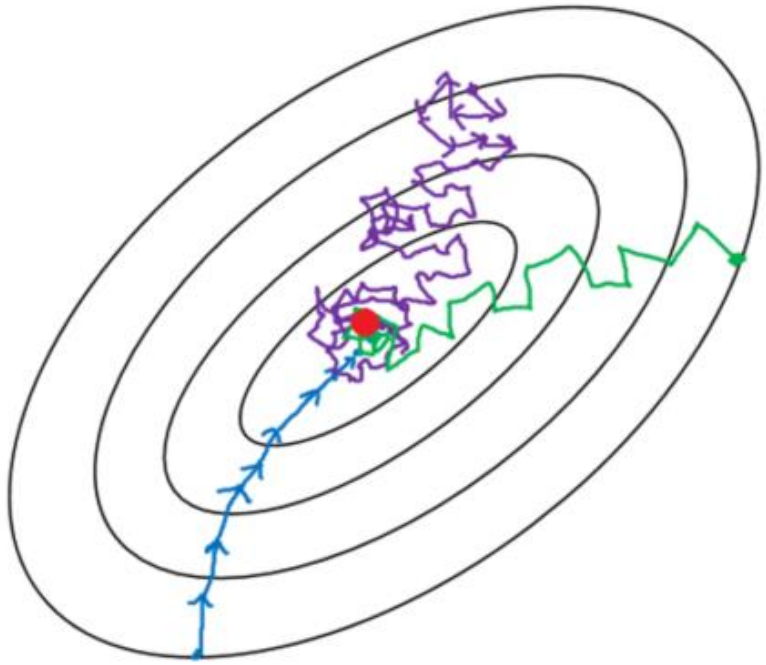
$$\theta := \theta + \Delta\theta$$

$$\theta := \theta - \alpha \nabla J(\theta)$$

(gradient descent)

Gradijentna metoda

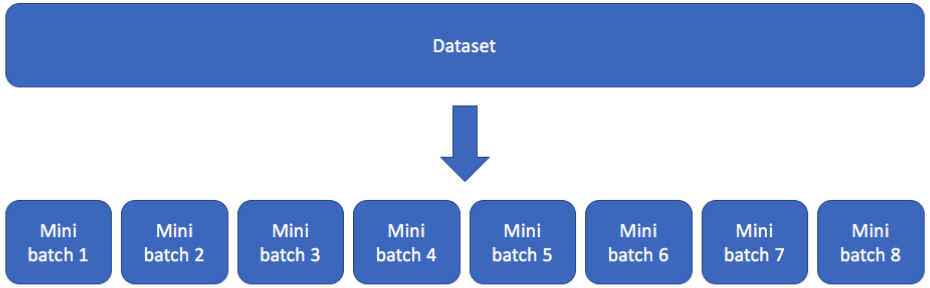
- Postupak optimizacije (minimizacija kriterijske funkcije)



Batch gradient descent
Mini-batch gradient Descent
Stochastic gradient descent

Izvor

Mini batches



Veličina batch-a

- Dominic Masters, Carlo Luschi, Revisiting Small Batch Training for Deep Neural Networks, 2018.

The collected experimental results for the CIFAR-10, CIFAR-100 and ImageNet datasets show that increasing the mini-batch size progressively reduces the range of learning rates that provide stable convergence and acceptable test performance. On the other hand, small mini-batch sizes provide more up-to-date gradient calculations, which yields more stable and reliable training.

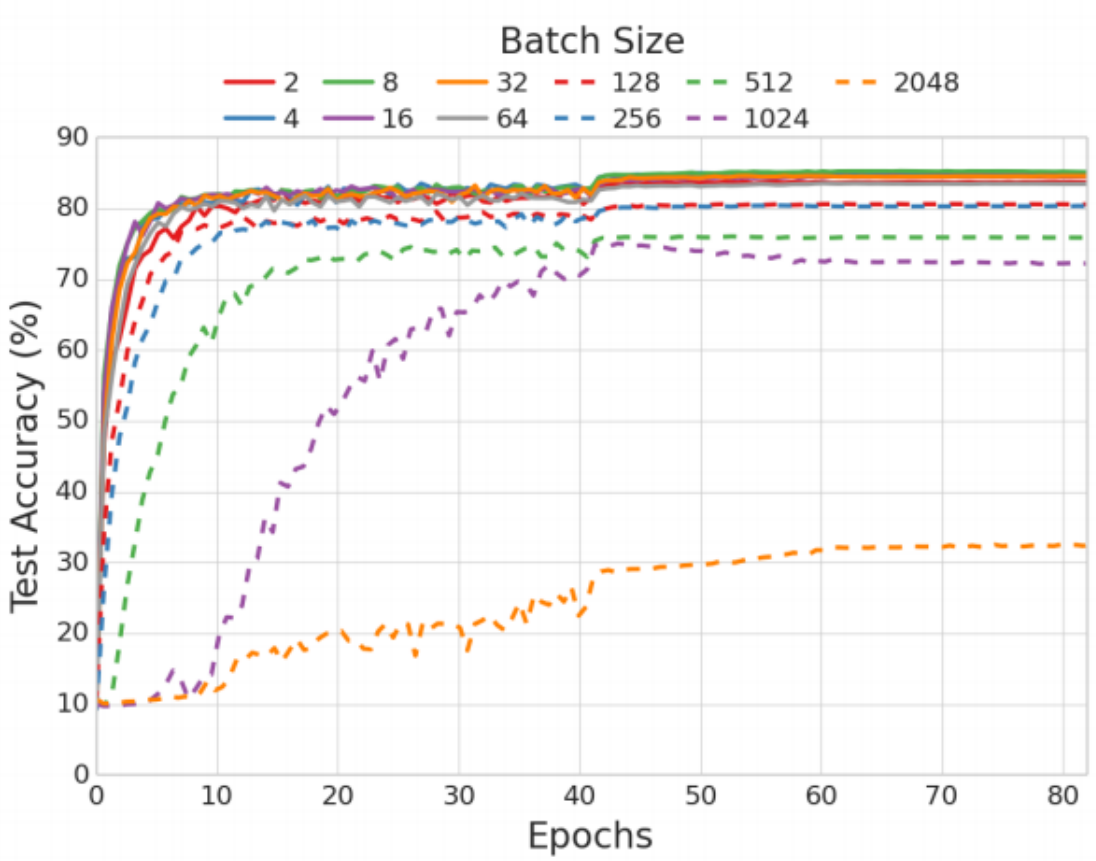


Figure 4: Convergence curves for ResNet-32 model with BN, for $\tilde{\eta} = 2^{-8}$ and for different values of the batch size. CIFAR-10 dataset without data augmentation.

Metode optimizacije

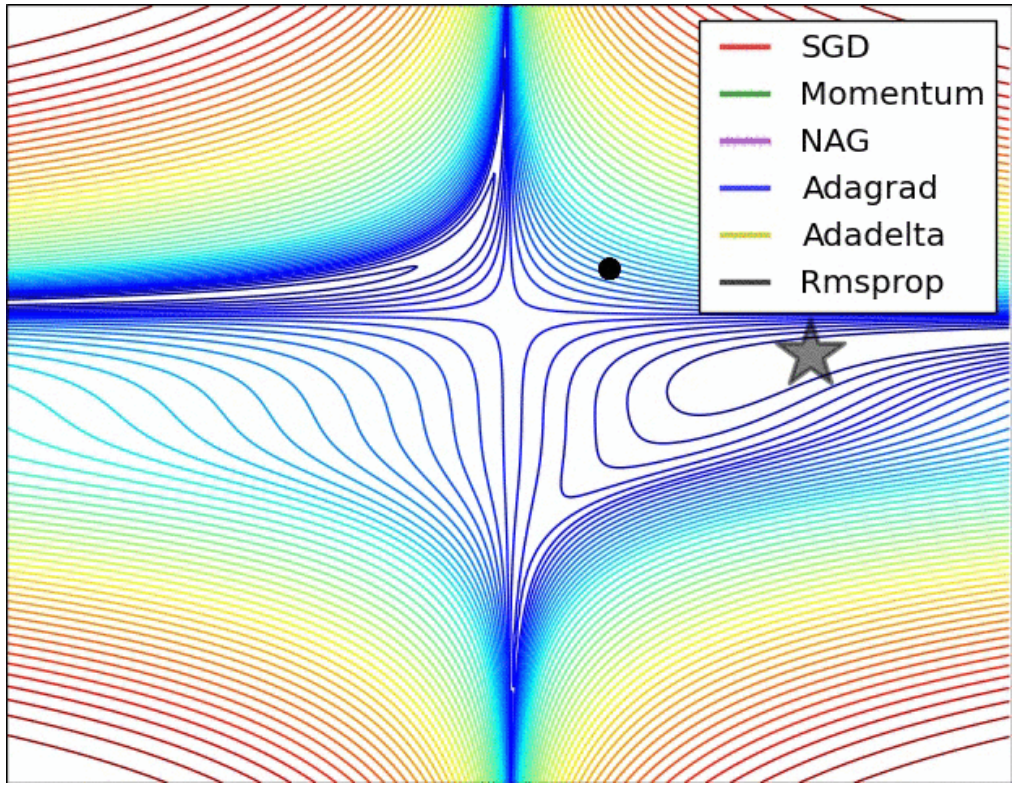
- SGD
- SGD s momentom
- Adagrad
- Adadelta
- RMSprop
- Adam…

