

# Duboko učenje 1

## Uvodno predavanje

Siniša Šegvić

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

# SADRŽAJ UVODNOG PREDAVANJA

- **Motivacija** za duboko učenje:
  - strojno učenje i umjetna inteligencija
  - kompozitni podatci (sastoje se od dijelova)
- **O predmetu:**
  - glavne teme
  - način održavanja nastave, razdioba bodova, literatura
- **Pregled osnova *strojnog učenja*:**
  - osnovni pojmovi i tehnike
  - primjeri algoritama
- **Pregled suvremenih izazova i prednosti *dubokog učenja***

# MOTIVACIJA: STROJNO UČENJE

- **Strojno učenje:** proučava algoritme čija funkcionalnost nije unaprijed isprogramirana nego je zadana primjerima za učenje:
  - jedan od središnjih problema **umjetne inteligencije**
  - zanimljivi su i hibridni algoritmi koji se dijelom uče a dijelom su ožičeni
- **Umjetna inteligencija:** proučava izradu strojeva koji "misle":
  - zadatci koji su laki za ljude, a vrlo teški za računala
- Primjer: program koji u ovim slikama pronađe kravu
  - inteligentno ponašanje lakše je naučiti nego konstruirati.



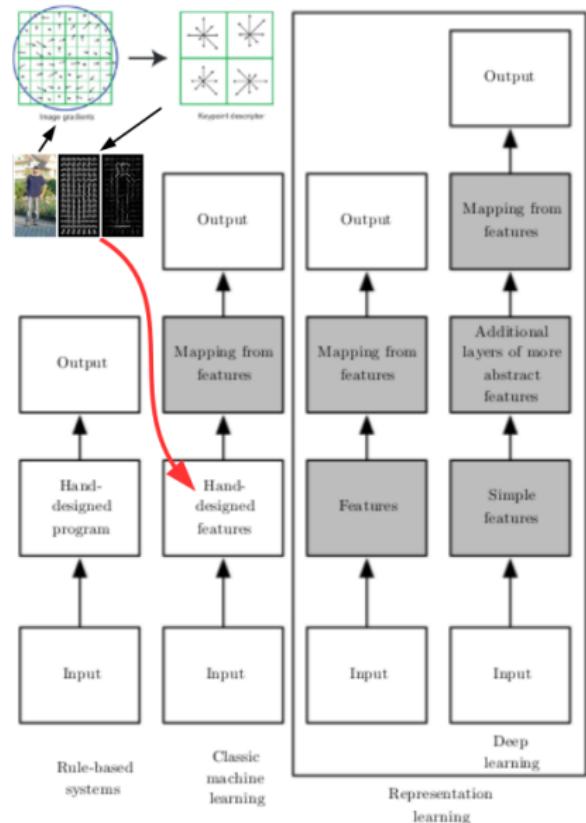
# MOTIVACIJA: UMJETNA INTELIGENCIJA

Odnos AI vs ML:

- rani pristupi: nema učenja
- klasični pristupi: naučiti plitki model na konstruiranim značajkama
- reprezentacijski pristupi: naučiti i značajke i model
- duboki pristupi: odjednom naučiti (engl. end to end) niz transformacija

**Učenje** postaje sve važnije!

**Podatkovne reprezentacije** isto!



[goodfellow16]

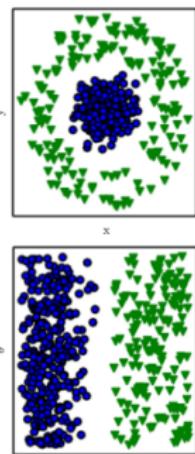
## MOTIVACIJA: PODATKOVNE REPREZENTACIJE

Složenost obrade često ovisi o reprezentaciji podataka:

- za dani zadatak, neke reprezentacije mogu biti pogodnije od drugih
- MCMLXXI + XIX vs 1971 + 19?
- polarna reprezentacija problema na slici desno može se klasificirati plitkim modelom

Ponekad značajke nije lako konstruirati ručno

- Grci i Rimljani u više od 1000 godina civilizacije nisu uspjeli izumiti položajni brojevni sustav
- to je značajno otežalo razvoj matematike
  - MCMLXXI + XXIX = ?
  - MXXIV : LXIV = ?
- ⇒ naučene reprezentacije su zanimljive!



[goodfellow16]

# MOTIVACIJA: PRIMJER

Kako razlikovati slike goveda od slika bizona?



[image-net.org]

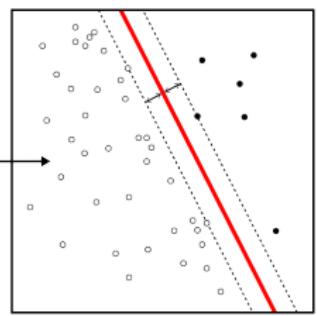
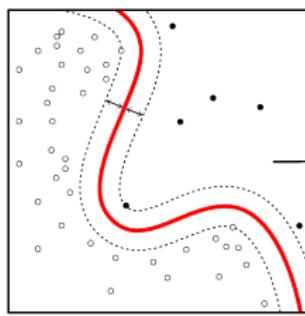
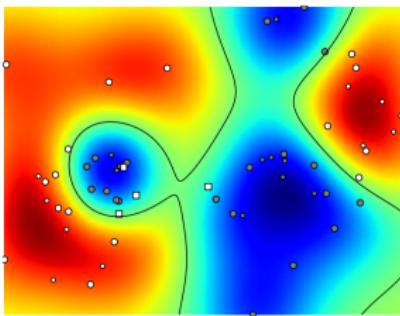
Težina klasifikacije jako ovisi o reprezentaciji podataka

- životinje bi bilo lako razlikovati kad bi neki magični algoritam pretvorio sliku u binarni vektor: [krzno?, grba?, divljina?, ...]
- većina bizona bi bili: [DA, DA, DA, ...]
- većina goveda bi bila: [NE, NE, NE, ...]

