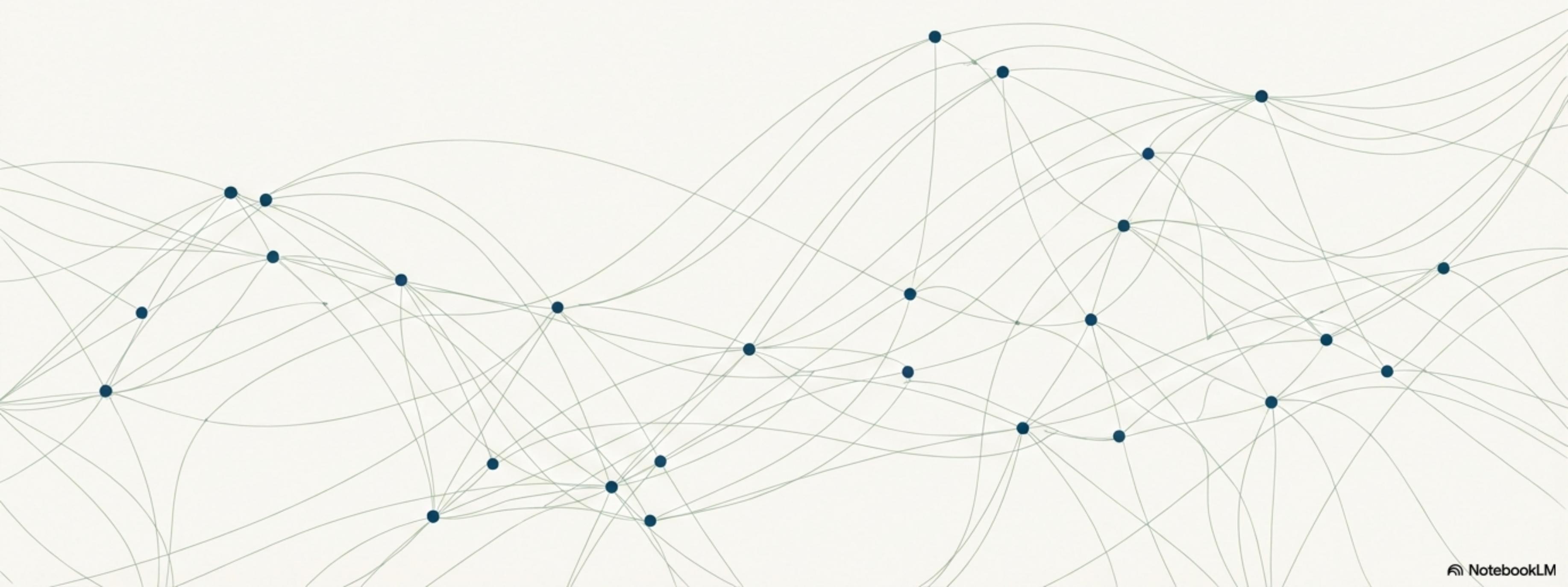


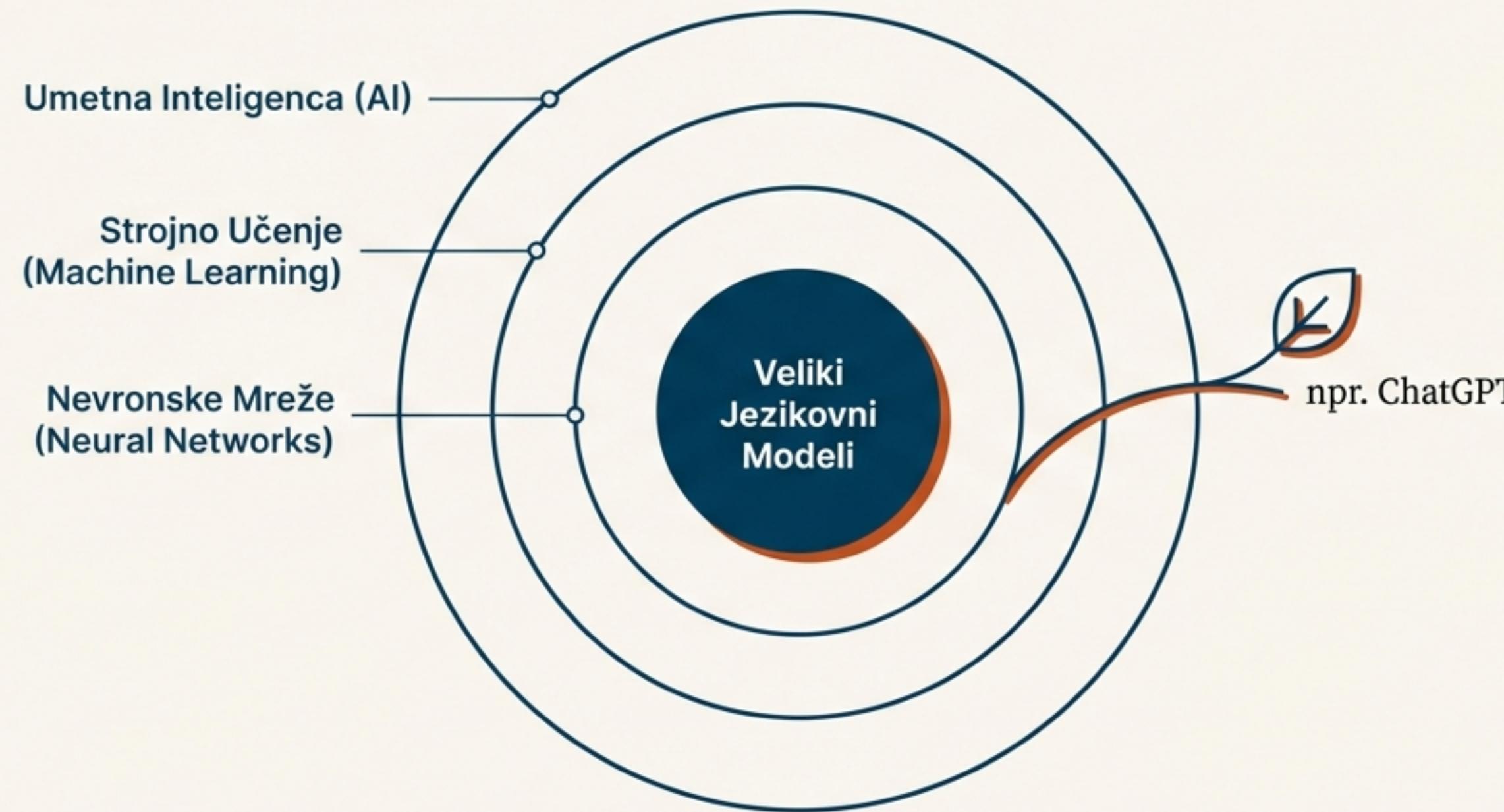
Demistificiranje Nevronskih Mrež

Od Širšega Koncepta Umetne Inteligence do Mehanike Učenja



Kje v svetu umetne inteligence se nahajajo nevronske mreže?

Umetna inteligenco je zelo širok pojem. Evropska unija jo opredeljuje kot "sposobnost stroja, da posnema človeške sposobnosti, kot so logično razmišljanje, učenje, načrtovanje in ustvarjalnost."



Čeprav danes večina ob omembi AI najprej pomisli na ChatGPT, so ti v resnici le ena izmed podkategorij nevronskih mrež, te pa le ena izmed podkategorij umetne inteligence.