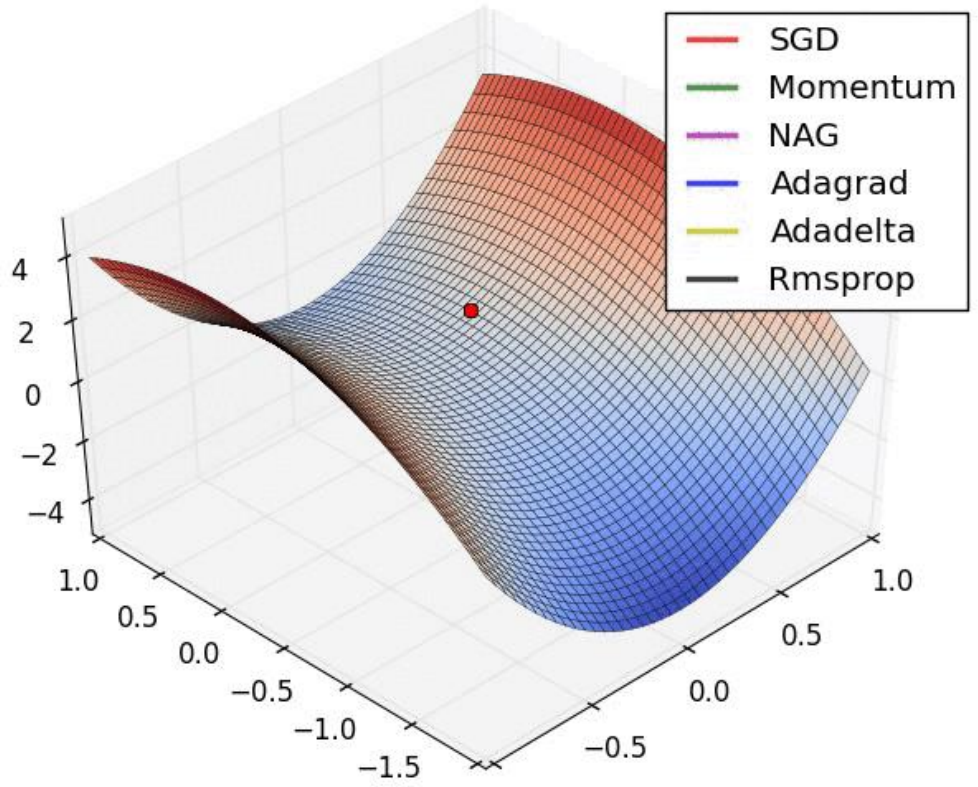
- Više infromacija na:

https://ruder.io/optimizing-gradient-descent/

Metode optimizacije

- Ponašanje blizu sedla





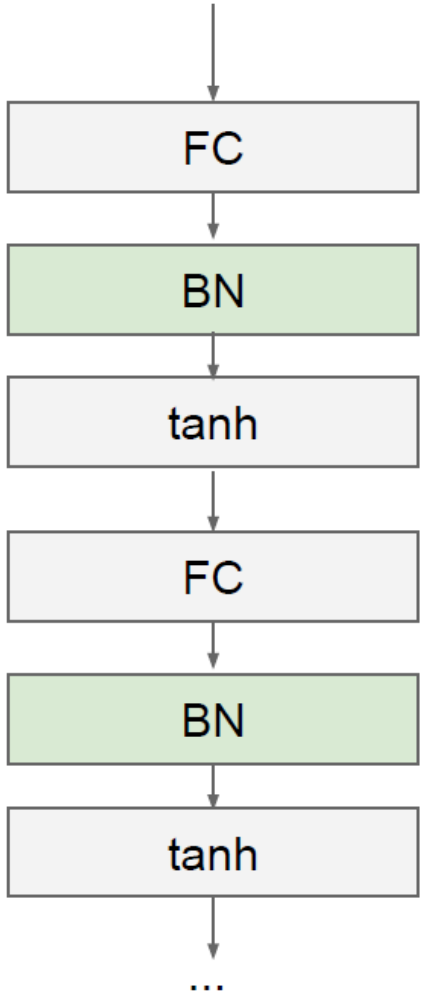
Batch normalization

- Sergey Ioffe, Christian Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015.
- Tijekom učenja mreže, distribucija ulaza svakog sloja se mijenja jer se mijenjaju parametri prethodnih slojeva nakon svakog mini batcha → ovo usporava proces učenja (potrebno je koristiti manju stopu učenja i pažljivo inicijalizirati parametre mreže)
- Rješenje: normalizacija ulaza pojedinog sloja tijekom učenja (podaci imaju srednju vrijednost 0 i std.devijaciju jednaku 1)

"Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin."

Very deep models involve the composition of several functions or layers. The gradient tells how to update each parameter, under the assumption that the other layers do not change. In practice, we update all of the layers simultaneously.

Batch normalization



- Pospješuje tok gradijenata kroz mrežu (stabilnost učenja)
- Omogućuje korištenje većeg learning rate-a (ubrzava učenje)
- Postupak učenja robusniji na lošu inicijalizaciju
- Može se shvatiti i kao „slaba” regularizacija (unosi šum u aktivacije svakog sloja) → sprječava overfitting

- Obično se prati srednja vrijednost i std. tijekom procesa učenja u obliku „running mean”
- Te vrijednosti se kasnije koriste za normalizaciju aktivacija

