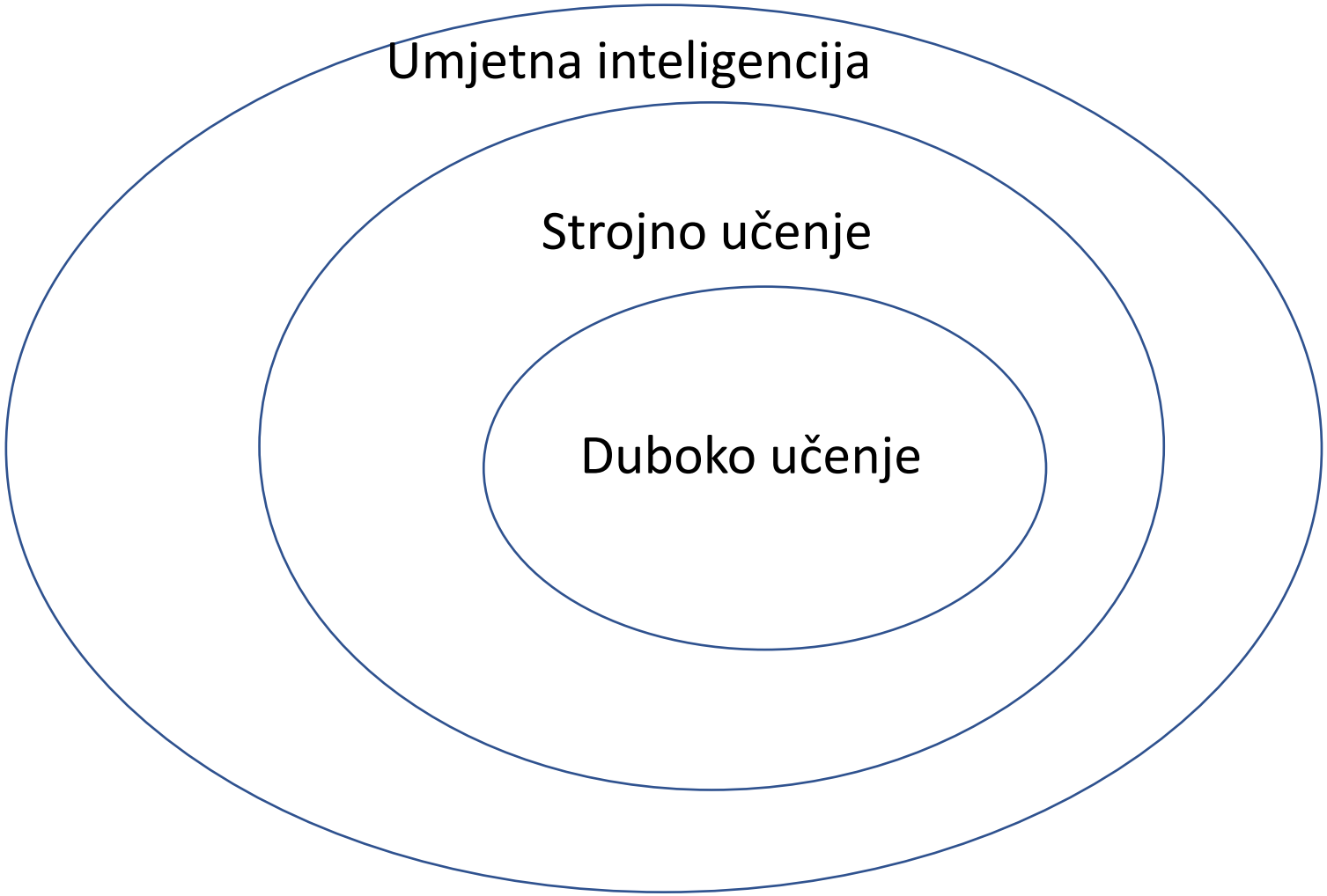
Uvod u znanost o podacima

Uvod u duboko učenje

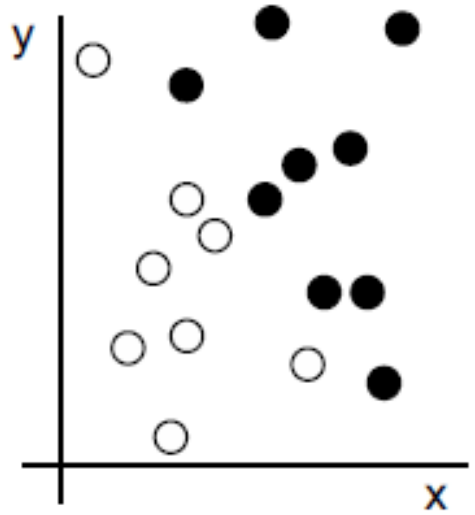
Prof. dr. sc. Mile Šikić

10. predavanje, 4. siječnja 2022.

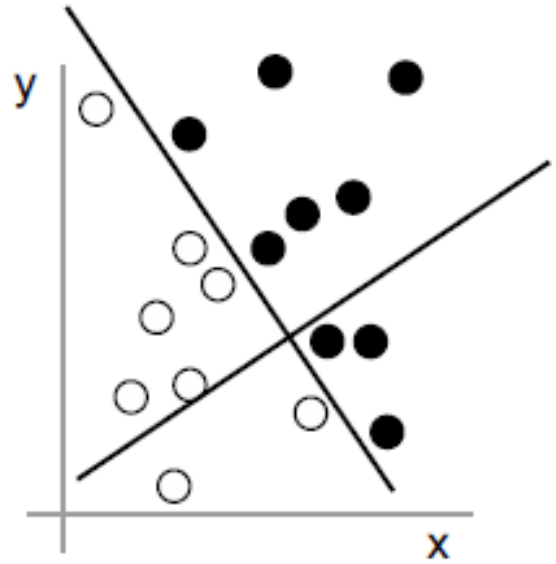
ak. god. 2021./2022.


Umjetna inteligencija
Strojno učenje
Duboko učenje

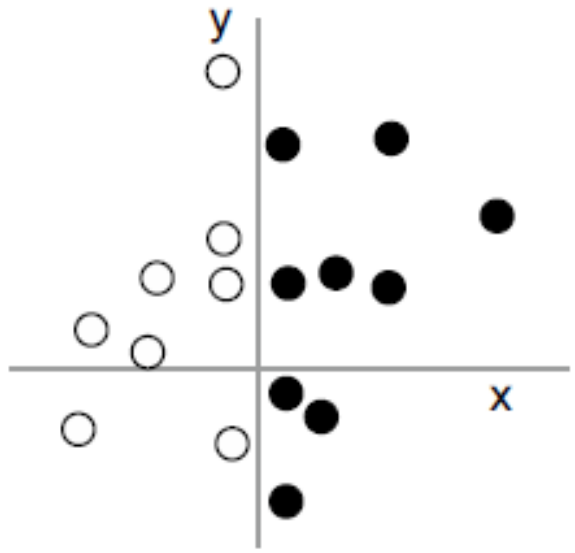
1: Sirovi podaci



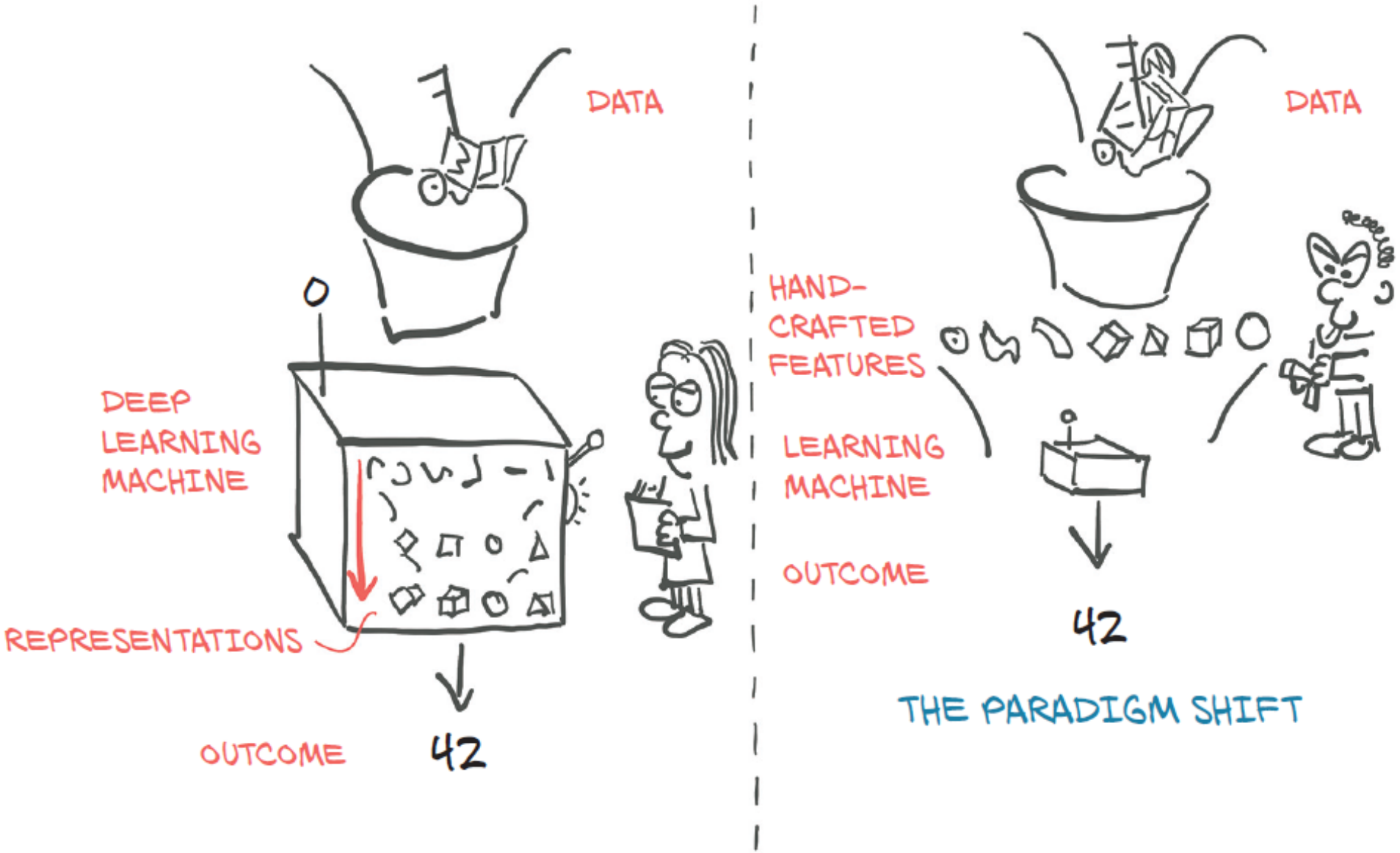
2: Promjena koordinata



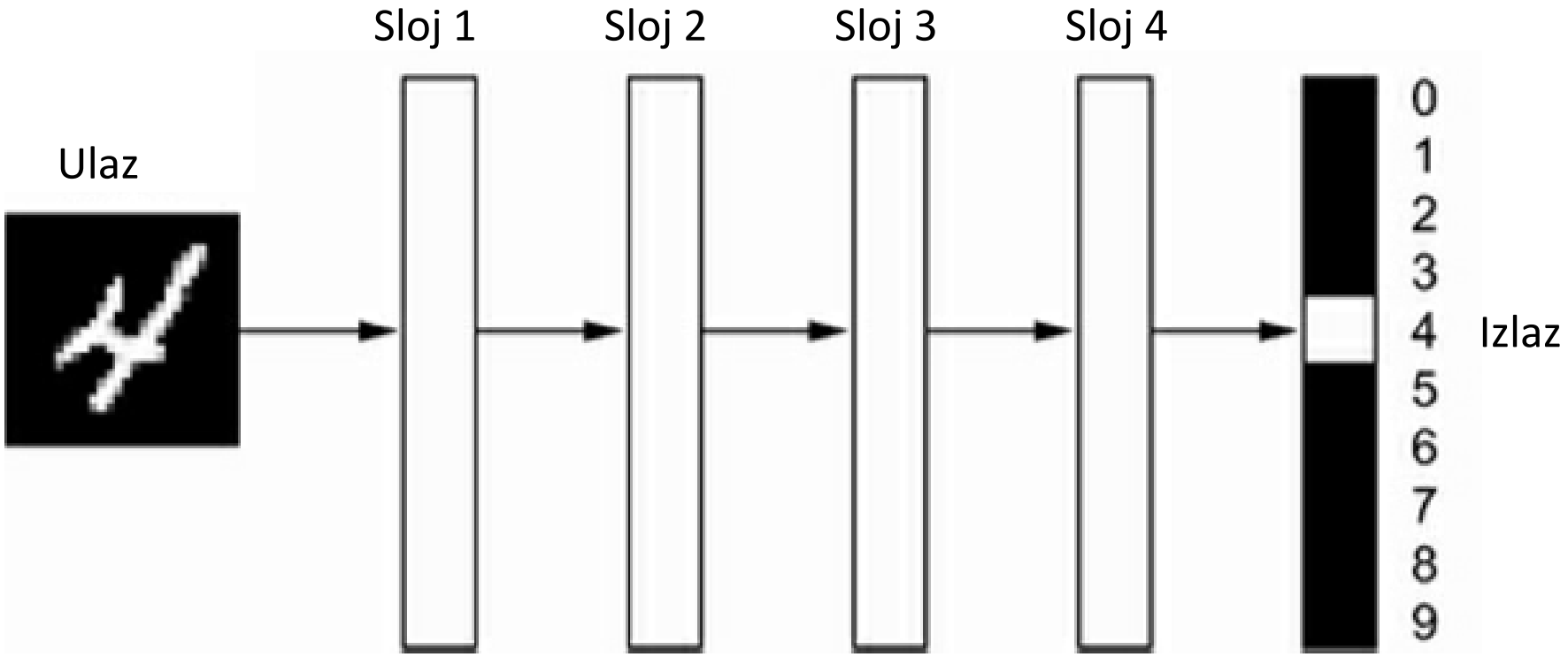
3: Bolja reprezentacija



Središnji problem u strojnom/dubokom učenju je smislena transformacija podataka – učenje smislene reprezentacije

