

Ispit iz algoritamskih heuristika april 2024

Izvestaj uz zadatak 3 – Treniranje neuralnih mreza

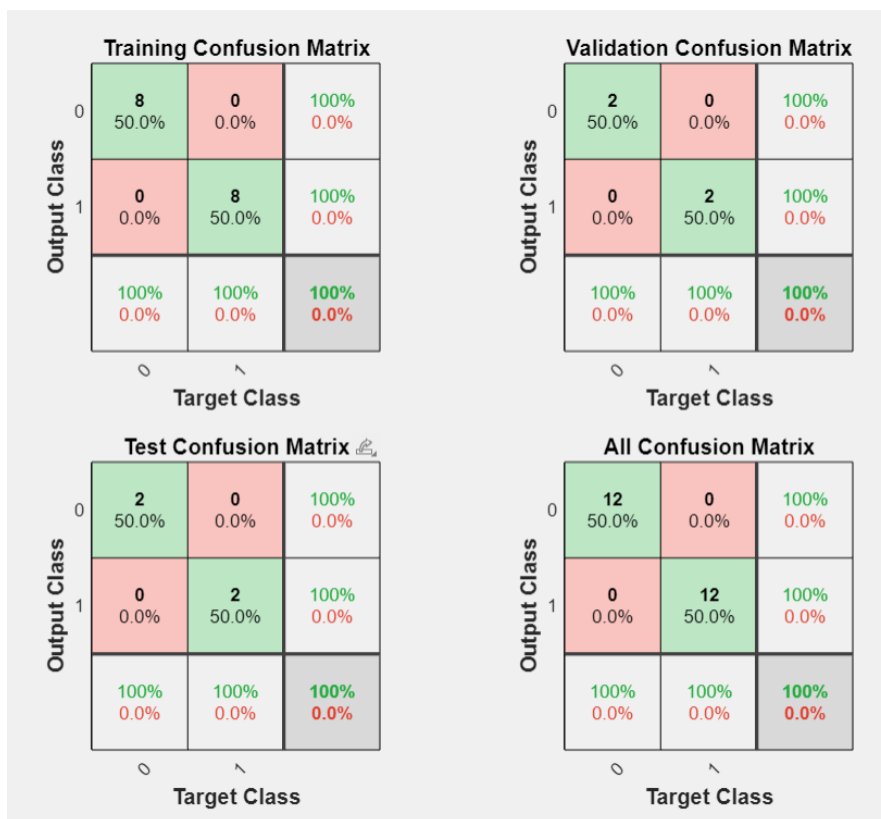
1. Sadržaj rada

Ovaj dokument pokriva zadatak 3. sa ispita iz predmeta algoritamske heuristike april 2024. Struktura rada ce biti po delovima zadatka, tj. deo pod a, pod b i pod c.

2. Neuralne mreze, treniranje neuralne mreze koja emulira MUX2-1

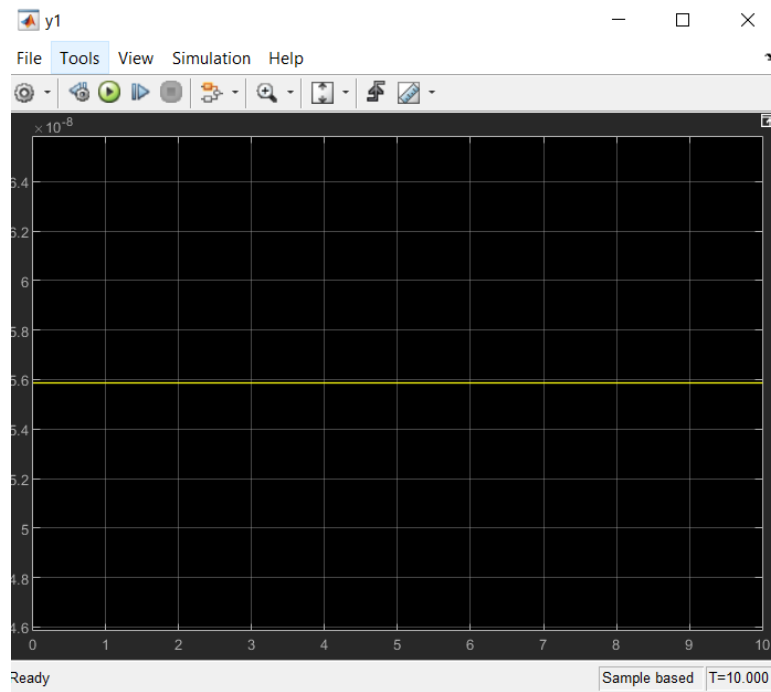
Nervni sistem je sastavljen od neurona koji komuniciraju preko sinapsi. Sam neuron meri slicnost izmedju dva vektora, onog sto mu se pojavio na ulazu i onih enkodiranih iz proslih merenja/iteracija. Ulazi se predstavljaju kao dendriti, jacine kao tezine sinapsi a bias nam predstavlja potencijal. Svaki ulaz se mnozi sa tezinom cime se zatim dobija tezinska suma. Neuralne mreze imaju ogroman spektar primena, medjutim ovaj dokument ce se fokusirati iskljucivo na dati zadatak sa ispita.