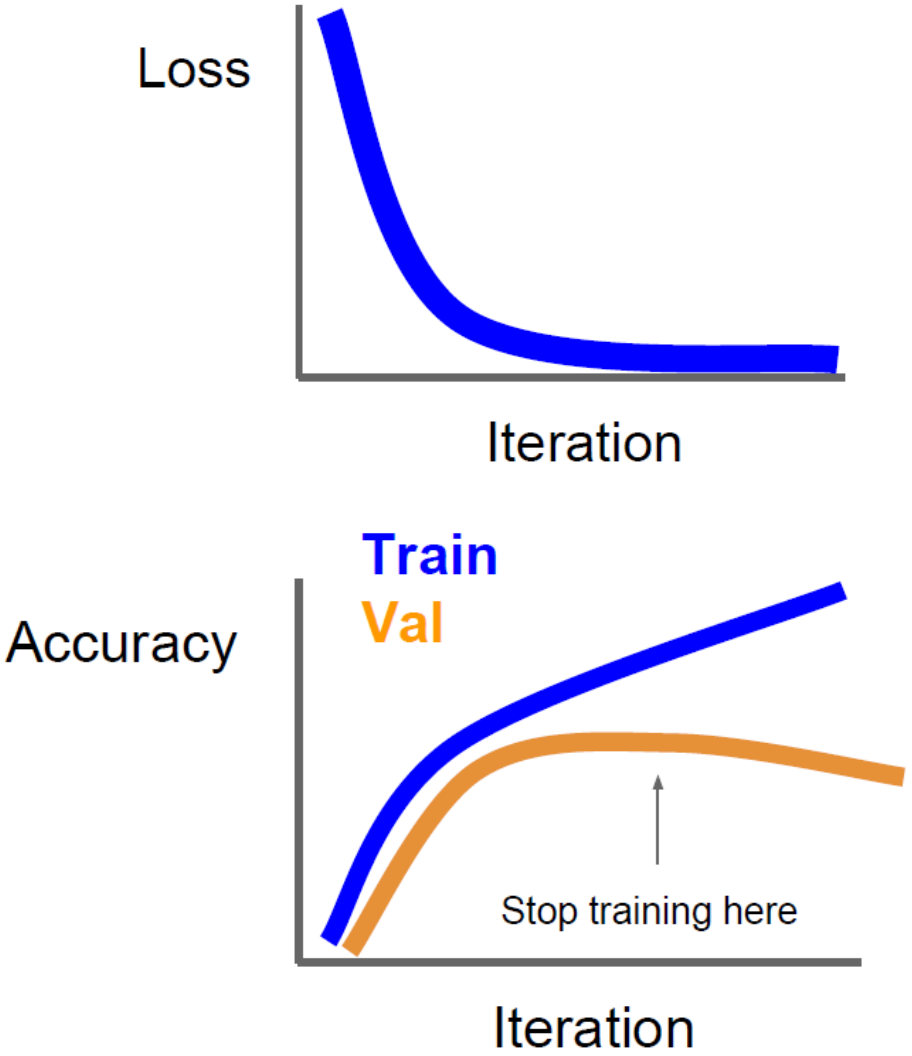
Regularizacija

- Kako unaprijediti performanse modela na „neviđenim“ podacima?
- Neuronske mreže imaju velik broj parametara (najčešće milijune), skup podataka je obično značajno manji
- Spriječiti pretjerano usklađivanje na podatke za učenje moguće je primjenom regularizacije:
  - L2 regularizacija
  - L1 regularizacija
  - Droput
- L2 regularizacija je najčešći tip regularizacije (često se naziva i weigh decay); parametar λ obično iznosi oko 0.01

$$w^* \leftarrow \arg\min_w \sum_{(x,y)\subseteq(X,Y)} \mathcal{L}(y, a_L(x; w_{1,\ldots,L})) + \frac{\lambda}{2}\sum_l \|w_l\|^2$$

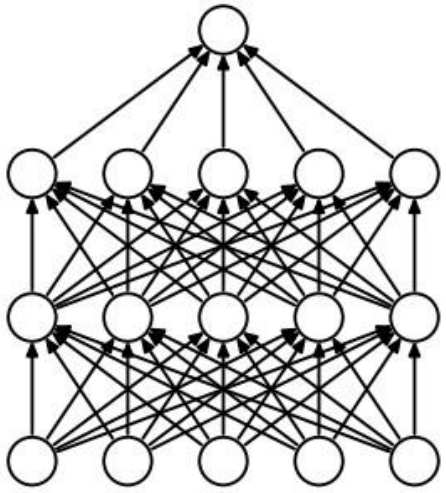
Rano zaustavljanje

- Drugi princip sprječavanja overfittinga
- Prate se performanse modela na zasebnom, validacijskom skupu
- Učenjem mreže smanjuje se pogreška na skupu za učenje kao i na validacijskom skupu (obično nešto sporije); tj. raste preciznost
- Zaustaviti proces učenja kada pogreška na validacijskom skupu počinje rasti (pretpostavljamo da mreža počinje overfittati skup za učenje)
- Ili učiti mrežu duže vrijeme, ali spremati slike modela (kako bi mogli izvući onaj koji je imao najmanju pogrešku na validacijskom skupu)

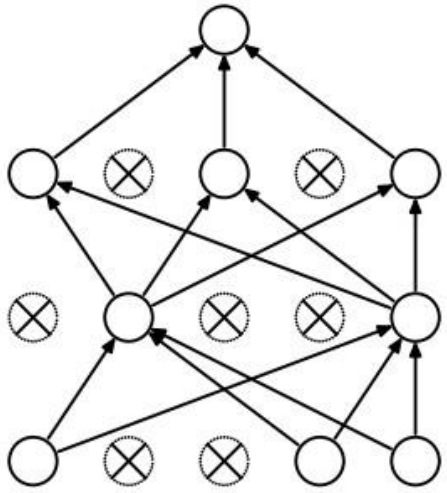


Dropout

- N. Srivastava et all. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, 2014.
- Vrlo efikasna i jednostavna tehnika regularizacije
- „The key idea is to randomly drop units (along with their connections) from the neural network during training.”
- „This significantly reduces overfitting and gives major improvements over other regularization methods”



(a) Standard Neural Net



(b) After applying dropout.

Tipično je vjerojatnost izbacivanja neurona jednaka 0.5

Znatno ubrzanje

Dodaje se slučajna varijabla koja sprječava pretjerano usklađivanje tijekom učenja.

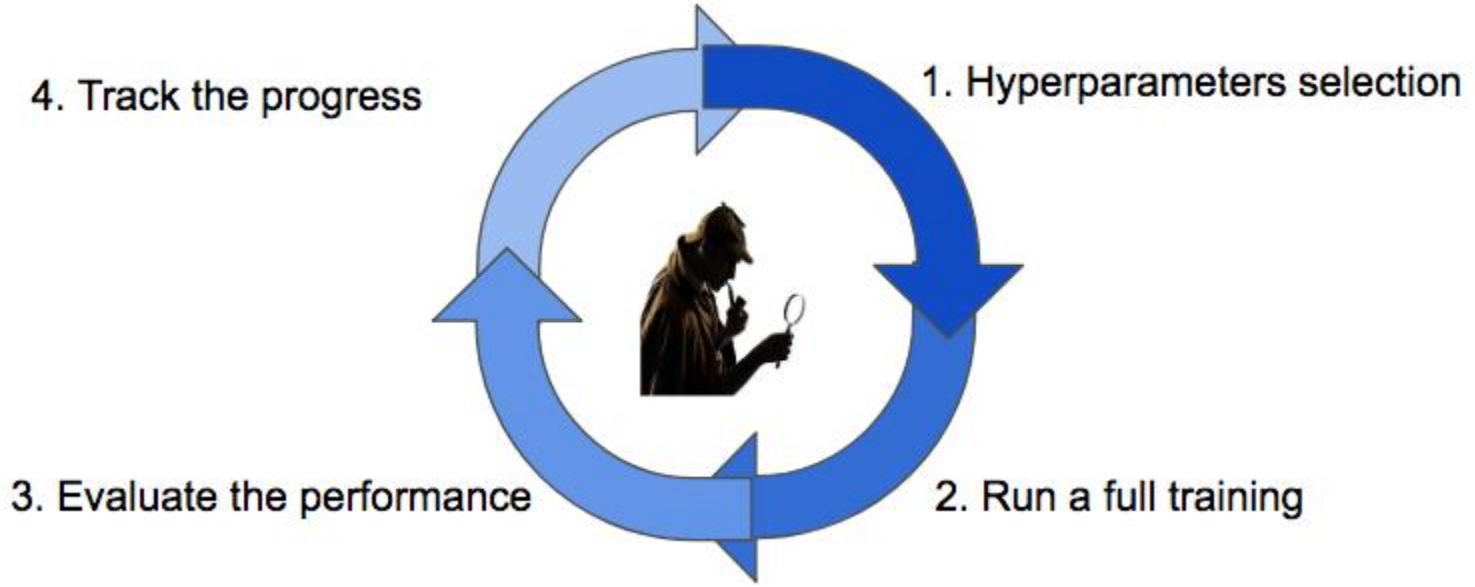
Robusniji neuroni, sprječava co-adaptaciju (node interactions).

