

7. zadatak

Prosinac, 2018.

U okviru posljednjeg zadatka spojiti ćemo neuro-računarstvo s evolucijskim računanjem: primijeniti ćemo genetski algoritam na učenje umjetne neuronske mreže koja obavlja klasifikacijski zadatak.

1 Uvod

U datoteci `zad7-dataset.txt` dostupan je skup uzoraka za učenje klasifikatorskog sustava. Taj se skup sastoji od 64 uzorka za učenje. Uzorci su izvučeni iz 2D prostora i mogu pripadati jednom od tri razreda: A , B ili C . Da bismo uzorke mogli obrađivati umjetnom neuronskom mrežom, pripadnost razreda kodirana je jednojedinичnim kodom: tako je razred A binarno kodiran kao 100, razred B binarno je kodiran kao 010 a razred C binarno je kodiran kao 001. Stoga je u datoteci svaki uzorak za učenje predstavljen kao niz od pet brojeva: x -koordinata uzorka, y -koordinata uzorka te tri bita koji kodiraju pripadnost točke (x, y) razredu.

U neuronskoj mreži koju ćemo koristiti za klasifikaciju uzoraka za učenje radit ćemo s dva tipa neurona. Neuroni tipa 2 su neuroni koji računaju težinsku sumu (*net*) i potom je propuštaju kroz sigmoidalnu prijenosnu funkciju. Od ovih neurona izgradit ćemo dio mreže koji će se ponašati kao standardna unaprijedna slojevitá neuronska mreža. Ovom vrstom neurona gradili smo neuronsku mrežu za koju smo izveli postupak učenja *backpropagation*.

Neuroni tipa 1 su neuroni koji na svojem izlazu u određenom smislu generiraju mjeru udaljenosti ulaza koji je predöčen neuronu i uzorka koji je pohranjen u neuronu. Preciznije bi bilo reći da su to neuroni koji na izlazu daju mjeru sličnosti ulaznog uzorka i pohranjenog uzorka (o čemu smo govorili kod samoorganizirajućih neuronskih mreža). Neuron koji ima k ulaza svoj bi izlaz y računao prema sljedećem izrazu:

$$y = \frac{1}{1 + \sum_{i=1}^k \frac{|x_i - w_i|}{|s_i|}}.$$

Pri tome su x_i vrijednosti koje su dovedene na ulaze neurona, w_i su vrijednosti koje neuron oduzima od predöčenih ulaza (to su komponente pohranjenog uzorka) a s_i su vrijednosti koje skaliraju utjecaj razlike $x_i - w_i$ na

mjeru udaljenosti. w_i i s_i su parametri neurona baš kao što su kod neurona koji je računao težinsku sumu težine w_i bile nepoznati parametri.

Zadatak 1.

Razmotrite jedan neuron koji ima samo jedan ulaz. Njegov izlaz tada će biti određen izrazom:

$$y = \frac{1}{1 + \frac{|x-w|}{|s|}}.$$

Pretpostavite da je u neuron pohranjena vrijednost $w = 2$. Nacrtajte *na istom grafu* ovisnost $y(x; w = 2)$ za tri slučaja: za $s = 1$, za $s = 0.25$ te za $s = 4$ (svaku različitom bojom ili stilom linije). Za raspon apscise uzmite interval $[-8, 10]$. Razumijete li sada kako s utječe na izlaz neurona y ? Kako će izgledati izlaz neurona koji ima dva ulaza i što se tada kontrolira parametrima s_1 i s_2 ?

Umjetna neuronska mreža koju trebate izgraditi sljedećeg je oblika.

1. sloj je ulazni sloj. Broj neurona u ovom sloju je 2 jer su ulazni uzorci oblika (x, y) . Ovi neuroni ništa ne računaju već dobiveni uzorak na svojim izlazima stavljaju ostatku mreže na raspolaganje.
2. sloj je sloj koji je izgrađen od N_1 neurona tipa 1.
3. sloj i svi ostali slojevi izgrađeni su od neurona tipa 2.

Najmanja moguća arhitektura ovakve mreže tada je $2 \times N_1 \times 3$: to je mreža koja ima jedan ulazni sloj, jedan sloj od N_1 neurona tipa 1 i samo jedan sloj od 3 neurona tipa 2 koji je odmah i izlazni sloj. Zadnji sloj mora imati tri neurona jer su izlazi u uzorcima za učenje predstavljeni s tri broja: svaki izlazni neuron zadužen je za generiranje jedne od tih izlaznih vrijednosti. Primjer bogatije arhitekture bio bi $2 \times N_1 \times 4 \times 3$ koji ima zadnja dva sloja izgrađena od neurona tipa 2. Između slojeva mreža je potpuno povezana: svaki neuron sloja i ima po jedan ulaz prema svakom neuronu sloja $i - 1$.