Prilozeni matlab kod *mux2_1.m* sadrzi sve moguće kombinacije na ulazima za a,b i sel, i sve vrednosti na izlazima istog. Setovi su replicirani jer Matlab2023 ne dozvoljava broj u *predictor* polju da bude manji od 10. Zatim je pokrenut alat za treniranje mreza za *nnstart*. Izabran je *pattern recognition* i dodeljeni su ulazi i izlaz. Treniranje je radjeno sa defaultnim layer size = 10, i dobijena je sledeca matrica koja pokazuje uspesnost treniranja na slici 1.

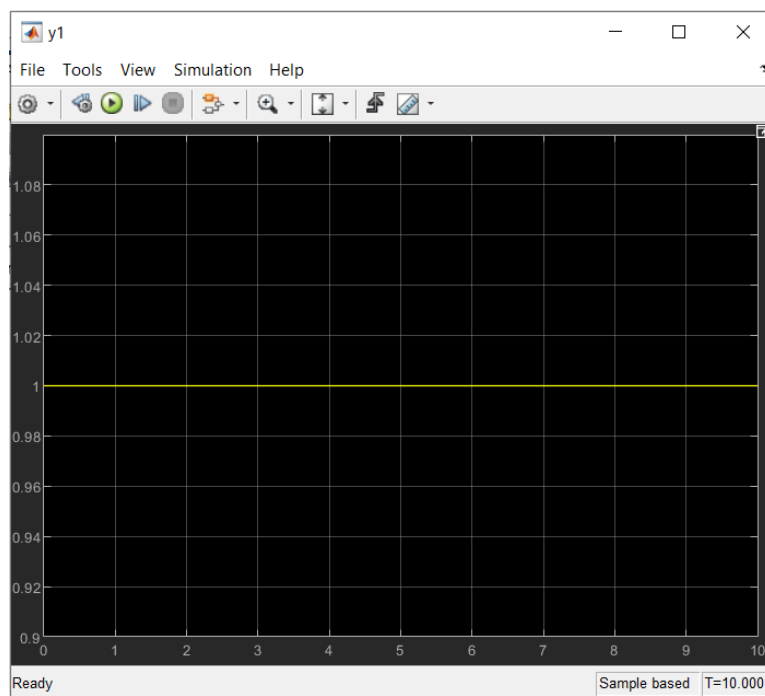


Slika 1 : Uspesno treniranje mreze

Ova mreza je zatim exportovana u Simulink kao poseban blok radi testiranja I dobijeni su sledeci rezultati na slikama 2 I 3.



