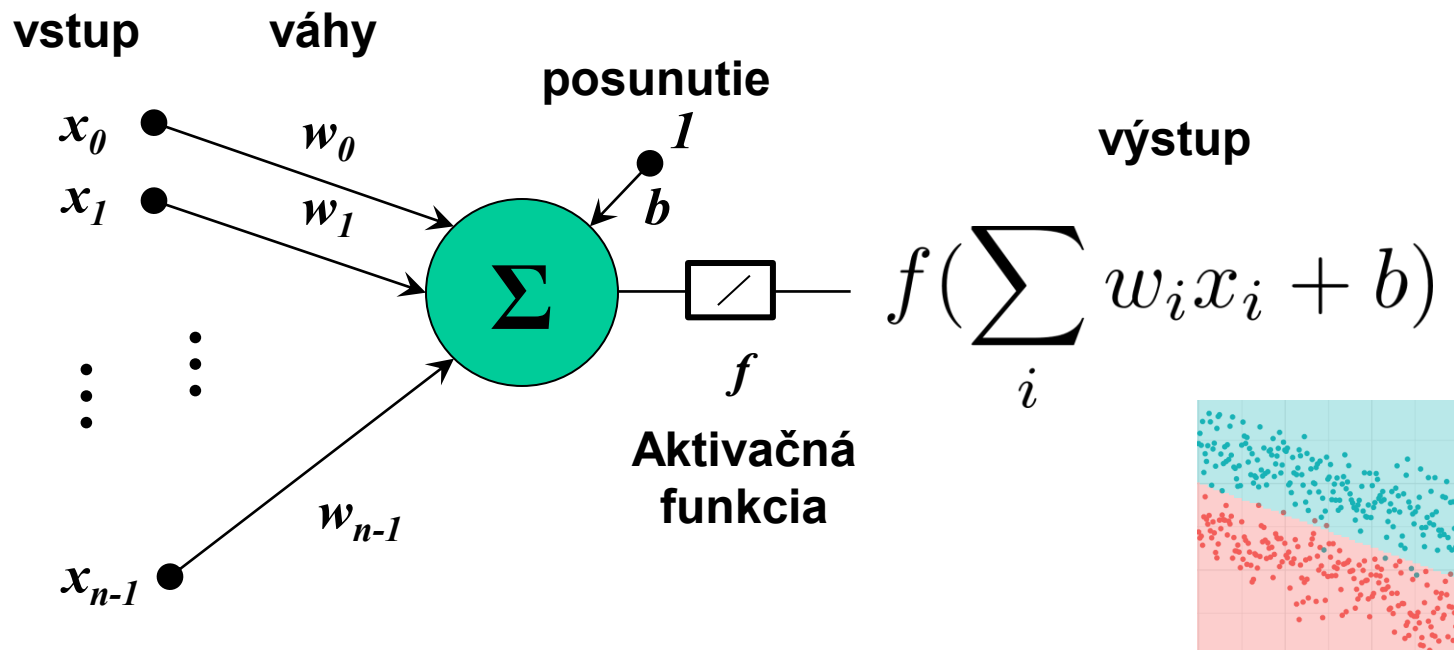


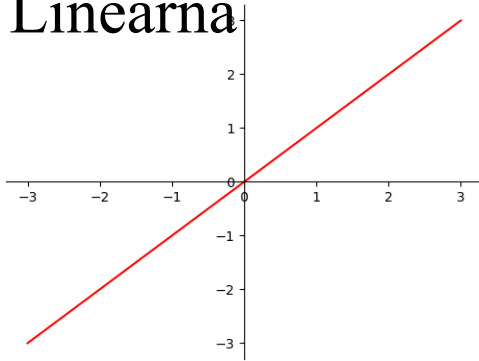
Neurón – Lineárny regressor



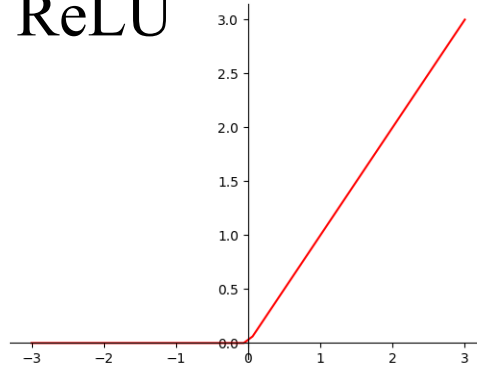
- Neurón počíta skalárny súčin vstupu s váhami, pripočíta posunutie a aplikuje aktivačnú funkciu
- Aktivačná funkcia môže byť: lineárna, sigmoida, hyperbolický tangens, ...

