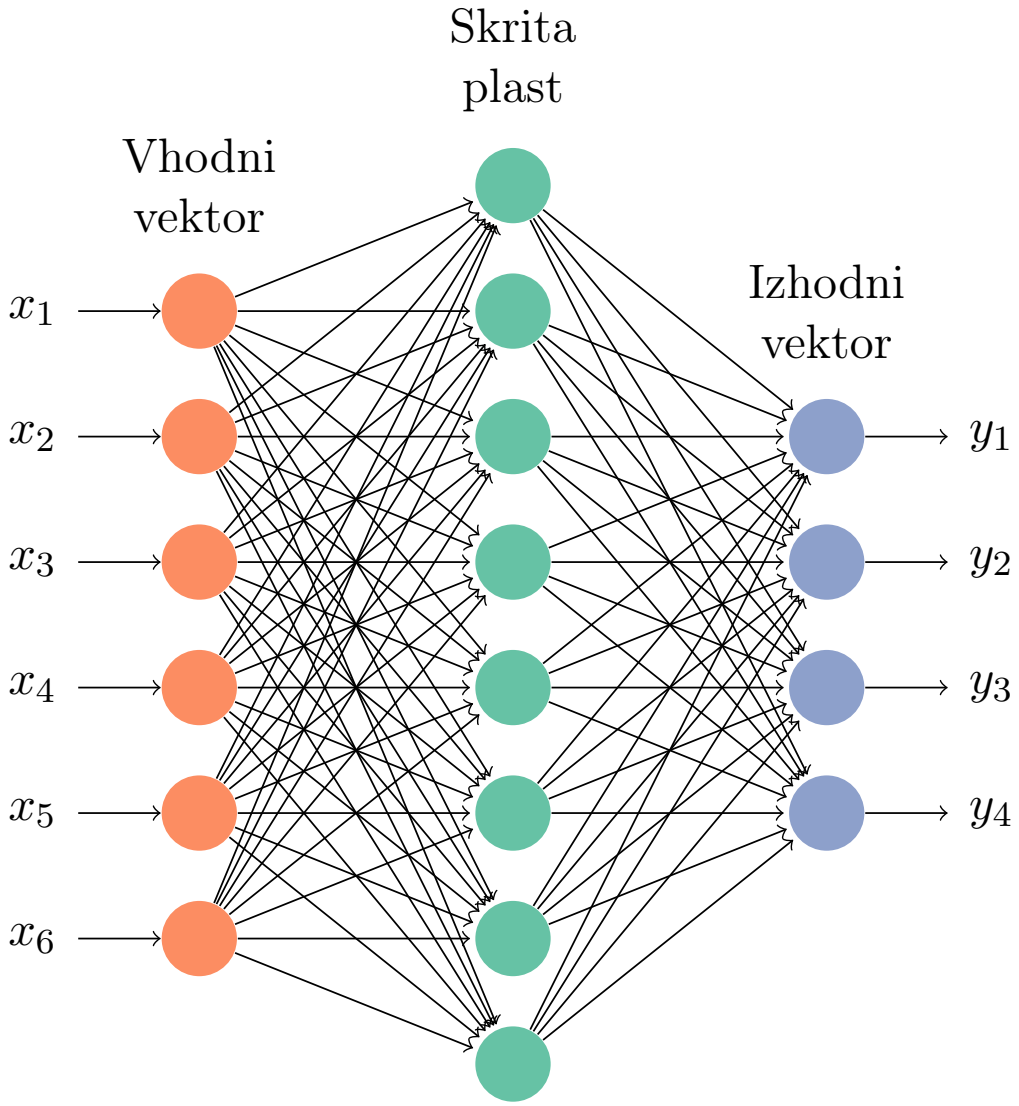
13. naloga - Nevronske mreže

Tina Klobas

4. avgust 2022

1 Opis problema

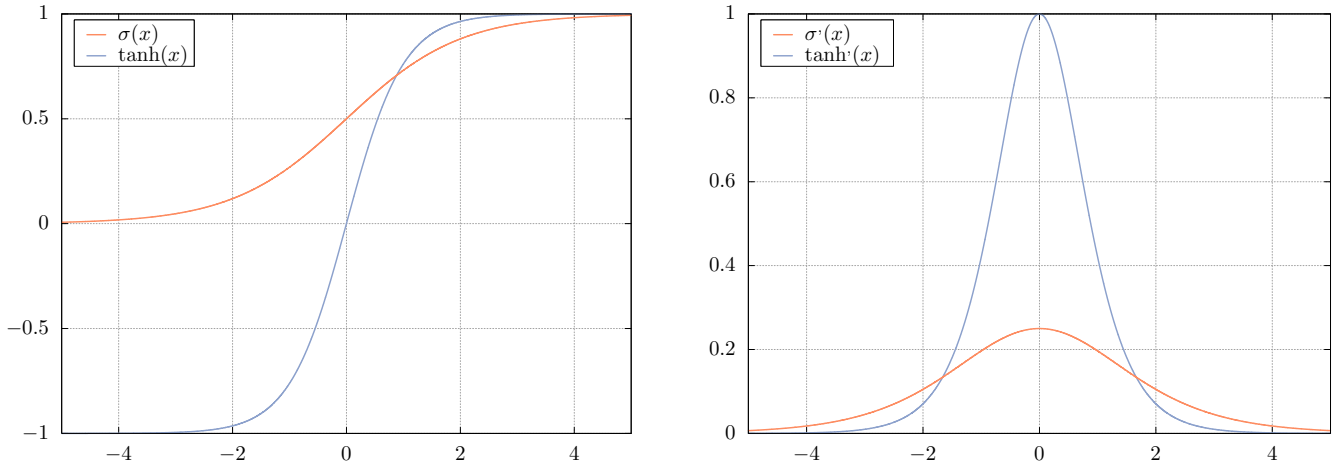
Model nevronske mreže so verige preslikav med vektorji in nelinearnih aktivacijskih funkcij in so tip univerzalnih funkcij, ki v principu lahko predstavljajo poljubno funkcijsko zvezo med vhodnim vektorjem $\mathbf{x}$ in izhodnim vektorjem $\mathbf{y}$.



