Robot iCub sa učí rozpoznávať objekty

Andrej Lúčny
Katedra aplikovanej informatiky,
FMFI (Matfyz), Univerzita Komenského v Bratislave

https://dai.fmph.uniba.sk/w/Project:TERAIS/sk

https://github.com/andylucny/whatiswhat

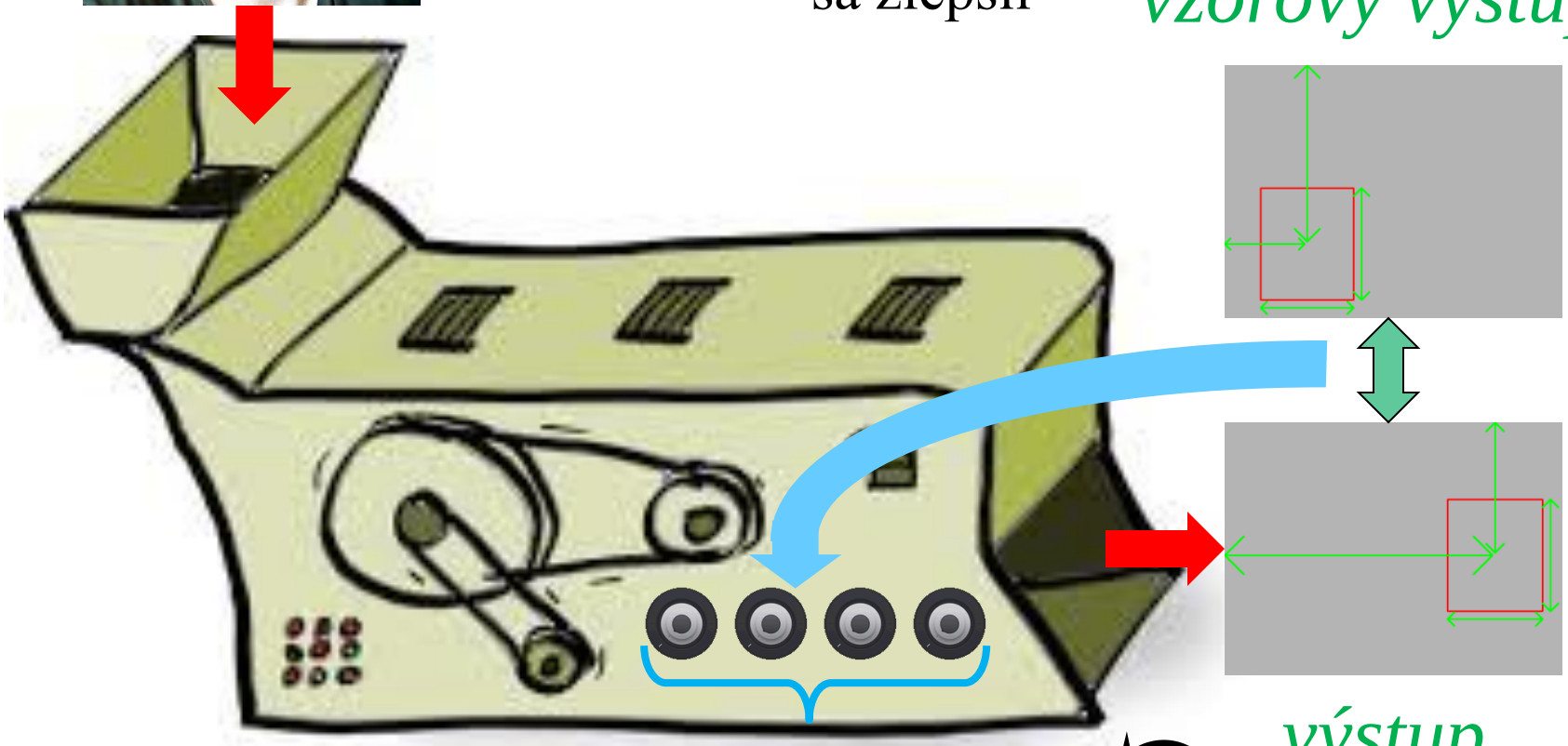
SEMINÁR
ROBOTIKA.SK

Neurónová sieť

Univerzálny stroj, pri ktorom z porovnania vzorových a skutočných výstupov vieme ako upraviť jeho parametre tak, aby sa zlepšil

vzorový vstup

vzorový výstup



parametre

výstup

trénovanie

Konvolučná neurónová sieť

Keď spracúvame obraz, vhodnou takou sieťou je konvolučná neurónová sieť, ktorá spracúva obraz pomocou:

- blokov konvolučných vrstiev, v ktorých každá vrstva aplikuje na obraz určitý kernel (= vážený priemer pixla a jeho okolia plus posunutie)
- redukcie dimenzie (zo štyroch pixelov jeden)
- expanzie dimenzie (z jedného pixela štyri)

Kernel

N=3 kanálov

3x3xN

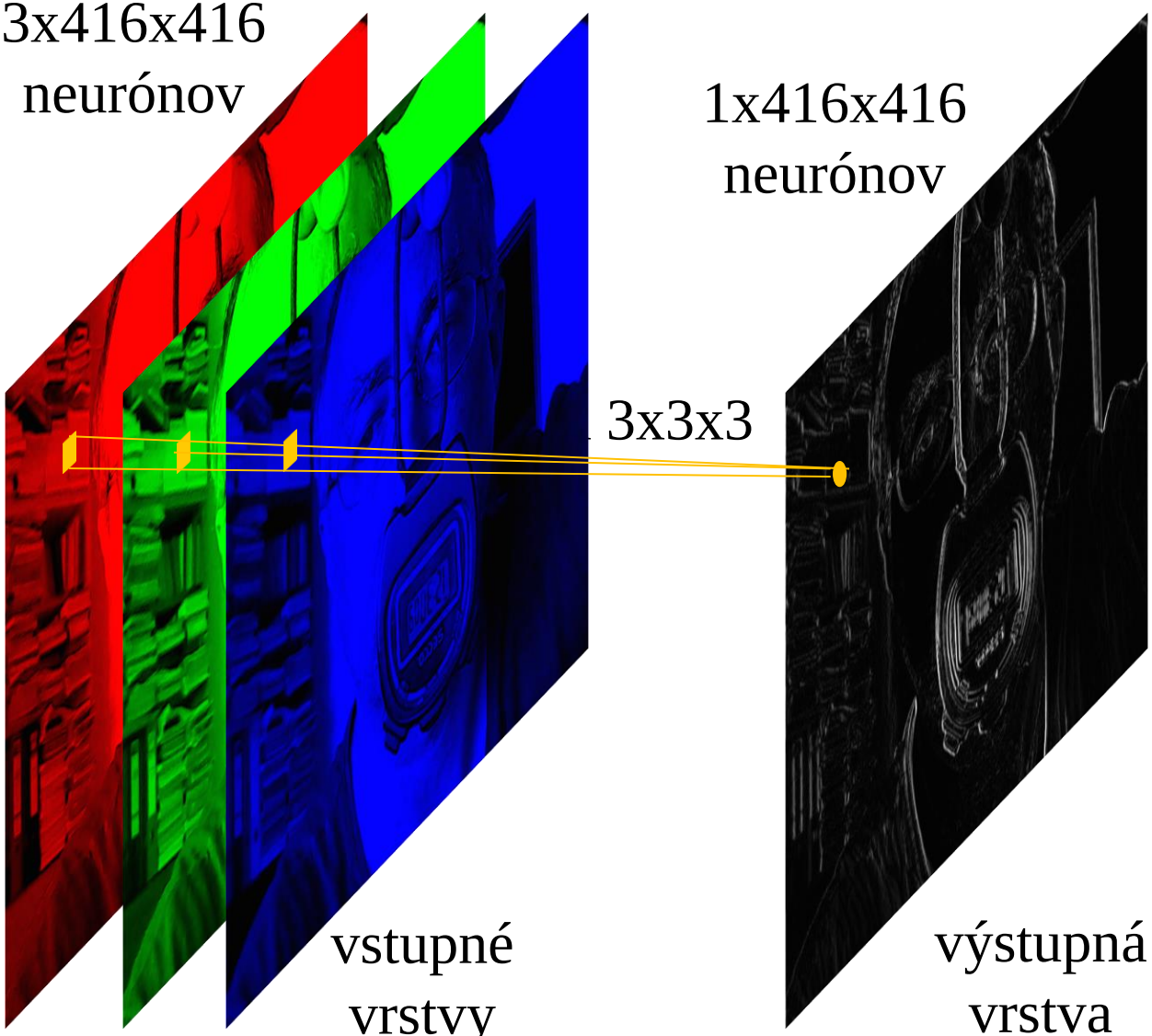
-1/3	0	1/3
-2/3	0	2/3
-1/3	0	1/3

+0

1 kanál

Napríklad: trojkanálový farebný obraz môžeme Sobelovým kernelom premeniť na zvislé hrany jednokanálového obrazu

