Convolutional Neural Network

MGTF 495

Class Outline

- Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

Key Terminologies

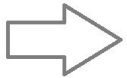
- **Input Image**
  - RGB image has 3 channels

What We See

What Computers See

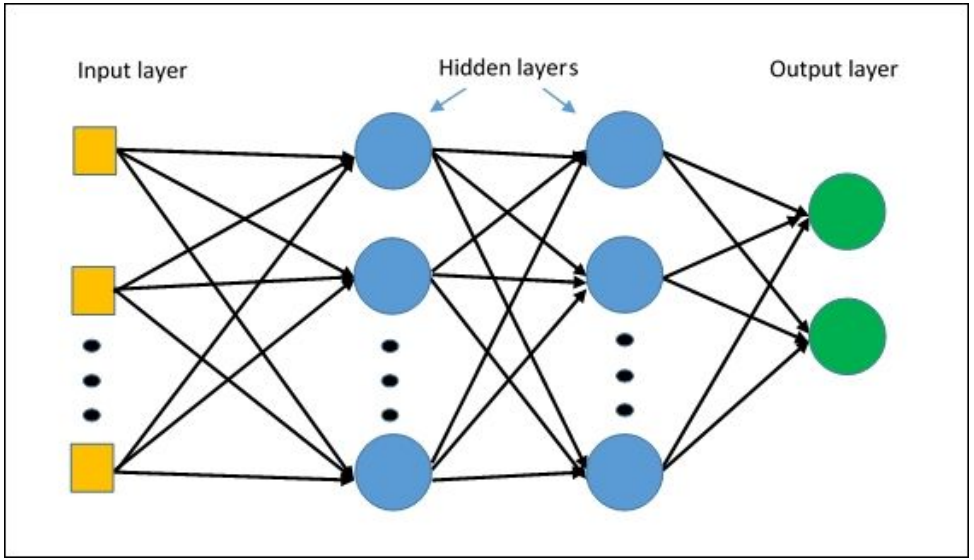
Multilayer Perceptron

1	1	0
4	2	1
0	2	1



1
1
0
4
2
1
0
2
1

Flatten the image



Feed the image to MLP

Why not Multilayer Perceptron ?

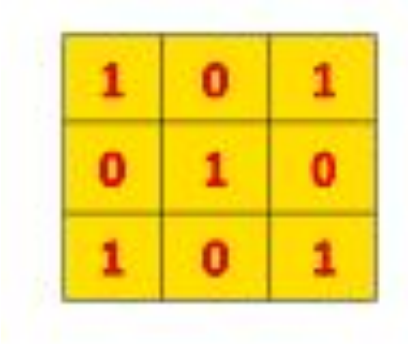
- If we flatten an image, we lose spatial information.
- Any pixel in image is related to its surrounding pixels.
- How to exploit the spatial information?

Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

Key Terminologies

- Input Image
- **Kernel/Filter/Feature Extractor**
  - Filters are matrix of numbers.
  - They extract interesting features from the image.
  - The extracted features are called **Feature Map/Activation map**

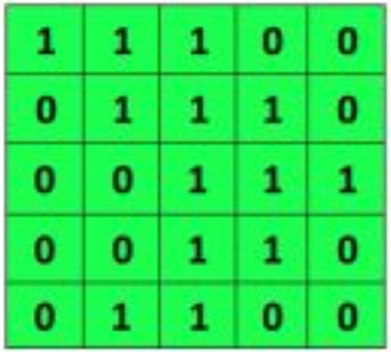


Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

Convolution - 2D

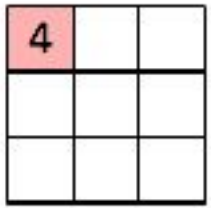
- Input Image
- **Kernel/Filter/Feature Extractor**
  - Filters are matrix of numbers.
  - They extract interesting features from the image.
  - Different filters extract different features.



Input image

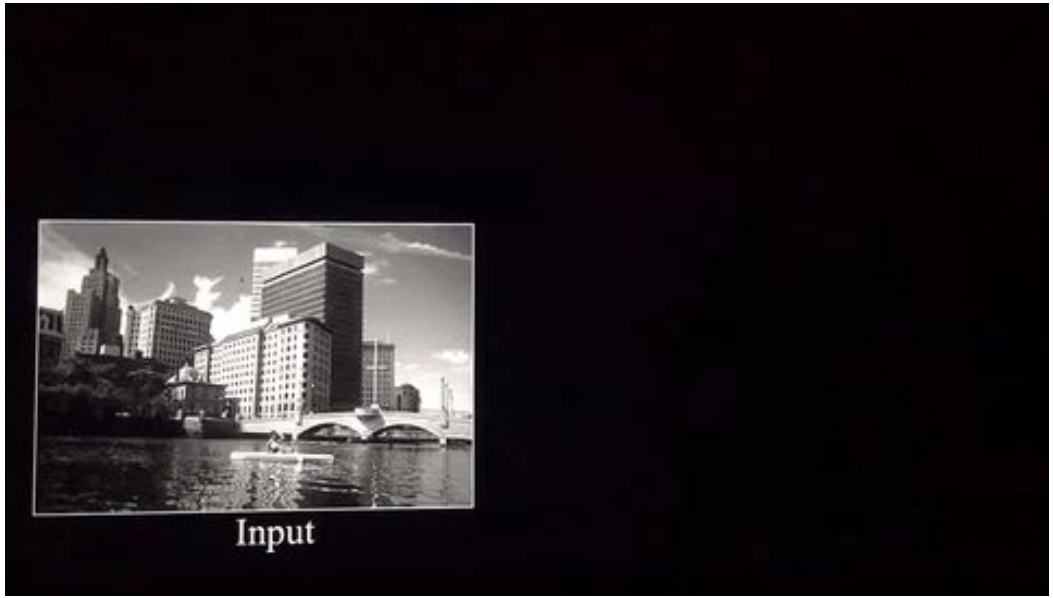
Kernel/Filter

Image



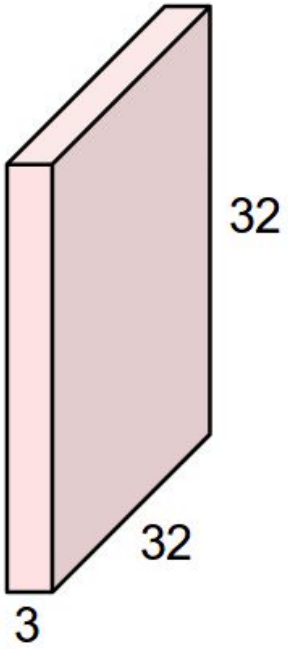
Convolved Feature

Convolution - 2D

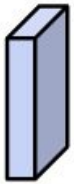


Convolutional Neural Network(CNN): Architecture

32x32x3 image



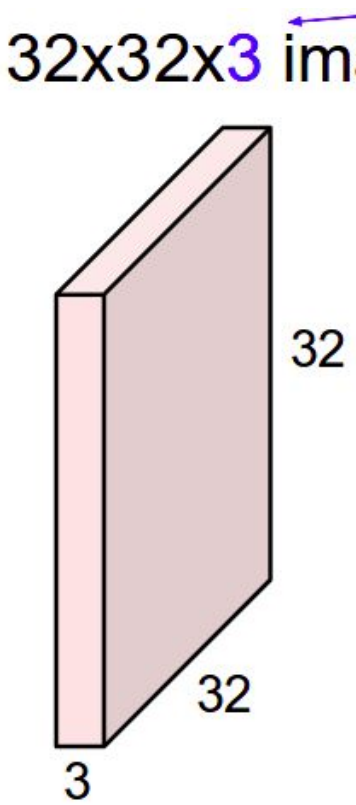
5x5x3 filter

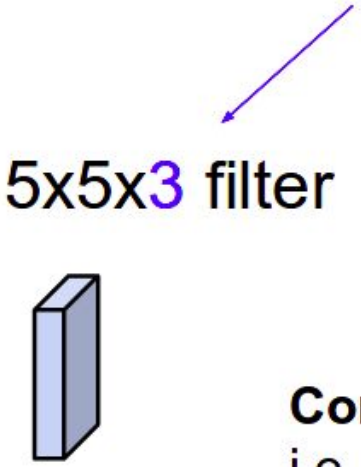


Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

CNN: Architecture

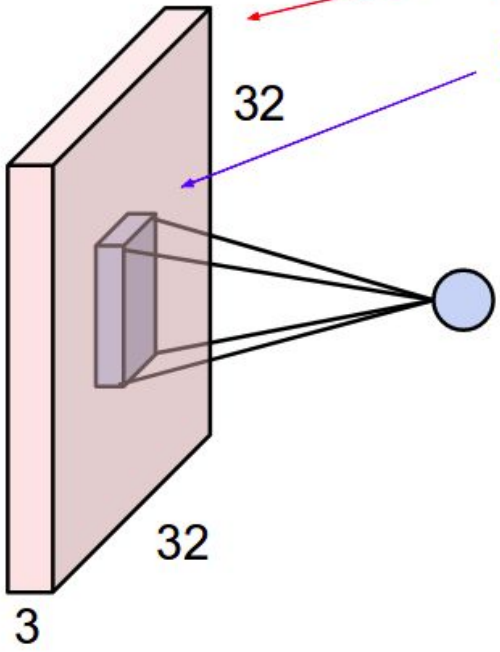
Filters always extend the full depth of the input volume





Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

CNN: Architecture



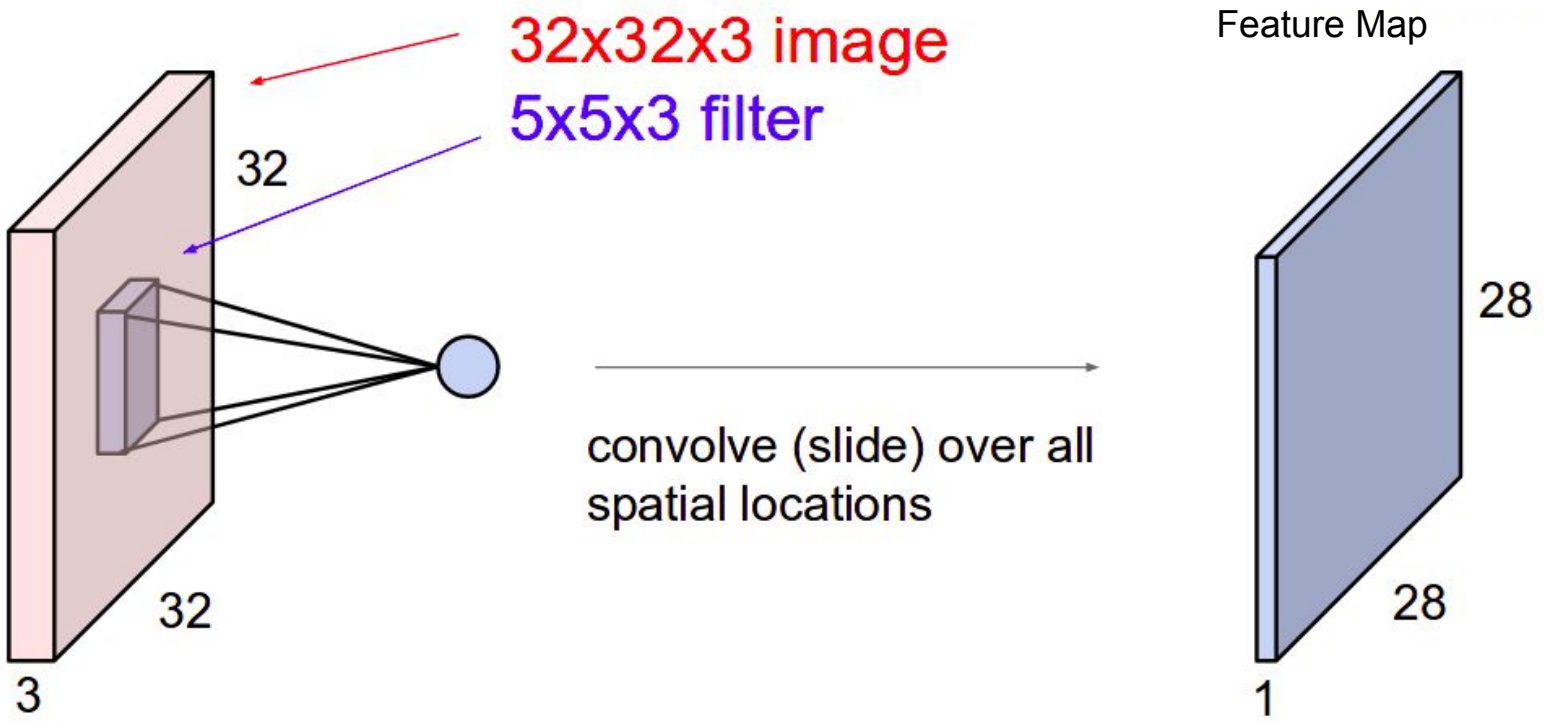
32x32x3 image

5x5x3 filter w

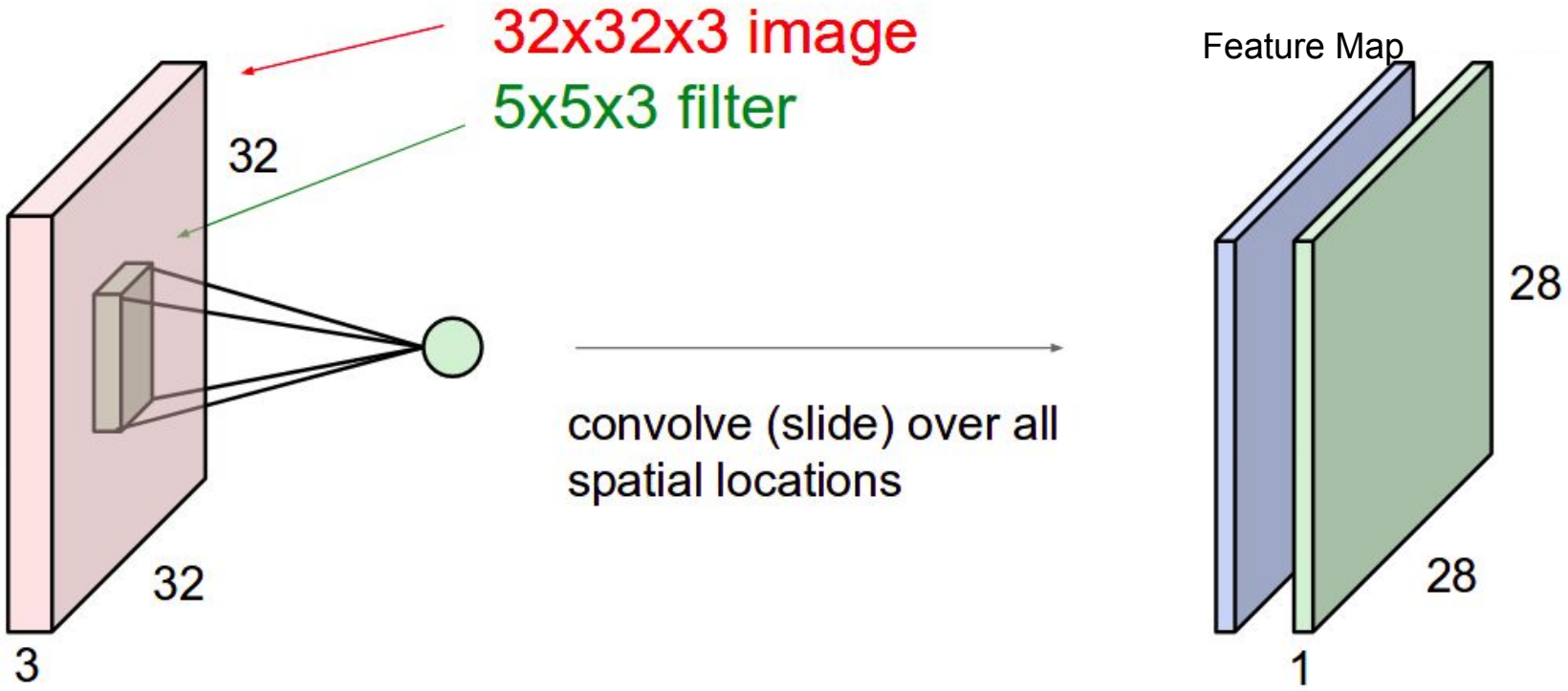
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

