

# Učenje sličnosti

metrička ugrađivanja složenih podataka

Siniša Šegvić  
UniZg-FER

# PLAN

- motivacija za učenje sličnosti
- sijamsko učenje
- trojni gubitak
- detalji izvedbe i vrednovanje
- primjene: stereoskopija, samonadziranje

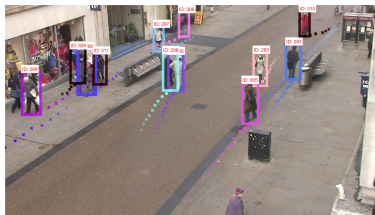
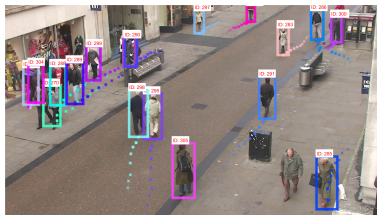
# UVOD : GRANICE KLASIFIKACIJSKE PARADIGME

Kada klasifikacijski modeli mogu postati nepraktični:

- razredi ne postoje ili nisu poznati
- nepraktično veliki broj razreda

Primjeri primjena:

- stereoskopska korespondencija
- samonadzirano učenje
- praćenje, asocijativno pretraživanje, biometrijska verifikacija



[bicanic19fusion]

## UVOD : GRANICE KLASIFIKACIJSKE PARADIGME (2)

Za takve primjene najpraktičnije prediktirati sličnost primjera

Lijevi par: različit. Desni par: sličan



[chopra05cvpr]

Ideja: ugraditi podatke u prikladni vektorski prostor

- standardne metrike (L2, ...) modeliraju sličnost među podacima
- kratki naziv: metrička ugrađivanja

## UVOD : METRIČKA UGRAĐIVANJA

Prije nego što nastavimo dalje, moramo se zapitati:

- zašto ne bismo mjerili sličnost u originalnom prostoru podataka?

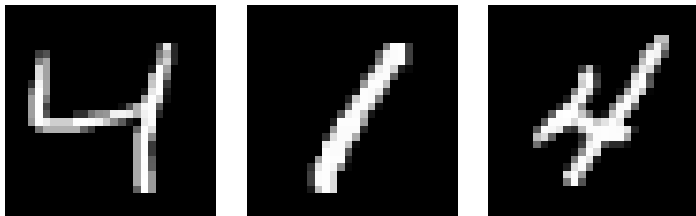
Odgovori:

- zato što udaljenosti u visokodimenzionalnim vektorskim prostorima nemaju smisla (prokletstvo dimenzionalnosti)
- zato što vektorske reprezentacije složenih podataka tipično nisu pogodne za njihovu usporedbu

## UVOD : METRIČKA UGRAĐIVANJA - PRIMJER

Promotrimo udaljenosti između znamenki skupa MNIST

- L2 udaljenost između prve i druge znamenke: 121.4
- L2 udaljenost između prve i treće znamenke: 133.2
- L2 udaljenost između druge i treće znamenke: 114.9



[lecun98piece]

Zaključak: ne postoji jaka veza između udaljenosti u originalnom prostoru i sličnosti podataka!

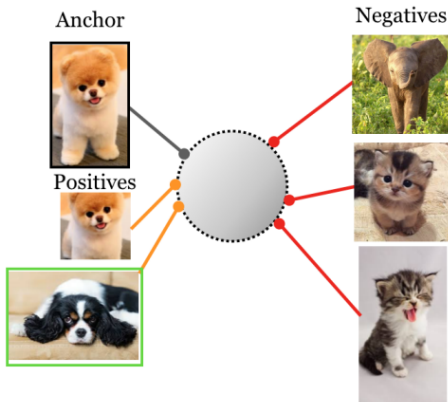
## UVOD : METRIČKA UGRAĐIVANJA - PRIMJER (2)

```
import torch
import torchvision
mnist = torchvision.datasets.MNIST('data', download=True)
print(mnist.targets[:10])
# tensor([5, 0, 4, 1, 9, 2, 1, 3, 1, 4])
print(torch.sqrt(torch.sum((mnist.data[2]-mnist.data[3])**2,
                           dtype=torch.float))))
# tensor(121.4)
print(torch.sqrt(torch.sum((mnist.data[2]-mnist.data[9])**2,
                           dtype=torch.float))))
# tensor(133.2)
print(torch.sqrt(torch.sum((mnist.data[3]-mnist.data[9])**2,
                           dtype=torch.float))))
# tensor(114.9)
import cv2
cv2.imwrite('m2.png', mnist.data[2].numpy())
cv2.imwrite('m3.png', mnist.data[3].numpy())
cv2.imwrite('m9.png', mnist.data[9].numpy())
```

## UVOD : METRIČKA UGRAĐIVANJA - CILJ

Ugraditi podatke u (relativno) niskodimenzionalni prostor gdje će standardna metrika modelirati sličnost među podacima

Ako reprezentacije normiramo (tj. smjestimo ih na hipersferu), udaljenost možemo mjeriti skalarnim produktom





## UVOD : METRIČKA UGRAĐIVANJA - PREDNOSTI

Podatke možemo asociirati iako u trenutku učenja nismo vidjeli sve razrede

Modelu je teže prenaučiti se:

- klasifikacija:  $O(N)$  podataka za učenje
- sličnost:  $O(N^2)$ ,  $O(N^3)$  ili  $O(N^B)$  podataka za učenje

Vrlo korisne reprezentacije mogu se naučiti i u samonadziranom kontekstu gdje zahtijevamo da podatak bude sličan perturbiranom sebi a različit od ostalih podataka; npr. SimCLR [chen20icml].

