



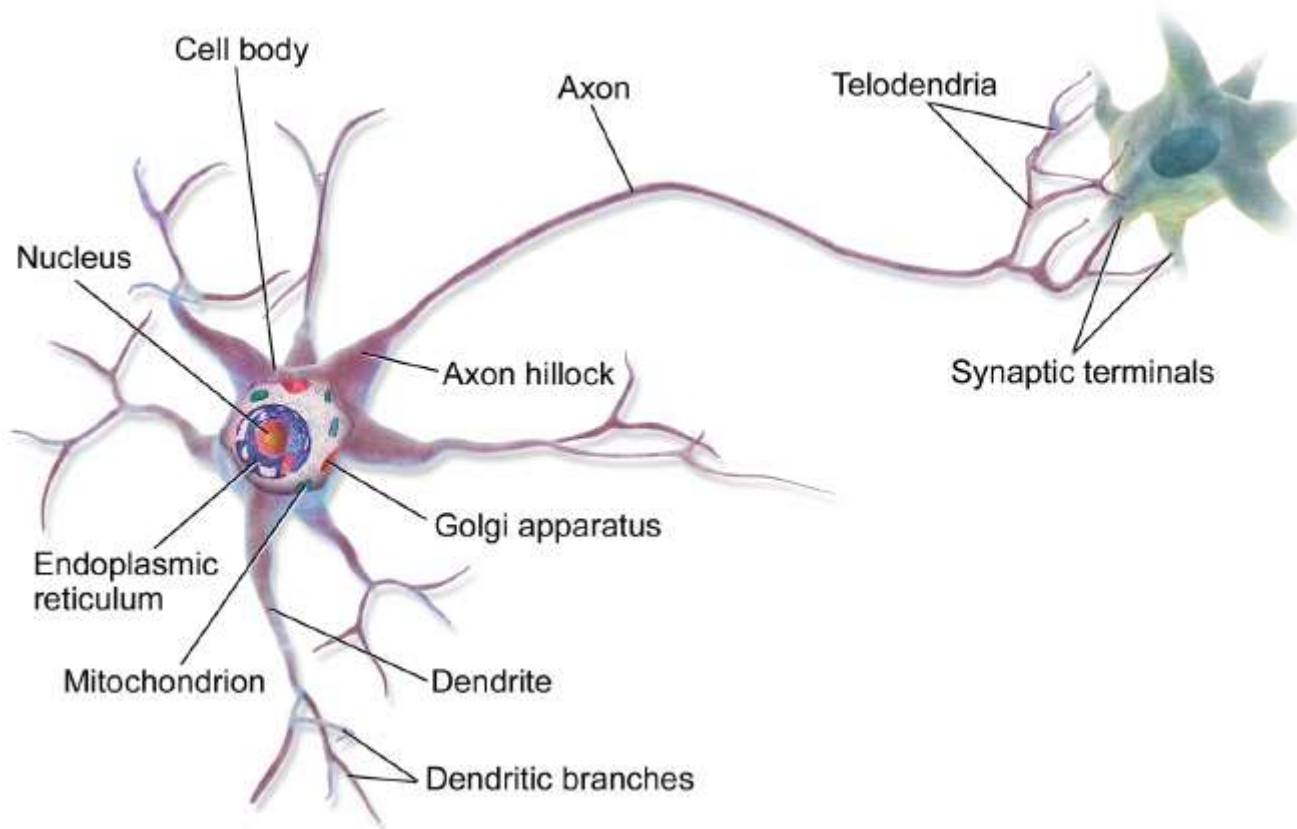
# Nevronske mreže

SSS, oktober 2023

Simon Čopar



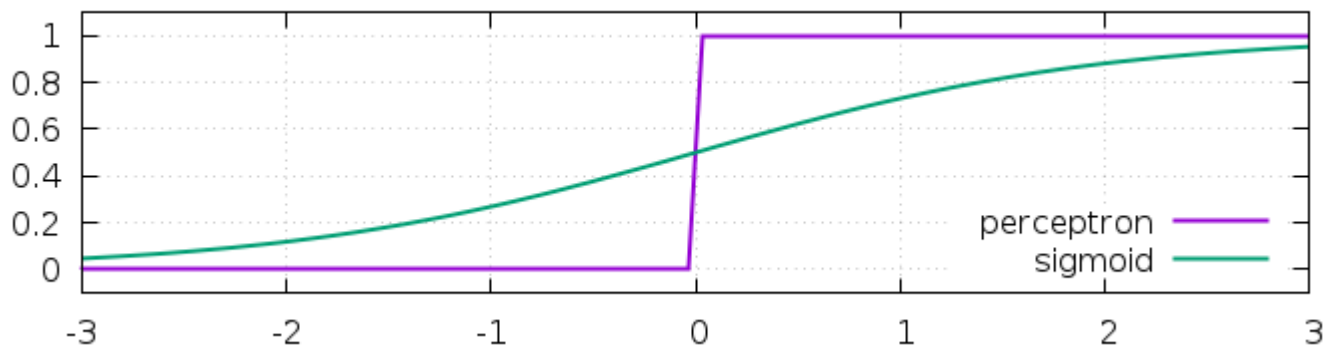
# Motivacija: kako deluje živčevje?