CNN: Architecture

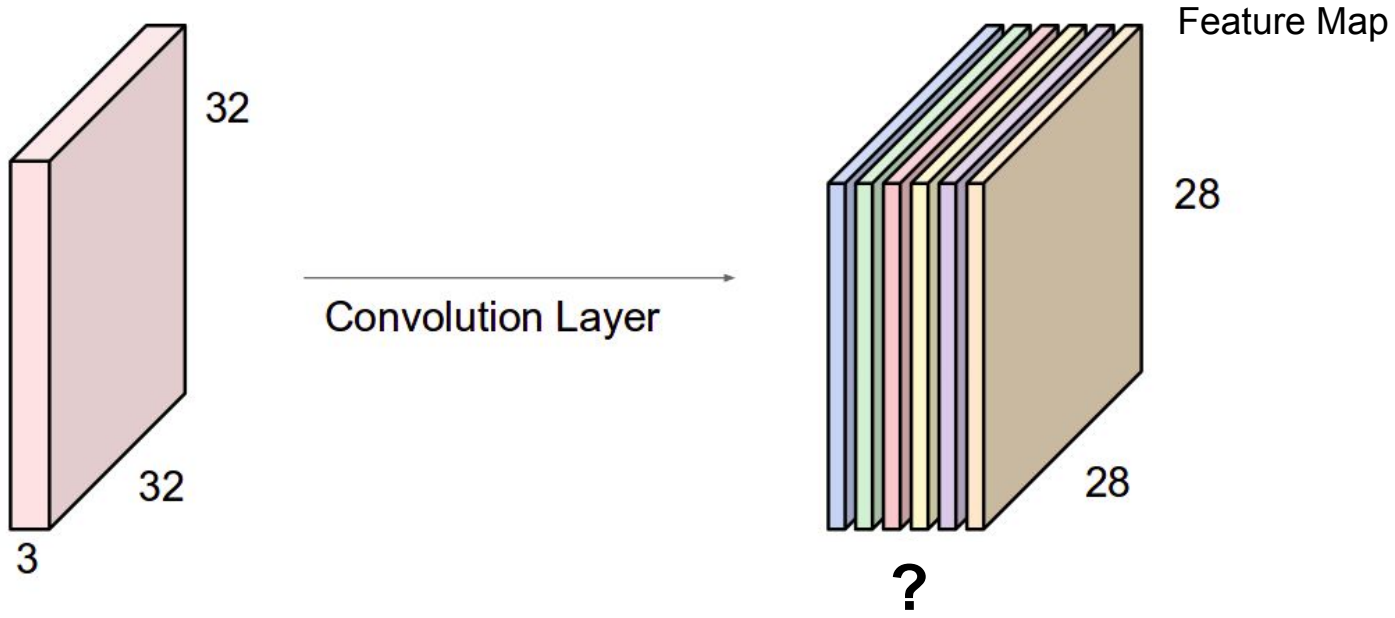


CNN: Architecture



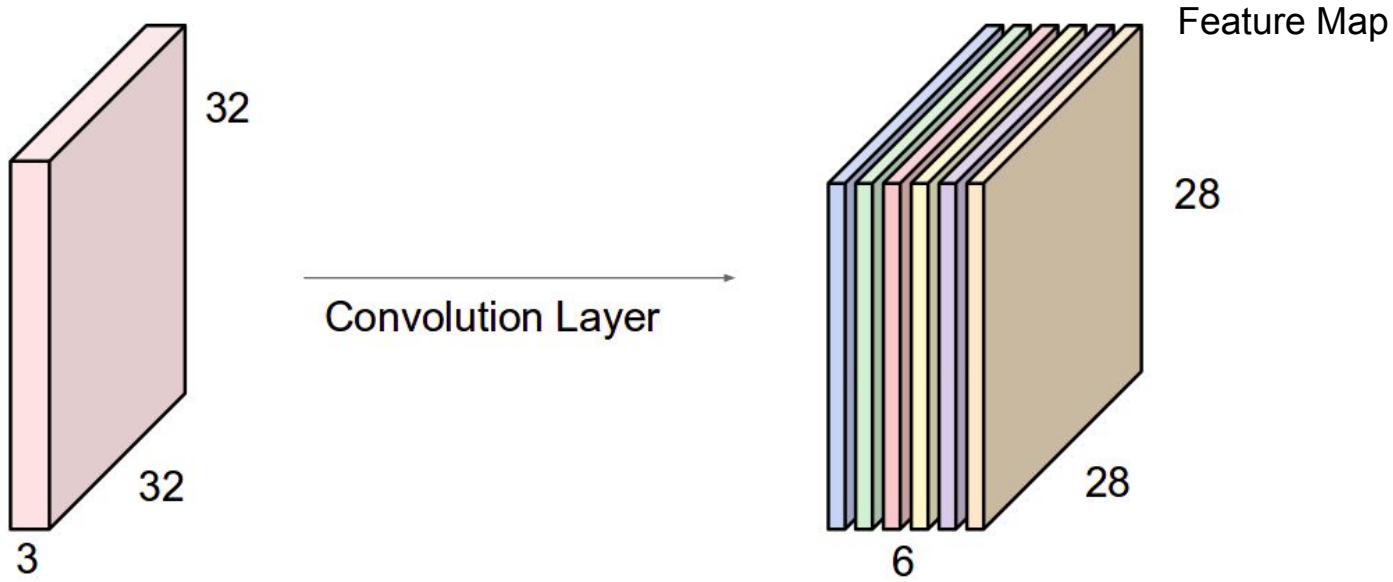
CNN: Architecture

For eg, if we have 6 5x5x3 filters,how many feature maps do we get?



CNN: Architecture

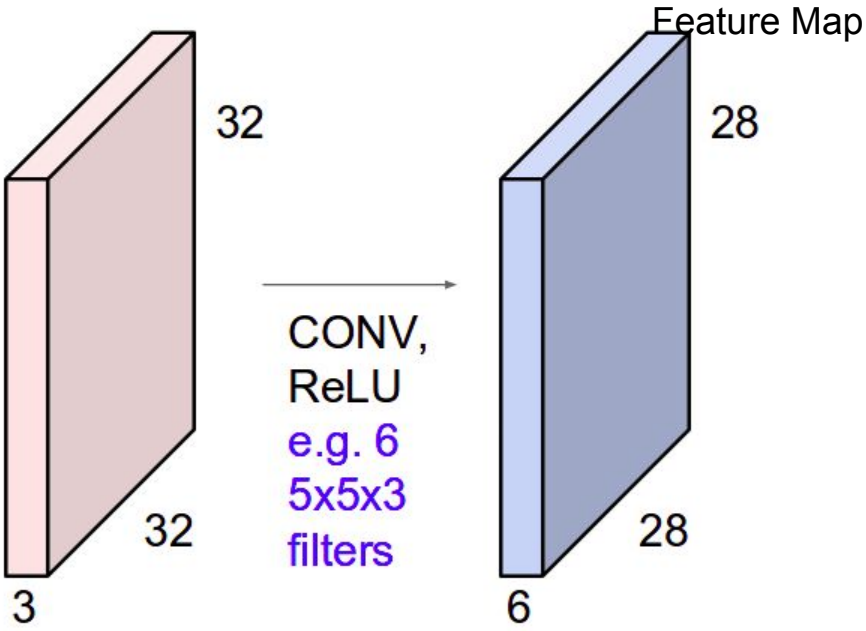
We get 6 feature maps!!!



We stack these up to get a “new image” of size 28x28x6!