Metrička ugrađivanja mogu ponekad pomoći i u klasičnim nadziranim zadacima [khosla20neurips].

## METRIKE : POJMOVI

Razmatramo skup  $X$  te preslikavanje  $d : X \times X \rightarrow R$ .

Kažemo da je  $d$  **metrika**, a  $(X, d)$  - metrički prostor akko:

1.  $d(a, b) \geq 0 \quad \forall a, b \in X$  (pozitivnost),
2.  $d(a, b) = 0 \iff a = b \quad \forall a, b \in X$  (strogost),
3.  $d(a, b) = d(b, a) \quad \forall a, b \in X$  (simetričnost),
4.  $d(a, b) \leq d(a, c) + d(c, b) \quad \forall a, b, c \in X$  (nejednakost trokuta).

Ova definicija dobro se uklapa u koncept sličnosti podataka.

Lako je pokazati da su aksiomi metrike redundantni npr. pozitivnost i simetričnost slijede iz strogosti i nejednakosti trokuta.

U praksi najčešće učimo **pseudo-metrik** koja relaksira strogost

- teško osigurati  $d(a, b) \neq 0 \quad \forall a \neq b$ , zahtijevamo samo  $d(a, a) = 0$

## METRIKE : STANDARDNI IZBORI

Euklidska metrika:

$$d_E(a, b) = \sqrt{(a - b)^\top (a - b)} \sim (a - b)^\top (a - b)$$

Ako su podatci normirani, skalarni produkt odgovara **kosinusnoj sličnosti** te inducira isto rangiranje kao i Euklidska metrika:

$$\begin{aligned}d_E(a, b) &\sim (a - b)^\top (a - b) \\ &\sim a^\top a - 2 \cdot a^\top b + b^\top b = 2 - 2 \cdot a^\top b \\ &\sim -a^\top b\end{aligned}$$

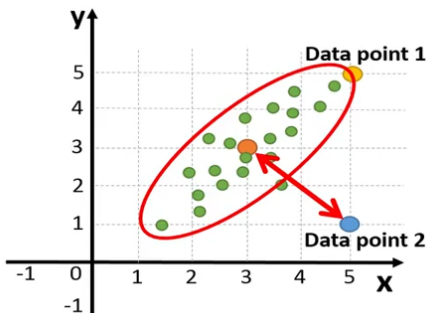
Mahalanobisova metrika (M odgovara inverznoj kovarijanci podataka):

$$d_M(a, b) \sim (a - b)^\top \cdot M \cdot (a - b)$$

# METRIKE : MAHALANOBIS

Mahalanobisova izohipsa označena je crvenom bojom

- Po Mahalanobisu, žuti podatak puno je bliži narančastom od plavog podatka
- možemo hipotetizirati da plavi podatak ne pripada zelenima
- Euklidska metrika ne podržava takvo zaključivanje!



## METRIKE : MAHALANOBIS (2)

Matrica  $M$  je realna i simetrična  $\Rightarrow$  može se dijagonalizirati:  $M = W^T W$

Korištenje Mahalanobisove metrike možemo interpretirati kao plitko ugrađivanje u Euklidski metrički prostor  $W$ :

$$\begin{aligned}d_M(a, b) &\sim (a - b)^T \cdot M \cdot (a - b) \\ &\sim (W \cdot (a - b))^T \cdot (W \cdot (a - b)) \\ &\sim (W \cdot a - W \cdot b)^T \cdot (W \cdot a - W \cdot b) \\ &\sim d_E(Wa, Wb)\end{aligned}$$

Postoje analogni plitki pristupi koji uzimaju u obzir informaciju o pripadnosti podataka simboličkim razredima (Fisher LDA)

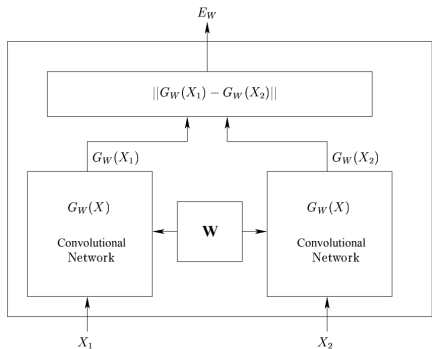
Logični korak dalje: zamijeniti  $W$  **dubokim modelom**  $f_\theta$

- ono što je logično danas, nije bilo logično 2005...

# SIJAMSKO UČENJE : IDEJA

Naučiti model  $G_W$  koji ugrađuje podatke  $X$  u prostor gdje euklidska metrika  $E_W$  odražava sličnost među podacima

Sijamsko učenje: parovi podataka prolaze kroz dva primjerka modela.



