

# Praktikum zo strojového učenia a umelej inteligencie na vizuálnych dátach

*Andrej Lúčny*

*Katedra aplikovanej informatiky FMFI UK*

*lucny@fmph.uniba.sk*

*[http://dai.fmph.uniba.sk/w/Andrej\\_Lucny](http://dai.fmph.uniba.sk/w/Andrej_Lucny)*

*[www.agentspace.org/praktikum](http://www.agentspace.org/praktikum)*

13

# Konvolučné neurónové siete

Spracovanie obrazu si teda žiadalo špecifický prístup a to dalo vzniknúť špeciálnemu druhu neurónových sietí, založených na spracovaní obrazu pomocou kernelov



3x3x3

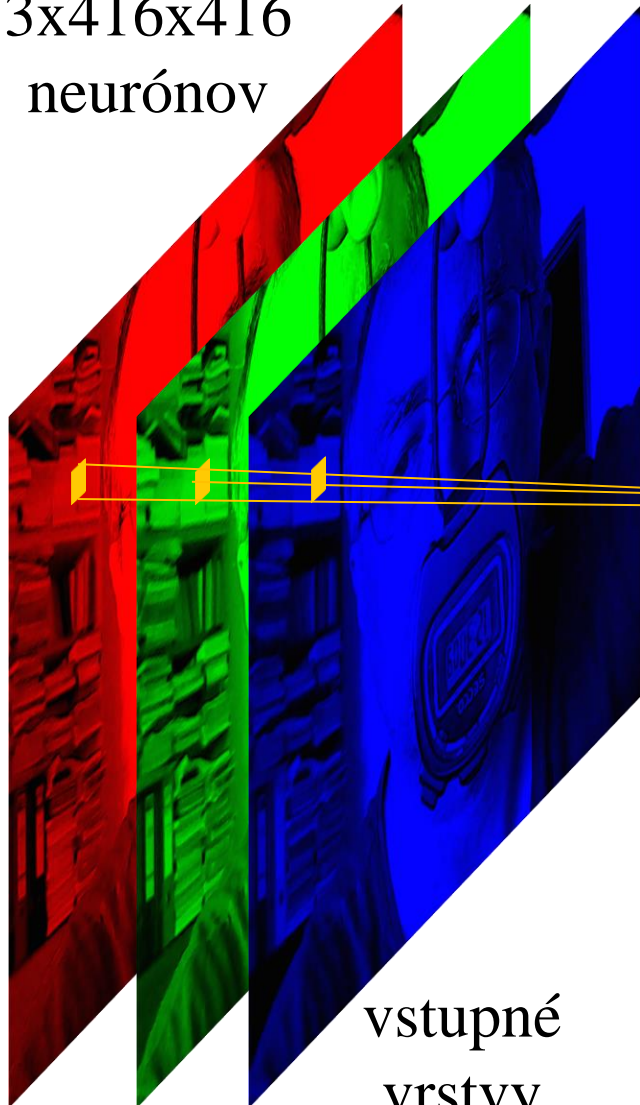
	1/3	0	1/3	
	1/3	0	1/3	
-1/3	0	1/3	3	
-2/3	0	2/3	3	
-1/3	0	1/3	3	



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

# Konvolučná vrstva

3x416x416  
neurónov



1x416x416  
neurónov

3x3x3



Každý neurón  
výstupnej vrstvy  
má spojenia na  
 $3 \times 3 \times 3 = 27$   
neurónov vstupnej  
vrstvy