2 Podatci za učenje

Prije no što nastavite dalje s projektom, riješite sljedeći zadatak.

Zadatak 2. Iskoristite neki gotov program (ili napišite vlastiti program, što god Vam je lakše) kako biste dobili 2D prikaz podataka koje ste dobili za učenje (`zad7-dataset.txt`). Pri tome uzorke različitih razreda prikažite ili različitim simbolom (npr. kvadratić, trokutić, kružić) ili različitom bojom. Ovu sliku spremite kao dio Vaše dokumentacije. Ako ste koristili gotov program, navedite naziv programa.

Proučite dobiveni prikaz. Postoji li kakav uzorak u tim podacima? Jesu li razredi međusobno linearno odvojivi?

Razmotrite neuronsku mrežu čije je arhitektura $2 \times 8 \times 3$ (skriveni sloj od neurona tipa 1, izlazni od neurona tipa 2). Riješite sljedeći zadatak.

Zadatak 3. Kada biste morali ručno odrediti vrijednosti svih parametara upravo zadane neuronske mreže, na koje biste ih vrijednosti postavili i zašto? Čime biste se vodili prilikom određivanja parametara neurona skrivenog sloja a čime prilikom određivanja parametara neurona izlaznog sloja? Nacrtajte tu neuronsku mrežu i na njoj prikažite vrijednosti svih parametara.

3 Genetski algoritam

Napišite genetski algoritam koji ćete primijeniti za pronalazak optimalnih parametara neuronske mreže; ako ste kod iz prethodnih zadataka pisali "modularno", možete iskoristiti Vaš prethodno napisani kod. Ako radite ispočetka, predlažem da mrežu i popratne dijelove modelirate na sljedeći način (nije nužno poslušati).

- Napišite razred **NeuronskaMreza** koji je implementacija neuronske mreže. Kroz konstruktor mreža bi trebala dobiti informaciju o arhitekturi (tako da se može zadati primjerice $2 \times 8 \times 3$, $2 \times 8 \times 4 \times 3$ i slično (dopustite proizvoljan broj slojeva).
- Pojedine vrste neurona zbog efikasnosti nemojte modelirati zasebnim razredima.
- Tijekom konstrukcije mreže stvorite pomoćno polje **double**-ova koje ima onoliko elemenata koliko ima neurona: to polje koristite za pohranu izlaza neurona (i istovremeno kao ulaze prilikom rada s neuronima sljedećeg sloja).
- Nemojte zasebno modelirati skupove parametara koji pripadaju pojedinim neuronima. Umjesto toga, pretpostavite da će svi parametri mreže mreži biti dani izvana kao jedno veliko polje **double**-ova: Vi se prilikom izračuna izlaza mreže morate sami sa sobom dogovoriti na kojim ćete pozicijama u tom polju očekivati da se nalaze pojedini parametri. Primjerice, najjednostavnije je očekivati da se parametri prvog neurona nalaze na prve četiri pozicije, parametri sljedećeg neurona na sljedeće četiri pozicije, itd. i tako sloj po sloj.
- Vaša neuronska mreža može imati metodu kojom pozivatelju dojavljuje koliko parametara toj mreži treba. Primjerice, kod mreže arhitekture $2 \times 8 \times 3$ trebao bi biti $8 * 2 * 2 + 3 * (8 + 1) = 59$.
- Napišite razred **Dataset** koji čuva i nudi pristup do svih uzoraka za učenje. Nudi uobičajene informacije poput "koliko ih ima" te pristup do i -tog uzorka.

- U razredu neuronske mreže napišite metodu `calcOutput` koja prima polje parametara mreže te ulaz i temeljem njega računa i vraća izlaz mreže.
- U razredu neuronske mreže napišite metodu `calcError` koja prima referencu na `Dataset` te na polje parametara mreže: temeljem tih podataka računa i vraća ukupnu pogrešku koju računa srednju kvadratnu pogrešku *mse*:

$$mse = \frac{1}{N} \sum_{s=1}^N \sum_{o=1}^M (t_{s,o} - y_{s,o})^2$$

pri čemu je N broj uzoraka za učenje, M je broj izlaza mreže (u našem slučaju bit će 3), $t_{s,o}$ je o -ta komponenta izlaza s -tog uzorka za učenje, $y_{s,o}$ je izlaz o -tog neurona u izlaznom sloju kada je na ulaz mreže bio narinut s -ti uzorak.