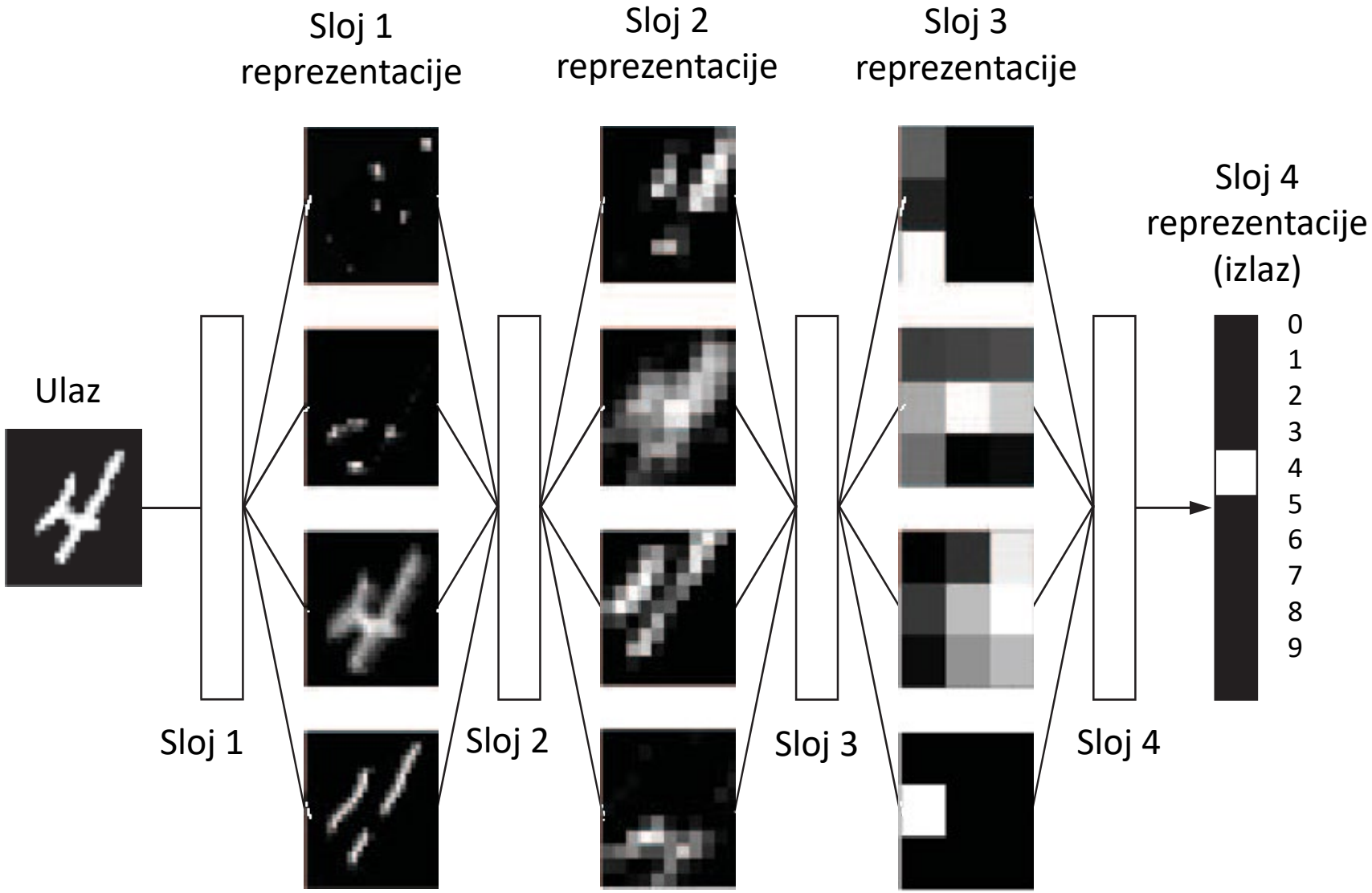


E. Stevens, L. Antiga, and T. Viehmann. Deep Learning with Pytorch



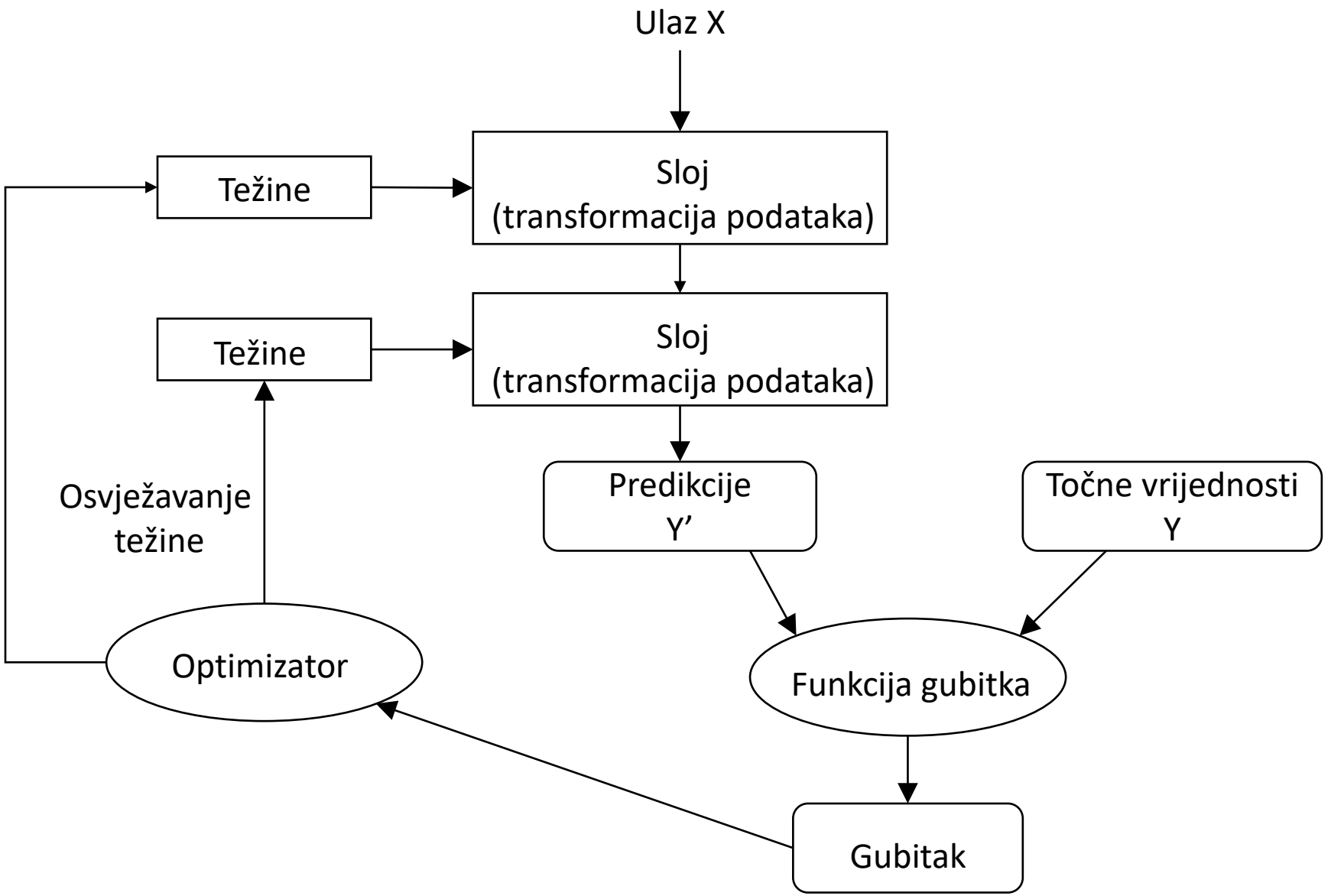
Ideja: uzastopni slojevi reprezentacije

Chollet, François. Deep Learning with Python, Second Edition



Informacija prolazi kroz uzastopne filtre i pročišćava se

Chollet, François. Deep Learning with Python, Second Edition



Chollet, François. Deep Learning with Python, Second Edition

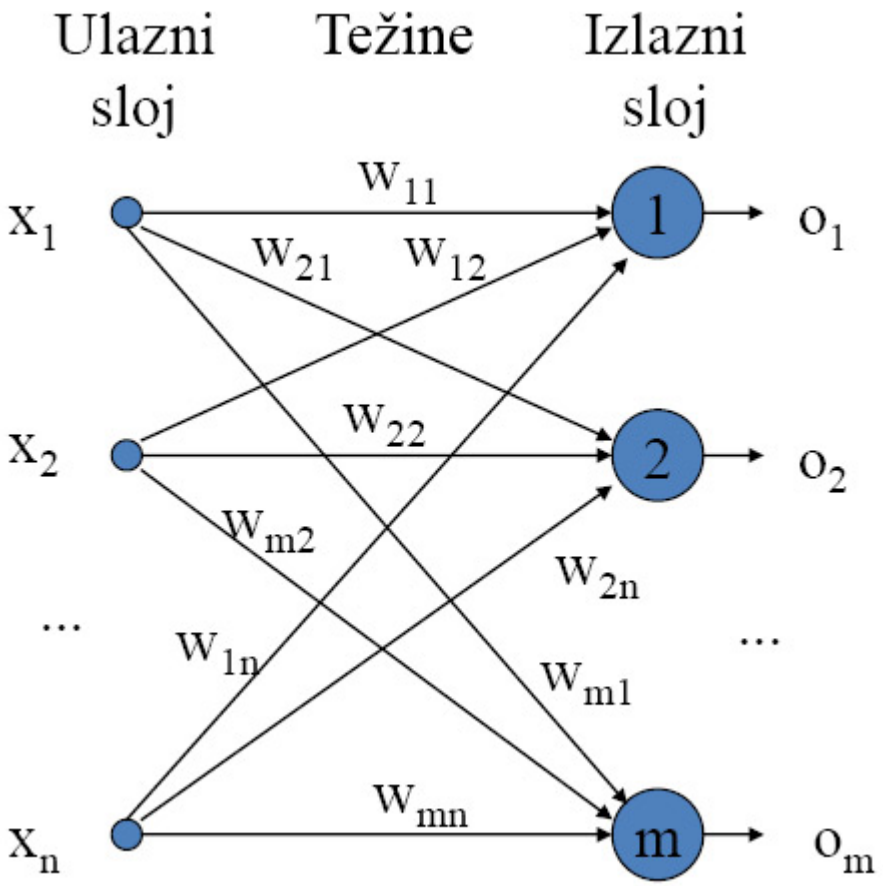
Funkcije gubitka

- Klasifikacija:
  - Binarna klasifikacija – binarna unakrsna entropija

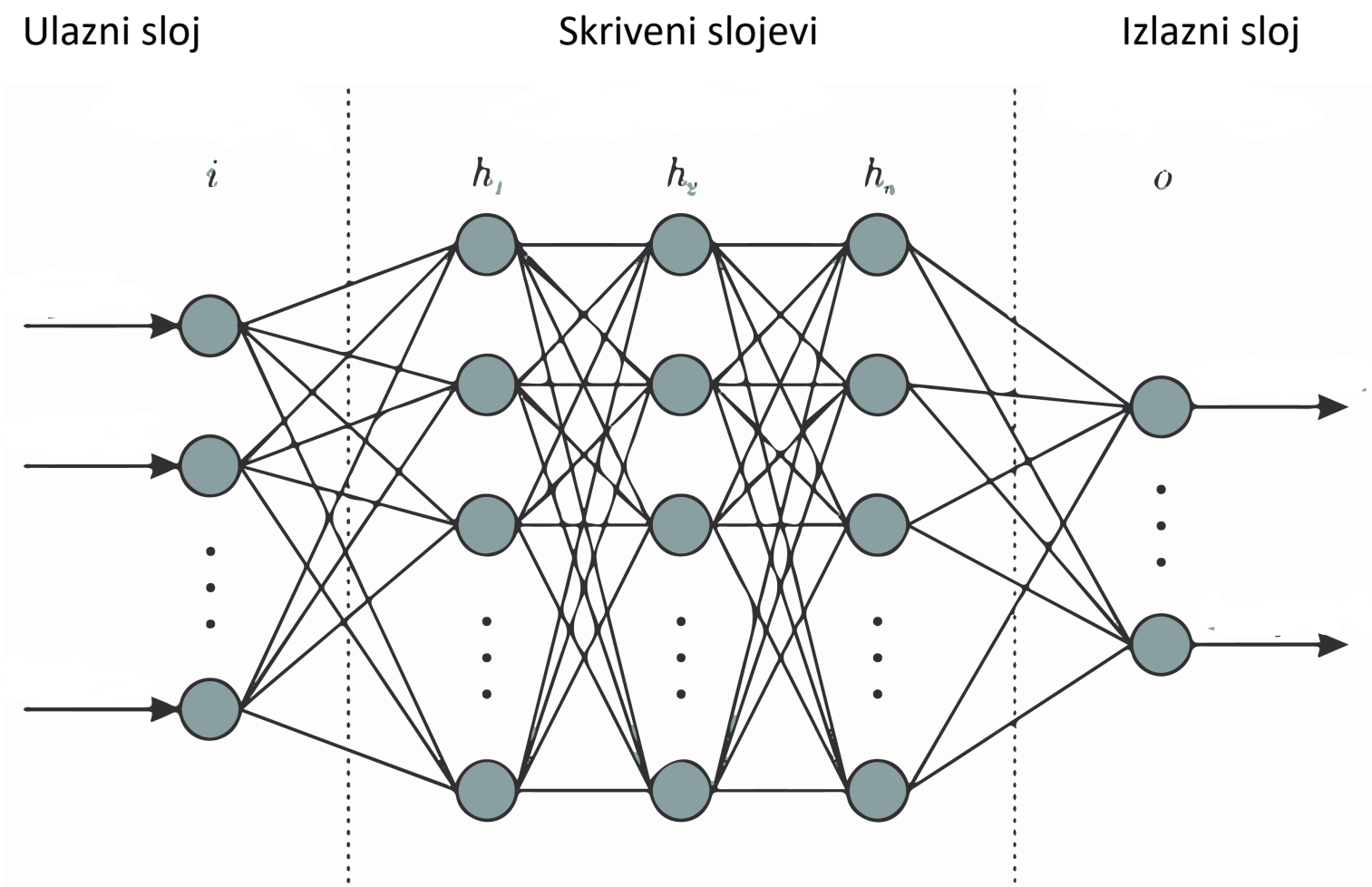
$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot logy_i' + (1-y_i) \cdot \log(1-y_i')\ ]$$

- Regresija
  - Srednja kvadratna pogreška (MSE)
  - Srednja apsolutna pogreška (MAE)

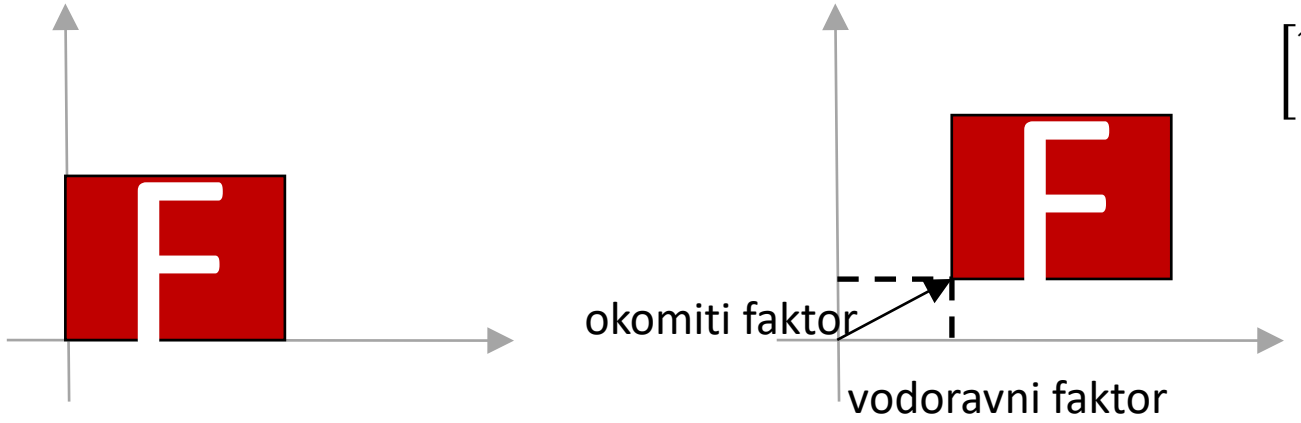
Jednoslojna neuronska mreža



Neuronska mreža

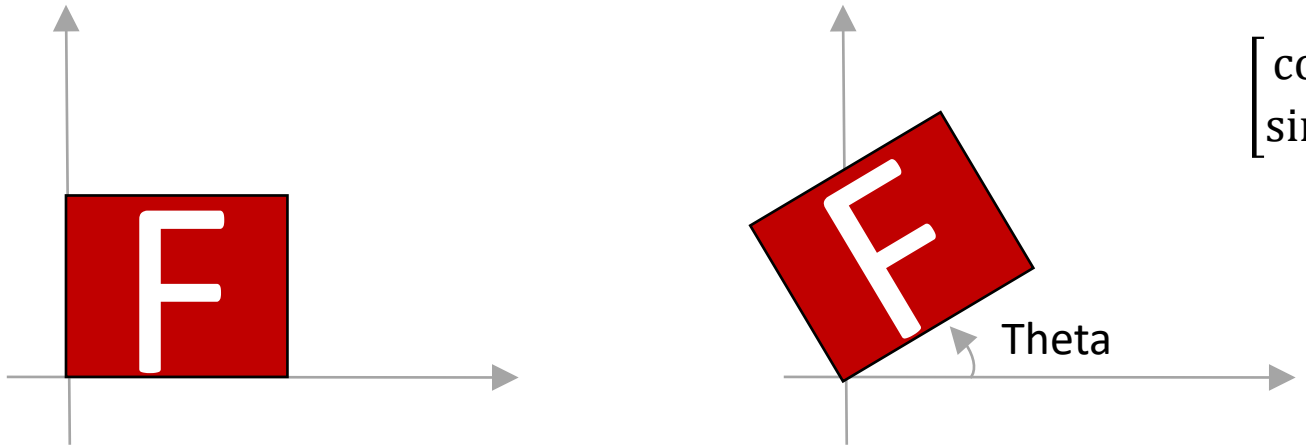


Linearne transformacije



$$\begin{bmatrix} vodoravni\ faktor \\ okomiti\ faktor \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

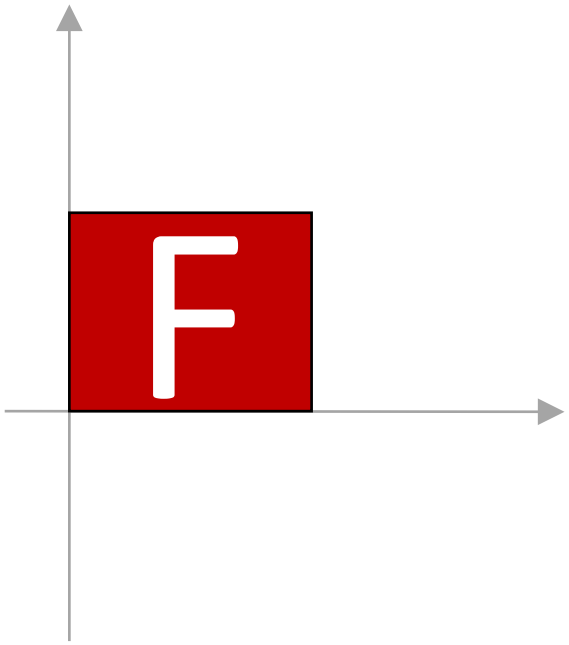
2D translacija (zbrajanje vektora)

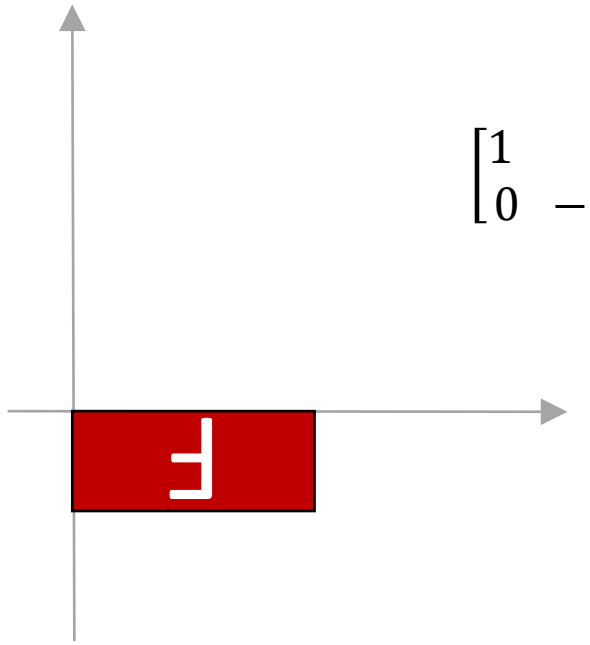


$$\begin{bmatrix} \cos(theta) & -\sin(theta) \\ \sin(theta) & \cos(theta) \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$$

2D rotacija (skalarni produkt)

2D skaliranje (skalarni produkt)

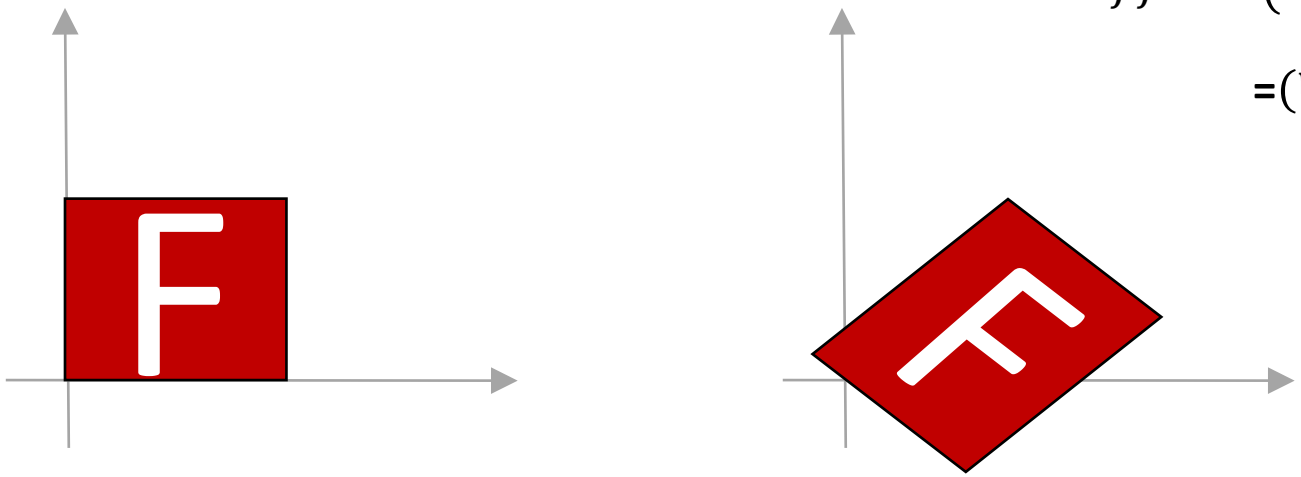




$$\begin{bmatrix} 1 & 0 \\ 0 & -0.5 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$$