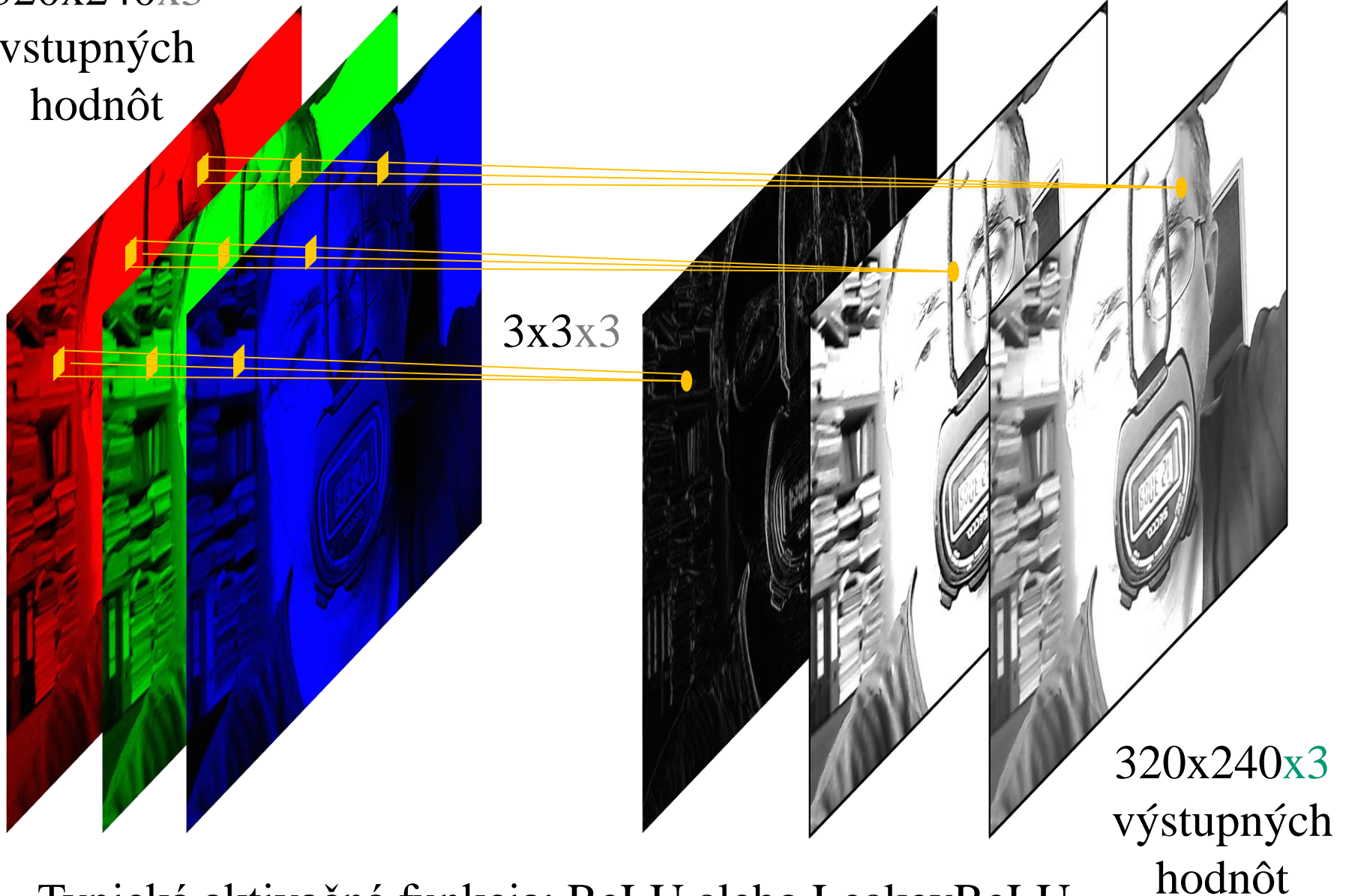
Váhy na spojeniach  
neuróny zdieľajú,  
t.j. celá vrstva má  
173056 neurónov,  
ale len 27  
parametrov,  
prípadne 28  
s biasom

[1998 LeCun, Bottou, Bengio & Haffner ]