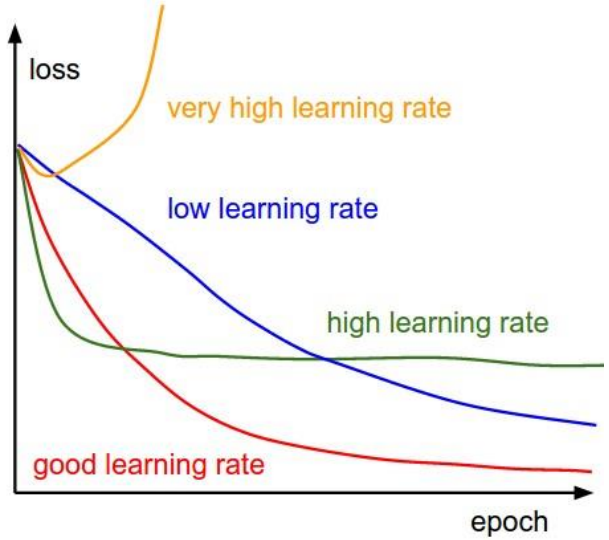
Augmentacija skupa za učenje

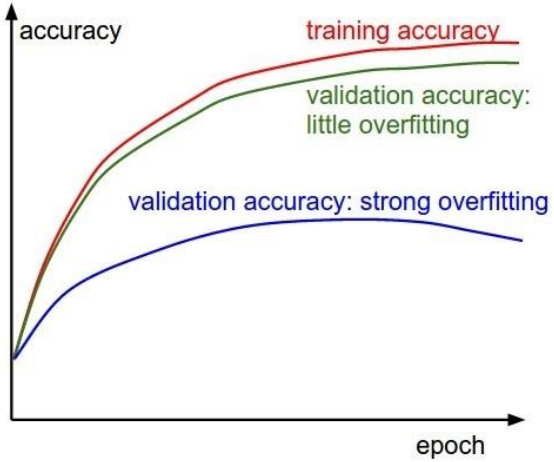
- Jedan od načina sprječavanja overfittinga; poboljšava generalizacijske sposobnosti mreže
- Podrazumijeva nasumične transformacije slika iz skupa za učenje pri čemu se zadržava oznaka (labela) slike:
  - Horizontal flip
  - Crop
  - Rotacije
  - Translacije
  - Uvećavanje
  - Mijenjanje osvjetljenja, kontrasta ili boje
  - ...

- Transformacije kakve bi se mogle pojaviti u pra

Tijek procesa učenja







Veliki „razmak” između train i val preciznosti → overfitting

Nema „razmaka” između train i val preciznosti → možebitna premala složenost modela

Transfer learning

- CNN mogu imati milijune parametara
- Naš raspoloživi skup podataka (slika) nije uvijek velik
- Postavlja se pitanje mogu li se CNN naučiti na takvim skupovima bez problema pretjeranog usklađivanja na podatke (engl. overfitting)?
- Jedan od načina učenja CNN mreže je transfer learning koji omogućuje učenje CNN mreže premda naš skup podataka nije pretjerano velik
- Pretpostavimo da imamo na raspolaganju dva skupa podataka
- Prvi skup je potpuno označen, ima velik broj slika
- Drugi skup slika je sličan, ali sadrži znatno manje slika
- Oznake ne moraju biti jednake u oba skupa!
- Na temelju prvog skupa izradi se bazni model
- Na temelju bazni model i drugog skupa izgradi se konačni model