Úvod: Aktivačné funkcie

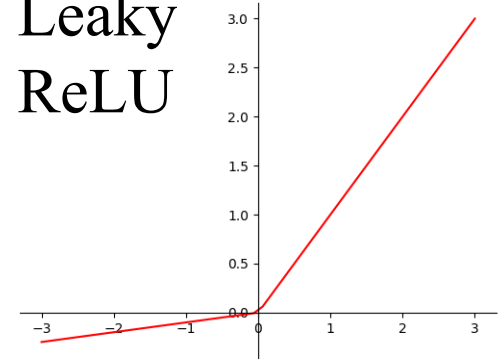
Lineárna



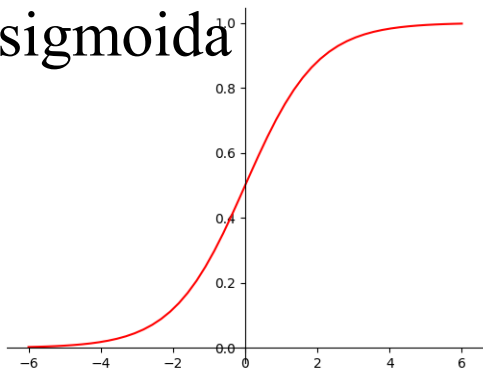
ReLU



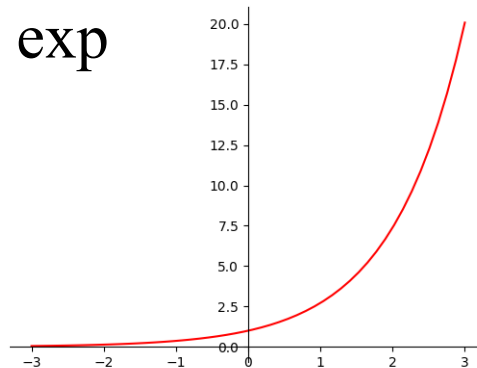
Leaky
ReLU



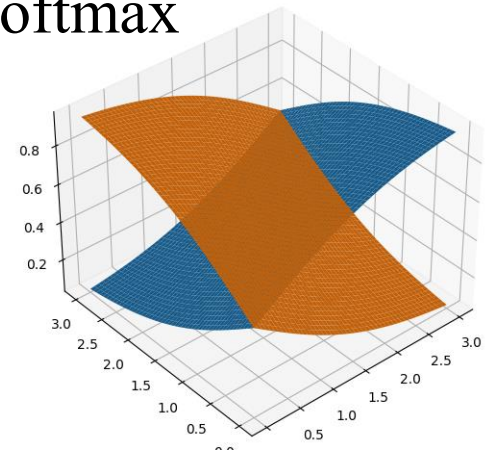
sigmoida