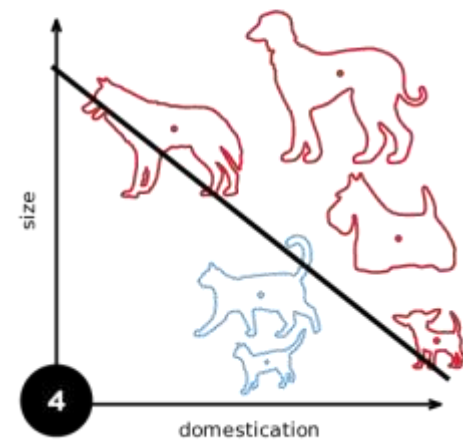
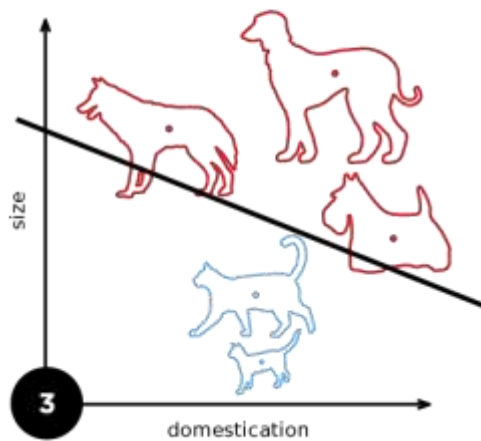
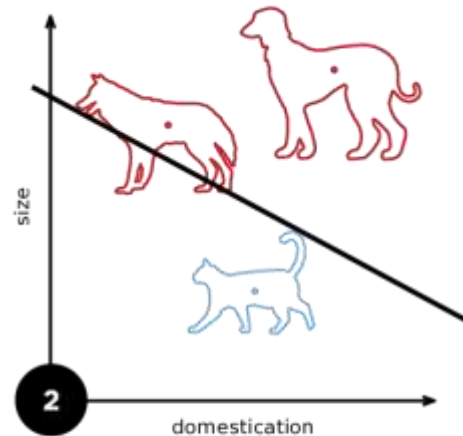
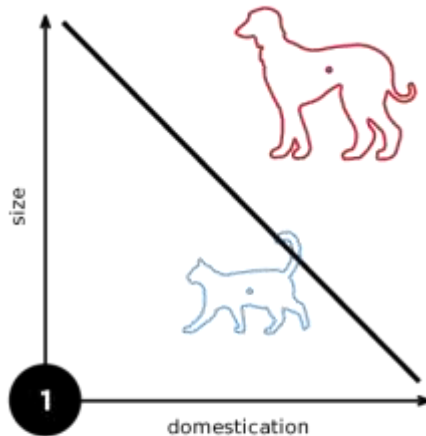
# Nevron: mehka binarna odločitev