Transfer learning

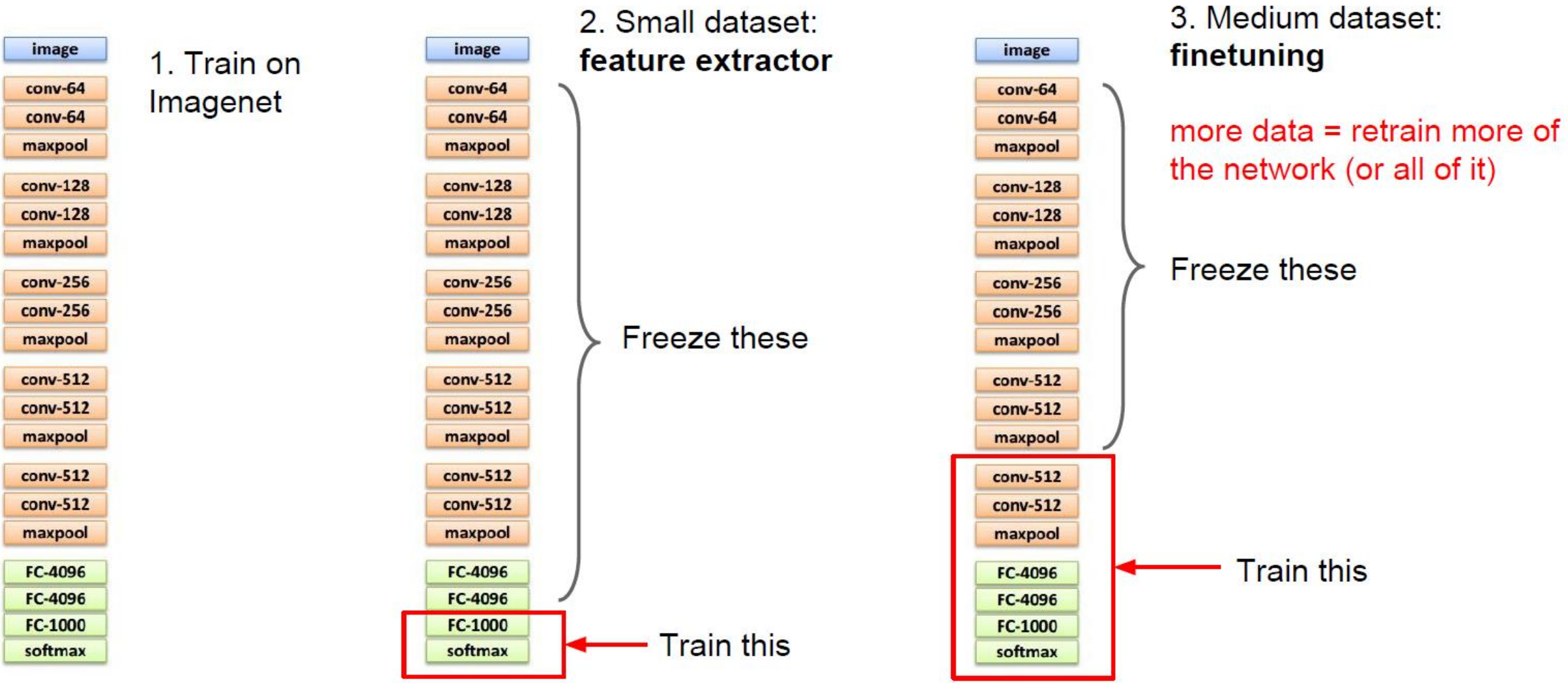
Imagenet: 1million

My dataset: 1,000

Bazni model

Konačni model

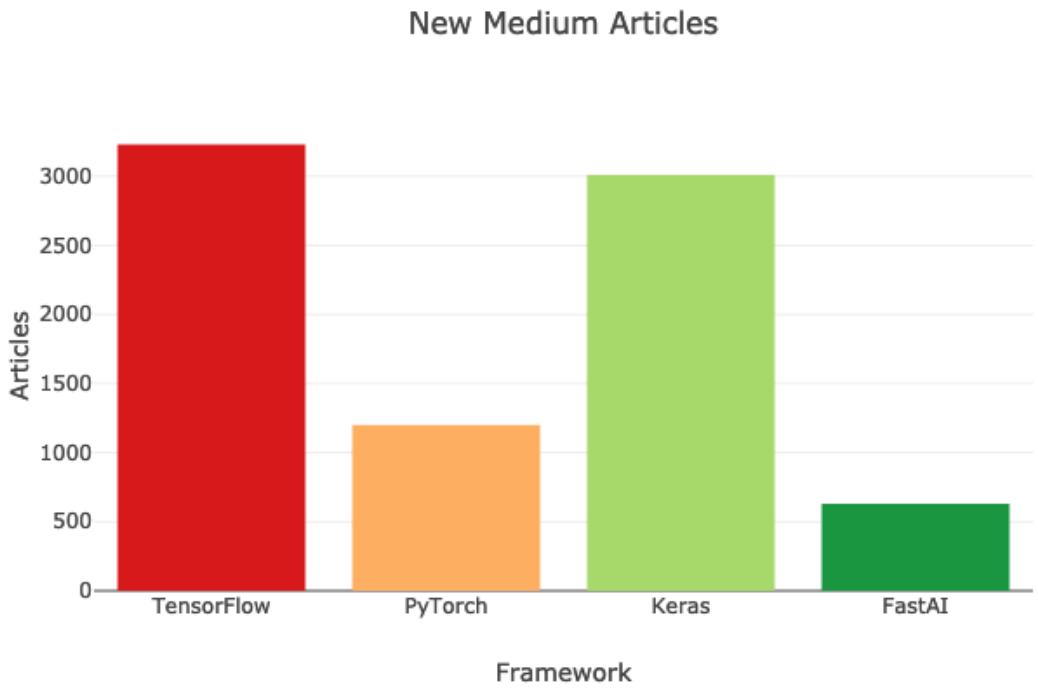
Transfer learning

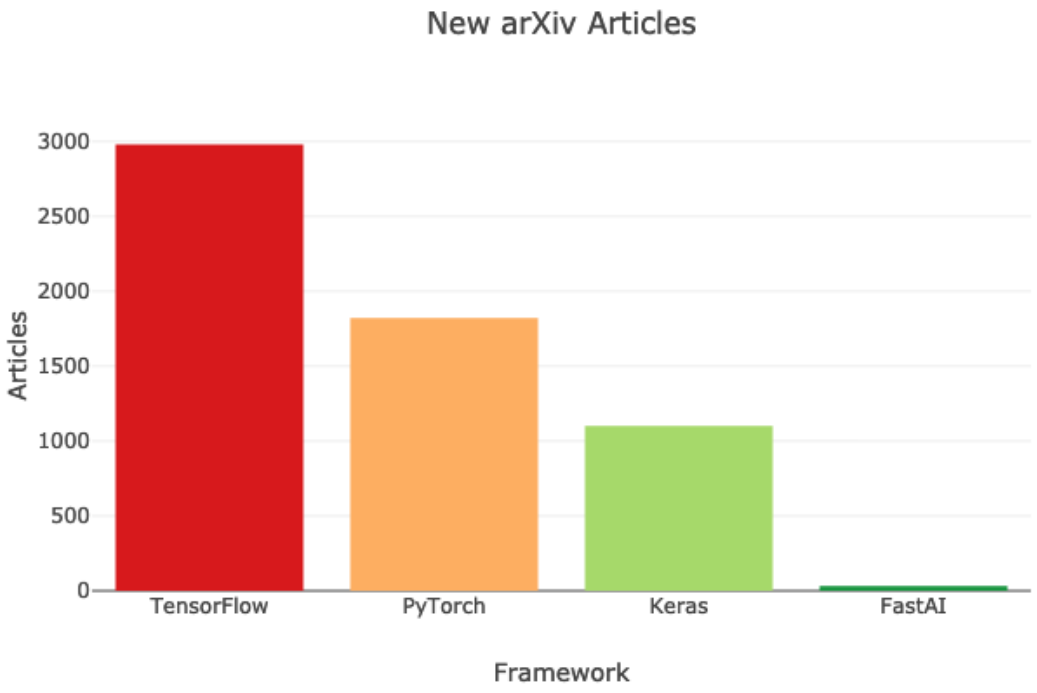


Biblioteke za duboko učenje

- trenutno su vrlo popularni: TensorFlow, Pytorch i high level API (Keras i fastai)

Izvor





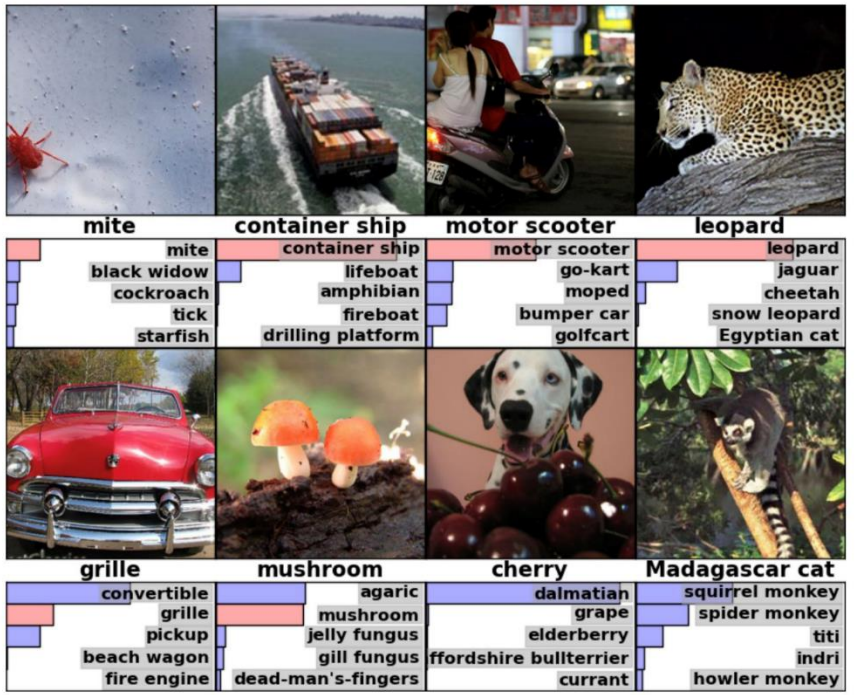
Klasifikacija slika

- ImageNet challenge – predstavljen 2009. godine

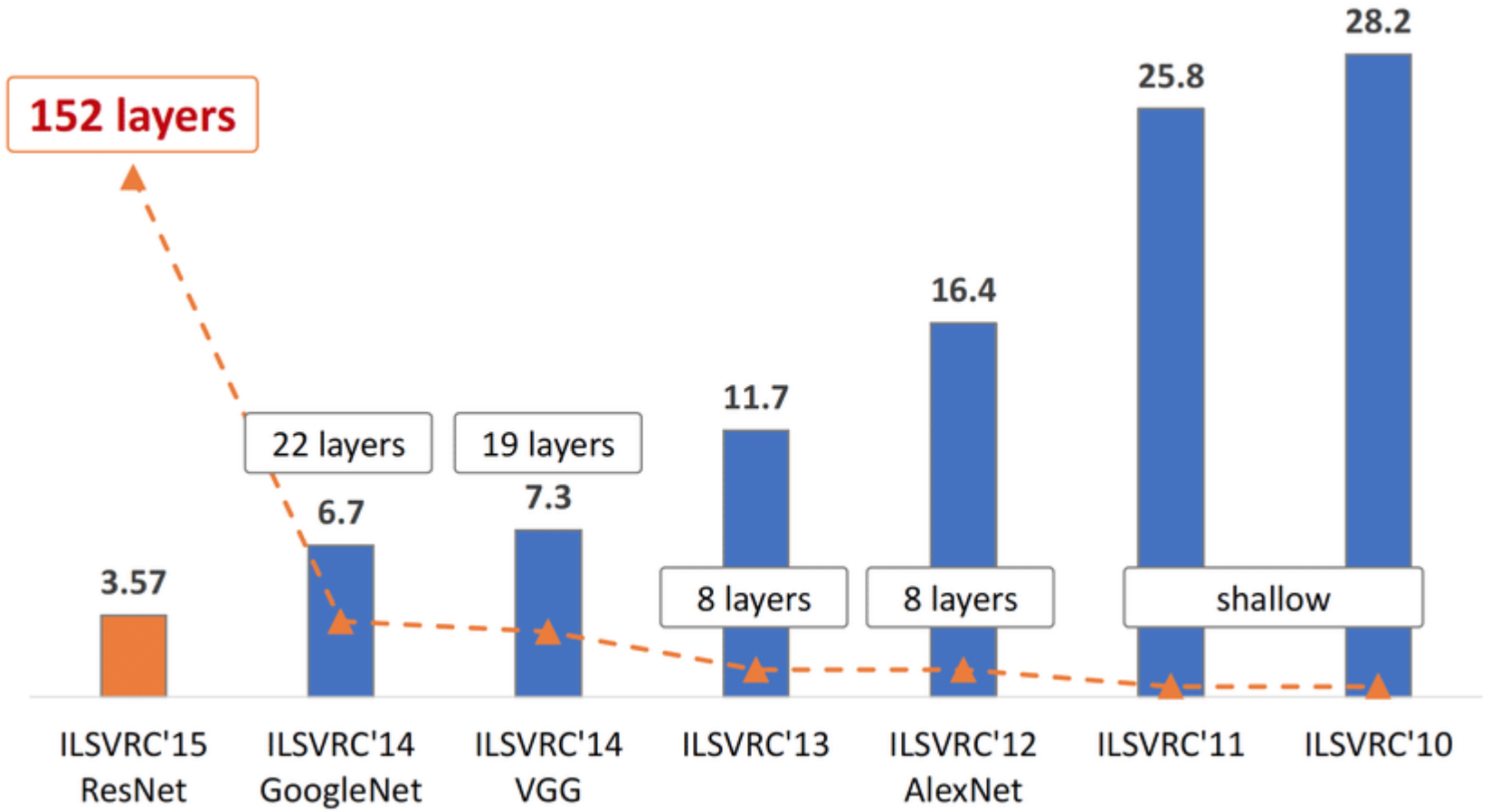
ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



