Machine Learning

Lecture 20: Convolution Neural Network

Chandresh Kumar Maurya
Assistant Professor in the CSE department

Indian Institute of Technology Indore, India

Slides Credit

Some of the slides are adapted from

- Dr. Piyush Rai machine learning course IITK CS 771A.
- Sudeep Raja, Columbia University
- Quoc Vi. Lee, A Tutorial on Deep Learning, Google Inc.

Table of Contents

Limitations of Feed-Forward Networks

- Require a huge number of parameters (note that the consecutive layers are fully connected)
- Not ideal for data that exhibit locality structure, e.g., (e.g., images, sentences)
  - $\rightarrow$ Kind of works but would be better to exploit locality in the data more explicitly
- Doesn't have a "memory", so not ideal when modeling sequence of observations

Table of Contents

The whole CNN

cat dog ……

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flattened

Fully Connected Feedforward network

CNN

- A feedforward neural network with a special structure
- Not all pairs of nodes are connected
- Weights are also "tied" (many connections have the same weights; color-coded above)
- The set of distinct weights defines a "filter" or "local" feature detector

CNN

- Applies 2 operations, convolution and pooling (subsampling), repeatedly on the input data
- Convolution: Extract "local" properties of the signal. Uses a set of "filters" that have to be learned (these are the "weighted" W between layers)
- Pooling: Downsamples the outputs to reduce the size of representation
- Note: A nonlinearity is also introduced after the convolution layer

Convolution

- An operation that captures local (e.g., spatial) properties of a signal
- Mathematically, the operation is defined as

$$h_{ij}^k = g((W^k * \boldsymbol{X}_{ij}) + b_k)$$

  where W^k is the filter, $*$ is the convolution operator, and g is nonlinearity
- Usually several filters $\{W^k\}_{k=1}^K$ are applied (each will produce a separate "feature map").
- These filters have to be learned (these are the weights of the NN)

Convolution

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

These are the network parameters to be learned.

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Dot product

3 -1

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3 -3

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Pooling/Downsampling

- Used to "downsample" the representation-size after convolution step.
- Also ensures robustness against minor rotations, shifts, corruptions in the image
- Popular approaches: Max-pooling, averaging pooling, etc

Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Why Pooling

- Subsampling pixels will not change the object

bird

Subsampling

bird

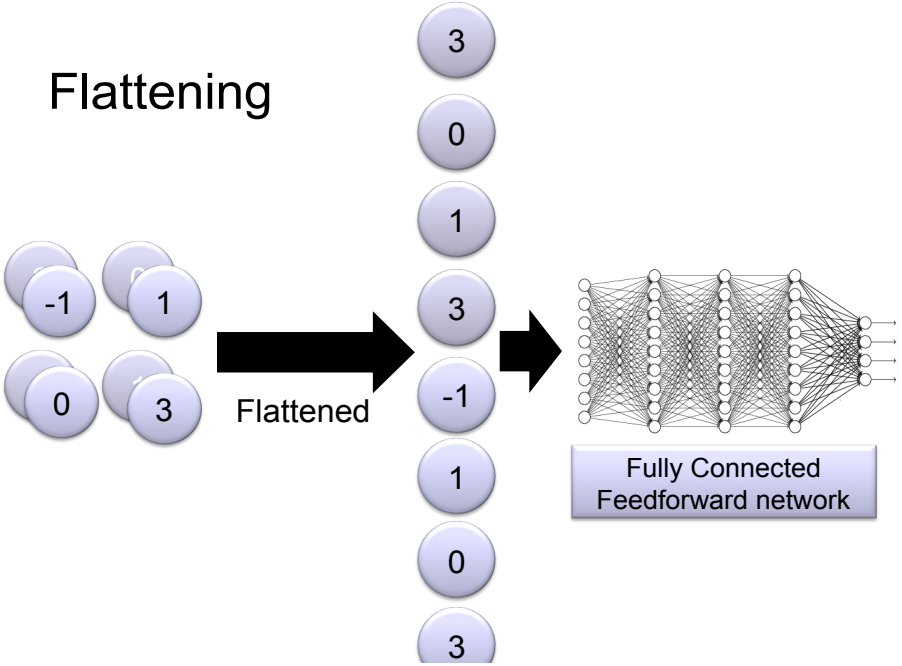
We can subsample the pixels to make image smaller