Konvolučná vrstva

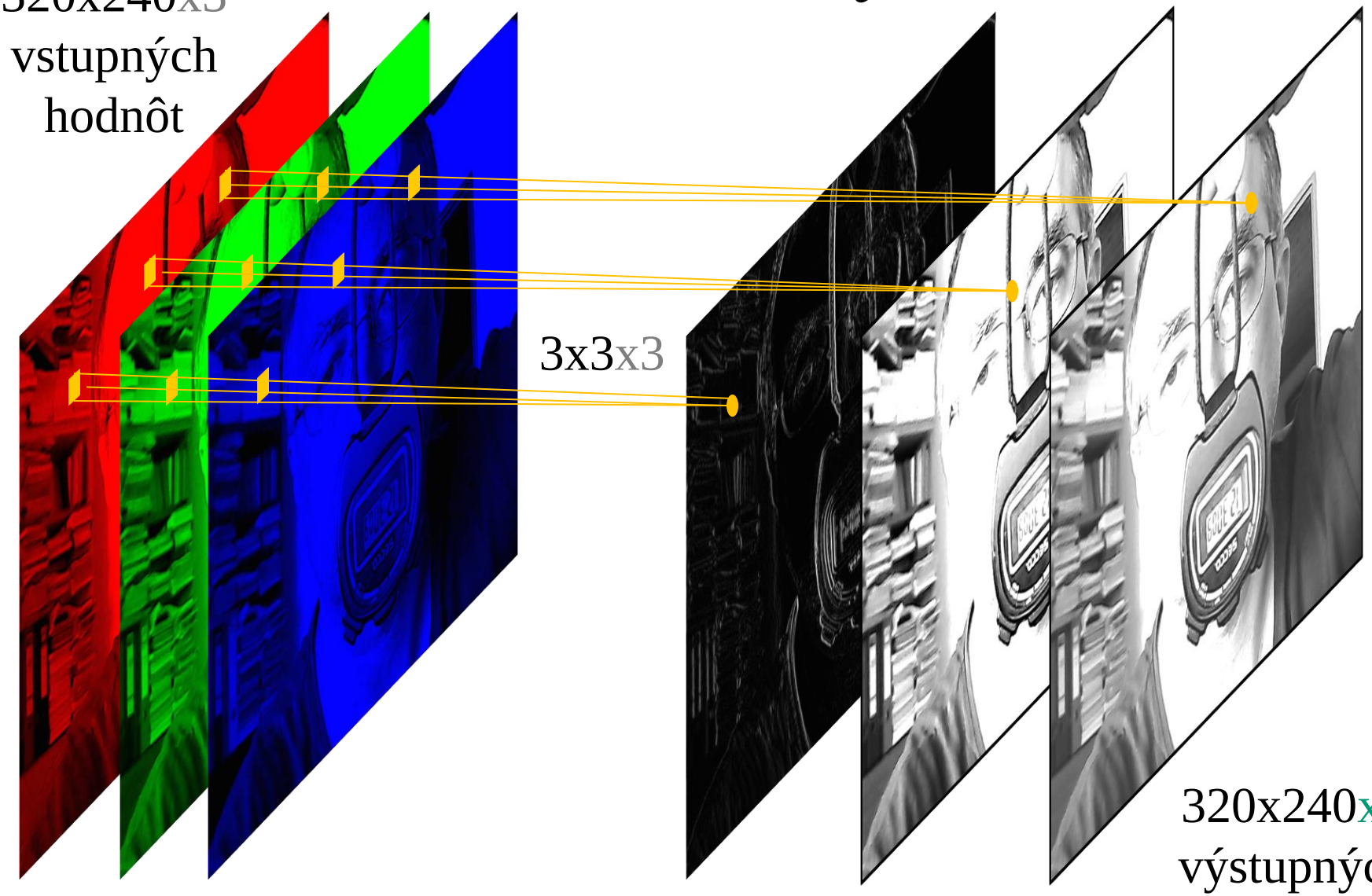


Každý neurón výstupnej vrstvy má spojenia na 3x3x3 = 27 neurónov vstupnej vrstvy

Váhy na spojeniach neuróny zdieľajú, t.j. celá vrstva má 173056 neurónov, ale len 27 parametrov, prípadne 28 s biasom

Blok konvolučných vrstiev

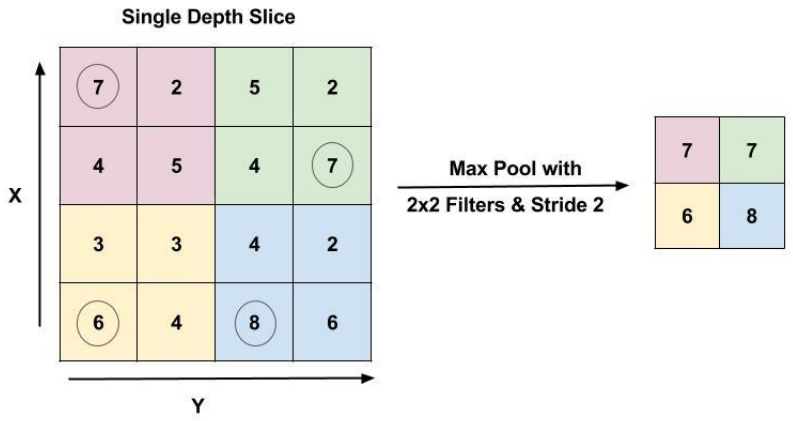
320x240x3
vstupných
hodnôt

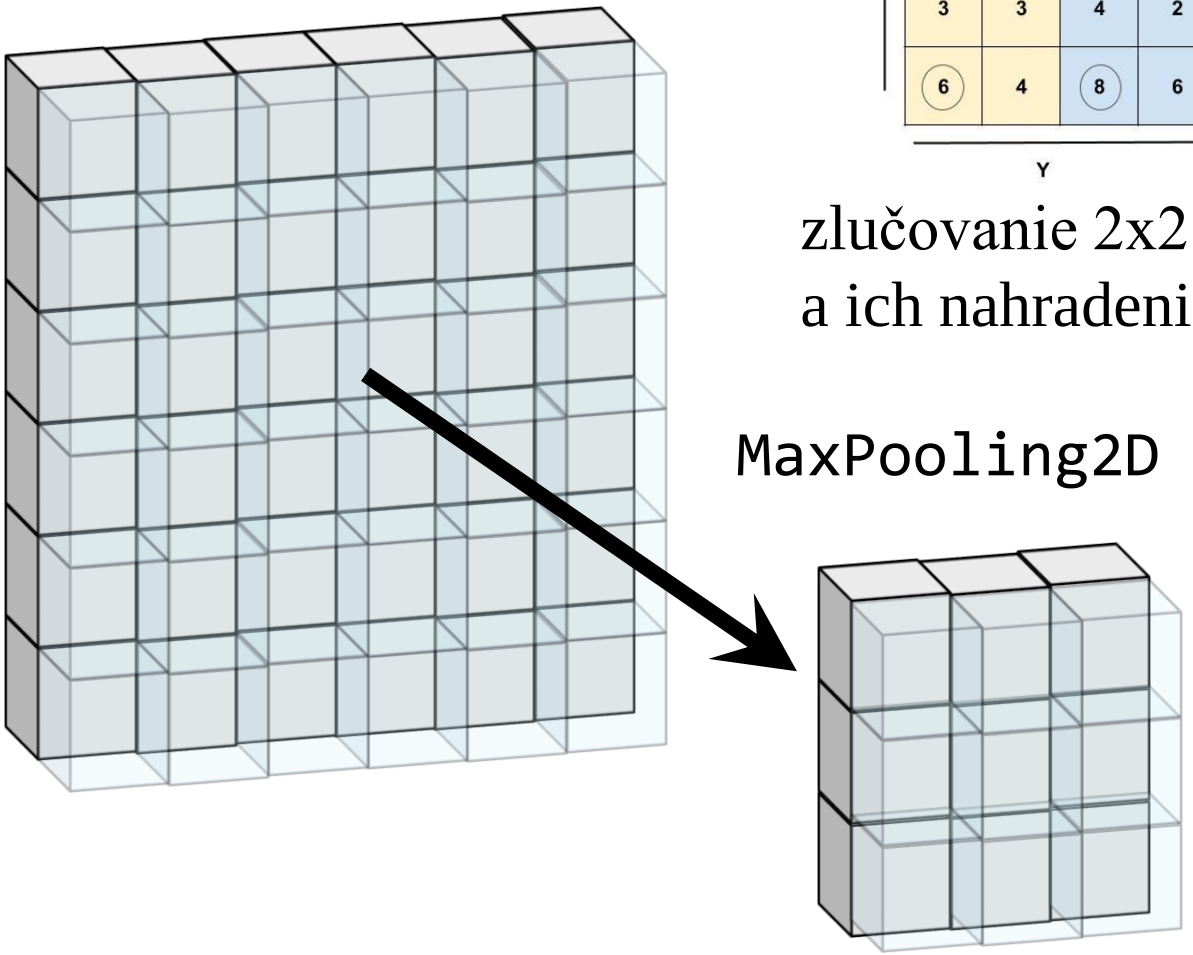


320x240x3
výstupných
hodnôt

Typická aktivačná funkcia: ReLU alebo LeakeyReLU

Redukcia dimenzie na báze maxima

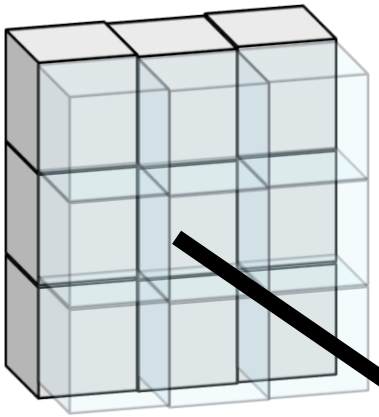




zlučovanie 2x2 pixelov s krokom 2
a ich nahradenie maximom

```
MaxPooling2D 2x2 stride=2
```