Slika 2 : Slučaj za ulazni vector $[0,0,0]$ -> Dobijena 0



Slika 3 : Slučaj za ulazni vector $[1,0,0]$ -> Dobijena 1

3. Bulova formula I upotreba prethodno korisnog mux2_1

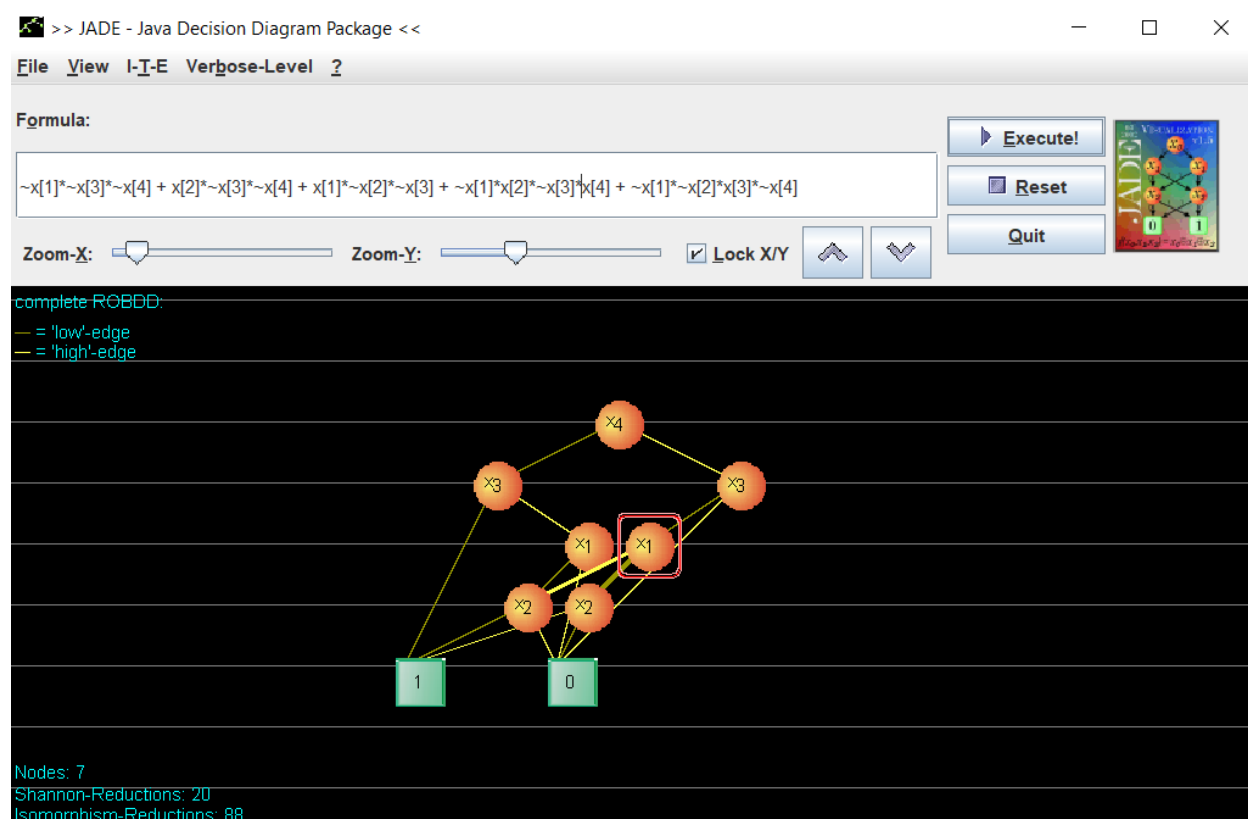
U zadatku pod b je data bulova formula koju je potrebno implementirati u Simulinku. Kako bi se efikasnije pronasao ROBDD ove funkcije koriscen je softverski alat JADE preuzet sa sajta nemackog univerziteta u Bremenu. Pomocu strelice pored opcije Lock X/Y cvorovi su pomerani po nivou kako bi se dobila jednostavnija slika. Bulova funkcija je data u prozoru *Formula* a u lepsem zapisu ona glasi :

$$y = f(a,b,c,d) = \bar{a} \cdot \bar{c} \cdot \bar{d} + b \cdot \bar{c} \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d}$$

Za potrebe kopiranja dacu je I u ovom formatu :

