

Assignment 10

Cloth Classification using RNN and CNN

Computing Lab - II, Spring 2021

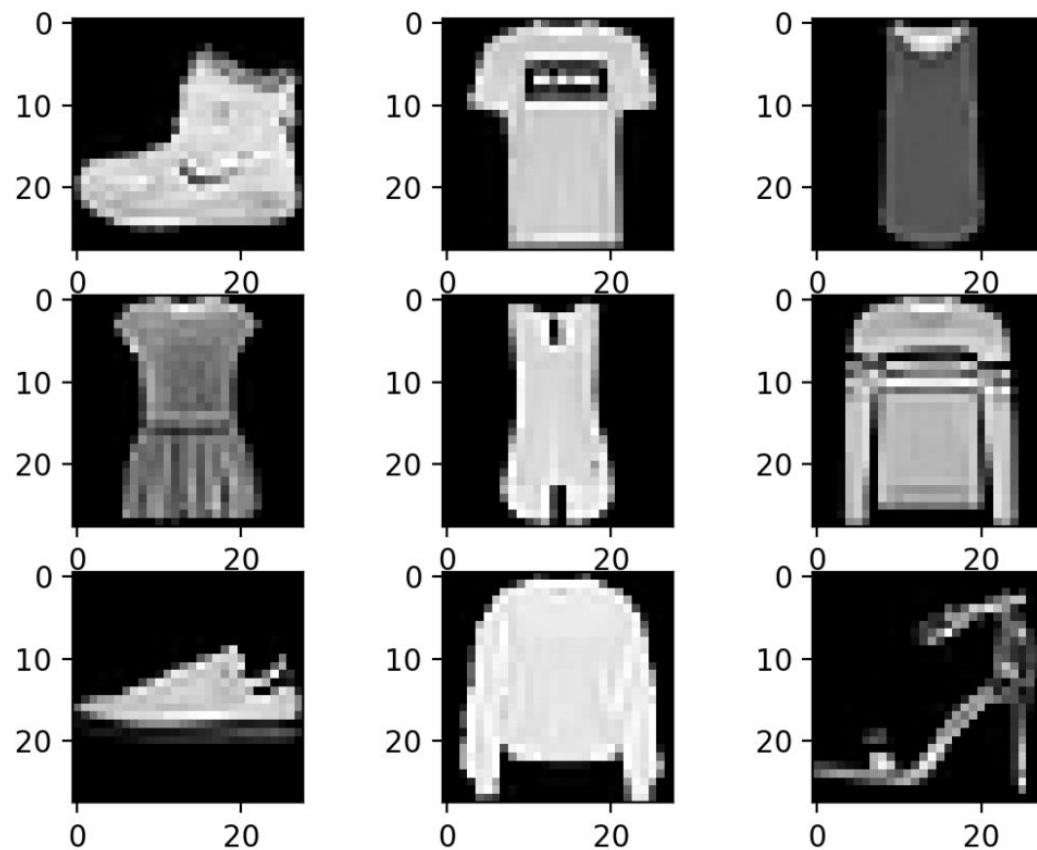
Fashion MNIST Dataset

60,000 small square 28×28 pixel grayscale images of items of 10 types of clothing, such as shoes, t-shirts, dresses, and more. The mapping of all 0-9 integers to class labels is listed below.

Task: Image Classification

- 0: T-shirt/top
- 1: Trouser
- 2: Pullover
- 3: Dress
- 4: Coat
- 5: Sandal
- 6: Shirt
- 7: Sneaker
- 8: Bag
- 9: Ankle boot

Images



Plot of a Subset of Images From the Fashion-MNIST Dataset

Tasks for the assignment

1. Use an RNN for the classification problem
2. Use a CNN for the same task

Recurrent Neural Networks

Task 1

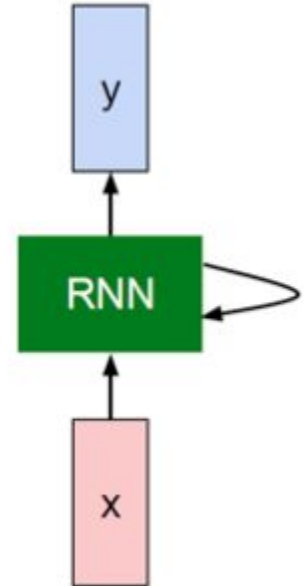
Recurrent Neural Network

We can process a sequence of vectors x by applying a recurrence formula at each step:

$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state some function with parameters W old state input vector at some time step

Notice: the same function and the same set of parameters are used at every time step.