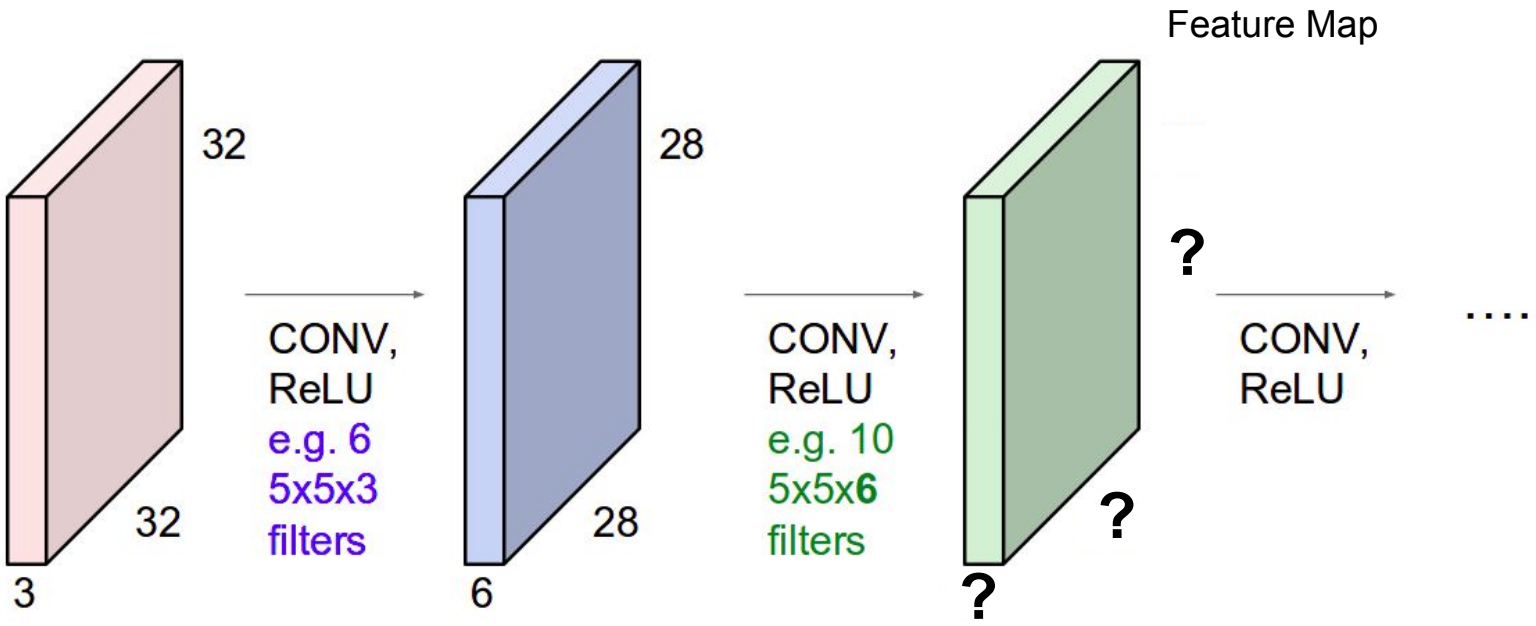
CNN: Architecture

ConvNet is a sequence of Convolutional Layers, separated by activation functions



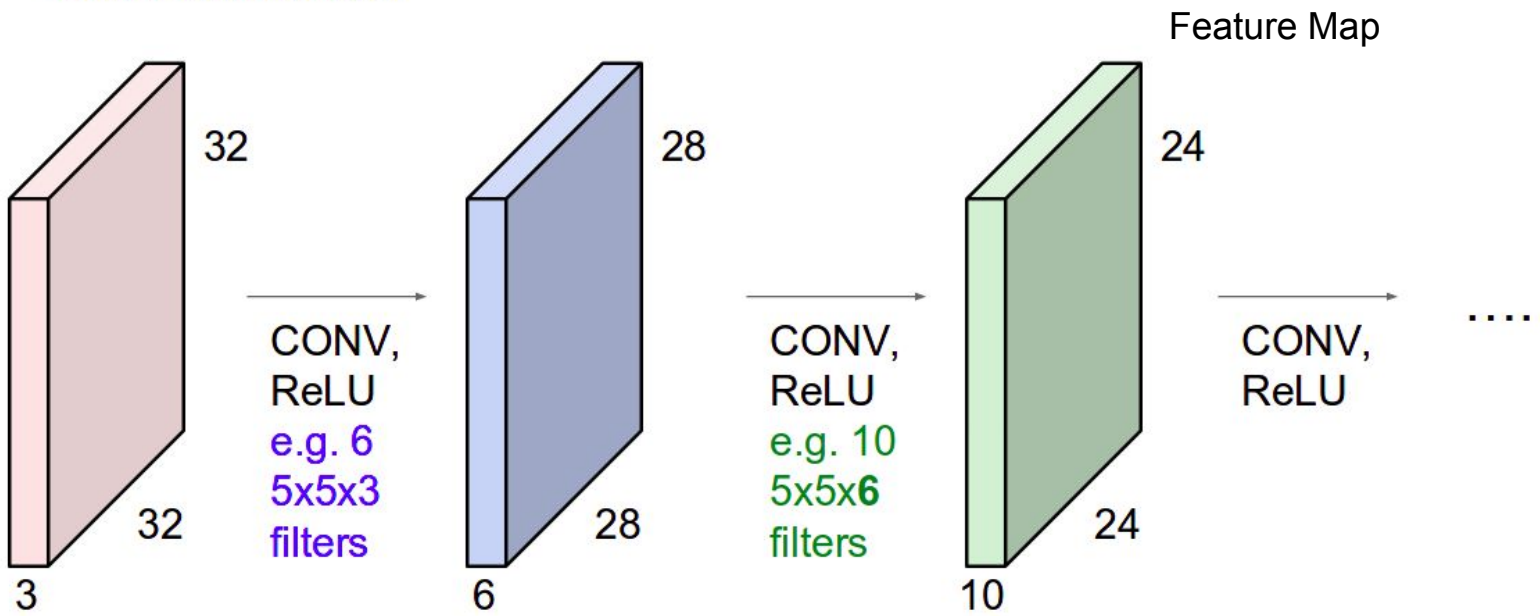
CNN: Architecture

ConvNet is a sequence of Convolutional Layers, separated by activation functions



CNN: Architecture

ConvNet is a sequence of Convolutional Layers, separated by activation functions



Convolutional Neural Network : Key Idea

- The idea of convolution of a filter over image was always there.
- Earlier, these filters were manually designed.
- With CNN, these filters are learnt.

-1	**0**	**1**
-1	**0**	**1**
-1	**0**	**1**

Filter: Detects Vertical Edge

-1	**-1**	**-1**
0	**0**	**0**
1	**1**	**1**

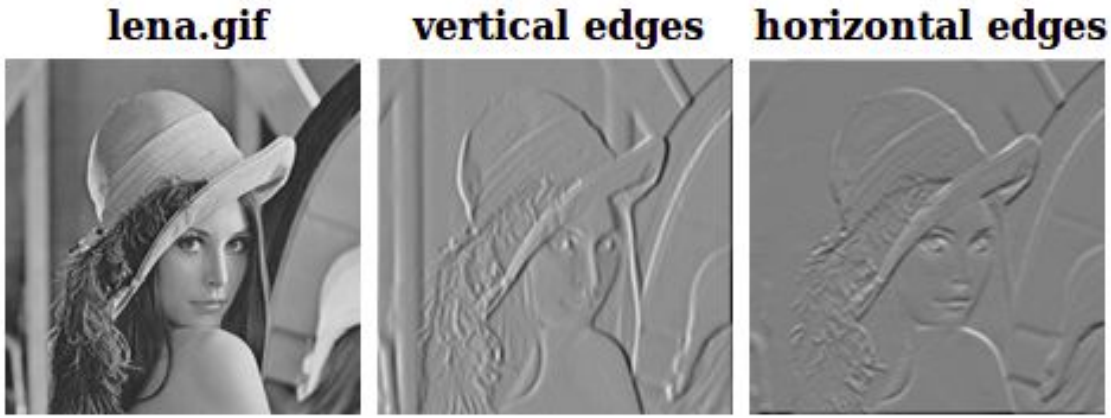
Filter: Detects Horizontal Edge

Convolutional Neural Network : Key Idea

- The idea of convolution of a filter over image was always there.
- Earlier, these filters were manually designed.
- With CNN, these filters are learnt.

