

# Klasifikacija znamenki pomoću konvolucijske neuronske mreže (CNN)

---

DANIEL JELUŠIĆ

# Uvod

---

- Problem: klasifikacija rukom pisanih znamenki
  - Za danu sliku želimo odrediti koju ona znamenku predstavlja
  - Želimo da takva klasifikacija bude što brža i što točnija
- Jedno rješenje: Konvolucijska neuronska mreža
  - Učenje s nadzorom
  - 3 sloja (Ulazni, Konvolucijski i Softmax sloj)
  - Slika se predprocesira (Konvolucija i Pooling) prije klasifikacije

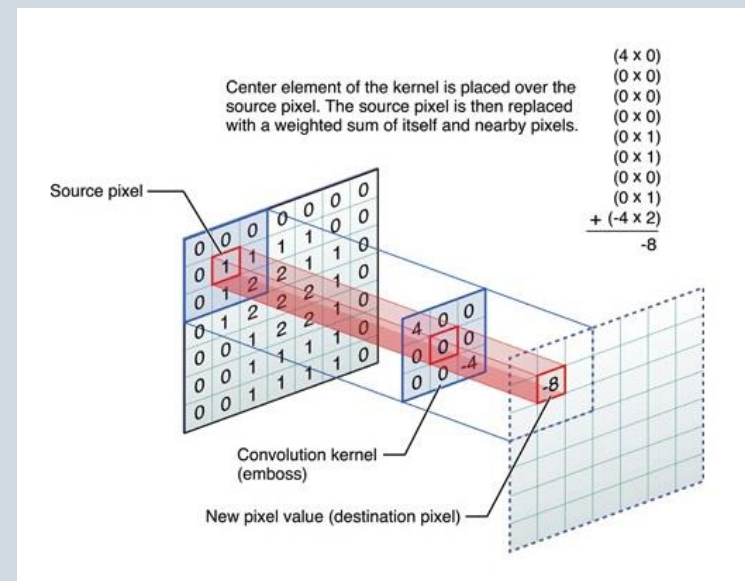
# Konvolucija

## ➤ Osnovni korak u konvolucijskom sloju

- Ulazna slika se konvoluirala sa k filtera (kernela) dobivenih učenjem
- Dobivamo k novih slika

## ➤ Zašto konvolucija?

- Slike koje predstavljaju istu znamenku imaju neka zajednička obilježja
- Konvolucija izolira te sličnosti i time olakšava softmax sloju klasifikaciju



# Pooling

---

- Redukcija dimenzije konvoluiranih slika
  - $n \times n$  sliku podijelimo na disjunktne  $p \times p$  kvadrate
  - Definiramo novu sliku tako da uzmemo srednje vrijednosti svih  $p \times p$  kvadrata
  - Dobivena slika je dimenzije  $(n/p) * (n/p)$
- Dva su razloga za pooling
  - Ubrzava račun u softmax sloju
  - Sprječava „overfitting” zbog prevelike količine podataka

# Softmax sloj

---

- Na ulazu prima konvoluirane i poolane slike
  - Njih ima točno  $k$  i svaka je dimenzije  $(n/p)^2$
  - Zato stvorimo vektor  $x$  dimenzije  $k * (n/p)^2$
- Parametri softmax regresije su  $\theta = (\theta_1, \dots, \theta_{10})$ 
  - Svaki od njih je dimenzije  $k * (n/p)^2$
- Konačno, izlaz softmax sloja je  $h_{\theta_i}(x) = \frac{1}{1 + \exp(-\theta_i^T x)}$ , za  $i = 1, \dots, 10$ .
  - Ako ovo normaliziramo dobijemo zapravo predviđanja našeg modela

# Učenje

---

- Faza učenja svodi se na optimizaciju filtera i parametara softmax regresije
  - Optimizacija se provodi pomoću trening skupa
- Koristi se Shohastički gradijentni spust (SGD)
  - SGD se realizira kroz nekoliko „epoha”
  - Nakon svake epohe smanjujemo veličinu koraka, odnosno  $\alpha$
- Za dobivanje gradijenata koristimo Backpropagation algoritam