# Blok konvolučných vrstiev

320x240x3

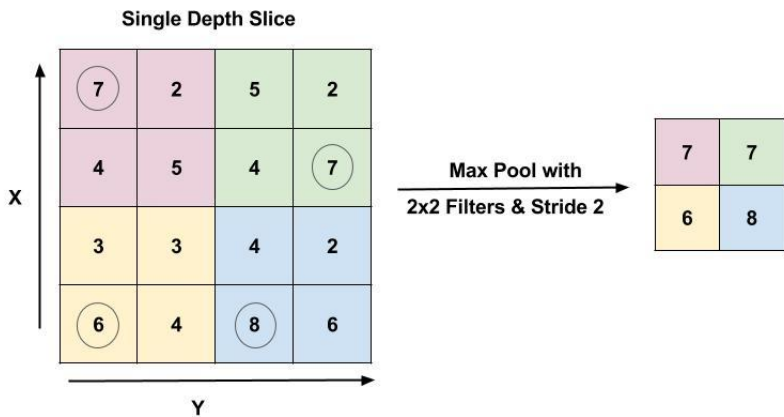
vstupných  
hodnôt



320x240x3  
výstupných  
hodnôt

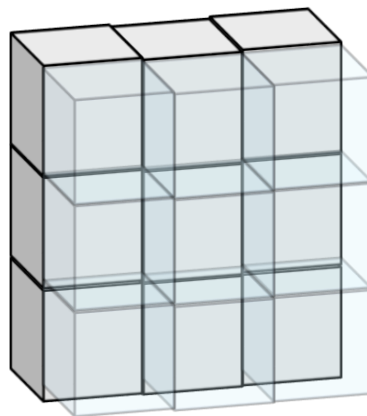
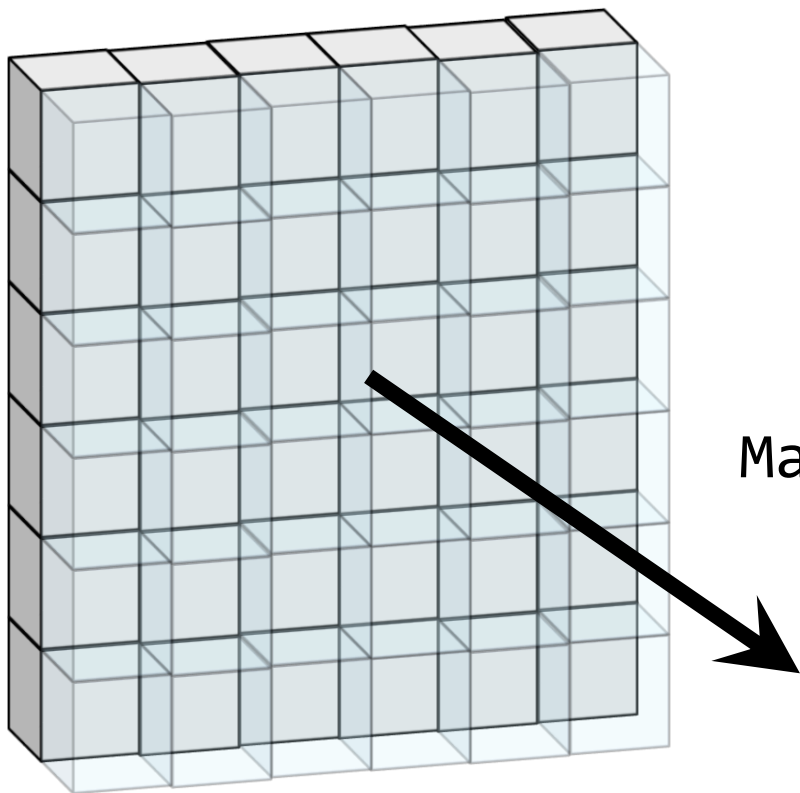
Typická aktivačná funkcia: ReLU alebo LeakeyReLU

# Redukcia dimenzie na báze maxima



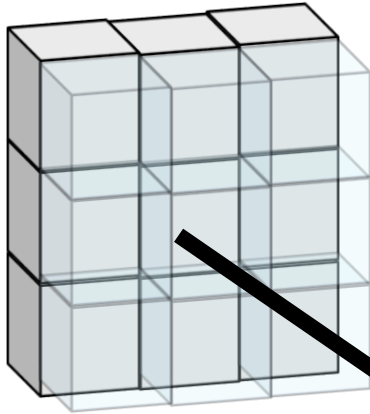
zlučovanie 2x2 pixelov s krokom 2  
a ich nahradenie maximom

