

Rukom pisani brojevi

Jelena Držaić
Sveučilište u Zagrebu
PMF-Matematika
Umjetna inteligencija

13. siječnja 2016.

Sadržaj

- 1 Uvod
- 2 Problemi implementacije
- 3 Varijante problema
- 4 pristupi rješavanju
- 5 Algoritmi

Neka od pitanja na koja ćemo pokušati odgovoriti kroz ovu prezentaciju su:

- 1 Zašto proučavamo dani problem?

Neka od pitanja na koja ćemo pokušati odgovoriti kroz ovu prezentaciju su:

- 1 Zašto proučavamo dani problem?
- 2 Koji su problemi konstrukcije algoritma za prepoznavanje rukom pisanih brojeva?

Neka od pitanja na koja ćemo pokušati odgovoriti kroz ovu prezentaciju su:

- 1 Zašto proučavamo dani problem?
- 2 Koji su problemi konstrukcije algoritma za prepoznavanje rukom pisanih brojeva?
- 3 Kako predstaviti rukom pisane brojeve u programu?

Neka od pitanja na koja ćemo pokušati odgovoriti kroz ovu prezentaciju su:

- ➊ Zašto proučavamo dani problem?
- ➋ Koji su problemi konstrukcije algoritma za prepoznavanje rukom pisanih brojeva?
- ➌ Kako predstaviti rukom pisane brojeve u programu?
- ➍ Kako pristupiti konstrukciji algoritma?
- ➎ Možemo li relativno jednostavnim algoritmom postići zadovoljavajuće rezultate?

Zašto je bitno imati softver za prepoznavanje rukom pisanih brojeva?

Primjena je mnogo.

- Prepoznavanje poštanskih brojeva kod sortiranja pisama u poštanskim službama.

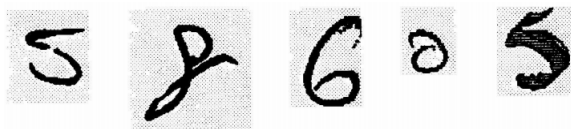
Zašto je bitno imati softver za prepoznavanje rukom pisanih brojeva?

Primjena je mnogo.

- Prepoznavanje poštanskih brojeva kod sortiranja pisama u poštanskim službama.
- Detektiranje rukom napisanih iznosa kod bankovnih transakcija.

Zašto je bitno imati softver za prepoznavanje rukom pisanih brojeva?
Primjena je mnogo.

- Prepoznavanje poštanskih brojeva kod sortiranja pisama u poštanskim službama.
- Detektiranje rukom napisanih iznosa kod bankovnih transakcija.
- Prepoznavanje rukom napisanih matematičkih formula.



- Pretpostavimo da nam je dan rukom napisan broj *504192*.
Većina nas bez previše razmišljanja prepoznaje broj jednak 504192.

- Pretpostavimo da nam je dan rukom napisan broj *504192*.
Većina nas bez previše razmišljanja prepoznaje broj jednak 504192.
- Za računalno je problem puno kompleksniji.
 - ▶ Kako zapisati zahtjev "broj je jednak 3 ako se sastoji od dva polukružna oblika, položena vertikalno jedan na drugome"?
 - ▶ Kako računalo natjerati da razlikuje rukom pisane znamenke 1 i 7?

- Pretpostavimo da nam je dan rukom napisan broj *504192*.
Većina nas bez previše razmišljanja prepoznaje broj jednak 504192.
- Za računalno je problem puno kompleksniji.
 - ▶ Kako zapisati zahtjev "broj je jednak 3 ako se sastoji od dva polukružna oblika, položena vertikalno jedan na drugome"?
 - ▶ Kako računalo natjerati da razlikuje rukom pisane znamenke 1 i 7?

Možemo pokušati program konstruirati tako da se ponaša slično čovjeku; da uči na temelju primjera te primjenjuje naučeno na nove zahtjeve.