[chopra05cvpr]

Primjerci dijele parametre  $W$ , a gradijenti odgovarajućih parametara dviju grana se akumuliraju.

## SIJAMSKO UČENJE : GUBITAK

Sijamsko učenje koristi neku od varijanti **kontrastnog gubitka**

Kontrastni gubitak ovisi o tome jesmo li na ulaze sijamskih modela doveli primjerke istog razreda

$$L(\theta) = \sum_{y_q=y_p} L_{\text{pos}}(\theta|x_q, x_p) + \sum_{y_q \neq y_n} L_{\text{neg}}(\theta|x_q, x_n)$$

Gubitak  $L_{\text{pos}}$  tjera primjerke istih razreda da se približe:

$$L_{\text{pos}}(\theta|x_q, x_p) = \|f_{\theta}(x_q) - f_{\theta}(x_p)\|^2$$

Gubitak  $L_{\text{neg}}$  tjera primjerke različitih razreda da se udalje:

$$L_{\text{neg}}(\theta|x_q, x_n) = [\max(0, m - \|f_{\theta}(x_q) - f_{\theta}(x_n)\|)]^2$$

## SIJAMSKO UČENJE : GRADIJENTI

Pogledajmo gradijente gubitaka  $L_{\text{pos}}$  i  $L_{\text{neg}}$  s obzirom na metrička ugrađivanja  $z_q = f_{\theta}(x_q)$ :

$$\frac{\partial L_{\text{pos}}}{\partial z_p} = 2 \cdot (z_p - z_q)$$

$$\frac{\partial L_{\text{neg}}}{\partial z_n} = -2 \cdot \max(0, m - \|z_q - z_n\|) \cdot \frac{z_n - z_q}{\|z_n - z_q\|}$$

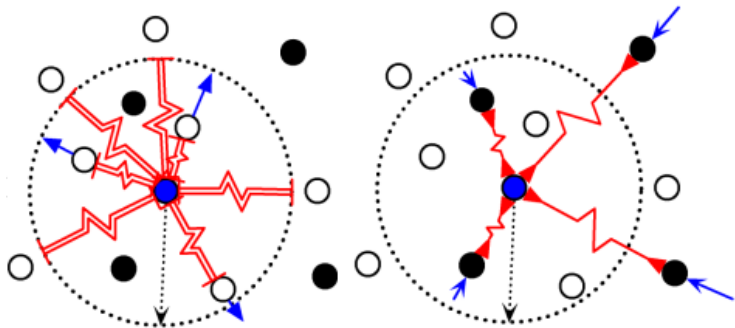
Ovi gradijenti potiču približavanje ugrađivanja pozitivnih parova te udaljšavanje ugrađivanja negativnih parova sve dok oni ne postanu udaljšeniji od  $m$ .



## SIJAMSKO UČENJE : GRADIJENTI (2)

Prikazanu dinamiku možemo ilustrirati sustavom mehaničkih opruga (sila opruge proporcionalna je udaljenosti)

- crni i bijeli krugovi predstavljaju pozitivne i negativne primjere s obzirom na plavi podatak
- negativni izvan radijusa  $m$  ne osjećaju odbijanje plavog podatka



[hadsell06cvpr]

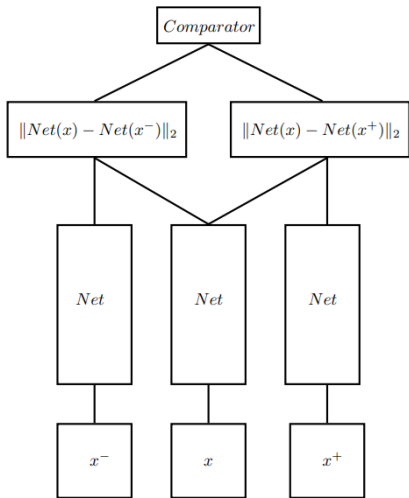
# TROJNO UČENJE : IDEJA

U sijamskom učenju jedan te isti podatak uspoređujemo i s negativnim i s pozitivnim primjerima.

Pri tome ugrađivanje promatranog primjera trebamo izračunati dva puta

Taj problem adresira trojno učenje:

- referentno ugrađivanje  
uspoređujemo s pozitivnim i negativnim primjerom



[hoffer15iclrw]

Tri primjerka modela dijele parametre i ravnopravno doprinose gradijentima gubitka

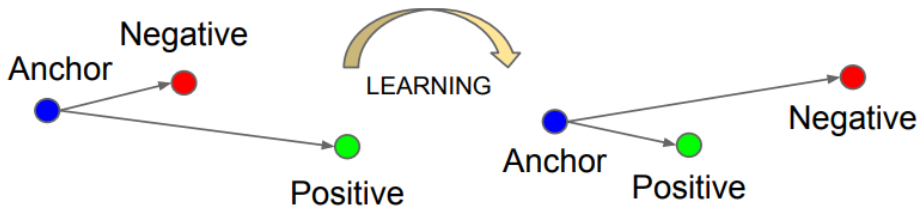
## TROJNO UČENJE : GUBITAK

Trojno učenje tipično koristi **trojni gubitak**

Trojni gubitak spaja obje komponente kontrastnog gubitka u jedan izraz:

$$L(\theta) = \sum_i \max(0, \|f_\theta(x_{ia}) - f_\theta(x_{ip})\| - \|f_\theta(x_{ia}) - f_\theta(x_{in})\| + \alpha)$$