Najbolje: **istovremeno** učiti i reprezentaciju i klasifikacijsko pravilo!

## MOTIVACIJA: PLITKI MODELI

- Dominantni pristup strojnom učenju 1990-2006: ručno konstruirane značajke i plitki klasifikatori s konveksnim gubitkom
  - SVM, logistička regresija, generalizirani linearni modeli.
- U to vrijeme, prednosti plitkih modela bile su jasne:
  - konvergencija učenja je garantirana i brza
  - jezgreni trik osigurava ogroman reprezentacijski potencijal
  - kompetitivna uspješnost raspoznavanja u praksi



[Wikipedia]

- Ipak, ti postupci ne mogu razlikovati krave od bizona...

# MOTIVACIJA: DUBOKO UČENJE

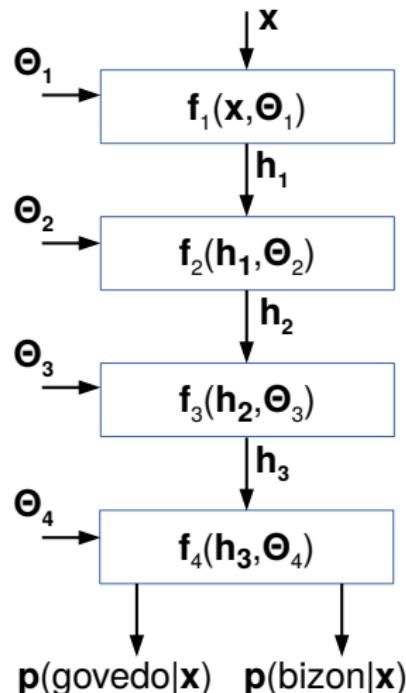
Duboki model: slijed naučenih nelinearnih transformacija.

Zašto duboko učenje nije bilo popularno?

- nema garancije uspjeha učenja
  - nekonveksni gubitak → lokalni minimumi
- nisu mogli nadmašiti stanje tehnike

Zašto je duboko učenje postalo uspješno?

- nove tehnike modeliranja i učenja
- veliki skupovi podataka ( $n=10^6$ )
- velika procesna moć (TFLOPS)
  - cuda, cuDNN, OpenBLAS, OpenMP



Primjene: razumijevanje slika, jezika i govora, bioinformatika itd.

## MOTIVACIJA: NO FREE LUNCH

Duboki modeli nisu "radili" kako treba jer nismo imali:

- dovoljno jake strojeve da dočekamo konvergenciju
- dovoljno podataka da ih naučimo
- tehnike koje pospješuju konvergenciju

Međutim, to ne pojašnjava zašto bi duboki modeli bolje generalizirali od drugih modela velikog kapaciteta (kSVM, stabla, ...)

- algoritme učenja ne možemo vrednovati neovisno o stvarnim podatcima: no free lunch theorem [domingos12cacm]

Uspješnost na neviđenim podatcima ovisi o **pristranosti** algoritma:

- prepostavke modela, gubitka ili optimizacijske metode
- pristranost odgovara podatcima  $\Rightarrow$  dobra generalizacija
- logistička regresija: izvrstan odabir za linearne razdvojive razrede

# MOTIVACIJA: KOMPOZITNI PODATCI

**Kompozitna struktura:** temeljna pretpostavka dubokih modela

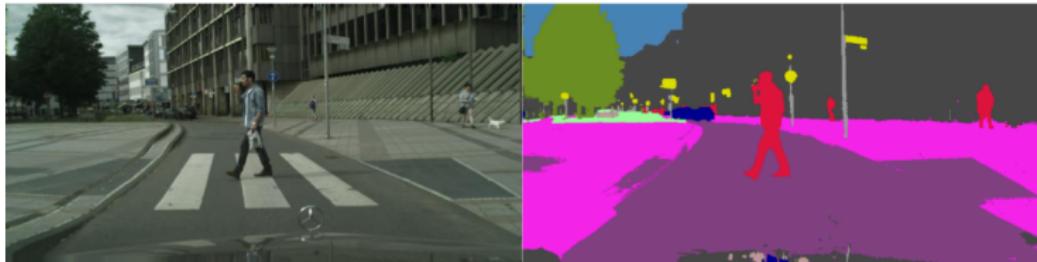
- podatkovna reprezentacija na razini n gradi se obradom reprezentacije na razini n-1
- to svojstvo dubokih modela dobro odgovara podatcima u mnogim teškim zadatcima
  - npr. auto ima kotače, kotači imaju naplatke, naplatci imaju rupice
  - slova čine riječi, riječi sintagme, a sintagme rečenice.



[zeiler14eccv]

# MOTIVACIJA: DUBOKO UČENJE - PRIMJENE

- Primjene:
  - računalni vid: klasifikacija slika, lokalizacija objekata, raspoznavanje simbola, semantička segmentacija



[kreso17iccvw]

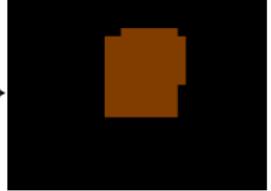
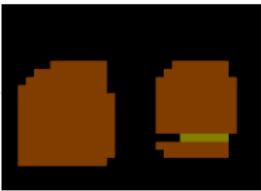
- obrada prirodnog jezika: analiza sentimenta, automatsko prevođenje, generiranje odgovora...
- pretraživanje informacija: učenje rangiranja
- raspoznavanje govora

# MOTIVACIJA: JOŠ JEDAN PRIMJER

- Neka su gornje slike zadane kao negativi te donje kao pozitivi:



- Kako biste pronašli gdje su objekti u prethodno neviđenim slikama?