Expanzia dimenzie



Nearest Neighbor

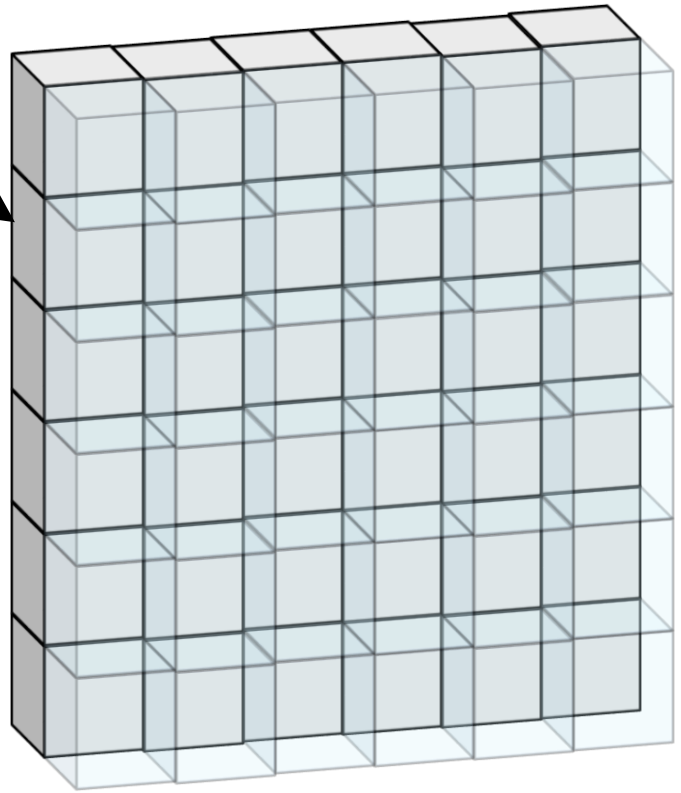
1	2
3	4

→

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Input: 2 x 2

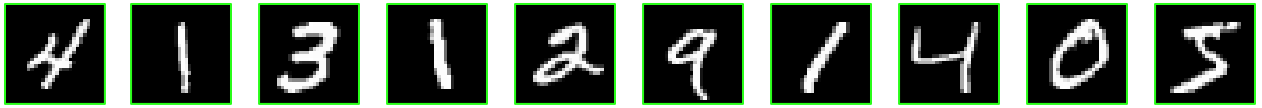
Output: 4 x 4



`UpSampling2D 2x2 stride=2`

Generatívne modelovanie

- Čím väčšiu sadu vzorových vstupov a výstupov máme, tým kvalitnejšiu neurónovú sieť dokážeme natrénovať

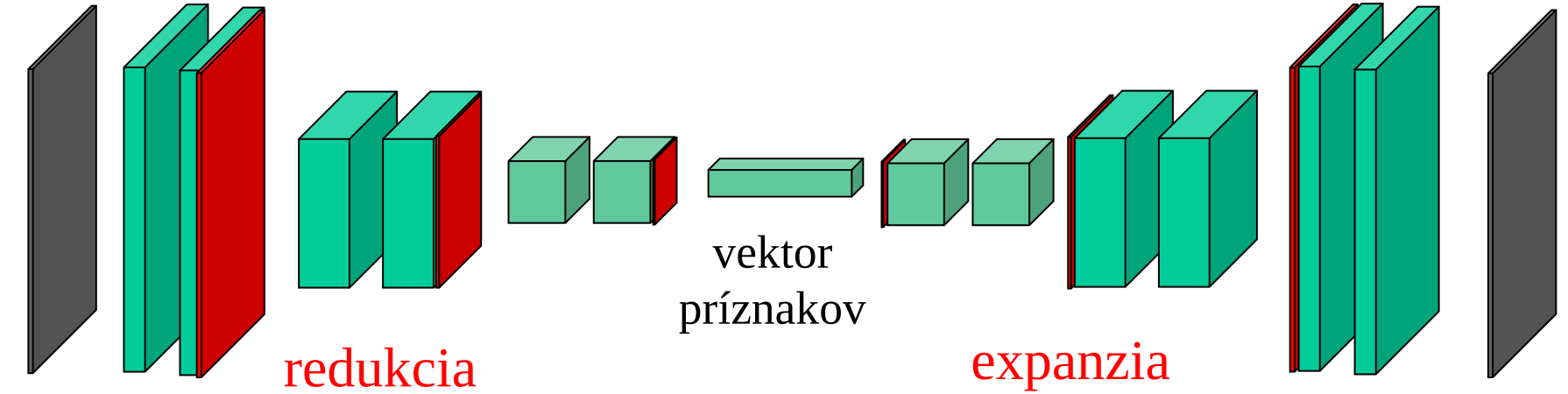


60000 obrázkov 28×28

- Ale tým sa aj viac narobíme, keď hovoríme aké majú vzorové výstupy byť (= anotácia dát)

