

Prepoznavanje gesti umjetnom neuronskom mrežom

Kolegij: Umjetna inteligencija (AG 2014/2015); ZEMRIS/FER

1. Snimanje geste

Napravili smo program koji korisniku nudi grafičko korisničko sučelje za unos geste. Korisnik pritisne tipku miša, napravi pokret mišem i otpusti tipku. Program “hvata” sve pozicije kroz koje je miš prošao i to sprema kao vektor 2D točaka. Ekranski koordinatni sustav ima ishodište u gornjem lijevom uglu; pozitivni smjer osi x je u desno, pozitivni smjer osi y prema dolje (y-koordinata raste prema dolje što je obrnuto od uobičajenog matematičkog koordinatnog sustava).

Primjerice, neka smo mišem napravili gestu koja odgovara znamenici 2.



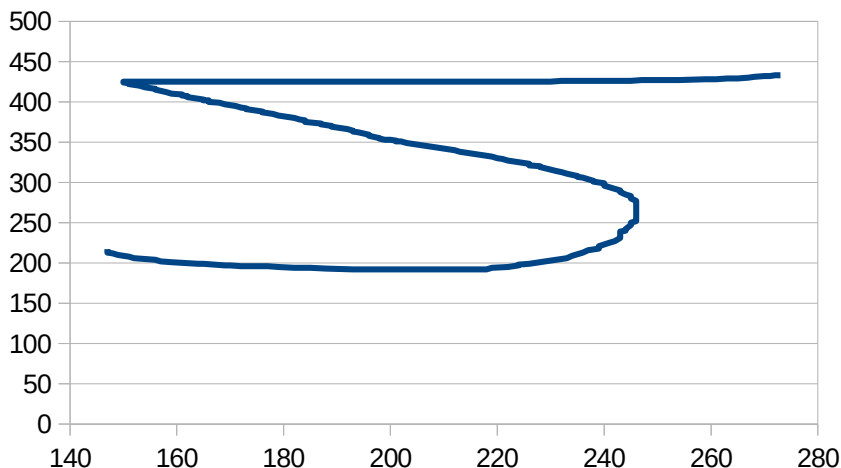
Uhvatili smo 215 točaka (x,y) kroz koje je prošao miš:

147	217
147	216
147	215
147	214
147	213
148	212
149	210
...	
268	431
270	432
271	432
272	433
273	433
273	433

Označimo ovaj skup točaka s $P = \{P_1, \dots, P_n\}$, gdje je $P_i = (x_i, y_i)$.

2. Predobrada podataka

Nacrtamo li prikupljene točke u uobičajenom koordinatnom sustavu, imamo:



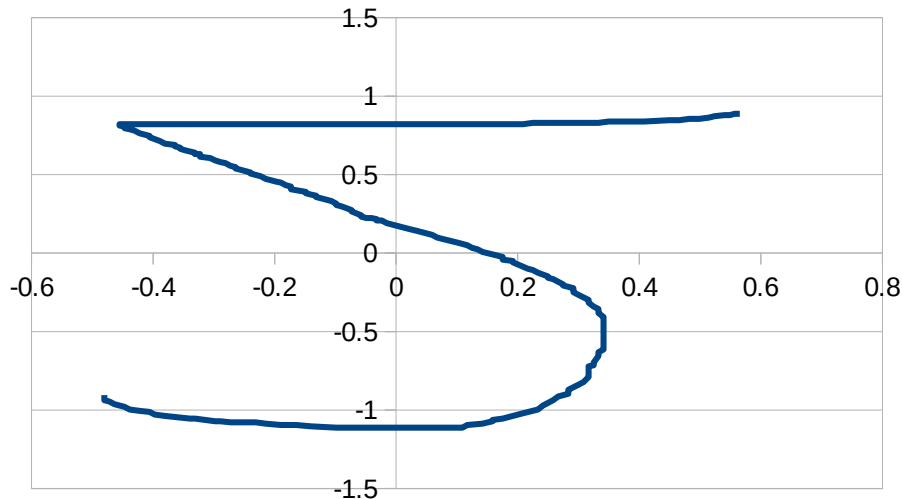
Kako bismo iz geste uklonili ovisnost o dijelu ekrana na kojem je napravljena te o njezinoj veličini, prikupljene točke normaliziramo sljedećim postupkom.