Chollet, François. Deep Learning with Python, Second Edition

Affina transformacija u ravnini



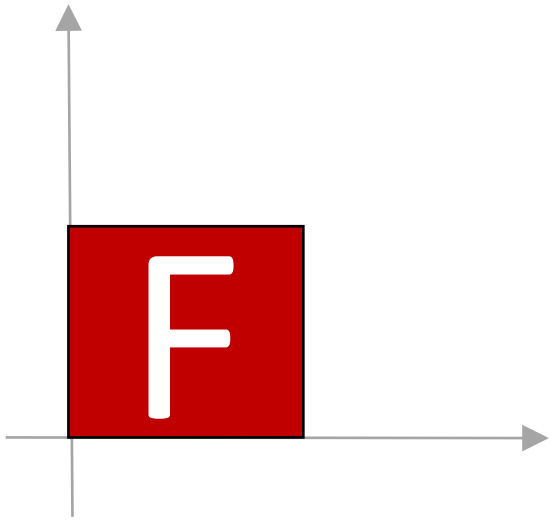
$$\boldsymbol{W} \times \boldsymbol{x} + \boldsymbol{b}$$

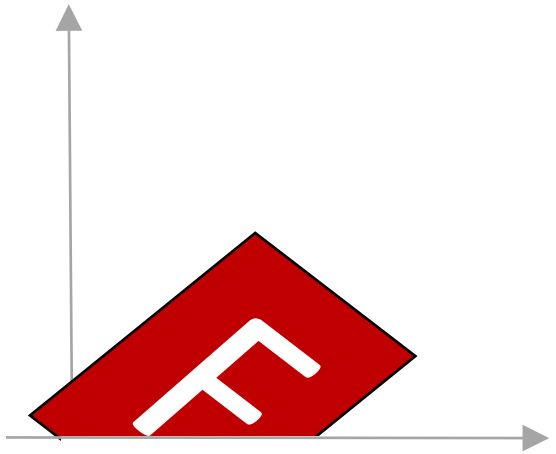
$$affina2\big(affina1(x)\big) = \boldsymbol{W2} \times (\boldsymbol{W1} \times \boldsymbol{x} + \boldsymbol{b1}) + \boldsymbol{b2} =$$

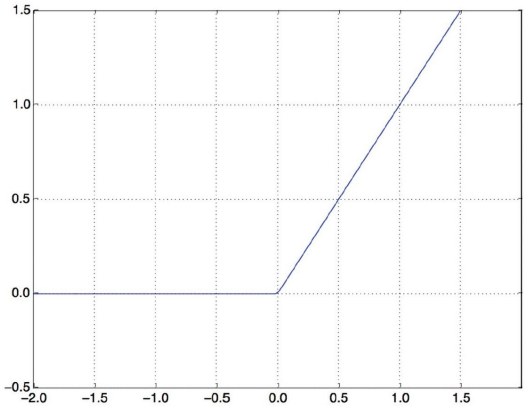
$$=(\boldsymbol{W2} \times \boldsymbol{W1}) \times \boldsymbol{x} + (\boldsymbol{W2} \times \boldsymbol{b1} + \boldsymbol{b2})$$

Chollet, François. Deep Learning with Python, Second Edition

Affina transformacija praćena relu aktivacijom

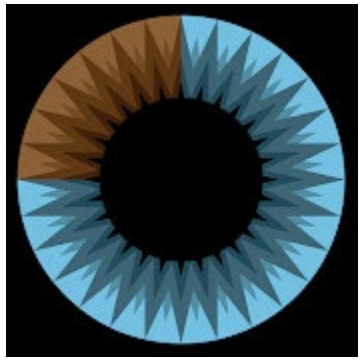






Chollet, François. Deep Learning with Python, Second Edition

Uvod u linearnu algebru



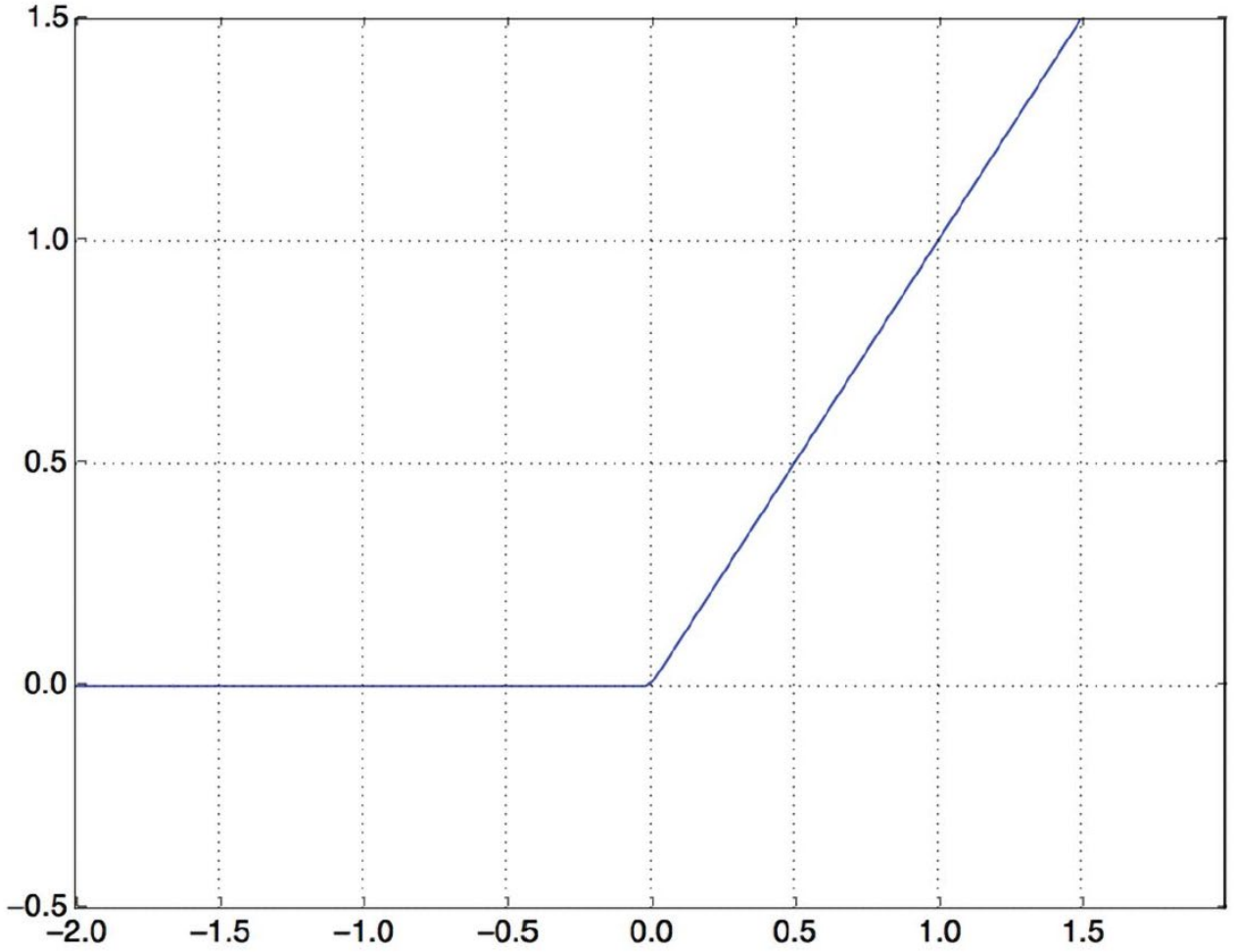
3Blue1Brown

https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

Zašto trebamo aktivacijsku funkciju

- Bez nje imamo samo linearnu transformaciju ulaznih podataka
- Skup svih mogućih linearnih transformacija ulazni podatka u višedimenzionalnom prostoru – previše restriktivno
- Za bogatiji prostor hipoteza trebamo nelinearnosti, odnosno aktivacijske funkcije

ReLu (rectified linear unit function)



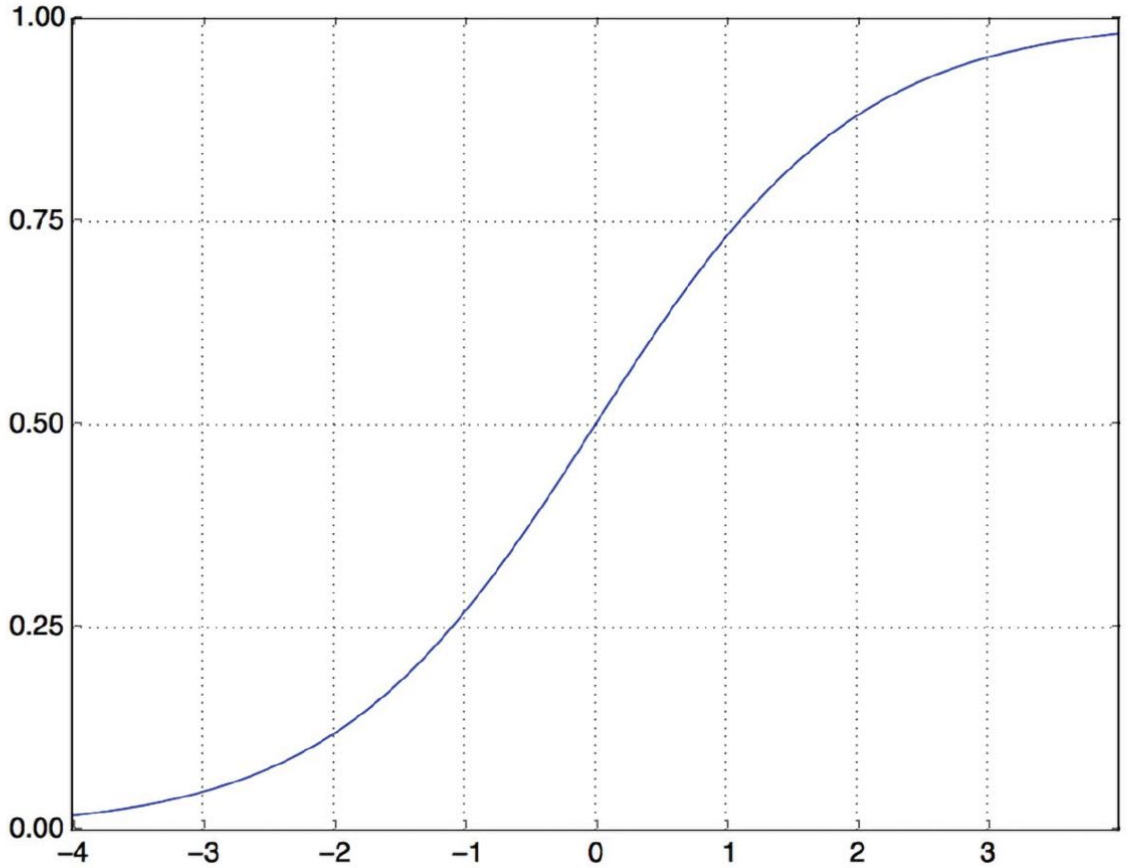
Prednosti:

- Smanjena izvjesnost nestajućeg gradijenta
- ReLu i njena derivacija su brže za izračunati od sigmoide.
- Jednostavna i dovoljno dobra u većini slučajeva

Nedostaci:

- u slučaju da previše aktivacija je ispod nule onda će velik broj neurona vraćati 0 i tako sprječavati učenje.
- Rješenje: Leaky-ReLu, ELU, ….

Sigmoida

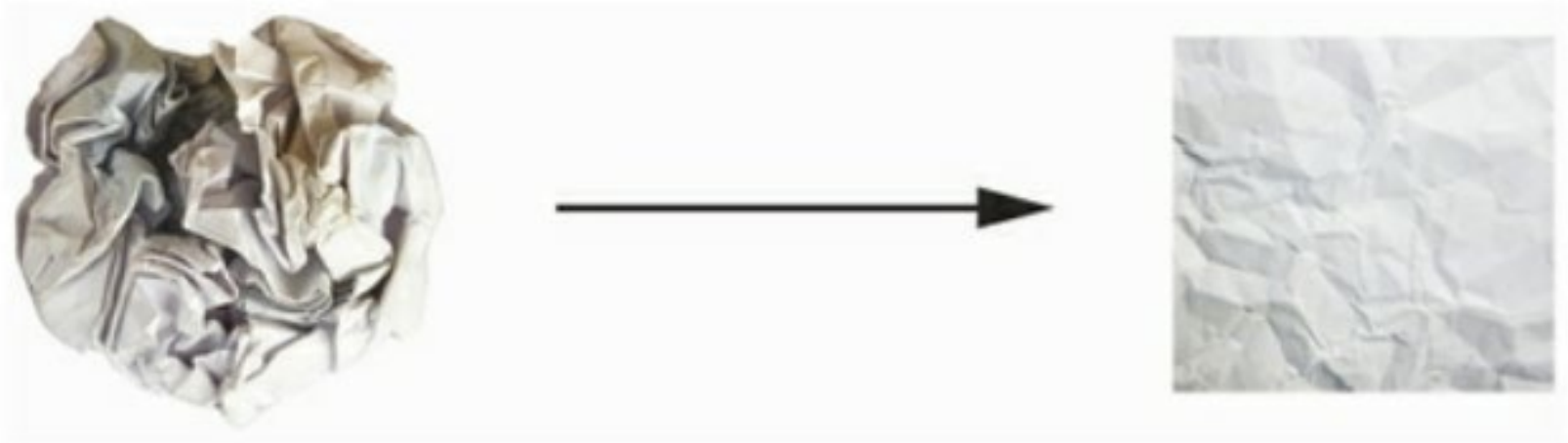


Koristimo u zadnjem sloju da dobijemo skor u rasponu [0, 1] koji možemo interpretirati kao vjerojatnost

Geometrijska interpretacija dubokog učenja

- Sve je vektor, odnosno sve je točka u geometrijskom prostoru
- Ulaze (slike, tekstove, zvučni signal) potrebno je prvo vektorizirati
- Svaki sloj radi jednu jednostavnu geometrijsku transformaciju na podacima koji prolaze kroz njega
- Transformacija mora biti diferencijabilna da model može učiti – postepena geometrijska promjena od ulaza do izlaza mora biti glatka i kontinuirana ☺

Geometrijska interpretacija dubokog učenja



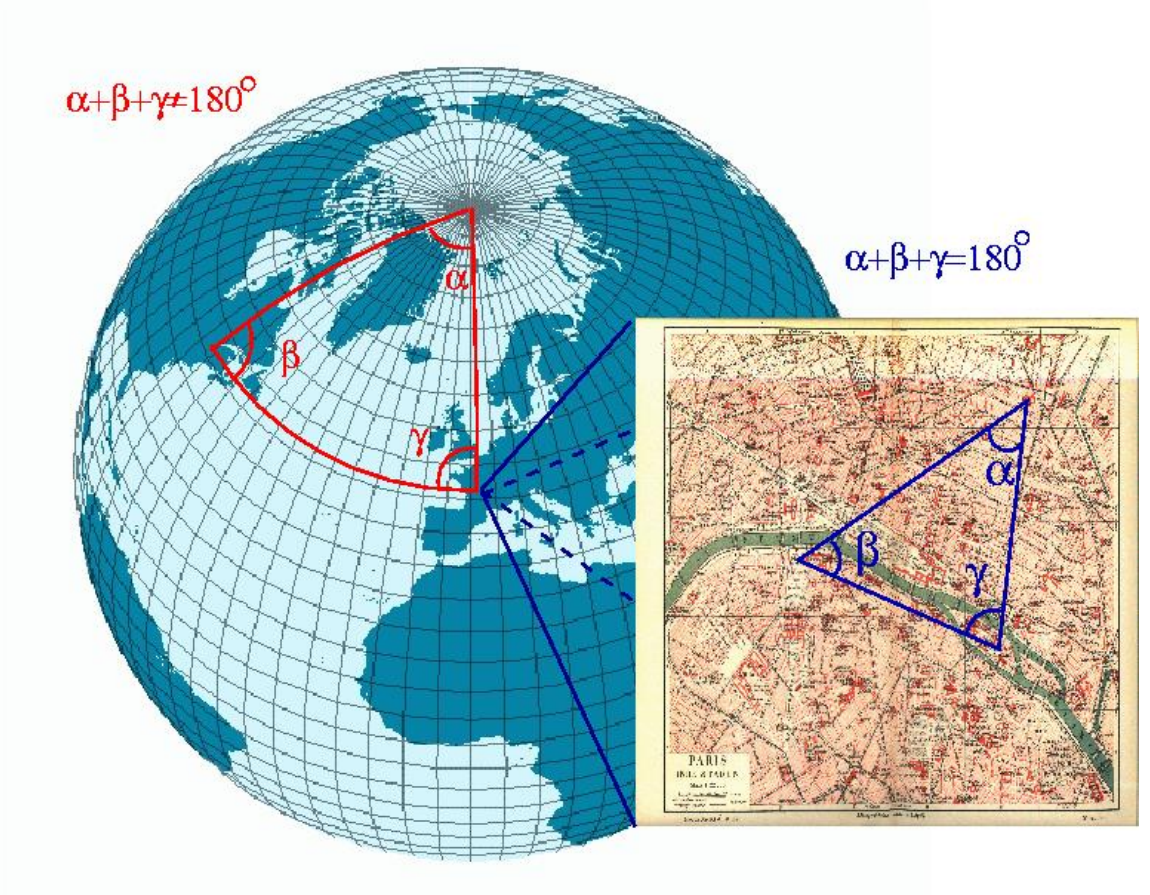
Dva lista papira (crveni i plavi), zgužvani zajedno.
Zgužvani papiri – klasa podataka