Osnovna ideja: Učenje iz podatkov za napovedovanje prihodnosti

Nevronska mreža se iz obstoječih podatkov nauči, kako so izhodni podatki (npr. D, E) odvisni od vhodnih (npr. A, B, C). Ko se tega nauči, zna za nove vhode napovedati pripadajoče izhode.

Primer 1: Zdravstvo



Vhod (Input)

Višina, teža, krvni tlak, krvna slika, kadilec (Da/Ne)...



[Nevronska Mreža]



Izhod (Output)

Verjetnost za določeno bolezen.

Primer 2: Prometna varnost



Vhod (Input)

Gostota prometa, vidljivost, signalizacija, geometrija ceste...



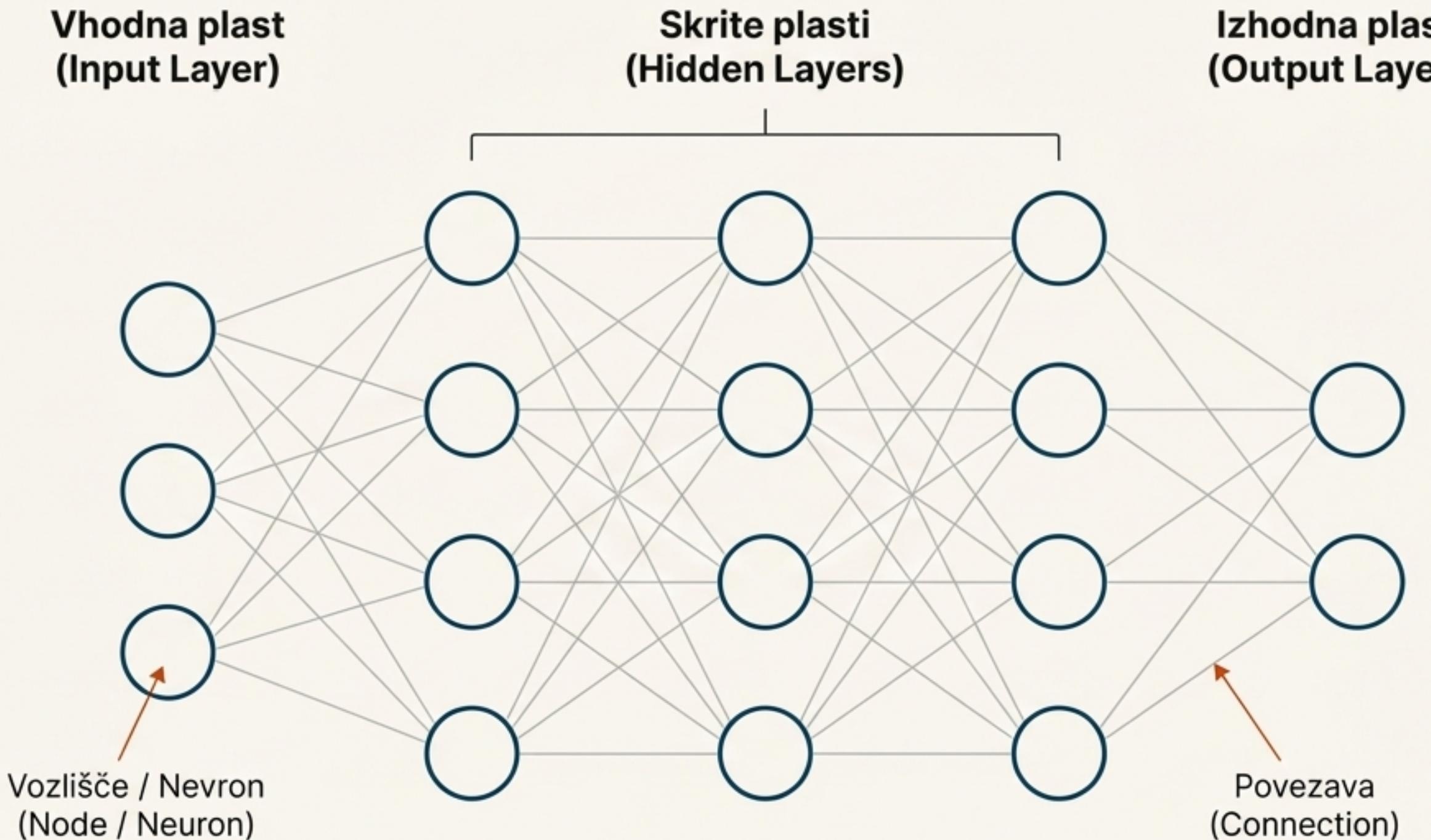
[Nevronska Mreža]



Izhod (Output)

Ocenjena nevarnost križišča (stevilo nesreč).