Mrežo z eno skrito plastjo $\mathbf{z}$ opišemo s setom enačb:

$$\mathbf{z}_j = \Sigma_i f(\beta_{j0} + \beta_{ji} x_i)$$
$$\mathbf{y}_k = \Sigma_j f(\alpha_{k0} + \alpha_{kj} z_j)$$

Člena $\beta_{j0} in \alpha_{k0}$ predstavljata *bias* in povečata fleksibilnost modela, sta pa ekvivalentna razširitvi vseh plasti razen zadnje za eno konstantno komponento $x_0 = z_0 = 1$.
Aktivacijska funkcija je lahko sigmoidna funkcija $\sigma = 1/(1 - \mathrm{e}^{-x})$, mi pa bomo uporabili $f(x) = \tanh(x)$.



Slika 1.1: Aktivacijski funkciji na levi in njuna odvoda na desni.

Izvrednotenje rezultata nevronske mreže je trivialna operacija, zahtevni del je inverzni problem – določitev linearnih preslikav α_{ij} in β_{ij} med plastmi nevronske mreže – saj je vse, kar poznamo, željeni rezultat za nekaj primerov vhoda, sama funkcija pa predstavlja kompleksno hevristiko odločanja. Ta problem rešujemo s treniranjem nevronske mreže na veliki količini vhodnih vektorjev $\mathbf{x}$ z znanim izidom $\mathbf{t}$. Funkcijo odstopanja $E = \frac{1}{2}||\mathbf{y} - \mathbf{t}||^2$ minimiziramo z obratno propagacijo napake, ki je poseben primer minimizacije z gradientnim spustom.
Definirajmo residual, ki se rekurzivno propagira z zadnje plasti nevronov proti prvi:

$$\delta_k^2 = \Sigma_j (y_k - t_k) f'(\alpha_{kj} z_j), \tag{1}$$
$$\delta_j^1 = \Sigma_{i,k} \delta_k^1 \alpha_{kj} f'(\beta j i x_i). \tag{2}$$

Popravek matrikam uteži, ki ga izvedemo na vsakem učnem primeru, potem zapišemo kot

$$\alpha_{kj} \to \alpha_{kj} - \eta \delta_k^2 z_j, \tag{3}$$
$$\beta_{ji} \to \beta_{ji} - \eta \delta_j^1 x_i, \tag{4}$$

pri čemer je η hitrost učenja – korak gradientnega spusta. Primerna vrednost je na primer $\eta = 0.01$. Začetne uteži α in β so običajno matrike Gaussovih naključnih števil z disperzijo $\sigma_\alpha^2 = 2/\dim(\mathbf{z})$, $\sigma_\beta^2 = 2/\dim(\mathbf{x})$.

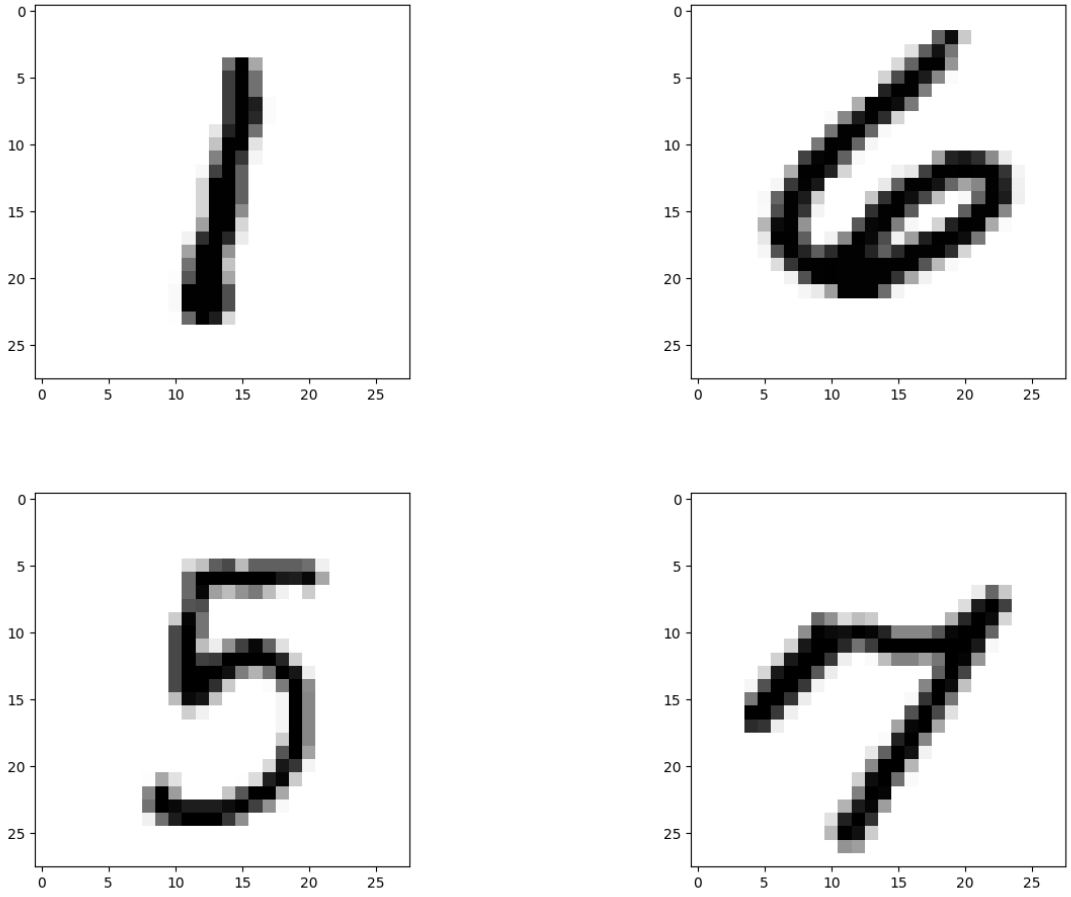
2 Učenje nevronske mreže

2.1 Naloga

V datotekah imamo nabor črnobelih slik ročno napisanih števk od 0 do p s pripadajočimi oznakami $0 - 9$. Skonstruirati in naučiti želimo nevronsko mrežo s tremi plastmi – na prvo plast pripeljemo vhodne podatke ($28^2 = 784$ pikslov slike, reskaliranih med 0 in 1), skrita plast je enake dimenzije kot prva, izhodna plast pa predstavlja 10 izhodov, za vsako števko enega, katerih ciljne vrednosti so 1 za pravo števko in -1 za napačno. Tekom učenja spremljamo napako E ter delež pravilno določenih števk.