- Dôležitú rolu hrajú preto **autoregresné** úlohy, ktoré anotáciu nevyžadujú

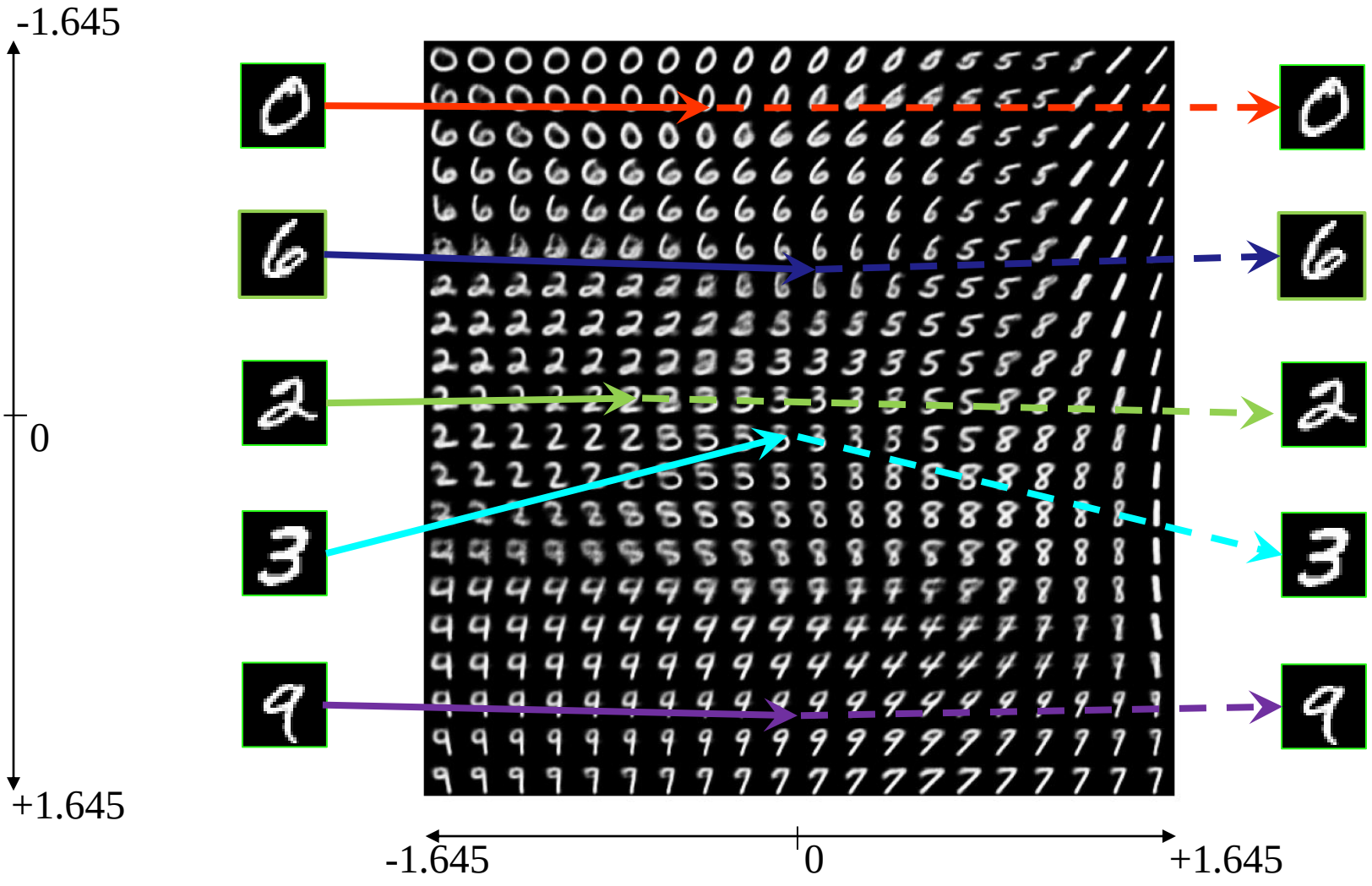
Konvolučný autokóder



bloky konvolučných vrstiev

chceme, aby:
VSTUP = VÝSTUP

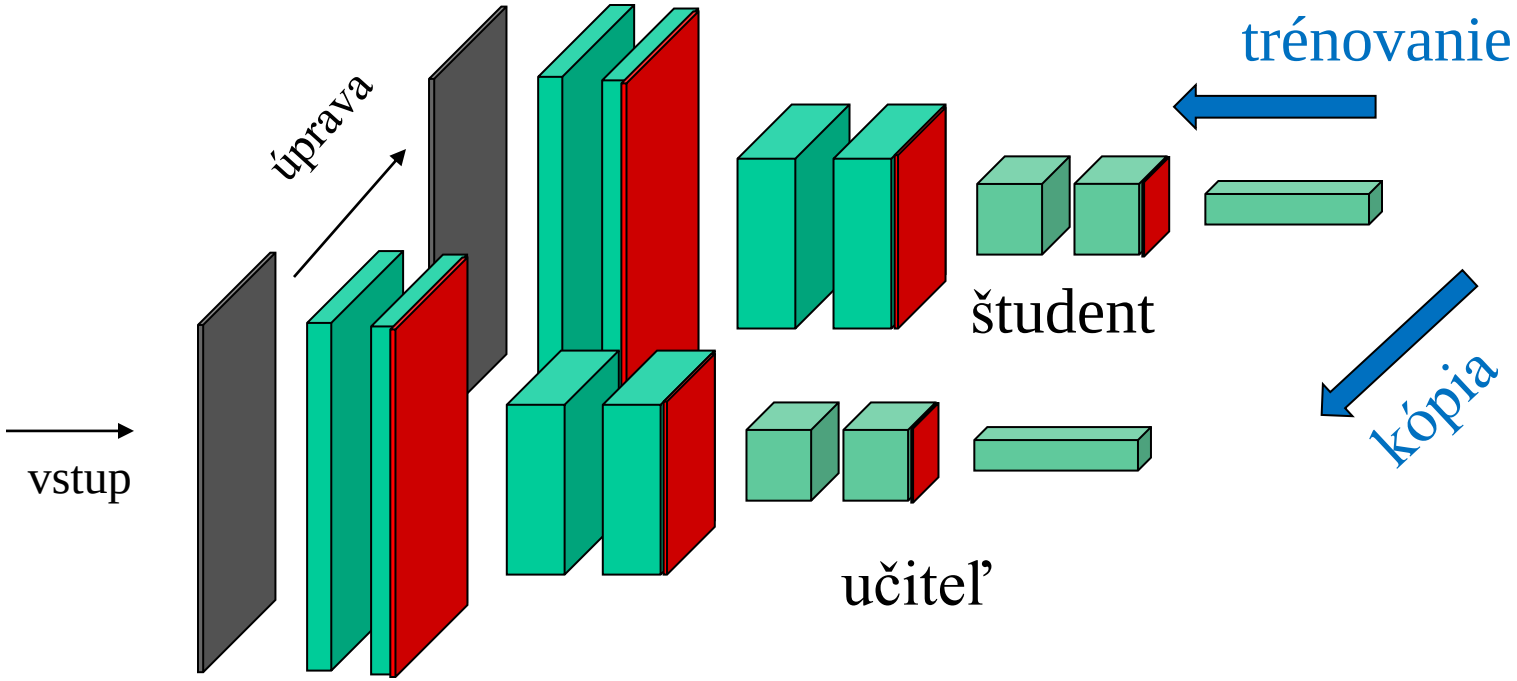
Autokóder



Extraktor obrazu

- Dokážeme to isté s obrazom, napríklad postavy stojacej pred ľubovoľnou scénou?
- Teoreticky by sme mohli podobne natrénovať autokóder z obrazov, ktoré môžeme vidieť a vziať z neho kóder a použiť ho ako extraktor
- Obraz má však oveľa väčšiu dimenziu, preto potrebuje príznakový vektor väčšej dimenzie aj oveľa dlhšie trénovanie
- V súčasnosti lepší výkon dosahujú metódy, ktoré trénujú priamo kóder pomocou tzv. samoučenia (self-supervision)

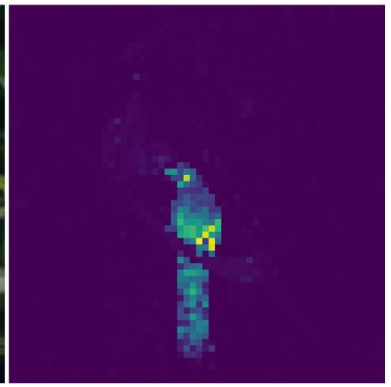
Samo sa učiaca sieť typu učiteľ-študent



- Trénujeme podľa učiteľa študenta: pre vstup, ktorý je upravený tak, že sa zachová jeho význam chceme od študenta odozvu čo najbližšiu tomu, čo povie učiteľ
- Z času na čas, skopírujeme študenta do učiteľa

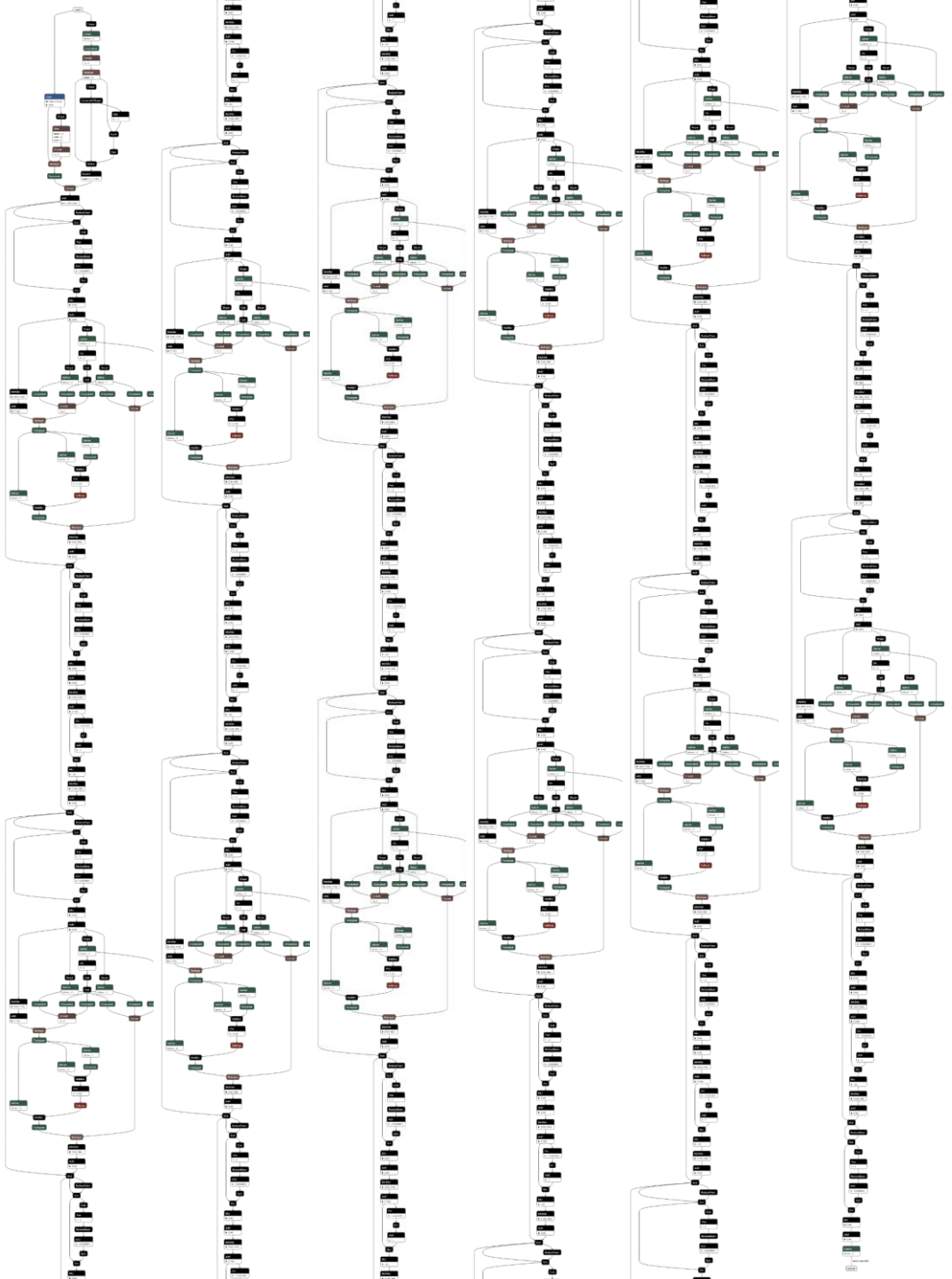
DINO

- Samo sa učiaca sieť typu učiteľ - študent, predtrénová pre pozornostné mapy



obraz 224x224

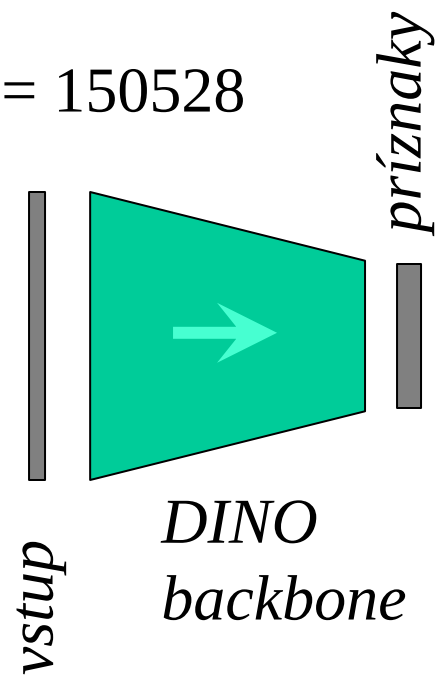
mení na púhych 384 príznakov



Extraktor obrazu

- Použijeme ViT backbone z DINO (small size)