# Primjer klasifikacije

---

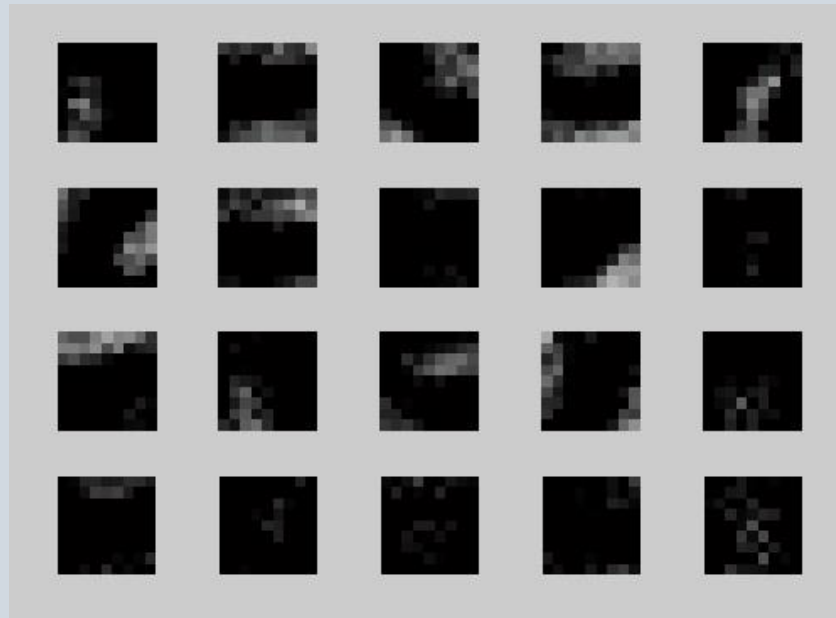
- Jednom kada smo izračunali optimalne parametre, klasifikacija slike se svodi na to da jednom prođemo kroz mrežu i izračunamo pripadne vjerojatnosti
- Znamenku pridružimo klasi koja ima najveću vjerojatnost
- Slijedi vizualni primjer klasifikacije
- Uzmimo prvu znamenku iz testnog skupa:
- Polazna slika je dimenzije 28x28



# Primjer klasifikacije (2)

---

- Odlučili smo se da je filtera za konvoluciju ukupno 20
- Nakon optimizacije, dobiveni su sljedeći filteri (dimenzije 9x9):