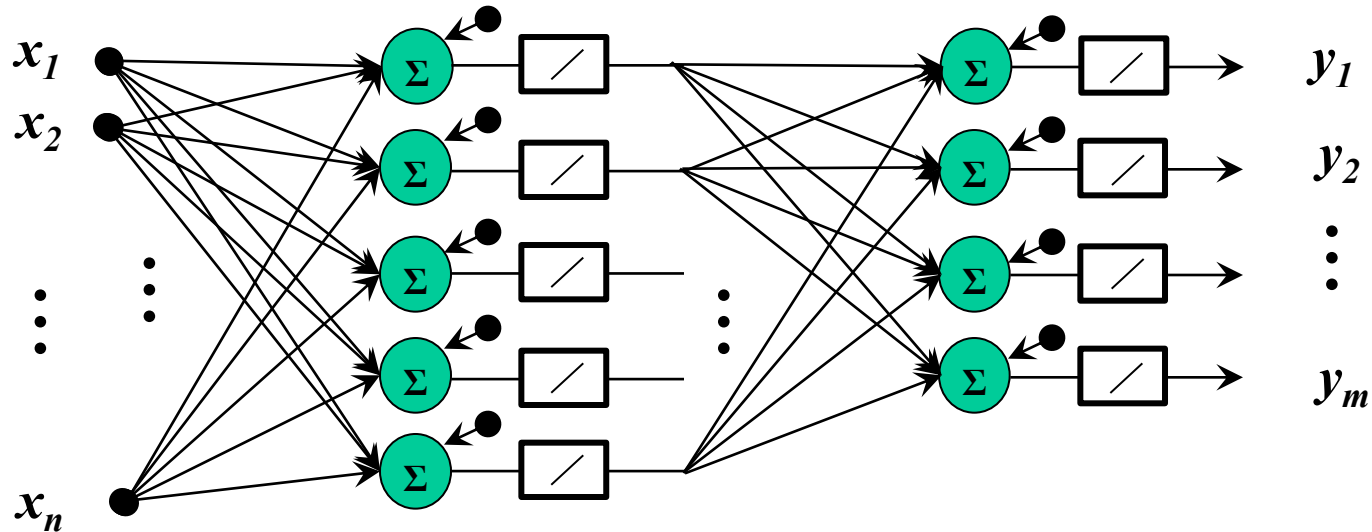
exp



softmax



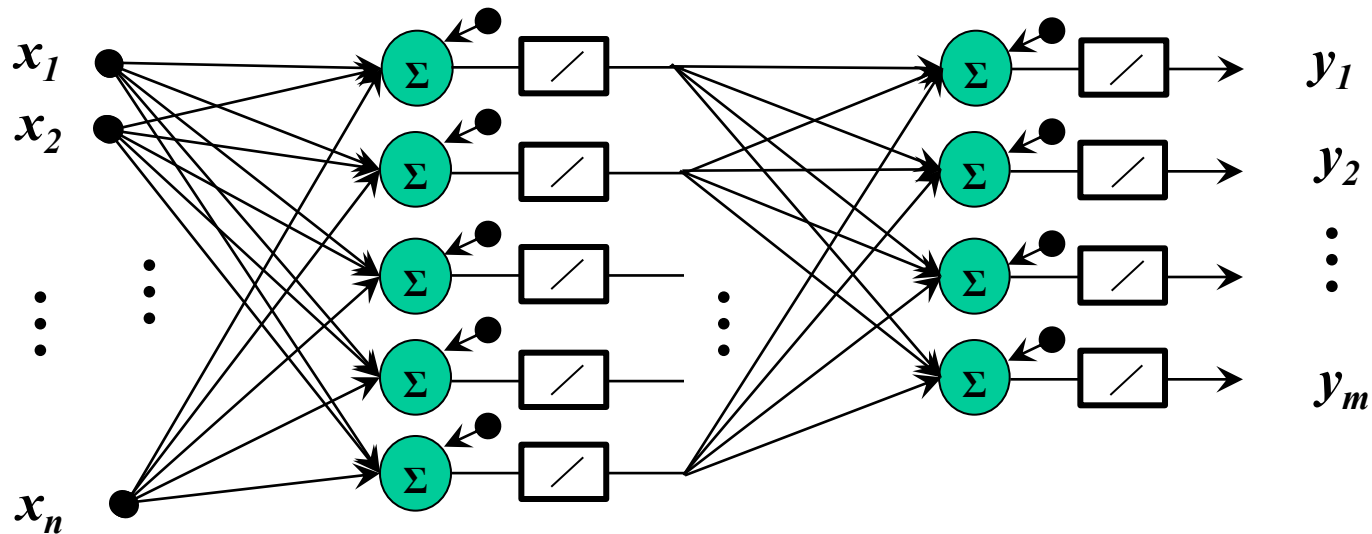
Perceptron



- jedna skrytá a jedna výstupná vrstva
- iba lineárna aktivácia
- možnosť trénovať len váhy výstupnej vrstvy

[1958 Rosenblatt]