-1	0	1
-1	0	1
-1	0	1

Filter: Detects Vertical Edge



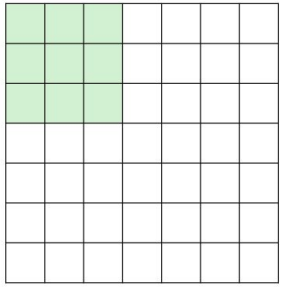
-1	-1	-1
0	0	0
1	1	1

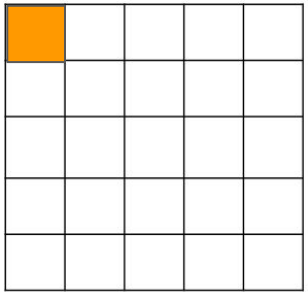
Filter: Detects Horizontal Edge

Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

Strides 1

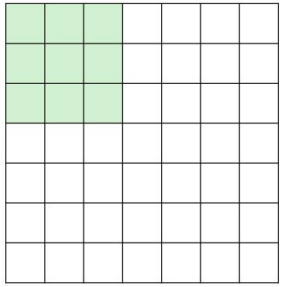


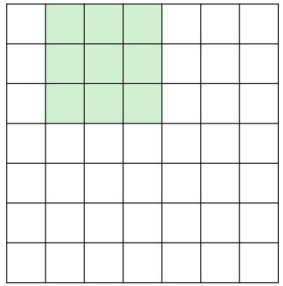


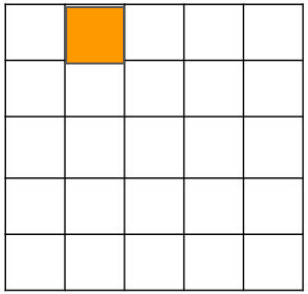
Generated feature map

Input Image Size: 7x7

Strides 1



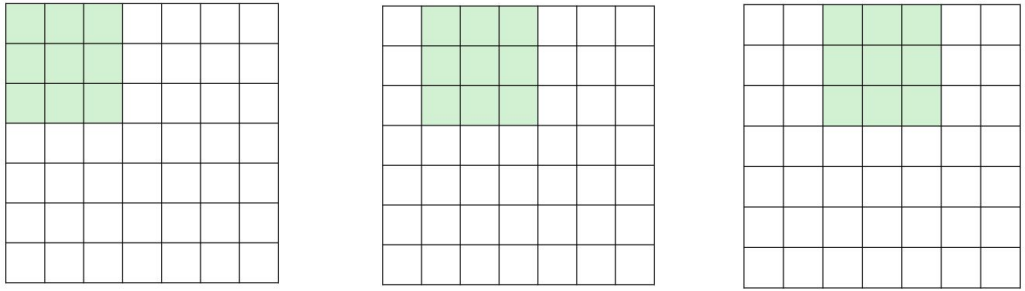


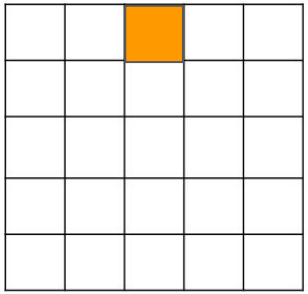


Generated feature map

Input Image Size: 7x7

Strides 1

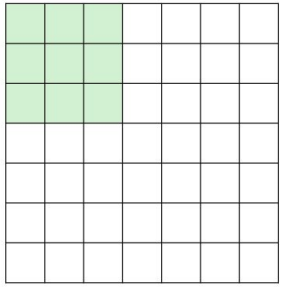


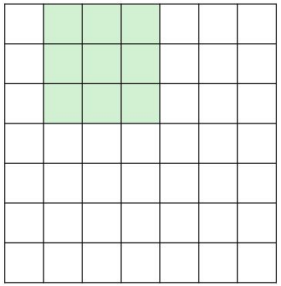


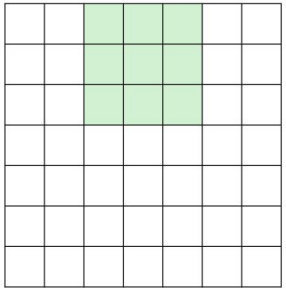
Generated feature map

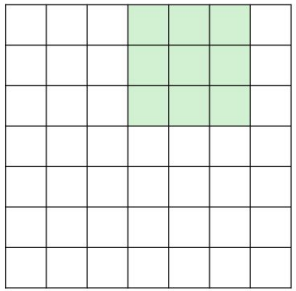
Input Image Size: 7x7

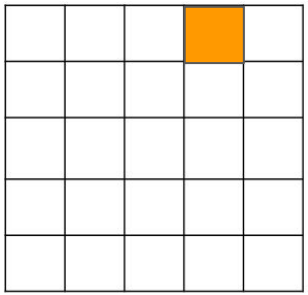
Strides 1







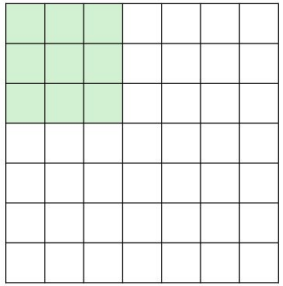


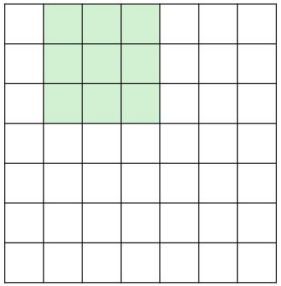


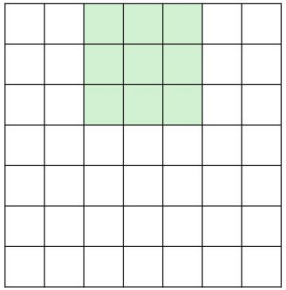
Generated feature map

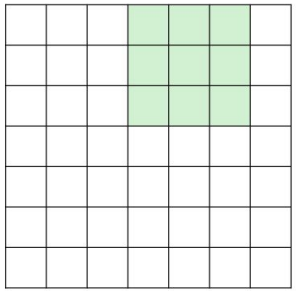
Input Image Size: 7x7

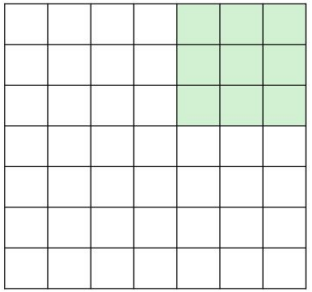
