

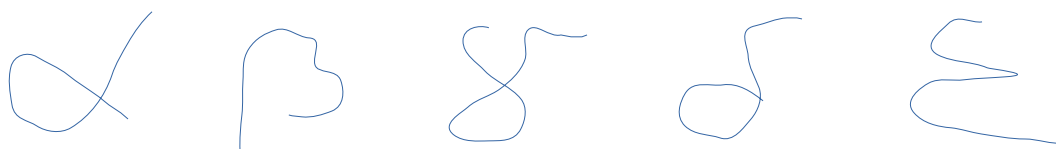
# Neizrazito, evolucijsko i neuroračunarstvo: zadatak 5

Tema ovog zadatka je učenje unaprijedne slojevite potpuno-povezane neuronske mreže (engl. *Multi-Layered Feed-forward Artificial Neural Network*).

## Opis problema

U ovom zadatku učite ćete neuronsku mrežu da klasificira geste koje korisnik radi mišem na ekranu. Pri tome ćemo razmatrati samo geste koje se sastoje od jednog kontinuiranog pokreta.

U nastavku je prikazano nekoliko slova grčkog alfabeta koje je moguće napisati jednim kontinuiranim pokretom miša (jednom gestom).



Želimo napraviti umjetnu neuronsku mrežu koja će moći klasificirati geste.

## Priprema uzoraka za učenje

Koristeći neki od programskih jezika kojim vladate napišite program koji će korisniku prikazati prozor u kojem će korisnik mišem moći napraviti gestu. Program treba pratiti miš od trenutka kada je korisnik pritisnuo tipku miša, kroz sve pokrete miša sve do trenutka kada korisnik otpusti tipku miša. Uobičajeno je moguće od prozora dobiti informaciju o  $(x,y)$  koordinatama točaka kroz koje prolazi pokazivač miša. Skupite sve točke kroz koje je miš prošao u jedan vektor točaka (redosljed je bitan!). Ovisno o duljini geste, vektor će imati različit broj elemenata.

Potom izračunajte aritmetičku sredinu svih točaka u vektoru (označimo je s  $T_c$ ; od sada na dalje komponente točaka pamтите kao decimalne brojeve). Potom korigirajte vektor točaka tako da svim točkama oduzmete točku  $T_c$ ; u ovom novom vektoru  $(x,y)$  koordinate su decimalni brojevi. Ovom operacijom iz zapisa geste uklonili informaciju o dijelu prozora u kojem je nacrtana tako da će sve geste koje jednako izgledaju (a iste su visine i širine), neovisno o tome jesu li nacrtane u lijevom dijelu prozora, na sredini prozora ili negdje drugdje, sada imati jednak zapis.

Potom izračunajte maksimalnu udaljenost  $x$  i  $y$  koordinata svih točaka koje su u vektoru od nule ( $i = 0$  do broj točaka u vektoru minus jedan):

$$mx = \max |x_i|$$

$$my = \max |y_i|$$

i označite veći od ta dva broja:

$$m = \max(mx, my)$$

Ponovno transformirajte sve točke u vektoru tako da  $x$  i  $y$  koordinate podijelite s  $m$ . Ovime ste osigurali da su sve  $x$  i  $y$  koordinate unutar raspona  $[-1, 1]$ , s time da su sve proporcionalno skalirane: ako je gesta bila šira no viša, i nakon skaliranja će ostati šira i viša.

Izračunajte duljinu tako dobivene geste: duljinu aproksimirajte sumom udaljenosti prve i druge točke, udaljenosti druge i treće točke, udaljenosti treće i četvrte točke... Neka je ukupna duljina geste  $D$ .

Označimo s  $M$  broj reprezentativnih točaka geste. Kao reprezentativne točke geste uzimamo točke koje su od prve točke udaljene za  $k \cdot D/(M-1)$ ,  $k=\{0,1,2,\dots,M-1\}$ . Ako se na toj udaljenosti ne nalazi niti jedna točka, ili uzmite najbližu, ili interpolirajte temeljem točaka u vektoru.

Ideja ovog postupka je gestu koja je snimljena kao varijabilni broj točaka predstaviti fiksnim brojem značajki. Stoga uzimamo točke po zamišljenoj krivulji koju opisuje gesta na uniformnim razmacima, tako da krenemo s prvom točkom i potom na jednakim razmacima po krivulji odaberemo još  $M-1$  točku. Stoga, ako je duljina geste  $D$ , biramo točke na razmacima  $D/(M-1)$ .

Nakon provedbe ovog postupka za svaku ćete gestu imati  $2 \cdot M$  brojeva ( $M$  x-koordinata i  $M$  y-koordinata).

Uporabom Vašeg programa snimite po 20 gesti svakog od prethodno prikazanih slova grčkog alfabeta i na opisani način izlučite značajke. Nemojte se truditi da svaka gesta izgleda identično: ideja je na različite načine "nacrtati" svako od slova i potom pustiti neuronsku mrežu da otkrije pravilnosti u načinu crtanja.