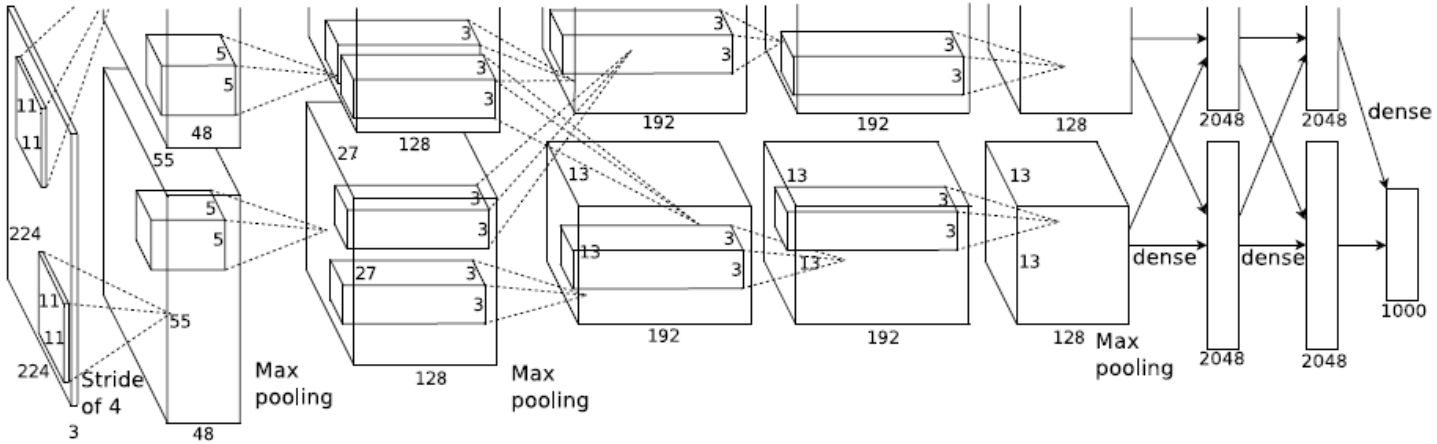
ImageNet challenge - pobjednici



Izvor

AlexNet (2012.)

- Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks, 2012.



Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

Zašto ne ranije, npr. 1998?

- Više podataka
- GPU i CUDA

VGG (2014.)

- Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014.

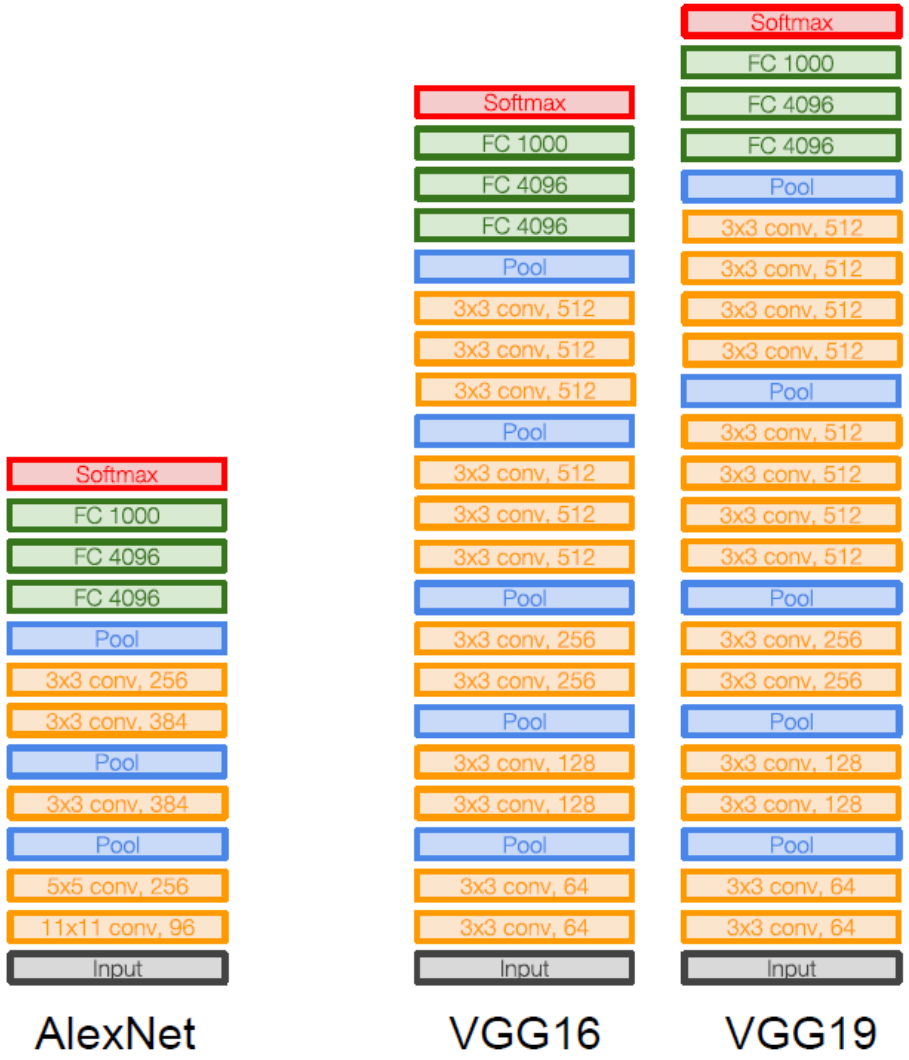
  - Manji filtri (3x3)
  - Više slojeva (dublja mreža)
  - ReLU
  - Sličan proces učenja kao kod AlexNet
  - 7.3% top 5 error in ILSVRC 2015
  - Najbolji model VGG19