Trojni gubitak privlači referentni i pozitivan podatak te odbija referentni i negativan podatak:



## TROJNO UČENJE : GRADIJENTI

Pogledajmo gradijente trojnog gubitka s obzirom na ugrađivanja

$$f_a = f_\theta(x_{ia}), f_p = f_\theta(x_{ip}) \text{ i } f_n = f_\theta(x_{in}).$$

Ti gradijenti potiču približavanje pozitivna i udaljšavanje negativna, ali samo ako je negativ bliže od udaljšnosti pozitivna uvećane za marginu  $\alpha$ :

$$\frac{\partial L}{\partial f_p} = [\|f_a - f_p\| + \alpha - \|f_a - f_n\|] \cdot \frac{f_p - f_a}{\|f_p - f_a\|}$$
$$\frac{\partial L}{\partial f_n} = [\|f_a - f_p\| + \alpha - \|f_a - f_n\|] \cdot \frac{f_a - f_n}{\|f_a - f_n\|}$$

## DETALJI : MEKA ZGLOBNICA

Klasični trojni gubitak zanemaruje trojke kod kojih je negativ dalji od pozitiva za više od margine.

Prednost tog pristupa jest onemogućavanje prenaučivosti: kada se podaci dovoljno dobro rasporede --- učenje prestaje

Međutim, s druge strane, takav ziheraški pristup može dovesti do lošije generalizacije

Stoga ponekad možemo poboljšati generalizaciju na način da čvrstu zglobnicu  $\text{ReLU}(x) = [x]_+$  zamijenimo njenom mekom varijantom:

$$\text{softplus}(x) = \ln(1 + e^x)$$

## DETALJI: VERZIJA SA SKALARNIM PRODUKTOM

Dogovorimo skraćenu notaciju ugrađivanja s obzirom na  $x_p$ ,  $x_p$ ,  $x_n$ :

$$f_a = f_\theta(x_{ia})$$

$$f_p = f_\theta(x_{ip})$$

$$f_n = f_\theta(x_{in})$$

Ako su latentne reprezentacije normirane,  $\|f_a\| = \|f_p\| = \|f_n\| = 1$ , gubitak možemo izraziti i kroz kosinusnu sličnost:

$$L(\theta) = \sum_i \max(0, f_\theta(x_{ia})^\top f_\theta(x_{in}) - f_\theta(x_{ia})^\top f_\theta(x_{ip}) + \alpha)$$

## DETALJI : FORMIRANJE TROJKI

Pojednostavnjeni izraz za osnovni trojni gubitak [hermans17arxiv]:

$$L_3(\theta) = \sum_{y_a=y_p \neq y_n} [\alpha + D_{ap} - D_{an}]_+$$

Formiranje trojki za učenje vrlo je važan izvedbeni detalj

- glavni problem je u učinkovitosti učenja
- broj svih trojki raste s  $O(N \cdot N_p \cdot N_n)$

Ponekad se isplati isprobati čvrsti trojni gubitak (eng. batch-hard [hermans17arxiv]) koji referentni podatak  $a$  povezuje s najtežim pozitivom i najtežim negativom:

$$L_{BH}(\theta) = \sum_a [\alpha + \max_{y_a=y_p} D_{ap} - \min_{y_a \neq y_p} D_{an}]_+$$

## DETALJI : N PAROVA

Pretpostavimo sada da imamo [sohn16neurips]:

- n parova koji dijele sidro ( $\mathbf{x}_a$ )
- samo jedan pozitivan podatak ( $\mathbf{x}_p$ ) te n-1 negativnih ( $\mathbf{x}_{ni}$ )

Gubitak definiramo tako da raste kad je sidro slično negativima a pada kad je sidro slično pozitivima:

$$\mathcal{L}_{N\text{-pairs}}(\mathbf{x}_a, \mathbf{x}_p, \{\mathbf{x}_{ni}\}) = \log \left( 1 + \sum_{i=1}^{N-1} e^{f_{\theta}(\mathbf{x}_a)^{\top} f_{\theta}(\mathbf{x}_{ni}) - f_{\theta}(\mathbf{x}_a)^{\top} f_{\theta}(\mathbf{x}_p)} \right)$$

Za n=2  $\mathcal{L}_{N\text{-pairs}}$  vrlo je sličan trojnom gubitku sa skalarnim produktom.