Kako imamo 5 slova, pripadnost razredima prikazat ćemo pet-bitnim jednojedinim kodom. Svim uzorcima koji čine gestu *alfa* pridjelit ćemo kod 1,0,0,0,0; svim uzorcima koji čine gestu *beta* pridjelit ćemo kod 0,1,0,0,0, i tako do posljednjeg slova koje će imati kod 0,0,0,0,1.

Pohranite sve ove uzorke u jednu datoteku: jedan redak po uzorku,  $2 \cdot M$  decimalnih brojeva koji čine reprezentativne točke geste te 5 decimalnih brojeva koji čine pridjeljeni kod.

Neuronsku mrežu trenirat ćemo da na temelju značajki geste na izlazu konstruira kod te geste.

## Neuronska mreža

Koristit ćemo unaprijednu višeslojnu neuronsku mrežu čija je arhitektura općenito  $a \times b \times \dots \times c$ ; primjerice, mreža  $2 \times 3 \times 2 \times 5$  bi označavala mrežu koja ima dva neurona u ulaznom sloju, tri u prvom skrivenom, dva u drugom skrivenom te pet u izlaznom sloju. Za potrebe ovog zadatka bit će nam prikladne arhitekture čiji je prvi broj  $2 \cdot M$  a zadnji 5 (zašto?).

Korisnik kao argument programa treba zadati putanju do datoteke s pohranjenim uzorcima, broj  $M$ , arhitekturu mreže (broj skrivenih slojeva treba vidjeti iz toga što je uneseno) te oznaku algoritma učenja (1, 2 ili 3); provjerite da prvi broj u arhitekturi odgovara  $2 \cdot M$  a posljednji da je 5.

Svi neuroni koriste sigmoidalnu prijenosnu funkciju.

## Algoritam učenja

U okviru zadatka potrebno je implementirati sljedeće algoritme učenja:

1. *Algoritam propagacije pogreške unatrag* (engl. *Backpropagation*). Algoritam koristi grupno učenje: težine se ažuriraju isključivo nakon što su viđeni svi uzorci te je na temelju njih izračunat pravi gradijent.
2. *Stohastički algoritam propagacije pogreške unatrag* (engl. *Stochastic Backpropagation*). Koristi se pojedinačno učenje. Na temelju svakog uzorka napravi se procjena gradijenta i odmah obavi korekcija težina.

3. *Algoritam propagacije pogreške unatrag uz male grupe* (engl. *Mini-batch Backpropagation*). Kako imamo po 20 uzoraka za svaki razred, oformite grupu koja se sastoji od prva dva uzorka iz svake grupe (ukupno  $2 \cdot 5 = 10$  uzoraka), na temelju nje izračunajte gradijent i napravite korekciju. Potom uzmite sljedeća dva uzorka iz svake grupe (sljedećih 10 ukupno), na temelju nje izračunajte gradijent i napravite korekciju. I tako redom. Ovdje će procjena gradijenata biti kvalitetnija u odnosu na pojedinačno učenje a opet manje je zahtjevna nego kod klasičnog (grupnog) algoritma.

Zgodna prezentacija povezana s ovim gradivom:

[http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf)

Prilikom učenja algoritam pokušava smanjiti srednju kvadratnu pogrešku nad skupom za učenje. Stoga tijekom izvođenja postupka učenja periodički prikazujte tu informaciju (ispisom u konzolu ili drugačije) kako biste mogli pratiti uči li još algoritam ili je došlo do stagnacije. Uočite da želite ispisati srednju kvadratnu pogrešku nad čitavim skupom uzoraka za učenje: nju treba izračunati uz držanje svih težina mreže fiksima.

Jednom kad mreža završi postupak učenja (omogućite korisniku da to nekako odredi: pogreška do koje treba stići, max. broj epoha, nekako drugačije?), ponovno prikažite prozor u kojem korisnik može zadavati geste i sada svaku gestu klasificirajte naučenom mrežom i prikažite utvrđeni razred.

Kao razred u koji je mreža klasificirala gestu uzmite onaj razred koji odgovara izlazu na kojem je mreža generirala najveću izlaznu vrijednost.

Implementacijski naputak: uočite što je zajedničko svim trima algoritmima učenja. Iskoristite zajedničke dijelove kako ne biste pisali tri kopije (gotovo) jednakih algoritama.

Pri pisanju modela neuronske mreže napravite model koji omogućava definiranje parametara arhitekture pa algoritam učenja najprije isprobajte na jednostavnijim zadacima; primjerice, pokušajte naučiti mrežu  $1 \times 6 \times 1$  da aproksimira  $f(x) = x^2$  (neka su uzorci za učenje primjerice  $\{(-1,1), (-0.8, 0.64), (-0.6,0.36), (-0.4,0.16), (-0.2,0.04), (0,0), (0.2,0.04), (0.4,0.16), (0.6,0.36), (0.8,0.64), (1,1)\}$ ). Potom promijenite arhitekturu na neku s dva skrivena sloja pa vidite radi li Vaša implementacija algoritma korektno.

Stopu učenja s kojom radi algoritam ili dopustite da korisnik zada ili fiksirajte u Vašem programu ili je ažurirajte na neki način (tu imate slobodu isprobati sve što Vam padne na pamet).