Varijante problema (1)

Prepoznavanje izoliranih znamenaka.

Varijante problema (1)

Prepoznavanje izoliranih znamenaka.

- Svakoj slici znamenke želimo pridružiti njenu vrijednost (preciznije, jedan od brojeva između 0 i 9).
- Osnovna varijanta problema.
 - ▶ Dodatnim pretprocesiranjem prepoznavanje rukom pisanih općenitih brojeva svodi se na ovaj problem.

Varijante problema (1)

Prepoznavanje izoliranih znamenaka.

- Svakoj slici znamenke želimo pridružiti njenu vrijednost (preciznije, jedan od brojeva između 0 i 9).
- Osnovna varijanta problema.
 - ▶ Dodatnim pretprocesiranjem prepoznavanje rukom pisanih općenitih brojeva svodi se na ovaj problem.
- Možemo direktno primjeniti kod prepoznavanja brojeva upisanih na formularima na kojima postoje definirana mjesta za upisivanje pojedinih znamenaka.

Varijante problema (1)

Prepoznavanje izoliranih znamenaka.

- Svakoj slici znamenke želimo pridružiti njenu vrijednost (preciznije, jedan od brojeva između 0 i 9).
- Osnovna varijanta problema.
 - ▶ Dodatnim pretprocesiranjem prepoznavanje rukom pisanih općenitih brojeva svodi se na ovaj problem.
- Možemo direktno primjeniti kod prepoznavanja brojeva upisanih na formularima na kojima postoje definirana mjesta za upisivanje pojedinih znamenaka.

The image shows a sample of a Croatian utility bill (Nalog za plaćanje) with a grid for handwritten digits. The grid is divided into sections for the payer (Platitelj) and the payee (Primatelj). The payer's section includes fields for name, address, and a grid for digits. The payee's section includes fields for name, address, and a grid for digits. The grid for digits is a 4x10 grid, with the first column for the sign (+/-) and the remaining columns for digits 0-9. The grid is labeled 'Or. N.B. 1.' in the bottom left corner.

Varijante problema (2)

Prepoznavanje višeznamenkastih brojeva.

Varijante problema (2)

Prepoznavanje višeznamenkastih brojeva.

- Češće korištena u primjenama.
 - ▶ Prepoznavanje rukom pisanih poštanskih brojeva.

Varijante problema (2)

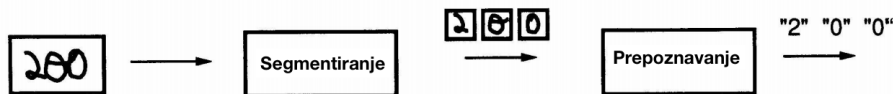
Prepoznavanje višeznamenkastih brojeva.

- Češće korištena u primjenama.
 - ▶ Prepoznavanje rukom pisanih poštanskih brojeva.
- Potrebno je izolirati pojedine znamenke, pa onda primijeniti algoritam za osnovnu varijantu problema.
 - ▶ Ovu fazu pretprocesiranja nazivamo *segmentiranje* (eng. *segmentation*).

Varijante problema (2)

Prepoznavanje višeznamenkastih brojeva.

- Češće korištena u primjenama.
 - ▶ Prepoznavanje rukom pisanih poštanskih brojeva.
- Potrebno je izolirati pojedine znamenke, pa onda primijeniti algoritam za osnovnu varijantu problema.
 - ▶ Ovu fazu pretprocesiranja nazivamo *segmentiranje* (eng. segmentation).



Varijante problema (3)

U nastavku je dan jedan način jednostavnog segmentiranja.

Varijante problema (3)

U nastavku je dan jedan način jednostavnog segmentiranja.

Primjer

Jedan od načina segmentacije svodi se na promatranje "komponenti povezanosti" na danoj slici, tj. kontinuirani skupovi crnih piksela. Ovaj pristup očigledno ne funkcionira na svim primjerima. Za generaliziranje rješenje potreban je sofisticiraniji pristup.

