SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Diplomski studij

RASPOZNAVANJE ZNAMENAKA KORIŠTENJEM KONVOLUCIJSKIH NEURONSKIH MREŽA

Meko računarstvo

Laboratorijska vježba 6

Andrej Bošnjak

DRB

Osijek, 2023.

SADRŽAJ

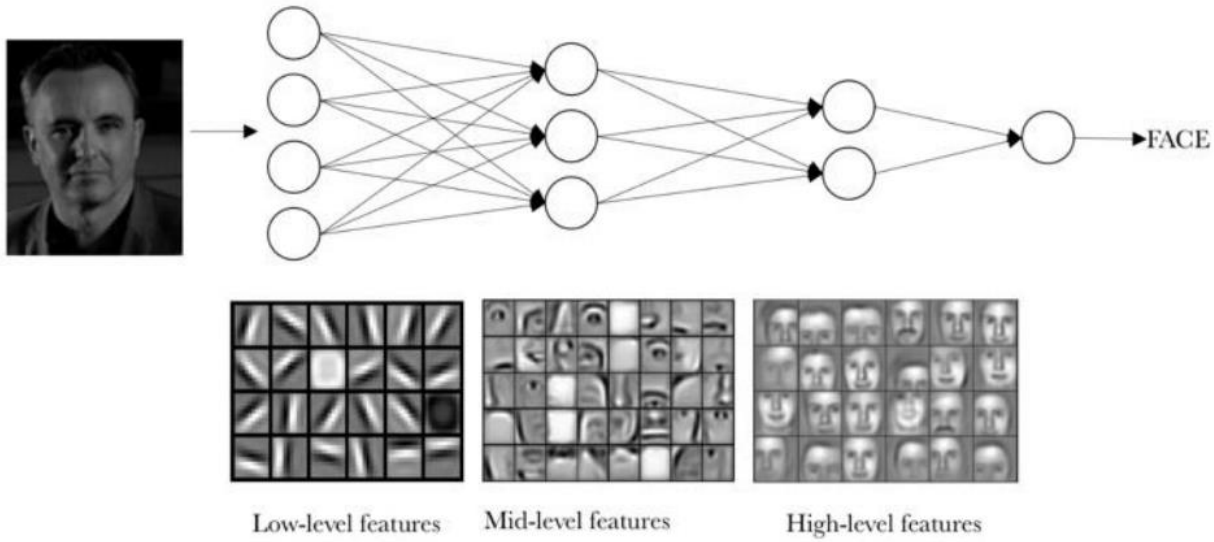
1. UVOD

Na šestoj laboratorijskoj vježbi se prepoznaju znakovi pomoću konvolucijskih neuronskih mreža. Nakon provedenog postupka učenja kroz 5 epoha se određuje preciznost cijele mreže, te se ona uzima kao indikator i evaluacija. Cilj je uočiti kako određeni parametri mreže utječu na dobivene rezultate.

2. OPIS PROBLEMA I RJEŠENJA

2.1. Konvolucijske neuronske mreže (CNN)

Duboko učenje je grana strojnog učenja, dok je konvolucijska neuronska mreža konkretna implementacija dubokih neuronskih mreža koja se najčešće primjenjuje za analizu i raspoznavanje slika. Standardne neuronske mreže nisu osobito prigodne u svrhu analize slika zbog zahtjeva da svaki neuron mora biti povezan sa svim elementima iz prethodnog sloja. Stoga ako npr. imamo sliku veličine 25 x 25 što znaci ukupno 225 piksela u konačnici će rezultirati s 225 težina koje je potrebno podesiti i optimizirati samo za jedan jedini neuron. Stoga, ako je je slika skromnih 100 x 100 piksela to je već 10000 težina po neuronu i ako imamo 100 neurona u jednom sloju ispada 1000000 težina koje se moraju podesiti. Konvolucijske neuronske mreže imaju takvu arhitekturu koja omogućava da se raspoznaju uzorci različitih dimenzija uz puno manje parametara koje je potrebno podesiti i optimizirati. Smisao konvolucijske neuronske mreže jest da se kroz slojeve neuronske mreže uče raspoznavati inkrementalno kompleksniji uzorci. Ako se uči npr. raspoznavanje ljudskih lica, na prvoj razini se raspoznaju linije, rubovi i različiti točkasti uzorci, na drugoj su to npr. dijelovi lica kao što je to nos, uho, usta, oko i sl., dok na idućoj razini su to veći dijelovi lica itd.