[krapac16gcpr]

## O PREDMETU: PLAN

Što ćemo proučavati?

- Osnove unaprijednog dubokog učenja:
  - potpuno povezani modeli  
(prva laboratorijska vježba)
  - diskriminativni konvolucijski modeli  
(druga laboratorijska vježba)
  - optimizacijske metode
  - regularizacija
- povratni modeli  
(treća laboratorijska vježba)
- metrička ugrađivanja  
(četvrta laboratorijska vježba)

## O PREDMETU: LITERATURA

- Deep Learning. Ian Goodfellow, Yoshua Bengio and Aaron Courville. MIT Press 2016.
- Jan Šnajder, Bojana Dalbelo Bašić. Strojno učenje. FER, Zagreb.
- Neural Networks and Deep Learning. Michael Nielsen. Determination press 2015.
- Deep Learning: Foundations and Concepts. Christopher M. Bishop and Hugh Bishop. Springer 2024.
- Understanding Deep Learning. Simon JD Prince. MIT Press 2023.
- PyTorch tutorials and documentation
  - <https://pytorch.org/tutorials/>
  - <https://pytorch.org/docs/stable/index.html>

## O PREDMETU: PREDZNANJE

Potrebno predznanje za praćenje kolegija:

- linearna algebra (§2.1 - §2.11) i teorija vjerojatnosti (§3.1 - §3.11)
- analiza vektorskih funkcija više varijabli (§4.3.1)
- osnovni pojmovi i tehnike strojnog učenja (§5.1 - §5.11)
- osnove programskog jezika Python

Nulta laboratorijska vježba: instrument za provjeru/sticanje predznanja

- možete (trebate!) je odmah riješiti
- ako ne riješite nultu vježbu - prva vježba će vam biti preteška
- gradivo svih laboratorijskih vježbi uči će u međuispit i ispit

## O PREDMETU: LABORATORIJ

Predviđjeli smo četiri laboratorijske vježbe:

0. vektorska algebra, plitki modeli, Python i numpy
1. potpuno povezani unaprijedni modeli
2. konvolucijski modeli
3. povratni modeli
4. metrička ugrađivanja

Vježbe se rade kod kuće, a predaju u terminu vježbi

- računalni zahtjevi: Python, Numpy, Scipy, Matplotlib i PyTorch
- krajem semestra: specijalni termin za nadoknadu jedne vježbe

## O PREDMETU: LABORATORIJ (2)

Laboratorijske vježbe (Python) su središnji dio predmeta:

- izravnih 20% bodova  $4 \times (2.5 + 2.5)$
- barem 20% zadataka na međuispitima i ispitima.

Potrebno je dobiti barem 50% laboratorijskih bodova za izaći na ispit.

Nulta vježba ne donosi bodove i ne provjeravamo je.

## O PREDMETU: UVJETI, BODOVI

**Aktivnosti:** predavanja, vježbe, međuispit, završni ispit, klasični ispit

**Kalendar nastave:**

kraj ožujka: L1

sredina travnja: L2

kraj travnja: MI

početak lipnja: L3

sredina lipnja: L4

kraj lipnja: ZI

početak srpnja: KI

**Kontinuirana provjera:**

laboratorij: 10 (A) + 10 (B)

ispiti: 40, 40

preduvjet: 50% laboratorijskih

**Klasični ispit:**

preduvjet: 50% laboratorijskih

Mogućnost dobivanja **bonus bodova** za: korisne sugestije, prijedloge problemskih zadataka ili vježbi, seminare

- javiti se e-mailom bilo kojem nositelju ili asistentu

**Burza grupa:** tjedan-dva prije odgovarajuće vježbe (pratite oglase!).

**Ocjenjivanje.** 2: 50%, 3: 63%, 4: 76%, 5: 89%.

## O PREDMETU: PREDAVAČI I ASISTENTI

- predavanja:
  - Siniša Šegvić
  - (Marko Čupić)
  - Marin Oršić
- laboratorijske vježbe:
  - Marin Oršić
  - Ivan Sabolić
  - Anja Delić
  - Ivan Martinović

# PREGLED STROJNOG UČENJA: OSNOVNI POJMOVI

Strojno učenje proučava oblikovanje postupaka čija se uspješnost poboljšava s iskustvom

Algoritam strojnog učenja obuhvaća sljedeća dva postupka:

- postupak obrade podataka (**model**), npr. klasificiranje slike
  - nužno zadati mjeru **performanse** (uspješnost) ili mjeru **pogreške**
    - ◊ npr. točnost klasifikacije na **ispitnom skupu**  $\{x_i, y_i\}$
    - ◊ npr. broj krivo klasificiranih ispitnih primjera
  - razlikujemo **empirijsku** i **generalizacijsku** uspješnost
- postupak za **optimiranje** slobodnih parametara postupka obrade na podatcima za učenje npr. gradijentni spust
  - nužno zadati **skup za učenje**  $\{x_i, y_i\}$ 
    - ◊ mora biti uzorkovan iz iste distribucije kao i ispitni skup!
  - nužno zadati kriterij optimizacije (**gubitak**)
    - ◊ npr. negativna log-izglednost parametara na skupu za učenje

# PREGLED STROJNOG UČENJA: DEFINICIJE

Algoritam strojnog učenja definiran je:

- **modelom**: postupak obrade sa slobodnim parametrima
- **gubitkom**: formalizacija (anti-)dobrote parametara modela
- **metodom optimizacije**: način pronalaženja parametara koji minimiziraju gubitak.