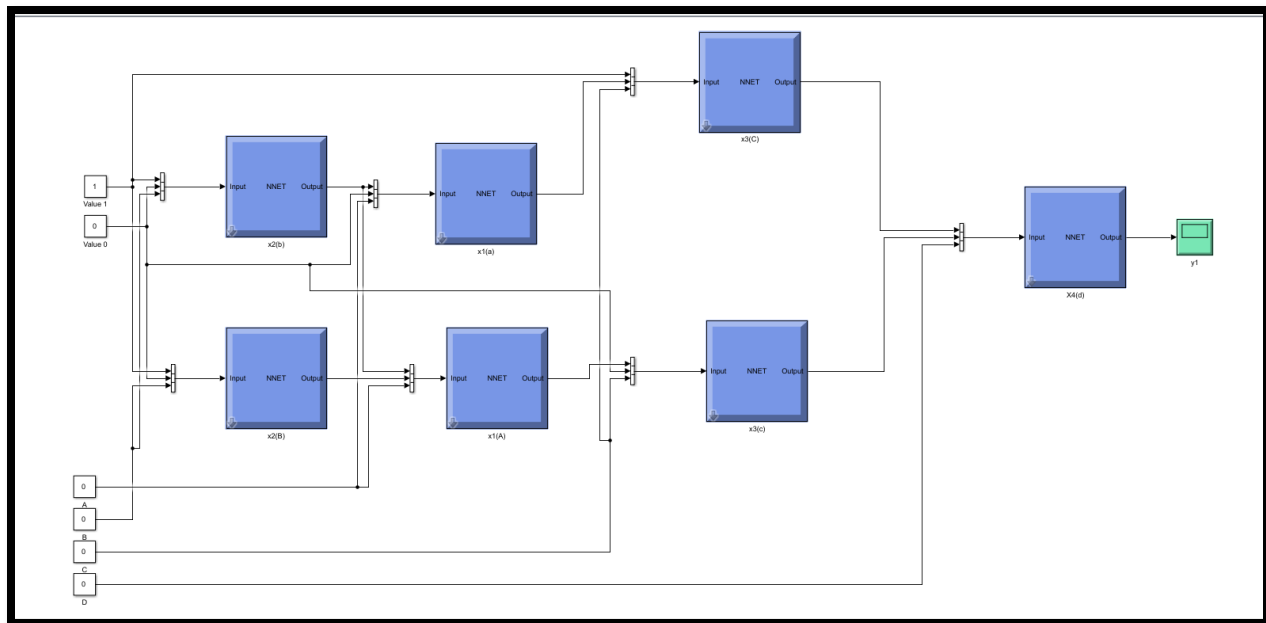
$\sim x[1] * \sim x[3] * \sim x[4] + x[2] * \sim x[3] * \sim x[4] + x[1] * \sim x[2] * \sim x[3] + \sim x[1] * x[2] * \sim x[3] * x[4] + \sim x[1] * \sim x[2] * x[3] * \sim x[4]$

Prozor iz alata JADE I minimizarana funkcija su prikazane na slici 4.



Slika 4 : Rezultat minimizacije JADE alata

Kada se prati napomena sa ispita (sa pdf dokumenta) da se svaki cvor zameni sa kreiranim muxom dobijamo sledeci Simulink model na slici 5.