224x224x3 = 150528

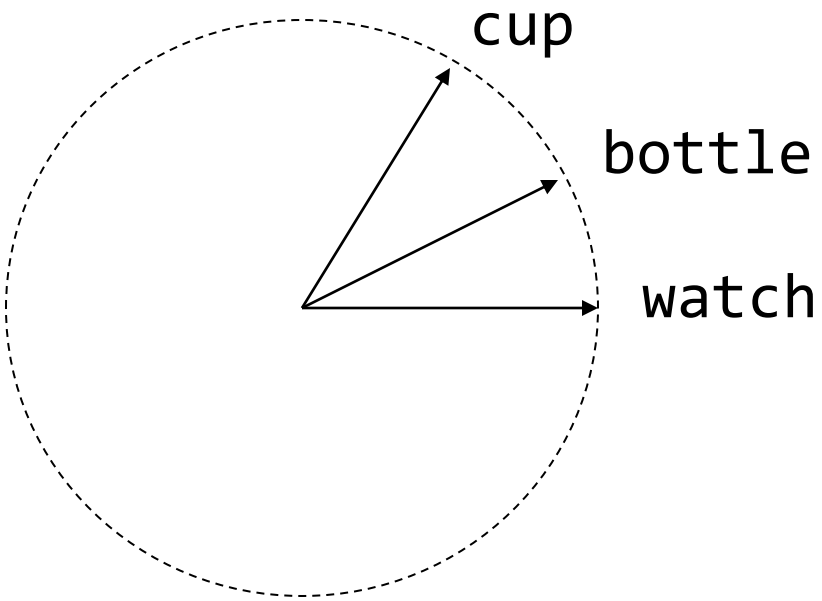


1x384

```
[[  0.37   2.63   0.7    0.66  -0.25  -2.94  -1.57   3.66  -1.14   2.21
   -2.61   6.49   3.55  -3.92  -1.89  12.06  -4.2    1.16   5.12  -2.41
    3.24  -3.82   0.14   5.06  -2.52  10.69  -1.63   1.73  -0.41   1.6
    0.02   0.11   0.33  -0.84  -2.36  -2.96  -4.43   1.32  -1.57   0.03
    2.32   3.31   1.93   1.46  -0.84  14.62  -0.1    0.49  -3.44   1.89
   -0.53   1.04  -2.     1.58  -3.18   0.46  -5.31   1.68   2.17  -4.7
    0.82  -1.25  -0.17  -5.52   1.06   5.82  -2.36  -1.86  -2.2   -1.93
   -5.48  -2.73  -2.02  -0.53  14.55  -4.19   5.7   -2.02   1.1  -10.93
   -0.3   -1.8    0.97   0.63  -4.91  -1.63  -0.21  -5.03  -3.25   7.83
   -4.9   -1.59  -1.32   1.73  -7.65   0.78   3.06  -2.85  -0.43   4.66
    5.16   2.61   5.53   0.82   0.05   3.62  -1.28   0.7    1.87  -1.19
   -8.28   2.16   0.5   -1.17   1.74   2.08  -4.38   6.68   5.02   6.27
   -3.14  12.75 -16.36  -1.21   7.25  -1.63   1.71  -5.21   6.9    1.98
   -1.75   2.8   -3.03   1.3    7.8    3.06  -4.18  -4.35 -12.24  -0.08
   -3.5    2.51  -0.19  -3.81  -4.18   7.67  -2.84   1.41   0.87  -3.27
    0.16  -0.09   1.73  -2.23  -9.82  -2.58  -3.4   -4.08   0.56  -2.48
   -5.39   4.59   0.72   3.32   3.29  -3.6   -0.13   4.65  -5.15  -5.24
   -2.32   5.93   0.89   2.02  -3.25  -1.98  -0.64   4.24   8.09  -5.61
   -3.6   -0.18  -5.54   0.6   -3.88  -5.02   2.02  -1.16   1.77   2.58
   -1.25   0.32  -4.24   3.61  -0.5   -2.89   1.52   7.71  -2.9   10.41
   -3.12  -5.3    4.03   2.     -5.6   -2.29   7.02   3.53   2.36  -2.59
    1.41   5.     2.18  -2.36   3.39   5.55   4.47   1.59   4.22   0.68
    1.92  -0.12  -3.52  -2.86   1.18  -1.92   9.13  -1.04   0.71   3.39
   -5.35  -1.52  -3.46   4.2   -0.22  -0.17   5.58   1.04  -5.02   0.95
   -0.57   4.32  -2.39   2.71  -1.65  -1.62   3.19  -1.44   0.61   2.51
   -2.11   2.93  -0.2  -13.94  -3.31  -5.4    6.45  -3.6    4.73  -2.23
   -1.02   0.57  -1.45   0.89  -3.3   -0.41   2.56 -15.4   -3.78  -6.35
    1.45   9.59  -3.38   6.18  -6.55   4.05  -2.75   3.43  -6.72  -7.45
    6.67  -7.03  -1.58   6.16  -0.84  -0.22   2.63  -2.92  -6.13  -4.14
    1.31  -2.06   1.31   0.02   1.42   1.36  -2.95   7.2  -10.27   1.29
    1.42   1.56   5.59   4.71  -3.84  -0.     2.94  -4.96  -1.7   -0.57
    6.49   4.24   3.33   4.18   0.52   2.82  -3.45  -6.27   1.52  -3.25
   -3.31   4.92   4.1    2.47  -0.99   9.92   2.36  -7.38   3.18   0.93
   -0.37   3.08  -2.4   -0.33  -2.97   5.68   1.38 -10.75   1.02   4.69
    4.61  -4.56   4.14  -4.58   0.44  -6.04  -6.2   -3.05  -3.61  -1.19
   -0.05   1.59   1.01   1.36  -1.4    4.09   4.56   6.13  -1.64  -0.25
   -1.57  -2.04   0.74  -6.78  -3.18   0.09  -0.53   6.95   5.38   4.57
    1.83  -2.76   0.59  -3.79   4.85  -4.15   1.11  -3.35   0.87  -4.42
   -1.34   4.35   7.02  -1.62]]
```

Čas inferencie na 4GB GPU: 0.05s

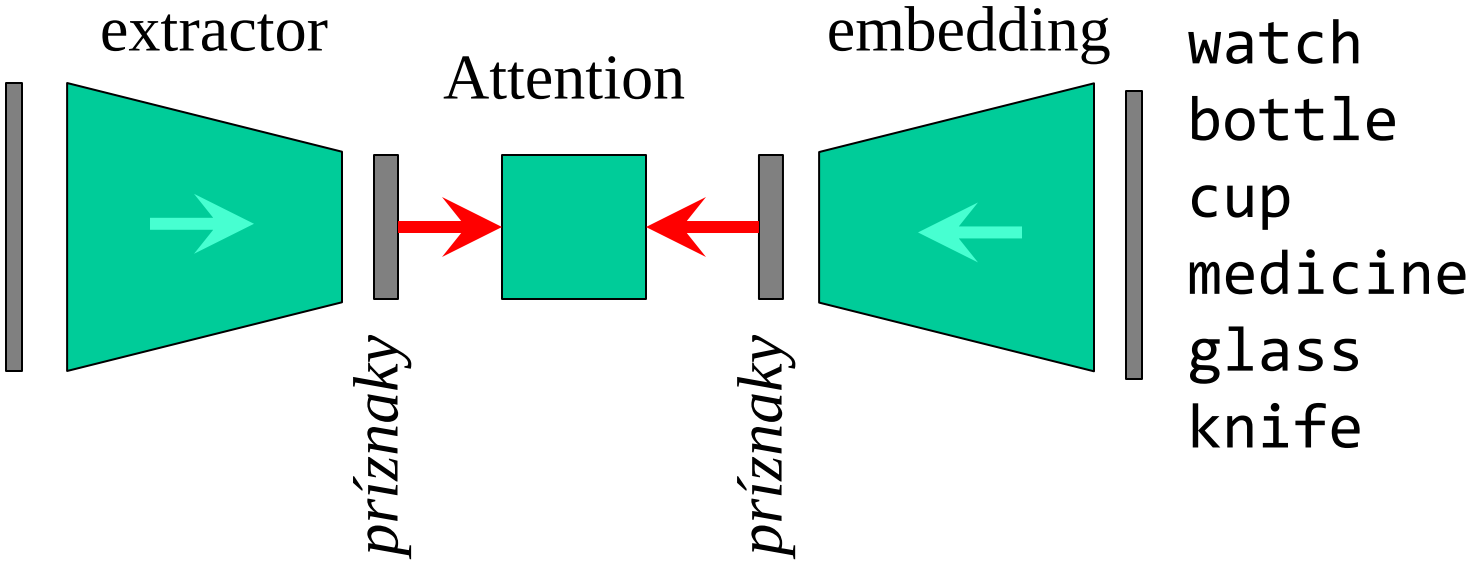
Embedding / Pozičné kódovanie



$$embed(name_i) = [cos(i\phi), sin(i\phi)]$$

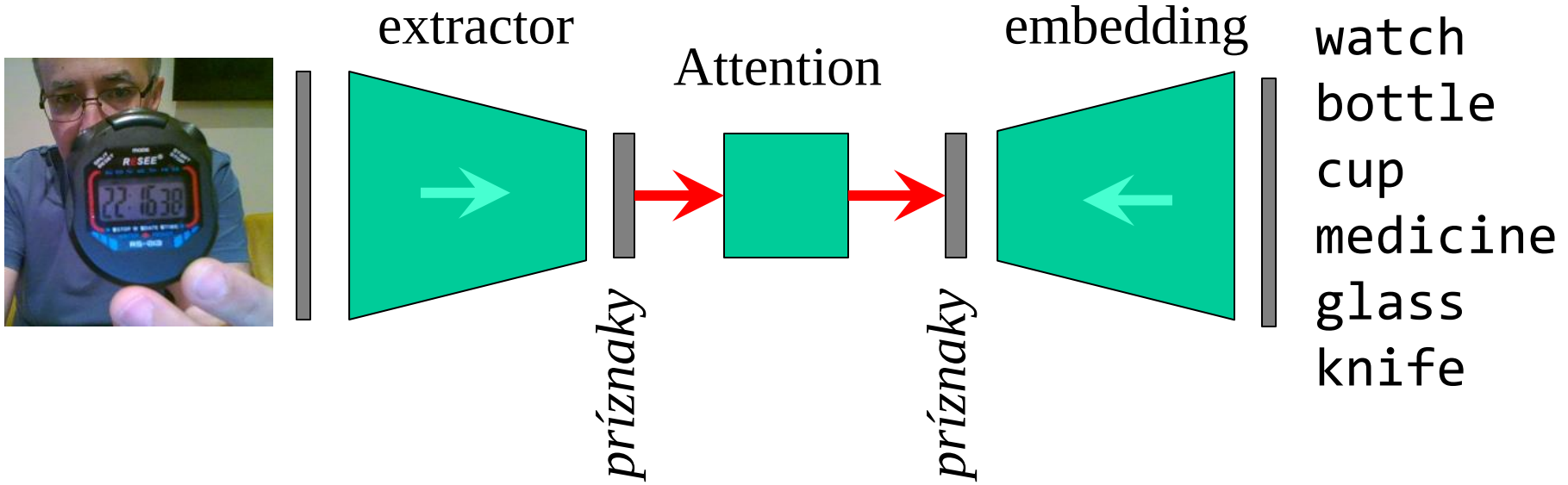
$$i = \lfloor arctan(embed(name_i))/\phi \rceil$$