## DETALJI: N PAROVA (2)

Nakon nekoliko jednostavnih koraka  $\mathcal{L}_{N\text{-pairs}}$  svodimo na sljedeći oblik:

$$\begin{aligned}\mathcal{L}_{N\text{-pairs}}(\mathbf{x}_a, \mathbf{x}_p, \{\mathbf{x}_{ni}\}) &= \\ &= -\log \frac{\exp(f_\theta(\mathbf{x}_a)^\top f_\theta(\mathbf{x}_p))}{\exp(f_\theta(\mathbf{x}_a)^\top f_\theta(\mathbf{x}_p)) + \sum_{i=1}^{N-1} \exp(f_\theta(\mathbf{x}_a)^\top f_\theta(\mathbf{x}_{ni}))} \\ &= -\log \frac{\exp(f_\theta(\mathbf{x}_a)^\top f_\theta(\mathbf{x}_p))}{\sum_{i=1}^N \exp(f_\theta(\mathbf{x}_a)^\top f_\theta(\mathbf{x}_i))}\end{aligned}$$

Vidimo da je  $\mathcal{L}_{N\text{-pairs}}$  ekvivalentan standardnoj unakrsnoj entropiji nad softmaxom vektora sličnosti parova podataka.

## DETALJI: N PAROVA (3)

Gubitak n parova može se poopćiti i na slučaj kad imamo više pozitivnih i više negativnih primjera u grupi  $\{\mathbf{x}_i, y_i\}_{i=1}^B$  (eng. soft nearest neighbours) [frosst19icml]:

$$\mathcal{L}_{\text{snn}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{1}{|\{y_j = y_i, j \neq i\}|} \frac{\sum_{j \neq i, y_j = y_i} e^{f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_j) / \tau}}{\sum_{y_i \neq y_k} e^{f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_k) / \tau}}$$

- hiper-parametar  $\tau$  (temperatura) modulira entropiju izlaza.

Stroža varijanta tog gubitka zahtijeva da **svaki** pozitiv iz grupe bude sličniji od negativa [khosla20neurips]:

$$\mathcal{L}_{\text{sum-out}} = -\frac{1}{B} \sum_{i=1}^B \frac{1}{|\{y_j = y_i, j \neq i\}|} \sum_{j \neq i, y_j = y_i} \log \frac{e^{f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_j) / \tau}}{\sum_{y_i \neq y_k} e^{f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_k) / \tau}}$$

- ova varijanta generalizira bolje iako  $\mathcal{L}_{\text{snn}} > \mathcal{L}_{\text{sum-out}}$  (Jensen)

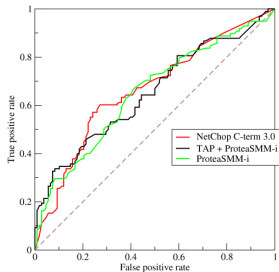
## DETALJI : VREDNOVANJE

Naučene mjere sličnosti induciraju rangiranje:

- fiksiramo podatak  $x_a$ , sortiramo sve druge podatke prema padajućoj sličnosti:  $s_{ai} = -d(f_\theta(x_a), f_\theta(x_i))$
- ako rangiranje savršeno generalizira (to obično nije slučaj) onda su svi pozitivni sortirani prije negativa

Kvalitetu rangiranja mjerimo površinom ispod krivulja PR ili ROC

- veća površina  $\Rightarrow$  bolji model



## DETALJI: VREDNOVANJE (2)

Evo kako formirati krivulje:

- postavljamo prag na svaki indeks  $i$ : podatci koji su sličniji od  $s_{ai}$  su pozitivni (ima ih  $P_i$ ), a ostali su negativni (ima ih  $N_i$ );
- uz pomoć oznaka dobivamo brojnosti
  - točnih pozitivna -  $TP_i$
  - lažnih pozitivna -  $FP_i$
  - lažnih negativna -  $FN_i$
  - i točnih negativna -  $TN_i$
- sada možemo (i dalje za taj isti prag  $i$ ) odrediti relevantne metrike:
  - odziv  $R_i = TPR_i = TP_i / (TP_i + FN_i)$
  - preciznost  $P_i = TP_i / (TP_i + FP_i)$
  - udio lažnih pozitivna =  $FPR_i = FP_i / (TN_i + FP_i)$
- krivulju preciznosti i odziva (PR) čine točke  $(R_i, P_i)$
- krivulju ROC čine točke  $(FPR_i, TPR_i)$

## DETALJI : ZADATAK

Zadani su podatci  $x_1$  do  $x_5$ .

Poznato je da su identiteti podataka redom  $Y=[1, 0, 0, 1, 1]$

Poznato je da udaljenosti od podatka  $x_1$  iznose:

$$d(x_1, X) = [0.0, 5.0, 2.0, 3.0, 1.0]$$

Odredite površinu ispod P-R krivulje (skraćeno AUPR) za predikcije modela u podatku  $x_1$

Napomena: AUPR često nazivamo i prosječnom preciznošću (eng. average precision, AP)