Vezano uz izradu genetskog algoritma, slobodni ste odabrati generacijsku izvedbu ili eliminacijsku izvedbu. Za prikaz rješenja koristite polje decimalnih brojeva: polje treba imati onoliko komponentata koliko odabrana neuronska mreža ima parametara. Implementirajte tri različita operatora križanja i tijekom izvođenja algoritma slučajno birajte koji ćete primijeniti. Implementirajte dvije vrste operatora mutacije:

1. za prvu vrstu operatora definirajte vjerojatnost mutacije komponente rješenja (p_{m1}) i potom se za svaku komponentu pitajte hoćete li je mutirati (u p_{m1} posto slučajeva odgovor treba biti da, inače ne); kad god je odgovor potvrđan, komponenti nadodajte slučajno generirani broj iz normalne distribucije s parametrima $N(0, \sigma_1)$; označimo ovaj operator s $mut_1(p_{m1}, \sigma_1)$; pojam "komponenta rješenja", kako radite s rješenjima koja su polja decimalnih brojeva, podrazumijeva svaki od decimalnih brojeva iz tog polja;
2. za drugu vrstu operatora definirajte vjerojatnost mutacije komponente rješenja (p_{m2}) i potom se za svaku komponentu pitajte hoćete li je mutirati (u p_{m2} posto slučajeva odgovor treba biti da, inače ne); kad god je odgovor potvrđan, komponentu zamijenite slučajno generiranim brojem iz normalne distribucije s parametrima $N(0, \sigma_2)$; označimo ovaj operator s $mut_2(p_{m2}, \sigma_2)$.

Vaš genetski algoritam treba koristiti dva primjerka prvo-opisane vrste mutacije (nazovimo ih M_1 i M_2), te jedan primjerak drugo-opisane vrste mutacije (nazovimo ga M_3). Razlika između M_1 i M_2 je u tome što ćete u M_1 uzeti manju vrijednost za parametar σ , a u M_2 neku veću vrijednost za parametar σ .

Svaki puta kada genetski algoritam treba napraviti mutaciju, mora odabrati jedan od tri prethodno opisana operatora (M_1 , M_2 ili M_3) i odabranim

operatorom treba provesti mutaciju. Da biste ovo napravili, definirajte poželjnost primjene mutacije M_i kao t_i , gdje je t_i nenegativan decimalni broj. Pri tome morate osigurati da je $t_0 + t_1 + t_2 > 0$ (drugim riječima, ne smiju sve poželjnosti biti 0). Definirajmo sada vjerojatnost da ćemo mutaciju obaviti operatorom M_i kao v_i , gdje je:

$$v_i = \frac{t_i}{\sum_{j=1}^3 t_j}.$$

Time ćemo, ako kao poželjnosti postavimo $t_1 = t_2 = t_3 = 1$ dobiti vjerojatnosti primjene svakog od operatora kao $\frac{1}{3}$. Ako bismo postavili $t_1 = 2$, a $t_2 = t_3 = 1$, vjerojatnost primjene mutacije M_1 bila bi $\frac{2}{4}$ tj. $\frac{1}{2}$, a vjerojatnost primjene mutacije M_2 odnosno M_3 bila bi $1/4$. Stoga, kada trebate raditi mutaciju, u skladu s danim vjerojatnostima odaberite pa primijenite operator. Prilikom pokretanja programa (ili negdje u metodi `main`) treba biti moguće podesiti poželjnosti, na temelju čega program sam treba odrediti vjerojatnosti. Razmislite: ima li više smisla da je v_1 bliži 0 ili bliži 1? Što biste rekli za v_2 te za v_3 ?

Tablica 1: Pregled parametara vezanih uz operatore mutacije. Korisnik zadaje vrijednosti za stupce $p_{m,i}$ i t_i .

Mutacija	$p_{m,i}$	t_i	v_i
M1	$p_{m,1}$	t_1	v_1
M2	$p_{m,2}$	t_2	v_2
M3	$p_{m,3}$	t_3	v_3

Optimizacijski problem koji rješavamo, na način kako je postavljen, direktno definira funkciju kazne: to je upravo *mse*. Kada stvorite novo rješenje (dijete), polje parametara koje to dijete predstavlja pošaljite neuronskoj mreži i zatražite izračun *mse*-a; njega transformirajte u funkciju dobrote i dalje u algoritmu postupajte kao da radite maksimizaciju.