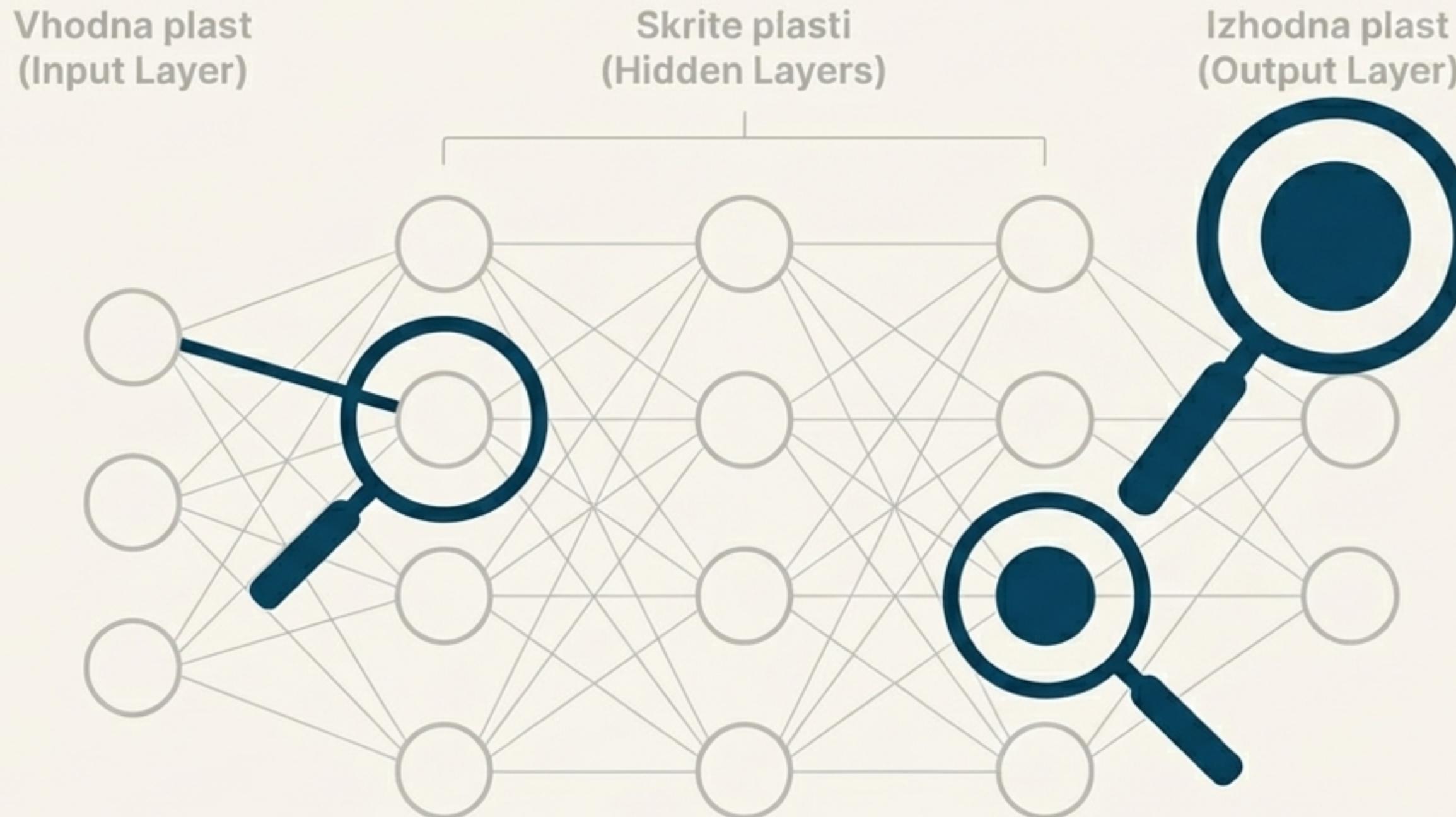
Anatomija nevronske mreže: Gradniki sistema



Ključna pravila

- **Vhodna plast:** Število vozlišč = število vhodnih (neodvisnih) podatkov.
- **Izhodna plast:** Število vozlišč = število izhodnih (odvisnih) podatkov.
- **Skrite plasti:** Število plasti in vozlišč je poljubno in odvisno od zahtevnosti problema.

Parametri mreže: 'Gumbi', ki se jih mreža nauči prilagajati



Utež (w)

Predstavlja moč povezave med dvema nevronoma.

Odmik (b)

Deluje kot prag, ki ga mora vsota vhodnih signalov preseči.

Na začetku so uteži (w) in odmiki (b) določeni naključno. Celoten cilj učenja je sistematično poiskati prave vrednosti teh parametrov.

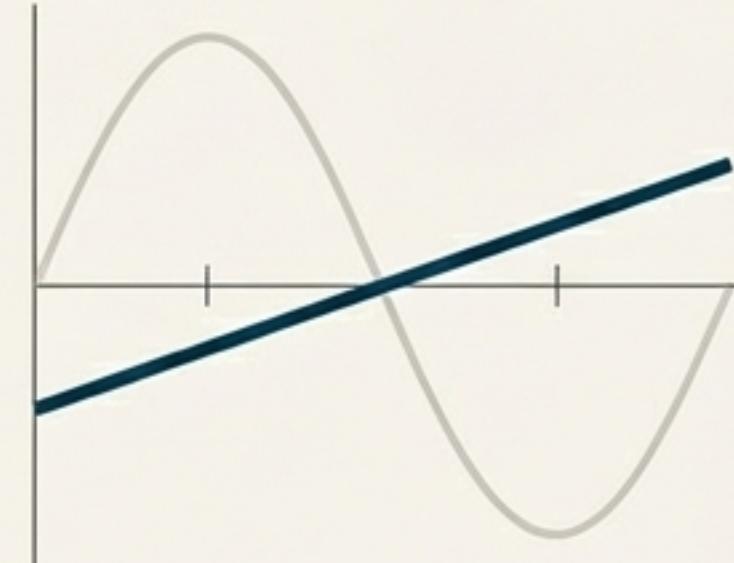
Problem linearnosti in rešitev: Aktivacijska funkcija

Del 1: Problem

Zakaj skrite plasti same po sebi niso dovolj?