RNN - further details

RNN update equations (forward pass)

Update Equations

Initial state - $h^{(0)}$

From $t = 1$ to $t = \tau$, the following update equation is applied:

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)}$$

$$h^{(t)} = \tanh(a^{(t)})$$

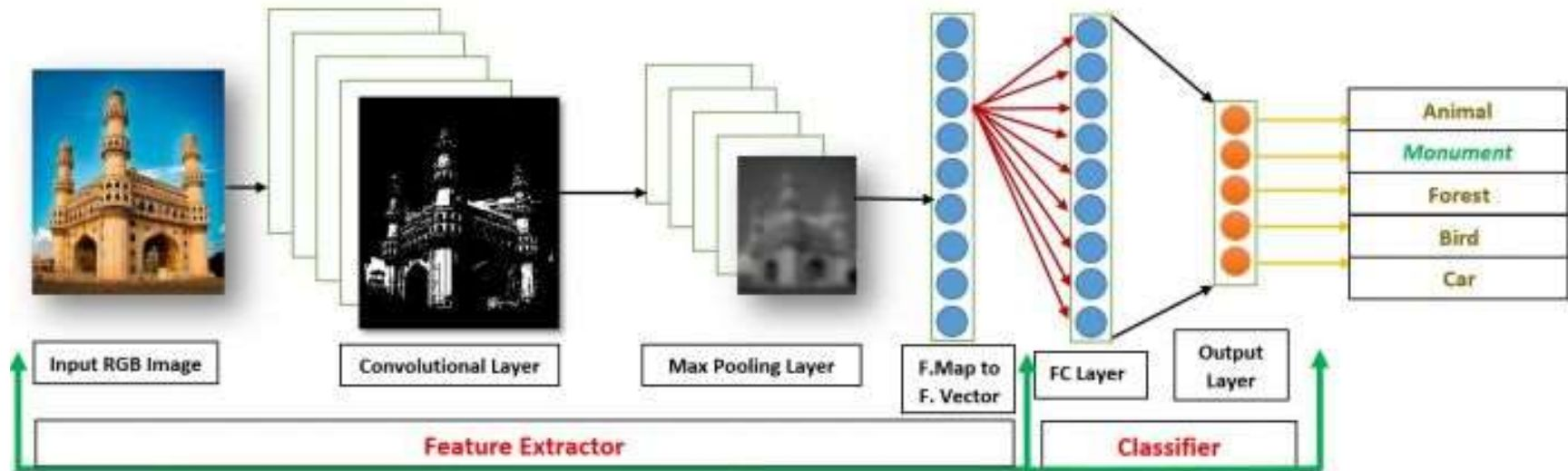
$$o^{(t)} = c + Vh^{(t)}$$

$$\hat{y}^{(t)} = \text{softmax}(o^{(t)})$$