S obzirom na kvalitetu podataka za učenje, razlikujemo:

- **nadzirano** učenje: svaki podatak označen željenim izlazom
  - tipični zadatci: **klasifikacija, regresija**
- **nenadzirano** učenje: dostupni su samo podatci
  - tipični zadatci: **estimiranje gustoće, generiranje podataka.**
- **podržano** (ojačano) učenje: dostupna je povratna veza o kvaliteti međudjelovanja s okolinom.

## PREGLED STROJNOG UČENJA: DEFINICIJE (2)

Izglednost parametara modela  $\text{lik}(\theta)$ :

- mjeri koliko dobro model objašnjava opažene podatke
- odgovara vjerojatnosti koju model dodjeljuje skupu za učenje

$$\begin{aligned}\text{lik}(\theta) &= P(\{y_i\} | \{\mathbf{x}_i\}, \theta) \\ &= \prod_{i=1}^N P(y_i | \mathbf{x}_i, \theta)\end{aligned}$$

Često razmatramo **log-izglednost** (prikladnija za diferenciranje):

- jednako rangira modele (logaritam monotono raste),
- pogodna za dosljednu formulaciju gubitka (NLL, MLE)

$$-\log \text{lik}(\theta) = -\sum_{i=1}^N \log P(y_i | \mathbf{x}_i, \theta)$$

# PREGLED STROJNOG UČENJA: PRIMJER

## Logistička regresija: nadzirana klasifikacija u više razreda

- model vraća aposteriornu distribuciju vjerojatnosti razreda:

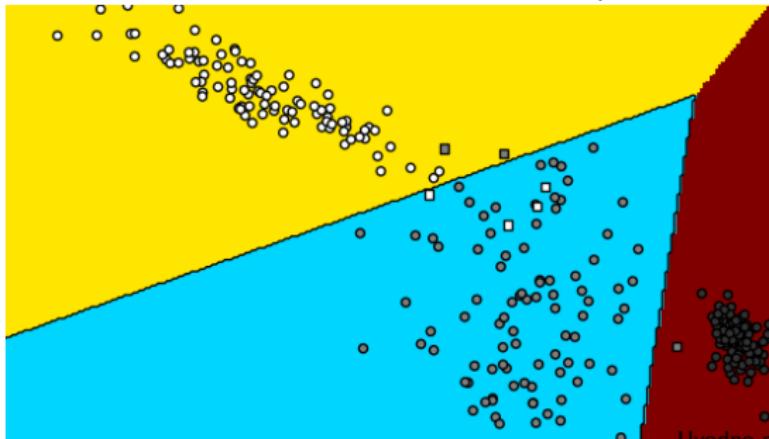
$$P(Y | \mathbf{x}) = \text{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b}), \text{softmax}(\mathbf{s}) = [e^{s_j} / \sum_k e^{s_k}]^\top.$$

- gubitak (negativna log-izglednost modela na skupu za učenje):

$$\mathcal{L}(\mathbf{W}, \mathbf{b} | \mathbf{Y}, \mathbf{X}) = - \sum_i \log P(Y = y_i | \mathbf{x}_i)$$

- optimizacijski postupak (gradijentni spust):

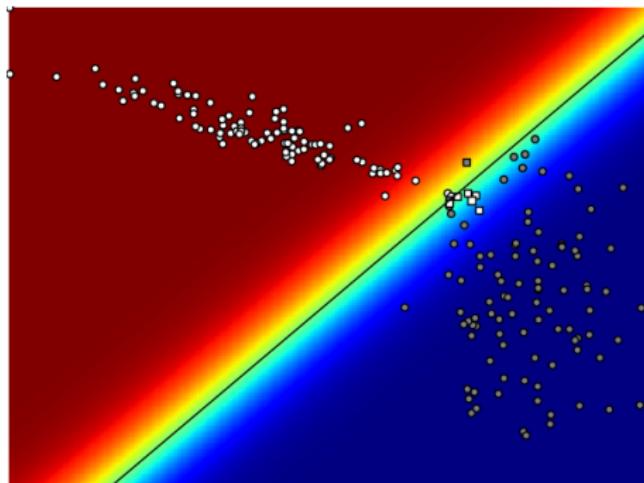
$$\mathbf{b}_{i+1} = \mathbf{b}_i - \delta \cdot \nabla_{\mathbf{b}_i} \mathcal{L} = \mathbf{b}_i - \delta \cdot \left( \frac{\partial \mathcal{L}}{\partial \mathbf{b}_i} \right)^\top$$



## PREGLED STROJNOG UČENJA: PRIMJER 2

Stroj s potpornim vektorima (nadzirana binarna klasifikacija):

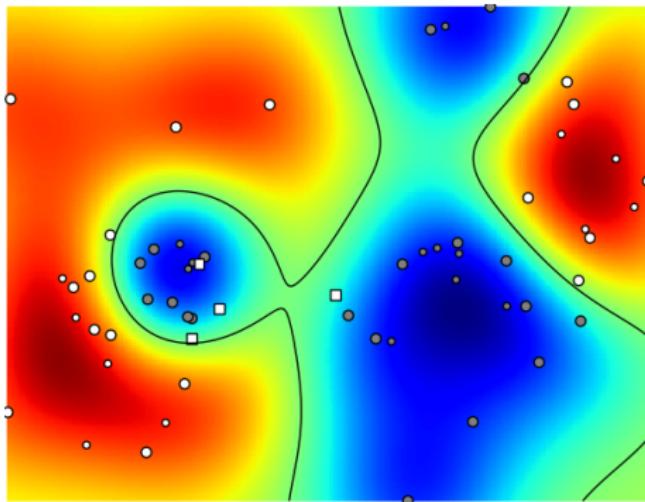
- model (pričinost podatka):  $f(\mathbf{x}) = \begin{cases} c_0 & \text{ako } \mathbf{w}^\top \mathbf{x} + b < 0 \\ c_1 & \text{ako } \mathbf{w}^\top \mathbf{x} + b > 0 \end{cases}$
- gubitak (ukupna povreda margin plus regularizacija):  
$$\mathcal{L}(\mathbf{w}, b \mid \mathbf{Y}, \mathbf{X}) = \lambda \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \max(0, 1 - (-1)^{\llbracket y_i \neq c_1 \rrbracket} (\mathbf{w}^\top \mathbf{x}_i + b))$$
- optimizacija: kvadratno programiranje ili gradijentni spust