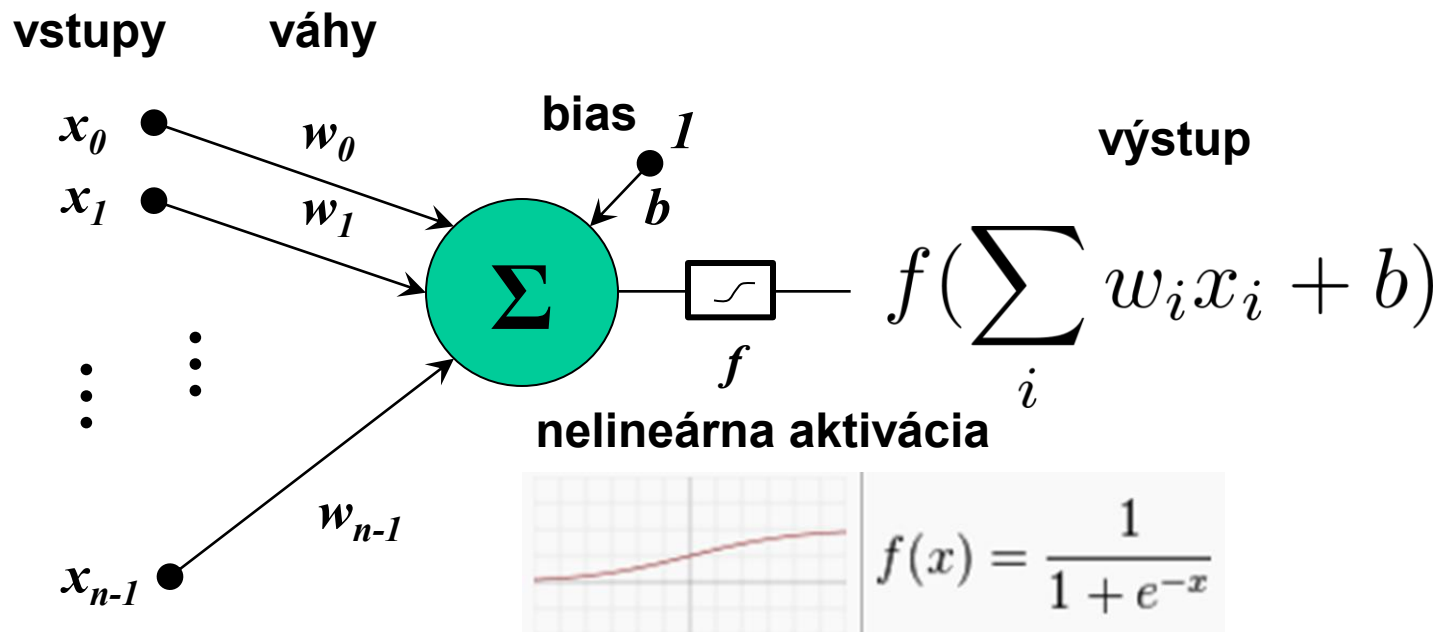
Perceptron



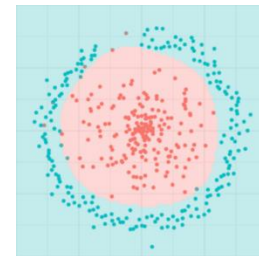
- analýza schopností perceptronu dokázala, že toho vie dosť málo
- neskôr sa však ukázalo, že jeho nevel'ká úprava situáciu dramaticky zmení

[1968 Minsky]

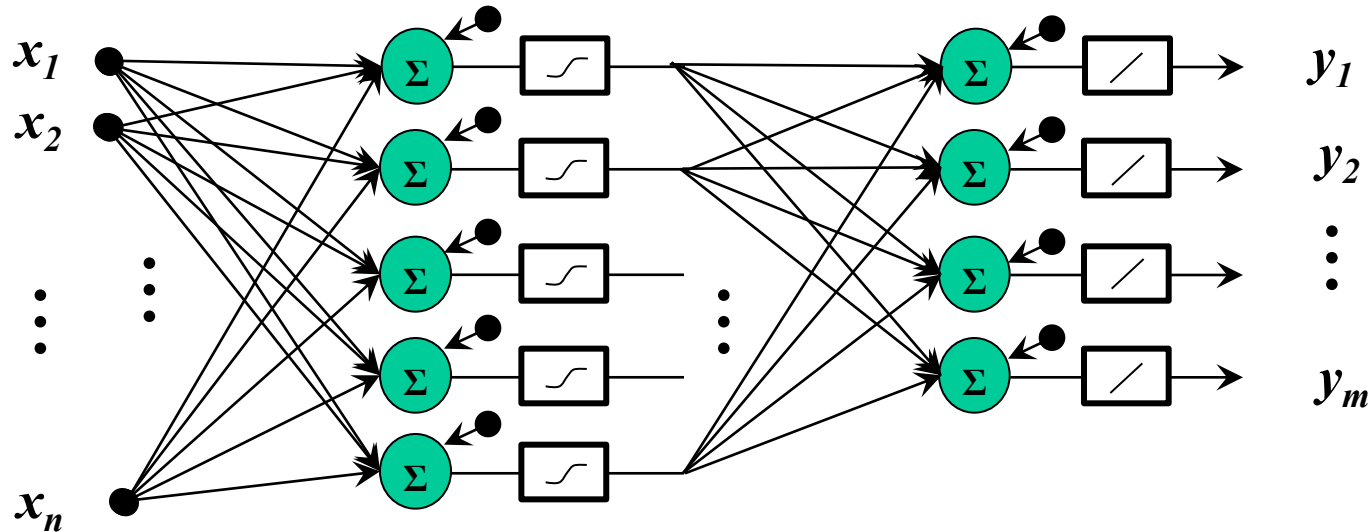
Neurón – Logistický regresor



- keď miesto lineárnej aktivácie použijeme logistickú funkciu (sigmoidu) alebo hyperbolický tangens, trénovanie jedného takéhoto neurónu zodpovedá logistickej regresii a potenciál neurónu byť stavebnou jednotkou siete sa dramaticky zlepší