- ↳ Osnovna enota enostavna: na podlagi intenzitet signalov na vseh vhodih nevron vrne impulz, ali pa ne
- ↳ Ostra stopnica nepraktična za računanje: zvezna je boljša
- ↳ Prednost zvezne: imamo naklon, torej vemo v kateri smeri popraviti, da se kaj **naučimo!**





# Več vprašanj: bolj uporabni odgovori





# Plast nevronov

↳ Več vprašanj: Ima ušesa? Je kosmat? Je maček? Je pes? Je avto? Ima veliko vogalov? Je črka?

↳ **Ideja:** iz več vprašanj lahko izvemo več

↳ **Kaj so sploh parametri?**

$$y = f(b + w_1x_1 + w_2x_2 + \dots)$$

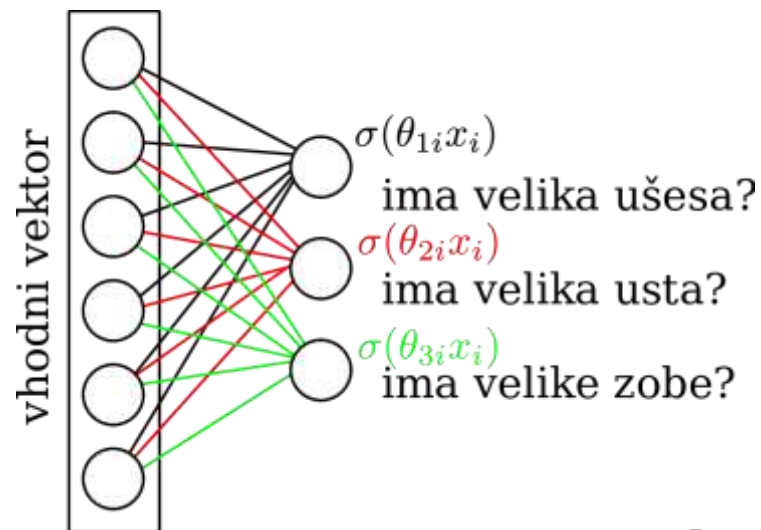
aktivacijska funkcija (nelinearna – recimo gladka stopnica)

vhodni podatki

bias (premik kje se stopnica aktivira)

uteži (kako pomembni so posamezni vhodi)