# PREGLED STROJNOG UČENJA: PRIMJER 3

Stroj s potpornim vektorima  $\mathbf{x}_j$ , težinama  $\alpha_j$  i jezgrenom funkcijom  $k$ :

- model:  $f(\mathbf{x}) = \begin{cases} c_0 & \text{ako } \sum_j (-1)^{\llbracket y_j \neq c_1 \rrbracket} \cdot \alpha_j \cdot k(\mathbf{x}_j, \mathbf{x}) + b < 0 \\ c_1 & \text{ako } \sum_j (-1)^{\llbracket y_j \neq c_1 \rrbracket} \cdot \alpha_j \cdot k(\mathbf{x}_j, \mathbf{x}) + b > 0 \end{cases}$
- gubitak (regularizacija + prijestup margine):  $\mathcal{L}(\boldsymbol{\alpha}, b) = h(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top) + \frac{1}{n} \sum_{i=1}^n \max \left( 0, 1 - (-1)^{\llbracket y_i \neq c_1 \rrbracket} (\sum_j (-1)^{\llbracket y_j \neq c_1 \rrbracket} \alpha_j \cdot k(\mathbf{x}_j, \mathbf{x}_i) + b) \right)$
- optimizacija: kvadratno programiranje ili gradijentni spust



# PREGLED STROJNOG UČENJA: POSREDNA OPTIMIZACIJA

Specifičnost strojnog učenja: uspješnost optimiramo **posredno**

- optimizacijska metoda **ne "vidi"** gubitak na ispitnom skupu
  - ali pretp. da empirijska distribucija odgovara generativnoj distribuciji
  - tj. podatke za učenje i ispitivanje generira isti slučajni proces
- gubitak se često ne može poistovijetiti s empirijskom pogreškom
  - tipično zato što formulacija pogreške nije derivabilna
  - tada je gubitak **zamjena** (engl. proxy) za empirijsku pogrešku
  - međutim, dobro definiran zamjenski gubitak (ML) može poboljšavati generalizaciju i nakon što empirijska greška padne na nulu!

# PREGLED STROJNOG UČENJA: KAPACITET

Kapacitet je osnovno svojstvo modela:

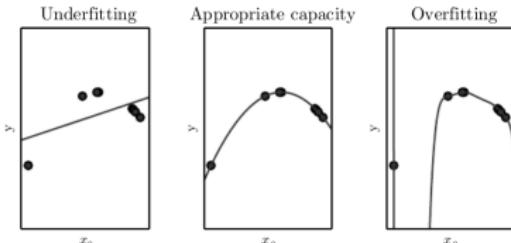
- opisuje sposobnost prilagođavanja podatcima
- najčešće je proporcionalan broju stupnjeva slobode modela
- formalno može se mjeriti brojem podataka (VC dimenzija) koje model može **razbiti** (tj. objasniti uz proizvoljno označavanje)

Modeli malog kapaciteta skloni su **podnaučenosti**:

- ne postoji skup parametara koji može objasniti skup za učenje.

Modeli velikog kapaciteta skloni su **prenaučenosti**:

- previše skupova parametara objašnjavaju skup za učenje.



[goodfellow16]

Uvodno predavanje → Pregled strojnog učenja (7) 27/43

## PREGLED STROJNOG UČENJA: UTJECAJ PODATAKA

Proučit ćemo kako empirijska i generalizacijska uspješnost naučenog modela ovise o veličini skupa za učenje

Razmotrit ćemo seriju eksperimenata vezanih uz skalarnu regresiju  $y = f_{\theta}(x)$  koji su prikazani na sl. 5.4 knjige [goodfellow16book]

Podatci:

- nezavisna varijabla  $x$ : slučajno uzorkovana na konačnom intervalu
- zavisna varijabla  $y$ : polinom 5. stupnja + umjereni šum

Modeli:

- polinom drugog stupnja (uče se tri parametra, potkapacitiran)
- polinom optimalnog kapaciteta (n određen iscrpnom validacijom)

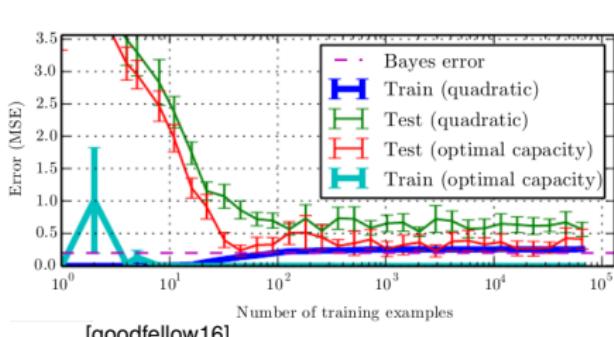
# PREGLED STROJNOG UČENJA: UTJECAJ PODATAKA (2)

Definirajmo **neizbjježnu** (Bayesovu) pogrešku(ljubičasto):