Varijante problema (3)

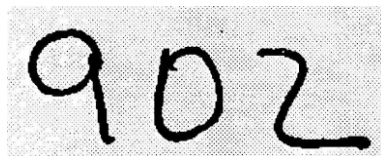
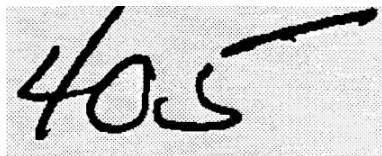
U nastavku je dan jedan način jednostavnog segmentiranja.

Primjer

Jedan od načina segmentacije svodi se na promatranje "komponenti povezanosti" na danoj slici, tj. kontinuirani skupovi crnih piksela.

Ovaj pristup očigledno ne funkcionira na svim primjerima.

Za generaliziranije rješenje potreban je sofisticiraniji pristup.



Pristupi problemu (1)

Offline prepoznavanje

Pristupi problemu (1)

Offline prepoznavanje

- Prepoznavanje brojeva nakon što je pisanje istih dovršeno.

Pristupi problemu (1)

Offline prepoznavanje

- Prepoznavanje brojeva nakon što je pisanje istih dovršeno.
- Uzorak kod offline prepoznavanja sadrži samo statičke podatke.

Pristupi problemu (1)

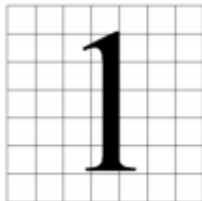
Offline prepoznavanje

- Prepoznavanje brojeva nakon što je pisanje istih dovršeno.
- Uzorak kod offline prepoznavanja sadrži samo statičke podatke.
- Ne koristi se dinamička informacija o zapisu uzorka (potez olovke, redosljed pisanja brojeva...).

Pristupi problemu (1)

Offline prepoznavanje

- Prepoznavanje brojeva nakon što je pisanje istih dovršeno.
- Uzorak kod offline prepoznavanja sadrži samo statičke podatke.
- Ne koristi se dinamička informacija o zapisu uzorka (potez olovke, redoslijed pisanja brojeva...).



Pristupi problemu (2)

Online prepoznavanje

Pristupi problemu (2)

Online prepoznavanje

- Korisnikov unos se evaluira za vrijeme zadavanja

Pristupi problemu (2)

Online prepoznavanje

- Korisnikov unos se evaluira za vrijeme zadavanja
- Nasuprot offline pristupu, koristimo informaciju o redoslijedu pisanja znamenki, potezu olovke ...
 - ▶ Koristi se kod pisanja rokom ili olovkom na tabletu, mobilnim uređajima.



Komponente algoritma (1)

Tipičan algoritam za prepoznavanje rukom pisanih znamenaka sastoji se od nekoliko faza.

- Pretprocesiranje

Komponente algoritma (1)

Tipičan algoritam za prepoznavanje rukom pisanih znamenaka sastoji se od nekoliko faza.

- Pretprocesiranje
- Ekstrakcija komponenti

Komponente algoritma (1)

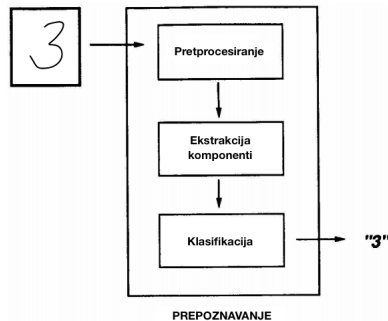
Tipičan algoritam za prepoznavanje rukom pisanih znamenaka sastoji se od nekoliko faza.

- Pretprocesiranje
- Ekstrakcija komponenti
- Klasifikacija

Komponente algoritma (1)

Tipičan algoritam za prepoznavanje rukom pisanih znamenaka sastoji se od nekoliko faza.

- Pretprocesiranje
- Ekstrakcija komponenti
- Klasifikacija



Komponente algoritma (2)

Pretprocesiranje

- Odvajanje slike od "okolne buke".