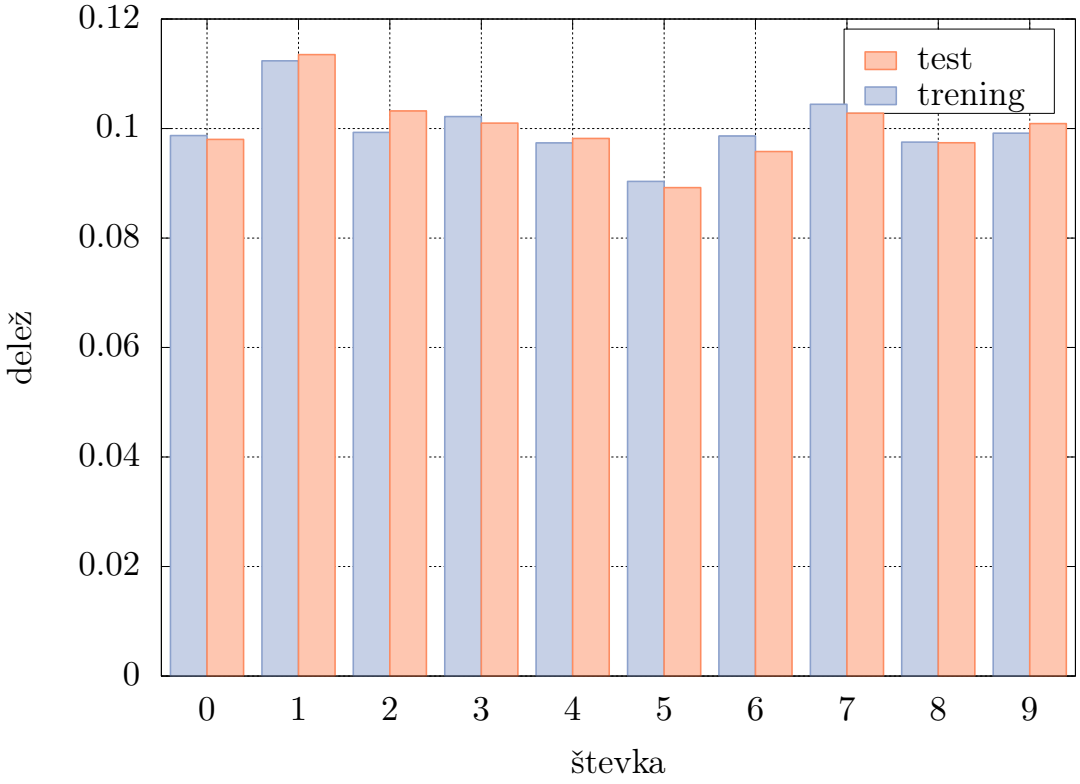
2.2 Pristop k problemu

Slike so pridobljene iz podatkovne baze *MNIST*, ki vsebuje 70.000 slik števk. Primeri so prikazani na slikah 2.1.



Slika 2.1: 4 slike iz baze podatkov MNIST.

Na grafu 2.2 vidimo, da imamo v podatkih približno enak delež vseh števk, rahlo sicer odstopa delež petk in enk. Poglejmo najprej kakšno porazdelitev števk imamo v podatkih:

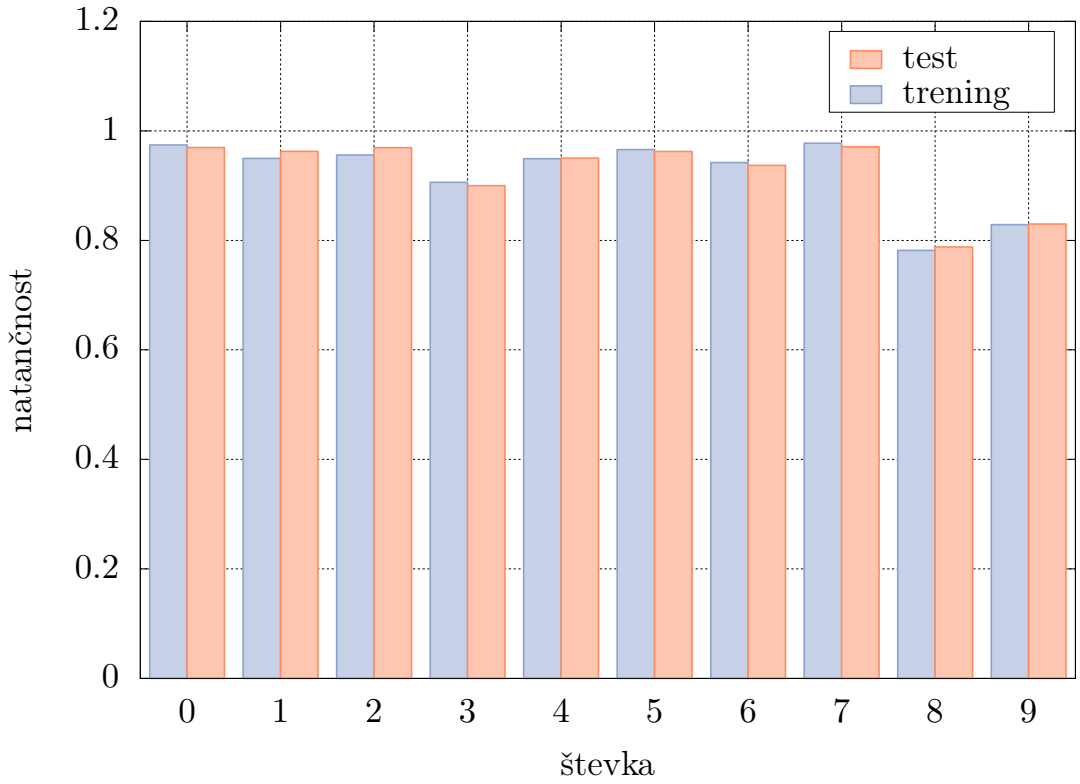


Slika 2.2: Delež števk v bazi podatkov MNIST za trening in test.

Za začetek naredimo učenje na eni skriti plasti, z dimenzijami uteži $\alpha = (784, 28)$ in $\beta = (28, 10)$. Delež pravilno ugotovljenih števk iz trening seta in testnega seta:

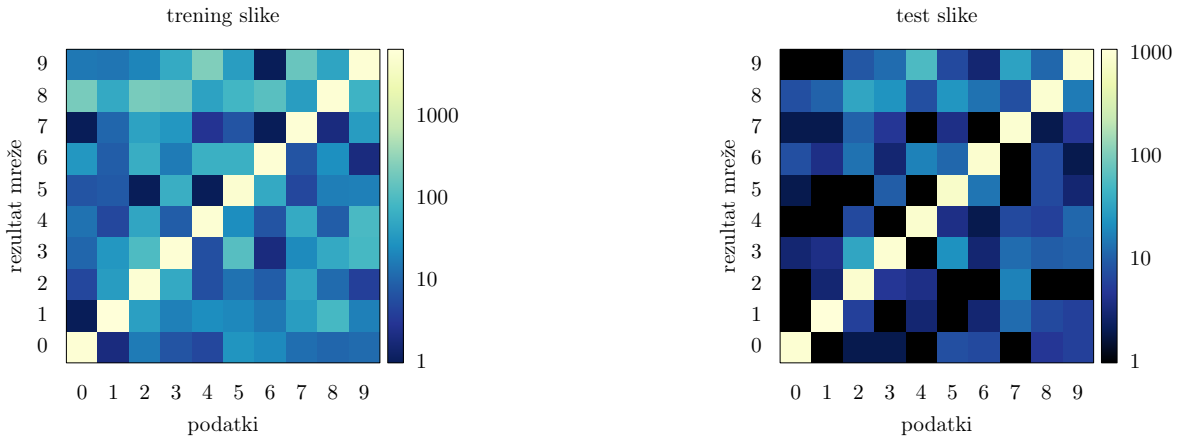
Podatki	Natančnost
Trening	0,9181
Test	0,9192

Kako je ta učinkovitost porazdeljena po števkah vidimo na grafu 2.3.



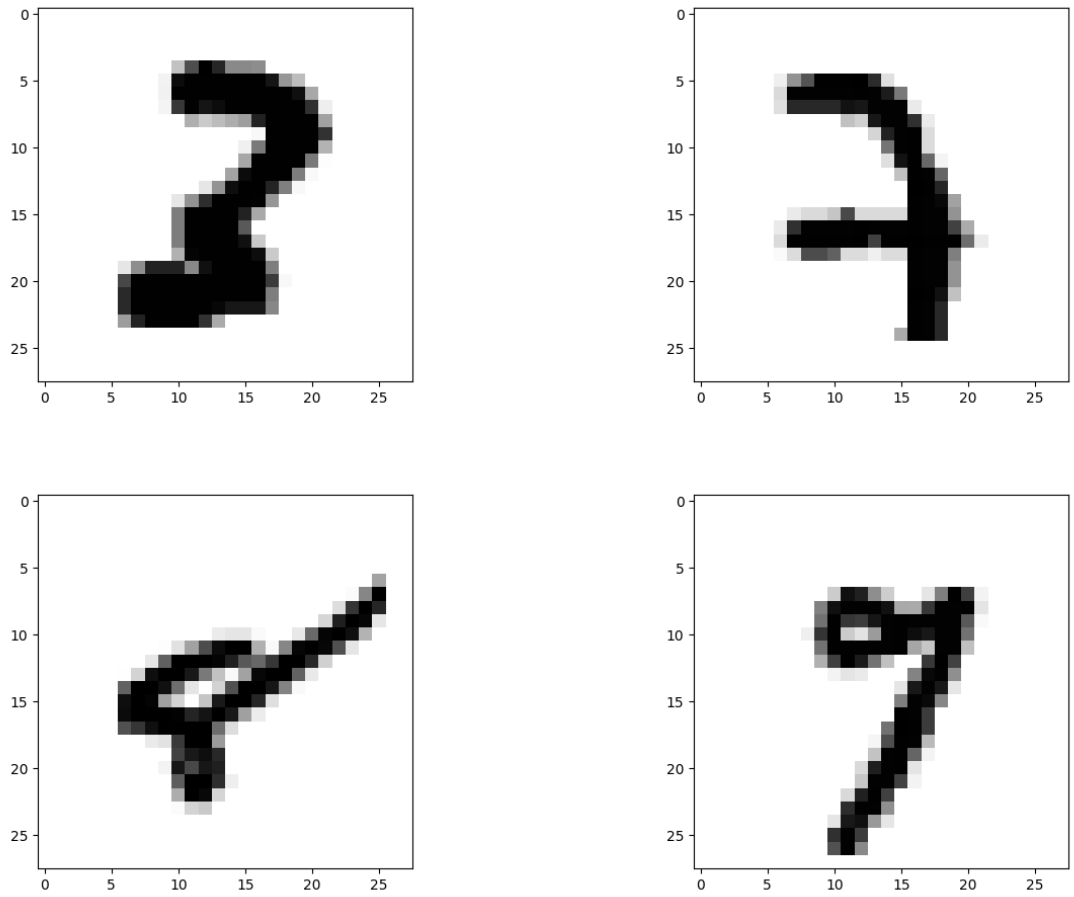
Slika 2.3: Delež pravilno ugotovljenih števk trening in testnega seta.