Asociovanie na jeden pokus



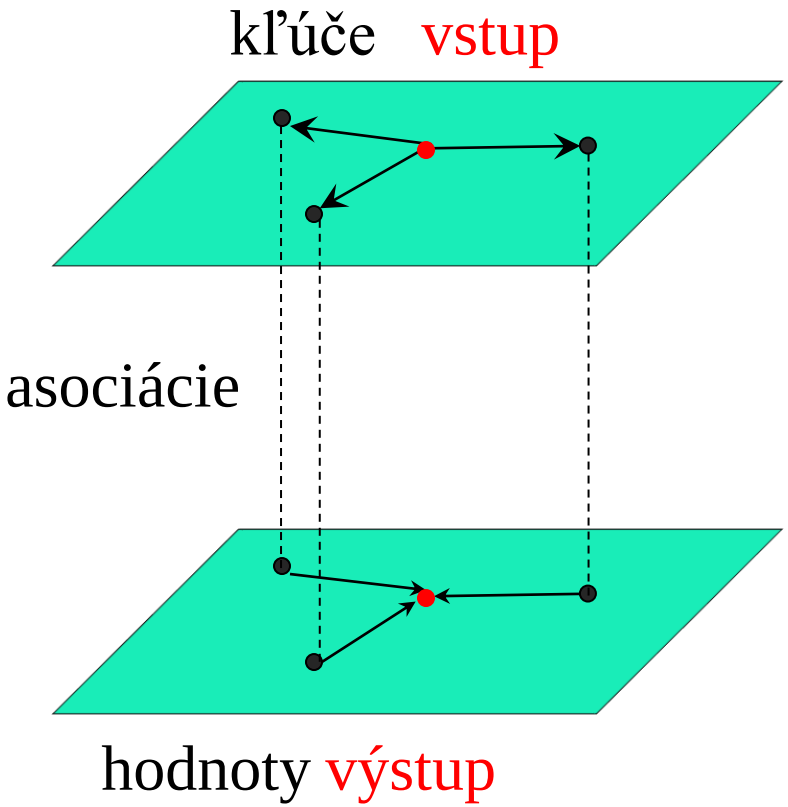
zbierame páry kľúč – hodnota

Asociovanie na jeden pokus



aplikujeme asociácie kľúč – hodnota

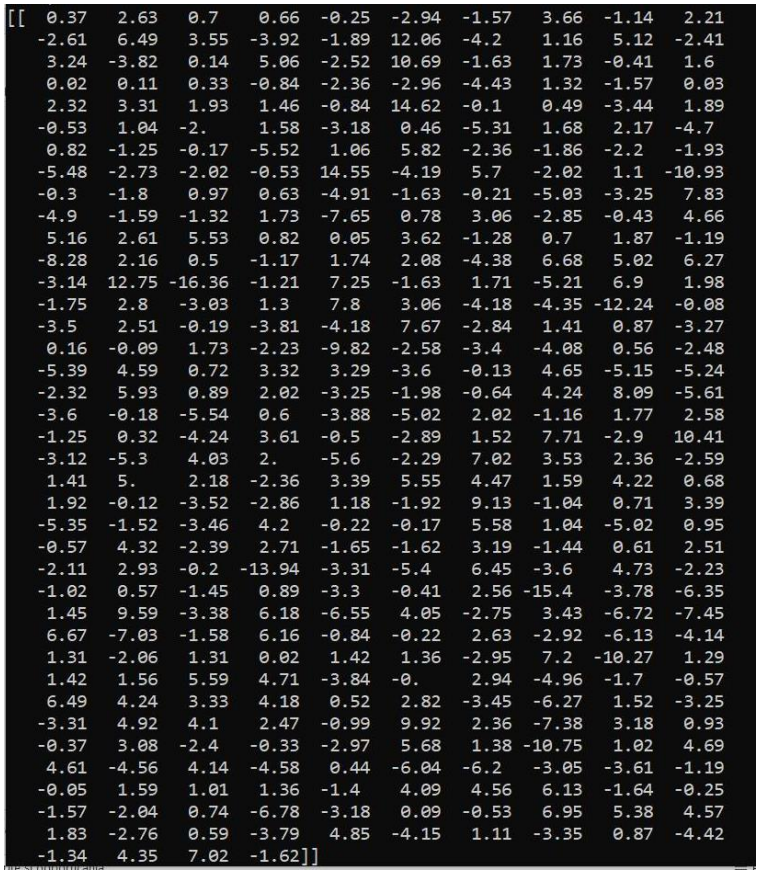
Asociovanie



- Vstup namiešane z tých kľúčov, ktoré poznáme
- Výstup bude analogickou miešaninou asociovaných hodnôt

Asociovanie

1x384



```
[[  0.37   2.63   0.7    0.66  -0.25  -2.94  -1.57   3.66  -1.14   2.21
   -2.61   6.49   3.55  -3.92  -1.89  12.06  -4.2    1.16   5.12  -2.41
    3.24  -3.82   0.14   5.06  -2.52  10.69  -1.63   1.73  -0.41   1.6
    0.02   0.11   0.33  -0.84  -2.36  -2.96  -4.43   1.32  -1.57   0.03
    2.32   3.31   1.93   1.46  -0.84  14.62  -0.1    0.49  -3.44   1.89
   -0.53   1.04  -2.     1.58  -3.18   0.46  -5.31   1.68   2.17  -4.7
    0.82  -1.25  -0.17  -5.52   1.06   5.82  -2.36  -1.86  -2.2   -1.93
   -5.48  -2.73  -2.02  -0.53  14.55  -4.19   5.7   -2.02   1.1  -10.93
   -0.3   -1.8    0.97   0.63  -4.91  -1.63  -0.21  -5.03  -3.25   7.83
   -4.9   -1.59  -1.32   1.73  -7.65   0.78   3.06  -2.85  -0.43   4.66
    5.16   2.61   5.53   0.82   0.05   3.62  -1.28   0.7    1.87  -1.19
   -8.28   2.16   0.5   -1.17   1.74   2.08  -4.38   6.68   5.02   6.27
   -3.14  12.75 -16.36  -1.21   7.25  -1.63   1.71  -5.21   6.9    1.98
   -1.75   2.8   -3.03   1.3    7.8    3.06  -4.18  -4.35 -12.24  -0.08
   -3.5    2.51  -0.19  -3.81  -4.18   7.67  -2.84   1.41   0.87  -3.27
    0.16  -0.09   1.73  -2.23  -9.82  -2.58  -3.4   -4.08   0.56  -2.48
   -5.39   4.59   0.72   3.32   3.29  -3.6   -0.13   4.65  -5.15  -5.24
   -2.32   5.93   0.89   2.02  -3.25  -1.98  -0.64   4.24   8.09  -5.61
   -3.6   -0.18  -5.54   0.6   -3.88  -5.02   2.02  -1.16   1.77   2.58
   -1.25   0.32  -4.24   3.61  -0.5   -2.89   1.52   7.71  -2.9   10.41
   -3.12  -5.3    4.03   2.    -5.6   -2.29   7.02   3.53   2.36  -2.59
    1.41   5.     2.18  -2.36   3.39   5.55   4.47   1.59   4.22   0.68
    1.92  -0.12  -3.52  -2.86   1.18  -1.92   9.13  -1.04   0.71   3.39
   -5.35  -1.52  -3.46   4.2   -0.22  -0.17   5.58   1.04  -5.02   0.95
   -0.57   4.32  -2.39   2.71  -1.65  -1.62   3.19  -1.44   0.61   2.51
   -2.11   2.93  -0.2  -13.94  -3.31  -5.4    6.45  -3.6    4.73  -2.23
   -1.02   0.57  -1.45   0.89  -3.3   -0.41   2.56 -15.4   -3.78  -6.35
    1.45   9.59  -3.38   6.18  -6.55   4.05  -2.75   3.43  -6.72  -7.45
    6.67  -7.03  -1.58   6.16  -0.84  -0.22   2.63  -2.92  -6.13  -4.14
    1.31  -2.06   1.31   0.02   1.42   1.36  -2.95   7.2  -10.27   1.29
    1.42   1.56   5.59   4.71  -3.84  -0.     2.94  -4.96  -1.7   -0.57
    6.49   4.24   3.33   4.18   0.52   2.82  -3.45  -6.27   1.52  -3.25
   -3.31   4.92   4.1    2.47  -0.99   9.92   2.36  -7.38   3.18   0.93
   -0.37   3.08  -2.4   -0.33  -2.97   5.68   1.38 -10.75   1.02   4.69
    4.61  -4.56   4.14  -4.58   0.44  -6.04  -6.2   -3.05  -3.61  -1.19
   -0.05   1.59   1.01   1.36  -1.4    4.09   4.56   6.13  -1.64  -0.25
   -1.57  -2.04   0.74  -6.78  -3.18   0.09  -0.53   6.95   5.38   4.57
    1.83  -2.76   0.59  -3.79   4.85  -4.15   1.11  -3.35   0.87  -4.42
   -1.34   4.35   7.02  -1.62]]
```

kľúč

$$K = \begin{pmatrix} k_1 \\ k_2 \\ \dots \\ k_l \end{pmatrix} \qquad V = \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_l \end{pmatrix}$$