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

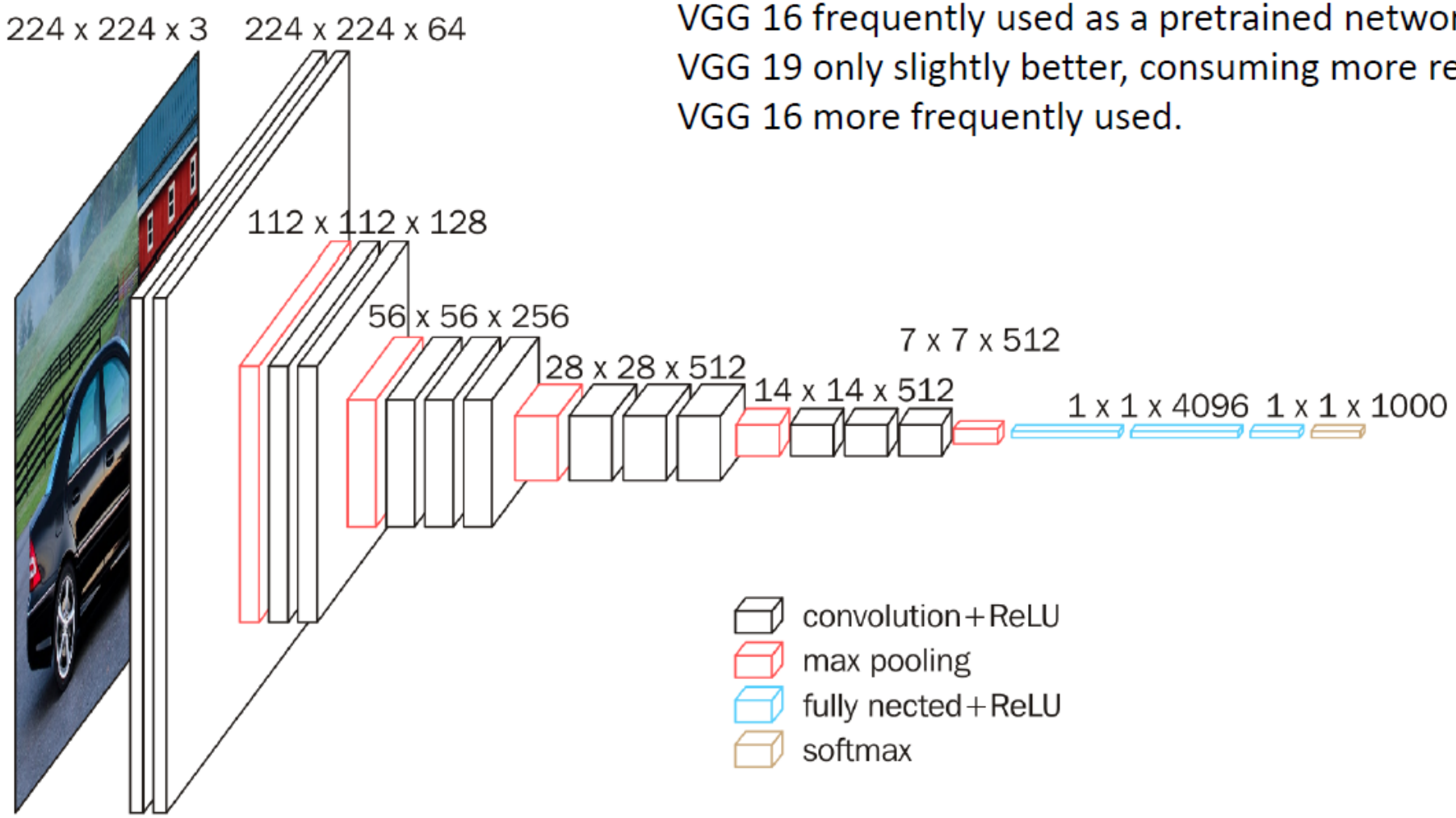
VGG vs AlexNet



Zašto 3x3 filtri?

„It is easy to see that a stack of two 3×3 conv. layers (without spatial pooling in between) has an effective receptive field of 5×5; three such layers have a 7 × 7 effective receptive field. So what have we gained by using, for instance, a stack of three 3×3 conv. layers instead of a single 7×7 layer? First, we incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative. Second, we decrease the number of parameters: assuming that both the input and the output of a three-layer 3 × 3 convolution stack has C channels…”

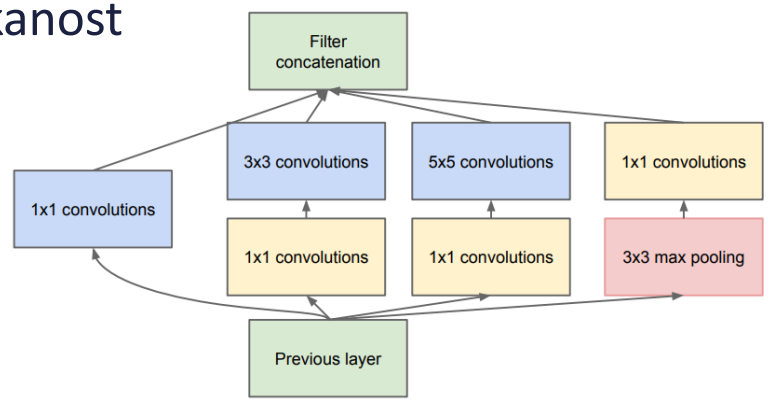
VGG16



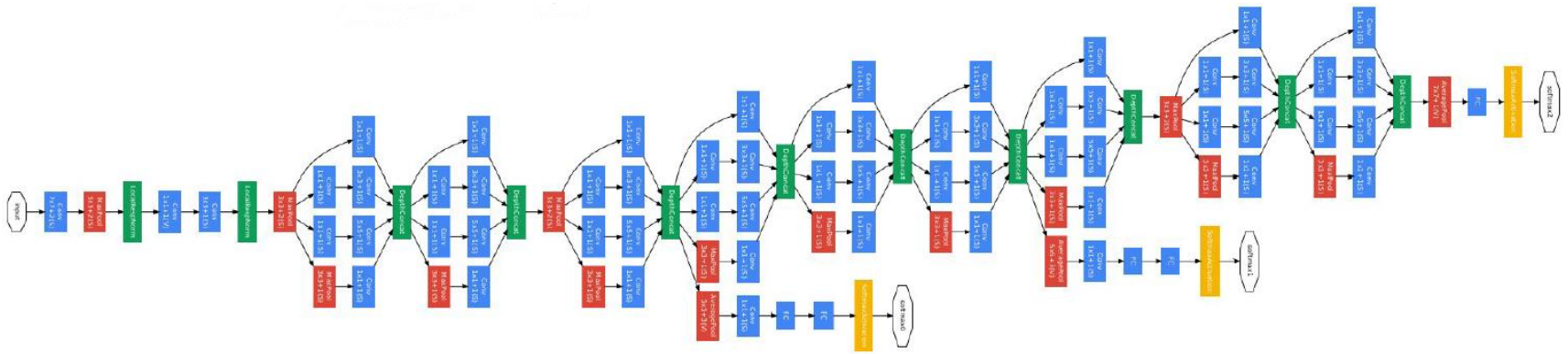
VGG 16 frequently used as a pretrained network.
VGG 19 only slightly better, consuming more resources.
VGG 16 more frequently used.

GoogLeNet (2014.)

- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going Deeper with Convolutions, 2014.
- Nastavlja se trend produbljivanja mreže s naglaskom na računalnu efikanost
- GoogLeNet:
  - Efikasni Inception moduli
  - Nema FC slojeva
  - Samo 5 milijuna parametara (12x manje nego AlexNet)
  - 6.7% top 5 error ILSVRC 2014



(b) Inception module with dimension reductions



Povećavanje dubine mreže

- Problem kada se uče izrazito duboke mreže na klasičan način
- Očekujemo da se poveća preciznost s povećanjem dubine mreže na skupu za učenje

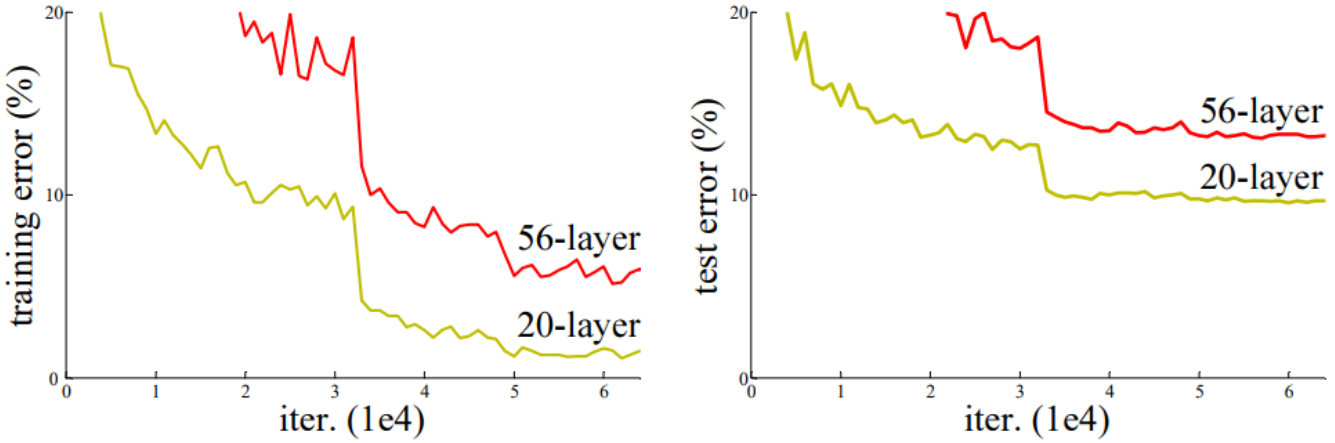
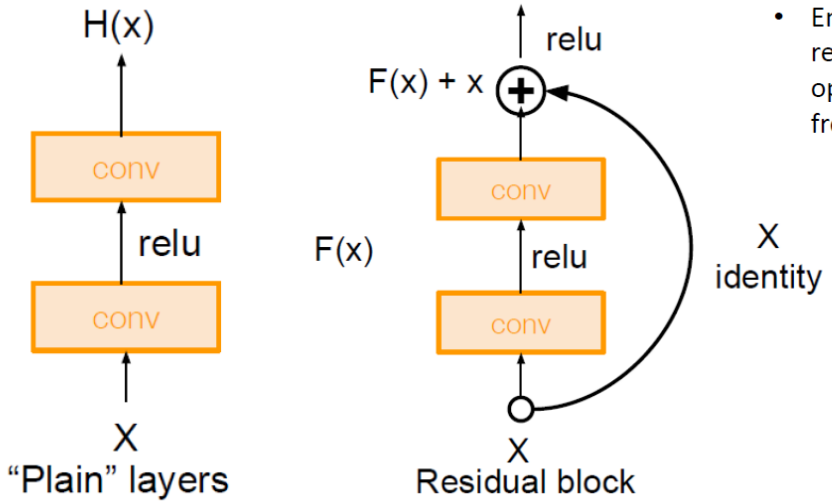