MaxPooling2D 2x2 stride=2





# Espansione dimensione



Nearest Neighbor

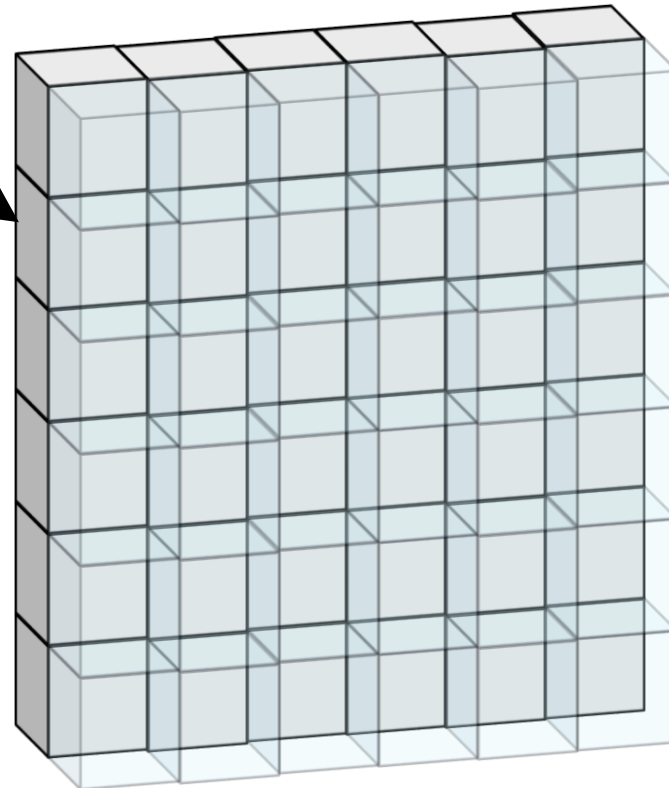
1	2
3	4



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

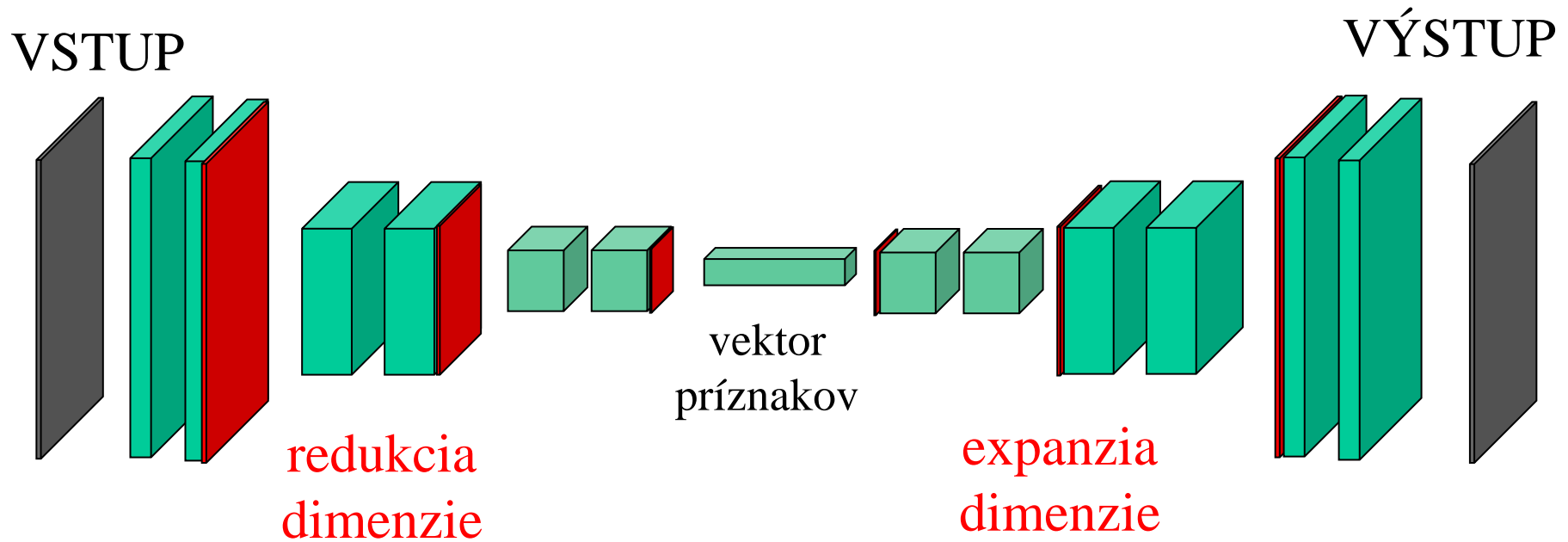
Input: 2 x 2

Output: 4 x 4



Upsampling2D 2x2 stride=2

# Konvolučný autokóder



bloky konvolučných vrstiev

chceme, aby:  
 $VSTUP = VÝSTUP$

# Autokóder

Vstup



```
array([[0.11400187, 0.47539777, 0.19979003, 0.03328802, 0.29702646,
0.6671412, 0.77884567, 0.98671937, 0., 0.7643514,
0.37023163, 1.1198556, 0.9364344, 0.27932528, 1.3840352,
1.1025171, 0., 1.127322, 0.6720804, 1.0088537,
0.91483283, 0., 1.3498058, 0.8023532, 0.13444054,
1.1076059, 0.805897, 0.4363817, 0.4396257, 0.,
0.5863744, 0.41566792, 0.22631842, 0.20689899, 0.28988916,
0.19635512, 0.92697734, 0.8330982, 0.8810159, 0.14809921,