Uvjet zaustavljanja postupka učenja treba biti dosegnut maksimalni broj iteracija/generacija ili postignut *mse* koji je manji od 10^{-7} . Ako radite eliminacijski GA, tada "generacije" brojite kao broj iteracija podijeljen s veličinom populacije. Npr. ako eliminacijski GA radi s 50 jedinki, tada ćemo reći da je nakon 50 iteracija prošla jedna generacija, nakon još 50 sljedeća generacija, itd.

Istestirajte Vašu implementaciju. Zapišite sve parametre algoritma (ne neuronske mreže) koje morate podesiti prije pokretanja algoritma. Pokušajte pronaći njihove prihvatljive vrijednosti.

Po završetku učenja na naučenu neuronsku mrežu dovedite uzorak po uzorak, izračunajte izlaze i svaki izlaz koji je manji od 0.5 spustite na 0 a preostale dignite na 1. Time ćete za svaki ulazni uzorak dobiti jedan binarni trobitni izlaz. Usporedite ga s ispravnim trobitnim izlazom koji je pohranjen

u uzorku za učenje. Za svaki uzorak iz skupa uzoraka za učenje na zaslon ispišite izlaz koji je pridružen uzorku, izlaz koji ste dobili od mreže opisanom binarizacijom te li taj uzorak korektno klasificiran ili nije. Potom i sumarno ispišite broj korektno klasificiranih i broj pogrešno klasificiranih uzoraka. Napomena: ovakvu interpretaciju izlaza radite samo **nakon** što je učenje završeno. Tijekom učenja za izračun mse -a koristite izravno vrijednosti koje se dobiju na izlaznim neuronima mreže bez zaokruživanja na 0 ili 1.

4 Preostali zadatci

Jednom kada ste se uvjerali da Vaša implementacija genetskog algoritma radi dobro, rješite sljedeće zadatke.

Zadatak 4. Naučite optimalne parametre mreže arhitekture $2 \times 8 \times 3$. Nacrtajte novu sliku na kojoj se vide svi ulazni uzorci s indikacijom razreda te uzorci koje je GA naučio za svaki neuron tipa 1. Prokomentirajte gdje se nalaze naučeni uzorci i je li to u skladu s očekivanjem. Kakve je vrijednosti parametara s_i naučio GA? Jesu li iste za x i y komponentu ili su različite? Objasnite!

Nacrtajte novu sliku na kojoj se vide svi neuroni neuronske mreže, pozicije koje su naučene u neuronima tipa 1 te vrijednosti težina za neurone tipa 2. Uočavate li kakvu pravilnost u tim težinama? Možete li je objasniti?

Zadatak 5. Naučite optimalne parametre mreže arhitekture $2 \times 8 \times 4 \times 3$. Je li postupak učenja trajao dulje ili kraće u odnosu na prethodnu arhitekturu? Možete li objasniti zašto?

Pogledajte naučene parametre u neuronima tipa 1 za ovaj slučaj. Možete li ih objasniti?

Zadatak 6. Možete li dobiti ispravnu klasifikaciju svih uzoraka u arhitekturi koja ima $N_1 < 8$? Provjerite to na arhitekturi $2 \times 6 \times 4 \times 3$. Na kraju (uspješnog ili neuspješnog) postupka učenja pogledajte za najbolje rješenje parametre u neuronima tipa 1 za ovaj slučaj. Što smo izgubili u odnosu na mrežu iz zadatka 4?

5 Izvještaj

U okviru rješenja ovog dijela projekta nužno je pripremiti datoteku s programskim rješenjem te izvještaj. Datoteku s programskim rješenjem trebate uploadati na Ferka. Izvještaj ne uploadate; njega trebate isprintati, zaklamati i imati sa sobom u terminu predaje vježbe.

Datoteka s programskim rješenjem treba biti ZIP arhiva u kojoj su smješteni svi izvorni kodovi te izvršna verzija programa. Ako to nije "exe" datoteka koja na dvoklik pokreće postupak učenja mreže one najjednostavnije arhitekture, pripremite i **README.txt** u kojoj ćete opisati kako se program pokreće.

U izvještaj uključite odgovore na sva ovdje postavljena pitanja (zadatci 1-6) te tražene slike.

5.1 Završne napomene

Ovaj zadatak rješava se samostalno i "timsko" rješavanje nije dopušteno. Ako imate pitanja ili problema oko rješavanja zadatka, potražite nas na Zavodu i dođite na konzultacije.