Za boljšo predstavo pa narišimo toplotno karto pravilno in napačno identificiranih števk. Vsak kvadrat na grafu nam pove kolikokrat je mreža pravilno ali napačno uganila posamezno števko. Graf 2.4 se ujema s prejšnjim histogramom 2.3, kjer lahko opazimo, da je mreža velikokrat najmanj natančno zadela števki 8 in 9. To lahko na toplotni karti vidimo kot svetlejši vrstici pri obeh števkah.



Slika 2.4: Prikaz zadetih in zgrešenih števk nevronske mreže med samim učenjem in na testnem naboru slik.

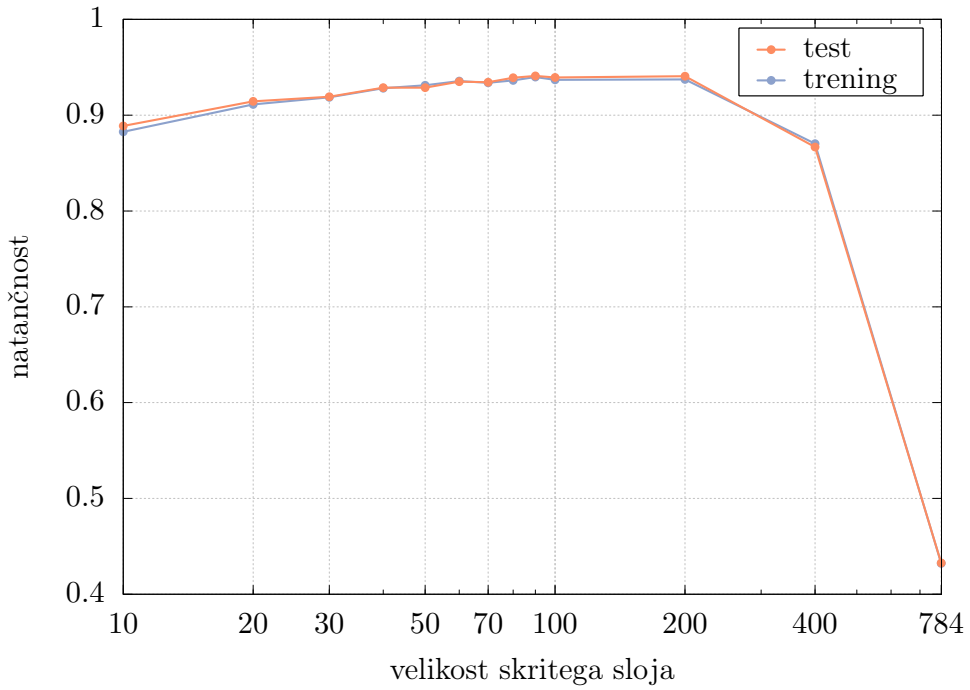
Spodaj na slikah 2.5 so prikazane napačno identificirane števke. V zgornji vrstici števki prebere kot 2. Pri 3 je spodnji zavoj premalo viden in bi ga verjento tudi sama prebrala napačno. 7 na desni se od ostalih 7 v bazi razlikuje po tem, da ima prečno črtico in naša mreža se s takimi sedmicami med učenjem ni srečala. Spodnja osmica je res grdo napisana in je verjetno tudi sama ne bi znala pravilno prebrati. Enako velja za 9 spodaj desno, kjer je zavoj preveč sploščen in ima netipično oster rob.



Slika 2.5: Zgrešene števke nevronske mreže. Na zgornjih slikah je obe števki označila za 2, spodaj levo 6 in 7 desno.

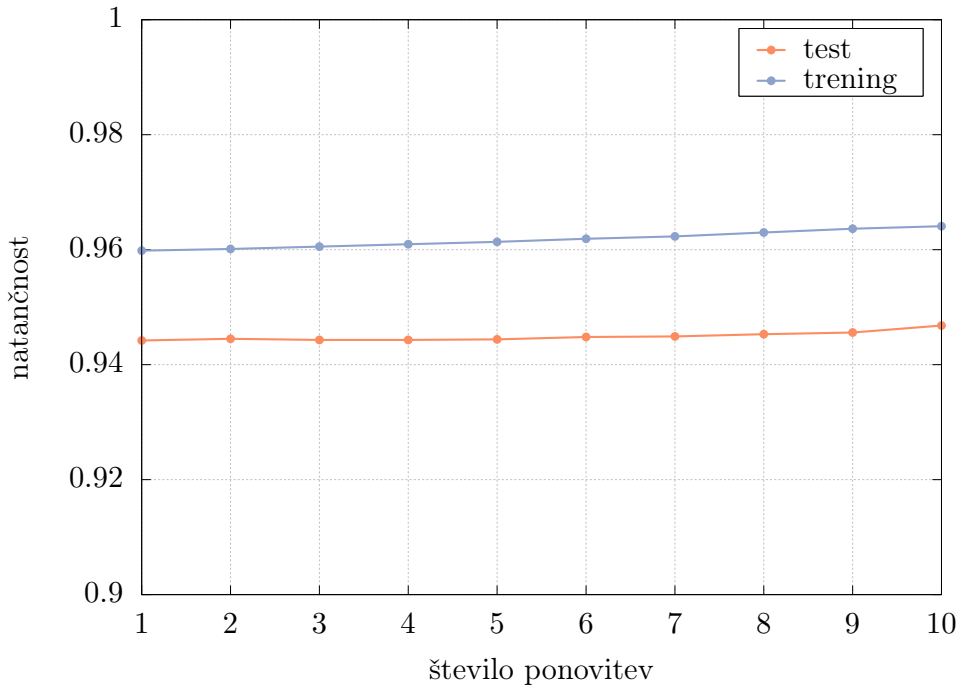
Poglejmo si kako se spreminja natančnost mreže s spreminjanjem velikosti skrite plasti. Na grafu 2.6 vidimo, da

z večanjem skritega sloja izboljšujemo natančnost naše mreže dokler ne pridemo do velikosti, ko začne ta padati. Vidimo, da se ne splača pretiravati in dati mreži prevelike prostosti in se najbolje odzove, ko je velikost skrite plasti enaka ali vsaj primerljiva z velikostjo vhodnega vektorja.