Strides 1

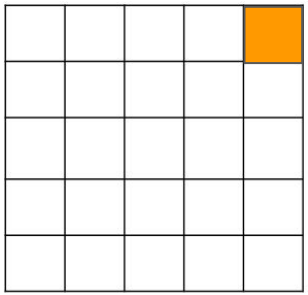








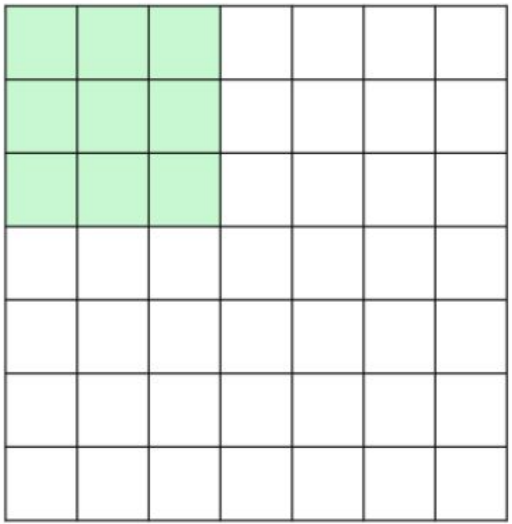


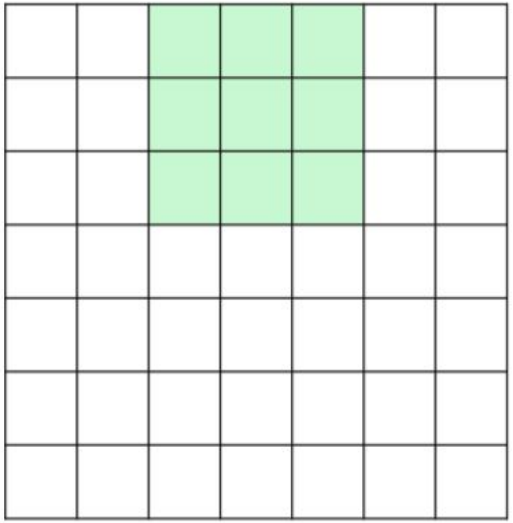


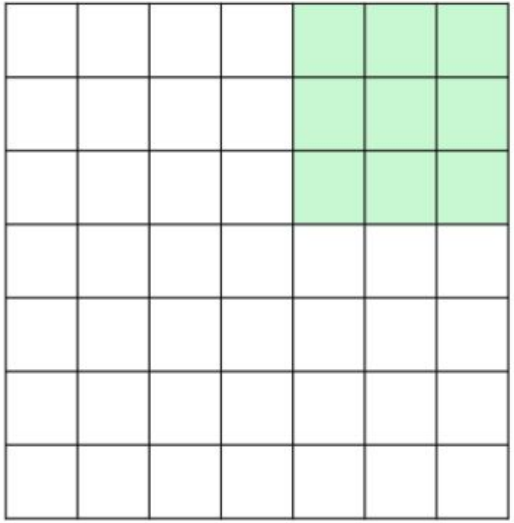
Generated feature map

Input Image Size: 7x7
Feature Map Size: 5x5

Strides 2

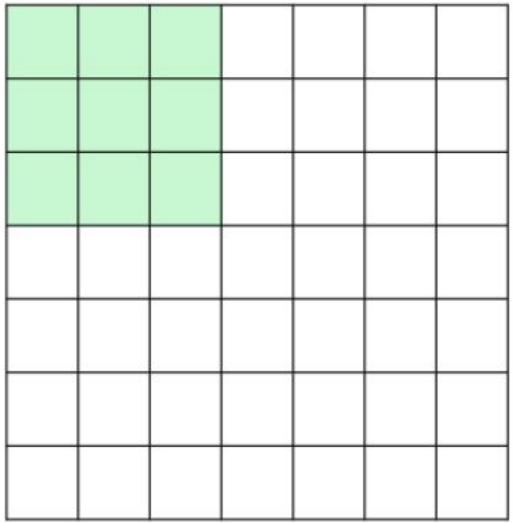






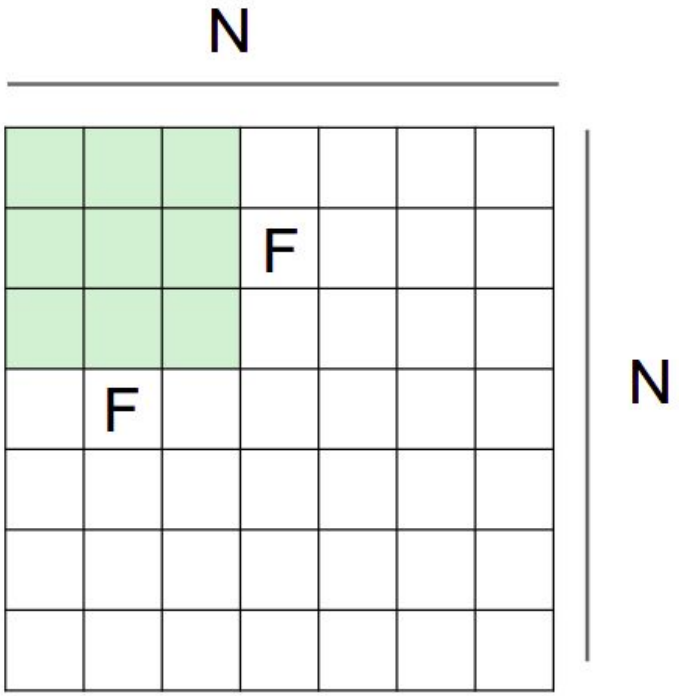
- Feature Map is of size: 3x3

Strides 3



- Strided(3) convolution not possible on image of size 7x7

Feature map dimensions



Output size:
(N - F) / stride + 1

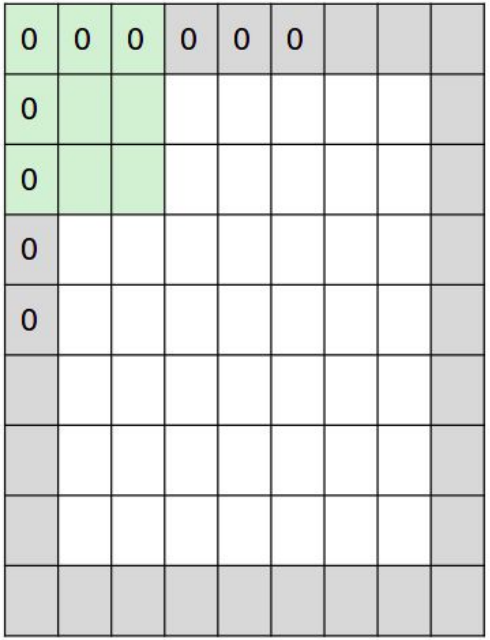
e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33

Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

Padding

In practice: Common to zero pad the border



e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

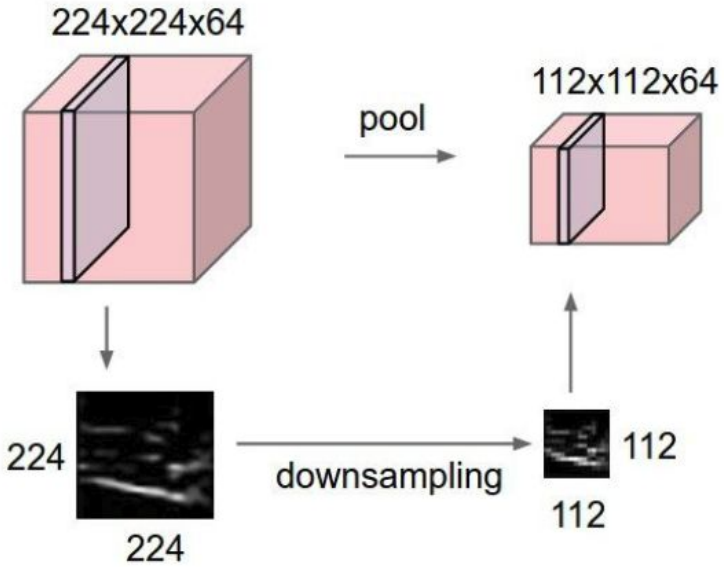
Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

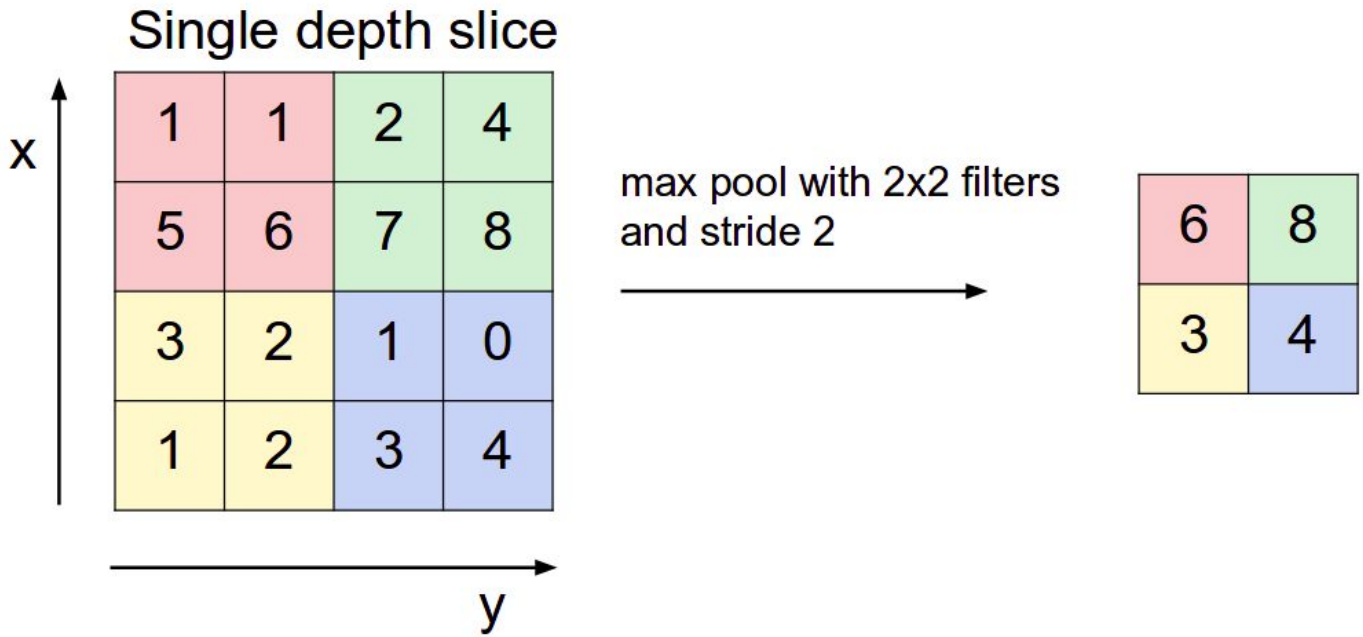
Subsampling: Pooling

Pooling layer

- Makes representation smaller and more manageable
- Operates over each feature map independently.
- Has no activation function.