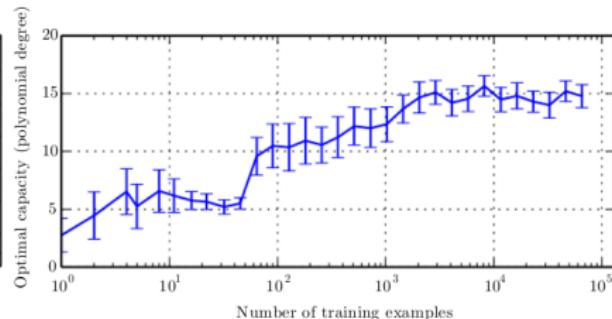
- javlja se zbog stohastičke prirode podataka ili šuma u označavanju

Kako skup za učenje raste, algoritam optimalnog kapaciteta teži:

- zanemarivoj empirijskoj grešci (svijetlo plavo)
- generalizacijskoj grešci (crveno) koja je bolja nego kod potkapacitiranog modela (zeleno) a konvergira **neizbjježnoj** (ljubičasto)



[goodfellow16]



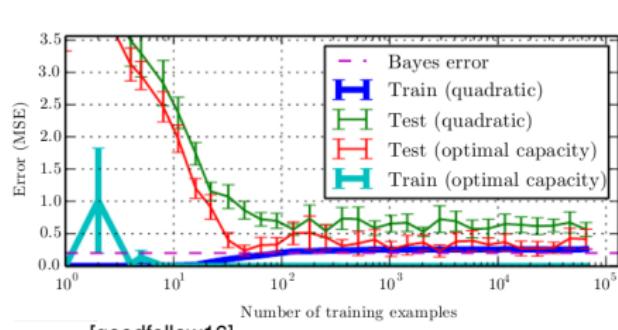
# PREGLED STROJNOG UČENJA: UTJECAJ PODATAKA (3)

Kako skup za učenje raste, potkapacitirani algoritam teži:

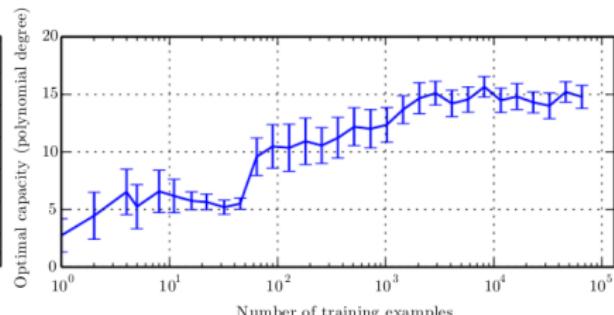
- zamjetnoj empirijskoj grešci (tamno plavo)
- generalizacijskoj grešci (zeleno) koja je veća od optimalne (crveno)

Model s viškom kapaciteta može biti bolji od modela koji zrcali složenost procesa koji je generirao podatke ( $n=5$  + šum, desno)

Kad imamo malo podataka za učenje, algoritam s prilagođenim kapacitetom može biti lošiji od potkapacitiranog algoritma (desno)



[goodfellow16]



# PREGLED STROJNOG UČENJA: LEX PARSIMONAE

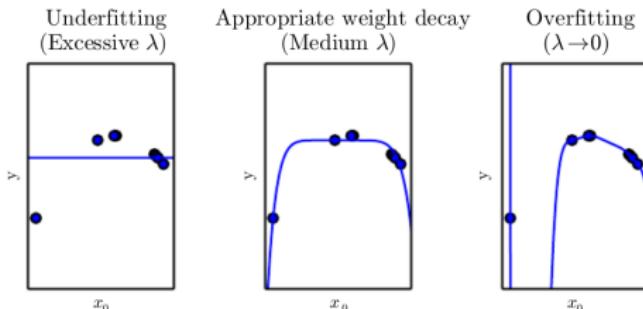
Ograničavanje kapaciteta modela nije jedini način usklađivanja algoritma s pristranošću podataka.

Drugi način je zadržati visoki kapacitet ali uvesti preferenciju prema **jednostavnijim** modelima modificiranjem **gubitka**.

Ako promatramo regresiju, jedan način regulariziranja gubitka bio bi:

$$J(\mathbf{w}) = \lambda \mathbf{w}^\top \mathbf{w} + \sum_i (\sum_j w_j x_i^j + b - y_i)^2.$$

Algoritam sada apriorno preferira rješenja u kojima promjene ulaza vode na blage promjene odluke modela.



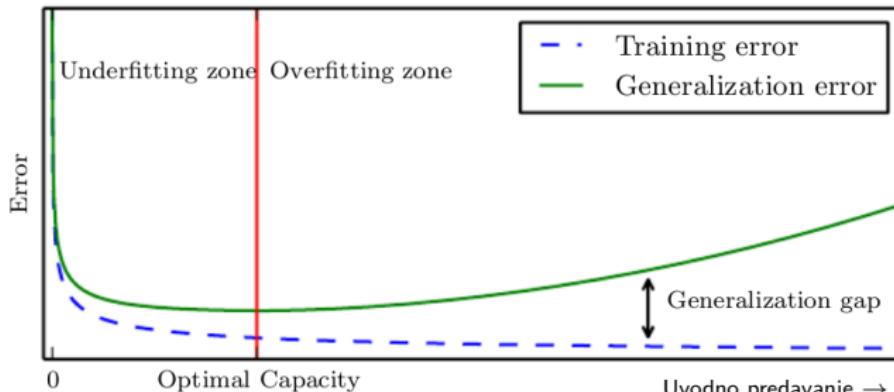
[goodfellow16]

# PREGLED STROJNOG UČENJA: REGULARIZACIJA

Svaku modifikaciju koja namjerava **poboljšati generalizaciju** bez smanjivanja empirijske pogreške nazivamo **regularizacijom** algoritma.