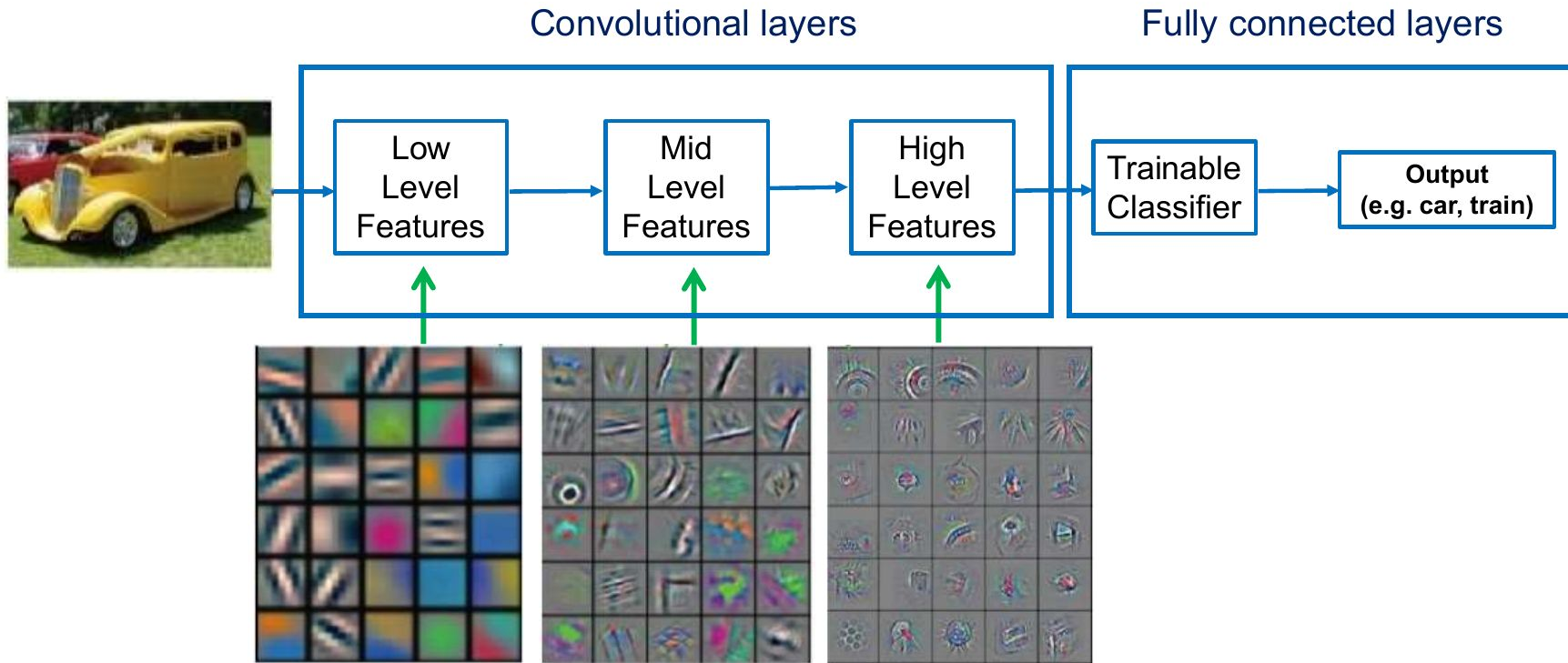
Convolutional Neural Networks

Introduction

Convolutional Neural Networks (CNNs) learn multi-level features and classifier in a joint fashion and performs much better than traditional approaches for various image classification and segmentation problems.



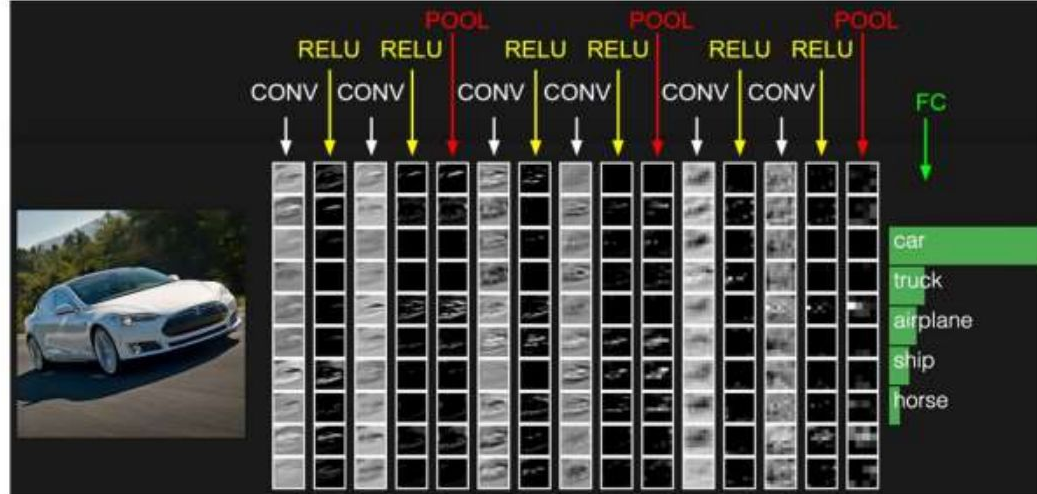
CNN – What do they learn?