Max-pooling

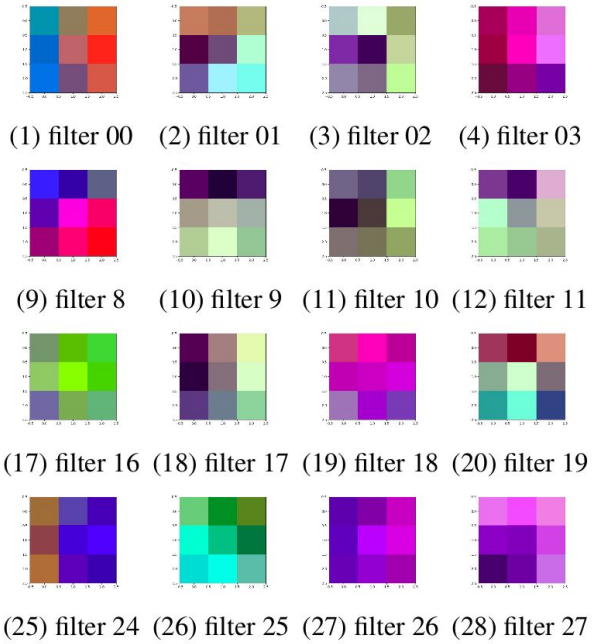

Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

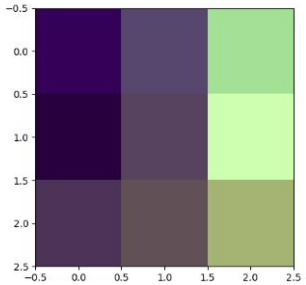
CNN: Visualization of filters

Input image

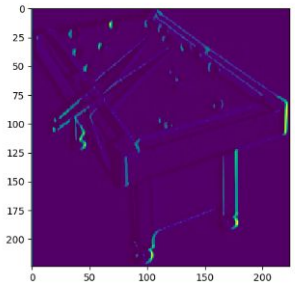


Feature map from first convolution layer

CNN: Visualization of filters - Conv 1 layer

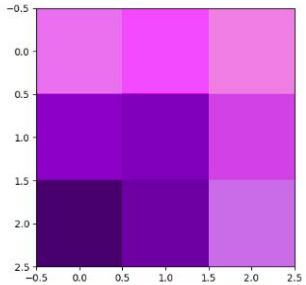


(1) Filter 6 (blue line)

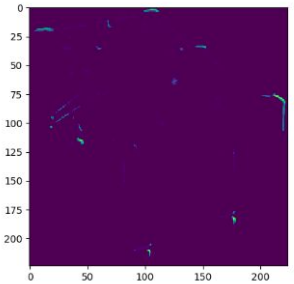


(2) Activation/Features from Filter 6 (dark edges)

CNN: Visualization of filters - Conv 1 layer

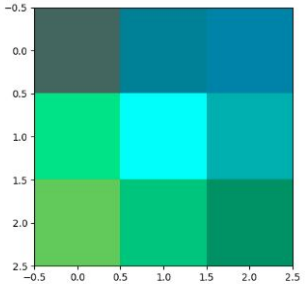


(3) Filter 27 (dark corner)

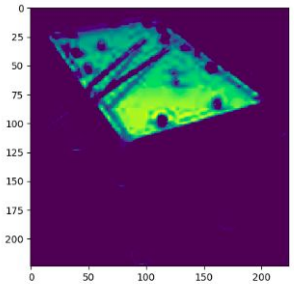


(4) Activation/Features from Filter 27

CNN: Visualization of filters - Conv 1 layer

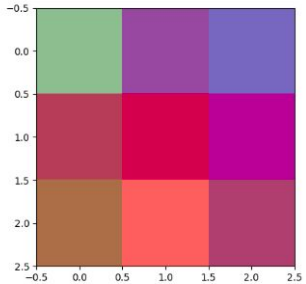


(5) Filter 15 (green surface)

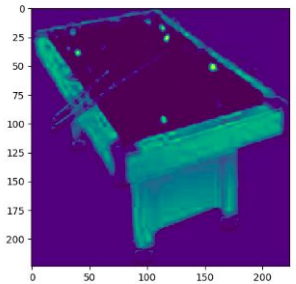


(6) Activation/Features from Filter 15

CNN: Visualization of filters - Conv 1 layer



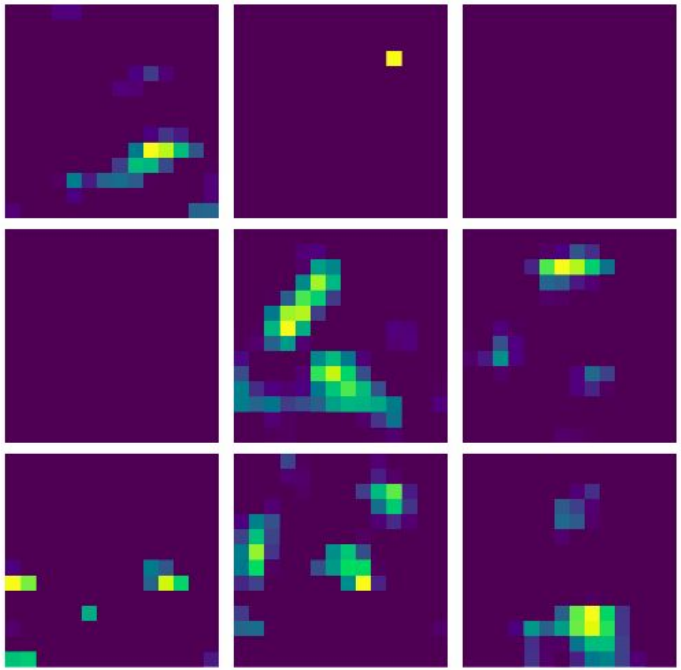
(7) Filter 47 (yellow surface)



(8) Activation/Features from Filter 47

Deeper layers

Feature Maps from deeper layers are very difficult to interpret.



Deep Visualization Toolbox

https://www.youtube.com/watch?v=AgkfIQ4IGaM

Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

What happens after Convolution & Activation ?

- Feature extraction is done by Convolution
  - The extracted features are 3D (Feature Maps)

What happens after Convolution & Activation ?

- Feature extraction is done by Convolution
  - The extracted features are 3D (Feature Maps)

- How to do classification using feature maps?

What happens after Convolution & Activation ?

- Feature extraction is done by Convolution
  - The extracted features are 3D (Feature Maps)
- How to do classification using feature maps?
- For classification, we need to add fully connected layers (MLP)
  - After last Convolutional Feature Map
  - Flatten the image - i.e. convert the 3D image to a 1D vector
  - Add Dense/Fully Connected layers.
  - Add classification head (output layer)