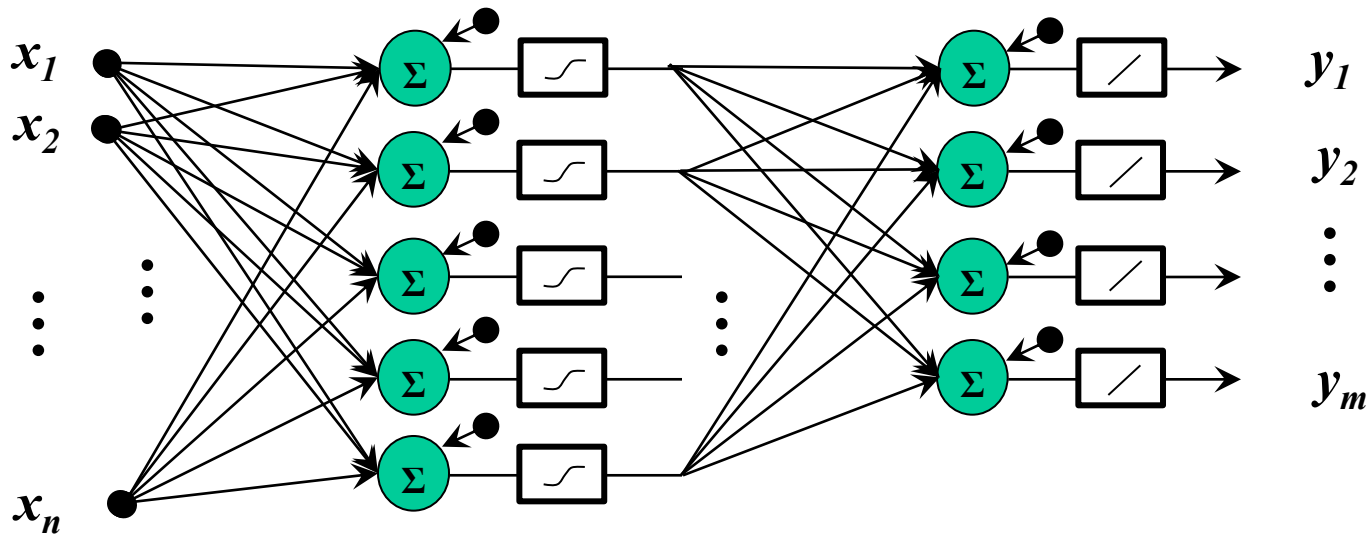
Perceptron + nelinearita + BP



- Algoritmus spätného šírenia (Back propagation) – pomocou pravidla derivácie zloženej funkcie [Leibnitz 1676] na trénovanie algoritmom spätnej propagácie (back propagation)