Komponente algoritma (2)

Pretprocesiranje

- Odvajanje slike od "okolne buke".
 - ▶ Skaliranje slika na standardnu veličinu.

Komponente algoritma (2)

Pretprocesiranje

- Odvajanje slike od "okolne buke".
 - ▶ Skaliranje slika na standardnu veličinu.
 - ▶ "grey-level normalization".

Komponente algoritma (2)

Pretprocesiranje

- Odvajanje slike od "okolne buke".
 - ▶ Skaliranje slika na standardnu veličinu.
 - ▶ "grey-level normalization".
 - ▶ Svođenje slike na oblik u kojem je "potez olovke" fiksne širine.

Komponente algoritma (2)

Pretprocesiranje

- Odvajanje slike od "okolne buke".
 - ▶ Skaliranje slika na standardnu veličinu.
 - ▶ "grey-level normalization".
 - ▶ Svođenje slike na oblik u kojem je "potez olovke" fiksne širine.
- Način pretprocesiranja uvelike ovisi o potrebama kasnijih koraka algoritma.

Komponente algoritma (2)

Ekstrakcija komponenti

- Izdvajanje skupa značajki koje će služiti kao kriterij za razlikovanje skupa pojedinih znamenki.

Komponente algoritma (2)

Ekstrakcija komponenti

- Izdvajanje skupa značajki koje će služiti kao kriterij za razlikovanje skupa pojedinih znamenki.
- Dobivamo vektor značajki ("feature vector").

Komponente algoritma (2)

Ekstrakcija komponenti

- Izdvajanje skupa značajki koje će služiti kao kriterij za razlikovanje skupa pojedinih znamenki.
- Dobivamo vektor značajki ("feature vector").
- Neki algoritmi preskaču ovu fazu procesiranja.

Komponente algoritma (3)

Klasifikacija

Dobivamo pripadnost znamenke jednoj od klasa. Postoje nekoliko pristupa klasifikaciji.

Komponente algoritma (3)

Klasifikacija

Dobivamo pripadnost znamenke jednoj od klasa. Postoje nekoliko pristupa klasifikaciji.

- Podudaranje s predloškom ("template matching")
 - ▶ Karakteriziramo vektor značajki (danu znamenku) s obzirom na njegovu *udaljenost* od skupova već otprije poznatih uzoraka.
 - ▶ Npr. algoritam k-klasteriranja.

Komponente algoritma (3)

Klasifikacija

Dobivamo pripadnost znamenke jednoj od klasa. Postoje nekoliko pristupa klasifikaciji.

- Podudaranje s predloškom ("template matching")
 - ▶ Karakteriziramo vektor značajki (danu znamenku) s obzirom na njegovu *udaljenost* od skupova već otprije poznatih uzoraka.
 - ▶ Npr. algoritam k-klasteriranja.
- Statistički klasifikatori
 - ▶ Koriste skup primjera (u ovom slučaju skup rukom pisanih znamenaka) kako bi se odredila distribucija u različite klase.
 - ▶ Nakon što je uzorak "istreniran", za novi primjer se određuje vjerojatnost da pripada nekoj klasi.
 - ▶ Npr. algoritmi neuronskih mreža.

Komponente algoritma (3)

Klasifikacija

- Sintaktičke metode
 - ▶ Opisuju svaku klasu (svaku znamenku 0-9) pomoću specifičnih kombinacija značajki.
 - ▶ Skup "legalnih" kombinacija za neku klasu definiran je zasebnim pravilima.

Testiranje (1)

Testirana su dva algoritma za prepoznavanje rukom pisanih znamenaka.

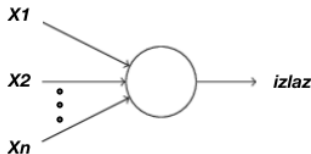
- Algoritmi neuronskih mreža.
- Algoritam korištenjem aproksimacije matricom nižeg ranga.