Duboko učenje
– transformacija zgužvane lopte do razine da može razdvojiti klase
- svaki sloj pomalo transformira

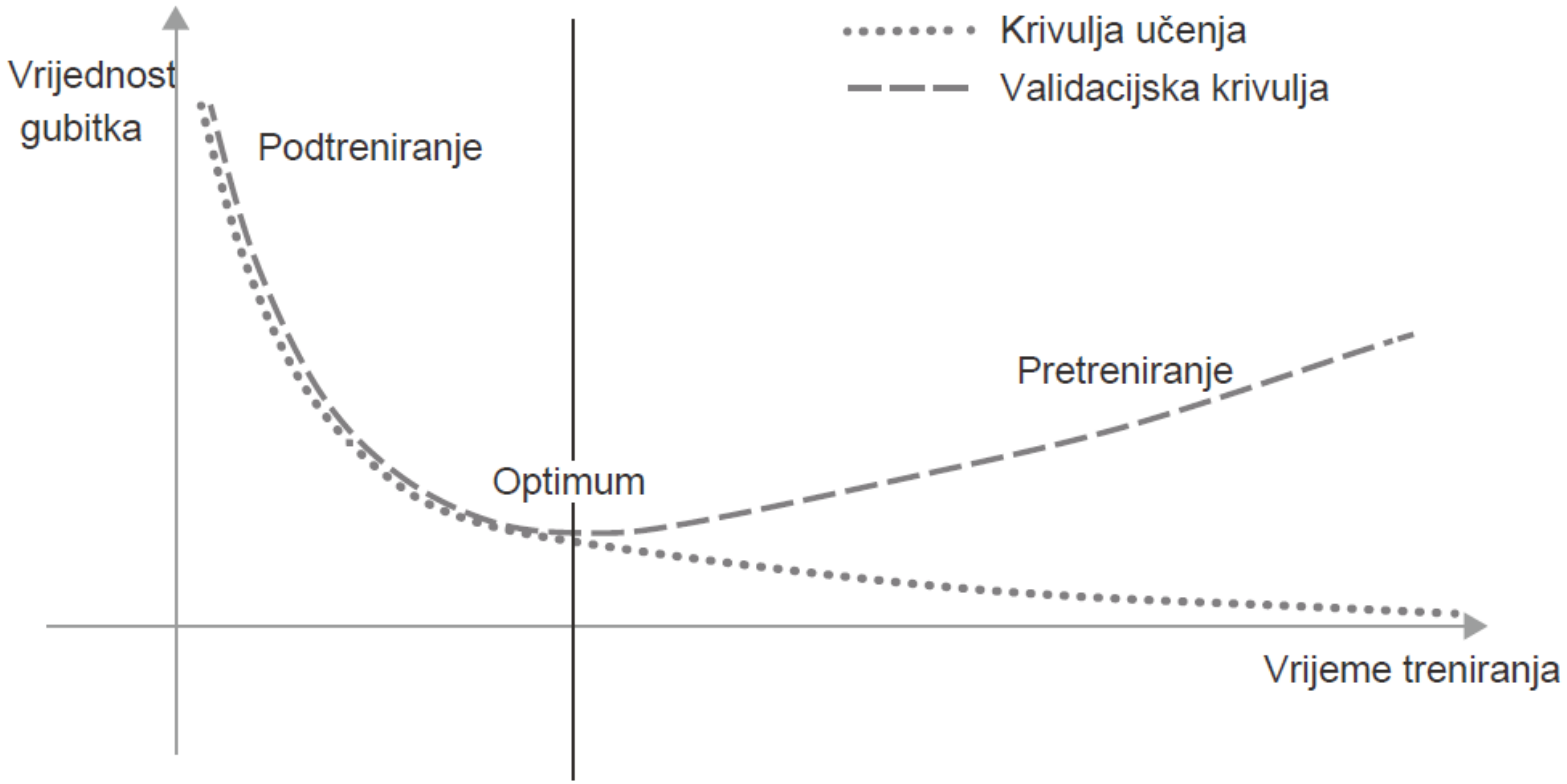
Chollet, François. Deep Learning with Python, Second Edition

Mnogostrukost (engl. Manifold)

Apstraktan topološki prostor u kojem svaka točka ima okolinu koja podsjeća na euklidski prostor, ali čija globalna struktura može biti kompliciranija. Kada se proučavaju mnogostrukosti, pojam dimenzije je važan.



Poopćenje: cilj strojnog učenja

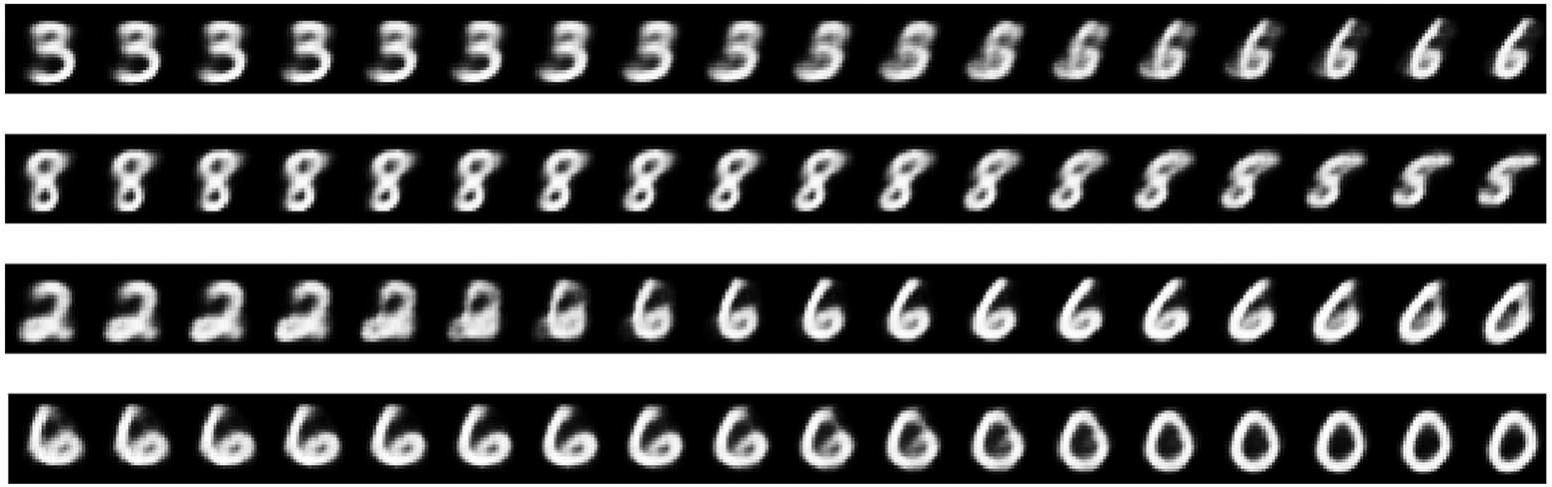


Optimizacija:
Proces prilagođenja modela u cilju postizanja najboljih performansi za podatke za učenje

Poopćenje:
Kako se trenirani model ponaša na podacima koje prije nije vidio. Ne možemo ju kontrolirati. Možemo jedino prilagoditi model podacima

Chollet, François. Deep Learning with Python, Second Edition

Hipoteza mnogostrukosti



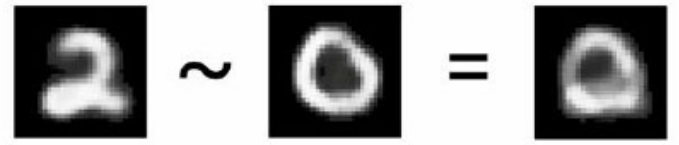
- rukom pisane znamenke zauzimaju sićušan, strukturiran podprostor (ukupan prostor 256^{784}) 28x28 pixela
- podprostor je kontinuiran (mala modifikacija -> još uvijek prepoznatljivo)
- podprostor je povezan (međuslike)

Teza povlači:

- Modeli trebaju prilagoditi relativno jednostavan, nisko-dimenzionalan visoko strukturiran prostor (latentna mnogostrukost)
- Unutar pojedine mnogostrukosti, moguće je INTERPOLIRATI između dva ulaza, odnosno pretvoriti iz jednog u drugi preko kontinuiranog kuta čije sve točke pripadaju mnogostrukosti

Razlika između linearne interpolacija i one u latentnoj mnogostrukosti

Interpolacija mnogostrukosti
(međutočka u latentnoj mnogostrukosti)



Linearna interpolacija
(srednja vrijednost u enkodiranom prostoru)

Chollet, François. Deep Learning with Python, Second Edition

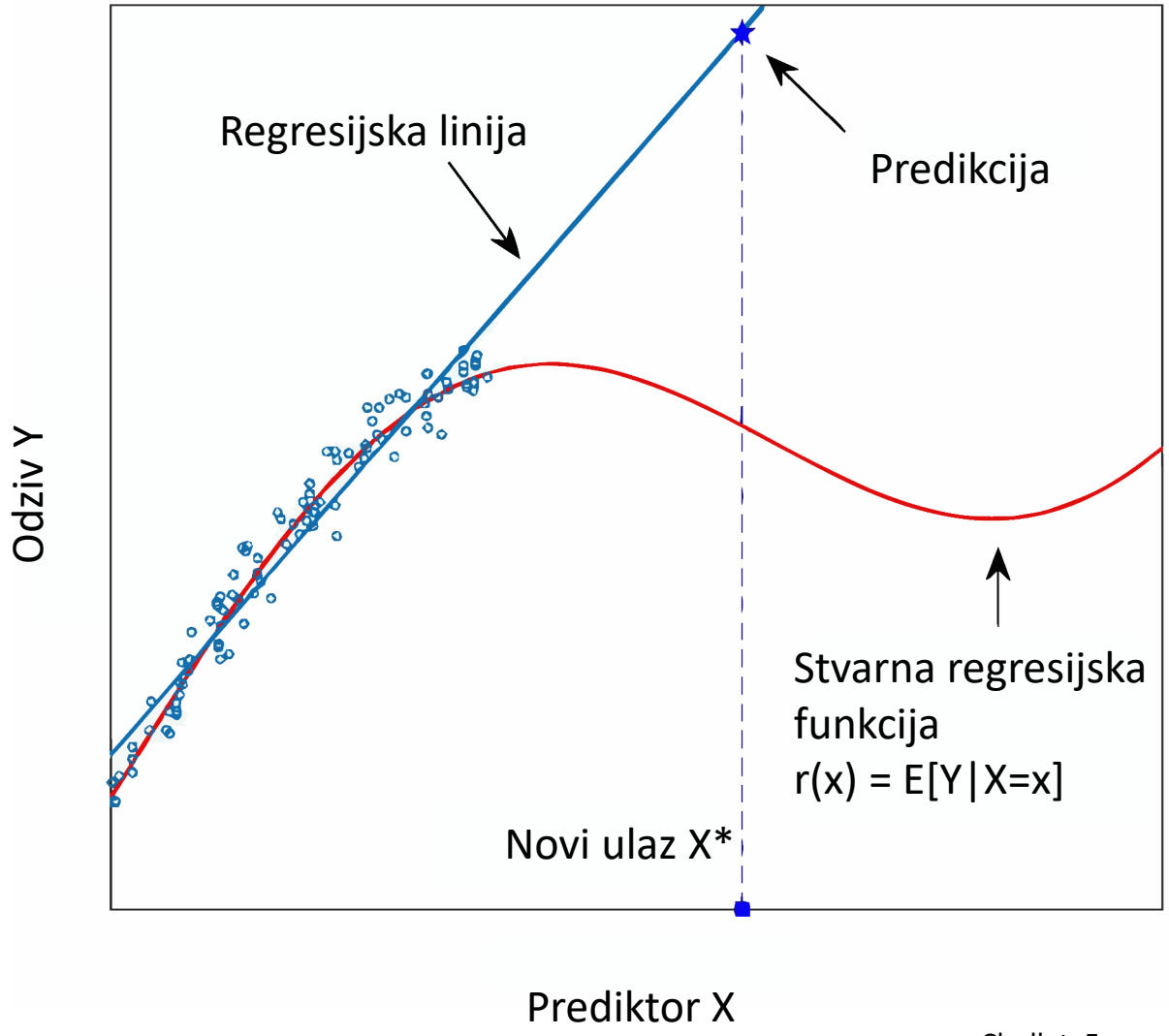
Interpolacija omogućuju samo lokalno poopćenje

Ljudi imaju sposobnost ekstremnog poopćenja koji su omogućeni kognitivnim mehanizmima drugačijim od interpolacije:

- apstrakcija,
- simbolički modeli svijeta,
- logika,
- rasuđivanje,
- apriorna znanja o svijetu.

Razum nasuprot intuiciji i prepoznavanju uzoraka!

Interpolacija vs ekstrapolacija

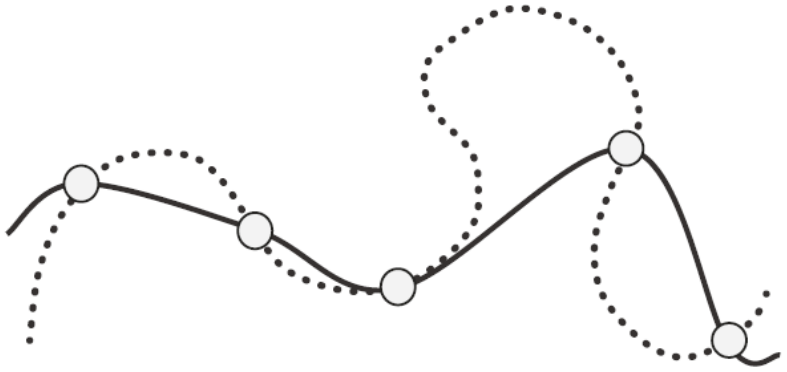


Chollet, François. Deep Learning with Python, Second Edition

Uzorkovanje

Originalni latentni prostor

Rijetko uzorkovanje:
Model ne odgovara latentnom prostoru i vodi krivoj interpolaciji



Gusto uzorkovanje: Model dobro aproksimira latentni prostor i interpolacija vodi k poopćenju

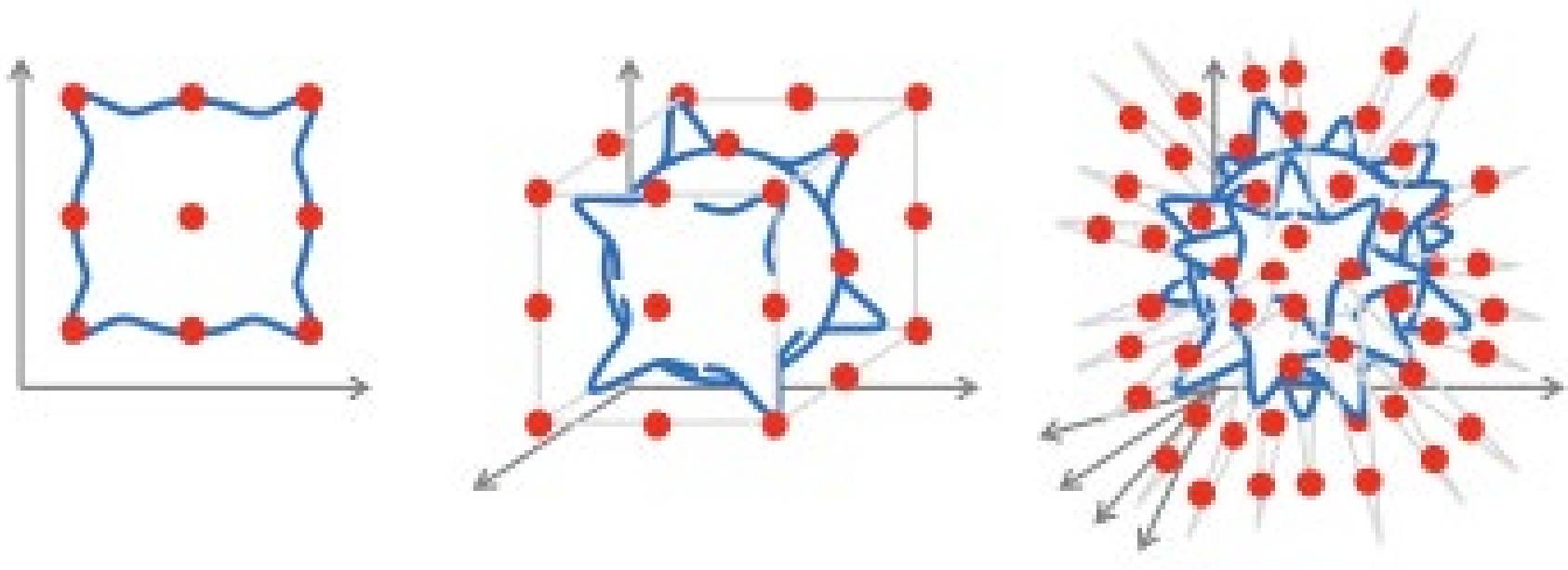
Chollet, François. Deep Learning with Python, Second Edition