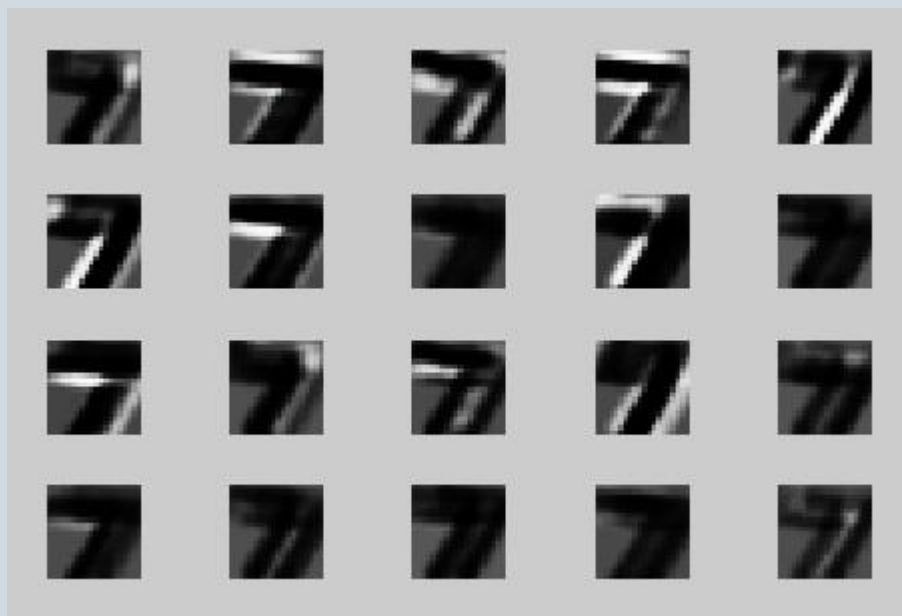




# Primjer klasifikacije (3)

---

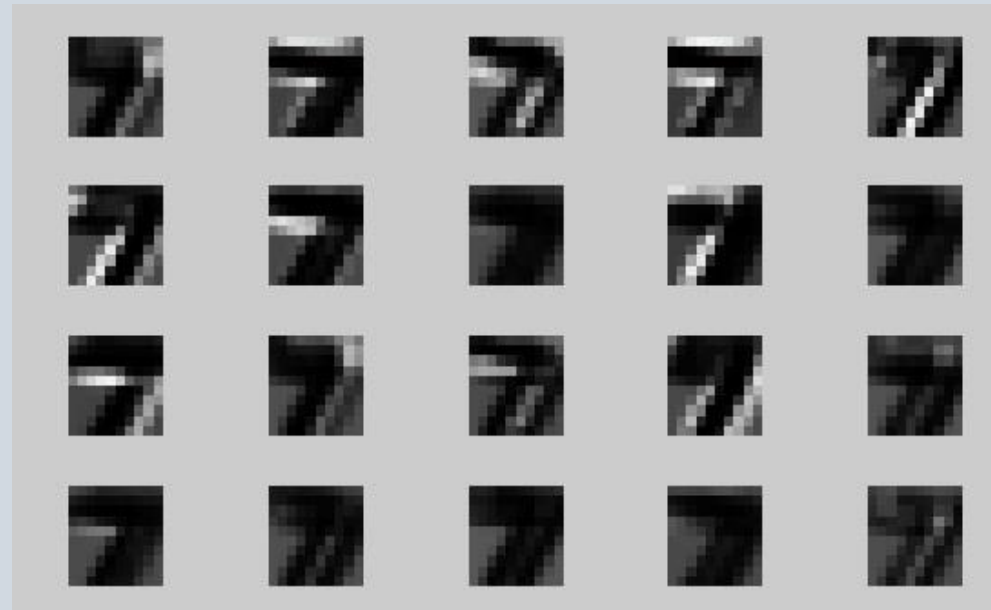
- Konvolucijom (i aktivacijom) početne slike sa filterima dobivamo sljedećih 20 20x20 slika:



# Primjer klasifikacije (4)

---

- Poolingom smanjimo dimenziju slika na 10x10:



# Primjer klasifikacije (5)

---

- Konačno, ovako poolane slike su ulaz softmax sloju
- Nakon prolaza kroz softmax sloj, dobivene vjerojatnosti su:
- Dakle, znamenku klasificiramo kao 7, što odgovara stvarnoj klasi