Regularizacija se može primijeniti na sve dijelove algoritma:

- gubitak: kažnjavanje norme vektora parametara
- optimizacijsku metodu: rano zaustavljanje učenja
- podatke: rastresanje, zašumljivanje oznaka
- model: složenost preslikavanja, vezivanje parametara



# PREGLED STROJNOG UČENJA: STATISTIČKI POGLED

Podnaučenost i prenaučenost možemo pojasniti ako generalizacijsku pogrešku rastavimo na doprinose pristranosti i varijance.

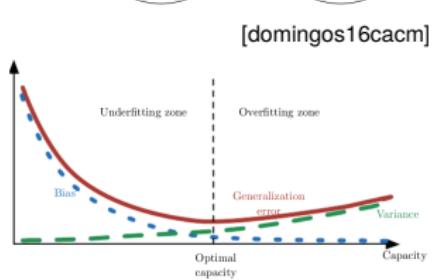
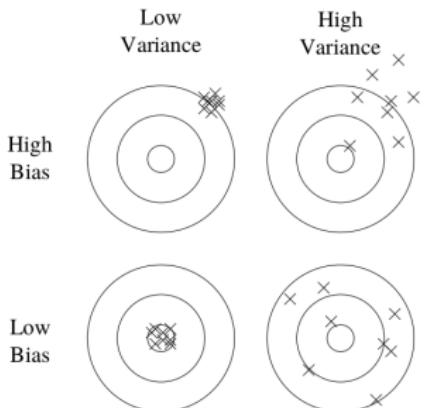
**Pristranost** algoritma: ugrađena sklonost prema nekim rješenjima.

**Varijanca** algoritma: sklonost da rezultat jako varira o ulaznim podatcima.

Regularizacija algoritma smanjuje varijancu i povećava pristranost.

Za najbolje rezultate mjeru regularizacije treba prilagoditi podatcima:

- izbjegći i lošu pristranost i pretjeranu varijancu.



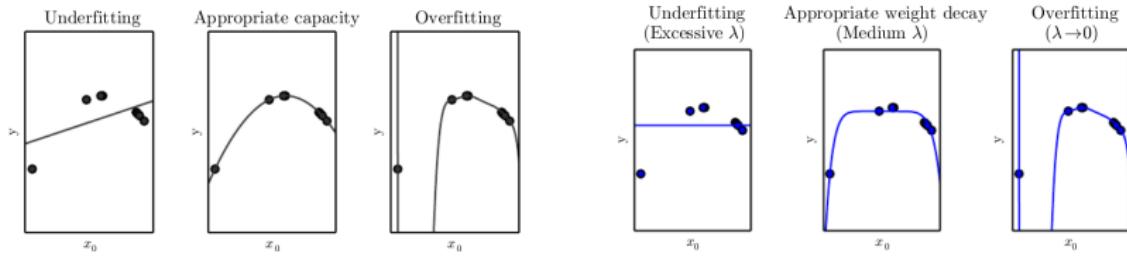
[goodfellow16]

# PREGLED STROJNOG UČENJA: HIPERPARAMETRI

Većina algoritama ima tzv. **hiperparametre** koji reguliraju njihovo ponašanje, a nisu obuhvaćeni optimizacijom

Primjeri: kapacitet modela, faktor kazne parametarske norme, korak optimizacije, broj epoha učenja...

Odabir takvih parametara najčešće provodimo iscrpnim ili slučajnim pretraživanjem prostora parametara na **validacijskom skupu**



[goodfellow16]

## PREGLED STROJNOG UČENJA: GUBITAK

Najintuitivniji gubitak je srednja kvadratna pogreška:

$$J(\theta) = \sum_i (\text{model}(\mathbf{x}_i) - y_i)^2$$

Taj gubitak nije prikladan za probabilističku klasifikaciju jer ignorira da model vraća vjerojatnosnu distribuciju

- npr.  $d_{L2}^2([1,0,0], [0.2, 0.4, 0.4]) < d_{L2}^2([1,0,0], [0.2, 0.0, 0.8])$

Dosljednija formulacija gubitka u probabilističkom okruženju jest negativna log-izglednost parametara modela:

$$J(\theta) = -\frac{1}{N} \sum_i \log P_{\text{model}}(y_i | \mathbf{x}_i, \theta).$$

Može se pokazati da je negativna log-izglednost specijalni slučaj unakrsne entropije (ekvivalent KL divergencije).

Ako modeliramo regresijsko odstupanje Gaussovom razdiobom, negativna log-izglednost se svodi na **kvadratni gubitak**.

## PREGLED STROJNOG UČENJA: SGD

Jedna od trenutno najpopularnijih optimizacijskih metoda u strojnom učenju jest **stohastički gradijentni spust**

Gradijentni spust po negativnoj log-izglednosti mora izračunati:

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_i^N \nabla_{\theta} L(\mathbf{x}_i, y_i, \theta).$$

Kod teških problema optimizacija sporo napreduje jer:  $N \cdot \dim(\theta) \sim 10^{12}$

Problem rješavamo odvajanjem podataka u manje **grupe** (engl. batch):

$$\nabla_{\theta} J(\theta) = \frac{1}{N'} \sum_i^{N'} \nabla_{\theta} L(\mathbf{x}_i, y_i, \theta).$$

- korak optimizacije izvodi se u vremenu  $O(N' \cdot \dim(\theta))$ ,  $N' \ll N$
- grupe se slučajno formiraju nakon svake epohe (stohastika!)
- u velikim modelima ( $\dim(\theta) \sim 10^6$ ): brže od metoda višeg reda
- koristi se i kod velikih plitkih modela!
- SVM s jezgrom: prostor  $\sim O(N^2)$ , vrijeme  $\sim O(N^3)$ .