Interpolacija u visokodimenzionalnom prostoru

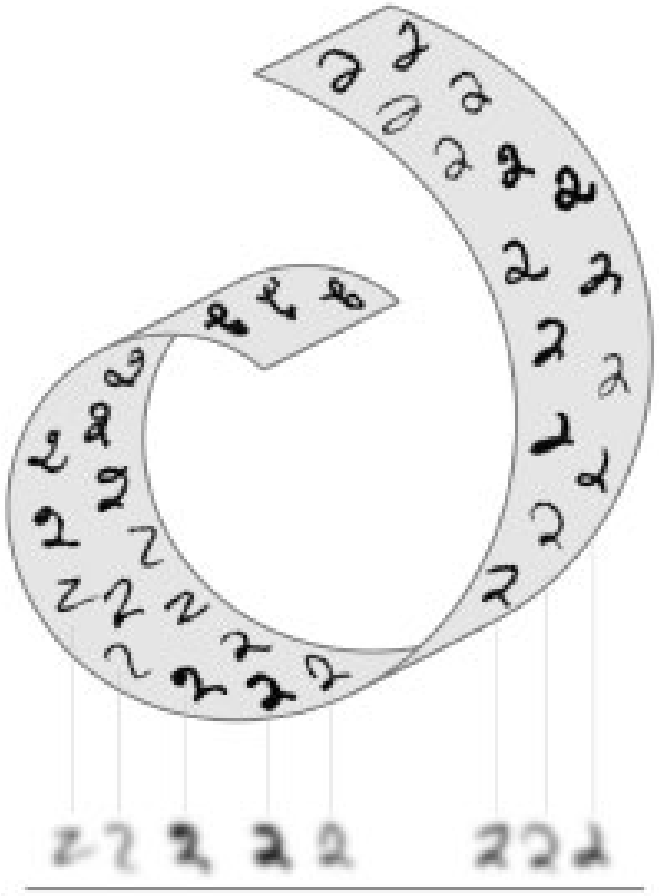
- Interpolacija u niskodimenzijalnom prostoru je klasičan problem obrade signala i možemo precizno matematički kontrolirati pogrešku
- Visokodimenzionalan prostor je problem – prokletstvo dimenzionalnosti

Prokletstvo dimenzionalnosti

- Porast broja dimenzija -> raste volumen, dostupni podaci postaju rijetki
- Potrebna količina podataka raste eksponencijalno s brojem dimenzija



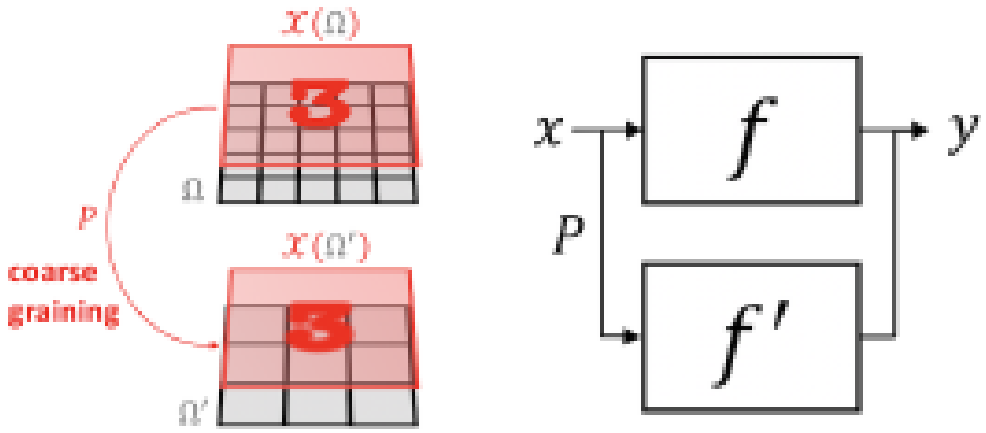
Redukcija dimenzionalnosti



- Potrebno je imati snažnu pretpostavku o ciljnoj funkciji (npr. ovisnost o skupu niskodimenzionalnih projekcija ulaza)
- U većini realnih primjera to nije moguće (udaljene interakcije među značajkama)
- Nadamo se da su podaci prostorno strukturirani -> korištenje prostorne strukture fizičke domene i inicijalnih geometrijskih znanja (prior)

Chollet, François. Deep Learning with Python, Second Edition

Inicijalna geometrijska znanja



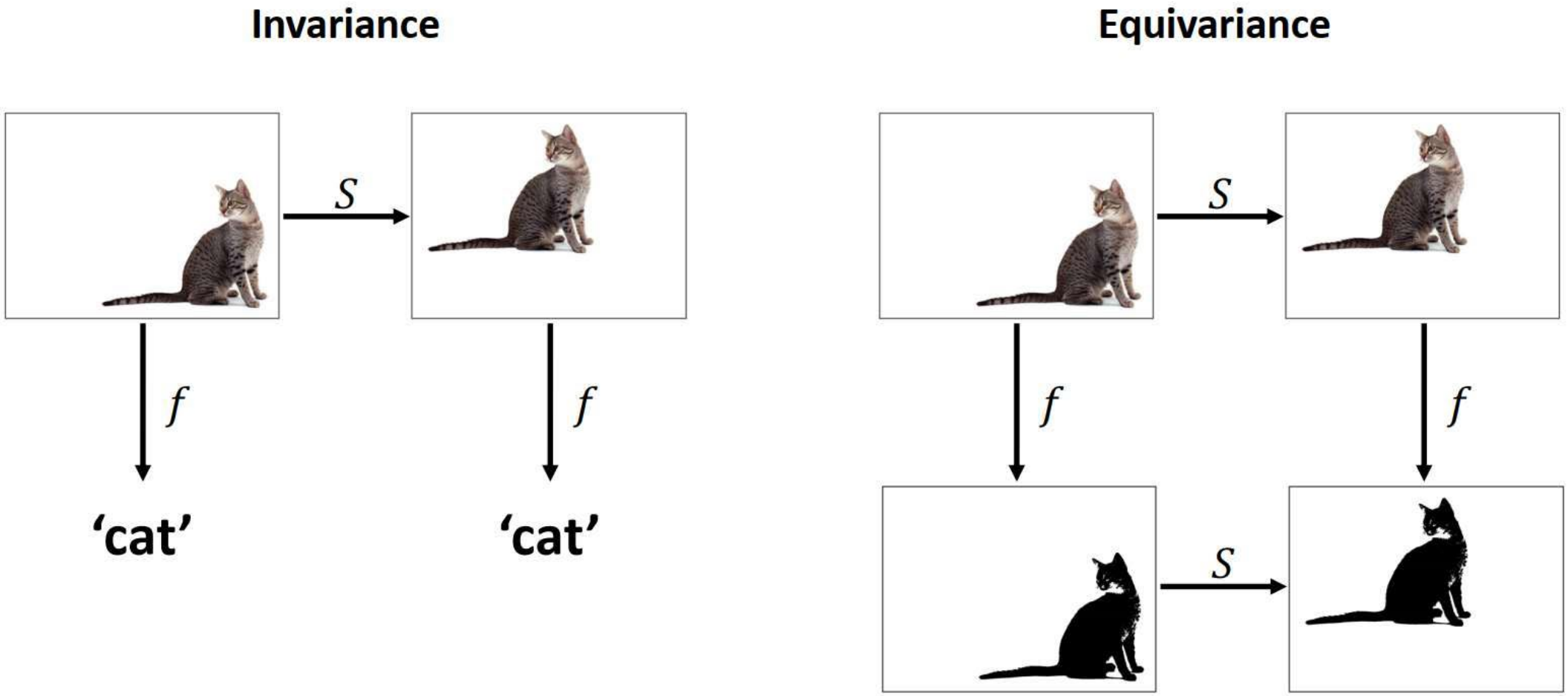
$P: X(\Omega) \rightarrow X(\Omega')$

- Simetrija – transformacija koja ostavlja određena svojstva objekta ili sustava nepromijenjenim ili invarijantnim
- Razdvajanje skala - mogućnost čuvanja važnih karakteristika signala kada ga se prebacuje u grublju verziju domene (npr. uzorkovanje slike pogrubljenjem koordinatne mreže)

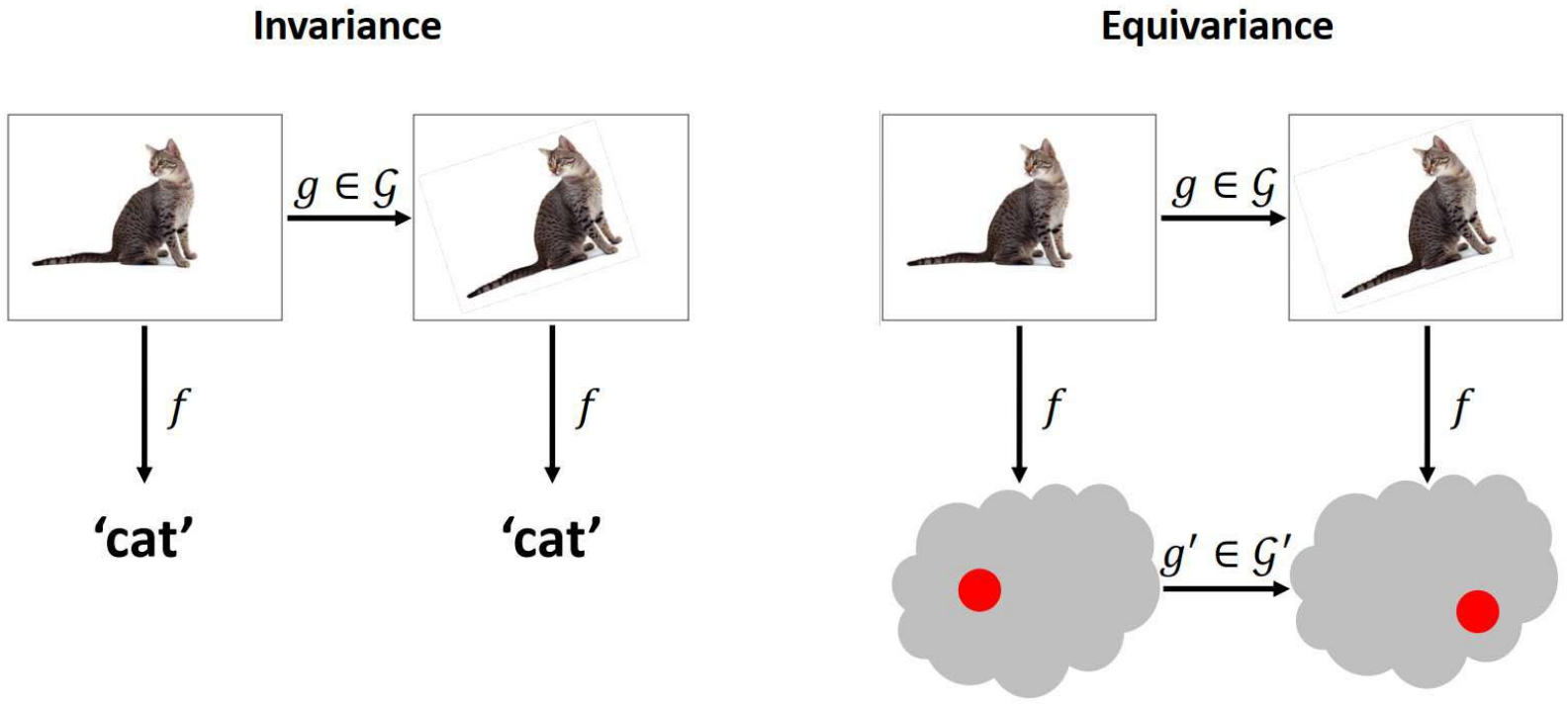
Induktivna pristranost

- Učenje – proces usvajanja korisnog znanja kroz promatranje i interakciju sa svijetom.
- Često više jednako dobrih rješenja
- Induktivna pristranost dopušta algoritmu za učenje da prioritizira jedno rješenje pred drugima, neovisno o podacima
- Primjeri: regularizacija, uključenost u samu arhitekturu
- Idealno induktivna pristranost poboljšava pretragu za rješenjem bez gubitka performansi te pomaže pronaći rješenje koje poopćava u poželjnom smjeru

Invarijantnost i ekvivarijatnost pomaka



Ekvivarijantnost grupa



$$f(g(\mathbf{x})) = g'(f(\mathbf{x}))$$

Grupe su jednostavno skupine operacija koje imaju svojstvo zatvorenosti. To su bilo koje operacije u kojima kombiniramo dva elementa grupe i dobivamo neki drugi element te grupe. Primjer: rotacije. Dvije uzastopne rotacije se može izvesti kao jedna veća.

Kako dizajnirati neuronske mreže

- Identificirati svojstva podataka/struktura/invarijantnosti koje su najviše opća i minimalistička
- Dizajnirati slojeve koji mogu uhvatiti te strukture, i naslagati puno njih
- Neuronske mreže (NN) su teške za debugiranje. Loše NN čine se da rade, ali su spore za treniranje, zahtijevaju puno podataka i loše poopćavaju

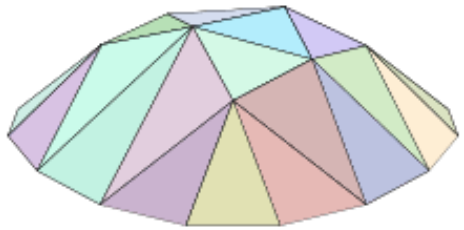
Glavne klase neuronskih mreža

- MLP (engl. multilayer perceptron) – potpuno povezane mreže
- CNN (engl. convolutional neural network) – konvolucijske mreže
- RNN (engl. Recurrent neural network) – povratne mreže

MLP/FC mreže

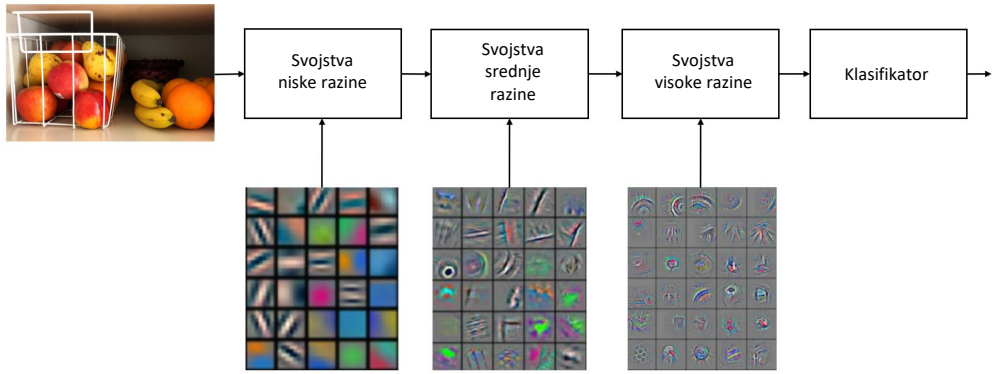
- Struktura podataka
  - Ulazni podaci se sastoje od uzoraka/predložaka fiksne veličine
- MLP izvodi podudaranje uzorka za fiksnu ulaznu veličinu
- Kompozicija funkcija:

  $F_0 \circ F_1 \circ \ldots \circ F_{L-1} \circ F_L$
  gdje $F_l(x_{l-1})=\sigma(W_l x_{l-1} + b_l) = x_l$
  gdje $W_l \; i \; b_l$ se uče koristeći SGD
- Performanse:
  - Odlično za po dijelovima linearne podatke



CNN

- Struktura podataka
  - Ulazni podaci imaju rešetkastu strukturu poput 2D/3D slika i videa
- CNN hvata svojstva slika
  - Lokalna i globalna translacijska invarijantnost na rešetci
    - Objekt je prepoznat neovisno o svojoj poziciji
  - Invarijantnost na rešetci za lokalne deformacije
  - Hijerarhijska reprezentacija podataka (svojstvo višeskalarnosti)
- Performanse
  - Izvrsno za rešetkasto strukturirane podatke



CNN

- Kompozicija funkcija:

$$F_0 \circ F_1 \circ \ldots \circ F_{L-1} \circ F_L$$

gdje $F_l(x_{l-1}) = \sigma(A_l * x_{l-1} + b_l) = x_l$

$*$ je operator konvolucije

A_l je jezgra fiksne veličine

A_l i b_l se uče koristeći SGD

- Konvolucijska operacija:
  - Linearna operacija (slična MLP), ali specijalizirana to podudaranje predloška sa pomičnim prozorom (translacija)
  - Neovisna o ulaznoj veličini, iako se slike obično prilagođavaju na istu veličinu (zbog računalne efikasnosti grupe/GPU)

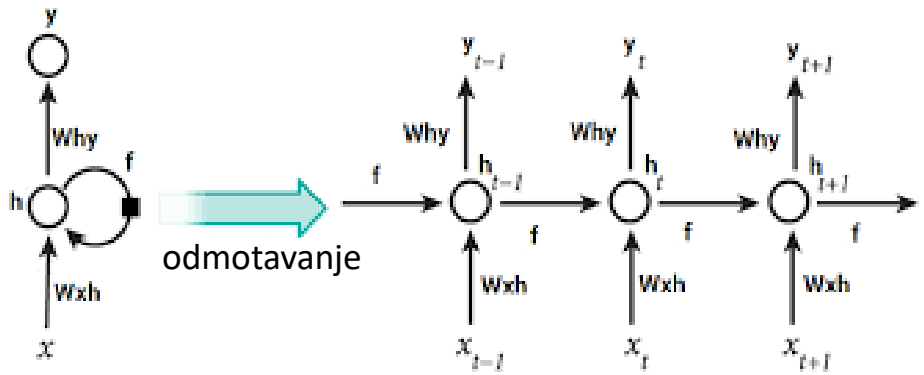
RNN

- Struktura podataka
  - Ulazni podaci su uređeni sljedovi (ili 1D rešetke)
  - Ulazne i izlazne duljine mogu varirati
- RNA su dizajnirane za sljedove
  - Uče reprezentaciju sljedova neovisno o duljini
    - Rekurzivna formula sumira sekvencu s vektorom h:

$$h \leftarrow f_W\ (h, x)$$

    - Dijeljenje težina kroz vrijeme (translacijska invarijantnost)
  - Zadržavanje ili ignoriranje informacije u slijedu za naredne zadatke
    - Mehanizam propusnice za zaboravljanje/pamćenje prošlosti ili novog ulaza:

$$\sigma \odot h$$



RNN

- Performanse:
  - Značajan napredak u NLP, ali ne proboj
  - Dominante u NLP do 2018
- Ograničenja
  - Ne mogu naučiti dugotrajne zavisnosti (ne više od 50 koraka)
  - Teške za treniranje zbog toga što su nelinearni dinamički sustavi
    - Bilo koja mala perturbacija može pojačati ili iščeznuti
  - Spore za treniranje zbog slijedne prirode (za razliku od CNN)
    - Važno ograničenje kada se trenira na velikim skupovima podataka

Optimizacija temeljena na gradijentu

- Rezultat = ReLu(dot(input, W) + b)
- W i b, težine, parametri koje možemo učiti (jezgra, pristranost)
- Inicijalizacija – male slučajne vrijednosti
- Učenje – postepeno podešavanje težina temeljeno na ulaznom signalu