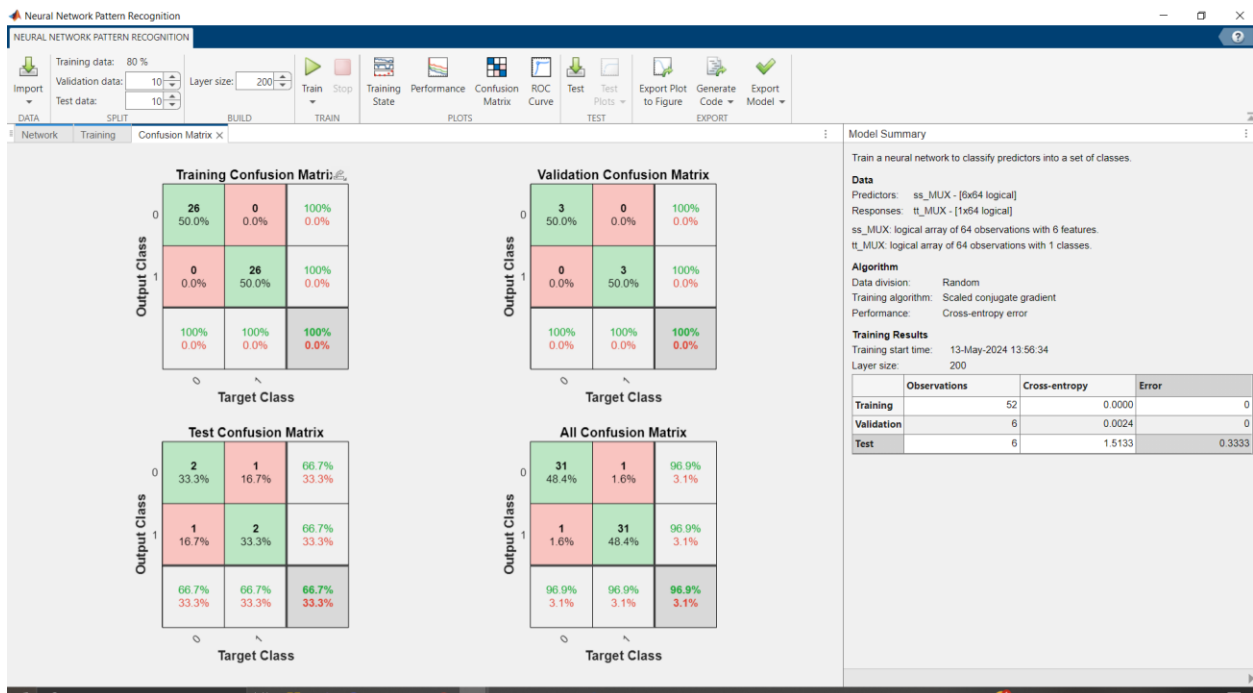


Slika 5 : Dobijeni Simulink model (za deo zadatka pod b)

Kao sto vidimo, koriscene su osnovne *constant* komponente za vrednosti 0 i 1 i za menjanje vrednosti ulaza A, B, C i D date Bulove funkcije. Na ekranu JADE alata cvorovi X1, X2, X3 i X4 su reprezentovani kao A, B, C i D respektivno. Simulink model je podeljen hijerarhijski od nizeg ka visem novi gledajuci sa leva na desno a ime svake komponente je povezano sa nazivima i iz JADE notacije i iz Bulove notacije.

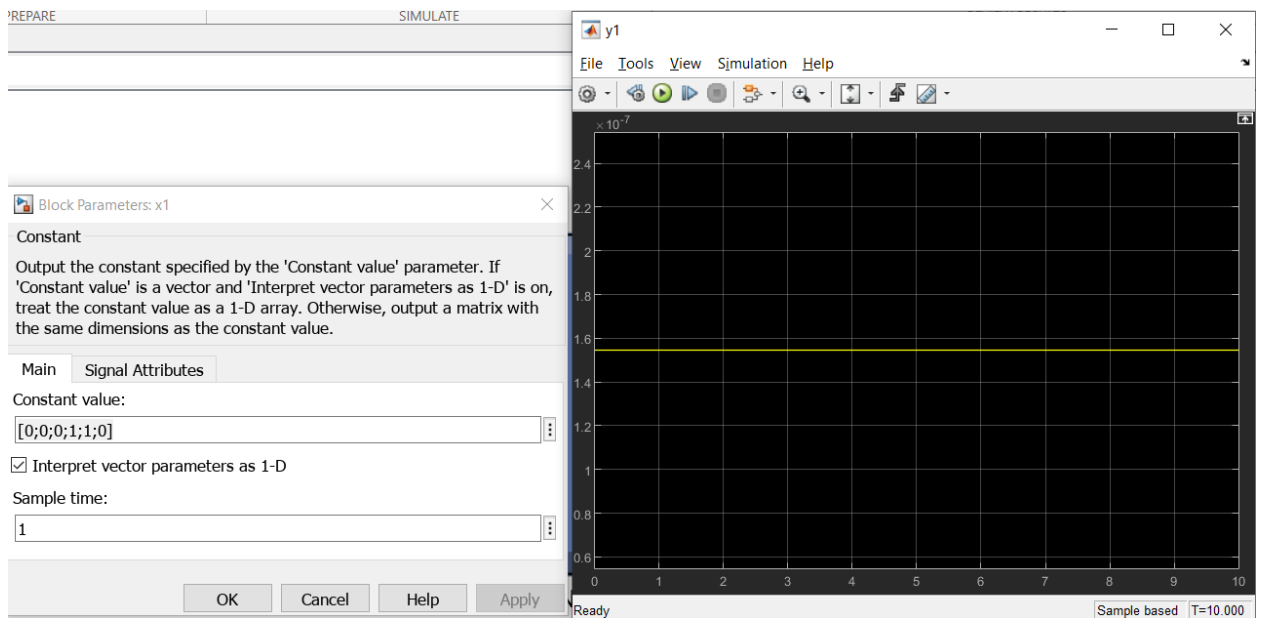
4. Treniranje Multipleksera 4 na 1 i odgovarajuci Simulink model