Testiranje (2)

Neuronske mreže

● Perceptron

- ▶ Tip umjetnog neurona.
- ▶ Kao unos prima bitove x_1, x_2, \dots, x_n te kao izlaz ima bit *izlaz*.
- ▶
$$\text{izlaz} = \begin{cases} 0, & \text{ako } \sum_{i=1}^n w_i x_i \leq \text{prag} \\ 1, & \text{ako } \sum_{i=1}^n w_i x_i > \text{prag} \end{cases}$$
, gdje je *prag* neka zadana vrijednost.
- ▶ Problem - mala promjena unosa može previše utjecati na izlaz.



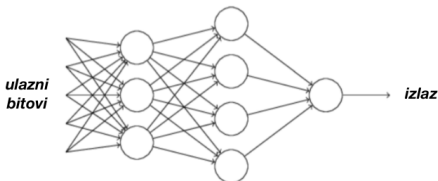
- Sigmoidni neuron

- ▶ Struktura slična kao kod perceptrona.
- ▶ Izlaz je realni broj između 0 i 1, rezultat funkcije $\frac{1}{1+e^{-w \cdot x - b}}$, gdje b (bias) jednak *-prag*.

- Sigmoidni neuron

- ▶ Struktura slična kao kod perceptrona.
- ▶ Izlaz je realni broj između 0 i 1, rezultat funkcije $\frac{1}{1+e^{-w \cdot x - b}}$, gdje b (bias) jednak -*prag*.

- Neuronska mreža



Testiranje (2)

Neuronske mreže

- Mreža u algoritmu je troslojna feedforward mreža.

Testiranje (2)

Neuronske mreže

- Mreža u algoritmu je troslojna feedforward mreža.
- Za učenje neuronske mreže koristimo metodu najbržeg silaska.
 - ▶ minimiziramo funkciju $C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2$, gdje $y := y(x)$ definiramo kao vektor dimenzije 10 koji označava stvarnu vrijednost primjera.
 - ▶ Iz skupa primjera za treniranje uzimamo slučajnim odabirom manji skup, treniramo mrežu na tim primjerima. Ponavljamo postupak dok ne iscrpimo sve primjere - završimo *epohu*.
 - ▶ Treniranje staje nakon zadanog broja epoha.

Testiranje (2)

Neuronske mreže

- Mreža u algoritmu je troslojna feedforward mreža.
- Za učenje neuronske mreže koristimo metodu najbržeg silaska.
 - ▶ minimiziramo funkciju $C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2$, gdje $y := y(x)$ definiramo kao vektor dimenzije 10 koji označava stvarnu vrijednost primjera.
 - ▶ Iz skupa primjera za treniranje uzimamo slučajnim odabirom manji skup, treniramo mrežu na tim primjerima. Ponavljamo postupak dok ne iscrpimo sve primjere - završimo *epohu*.
 - ▶ Treniranje staje nakon zadanog broja epoha.
- Znamenka pripada skupu (znamenki) za koju je izlaz neuronske mreže najveći.

Testiranje (2)

Neuronske mreže

Rezultati

n-srednji sloj	epohe	η	"mini-batch"	točnost
30	25	3.0	10	95.16%
100	30	3.0	10	96.49%
50	50	3.0	10	95.64%
50	100	3.0	15	95.7%

- n - broj neurona u srednjem sloju.
- epohe - broj epoha.
- η - tzv. "learning rate" parametar metode najbržeg silaska.
- "mini-batch" - veličina podskupova koje obradimo odjednom kod metode najbržeg silaska.

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

U članku [2] predstavljen je algoritam za pretraživanje tekstualnih dokumenata.

- Bazira se na SVD dekompoziciji, te aproksimaciji matrice matricom manjeg ranga.

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

U članku [2] predstavljen je algoritam za pretraživanje tekstualnih dokumenata.