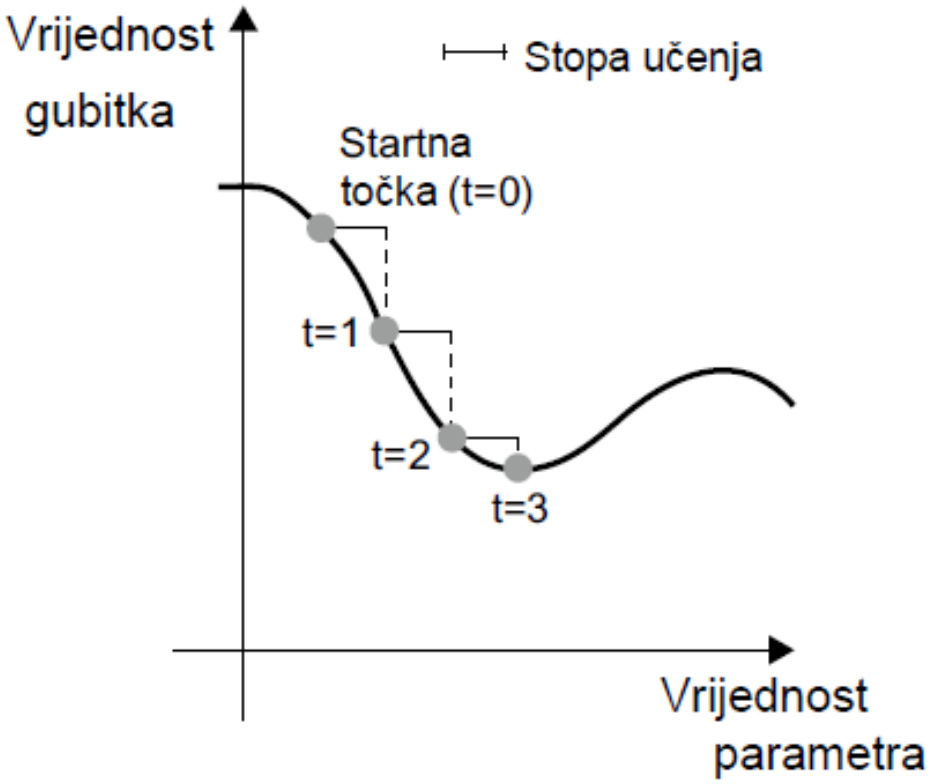
Učenje

1. Izvuči grupu primjera za učenje, **x** i pripadajućih izlaza **y_true**
2. Pokrenuti model na **x** (prolaz prema naprijed) i dobiti predviđanja, **y_pred**
3. Izračunati gubitak na toj grupi, mjera nepodudaranja između **y_true** i **y_pred**
4. Izračunati gradijent gubitka u odnosu na parametre modela (prolaz unatrag)
5. Malo pomaknuti parametre u smjeru suprotnom od gradijenta

Stohastički gradijentni spust (SGD)

- Stohastički – u svakom ciklusu uzimamo mini grupu podataka na slučajan način (mini grupni SGD)
- Stvarni SGD - u svakom krugu uzimamo samo jedan podatak
- Grupni SGD – uzeti sve podatke. Osvježavanje točnije, no računalno skupo
- Kompromis s manjom grupom

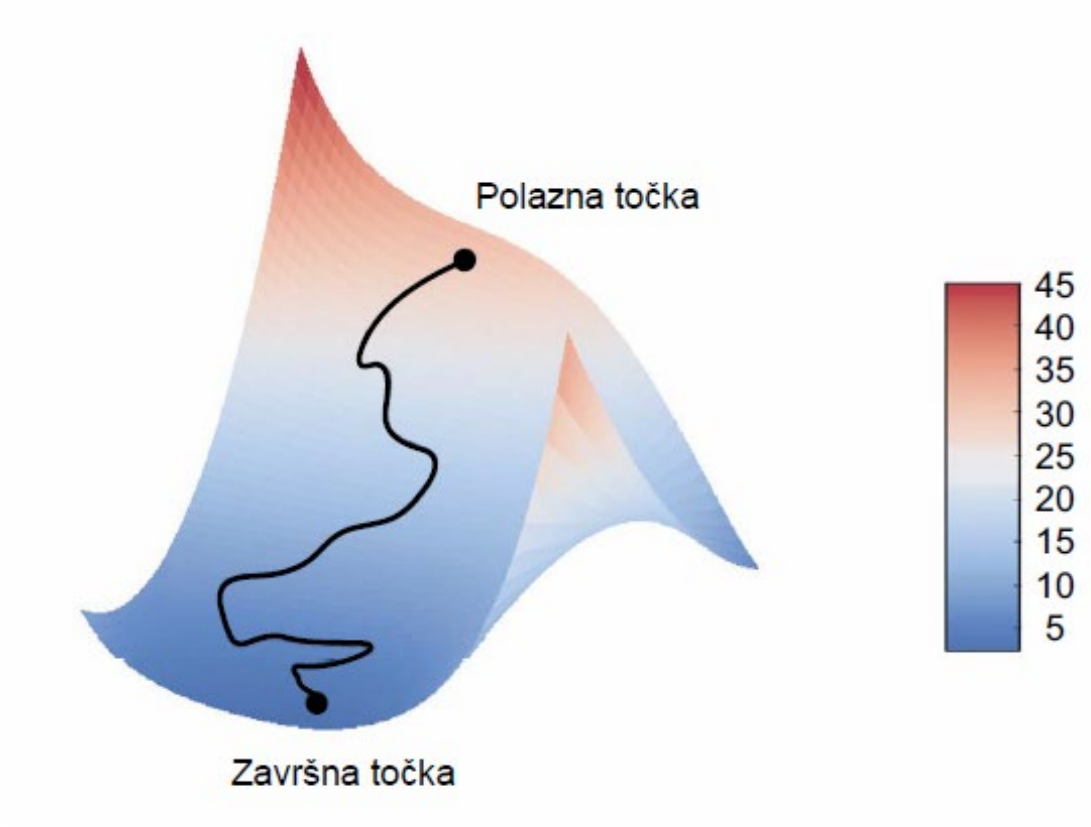
Stopa učenja



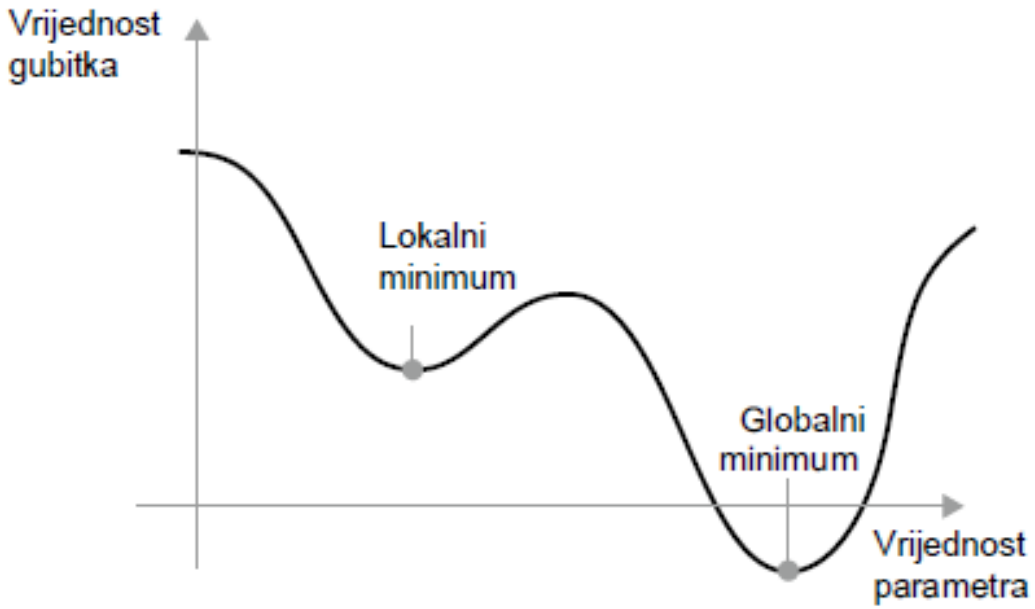
Premala stopa – polagano učenje i mogućnost da zaglavimo u lokalnom minimumu
Prevelika stopa – prilikom osvježavanja možemo završiti u slučajnom dijelu krivulje

Chollet, François. Deep Learning with Python, Second Edition

Gradijentni spust u 2D



Moment



Ideja pokušati riješiti brzinu konvergencije i lokalnog minimuma.

Primjer iz fizike Kuglica s dovoljnim momentom će izletjeti iz lokalnog minimuma i završiti u globalnom. Koristimo ne samo lokalnu brzinu već i ubrzanje

Chollet, François. Deep Learning with Python, Second Edition

Evaluacija modela

- Podjela na skup za učenje, validaciju i testiranje
- K-struka unakrsna validacija
- Očekivana osnovna vrijednost točnosti (npr. 0.1 za MNIST)

Važno kod validacije modela

- Reprezentativnost podataka
  - Učenje na znamenkama 0 – 7 a testiranje na 8 i 9 (LOŠE!)
  - **Rješenje**: Slučajno uzorkovanje
- Smjer u kome vrijeme teče
  - Predviđanje događanja u budućnosti na osnovi prošlosti (vrijeme, kretanje dionica, …)
  - Podaci se ne smiju promiješati na slučajan način –> vremensko curenje (učenje koristeći podatke iz budućnosti
  - Podaci u test skupu moraju biti noviji od podatak u skupu za učenje
- Redundantnost u podacima
  - Često u stvarnim podacima imamo duple podatke
  - Problem ako završe i u skupu za treniranje i skupu za validaciju

Poboljšanje prilagođenja modela

- **Za postići perfektno prilagođenje, potrebno je prvo pretrenirati!**
- S obzirom da ne znamo gdje je granica, moramo ju prvo proći da bi ju mogli naći.
- Inicijalni cilj je pronaći model koji pokazuju neku snagu poopćenja, može pretrenirati
- Kada pronađemo takav model, fokus je na profinjenju poopćenja borbom protiv pretreniranja

Osnovni problemi

1. Učenje nije počelo. Gubitak učenje ne opada s vremenom
2. Učenje je počelo u redu, ali model ne poopćava dovoljno – ne postižemo bolje od podrazumijevane osnovne točnosti (engl. common-sense baseline)
3. Gubitak učenja i validacije opada, postižu se rezultati bolji od podrazumijevane osnovne točnosti, no ne može se pretrenirati -> podtreniranje

Moguća rješenja

- Podešavanje parametara gradijentnog spusta
- Korištenje adekvatnije arhitekture
- Povećanje kapaciteta modela

Podešavanje parametara gradijentnog spusta

Problem: Učenje ne krene, ili se zaustavi prerano.

- Mora se moći popravi. Svaki model bi se morao prilagoditi makar i sa slučajnim ulaznim podacima. Ako ništa drugo onda memoriranje podataka za učenje
- Problem gradijentnog spusta. Odabir optimizera, distribucije inicijalni težina modela, stope učenja ili veličine grupe (engl. batch)
- Ovi parametri su međusobno zavisni -> često je dovoljno promijeniti stopu učenja i veličinu grupe zadržavajući vrijednosti ostalih parametara.

Podešavanje gradijentnog spusta – praktični savjeti

- **Stopa učenja**
  - Previsoka - može voditi u osvježavanja koja ne mogu pogoditi ispravno prilagođenje
  - Preniska – učenje presporo pa imamo dojam da je učenje stalo

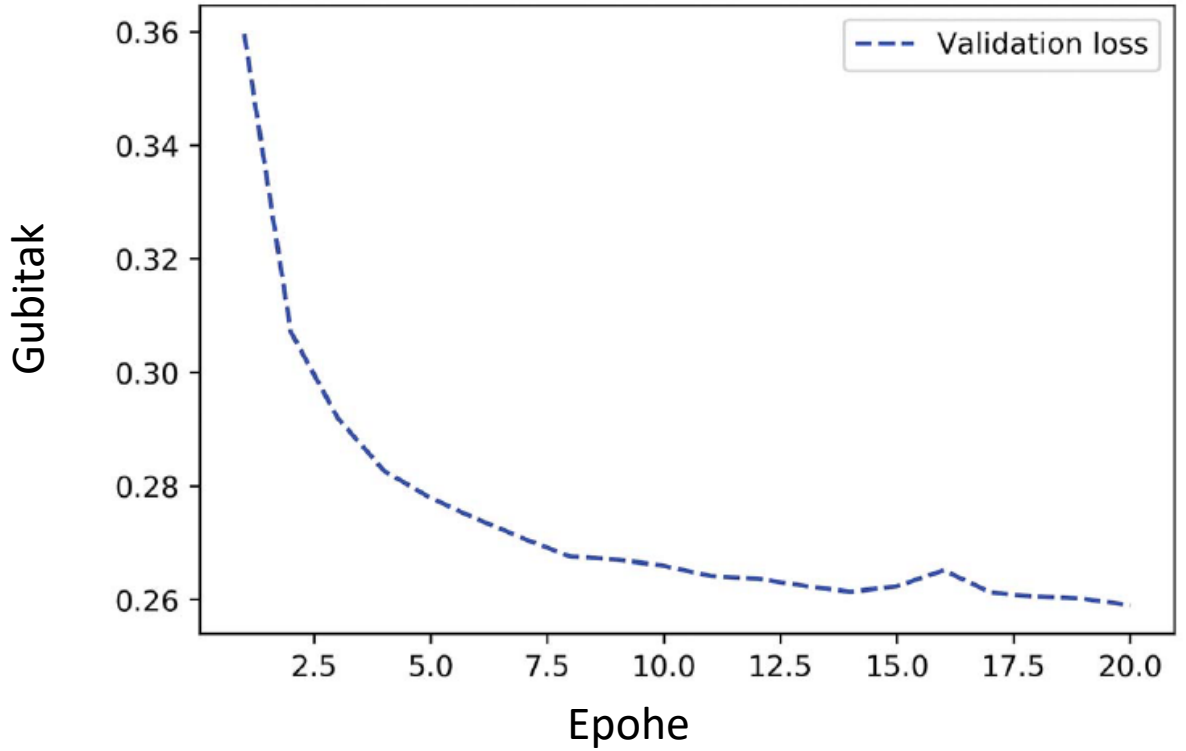
- **Veličina grupe**
  - Veća grupa će voditi gradijentima koji su informativniji i manje šumoviti (niža varijanca)

Korištenje adekvatnije arhitekture

- Model se prilagođuje, no metrike validacije pokazuju da rezultati nisu bolji od slučajnog klasifikatora
- Model uči, ali ne poopćava
- Mogući uzroci:
  - Ulazni podaci jednostavno nemaju dovoljno informacija za predviđanje. Problem na način na koji je definiran je nemoguće riješiti
  - Korištena arhitektura modela nije odgovarajuća

Povećanje kapaciteta modela

- Model se prilagođuje, metrike na validacijskom skupu se spuštaju i čin se da se postiže barem neka razina poopćenja. Na dobrom smo putu!

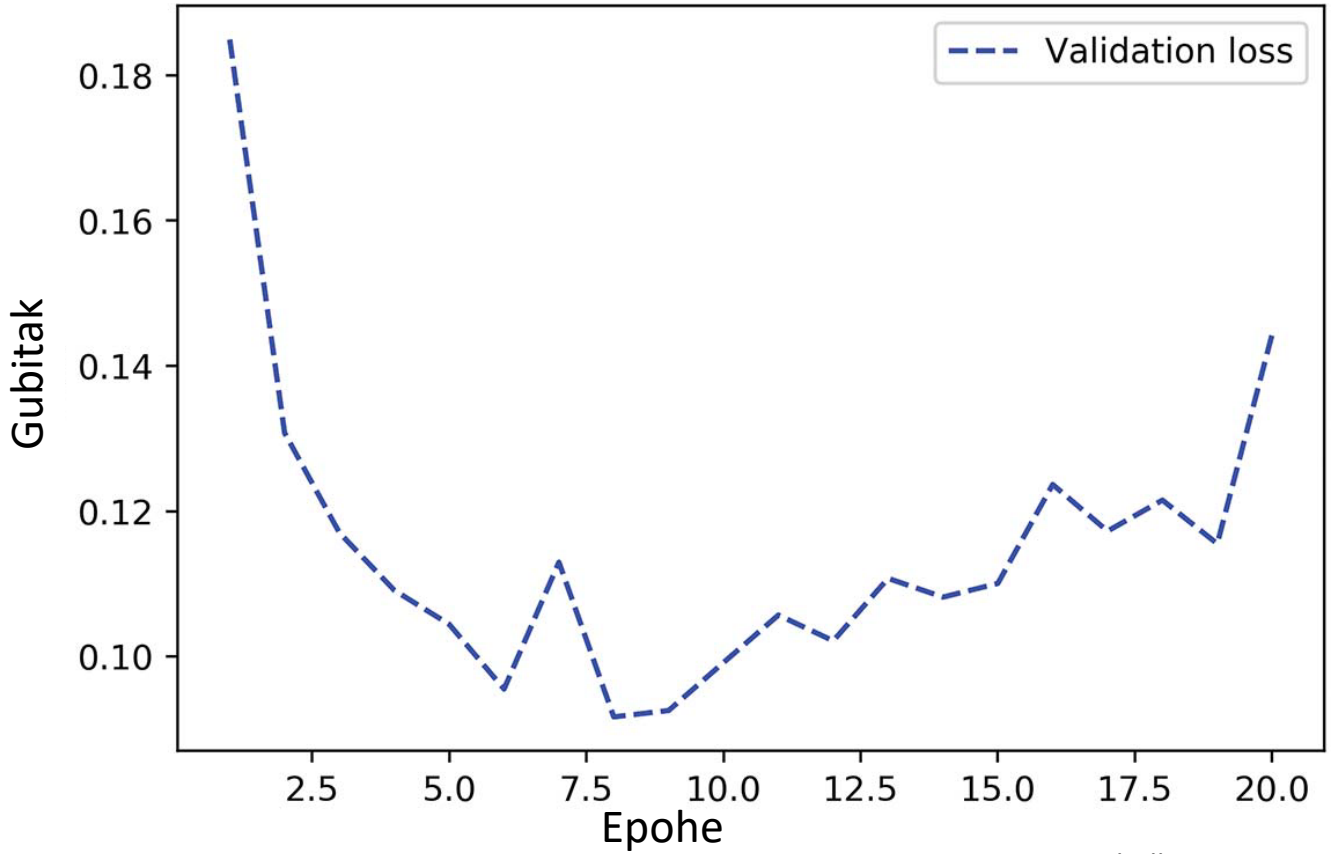


MNIST korištenjem obične logističke regresije

Chollet, François. Deep Learning with Python, Second Edition

Povećanje kapaciteta modela

- Uvijek moramo moći pretrenirati!
- Dodavanje dva dodatna sloja u jednostavni model