## DETALJI : ZADATAK - RJEŠENJE

Rangiranje podataka je:  $[x_5^{(1)}, x_3^{(0)}, x_4^{(1)}, x_2^{(0)}]$

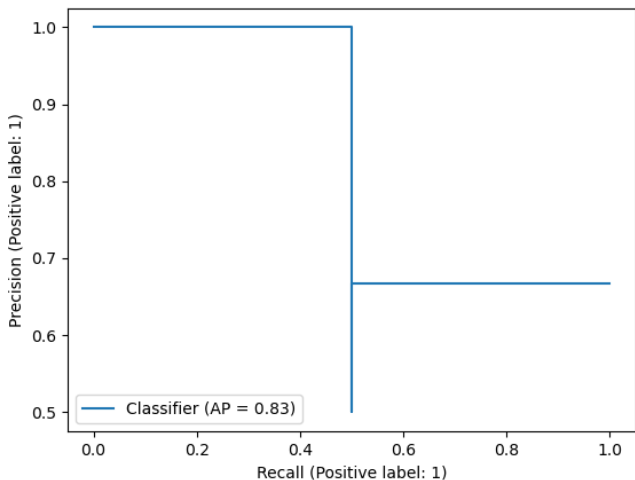
Postavljamo prag na svaki podatak i mjerimo preciznost  $P=TP/(TP+FP)$   
i odziv  $R=TP/(TP+FN)$ :

pozitivne predikcije	TP	FP	FN	P	R
$x_5, x_3, x_4, x_2$	2	2	0	0.5	1.0
$x_5, x_3, x_4$	2	1	0	<b>0.7</b>	<b>1.0</b>
$x_5, x_3$	1	1	1	0.5	0.5
$x_5$	1	0	1	<b>1.0</b>	<b>0.5</b>

Za niti jedan prag nemamo  $R=0$ . Zato dogovorno dodajemo točku ( $R=0$ ,  $P$ =preciznost za najmanji  $R$ ).

Ako za isti  $R$  imamo više  $P$ -ova — smijemo izabrati bolji.

## DETALJI : ZADATAK - GRAF



Rješenje:  $AUPR = (0.5-0) \times 1 + (1-0.5) \times 0.66 = 0.83$

## DETALJI: ZADATAK - KOD

```
import numpy as np
from sklearn.metrics import average_precision_score
from sklearn.metrics import PrecisionRecallDisplay
import matplotlib.pyplot as plt

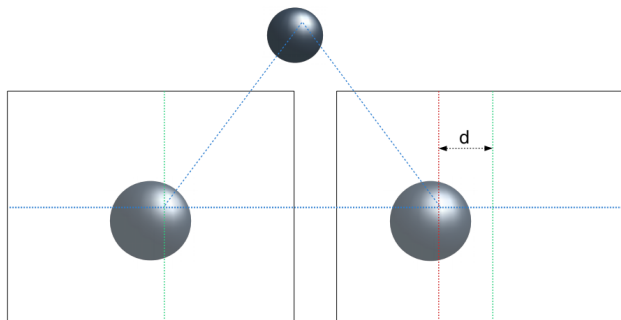
y_true = np.array([0, 0, 1, 1])
y_scores = np.array([-5, -2, -3, -1])
print(average_precision_score(y_true, y_scores))

PrecisionRecallDisplay.from_predictions(y_true, y_scores)
plt.show()
```



## STEREO : ZADATAK

Za svaki piksel lijeve slike tražimo korespondentni piksel u desnoj slici:



[orsic17ms]

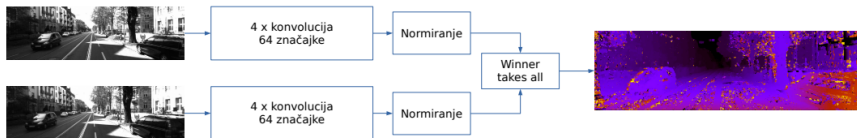
Pretpostavljamo kalibrirani slučaj: korespondencija je u istom retku

- tražimo gusto polje horizontalnih pomaka (dispariteta)
- ako znamo disparitet, širinu vidnog polja kamere i udaljenost među kamerama - možemo odrediti dubinu tog dijela scene u metrima

## STEREO : IDEJA

Ugraditi piksele obje slike u metrički prostor konvolucijskim modelom

$$f_{\theta} : \mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^{F \times H \times W}, F=64 \text{ [zbontar15cvpr].}$$



[orsic17ms]

Formirati gusti volumen cijene  $V$  oblika  $D \times H \times W$ :

$$\square V_{ijd} = \text{cost}(f_{\theta}(I_L)_{i,j}, f_{\theta}(I_R)_{i,j+d})$$

U svakom pikselu odrediti najbolji disparitet (winner takes all):

$$\square D_{ij} = \arg \min_d V_{ijd}$$

## STEREO : DETALJI

Trojni gubitak izražavamo skalarnim produktom nad normiranim metričkim ugrađivanjima isječaka  $9 \times 9$ :