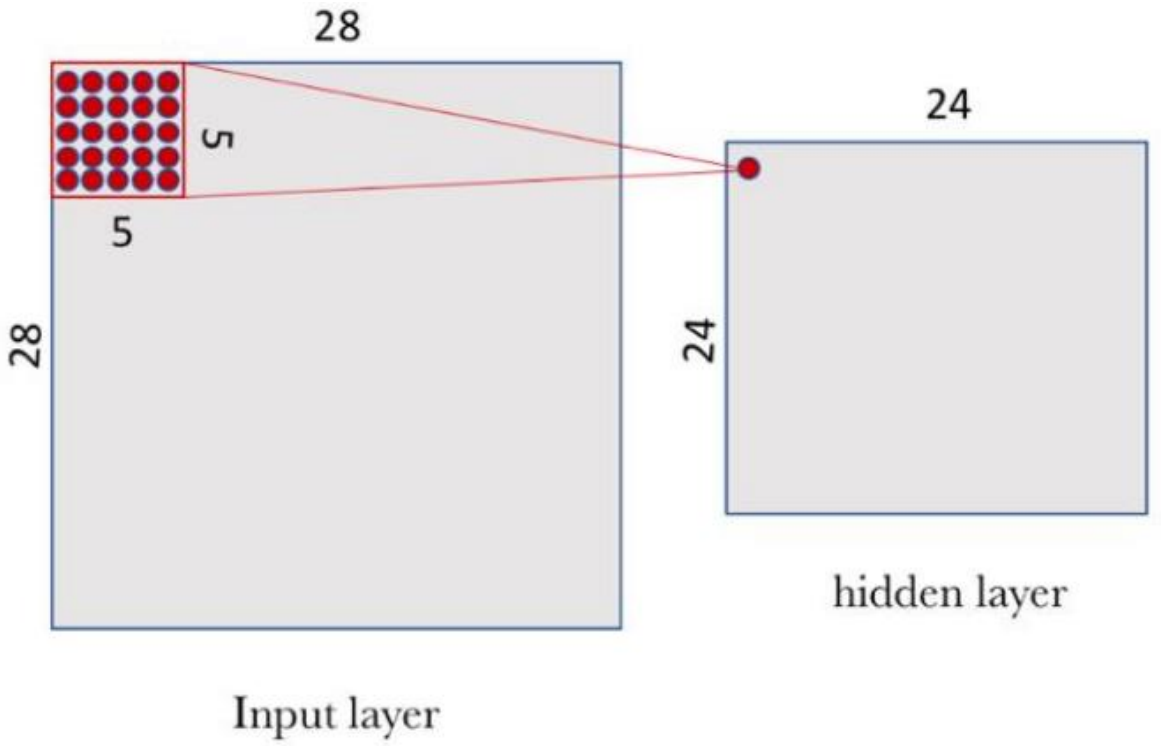
Slika 2.1 Kompleksnost raspoznavanja uzoraka se inkrementalno povećava kroz slojeve

Drugim riječima, svaki sloj konvolucijske neuronske mreže uči različitu razinu apstrakcije ulaznih podataka. Iako konvolucijska neuronska mreža može imati mnogo različitih dijelova, tipični dijelovi uključuje sljedeće slojeve:

- Konvolucijski sloj (engl. Convolution layer)
- Sloj sažimanja (engl. Pooling Layer),
- Potpuno povezani sloj (engl. Fully-Connected (dense) Layer)

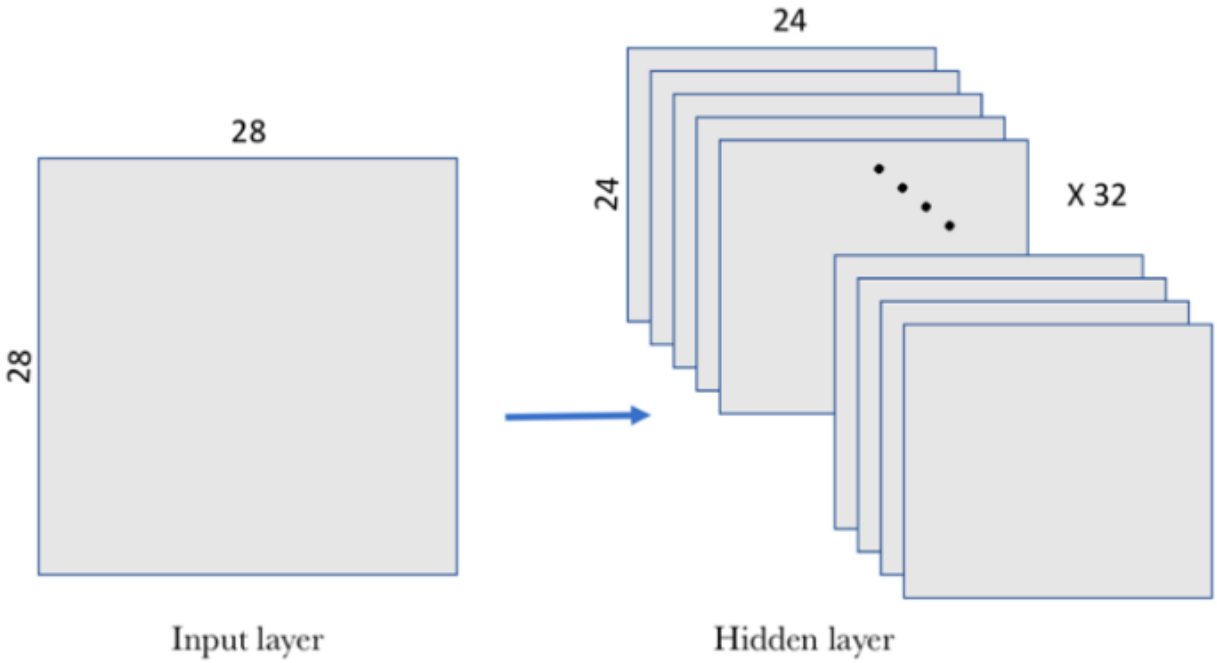
Unutarnja struktura tipične konvolucijske neuronske mreže sastoji se od nekoliko naizmjenično poslaganih višedimenzionalnih konvolucijskih slojeva i slojeva sažimanja. Na kraju se nalazi potpuno povezan sloj, koji je jednodimenzionalan, kao i izlazni sloj. Fundamentalna razlika između konvolucijskog sloja i potpuno povezanog sloja (istovrstan standardnoj neuronskoj mreži), jest to što potpuno povezan sloj uči uzorke globalno dok konvolucijski sloj uči lokalne uzorke unutar malih dvodimenzionalnih prozora nazvanih jezgra (engl. kernel). Tako naučne lokalne uzorke konvolucijski sloj nauči raspoznavati na bilo kojem mjestu na slici, dok bi na primjer standardne neuronske mreže naučile raspoznavati samo na točno određenom mjestu. Stoga ako bi neka značajka promijenila svoj položaj, standardna neuronska mreža bi morala ponovo biti naučena. Još jedna bitna značajka konvolucijskih slojeva je ta da može naučiti odnosno zadržati prostorne odnose između uzoraka. Ako se na primjer u prvom konvolucijskom sloju uče osnovni uzorci kao što su rubovi, linije, prijelazi i sl. tada se u drugom sloju mogu naučiti kompleksne kompozicije tih osnovnih uzoraka iz prethodnog sloja. Time se iz sloja u sloj povećava kompleksnost apstraktnih vizualnih koncepata. Konvolucijski slojevi i slojevi sažimanja se baziraju na 3D tenzorima nazvanima mape značajki (engl. feature map), koje imaju širinu, visinu i dubinu. Ulazni slojevi mogu isto imati te tri dimenzije, ako se radi o crno-bijeloj slici tada pored širine i visine ima samo dubinu od 1, dok slike u boji imaju dubinu 3 (crvena, zelena i plava komponenta slike). Konvolucijski sloj radi na principu konvolucije gdje maleni prozor „klizi“ po slici, umnaža elemente (piksele) slike i potom ih sumira, te uz dodatak bias-a „pridružuje“ neuronu na odgovarajućem mjestu konvolucijskog sloja. Vizualno, obradu krećemo s prozorom u gornjem lijevom kutu slike koji daje potrebne informacije prvom (opet, gornji lijevi kut) neuronu konvolucijskog sloja. Nakon toga pomičemo prozor za jedno mjesto u desno i povezujemo