## PREMA DUBOKOM UČENJU: IZAZOV

Problemi UI-razine razmatraju složene podatke ( $D=\dim(\mathbf{x}_i) \sim 10^5$ )

Zbog toga broj svih mogućih podataka iznosi barem  $O(2^D)$

Ako pristranost klasifikatora ne pogađa pristranost podataka,  
moramo imati predstavnika u svakoj hiper-kocki prostora podataka

- u *pesimističnom* slučaju treba nam  $O(2^D)$  primjera za učenje!
- oblik **prokletstva dimenzionalnosti** (curse of dimensionality)



[goodfellow16]

## PREMA DUBOKOM UČENJU: KLASIČNI ODGOVOR

Klasični klasifikatori pristupaju tom problemu pod pretpostavkom **glatkoće** (ili **lokalne konstantnosti**) modela:

*rezultat modela ne bi se smio "mnogo" mijenjati unutar kompaktne regije prostora podataka.*

Ovakvu pristranost implementiraju k-NN, jezgrene metode, stabla

- svi ti pristupi zahtijevaju  $O(n)$  primjera za učenje za razlikovanje  $O(n)$  regija u prostoru podataka
- takvi pristupi ne mogu funkcionirati kad je  $n = 2^{10^5}$

Možemo izvesti sljedeće zaključke:

- glatkoća je OK, ali ne može se nositi s porastom dimenzionalnosti
- trebaju nam **pristrani** modeli prilagođeni **stvarnim podatcima**.

## PREMA DUBOKOM UČENJU: KOMPOZITNI PODATCI

Temeljna pretpostavka **dubokih modela**: podatci su generirani rekurzivnom **kompozicijom dijelova**

- osoba ima glavu, glava ima lice, lice ima oči, oko ima šarenicu

Potencijal za nezavisno učenje značajki nižih razina:

- osoba s plavim očima i crnom kosom može doprinijeti prepoznavanju osoba s plavim očima i crvenom kosom.

Duboke modele možemo izraziti manjim brojem značajki

- npr. naučiti xor<sub>n</sub> kao zbroj minterma:  $f_n^{\text{exp}}(\mathbf{b}) = \sum_{j=1}^{2^n} w_j \cdot [\![\mathbf{m}_j = \mathbf{b}]\!]$ 
  - O(2<sup>n</sup>) binarnih značajki (i parametara)
- npr. naučiti xor<sub>n</sub> kao kompoziciju *dvoulaznih* logičkih funkcija  $f_i$ :
  - $f_n^{\text{lin}}(\mathbf{b}) = f_1(b_1, f_2(b_2, \dots, f_{n-1}(b_{n-1}, b_n) \dots))$
  - O(4n) binarnih značajki (i parametara)

Ovakav pristup može se suprotstaviti prokletstvu dimenzionalnosti.

# PREMA DUBOKOM UČENJU: KOMPOZITNI PODATCI (2)

**Kompozitna struktura:** važan oblik pristranosti dubokih modela

- podatkovna reprezentacija na razini n gradi se od reprezentacija na razini n-1
- to svojstvo dubokih modela dobro odgovara podatcima u mnogim teškim zadatcima
  - npr. auto ima kotače, kotači imaju naplatke, naplatci imaju rupice
  - slova čine riječi, riječi sintagme, a sintagme rečenice.



[zeiler14eccv]

# PREMA DUBOKOM UČENJU: UČENJE MNOGOSTRUKOSTI

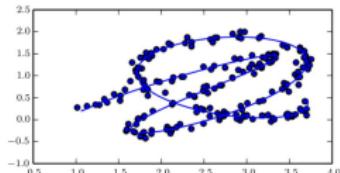
**Mnogostrukost:** povezani skup  $\{x_i\} \in \mathbb{R}^n$  koji se lokalno može aproksimirati skupom  $\{x'_i\} \in \mathbb{R}^m$ ,  $m \ll n$  (lijevo!).

Duboki modeli pretpostavljaju da podatci stanuju na niskodimenzionalnoj mnogostruktosti

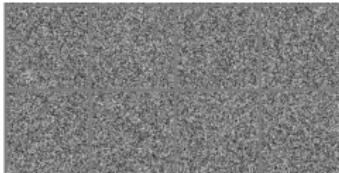
- model se specijalizira na dijelove podatkovnog prostora s velikom gustoćom vjerojatnosti podataka

Koliko je ta pretpostavka zadovoljena u praksi?

- većina svih mogućih ulaznih vektora nisu valjani podatci (sredina!)
- možemo zamisliti skup transformacija koje definiraju obrise mnogostrukosti: svjetlina, kontrast, rotacija (desno!) itd.



[goodfellow16]



## PREMA DUBOKOM UČENJU: ZAKLJUČAK

Duboki modeli su **pristrani**: bolje se mogu prilagoditi podatcima koji se sastoje od dijelova

- ima ih smisla koristiti kad su podatci **kompozitni**
- inače, bolje rezultate mogli bi dati plitki modeli

Duboki modeli su **skalabilni**, mogu raditi s:

- visokodimenzionalnim podatcima ( $D=10^5$ )
- огромним skupovima za učenje ( $N=10^6$ )
- огромним brojem parametara ( $\dim(\theta)=10^9$ )

Zbog toga su duboki modeli **metoda izbora** kod mnogih problema koje danas svrstavamo u UI

## ZAHVALA

Ova predavanja proizšla su iz istraživanja koje je finansirala Hrvatska zaklada za znanost projektom I-2433-2014 MultiCLoD.



<http://multiclod.zemris.fer.hr>