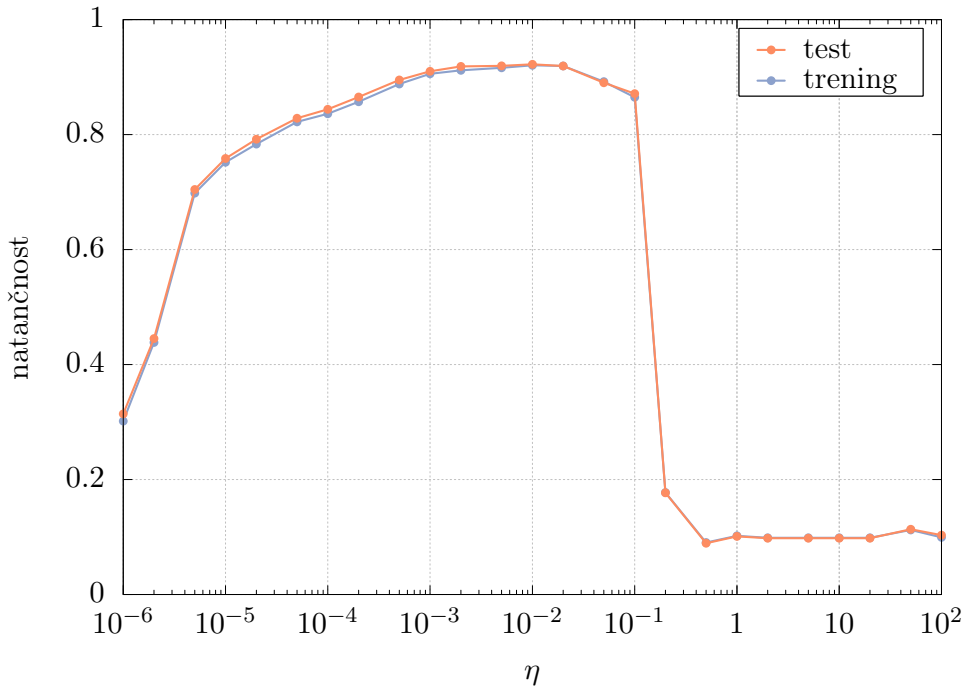
Slika 2.6: Spreminjanje natančnosti mreže z večanjem velikosti skrite plasti.

Natančnost lahko poskusimo izboljšati tudi z večkratnim učenjem na enakem setu podatkov. Graf 2.7 prikazuje naraščanje natančnosti z večanjem števila ponovitev, vendar to naraščanje v našem primeru ni dovolj izrazito, da bi se nam časovno splačalo iti več kot denimo dvakrat skozi podatke.



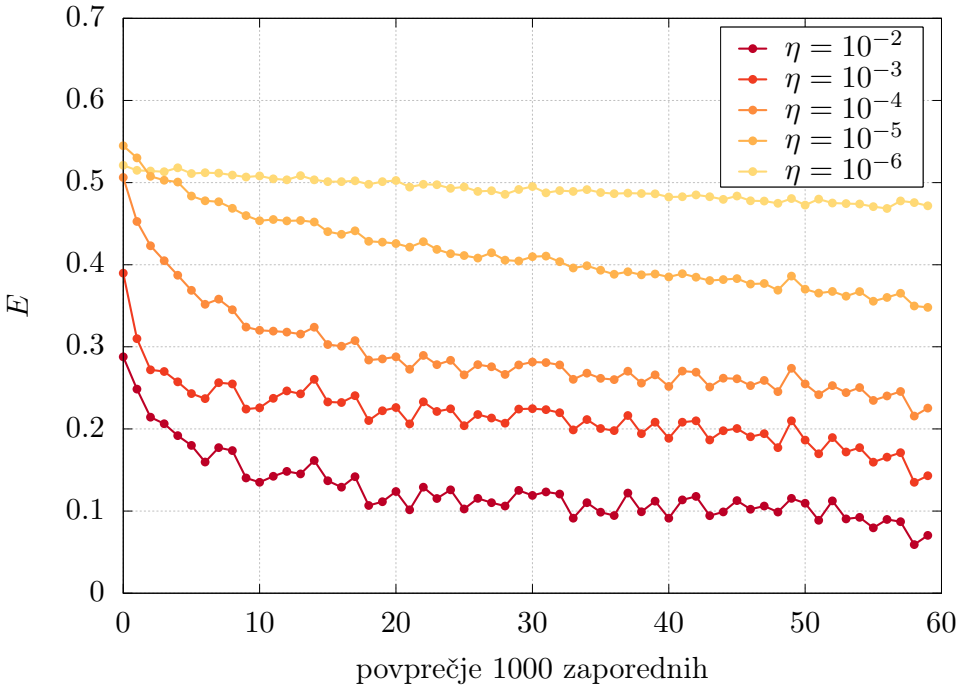
Slika 2.7: Natančnost modela v odvisnosti od števila ponovitev.

Preverimo še kako se učinkovitost mreže spreminja s spreminjanje hitrosti učenja η. Na sliki 2.8 vidimo, da dosežemo maksimalno natančnost pri vrednosti $\eta \approx 0.01$ po tem, pa začne naglo padati.