- Računamo srednju vrijednost x i y koordinate nad prikupljenim točkama. Označimo ih x_{avg} i y_{avg} .
- Utvrđujemo minimalnu i maksimalnu x i y koordinatu: x_{min} , x_{max} , y_{min} , y_{max} .
- Neka je $M = \max(x_{max} - x_{min}, y_{max} - y_{min})$.
- Računamo normalizirani skup točaka $T = \{T_1, \dots, T_n\}$, gdje je
$$T_{i,x} = 2 \cdot (P_{i,x} - x_{avg}) / M \quad \text{i} \quad T_{i,y} = 2 \cdot (P_{i,y} - y_{avg}) / M .$$

Time dobivamo skup točaka:

-0.4802856315738686	-0.9046801119366976
-0.4802856315738686	-0.9129788671234197
-0.4802856315738686	-0.9212776223101417
-0.4802856315738686	-0.9295763774968636
-0.4802856315738686	-0.9378751326835856
-0.47198687638714665	-0.9461738878703077
-0.46368812120042463	-0.9627713982437516
...	
0.5238637460194924	0.8712534980218086
0.5404612563929364	0.8795522532085306
0.5487600115796584	0.8795522532085306
0.5570587667663803	0.8878510083952526
0.5653575219531023	0.8878510083952526
0.5653575219531023	0.8878510083952526

koji je ilustriran na sljedećoj slici:



Da bismo geste prepoznavali umjetnom neuronskom mrežom, prikladno je svaku gestu predstaviti nizom brojeva fiksne duljine. Stoga ćemo od svih prikupljenih točaka pokušati uzeti fiksni broj uzoraka koji nacrtanu gestu dijele na segmente jednake duljine.

Duljinu dijela geste od početne točke do L-te točke računamo kao sumu euklidskih udaljenosti susjednih točaka koje čine gestu, do L-te točke:

$$len(L) = \sum_{i=2}^L \sqrt{(T_{i,x} - T_{i-1,x})^2 + (T_{i,y} - T_{i-1,y})^2}$$

Ukupnu duljinu geste računamo kao sumu euklidskih udaljenosti susjednih točaka koje čine gestu:

$$totalLen = \sum_{i=2}^n \sqrt{(T_{i,x} - T_{i-1,x})^2 + (T_{i,y} - T_{i-1,y})^2} = len(n)$$

Za prethodno prikazanu znamenku ukupna duljina je 4.059601006165824.

Neka krivulju želimo podijeliti na K-1 segmenata (npr. K=10). Tražimo točke R_1 do R_K . Točka R_1 će biti jednaka početnoj točki geste T_1 a točka R_K krajnjoj točki geste T_n . Točku T_j , $1 < j < K$, ćemo odrediti kao onu točku T_p za koju je:

$$len(p) \approx \frac{j-1}{K-1} \cdot totalLen$$

Ako takva točka ne postoji, odnosno ako je

$$len(p) < \frac{j-1}{K-1} \cdot totalLen < len(p+1)$$

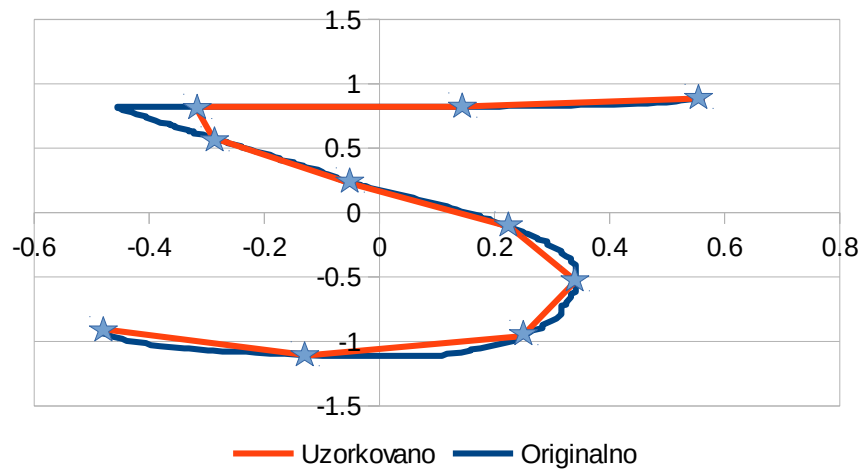
napraviti ćemo linearnu interpolaciju:

$$R_j = T_p + (T_{p+1} - T_p) \cdot \frac{\frac{j-1}{K-1} \cdot totalLen - len(p)}{len(p+1) - len(p)}$$

gdje točke gledamo kao vektore. Npr. Za K=10 dobivamo točke:

-0.4802856315738686	-0.9046801119366976
-0.1234318476465665	-1.1071712006723655
0.2482766586305006	-0.9579289755113617
0.34129113191160854	-0.5305679490905975
0.22656028031020756	-0.11090305603680023
-0.0474092494372347	0.2239505934574932
-0.2892685742125661	0.580651378419842
-0.3192840306088386	0.8214609669014766
0.13178274785403182	0.8214609669014766
0.5653575219531023	0.8878510083952526

koje su prikazane na sljedećoj slici:



Na ovaj način dobivamo konačan vektor duljine $2*K$: $[R_{1,x}, R_{1,y}, R_{2,x}, R_{2,y}, \dots, R_{K,x}, R_{K,y}]$, odnosno u slučaju da je $K=10$, dvadeset-komponentne vektore.

3. Prikupljanje skupa uzoraka za učenje

Pretpostavimo da želimo prepoznavati geste koje odgovaraju znamenkama. Za svaku znamenku potrebno je prikupiti određen broj uzoraka i njih predobraditi na prethodno opisani način.

Svakom uzorku potrebno je dodijeliti jedinstveni vektor koji predstavlja razred kojem pripada gesta (u ovom slučaju znamenka) i koji ćemo htjeti da neuronska mreža nauči.

Za potrebe ovog primjera koristit ćemo takozvano *1-hot* kodiranje: imamo li Q razreda gesti, razred ćemo predstaviti kao Q -dimenzijski vektor u kojem su sve komponente postavljene na 0 osim jedne (koja odgovara razredu). Primjerice, ako želimo razlikovati 10 znamenaka, imamo 10 razreda pa ćemo koristiti vektore:

$[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ (za znamenku 0)

$[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]$ (za znamenku 1)

$[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$ (za znamenku 2)

...

$[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$ (za znamenku 9)

4. Učenje neuronske mreže

Mrežu učimo tako da minimiziramo ukupnu funkciju pogreške koja je definirana kao suma kvadratnog odstupanja željenog izlaza i izlaza koji daje mreža (po svim izlazima i svim uzorcima za učenje, usrednjena po svim uzorcima).