vrijednosti s drugim neuronom u konvolucijskom sloju i tako obilazimo cijelu sliku, s lijeva na desno, od gore prema dolje.



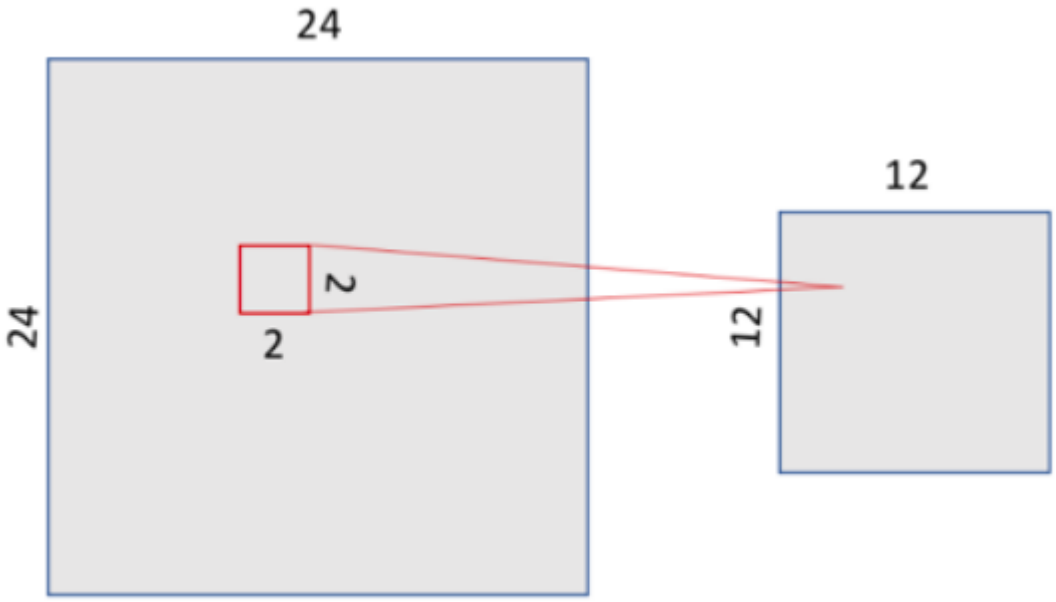
Slika 2.2 Konvolucija se obavlja prozorima predefinirane veličine