CNN - Components

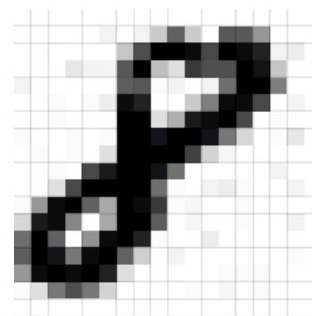
There are four main components in the CNN:

1. Convolution
2. Non-Linearity
3. Pooling or Sub Sampling
4. Classification (Fully Connected Layer)



Input

- An Image is a matrix of pixel values.
- If we consider a gray scale image, the value of each pixel in the matrix will range from 0 to 255.
- If we consider an RGB image, each pixel will have the combined values of R, G and B.



What We See

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 82 12 50 77 91 08
49 49 99 40 17 81 18 57 40 87 17 40 88 43 49 48 04 86 42 00
81 49 31 73 55 78 14 23 93 71 40 47 53 68 30 03 49 13 36 45
92 70 95 23 04 60 11 42 49 24 46 56 01 32 56 71 37 02 36 91
22 31 14 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 50
24 47 32 60 99 03 85 02 44 78 33 53 78 34 84 20 35 17 12 50
32 98 81 28 44 23 47 10 24 38 40 47 59 54 70 44 18 38 44 70
47 24 20 69 02 42 12 20 95 43 94 33 43 08 40 91 66 49 94 21
24 33 58 08 44 73 89 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 76 00 74 44 20 45 35 14 00 40 33 97 34 31 33 95
78 17 33 28 22 75 31 67 15 94 03 80 04 42 16 14 09 53 56 92
14 38 05 42 94 35 31 47 55 58 58 24 00 17 54 24 96 29 85 57
86 54 00 45 35 71 89 07 05 44 44 37 44 40 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 82 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 14 07 97 57 32 16 26 26 79 33 27 98 66
88 34 48 87 57 42 20 72 03 46 33 47 46 55 13 32 43 93 53 49
04 42 14 73 35 20 30 11 24 94 72 18 08 46 29 32 40 42 76 94