Za to možemo koristiti Backpropagation, stohastički Backpropagation, otporni Backpropagation ili poboljšani otporni Backpropagation: demonstracijski program koji Vam stoji na raspolaganju podržava sva četiri algoritma i dostupan je na adresi:

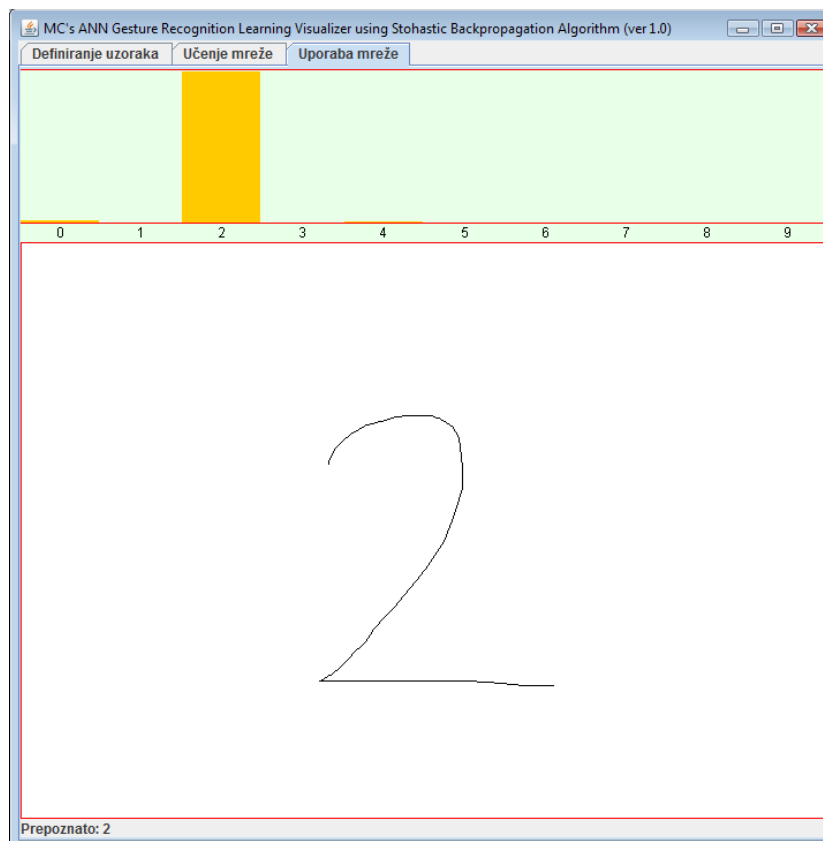
<http://java.zemris.fer.hr/nastava/ui/ann/ffmlp-ann.zip>

Alternativno, neuronsku mrežu možemo učiti i drugim algoritmima, poput algoritama evolucijskog računanja (evolucijski algoritmi, algoritmi rojeva, ...).

Isprobajte različite arhitekture mreže. U programu možete namještati samo broj neurona po skrivenim slojevima, s obzirom da je broj neurona ulaznog sloja određen dimenzionalnošću uzoraka koje dovodimo na mrežu a broj neurona izlaznog sloja dimenzionalnošću vektora koji koristimo za reprezentaciju razreda uzorka.

5. Uporaba mreže

Jednom naučenu mrežu možete isprobati. U programu na posljednoj kartici možete unositi geste a naučena mreža će ih prepoznavati.



Pitanja:

1. Što će mreža napraviti za gestu koja nije slična niti jednom od uzoraka razreda?
2. Zašto?
3. Kako biste popravili "kvalitetu" naučenoga?
4. Kakav utjecaj ima veći broj slojeva ili neurona po slojevima na brzinu učenja?
5. Kakav utjecaj ima veći broj slojeva ili neurona po slojevima na sposobnost generaliziranja (drugim riječima, kakve se pogreške javljaju pri uporabi složenijih mreža? Probajte!).
6. Kada biste ovakvu mrežu htjeli koristiti u računalnom programu koji bi omogućavao takav način unosa, kako biste omogućili da Vas program pita ako nije siguran u razred napravljene geste?
7. Prikazan način izrade vektora koji reprezentira pojedini razred svakako nije jedini mogući.
 1. Koje su druge mogućnosti?
 2. Što se time dobiva a što gubi?
 3. Koliki je minimalni broj izlaza neuronske mreže koju bismo koristili za prepoznavanje gesti znamenaka? Obrazložite Vaše razmišljanje.