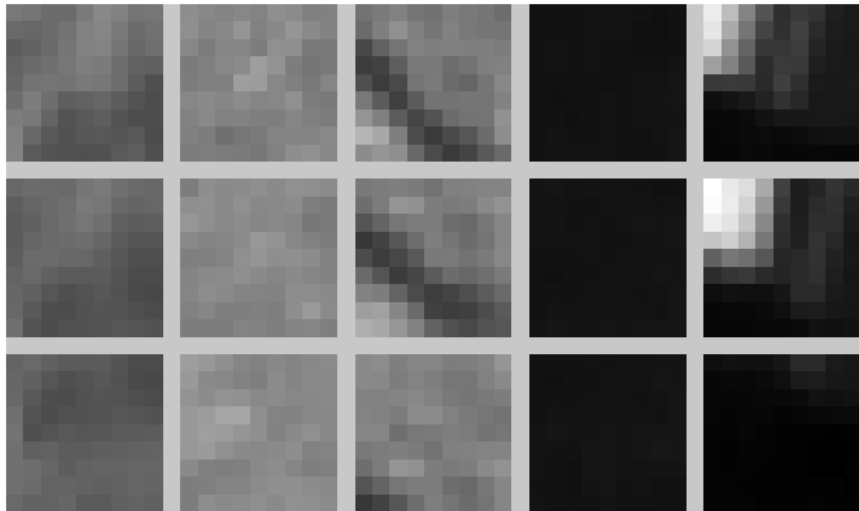
- tri primjerka modela  $f_\theta$  dijele parametre
- svaki primjerak pamti aktivacije i računa gradijente
- ukupni gradijent za svaki parametar dobivamo agregacijom doprinosa primjeraka modela

Model  $f_\theta$  je ekvivarijantan:  $4 \times \text{conv}3 \times 3$  bez sažimanja:

- ulazni podatci za učenje imaju dimenzije  $128 \times 3 \times 9 \times 9$
- pri učenju ne koristimo nadopunjavanje (pri zaključivanju - da)
- ugrađivanja imaju 64 dimenzije,  $f_\theta : \mathbb{R}^{3 \times 9 \times 9} \rightarrow \mathbb{R}^{64}$
- pikseli slika se normiraju na  $N(0, \mathbf{I})$  pri učenju i zaključivanju
- zaključivanje primjenjujemo na cjelokupne slike  $2 \times 3 \times H \times W$

## STEREO : TROJKE

Stupci prikazuju trojke za učenje:

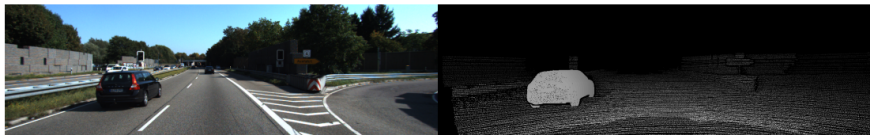


[orsic17ms]

## STEREO : EKSPERIMENTI

Skup podataka KITTI [geiger13ijrr]:

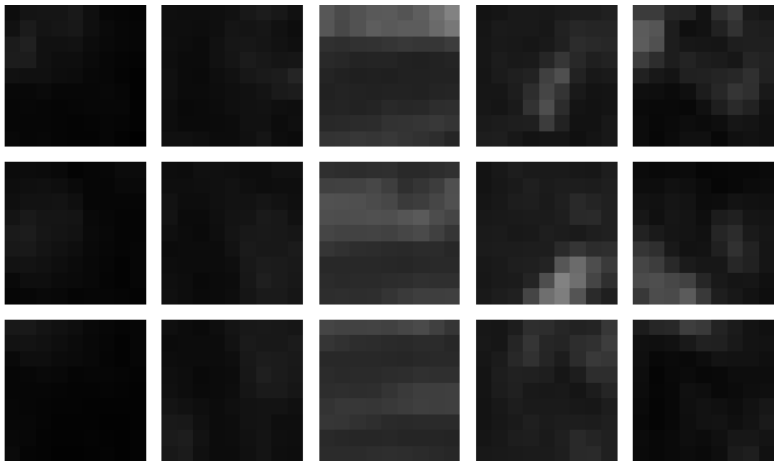
- 200 rektificiranih slika 1382 x 512
- skup za učenje: 80% slika (ostatak - skup za validaciju)
- točni dispariteti izmjereni LIDAR-om u 30% piksela
- gusti dispariteti na automobilima dobiveni fitanjem CAD modela



[orsic17ms]

## STEREO : GREŠKE

Mnogi pikseli nemaju korespondenciju zbog "stereoskopske sjene":



[orsic17ms]

Retci prikazuju i) upit, ii) korespondenciju, iii) najsljedniji negativ.

## STEREO : TOČNOST

Stereoskopske metode na KITTI-ju uspoređujemo prema postotku točnih dispariteta s tolerancijom  $\pm 3$  piksela.

Eksperimentalna točnost je solidna, iako lošija od stanja tehnike:

Model	Točnost - treniranje	Točnost - testiranje
Sive ulazne slike	84.99%	82.32%
Sive ulazne slike - BN	74.51%	71.61%
Ulazne slike u boji	<b>85.50%</b>	<b>82.84%</b>
Ulazne slike u boji - BN	74.40%	71.33%

[orsic17ms]

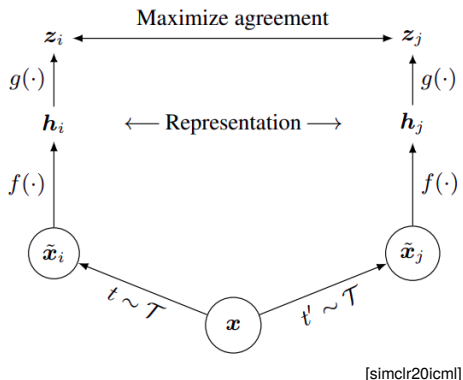
Velika prednost ovakvih pristupa: otpornost na prenaučenosť.