$$cos\alpha_i = \frac{q \cdot k_i}{\|q\| \|k_i\|}$$

1x2

```
array([[ 0.64, -0.32]])
```

hodnota

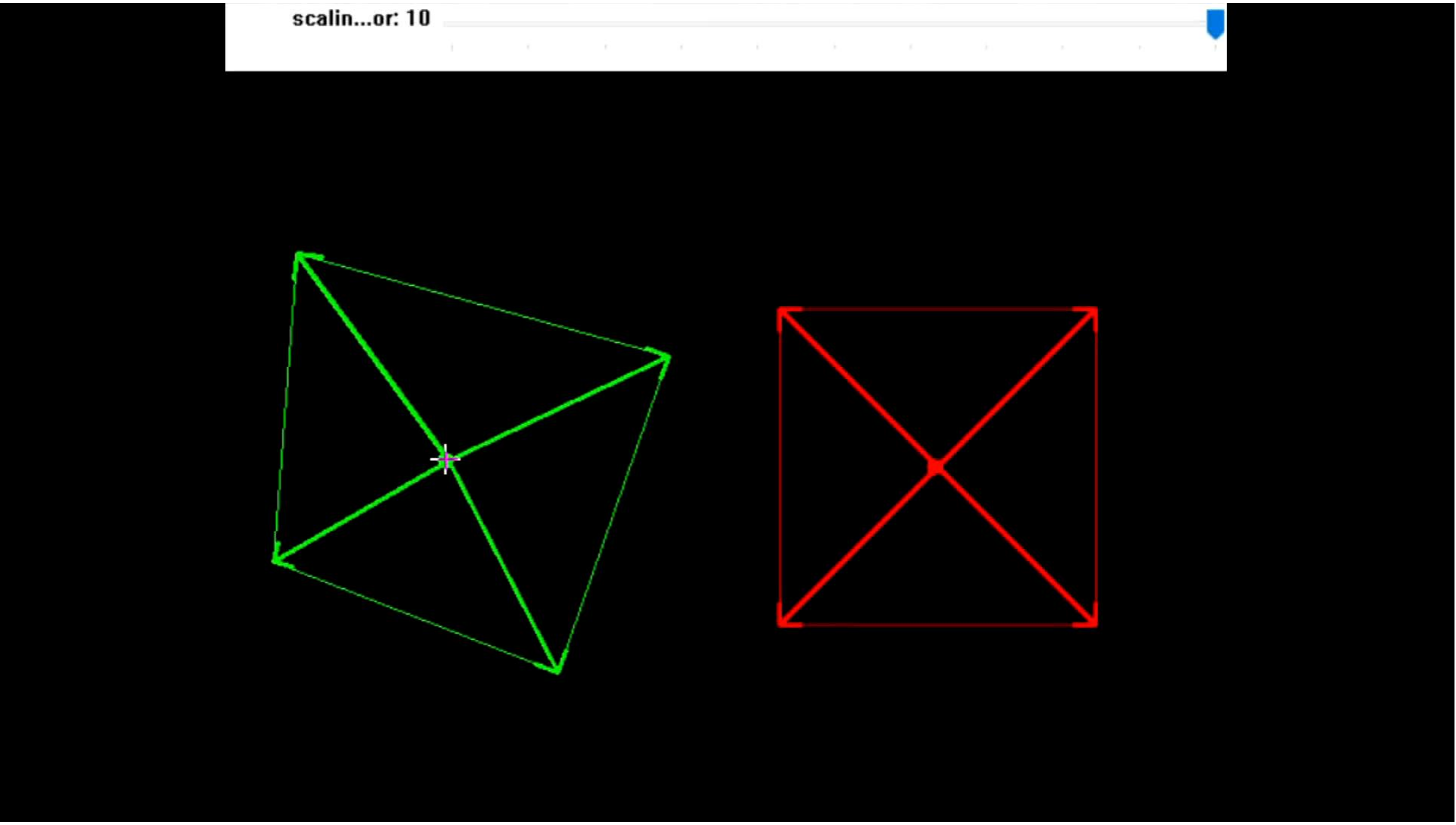
$$A(q, K, V) = softmax\left(\frac{qK^T}{d}\right)V$$

$$softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

škálovací faktor

$1/n^2$

$\sqrt{(n)}$

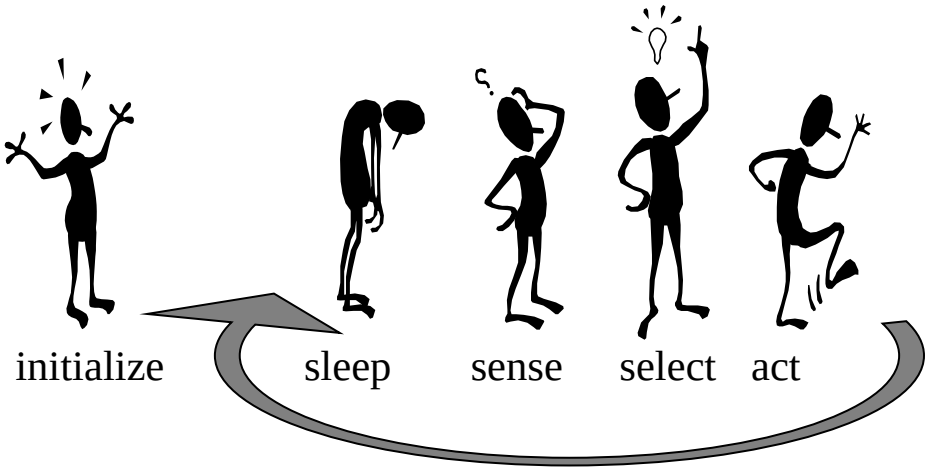


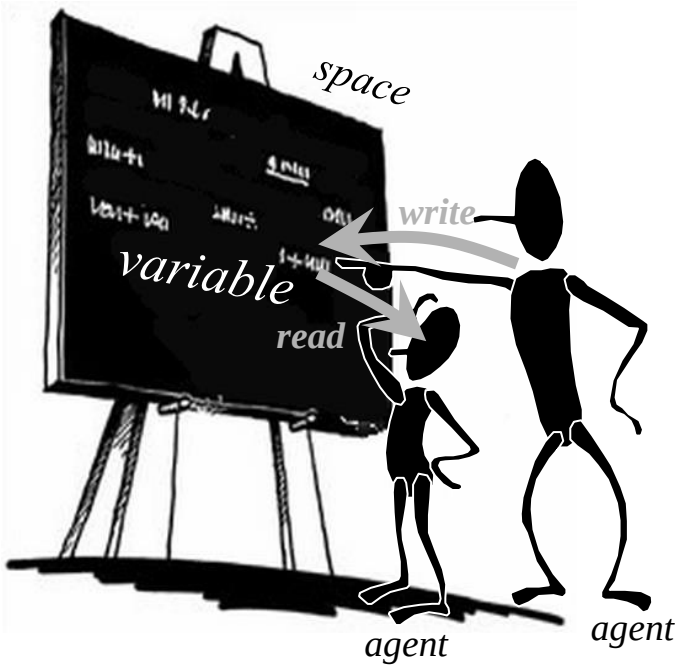
Integrácia

- audio recording, silence detection (sr)
- transcription from speech to text (torch, whisper)
- text pattern matching (re)
- image grabbing from camera (opencv)
- feature detection (onnxruntime)
- asociating of features with names embedding (numpy)
- application of associations via attention (numpy)
- speech synthesis (espeak)
- lips movements during speaking (pyicubsim)

Architektúra Agent-Space

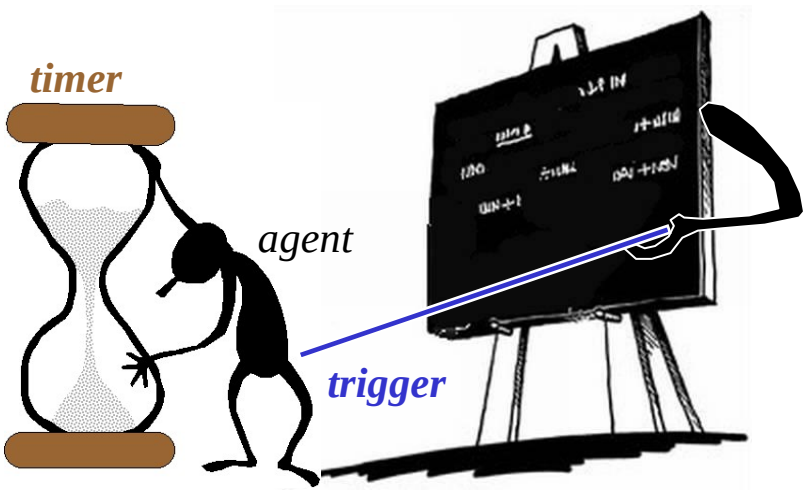
- Systém sa skladá z agentov
- Agenti komunikujú cez blackboard (Space)





Implementácia v Pythone

- Každý agent má vlastné vlákno
- Reentrantne volá metódy *read* a *write* sigletonu *Space*
- Agent je budený časovačom (pravidelne) alebo spúšťačom (zavolaním metódy *write* iným agentom)



Implementácia v Pythone

Blackboard (rozšírené dictionary)

```
from agentspace import space

space['a'] = 2
print(space['a']) # 2

print(space['b']) # None

print(space(default=-1)['b']) # -1
```

Implementácia v Pythone

validita a priority

```
space(validity=0.5)['b'] = 1
print(space['b']) # 1
time.sleep(1)
print(space(default=-1)['b']) # -1

space['c'] = 1 # default priority is 1.0
space(validity=0.5,priority=2.0)['c'] = 2
space(validity=2.0,priority=1.5)['c'] = 3
print(space(default=0)['c']) # 2
time.sleep(1)
print(space(default=0)['c']) # 0
```

Implementácia v Pythone

```
from agentspace import Agent

class MyAgent(Agent):

    def __init__(self, args):
        process(args)
        super().__init__()

    def init(self):
        self.attach_timer(1.0)

    def senseSelectAct(self):
        print('hallo')

MyAgent()
```

agenti (objekty ktoré majú vlastné vlákno)

budený časovačom

Implementácia v Pythone

```
from agentspace import Agent

class MyAgent(Agent):

    def __init__(self, args):
        process(args)
        super().__init__()

    def init(self):
        space.attach_trigger('a',self)

    def senseSelectAct(self):
        print('hallo')

MyAgent()
```

agenti (objekty ktoré majú vlastné vlákno)