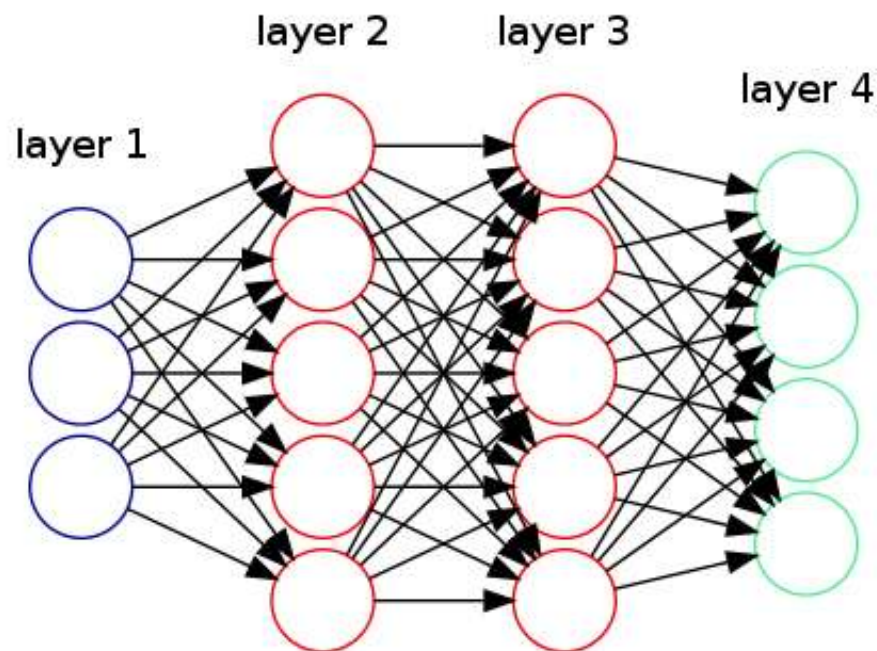
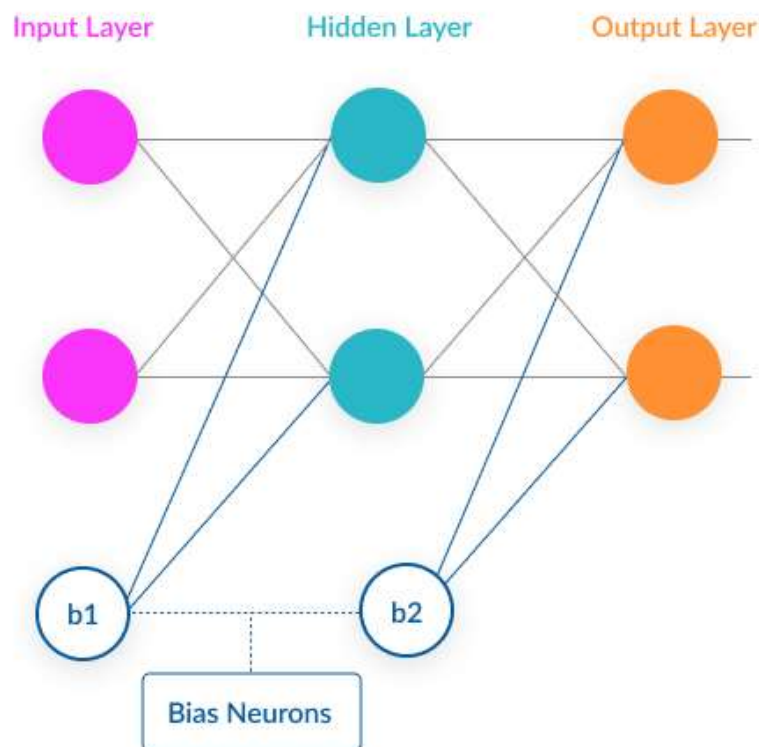
↳ **Problem:** kje dobimo te parametre?





# Skrite plasti

- ↳ **Hierarhija:** odgovor gradimo po korakih
- ↳ Rezultat prve plasti so značilke (features) za naslednjo plast
- ↳ **Upamo:** vmesna plast se sama nauči, kaj so pametne lastnosti





## Izrek o univerzalni aproksimaciji

↳ S kompozicijo dovolj velike količine **nelinearnih** preslikav lahko vsako funkcijo aproksimiramo poljubno natančno