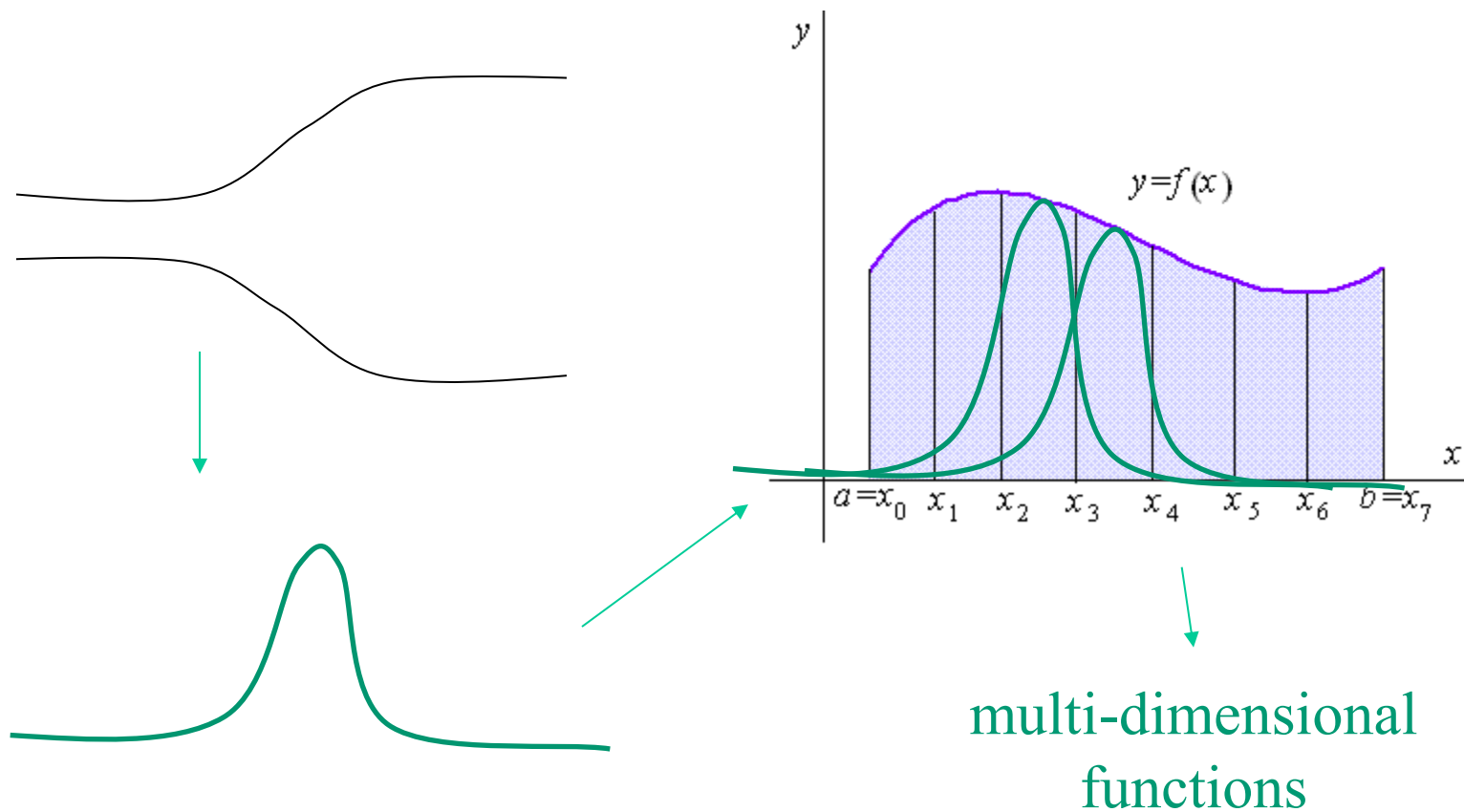
[1986 Rumelhart, Hinton & Williams]

Univerzálny aproximátor



- matematicky bolo dokázané, že perceptron sa teoreticky dokáže s ľubovoľnou presnosťou naučiť akúkoľvek rovnomerne spojitú funkciu [1989 Cybenko] [1989 Hornik]
- prakticky sa to však podarilo uskutočniť len pre vstupy pomerne malej dimenzie, pre naše obrazové dáta s dimenziou $519168 = 416 \times 416 \times 3$ je to nepoužiteľné