0., 0.20924592, 0., 2.7036035, 1.7514778,
0.84079874, 1.6247051, 0., 0.25012553, 0.70244396,
0.6414403, 2.4568624, 1.4004446, 0., 1.3100656,
0., 0.5075829, 0.68791926, 0.65375787, 0.82646245,
0.4355916, 0., 0.19816275, 0., 0.,
0.4575198, 0.18170735, 0.12635085, 0.17334037, 0.4582858,
1.0187354, 0.75260663, 0., 0.4846586, 0.,
1.9696081, 1.12253, 0.8872602, 1.3111267, 0.,
0., 1.1081457, 0.5976082, 2.1433632, 1.2630261,
0., 1.4435332, 0., 0.43087044, 0.50078976,
0.85700417, 0.23156954, 0.3238153, 0.19322284, 0.23595949,
0., 0., 0.11734977, 0., 0.8126336,
1.2869604, 0.65106845, 1.012244, 0., 0.07893795,
0.16735056, 0.15048887, 2.1369095, 1.2226689, 0.,
1.0082622, 0., 0.9999559, 0.35816067, 0.4425221,
1.7548463, 0.36668733, 0.25854337, 0.35278222, 0.,
0.7470093, 0.42634767, 0.5120847, 0.24160625, 0.23943251,
0.61216664, 0.171287, 0.35395604], dtype=float32)
```