Chollet, François. Deep Learning with Python, Second Edition

Poboljšanje poopćenja

- Prikupiti više podataka za učenje ili bolje podatke za učenje
- Razviti bolje značajke
- Smanjenje kapaciteta modela
- Dodati regularizaciju težina (za manje modele)
- Dodati isključivanje pojedinih čvorova (*engl. dropout*)

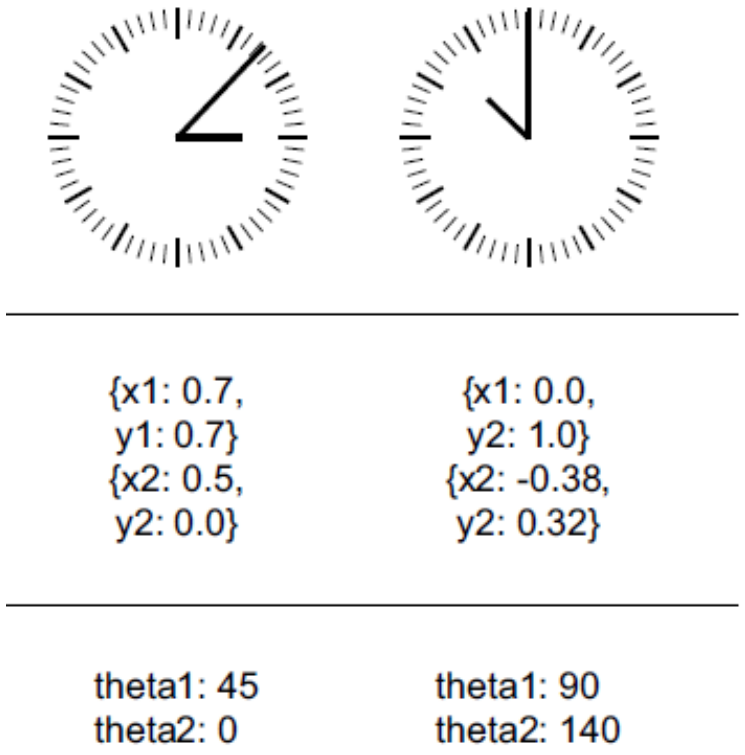
Trošenje više truda i novaca na prikupljanje podataka gotovo uvijek donosi veći povrat investicije nego trošenje istog iznosa na razvijanje boljih modela

Prikupljanje više podataka za učenje ili boljih podataka za učenje

- Biti sigurni da imamo dovoljno podataka. Važno je da zapamtiti da trebamo imati gusto uzorkovanje podataka da bi mogli dobro interpolirati. Više podataka -> bolji model
- Smanjiti greške u označavanju – vizualizirati ulazne podatke i provjeriti za anomalije te pažljivo provjeriti oznake
- Očistiti podatke i znati kako rukovati s podacima koji nedostaju
- Ako imamo puno značajki i nismo sigurni koje su korisne, odabrati značajke (*engl. feature selection*)

Ručna konstrukcija značajki

- Proces korištenja vlastitog znanja o podacima i korištenom algoritmu strojnog učenja u cilju postizanja da algoritam radi bolje primjenom rukom kodiranih transformacija podataka
- Vrlo često nije razumno očekivati da će model naučiti iz potpuno arbitrarnih podataka
- Podaci trebaju biti predstavljeni modelu na način da olakšaju njegov posao

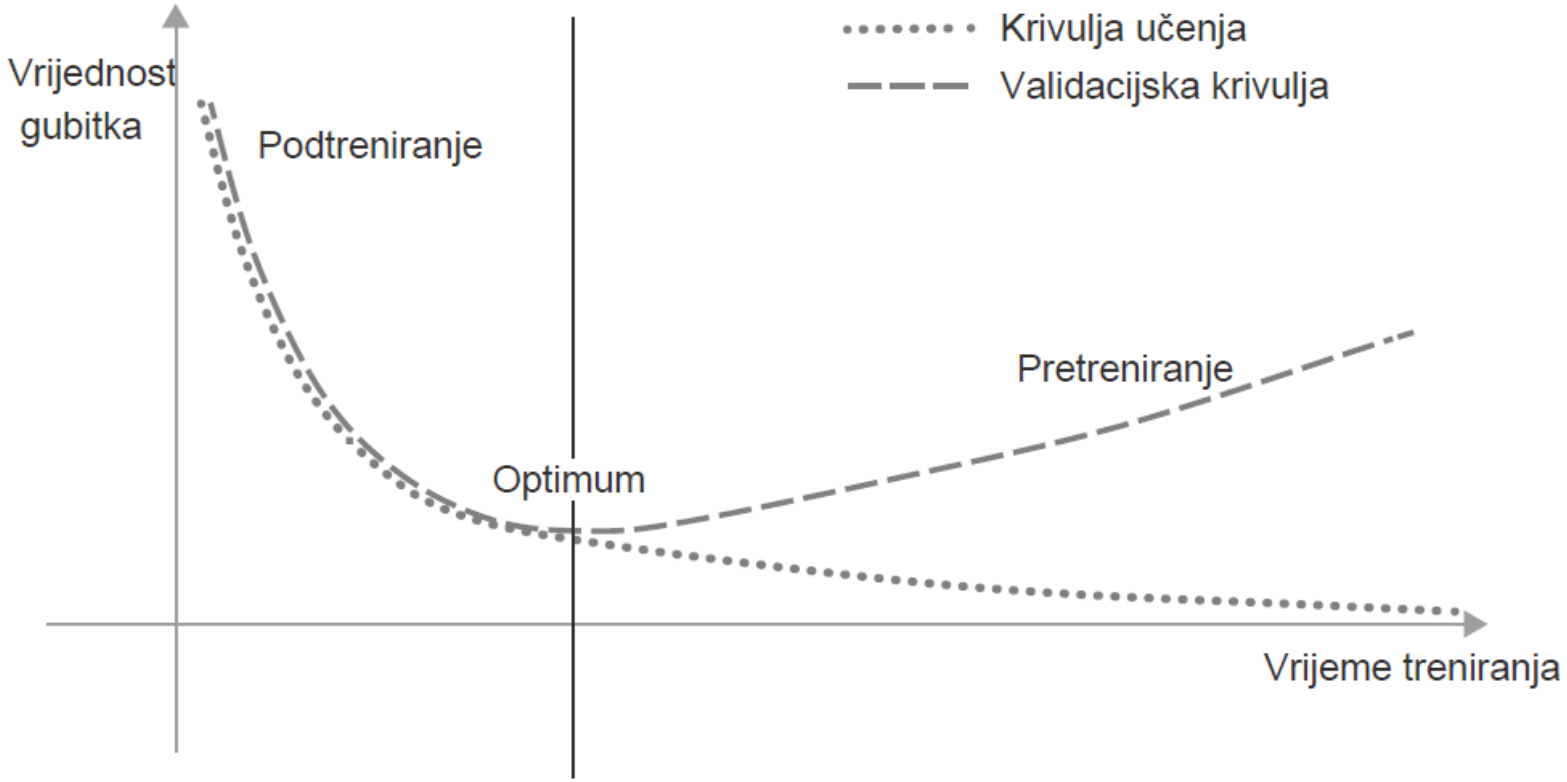


Chollet, François. Deep Learning with Python, Second Edition

Ručna konstrukcija značajki i duboko učenje

- Suvremene neuronske mreže su sposobne automatski izvući korisne značajke iz sirovih podataka. Pa ipak…
- Za učenje značajki modelima dubokog učenja treba ogromna količina dostupnih podataka
- Dobre značajke omogućavaju elegantnije rješavanje problema s manje resursa
- Dobre značajke omogućavaju rješavanje problema s manje podataka.

Rano zaustavljanje



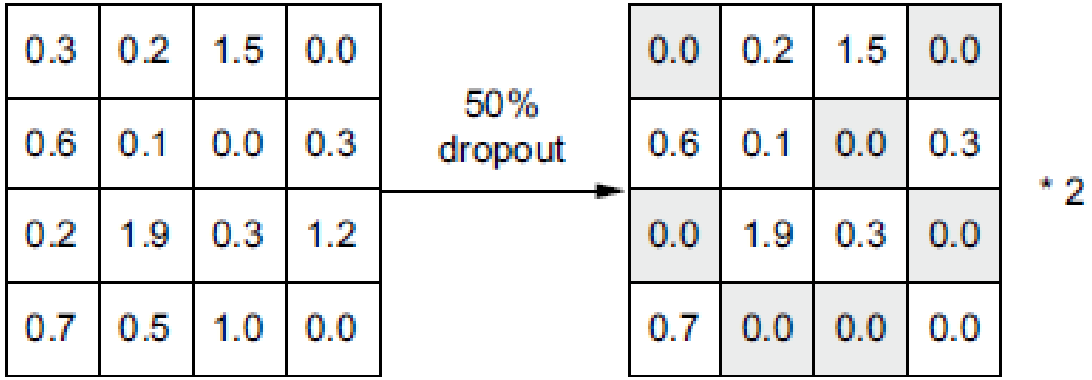
Chollet, François. Deep Learning with Python, Second Edition

- **Regularizacijske tehnike**. Skup postupaka koji aktivno ometaju sposobnost modela perfektnom prilagođenju podacima za treniranje sa ciljem boljeg ponašanja modela za vrijeme validacije

Regularizacija

- Occamova britva (obrazloženje koje uzima manje pretpostavki)
- Reduciranje veličine modela (kompromis između previše i premalo kapaciteta)
- Smanjenje kompleksnosti modela regularizacijom težina
  - Forsiranje da težinu imaju manju vrijednost
  - L2 regularizacija
  - L1 regularizacija
  - Koristi se za manje modele
- Isključivanje pojedinih čvorova (*engl. dropout*)

Isključivanje pojedinih čvorova



Slučajno ispuštanje određenog broja izlaznih značajki sloja tijekom učenja

U testnom vremenu, ni jedna jedinica nije isključena, a vrijednost izlaza su umanjene s faktorom jednakim stopi isključivanja zbog proporcionalno više aktivnih jedinica nego tijekom učenja.

Chollet, François. Deep Learning with Python, Second Edition

Konvolucija - ideja

- F = [.05, .03, .01] postotak spajanja na respirator tjedno
- G = [100, 200, 300, 200, 100, 100] broj novo hospitaliziranih tjedno
- F * G (* simbol za konvoluciju)

Hospitalizirani 100 200 300 200 100 100

Postotak na respiratoru 0.01 0.03 0.05

Broj na respiratoru 5

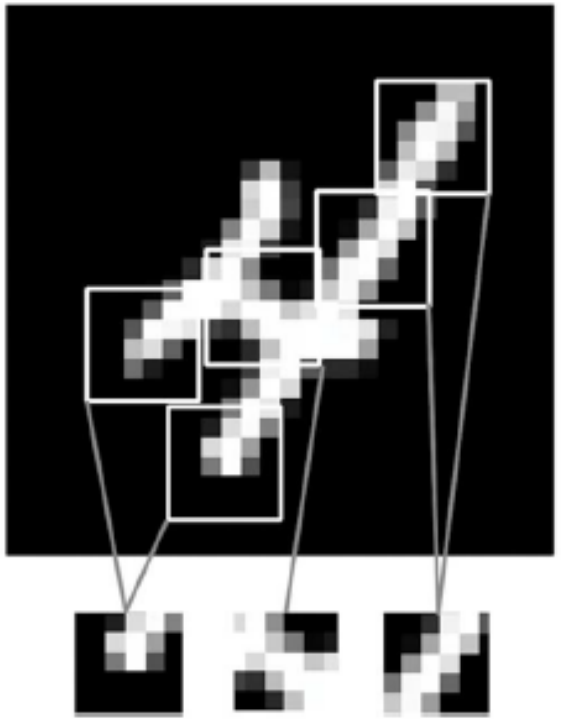
$$C[n] = \sum_{k=-\infty}^{k=\infty} G[k] \cdot F[n - k]$$

Primjer konvolucijske mreže

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 5, 5, 64)	0
conv2d_2 (Conv2D)	(None, 3, 3, 128)	73856
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 10)	11530

[batch_size, image_height, image_width, no_channels]