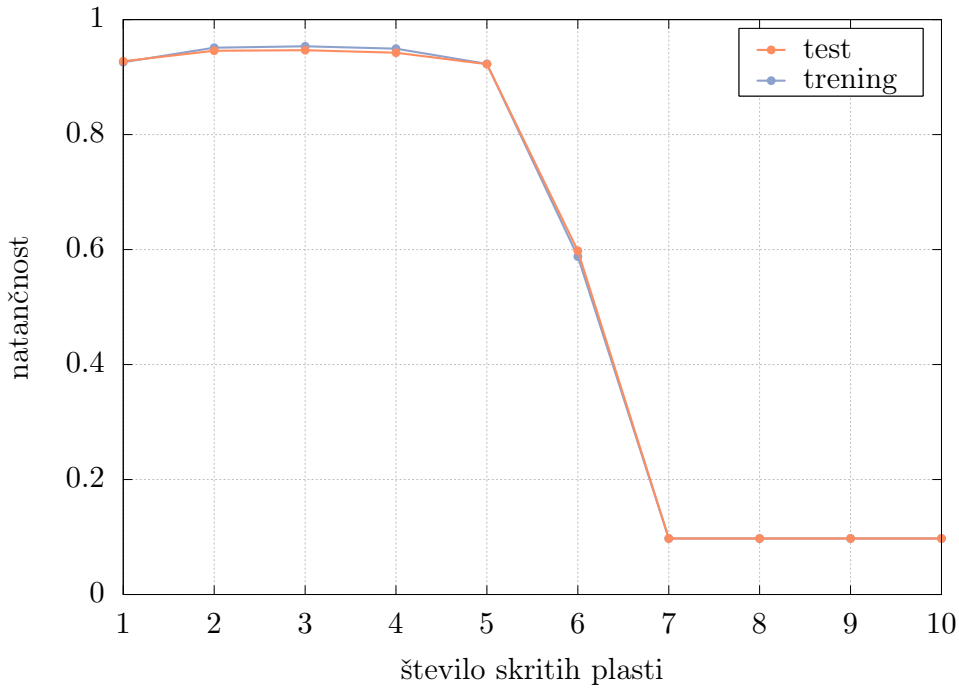
Slika 2.8: Spreminjanje natančnosti mreže s spreminjanjem hitrosti učenja η, kjer smo povprečili 1000 zaporednih korakov.

Na grafu 2.9 vidimo kako se tekom učenja izboljšuje napaka: $E = \frac{1}{2}||y^T - t||$, dokler se ne ustali na neki asimptotični vrednosti.



Slika 2.9: Spreminjanje napake E tekom učenja pri različnih parametrih η.

Napake pri vzvratnem razširjanju skozi veliko plasti postanejo zelo majhne, kar oteži učenje, kar je tudi razlog za manjši natančnost pri prevelikem številu plasti. To lahko tudi vidimo na grafu 2.10 kjer imamo padec natančnosti pri večjem številu skritih plasti.



Slika 2.10: Spreminjanje natančnosti mreže s spreminjanjem števila plasti.

3 Deep dream

3.1 Naloga

Pri učenju mreže smo minimizirali napako z iterativnim popravljanjem uteži W^l. Na že naučeni mreži pa lahko vprašamo, kateri vhod minimizira napako na izhodu za izbrani izhod. Z enakim postopkom posrednega odvajanja, s katerim smo določili popravke uteži, dobimo iteracijski popravek za vhodni vektor

$$\mathbf{y}^0 \to \mathbf{y}^0 + \eta\delta^0 W^0, \tag{5}$$

pri čemer do δ^0 pridemo s pomočjo rekurzije. To iteracijsko shemo ponavljamo, dokler napaka na izhodu ne doseže zadovoljivo majhne vrednosti. Začetni približek za sliko $\mathbf{y}^0$ je v našem primeru lahko kar črna slika, na katero želimo, da nevronska mreža nariše izbrano števko glede na naučene značilnosti. Ta postopek je znan tudi pod frazo *"deep dream"* in ga lahko uporabimo kot vpogled v lastnosti, ki jih mreža zazna kot pomembne za klasifikacijo.