Ako imamo ulaznu sliku 28 x 28 pikesela (npr. MNIST baza slika znamenki) i prozor od 5 x 5, tada veličina konvolucijskog sloja može biti samo 24 x 24, jer nam ta dimenzija omogućava da obiđemo sve elemente ulazne slike s našim prozorom bez da narušavamo granice slike, odnosno možemo pomaknuti prozor samo za 23 mjesta u desno (odnosno prema dolje) bez da pređemo granicu slike. Naravno prozor se po želji može pomicati za više mjesta odjednom što se definira kao korak (engl. stride) uzorkovanja, a također na rubove slike se mogu dodavati odgovarajući broj nula čime se omogućava da unutarnji (konvolucijski) slojevi neuronske mreže imaju istu širinu i visinu kao i ulazna slika (engl. padding). Dakle prema korištenom primjeru, svaki neuron iz skrivenog sloja je povezan s odgovarajućom regijom ulazne slike preko 5 x 5 prozora. To povezivanje je definirano s 5 x 5 matricom težina koja se naziva filter i pripadajućim bias-om.



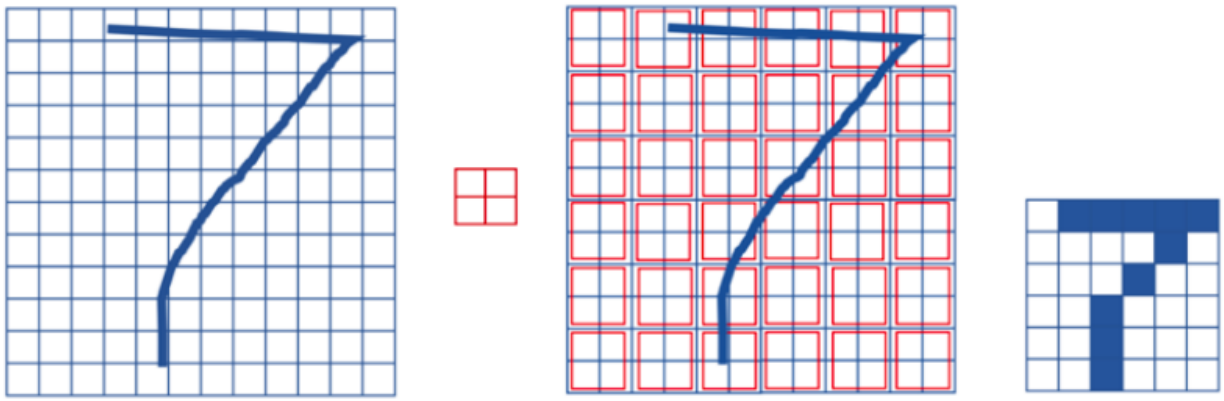
Slika 2.3 Svaki konvolucijski sloj tipično ima više filtera, svaki povezan sa svojim podslojem

U primjeru prikazanom na Slici 2.3, prvi konvolucijski sloj prima tenzor dimenzija (28,28,1) i rezultira tenzorom (24,24,32) za koji je potrebno proračunati i optimizirati 32 x 5 x 5 + 32 x 1 = 832 vrijednosti. Često se nakon konvolucijskih slojeva postavljaju slojevi sažimanja (engl. pooling). Njihova je svrha da pojednostave informaciju koja dolazi iz konvolucijskih slojeva i time kondenzira njihov sadržaj. Drugim riječima funkcija sažimanja je da mapira skup prostorno bliskih značajki na ulazu u jednu značajku na izlazu, u tu svrhu se koristi statistički pokazatelji kao što su maksimum ili srednja vrijednost. Ako npr. u našem slučaju koristimo funkciju sažimanja s 2 x 2 prozorom, dobit ćemo na izlazu tenzor s duplo manjom širinom i visinom.