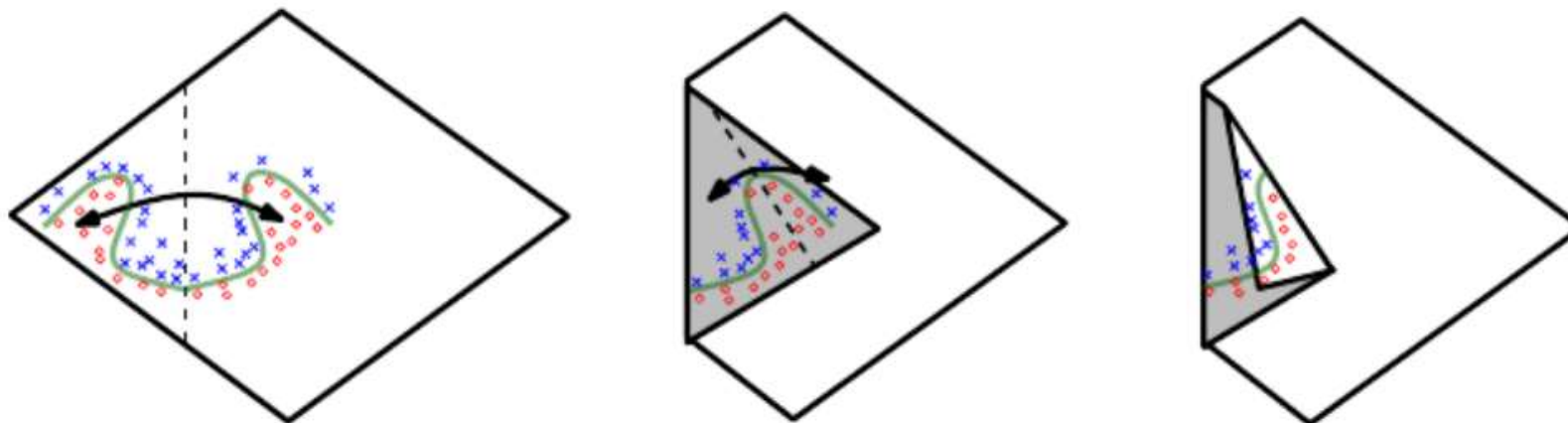
↳ Če je **linearna**, kolapsira na eno preslikavo:

$$\mathbf{y} = \mathbf{W}^{(3)}\mathbf{W}^{(2)}\mathbf{W}^{(1)}\mathbf{x} = \mathbf{W}'\mathbf{x}$$

Slabo – nič ne pridobimo

↳ „Deep Neural Network“ – veliko plasti

↳ Težje minimizirat, potrebuje več podatkov, numerični problemi

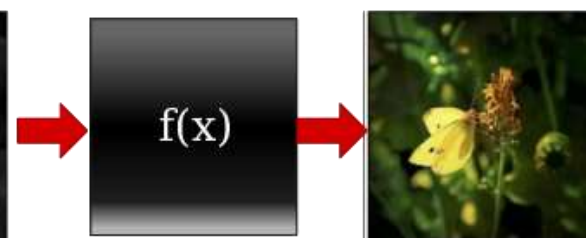
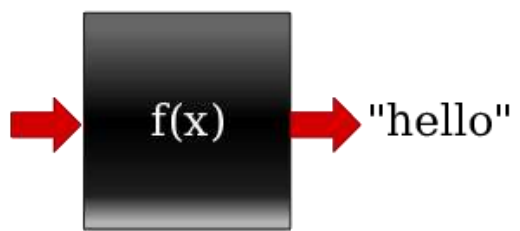
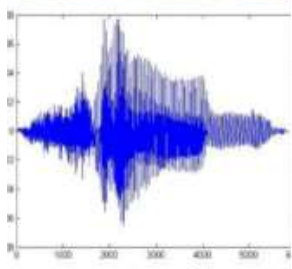
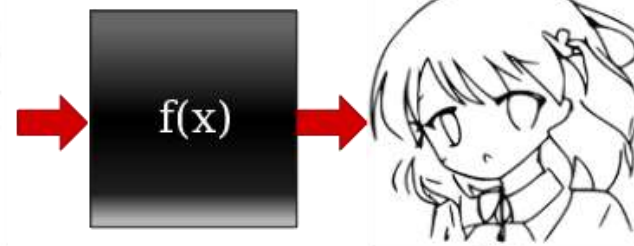
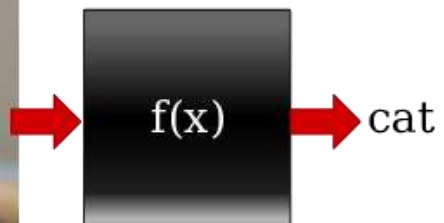