Linearna kombinacija linearnih kombinacij je zopet le linearja kombinacija. Brez dodatnega koraka bi bila celotna mreža, ne glede na število plasti, le zapleten linearni model.

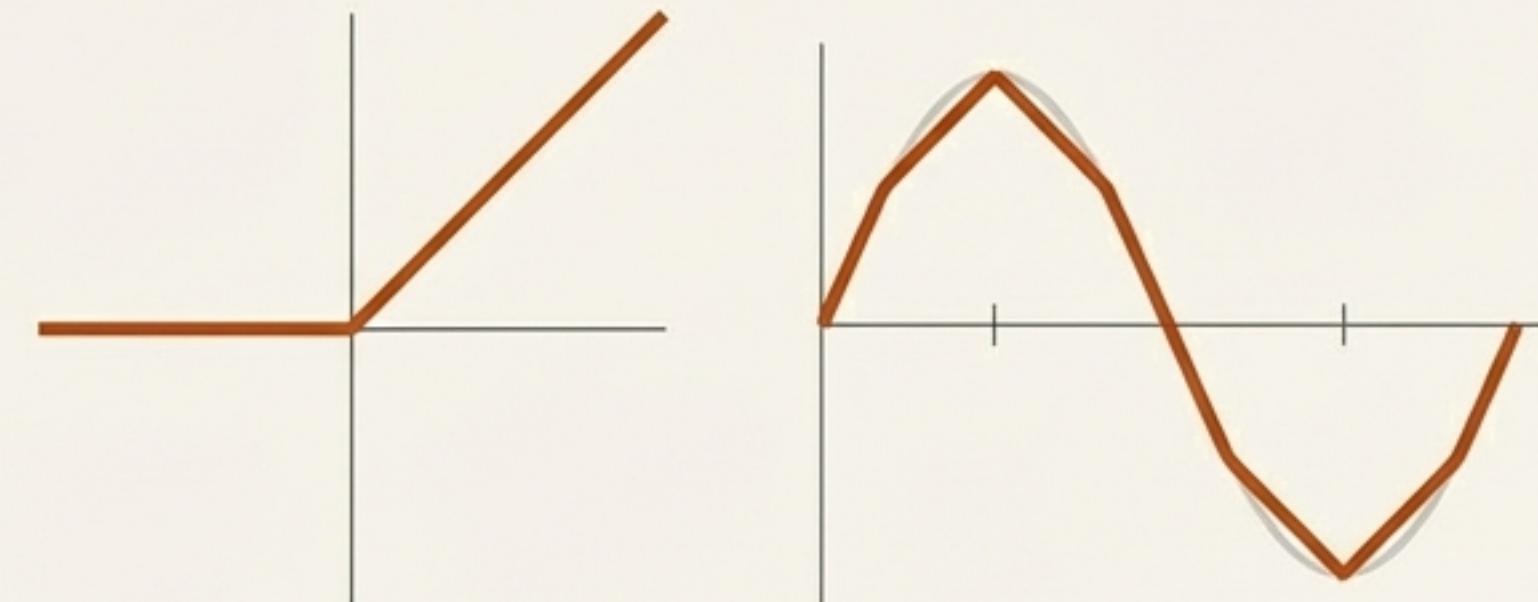
$$o = w_1(w_{11}I_1 + w_{12}I_2) + \dots = w'_1I_1 + w'_2I_2 + b'$$



Del 2: Rešitev

Nelinearni 'preklop' z aktivacijsko funkcijo ReLU.

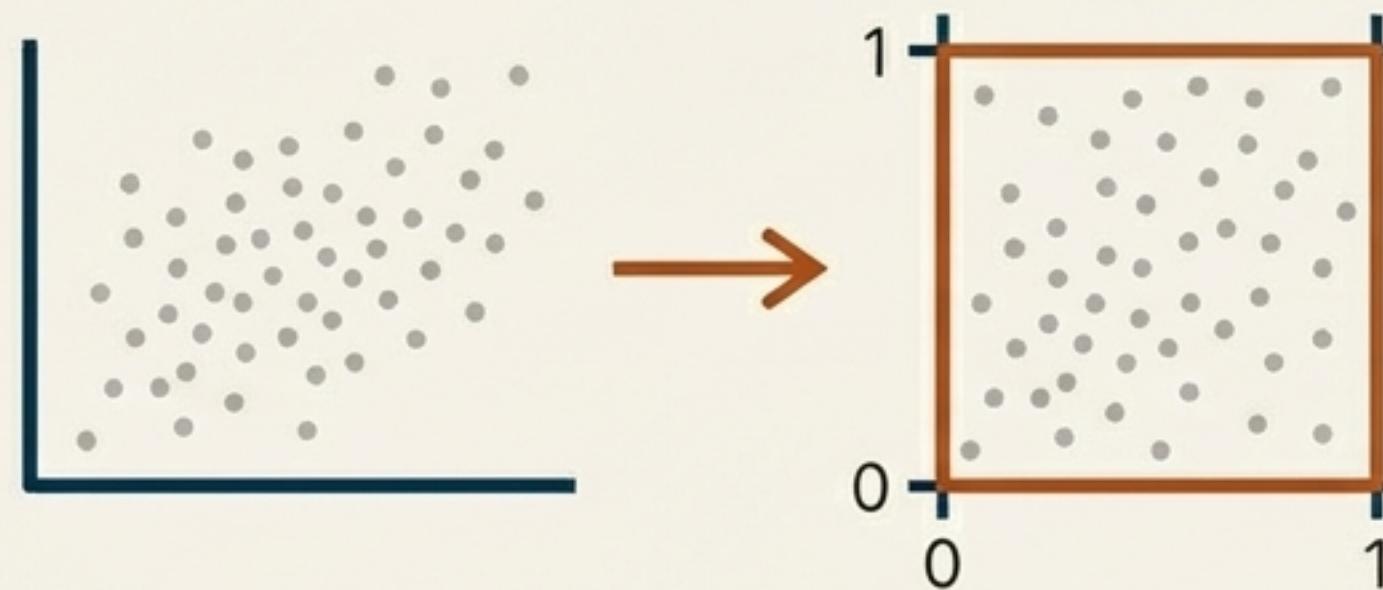
Na vrednost v vsakem vozlišču (razen vhodnih) uporabimo nelinearno funkcijo. Najpogostejša je ReLU (Rectified Linear Unit), ki preprosto odreže vse negativne vrednosti.