20 49 34 41 72 30 23 88 34 42 89 49 82 47 59 55 74 04 36 14
20 73 35 29 78 31 80 01 74 31 49 71 48 86 81 16 23 87 05 54
01 70 54 71 83 51 54 49 14 92 33 48 41 43 52 01 89 19 47 48
```

What Computers See

Convolution

The primary purpose of Convolution in case of a CNN is to extract features from the input image.

1	0	1
0	1	0
1	0	1

Filter / Kernel / Feature detector

Image

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

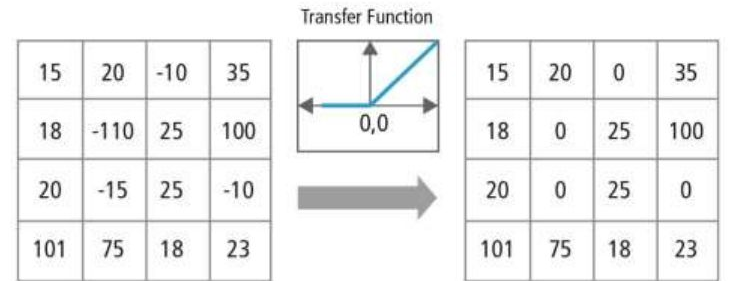
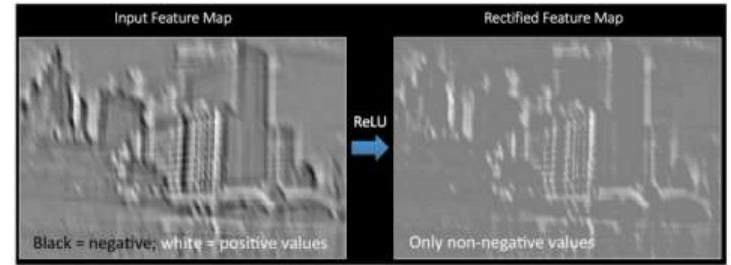
Convolved Feature /
Activation Map /
Feature Map

Convolution...

- The size of the output volume is controlled by three parameters that we need to decide before the convolution step is performed:
 - ✓ **Depth:** Depth corresponds to the number of filters we use for the convolution operation.
 - ✓ **Stride:** Stride is the number of pixels by which we slide our filter matrix over the input matrix.
 - ✓ **Zero-padding:** Sometimes, it is convenient to pad the input matrix with zeros around the border, so that we can apply the filter to bordering elements of our input image matrix.
 - With zero-padding *wide convolution*
 - Without zero-padding *narrow convolution*

Non-Linearity (ReLU)

- Replaces all negative pixel values in the feature map by zero.
- The purpose of ReLU is to introduce non-linearity in CNN, since most of the real-world data would be non-linear.
- Other non-linear functions such as tanh $(-1,1)$ or sigmoid $(0,1)$ can also be used instead of ReLU $(0,\text{input})$.



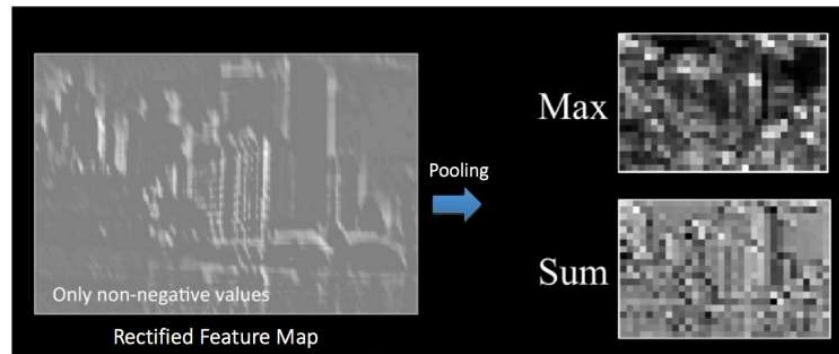