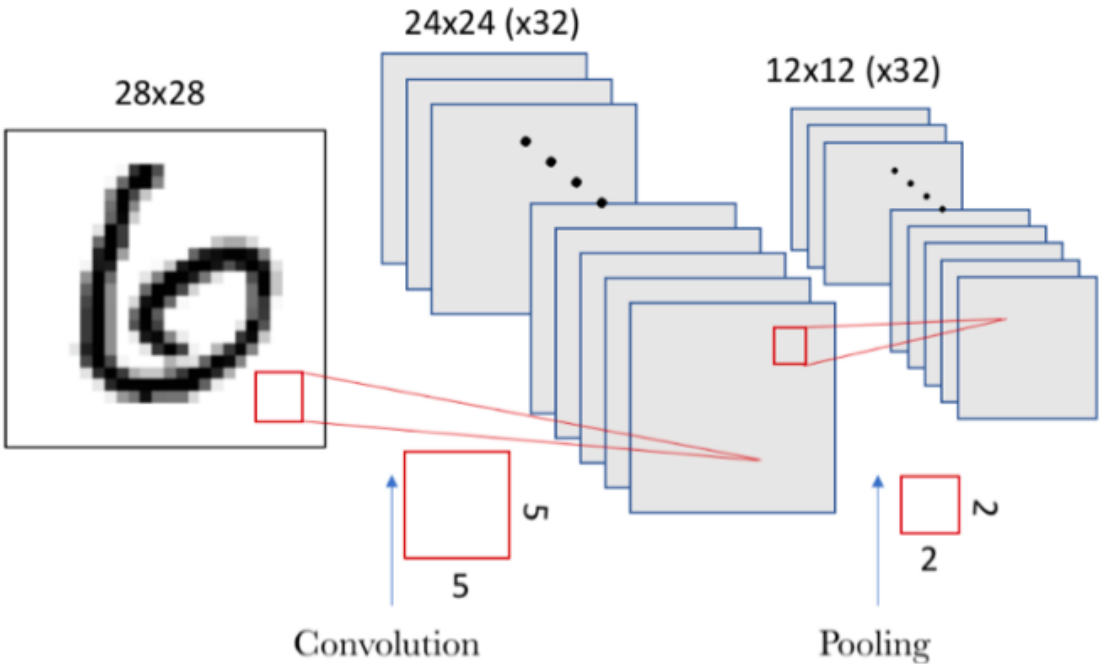
Slika 2.4 Sloj sažimanja rezultira s manjom količinom podataka koje je potrebno obraditi u narednim slojevima

Ovakvim pristupom sažimanja se i dalje održavaju prostorni odnosi npr.



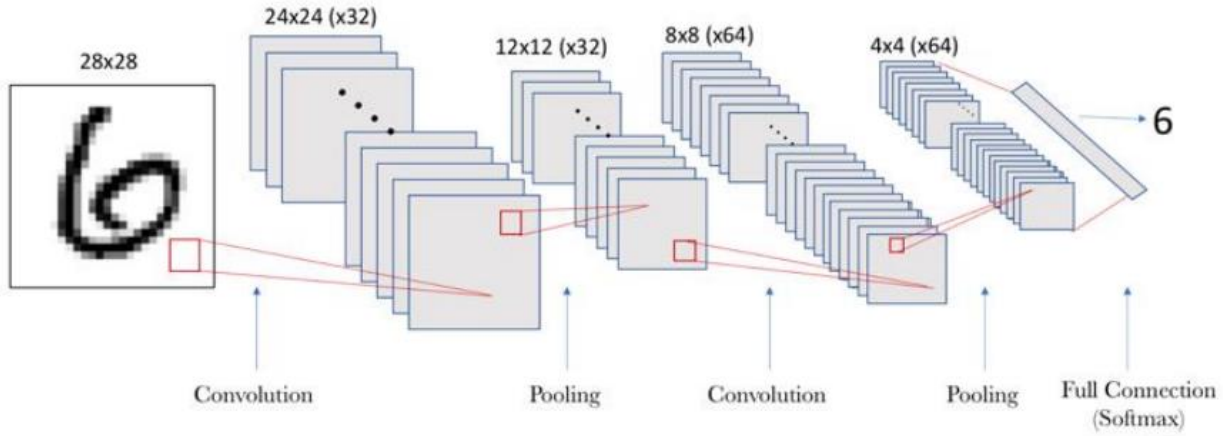
Slika 2.5 Prostorni odnosi se održavaju bez obzira na smanjivanje rezolucije