Priprava podatkov: Mreža potrebuje urejen in strukturiran vhod

Zvezni podatki

Normalizacija



Podatke preslikamo na interval med 0 in 1 (ali -1 in 1). S tem zagotovimo, da imajo vsi vhodi primerljivo velikost in se izognemo numeričnim težavam.

Kategorični podatki

'One-Hot' kodiranje

Da bi se izognili ustvarjanju lažnih hierarhičnih odnosov (npr. rumena > modra), vsako kategorijo predstavimo z vektorjem.



Korak 1: Izračun napake s stroškovno funkcijo

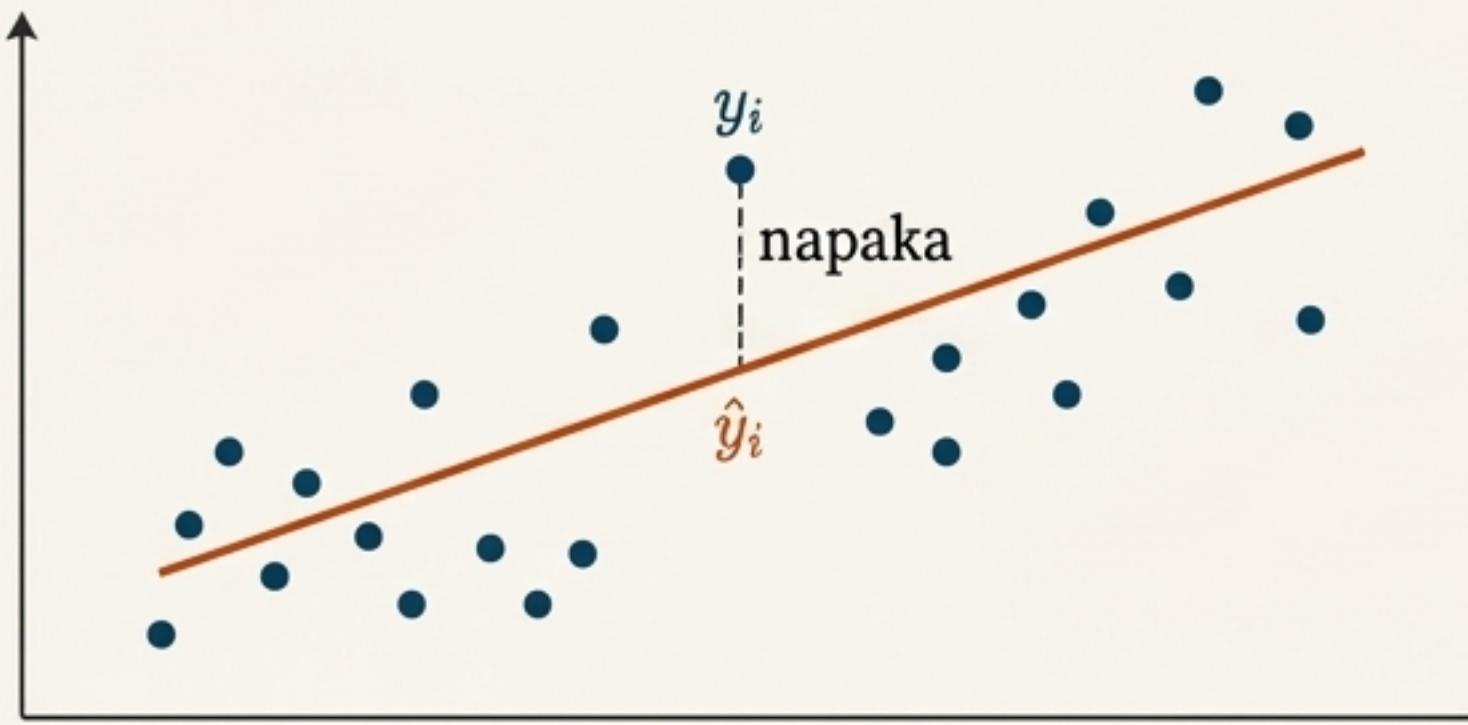
Kako 'narobe' je naša napoved?

Za zvezne podatke (npr. napoved cene)

Povprečna kvadratna napaka (MSE)

Izračuna povprečje kvadratov razlik med dejanskimi vrednostmi (y) in napovedmi modela (\hat{y}).

$$\text{MSE} = \frac{1}{N} \sum (y_i - \hat{y}_i)^2$$



Za klasifikacijo (npr. prepoznavanje kategorij)

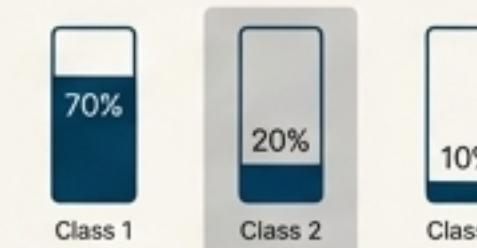
Križna Entropija (Cross-Entropy)

Mera, ki je nizka, če je model zelo prepričan v *pravilen* odgovor, in visoka, če je prepričan v napačnega.