Pooling

Reduces the dimensionality of each feature map but retains the most important information. Pooling can be of different types: Max, Average, Sum etc.

1	3	2	9
7	4	1	5
8	5	2	3
4	2	1	4

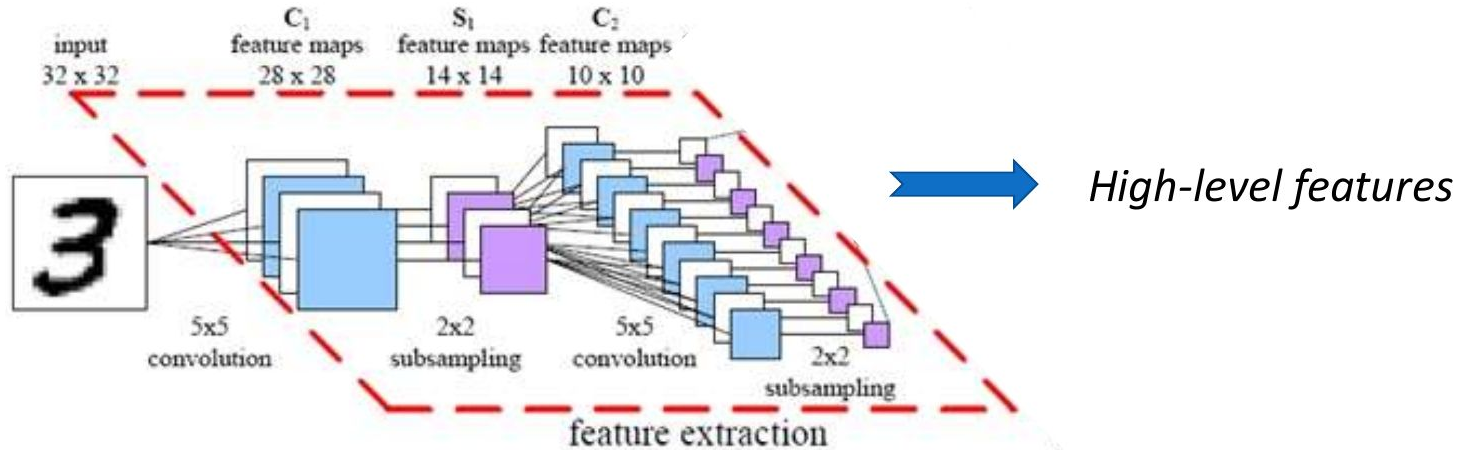
2x2 region

7	9
8	



Story so far

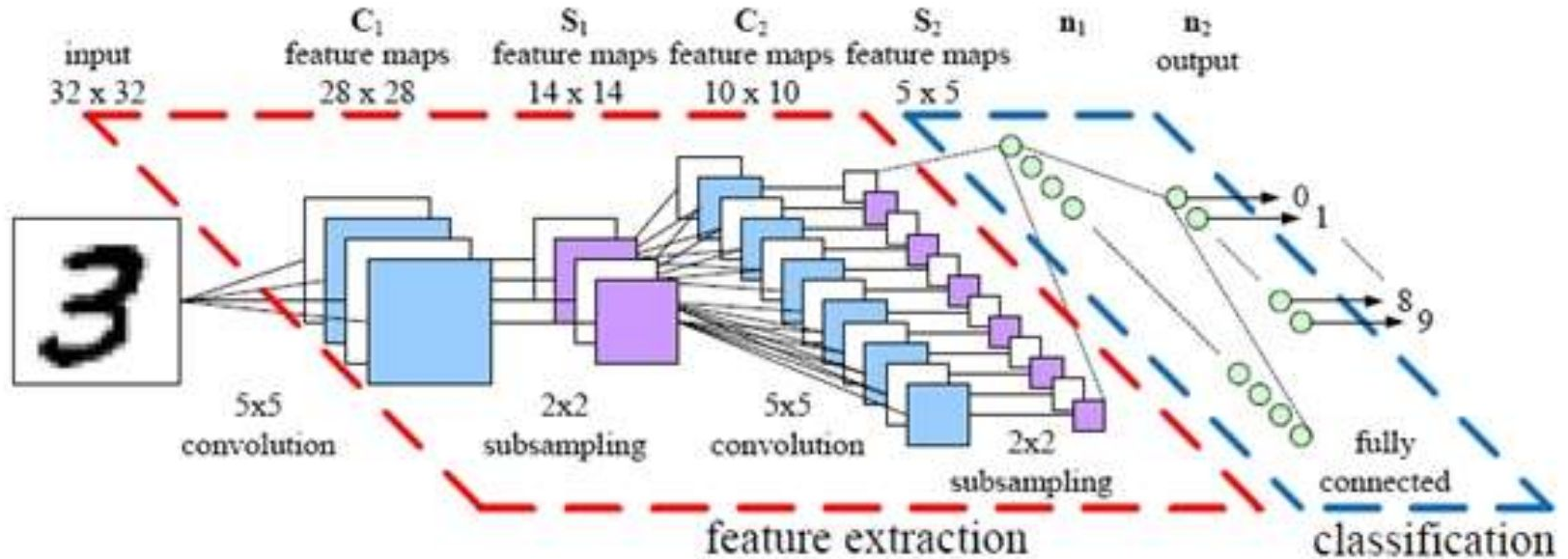
- Together these layers extract the useful features from the images.
- The output from the convolutional and pooling layers represent high-level features of the input image.



Fully Connected Layer

- A traditional Multi-Layer Perceptron.
- The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer.
- Their activations can hence be computed with a matrix multiplication followed by a bias offset.
- The purpose of the Fully Connected layer is to use the high-level features for classifying the input image into various classes based on the training dataset.

Overall CNN Architecture



Summary. To summarize, the Conv Layer:

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
 - Number of filters K ,
 - their spatial extent F ,
 - the stride S ,
 - the amount of zero padding P .
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
 - $W_2 = (W_1 - F + 2P)/S + 1$
 - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
 - $D_2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.
- In the output volume, the d -th depth slice (of size $W_2 \times H_2$) is the result of performing a valid convolution of the d -th filter over the input volume with a stride of S , and then offset by d -th bias.