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

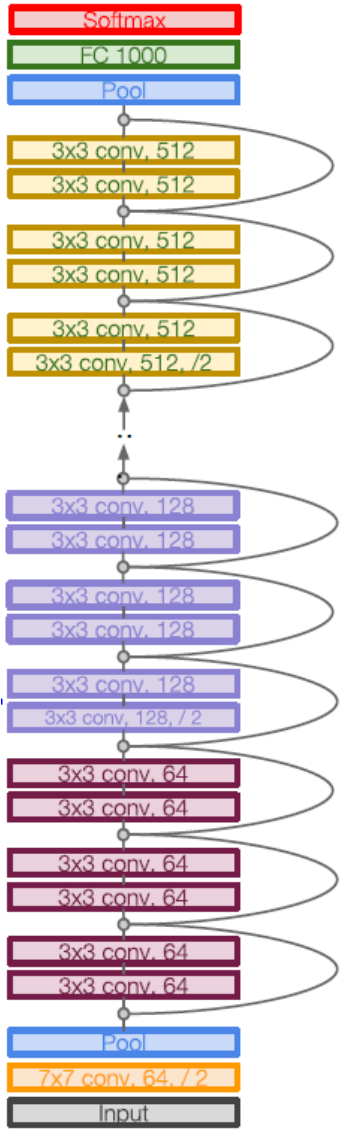
Ovo nije problem pretjeranog usklađivanja!

ResNet (2015.)

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2015.
- Revolucija u dubini mreže
- Microsoft
- Prvo mjesto u 5 različitih natjecanja (ImageNet i COCO)
- Koristi rezidualne blokove; svaki ima dva 3x3 konvolucijska sloja



- Empirical evidence showing that residual networks are easier to optimize, and can gain accuracy from increased depth.



ResNet (2015.)

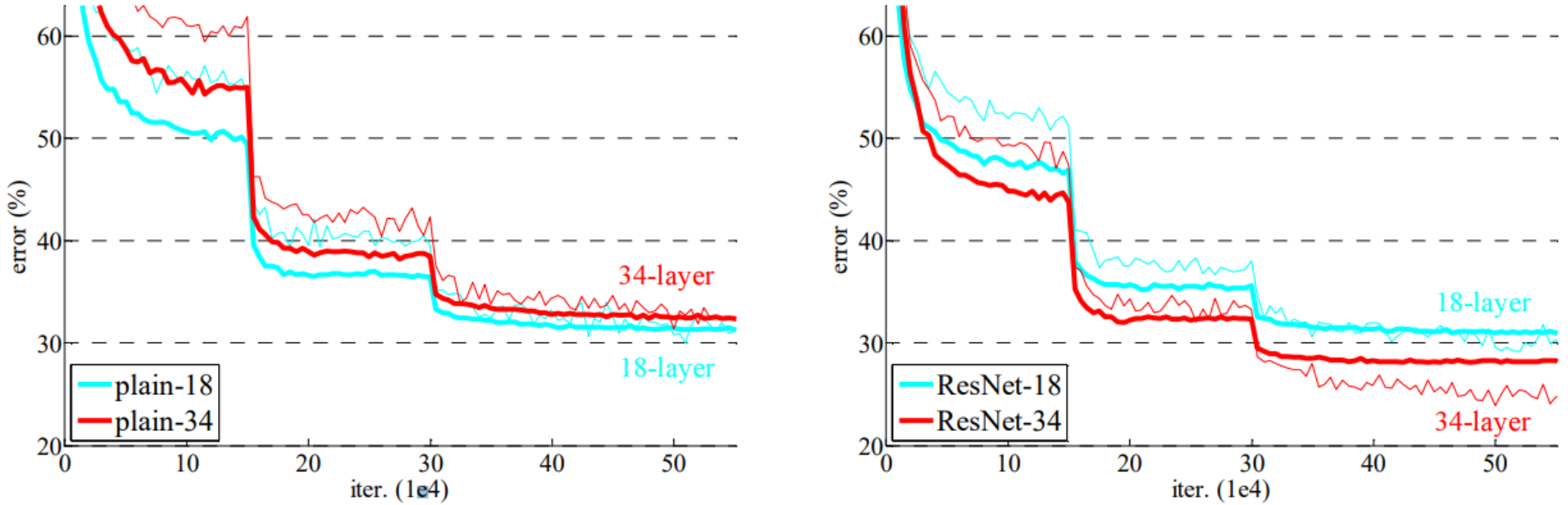


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

Usporedba računalnih zahtjeva i performansi

- A. Canziani, E. Culurciello, A. Paszke, AN ANALYSIS OF DEEP NEURAL NETWORK MODELS FOR PRACTICAL APPLICATIONS, 2017.

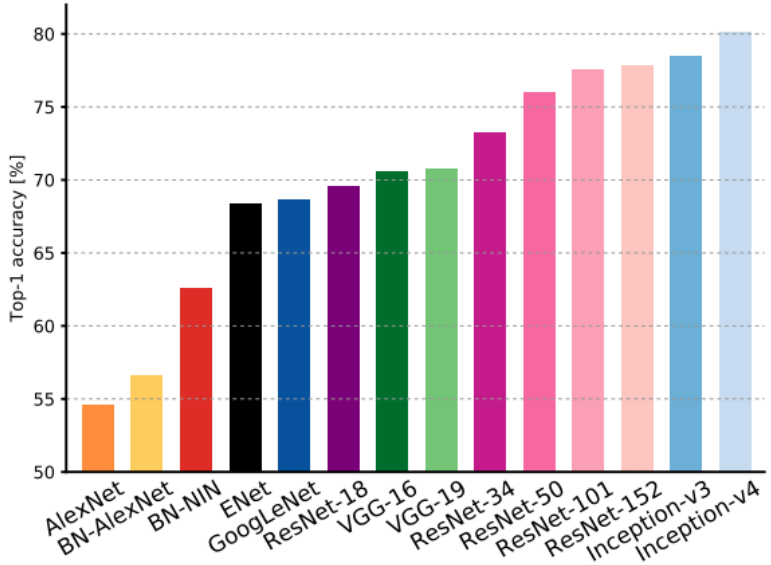


Figure 1: **Top1 *vs.* network.** Single-crop top-1 validation accuracies for top scoring single-model architectures. We introduce with this chart our choice of colour scheme, which will be used throughout this publication to distinguish effectively different architectures and their correspondent authors. Notice that networks of the same group share the same hue, for example ResNet are all variations of pink.

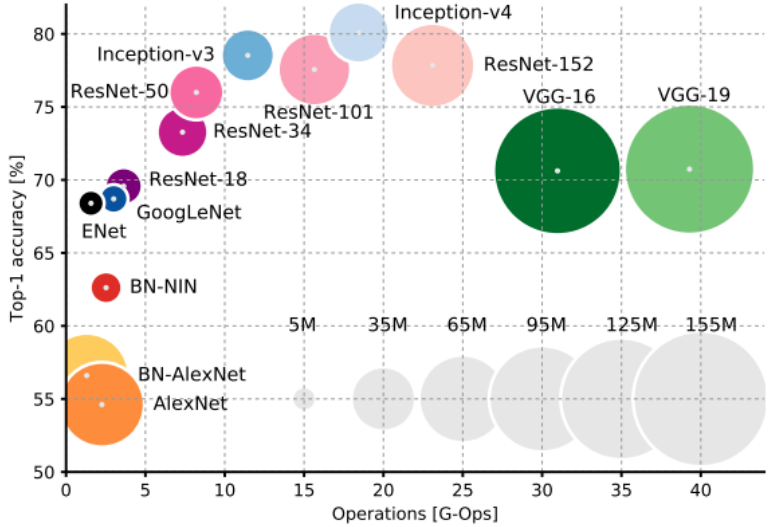


Figure 2: **Top1 *vs.* operations, size $\propto$ parameters.** Top-1 one-crop accuracy versus amount of operations required for a single forward pass. The size of the blobs is proportional to the number of network parameters; a legend is reported in the bottom right corner, spanning from 5×10^6 to 155×10^6 params. Both these figures share the same y-axis, and the grey dots highlight the centre of the blobs.