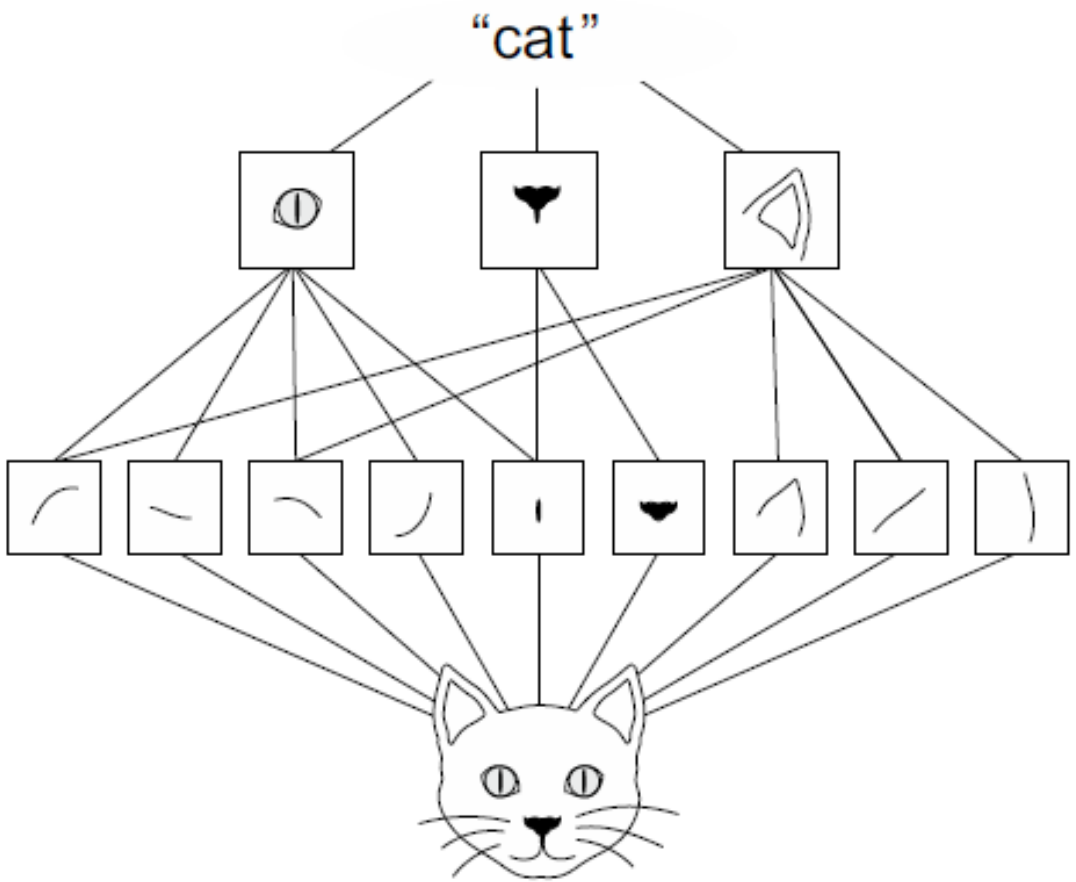
Konvolucijske mreže



MLP uče globalne uzorke
CNN uče lokalne uzorke

Chollet, François. Deep Learning with Python, Second Edition

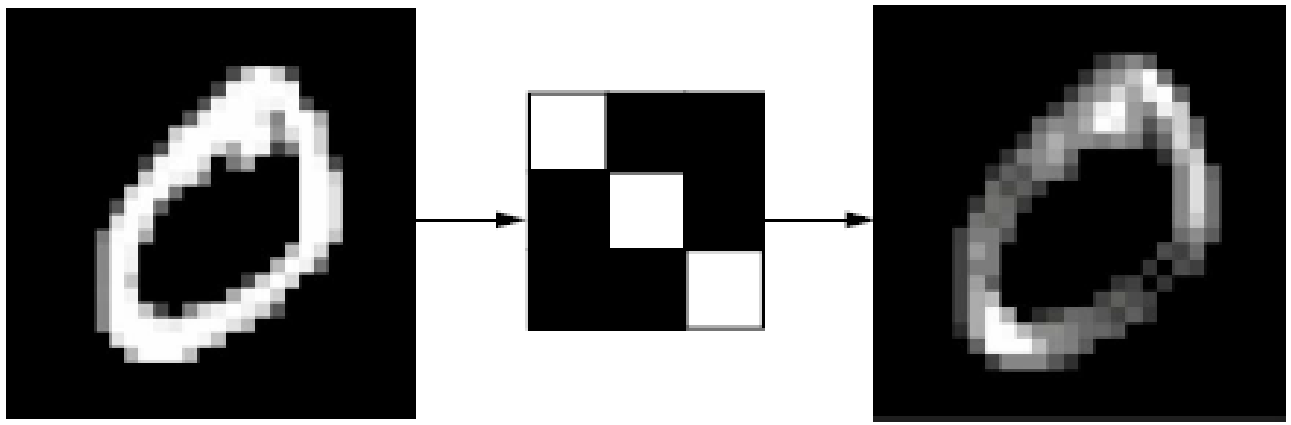
Konvolucijske mreže



1. Naučeni uzoci su invarijantni na translaciju
2. Konvolucijske mreže uče prostorne hijerarhije uzoraka

Chollet, François. Deep Learning with Python, Second Edition

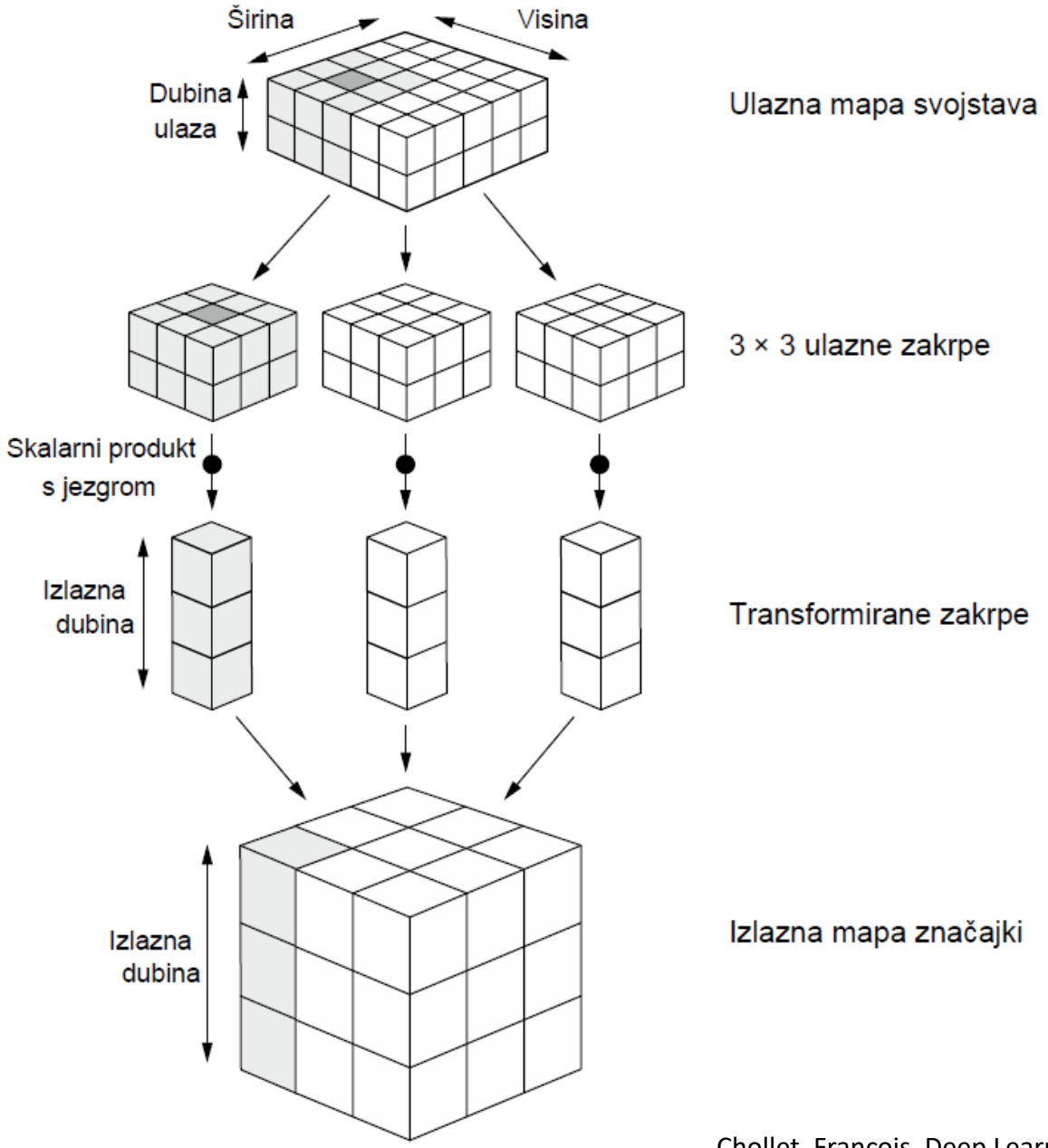
Konvolucijske mreže



Ključni parametri

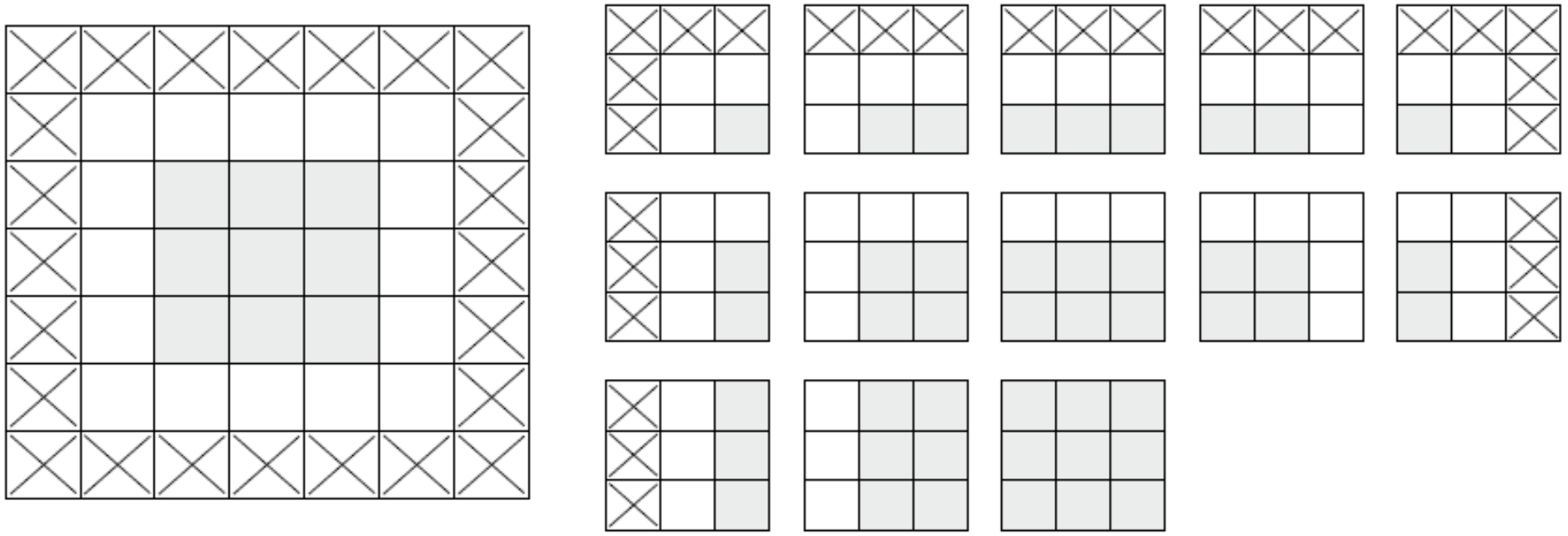
1. Veličina zakrpa izvučenih iz inputa
   - obično 3 × 3 ili 5 × 5
2. Dubina izlazne mape svojstava
   - broj filtara izračunatih konvolucijom
   - krenuli s 32 i završili sa 64

Chollet, François. Deep Learning with Python, Second Edition



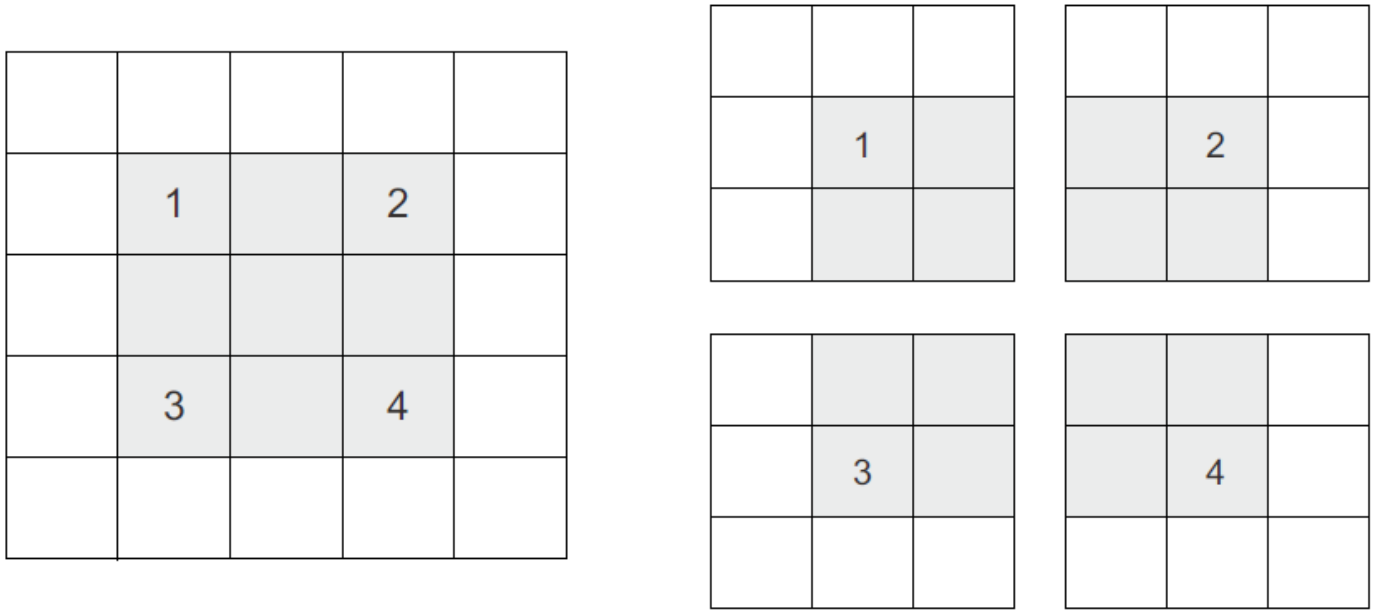
Chollet, François. Deep Learning with Python, Second Edition

Nadopunjavanje (engl. padding)



Chollet, François. Deep Learning with Python, Second Edition

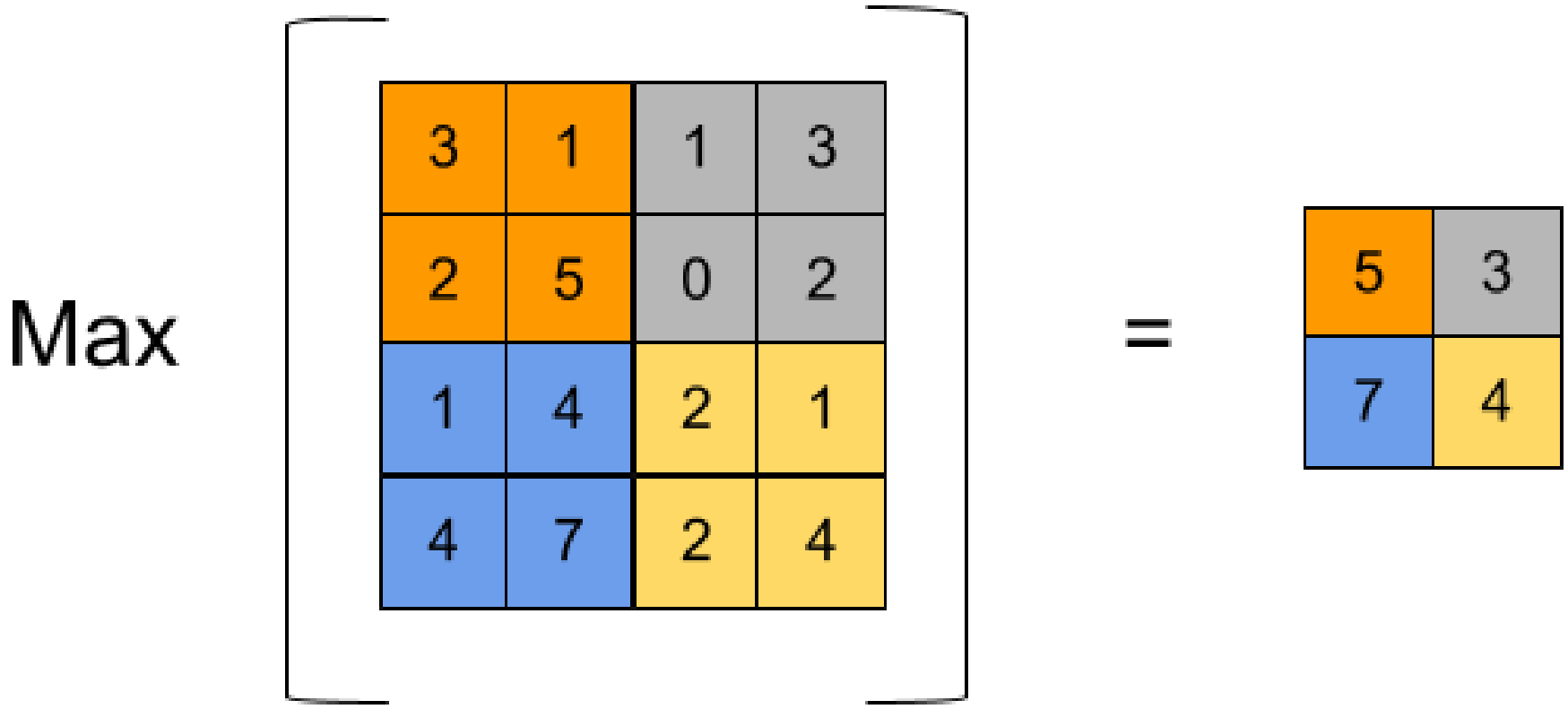
Korak (*engl. stride*)



3 × 3 konvolucijske zakrpe s koracima 2 × 2

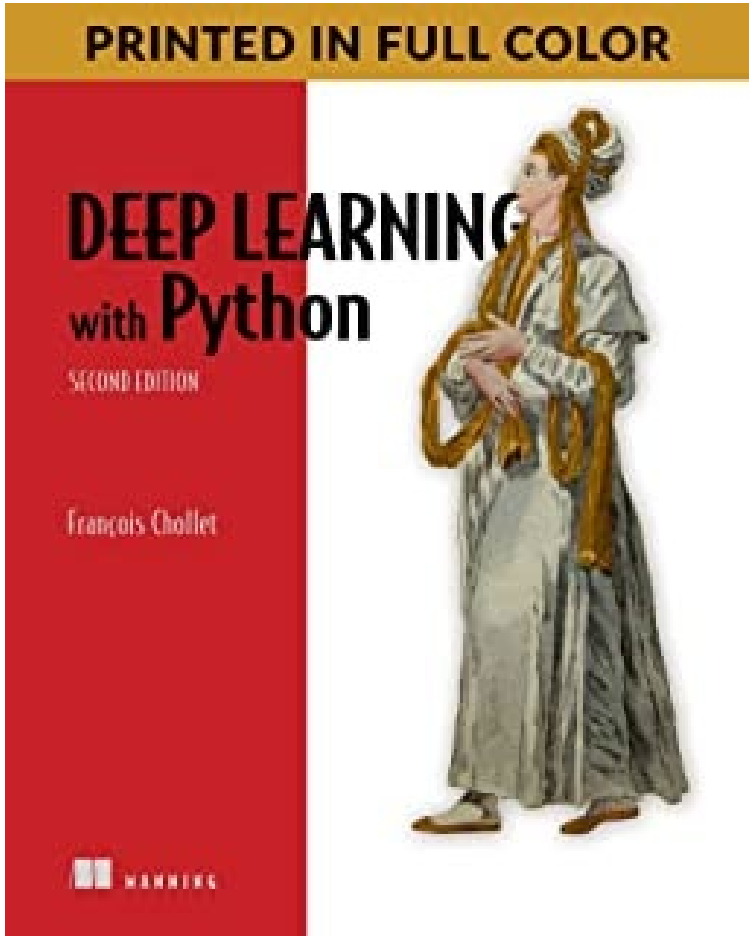
Chollet, François. Deep Learning with Python, Second Edition

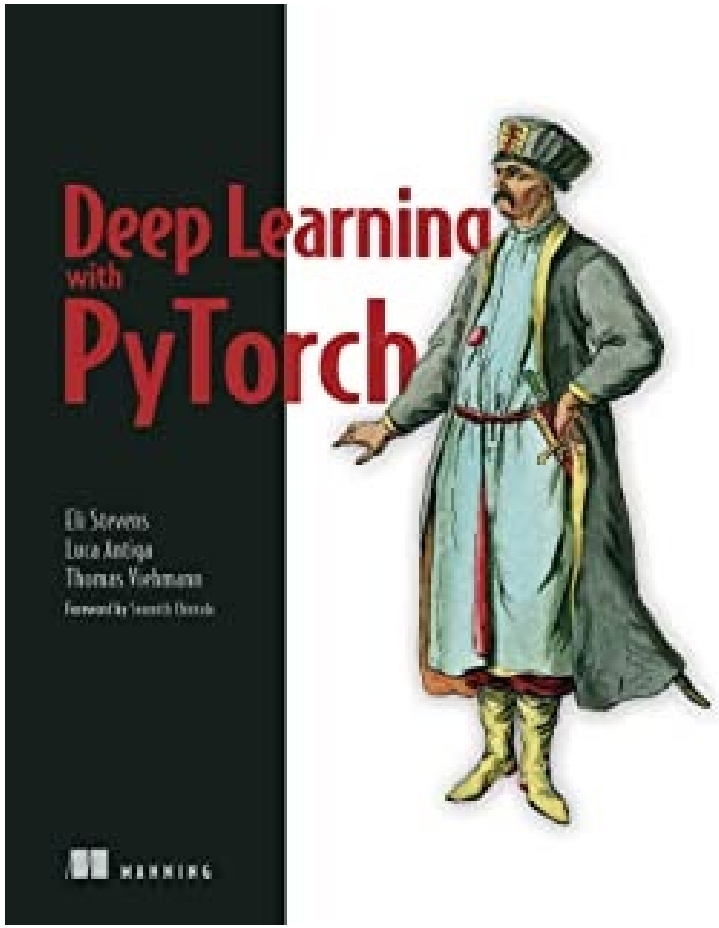
Združivanje maksimalnih vrijednosti (engl. Max pooling)



2 × 2 prozori s korakom 2

Literatura


PRINTED IN FULL COLOR
DEEP LEARNING
with Python
SECOND EDITION
François Chollet
MANNING


Deep Learning
with
PyTorch
Eli Stevens
Luca Antiga
Thomas Viehmann
MANNING

Znanstvenici i podcasti

- Geoffrey Hinton, Yann LeCun, Yoshua Bengio
- François Chollet
- Max Welling
- Michael Bronstein
- Judea Pearl

- Podcasts: Machine learning street talk

Pošaljite mi svoje mišljenje o ovom predavanju na:
mile.sikic@fer.hr

- Što (ni) ste voljeli na ovom predavanju?
- Što (ni) je bilo dobro objašnjeno?
- O čemu (ne) biste željeli čuti više detalja?
- …