Universal approximation



Kernel

240x320x1



1x1x1

1.5
weight

0.15
bias

240x320x1



Kernel

240x320x1



1x1x1

1.5
weight

0

bias

240x320x1



weight changes contrast

Kernel

240x320x1



1x1x1

1.0

weight

0.15

bias

240x320x1



bias changes brightness

Convolutional Layer

320x240x1
input
values

input
dimension
76800



1x1x1
weight



1 convolutional layer
320 x 240 neurons

neurons share
1 weight and
1 bias

the layer contains
76800 neurons
but has 2
parameters only

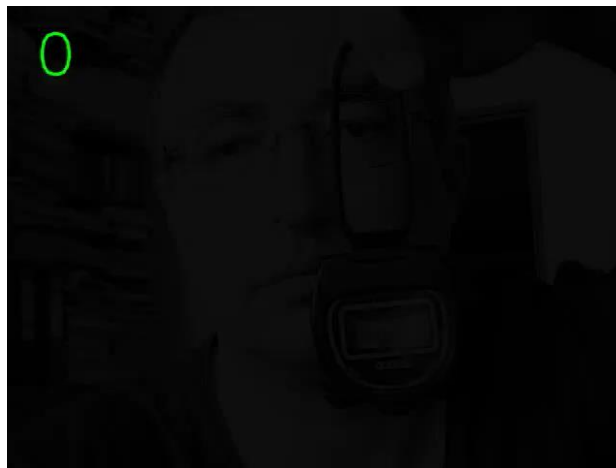
activation: linear

320x240x1
output
values

Training



sample input



sample output

1.5 contrast
weight

0.15 brightness
bias

Kernel

240x320x3



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	

weights

+0

bias

240x320x1

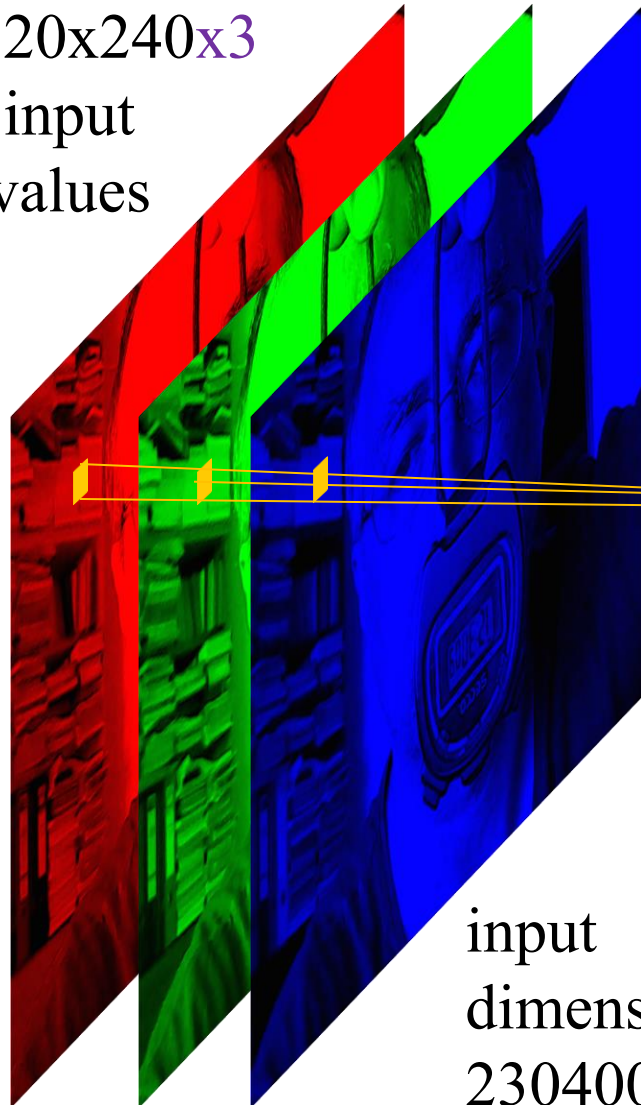


Sobel (vertical) kernel provides vertical edges