- Bazira se na SVD dekompoziciji, te aproksimaciji matrice matricom manjeg ranga.
- Može se prilagoditi za problem prepoznavanja rukom pisanih znamenki.

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

U članku [2] predstavljen je algoritam za pretraživanje tekstualnih dokumenata.

- Bazira se na SVD dekompoziciji, te aproksimaciji matrice matricom manjeg ranga.
- Može se prilagoditi za problem prepoznavanja rukom pisanih znamenki.
- Aproksimacija matricom nižeg ranga koristi se kako bi se iz ulaznih primjera eliminirala "buka".

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

Ideja algoritma

- $A = [S_1 \ S_2 \ \dots \ S_n]$, j -ti stupac, S_j , reprezentacija jedne znamenke (po jedna komponenta za svaki piksel).

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

Ideja algoritma

- $A=[S_1 \ S_2 \ \dots \ S_n]$, j -ti stupac, S_j , reprezentacija jedne znamenke (po jedna komponenta za svaki piksel).
- $D=[d_1 \ d_2 \ \dots \ d_n]$, d_j ima vrijednost znamenke čija je slika pohranjena u j -tom stupcu matrice A .

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

Ideja algoritma

- $A = [S_1 \ S_2 \ \dots \ S_n]$, j -ti stupac, S_j , reprezentacija jedne znamenke (po jedna komponenta za svaki piksel).
- $D = [d_1 \ d_2 \ \dots \ d_n]$, d_j ima vrijednost znamenke čija je slika pohranjena u j -tom stupcu matrice A .
- Definiramo X_i kao matricu sastavljenu od stupaca matrice A koji reprezentiraju znamenku i .

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

Ideja algoritma

- $A = [S_1 \ S_2 \ \dots \ S_n]$, j -ti stupac, S_j , reprezentacija jedne znamenke (po jedna komponenta za svaki piksel).
- $D = [d_1 \ d_2 \ \dots \ d_n]$, d_j ima vrijednost znamenke čija je slika pohranjena u j -tom stupcu matrice A .
- Definiramo X_i kao matricu sastavljenu od stupaca matrice A koji reprezentiraju znamenku i .
- Koristeći SVD i aproksimaciju matricom nižeg ranga dobivamo potprostore za pojedine znamenke.

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

Ideja algoritma

- $A = [S_1 \ S_2 \ \dots \ S_n]$, j -ti stupac, S_j , reprezentacija jedne znamenke (po jedna komponenta za svaki piksel).
- $D = [d_1 \ d_2 \ \dots \ d_n]$, d_j ima vrijednost znamenke čija je slika pohranjena u j -tom stupcu matrice A .
- Definiramo X_i kao matricu sastavljenu od stupaca matrice A koji reprezentiraju znamenku i .
- Koristeći SVD i aproksimaciju matricom nižeg ranga dobivamo potprostore za pojedine znamenke.
- Potprostor kojem pripada neki primjer određujemo tako da računamo udaljenost primjera (reprezentiran jednako kao stupac matrice A) od svakog od konstruiranih potprostora.

Testiranje (3)

Algoritam aproksimacije matricom nižeg ranga

Dodajmo još da je rang aproksimacije dimenzija potprostora koje koristimo.

Rezultati:

rang aproksimacije	točnost
5	90.28%
6	90.53%
10	93.17%
14	93.92%
15	94.16%
18	94.17%



Michael A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.



M. W. Berry, Z. Drmač, E.R. Jessup, *Matrices, Vector Spaces, and Information Retrieval*, Society for Industrial and Applied Mathematics, 1999.



O. Matan, J. Bromley, C.J.C. Burges, J.S. Denker, L.D. Jackel, Y.L. Cun, E.P.D. Pednault, W.D. Satterfield, C.E. Stenard, T.J. Thompson, *Reading Handwritten Digits: A Zip Code Recognition System*, 1991.



https://en.wikipedia.org/wiki/Handwriting_recognition, (pristupano 9.1.2016.)