Sastavna komponenta novijih pristupa koji rješavaju preostale izazove:

- učenje na neoznačenom videu [liu20cvpr]
- popunjavanje područja bez korespondencija analizom konteksta

## SAMONADZIRANJE : ZADATAK

Naučiti korisne reprezentacije bez korištenja semantičkih oznaka.



- $f$  - konvolucijska okosnica
  - npr. ResNet-50
- $g$  - projekcijski modul
  - npr.  $2 \times \text{FC}(128)$
- $t, t' \in \mathcal{T}$  - slučajne perturbacije
  - npr. izrezivanje, rastresanje boje, zaglađivanje

Prednosti: možemo učiti bez oznaka, bolji prijenos učenja!

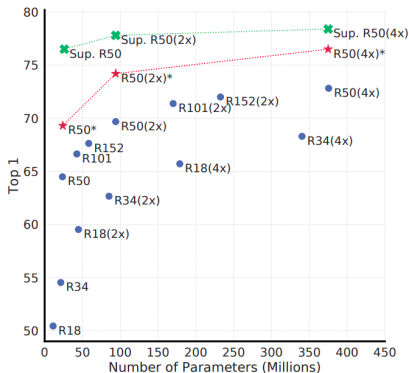
Nedostatci: dugotrajno učenje, velike grupe, trivijalna rješenja?



# SAMONADZIRANJE : SIMCLR

SimCLR: Simple Contrastive Learning of visual Representations

- gubitak n parova: za svaki pozitivni par imamo  $2(B-1)$  negativa
- zaključivanje: odbaciti  $g$ , koristiti  $f$  za prijenos učenja
- evaluacija: nad  $f$  naučiti višerazrednu logističku regresiju
- ostale primjene: predtreniranje + ugađanje, polunadzirano učenje



- veći modeli bolje uče
- SimCLRv2 + linear gotovo jednako dobar kao i nadzirano učenje
- za najbolje rezultate trebamo više epoha učenja ( $10\times$ ) i velike grupe (4096)

# SAMONADZIRANJE: SIMCLR (2)

---

**Algorithm 1** SimCLR's main learning algorithm.

---

**input:** batch size  $N$ , constant  $\tau$ , structure of  $f, g, \mathcal{T}$ .  
**for** sampled minibatch  $\{\mathbf{x}_k\}_{k=1}^N$  **do**  
  **for all**  $k \in \{1, \dots, N\}$  **do**  
    draw two augmentation functions  $t \sim \mathcal{T}, t' \sim \mathcal{T}$   
    # the first augmentation  
     $\tilde{\mathbf{x}}_{2k-1} = t(\mathbf{x}_k)$   
     $\mathbf{h}_{2k-1} = f(\tilde{\mathbf{x}}_{2k-1})$  # representation  
     $\mathbf{z}_{2k-1} = g(\mathbf{h}_{2k-1})$  # projection  
    # the second augmentation  
     $\tilde{\mathbf{x}}_{2k} = t'(\mathbf{x}_k)$   
     $\mathbf{h}_{2k} = f(\tilde{\mathbf{x}}_{2k})$  # representation  
     $\mathbf{z}_{2k} = g(\mathbf{h}_{2k})$  # projection  
  **end for**  
  **for all**  $i \in \{1, \dots, 2N\}$  and  $j \in \{1, \dots, 2N\}$  **do**  
     $s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_j\|)$  # pairwise similarity  
  **end for**  
  **define**  $\ell(i, j)$  as  $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$   
   $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$   
  update networks  $f$  and  $g$  to minimize  $\mathcal{L}$   
**end for**  
**return** encoder network  $f(\cdot)$ , and throw away  $g(\cdot)$

---

[simclr20icml]

Gubitak ima 2B članova:

- svaki član je  $\ell_{NP}(i, j)$ 
  - $i$ : sidro
  - $j$ : pozitiv
  - $k \neq i$ : negativni
- $\ell_{NP}(i, j)$  je izražen s obzirom na kosinusne sličnosti  $s_{ij}$  i  $s_{ik}$
- negativni su svi preostali podatci minigrupe
  - $k \in [1..2N], k \neq i$

# ZAKLJUČAK

Kvantificiranje sličnosti jedan od temeljnih zadataka strojnog učenja

Metrička ugrađivanja primjenjujemo kad klasifikacija nije praktična

- broj razreda prevelik ili unaprijed nepoznat
- malo označenih podataka za učenje

Danas metrička ugrađivanja tipično ostvarujemo dubokim modelima koje učimo u sijamskim konfiguracijama.

Primjene:

- praćenje osoba, verifikacija lica, stereo
- učenje s malenim brojem označenih primjeraka (few-shot)
- samonadzirano učenje
- detekcija izvandistribucijskih podataka.