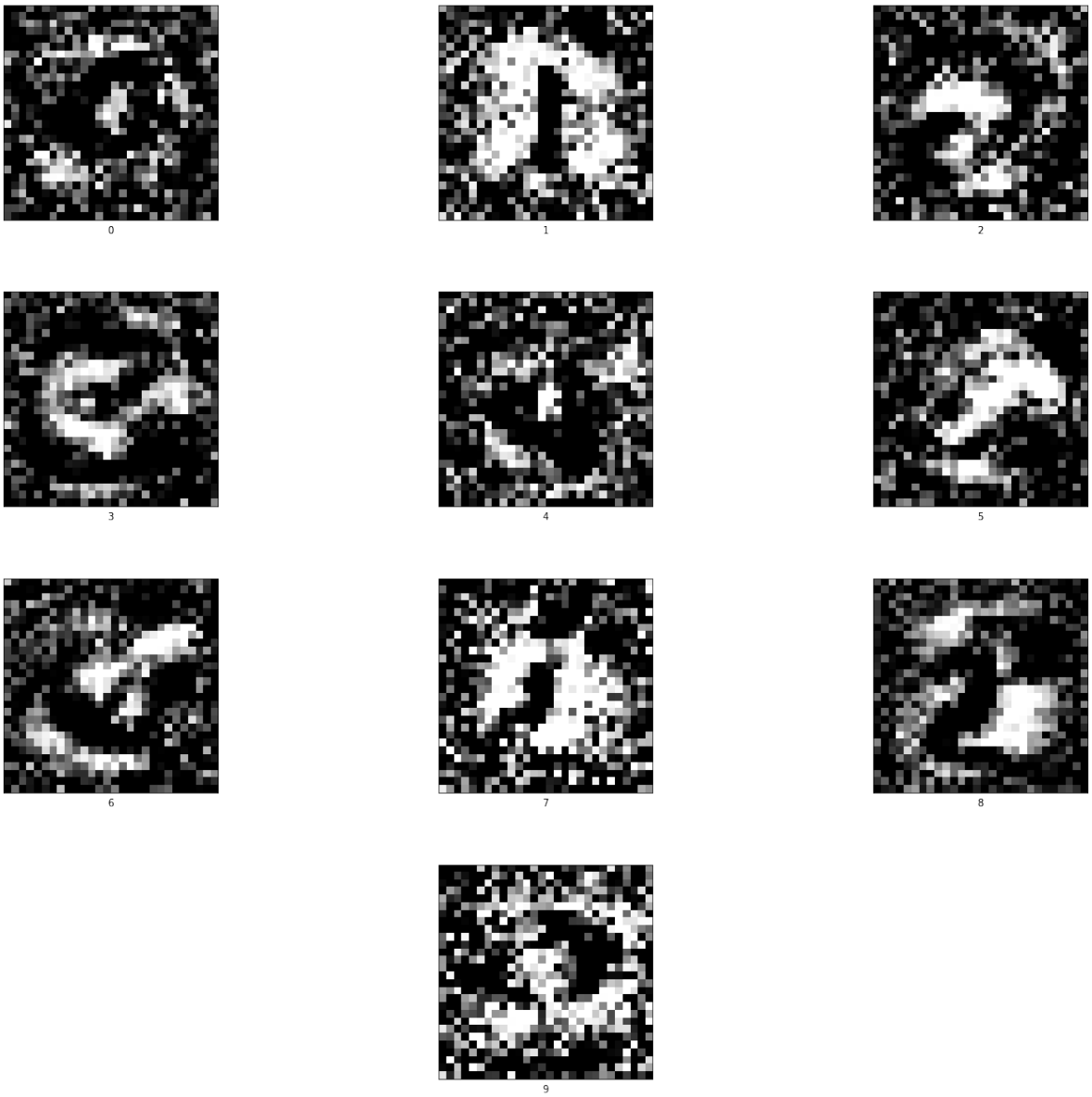
3.2 Pristop k problemu

Rekurzivno računamo residual po plasteh:

$$\delta^{l-1} = -(\mathbf{y}^l - \mathbf{t}) \cdot f'(W^{l-1}\mathbf{y}^{l-1}) \tag{6}$$

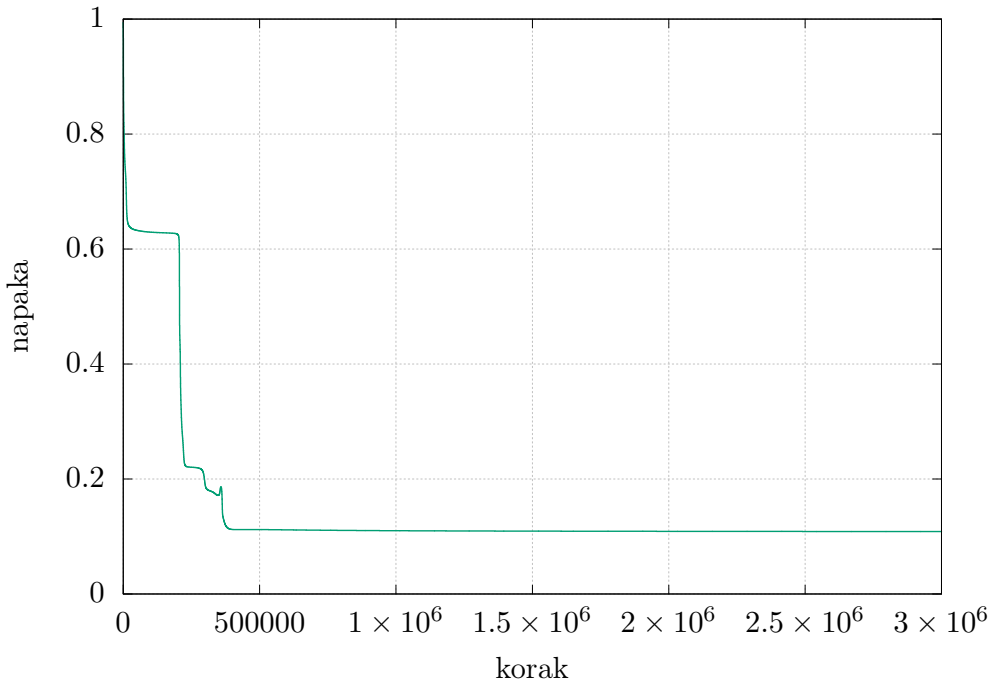
$$\delta^l = \delta^{l+1} W^{l+1} \cdot f'(W^l \mathbf{y}^l) \tag{7}$$

s pomočjo česar izračunamo nov $\mathbf{y}_0$.
Poglejmo si kar rezultate postopka minimizacije napake:



Slika 3.1: Metoda deep dream na mreži s skrito plastjo dimenzije 80.

Na zgornjih grafih 3.1 so prikazani rezultati za vseh deset števk. Pri večini lahko razločimo obrise, ki jih mreža prepozna, da nato določi za katero števko gre. Najbolje to opazimo pri števkah $0, 1$ in 3, zanimive so pa tudi ostale, pri katerih prepozna le del giba. Denimo pri dvojki prepozna zgornji lok in zavoj, pri številki pa devet vidimo zgornji krogec.
Pri risanju teh slik je bila zelo pomembna izbira dimenzije skrite plasti; pri plasteh dimenzije 28 rezultati niso bili razločni, z večanjem pa smo dobili vedno boljše slike, vendar se je tudi računski čas izjemno poslabšal.
Tekom modela smo manjšali residual oziroma napako, ki jo lahko vidimo na grafu 3.2, ki je zelo podoben grafu 2.9, kjer smo risali napako tekom učenja.



Slika 3.2: Norma napake v odvisnosti od števila korakov pri deep dream metodi za števko 9.