Pravi razred: $y = [0, 1, 0]$

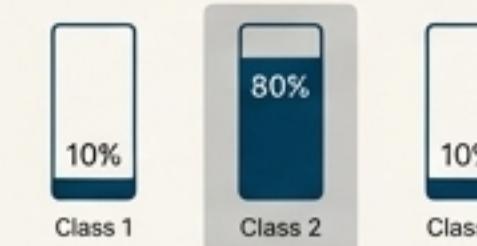
Slaba napoved

$\hat{y} = [0.7, 0.2, 0.1] \rightarrow$ Visoka napaka ($H = 1.609$)



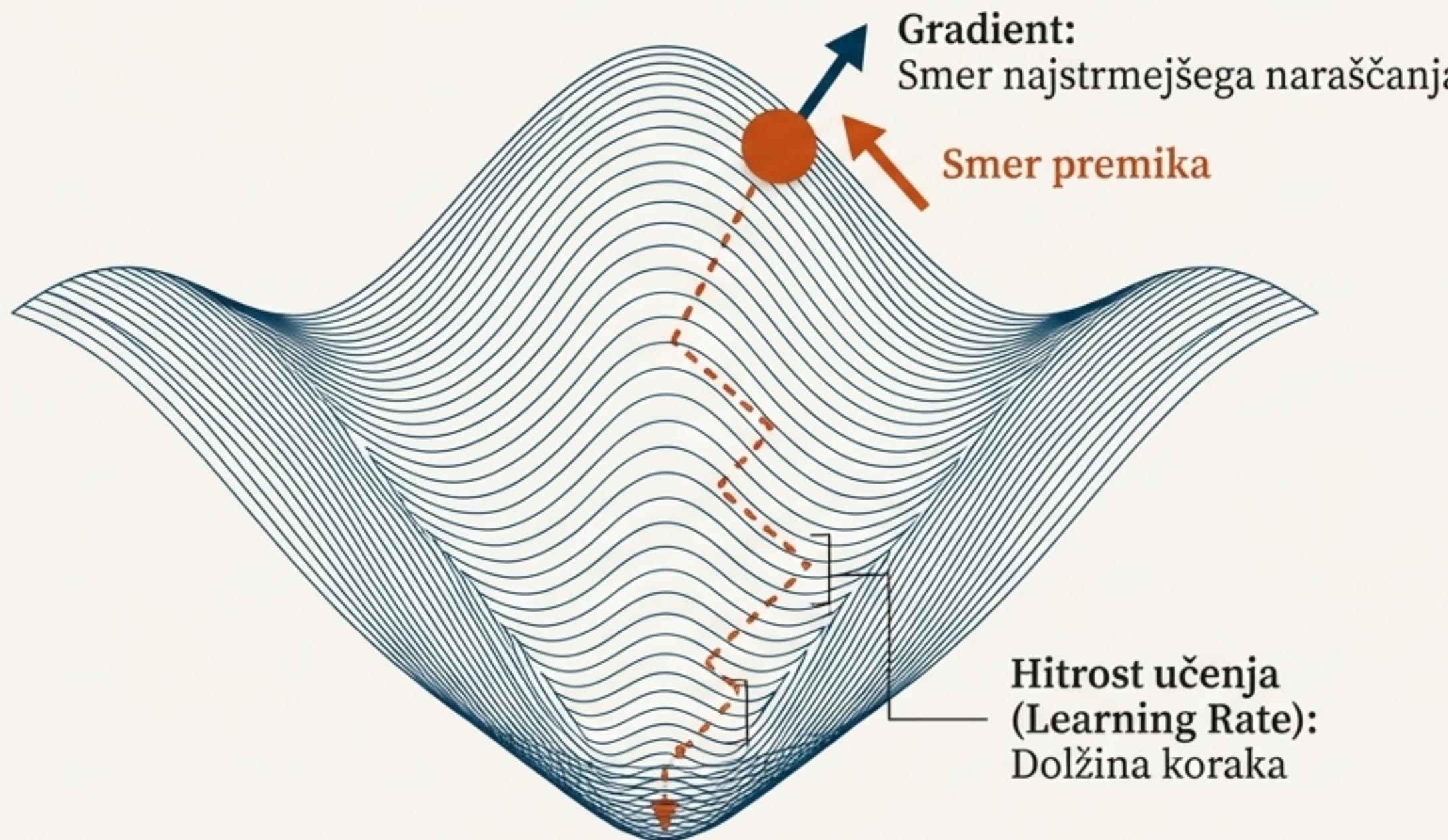
Dobra napoved

$\hat{y} = [0.1, 0.8, 0.1] \rightarrow$ Nizka napaka ($H = 0.223$)



Korak 2: Optimizacija z gradientnim spustom

Kako prilagodimo parametre, da zmanjšamo napako?



Prevelika
hitrost

Prevelika hitrost

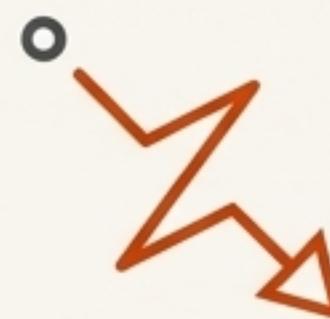
Premajhna
hitrost

Premajhna hitrost

Pametnejša optimizacija: Družina naprednih algoritmov

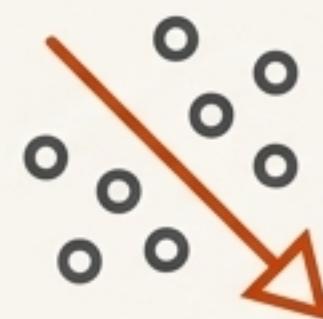
Osnovni gradientni spust je počasen. V praksi uporabljamo izboljšane različice, ki pospešijo in stabilizirajo učenje.

SGD (Stochastic Gradient Descent)



Prilagajanje uteži po vsakem posameznem vzorcu. Hitro, a potencialno nestabilno.