Za treniranje multipleksera 4 na 1 je koriscen gotovo identican nacin kao i za 2 na 1. Matlab kod se razlikuje samo po vrednostima i broju ulaza. Naime, za mux4-1 postoje 4 informaciona ulazna signala(a,b,c,d) i 2bitni selekcionni signal(sel) koji sam podelio u dva jednobitna kako bi lakse prosao kroz sve kombinacije. Ovo znaci da imamo 6 vrsta u s matrici i potrebno je da budu 64bitne kako bi prosli kroz sve kombinacije tih ulaza (2^6). Koriscena je Matlab sekvenca : `s=logical(dec2bin(0:63, 6) - '0')` kako bi se kreirala na brzi i efikasniji ulazna matrica, a matrica izlaza t je rucno ispisana prolaskom kroz sve vrednosti. Ovde nije bilo potrebe za multipliciranjem jer su ulazi i izlaz dovoljno široki. Treniranje je radjeno na isti nacin, sa manjom modifikacijom parametara *layer size*, *validation data* i *test data* kako bi se dobio dosta dobar rezultat treninga, koji cu prikazati na slici 6.

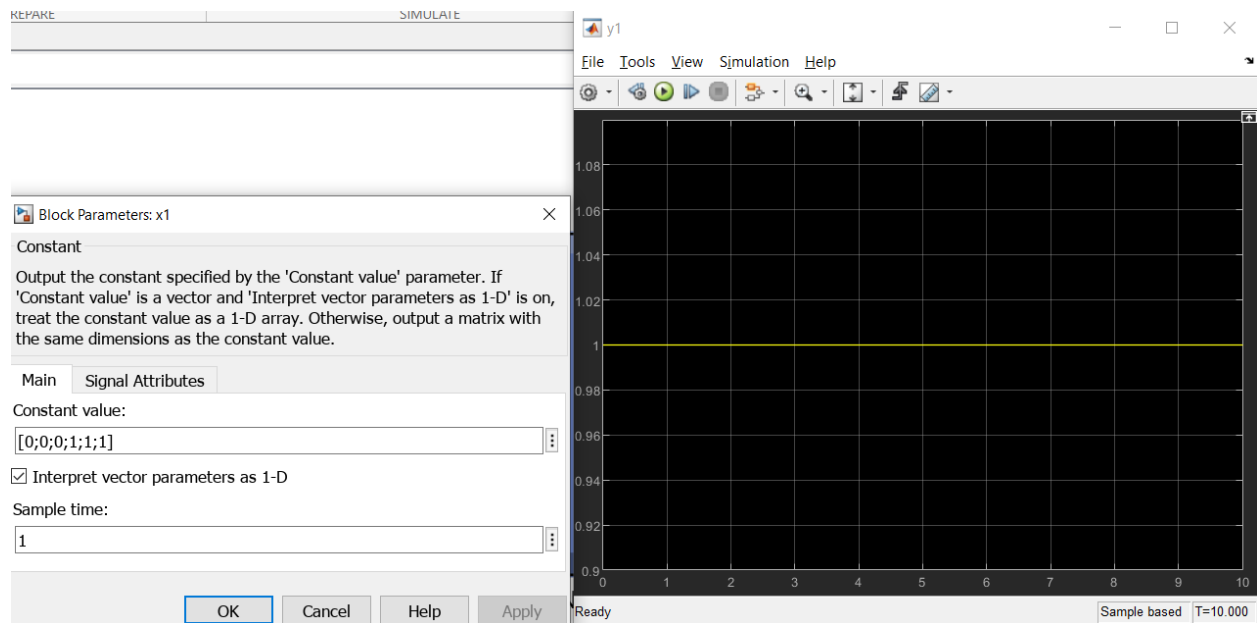


Slika 6 : Rezultati testiranja ANN za mux 4-1

Blok je eksportovan u Simulink I uradjena je provera sa par vrednosti kako bi se potvrdilo korektno testiranje. Prilozicu dva primera na slikama 7 I 8.



Slika 7 : Ocekivana niska vrednost za dati vector – Sel je 10 I ocekuje se treca vrednost koja je 0



Slika 8 : Ocekivana visoka vrednost za dati vektor – Sel je 11 I ocekuje se cetvrta vrednost koja je 1