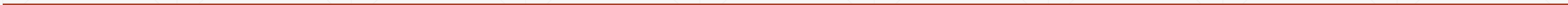


INFO251 – Applied Machine Learning

Lab 10
Suraj R. Nair

Topics

- Convolutional neural networks (CNNs)



Convolutional Layers

- Goal: Capture the spatial dependencies in parts of an image
- Multiply a **kernel matrix** (“**filter**”) k by subsets of the input image
 - **Hyperparameter**: Size of k (often 3x3, 5x5, or 7x7)
 - **Learn**: The weights of k
- **Stride**: How to shift the kernel matrix
 - **Hyperparameter**: Stride value (integer)

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

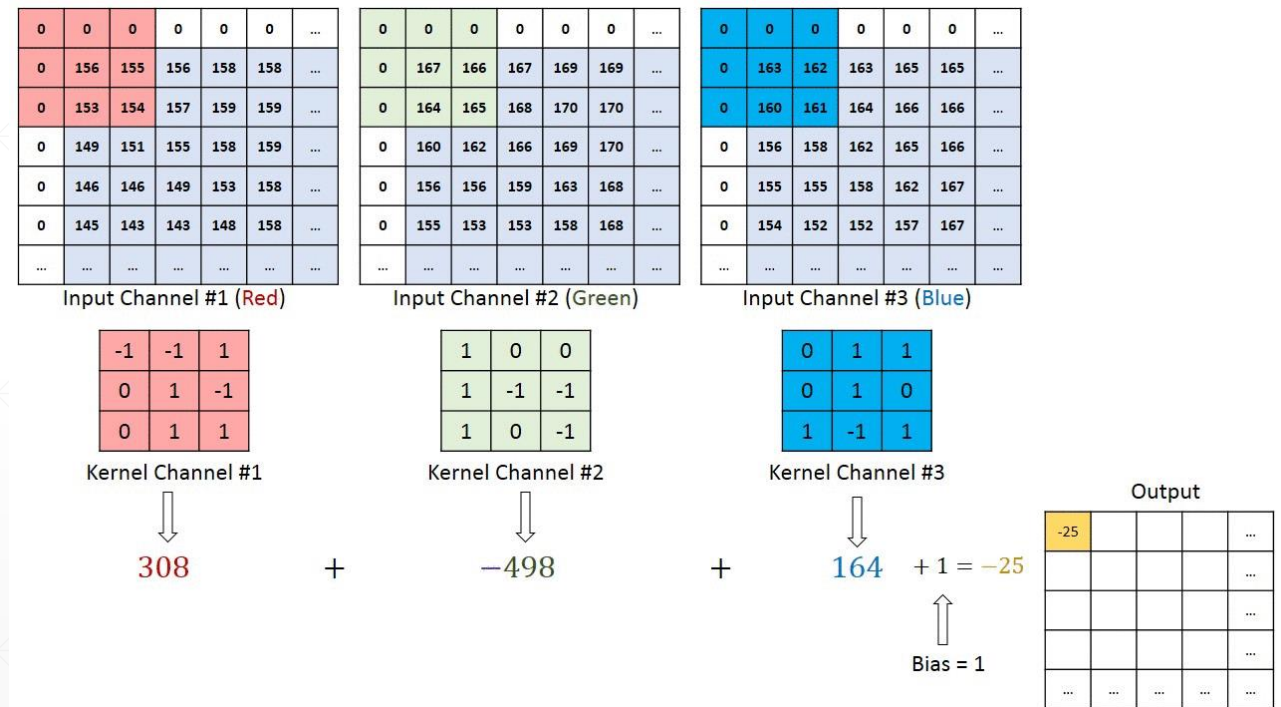
Image

4		

Convolved
Feature

Convolutional Layers

- **Channels:** Number of "layers" in the input image
 - Grayscale: 1 channel
 - RGB: 3 channels
 - RGBA: 4 channels
- Same filter size and stride length, but each channel has different weights
- Outputs of channels are summed up



Pooling Layers

- Goal: Reduce size of convolved layer to decrease compute cost
- Again, operates kernel matrix k over the convolved matrix
 - **Hyperparameter:** Size of k (usually 2x2)
 - **Hyperparameter:** Stride width (usually 2)
- **Max pooling:** Return maximum value in area covered by kernel
- **Average pooling:** Return average value in area covered by kernel

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

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

Convolutional Neural Network Structure

- Convolutional layer
- Pooling layer
- Convolutional layer
- Pooling layer

*Repeat convolution followed by pooling any number of times.
Option to add dropout after pooling.*

- Flatten
- Fully connected layer(s) (activation: sigmoid/tanh/relu)
- Output layer (activation: determined by problem type)

Add any number of fully connected layers. Option to add dropout.

Dimensions

- In general, output size from a convolutional layer: $[(N - F + 2P)/S] + 1$
 - N: Input image dimensions (e.g. **28 X 28**)
 - F: Filter / Kernel dimension(e.g. **3 X 3**)
 - P: Padding (default 0)
 - S: Stride (default 1)
 - Number of parameters associated with a convolutional layer: $((f * f * d) + 1) * k$
 - d is the number of filters/ channels in the previous layer
 - k is the number of filters in the current layer
-

Example: VGG-16 (Simonyan & Zisserman, 2015)