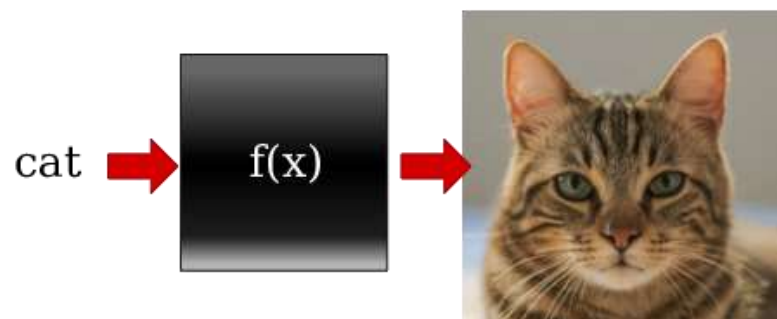


# Kaj mislimo s pojmom univerzalna funkcija?

- ↳ Slikamo katerikoli objekt v katerikoli rezultat
- ↳ Matematično: vse je lahko „vektor“



**Bolj ambiciozno:**

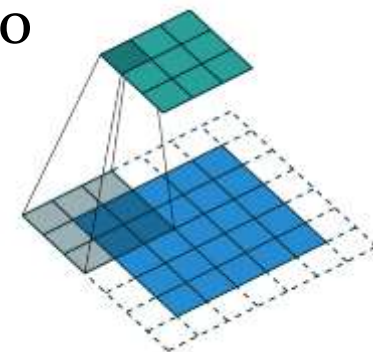
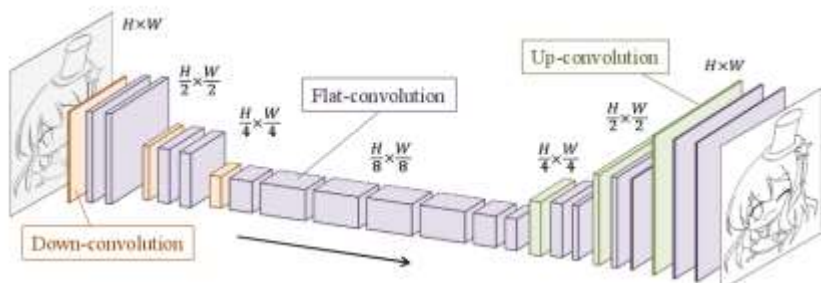






# Različne vrste nevronske mreže

- ↳ Gosto povezane – vsak nevron je povezan z vsemi iz prejšnje plasti
- ↳ Konvolucijske: za slikovni material – povezani samo sosednji deli slike (lokalnost!)




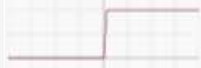







- ↳ Za jezikovne vsebine in generiranje vsebin (slik, videov, besedil) v zadnjem času najbolj primerni Transformerji (recimo GPT-3, GPT-4, BERT, Dall-E, Midjourney,...)

- ↳ Besede in njihovi položaji v besedilu so tudi števila
- ↳ Prav tako segmenti slik in njihove lokacije
- ↳ Deluje na principu pozornosti: pozornost oziroma asociacija v matematičnem smislu je tudi opisljiva z matrikami (če beseda ali fraza spomni na nek koncept, je v ustreznem stolpcu in vrstici velika vrednost).





# Aktivacijske funkcije

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$



# Cenovna funkcija

↳ Za učenje moramo vedeti pravilni odgovor

↳ Kako sploh vemo, kaj je prav?

↳ Minimizacija funkcije izgube oz. cenovne funkcije („Loss function“)

↳ Primer: vsota kvadratov odmikov (kot pri fitanju!)