U primjeru prikazanom na Slici 2.5, za 12 x 12 ulaz i prozor od 2 x 2, na kraju se dobije 6 x 6 izlaz na kojemu se i dalje može razaznati da se radi o broju 7 iako smo time znatno smanjili količinu informacija koju je potrebno obraditi. Kao što je prethodno navedeno, konvolucijski slojevi imaju više filtera odnosno više podslojeva, te je za svaki taj podsloj individualno potrebno provesti sažimanje.



Slika 2.6 Sažimanje se provodi za svaki podsloj konvolucijskog sloja individualno

Na kraju konvolucijske neuronske mreže se nalazi potpuno povezani sloj koji predstavlja jedan niz neurona gdje je svaki neuron povezan s svim neuronima prethodnog sloja na sličan način kao što se to radi i kod običnih neuronskih mreža. Nakon takvog sloja se nalazi još jedan sloj koji ima onoliko neurona koliko ima i klasa ulaznih podataka (npr. ako raspoznajemo znamenke onda ima 10 neurona, za znamenke od 0..9) nakon čega se najčešće nekom softmax metodom određuje izlaz iz neuronske mreže.



Slika 2.7 Kompletna implementacija jedne konvolucijske neuronske mreže

Na slici iznad se može vidjeti tipična konvolucijska neuronska mreža s višestrukim konvolucijskim slojevima i slojevima sažimanja.

2.2. Parametri

Parametri neuronske mreže koji se mijenjaju:

- Aktivacijska funkcija: 'relu', 'tanh', 'sigmoid'
- Veličina filtera (kernel-a) konvolucijskih slojeva: 3 x 3, 5 x 5, 7 x 7

Korištena su dva modela neuronskih mreža. Prvi model sadrži samo dva konvolucijska sloja, prvi sa 64 filtera i drugi sa 32 filtera. Drugi model isto ima dva konvolucijska sloja s istim brojem filtera kao i prethodni model, ali za razliku od prethodnog modela ima i dva sloja za sažimanje, jedan između konvolucijskih slojeva i jedan na kraju, prije potpuno povezanog modela. Slojevi za sažimanje koriste max funkciju i veličinu prozora 2 x 2.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
```

Slika 2.8 Arhitektura prvog modela mreže

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 24, 24, 64)        1664

 max_pooling2d (MaxPooling2D  (None, 12, 12, 64)       0
 )

 conv2d_3 (Conv2D)           (None, 8, 8, 32)          51232

 max_pooling2d_1 (MaxPooling  (None, 4, 4, 32)         0
 2D)

 flatten_1 (Flatten)         (None, 512)               0

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 58,026
Trainable params: 58,026
```

Slika 2.9 Arhitektura drugog modela mreže

Također će se rezultati konvolucijskih neuronskih mreža usporediti i sa standardnom neuronskom mrežnom. Standardna neuronska mreža ima sljedeće parametre:

- 100 neurona
- Aktivacijska funkcija: „logistic“
- Optimizacijski postupak „adam“
- 100 epoha, sa rano zaustavljanje omogućeno