Mini-Batch Gradient Descent



Najboljše iz obeh svetov: učenje v manjših skupinah (batch-ih). Stabilnejše od SGD, hitrejše od klasičnega GD.

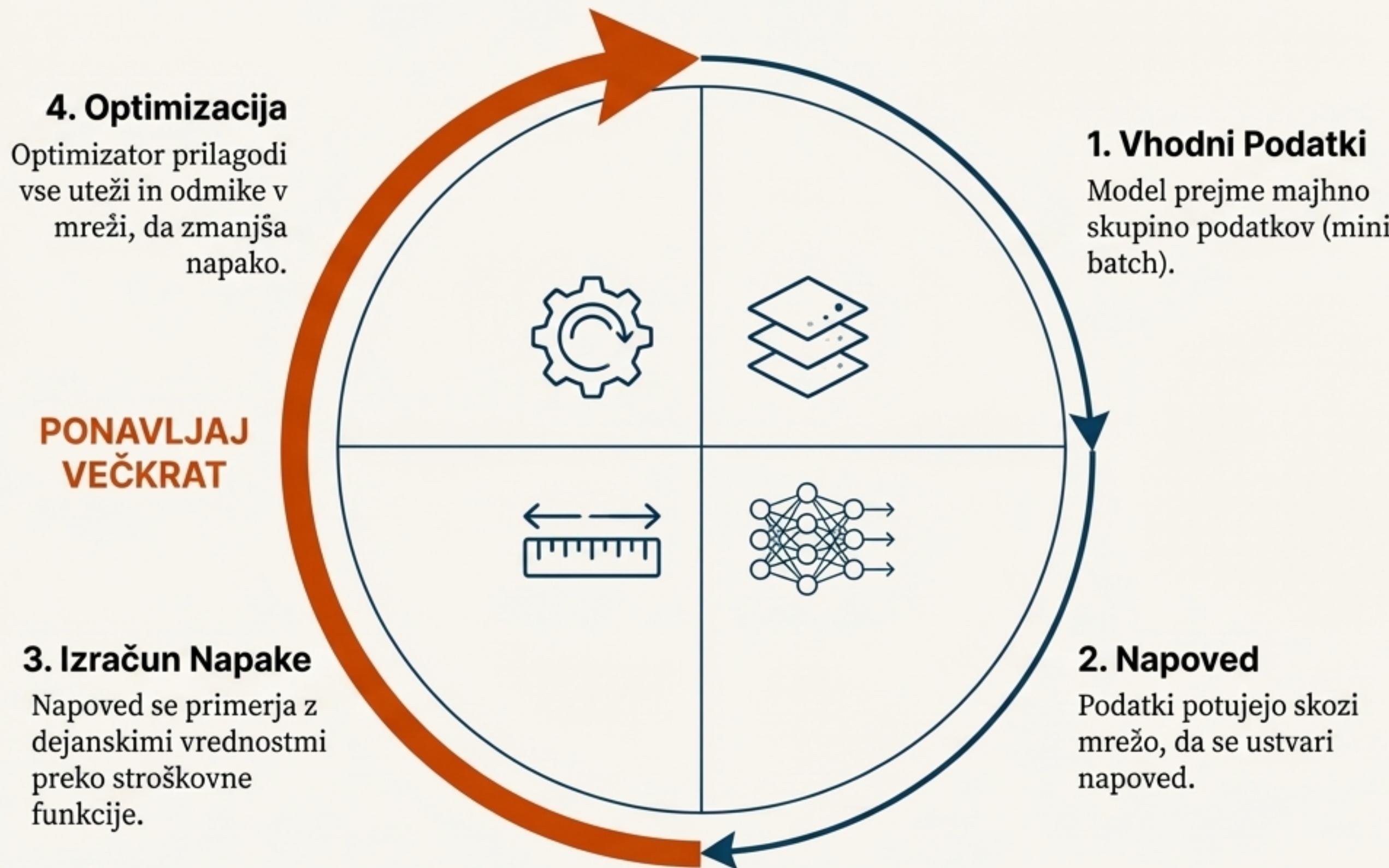
Adam (Adaptive Moment Estimation)

Današnji standard



Prilagaja hitrost učenja za vsak parameter posebej in uporablja ‘zagon’ (momentum). Učinkovit in dobro deluje v večini primerov.

Cikel učenja: Ponavljajoč se proces izboljševanja



Ključni pojmi

Epoha: En prehod skozi *celoten* nabor učnih podatkov.

Velikost paketa (Batch Size):
Število vzorcev, po katerih se posodobijo uteži v enem koraku optimizacije.

Končni izpit: Validacija in nevarnost 'prekomernega prilagajanja' (Overfitting)



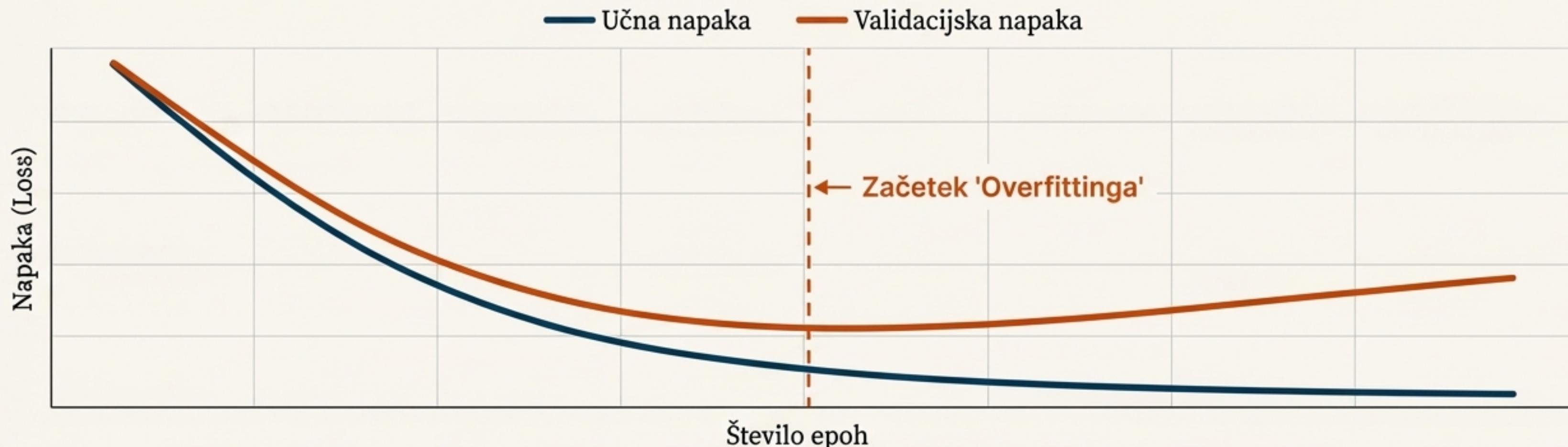
Učenje (Training)