→ fewer parameters to characterize the image

Strides

- Stride defines the number of nodes a filter moves between two consecutive convolution operations
- Likewise, we have a stride to define the same when applying pooling

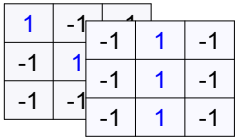
Flattening


-1
1
0
3
Flattened
3
0
1
3
-1
1
0
3
Fully Connected
Feedforward network

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***

```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```



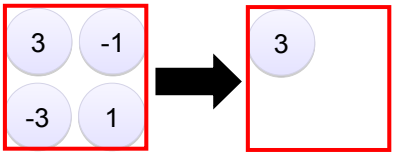
There are 25 3x3 filters.

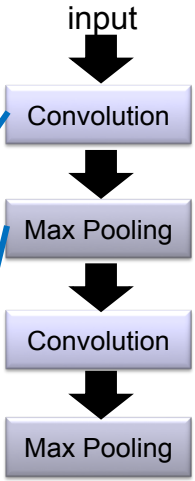
Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: black/white, 3: RGB

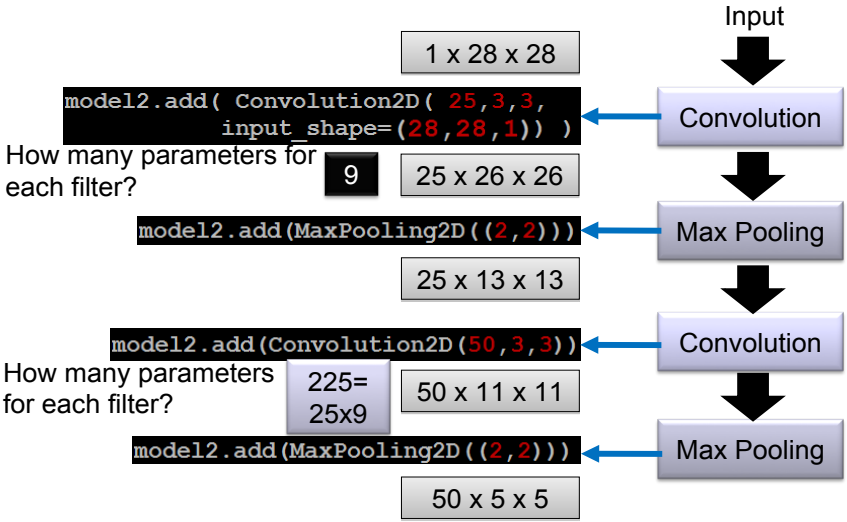
```
model2.add(MaxPooling2D((2,2)))
```





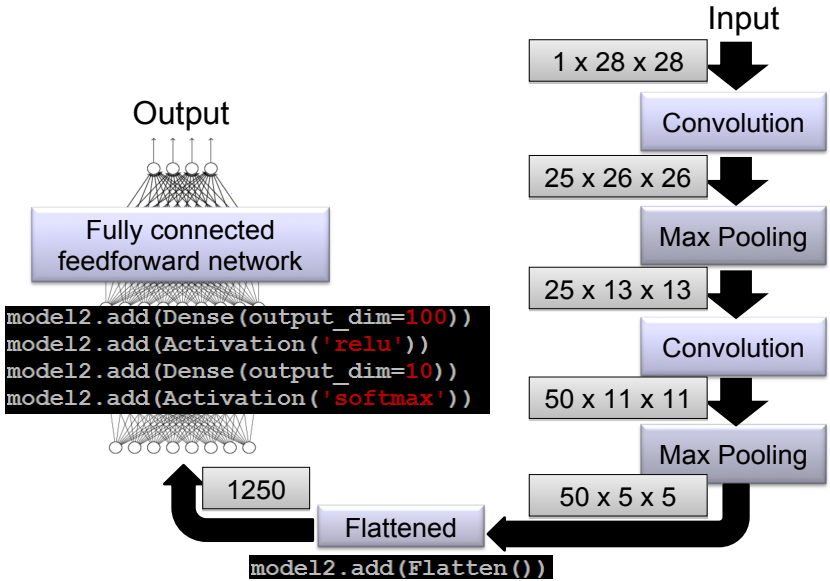
CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D array)***



CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D array)***



CNN for Sentiment Classification

See demo folder in PPT/Talks/DL

Bibliography I