Convolutional Layer

320x240x3

input
values



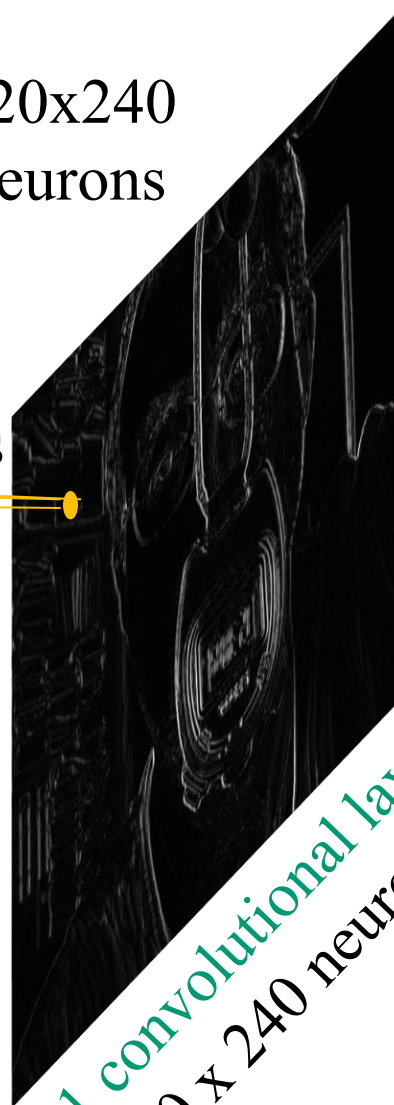
input
dimension
230400

activation: linear

320x240
neurons

3x3x3
weights

1 bias



1 convolutional layer
320 x 240 neurons

neurons share
27 weights and
1 bias

the layer contains
76800 neurons
and 28 parameters

320x240x1
output
values

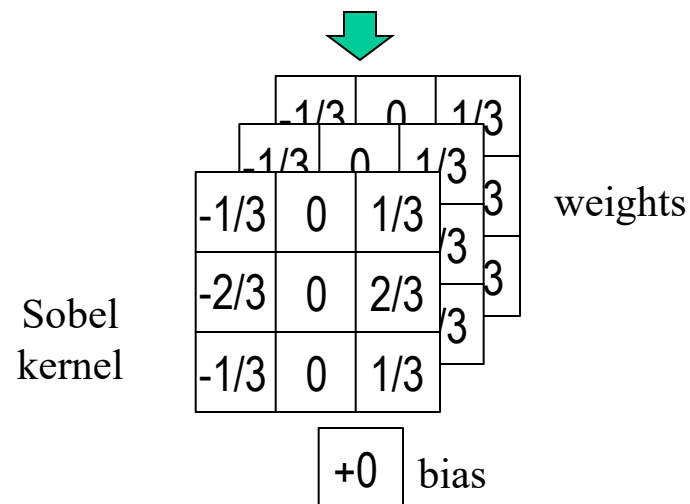
Training



sample input



sample output



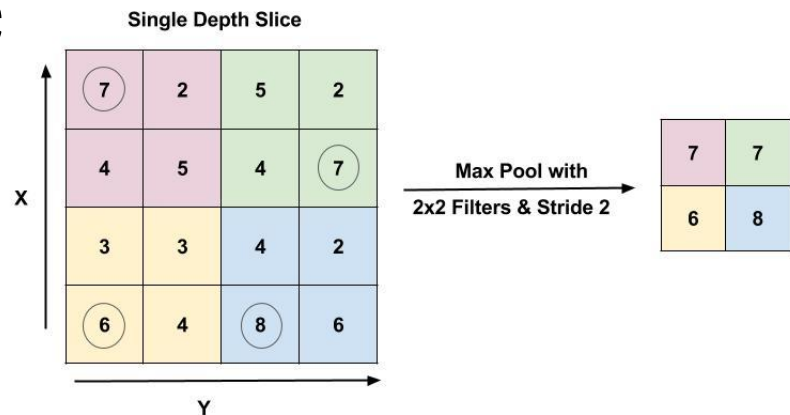
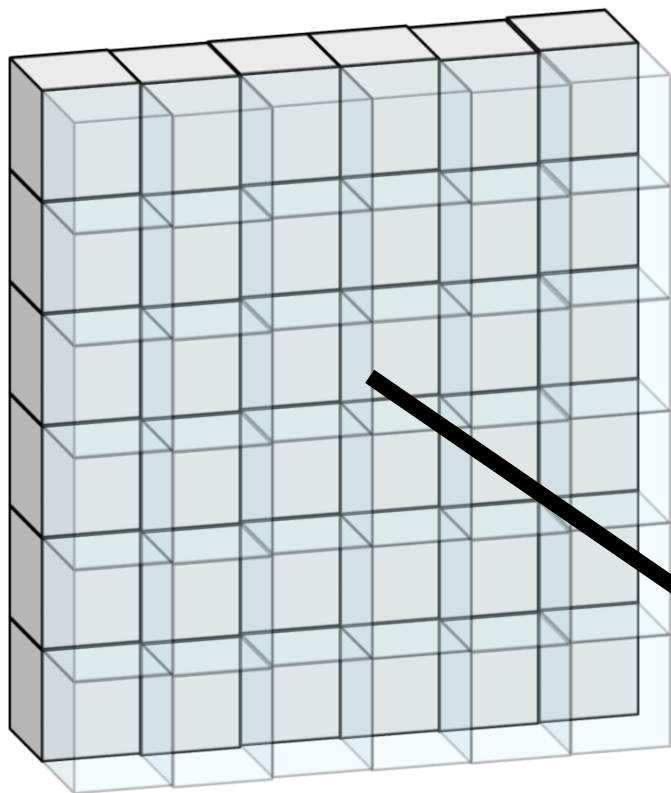
Blok konvolučných vrstiev

320x240x3
vstupných
hodnôt

3x3x3

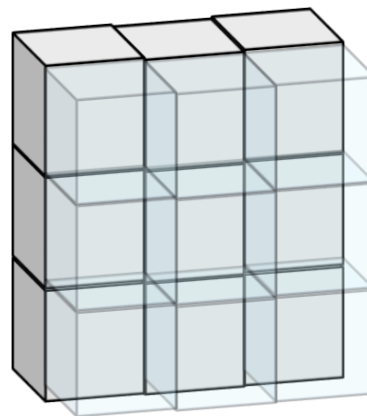
320x240x3
výstupných
hodnôt

Redukcia dimenzie na báze maxima

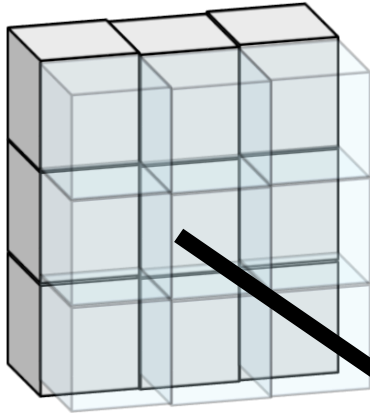


zlučovanie 2x2 pixelov s krokom 2
a ich nahradenie maximom

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