$$J(\mathbf{y}, \mathbf{t}) = \sum_i (y_i - t_i)^2$$

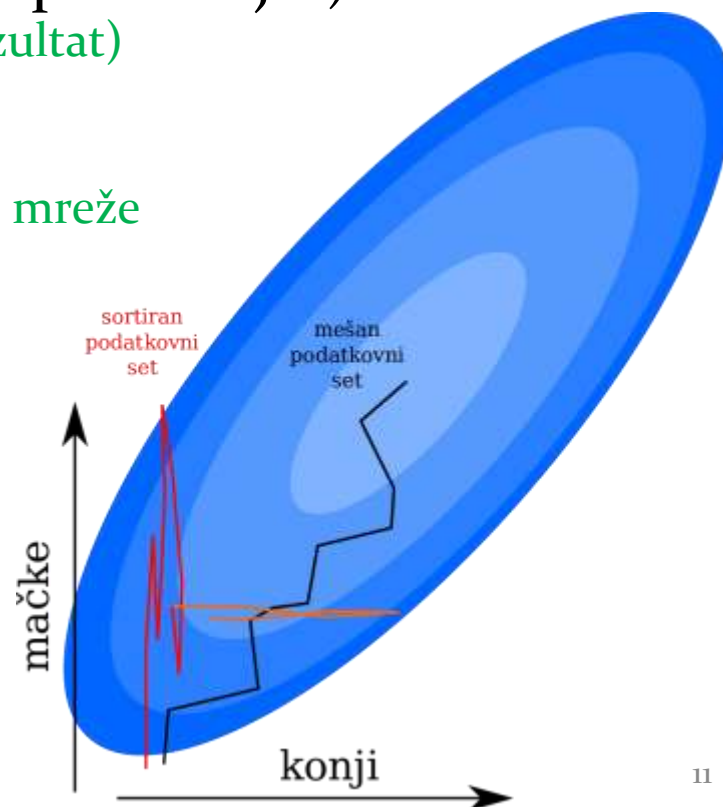
Target (pravi rezultat)

Izhod iz nevronske mreže

MSE (Mean squared error) – najmanjši kvadrati

↳ Kako pa „bližamo“?

Kako znižamo cenovno funkcijo?





# Vzratna propagacija napake

↳ Optimiziramo kompozitum: gradient po verižnem pravilu

$$\mathbf{y} = f \left( W^{(3)} f \left( W^{(2)} f \left( W^{(1)} \mathbf{x} \right) \right) \right)$$

$\tilde{\mathbf{y}}^{(2)}$  izhod prejšnje plasti  
 $\mathbf{y}^{(2)}$  po aktivaciji

$$\frac{\partial J}{\partial W^{(3)}} = \underbrace{(\nabla J)(f'(W^{(3)} \mathbf{y}^{(2)}))}_{\text{red}} \otimes \mathbf{y}^{(2)}$$

$$\frac{\partial J}{\partial W^{(2)}} = \left( \underbrace{(\nabla J)(f'(W^{(3)} \mathbf{y}^{(2)}))}_{\text{red}} \right) \underbrace{W^{(3)}(f'(W^{(2)} \mathbf{y}^{(1)}))}_{\text{green}} \otimes \mathbf{y}^{(1)}$$

$$\frac{\partial J}{\partial W^{(1)}} = \left( \underbrace{(\nabla J)(f'(W^{(3)} \mathbf{y}^{(2)}))}_{\text{red}} \right) \underbrace{W^{(3)}(f'(W^{(2)} \mathbf{y}^{(1)}))}_{\text{green}} \underbrace{W^{(2)}(f'(W^{(1)} \mathbf{x}))}_{\text{blue}} \otimes \mathbf{x}$$

↳ Kaj pa to pomeni **v vsakdanjem jeziku**?

- ↳ Zadnja plast malo spremeni uteži v taki smeri, da bi bila napaka manjša
- ↳ Zadnja plast pove predzadnji plasti, kam bi se njihov izhod moral spremeniti, da bi bil rezultat boljši
- ↳ Predzadnja plast spremeni uteži v taki smeri, da se približa željam zadnje plasti
- ↳ Predzadnja plast pove predhodnici, kam bi se njihov izhod moral spremeniti...

# Postanite nevronska mreža



↳ Udeleženci boste postali živčni sistem „roomba“ sesalca (interaktivna delavnica)

↳ Vhodna plast: „oči“ povedo, ali vidijo zid, ali ne

↳ Skrita plast: vi samo sledite temu kar vam daje prva plast

↳ Izhodna plast: pogon levega kolesa in pogon desnega kolesa