Učenje na pamet iz starih izpitov.
Model se uči na znanih podatkih.
Napaka se zmanjšuje.



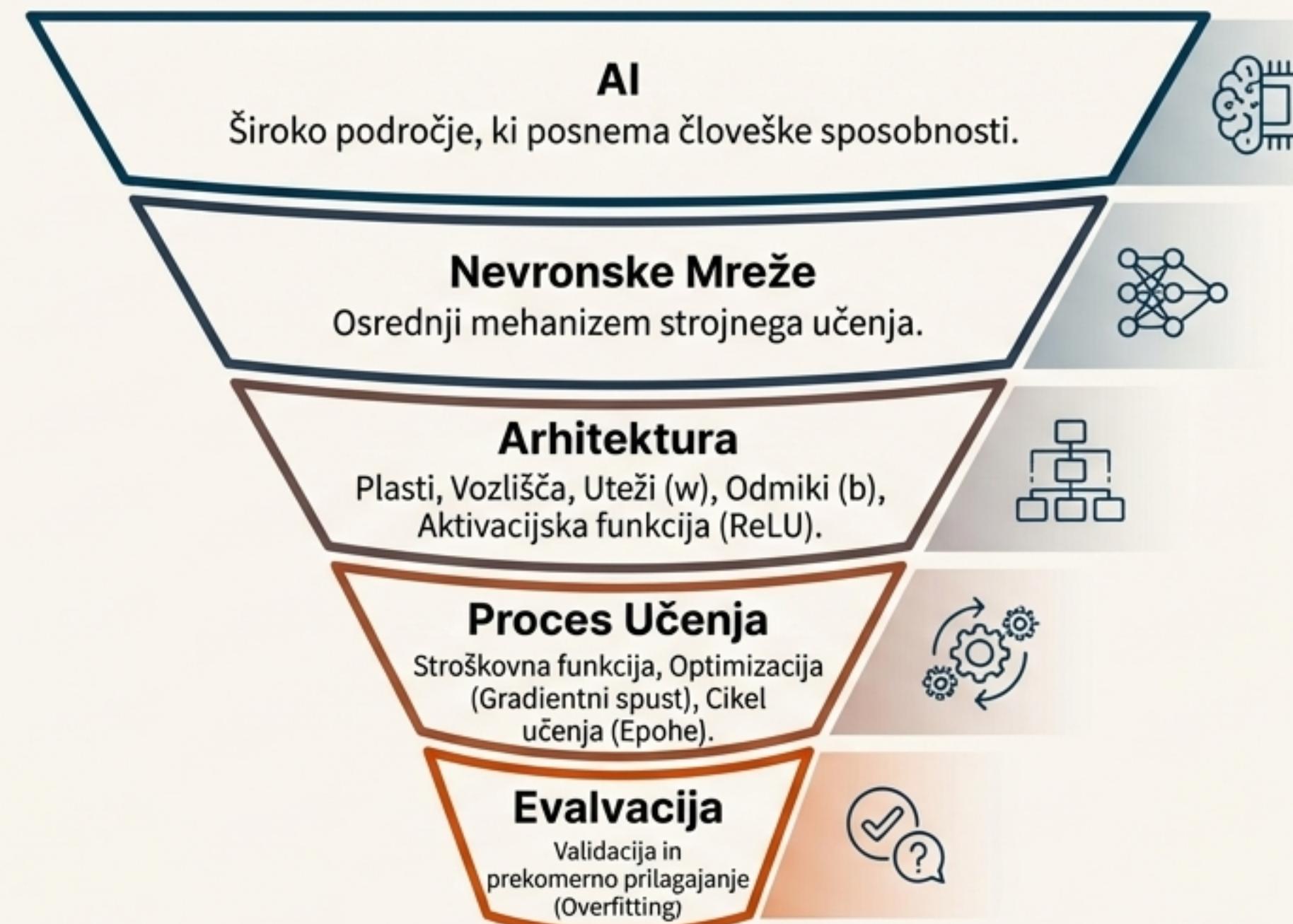
Validacija (Validation)

Reševanje novega izpita z neznanimi vprašanji.
Model preizkusimo na podatkih, ki jih med
učenjem ni videl. To je pravi test njegovega znanja.



Overfitting se zgodi, ko se model podatke nauči 'na pamet', namesto da bi razumel dejanske povezave. Posledično je odličen na učnih podatkih, a slab na novih.

Povzetek: Naše potovanje skozi lijak razumevanja



Začeli smo pri splošnem pojmu AI in se prebili do samega jedra delovanja nevronske mreže – sistematičnega procesa učenja iz napak.

Ključna spoznanja in naslednji koraki

-  Nevronske mreže se učijo kompleksnih vzorcev s postopnim prilagajanjem notranjih parametrov (uteži in odmikov).
-  Učenje je iterativen cikel napovedovanja, merjenja napake in optimizacije za zmanjšanje te napake.
-  Uspeh modela se ne meri le z nizko napako pri učenju, ampak z njegovo zmožnostjo posploševanja na novih, neznanih podatkih (validacija).

Teorijo smo osvojili.

Naslednji korak? Gradnja prve mreže v Kerasu.