Class Outline

- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

CNN Architecture: Overview

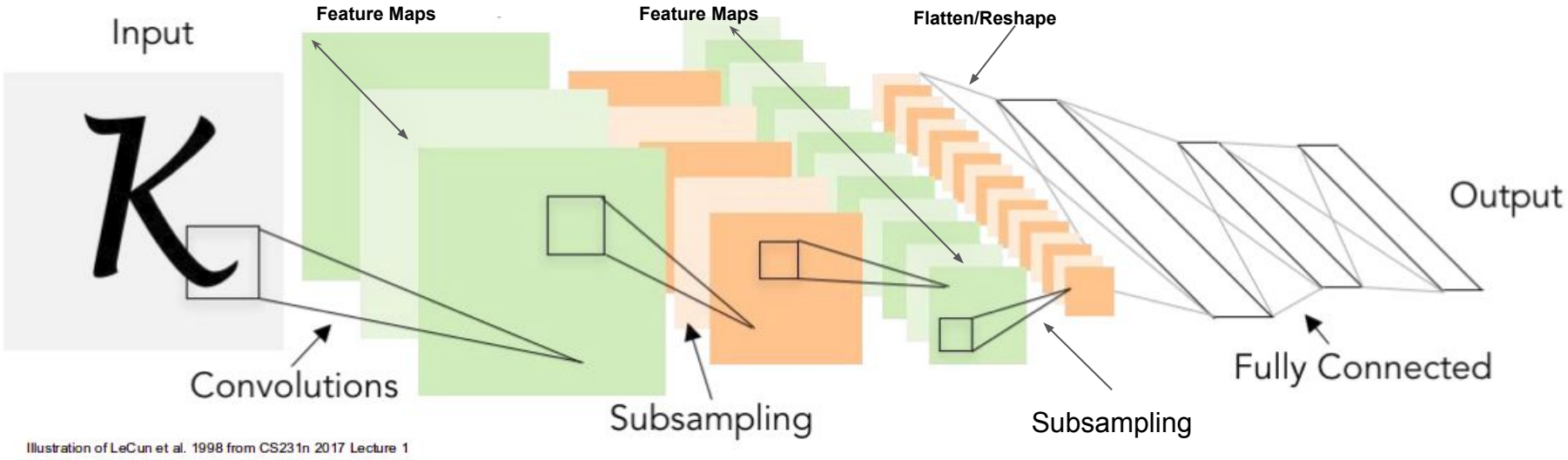
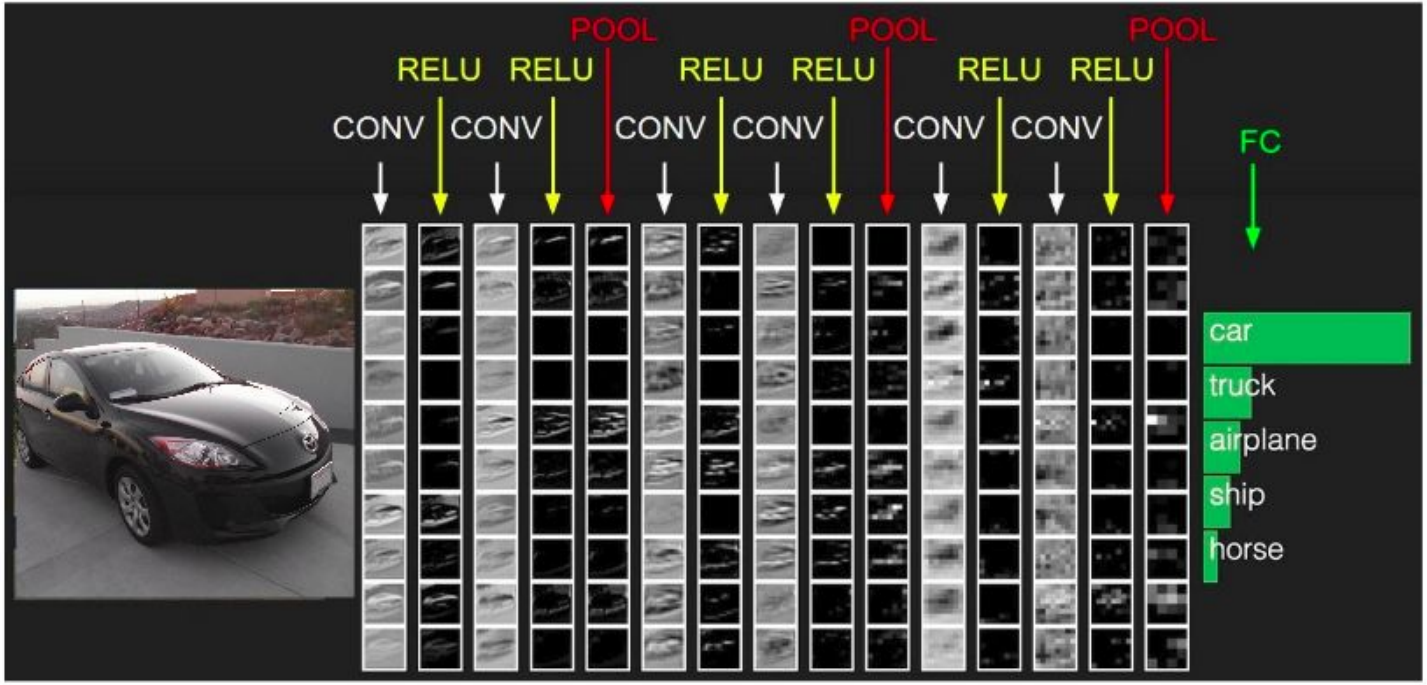


Illustration of LeCun et al. 1998 from CS231n 2017 Lecture 1

Model Summary in Keras

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 9216)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               1179776
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 1,199,882
Trainable params: 1,199,882
Non-trainable params: 0
_________________________________________________________________
```

Overall Architecture



Demo

Demo

Class Outline

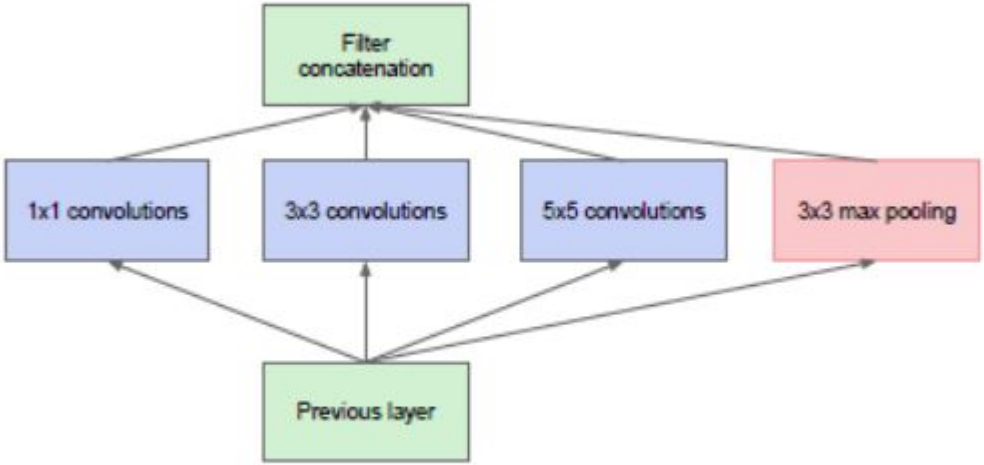
- Problem with Multilayer Perceptron
- CNN Architecture
  - Kernel
  - Convolution 2D
  - Strides
  - Padding
  - Subsampling/Pooling
  - Feature Visualization
  - What happens after Convolution
  - Overview
- Latest trends in CNN

Latest trends

- In 2012, Neural Networks were typically ~ 8 layer deep.
- Now networks are more than 100 layers deep.
- Most commonly used networks are:
  - Google: GoogLeNet
  - Microsoft: ResNet

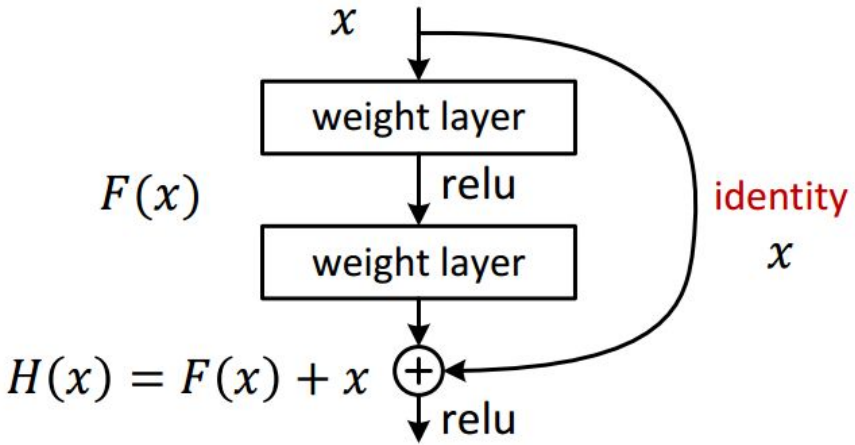
GoogLe Net : Inception Module

- Let the network decide the best filter size by itself.



Microsoft : Residual Net (ResNet)

- Enables deeper network
- Backpropagation is easier, and deals with Vanishing gradient problem
- Learns identity mapping



Network Examples