Výstup



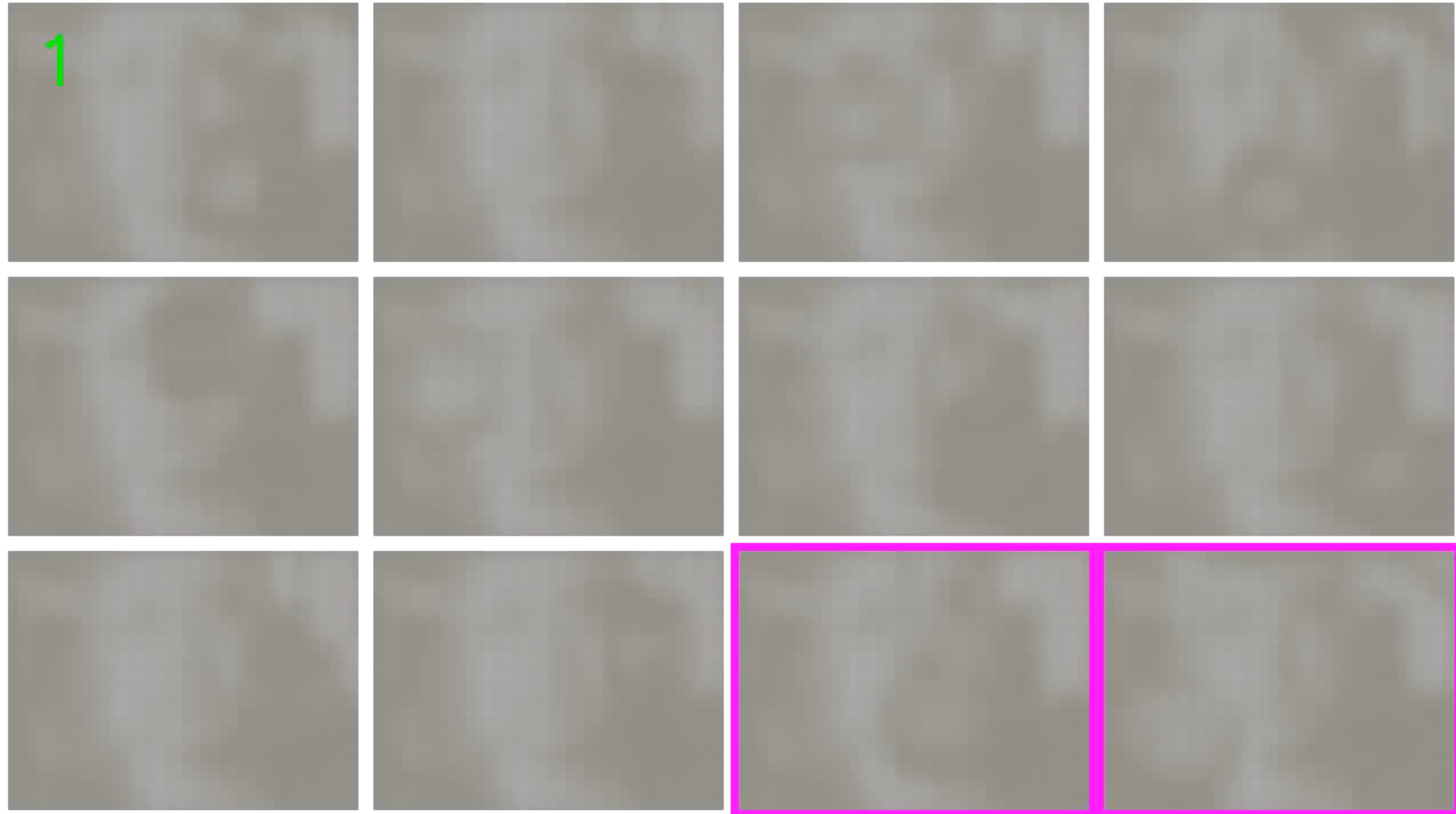
vektor príznakov

Na natréňovanie akejkoľvek neuronovej siete potrebujeme dostatočné množstvo príkladov – **sadu vzorových vstupov a výstupov**

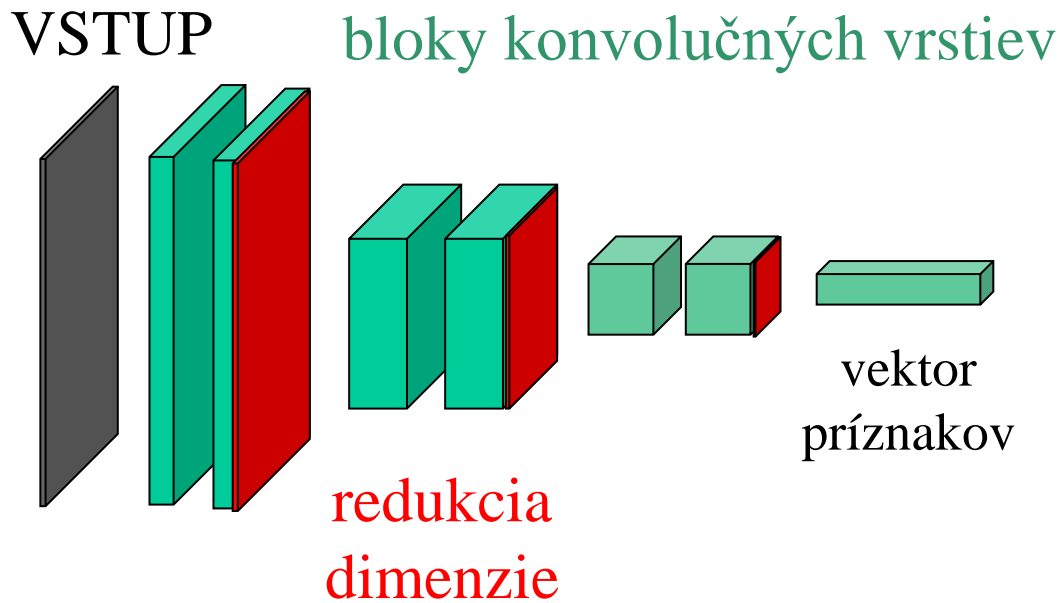


# Trénovanie autokódera

1



# Kóder



Kóder redukuje dimenziu dát,  
premieňa obraz na príznaky

# Dekóder