budený spúšťačom

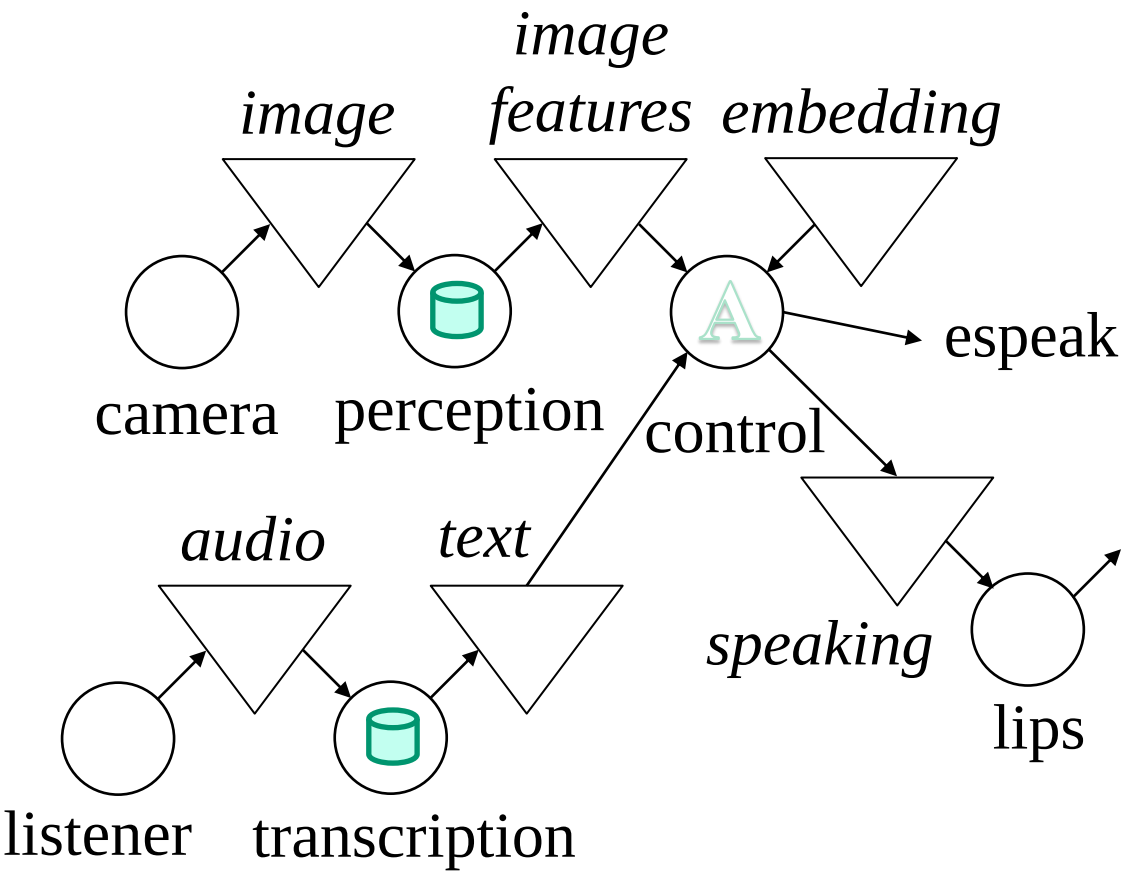
Príklad: Transcripcia (STT)

```
class TranscriptionAgent(Agent):

def init(self):
    self.audio_model = whisper.load_model("base.en").to('cuda')
    space.attach_trigger('audio',self)

def senseSelectAct(self):
    audio_data = space[self.nameAudio]
    if audio_data is not None:
        result = self.audio_model.transcribe(audio_data)
        space(validity=1.0)['text'] = result['text']
```

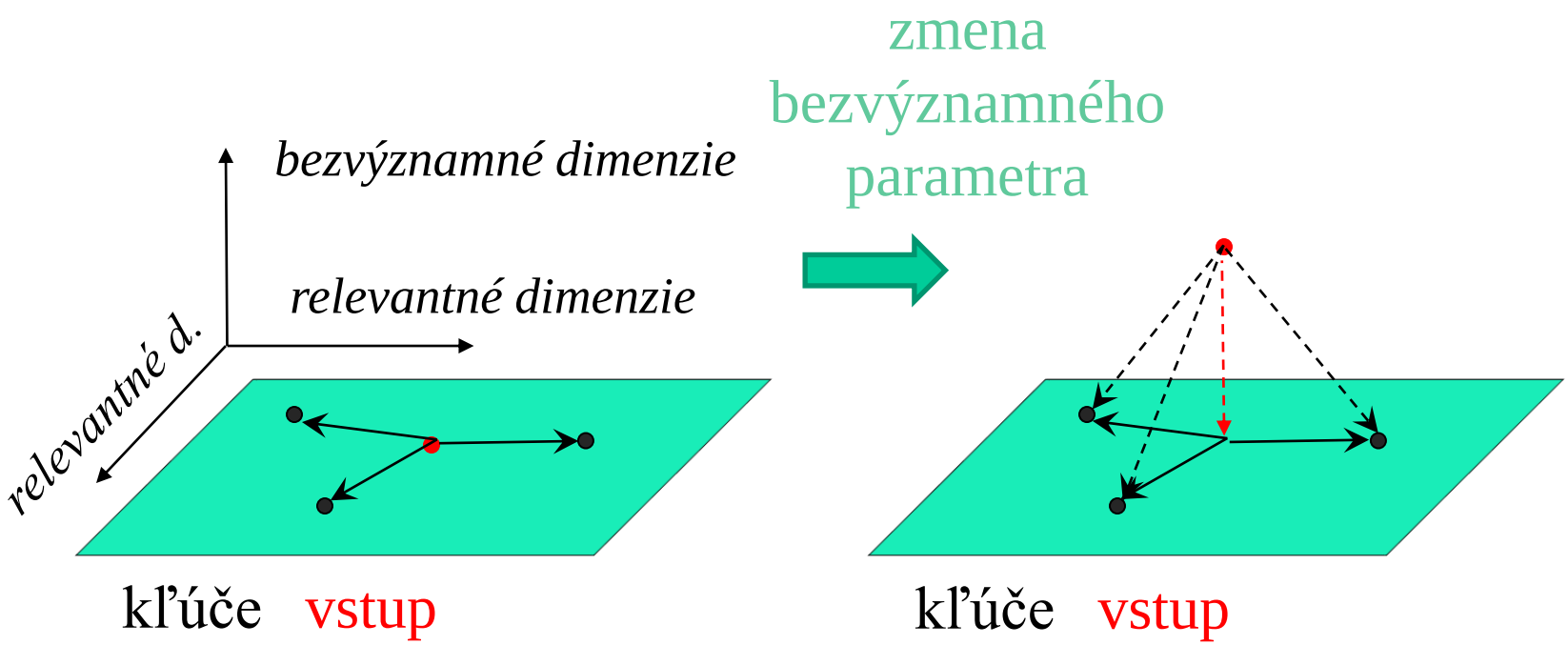
Systém na pomenovávanie objektov



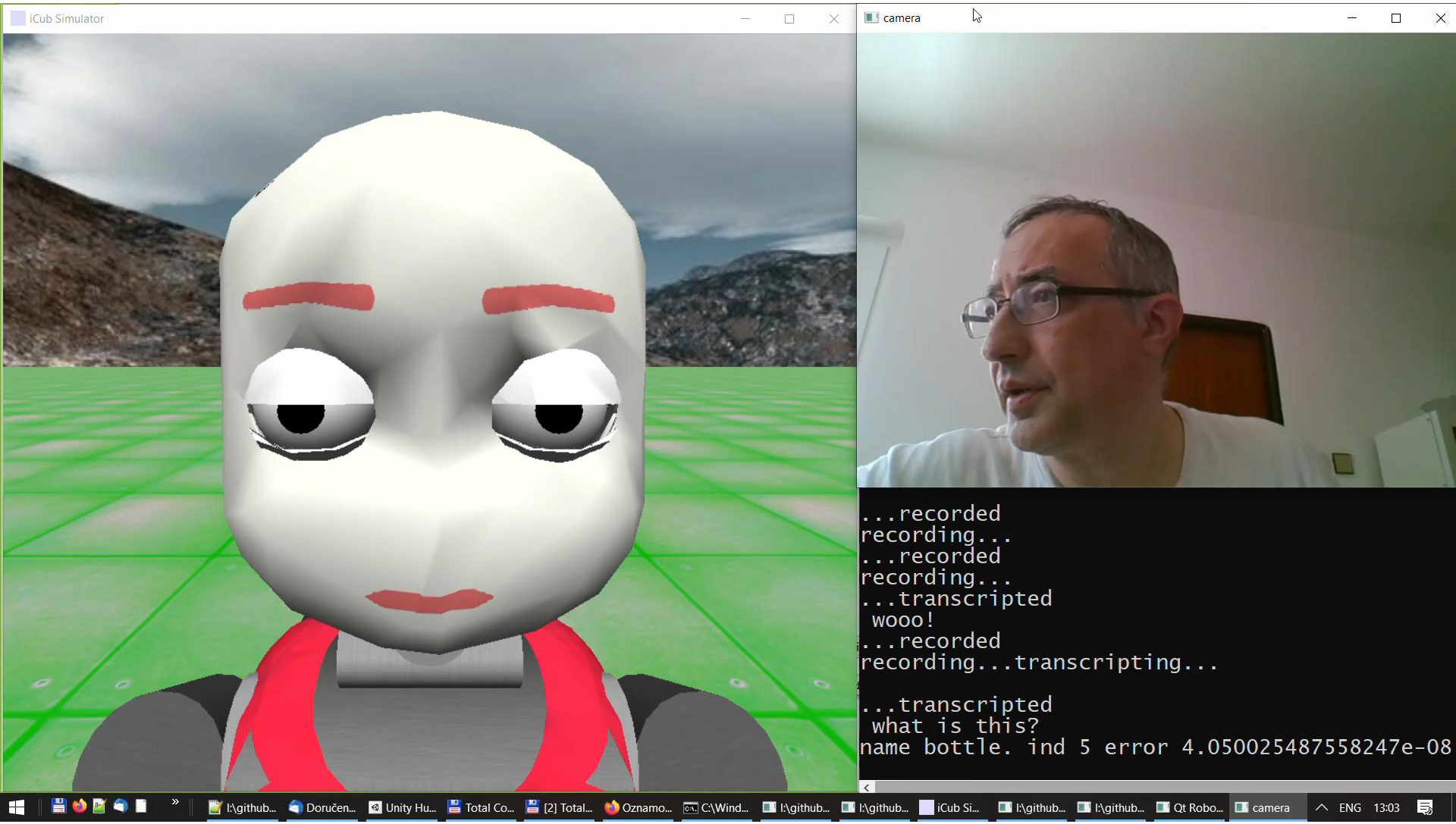
Pomenovávanie objektov

https://youtu.be/nQnEdyfDh5I

Nezlyhá to, keď zmeníme farbu steny za učiteľom?



Pomenovávanie objektov



https://youtu.be/VFPgt9UyUbI

Ďakujem za pozornosť !

https://github.com/andylucny/whatiswhat

Výzva: prispejte nápadom na prednášajúceho alebo tému!
Ponúknite vlastnú prednášku!