3. KOD

Kod s kojim se izvršava treniranje svih kombinacija parametara mreža dan je u nastavku:

```
from keras import layers
from keras import models
from keras.datasets import mnist
from keras.utils import to_categorical


(X_train, y_train), (X_test, y_test) = mnist.load_data()


X_train = X_train.reshape(60000,28,28,1)
```

```
X_train = X_train.astype('float32') / 255
X_test = X_test.reshape(10000,28,28,1)
X_test = X_test.astype('float32') / 255
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)

filters=[(3,3), (5,5), (7,7)]
activations=["relu", "tanh", "sigmoid"]

for filter in filters:
  for activationF in activations:
    model1 = models.Sequential()

    model1.add(layers.Conv2D(64, filter, activation=activationF, strides=(1,1), input_shape=(2
8, 28, 1)))
    model1.add(layers.Conv2D(32, filter, activation=activationF, strides=(1,1)))
    model1.add(layers.Flatten())
    model1.add(layers.Dense(10, activation="softmax"))
    model1.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])

    history1 = model1.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=5)
    print("==========================================================
====")
    print("Filter: " + str(filter) + ", activation function: " + str(activationF) + ", accuracy:" + str(
history1.history['val_accuracy'][-1]))
    print("==========================================================
====")

    model2 = models.Sequential()
    model2.add(layers.Conv2D(64, filter, activation=activationF, strides=(1,1), input_shape=(2
8,28,1)))
    model2.add(layers.MaxPool2D((2,2)))
```

```
    model2.add(layers.Conv2D(32, filter, activation=activationF, strides=(1,1), input_shape=(2
8,28,1)))
    model2.add(layers.MaxPool2D((2,2)))
    model2.add(layers.Flatten())
    model2.add(layers.Dense(10, activation="softmax"))
    model2.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])

    history2 = model2.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=5)
    print("==========================================================
====")
    print("Filter: " + str(filter) + ", activation function: " + str(activationF) + ", accuracy:" + str(
history2.history['val_accuracy'][-1]))
    print("==========================================================
====")
```

```
from keras import layers
from keras import models
from keras.datasets import mnist
from keras.utils import to_categorical
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix

from sklearn.neural_network import MLPClassifier, MLPRegressor
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels

(X_train, y_train), (X_test, y_test) = mnist.load_data()

X_train = X_train.reshape(60000,28*28)
X_train = X_train.astype('float32') / 255
X_test = X_test.reshape(10000,28*28)
X_test = X_test.astype('float32') / 255
y_train = to_categorical(y_train)
```

```
y_test = to_categorical(y_test)


#Create neural network
mlp = MLPClassifier(hidden_layer_sizes=(100), activation="logistic", alpha=0.00001, solver=
"adam", max_iter=100, verbose=True, early_stopping=True)


#Learn neural network
mlp.fit(X_train,y_train)
```

4. REZULTATI

Dobiveni rezultati su preciznosti nad validacijskim skupom, prikazani u sljedećoj tablici:

	Model 1			Model 2		
	Relu	Tanh	Sigmoid	Relu	Tanh	Sigmoid
3x3	0.97930	0.97049	0.92619	0.97570	0.97109	0.90100
5x5	0.98259	0.980499	0.93320	0.98079	0.97649	0.91589
7x7	0.98479	0.981999	0.94739	0.97820	0.97000	0.90119
	Standardna neuronska mreža					
	0.954167					

5. ZAKLJUČAK

Nakon izvršenih kombinacija parametara za navedene neuronske mreže, vidljivo je da promjenom bilo kojih parametara utječemo na rezultat.

Iz slika 2.8 i 2.9 je vidljivo da drugi model mreže ima četiri puta manje parametara za učiti zbog slojeva za sažimanje. To također smanjuje vrijeme učenja mreže.

Aktivacijska funckija „sigmoid“ daje najlošiju validacijsku preciznost u oba modela, dok „relu“ i „tanh“ daju približno jednake rezultate.

Općenito povećanjem veličine kernel-a se dobivaju bolji rezultati.

Standardna neuronska mreža ima najveće vrijeme učenja zbog velikog broja epoha, no zbog parametra ranog zaustavljanja se to vrijeme malo skratilo. U konačnici je imala prosječan rezultat ako se usporede najbolji i najlošiji rezultati konvolucijskih neuronskih mreža.