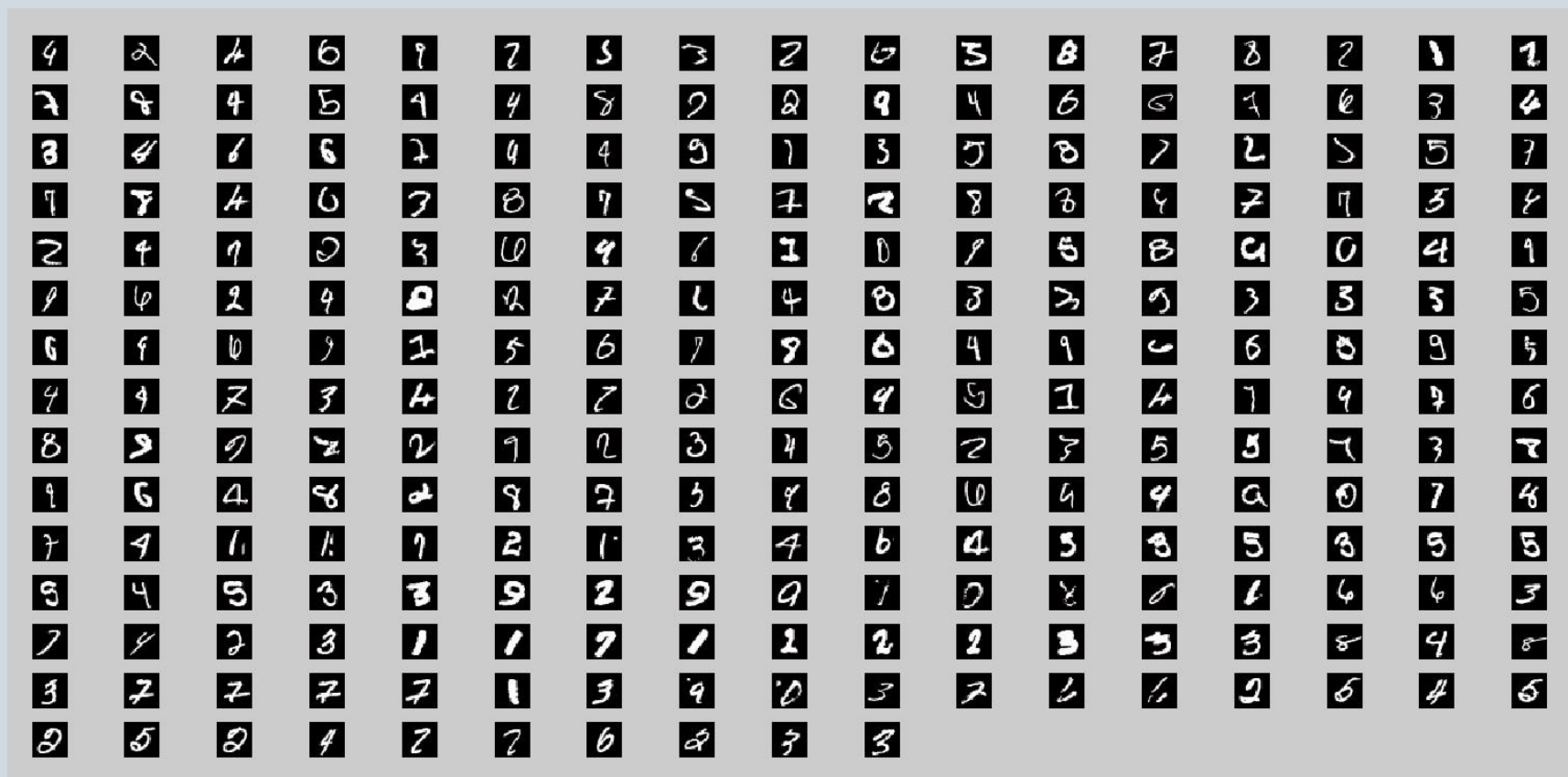
```
0.0000  
0.0000  
0.0002  
0.0000  
0.0000  
0.0000  
0.0000  
0.9997  
0.0000  
0.0000  
0.0000
```

# Rezultati

---

- Klasifikacija cijelog testnog skupa trajala je otprilike 45s
- Najbolja postignuta točnost je 97.9%
- Za kraj, na idućem slajdu prikazane su sve znamenke koje je algoritam krivo klasificirao
  - Uočimo da je dobar dio tih slika neprepoznatljiv čovjeku, pa je teško kriviti algoritam za te greške
  - Međutim, neke slike su vrlo jasne i nije jasno zašto ih algoritam nije uspio pravilno klasificirati

# Krivo klasificirane slike



# Literatura

---

- ❑ UFLDL Tutorial [http://ufldl.stanford.edu/wiki/index.php/UFLDL\\_Tutorial](http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial)
- ❑ UFLDL Exercise: Convolutional Neural Network  
[http://ufldl.stanford.edu/tutorial/index.php/Exercise: Convolutional Neural Network](http://ufldl.stanford.edu/tutorial/index.php/Exercise:_Convolutional_Neural_Network)