Applied Machine Learning

Course number: W207

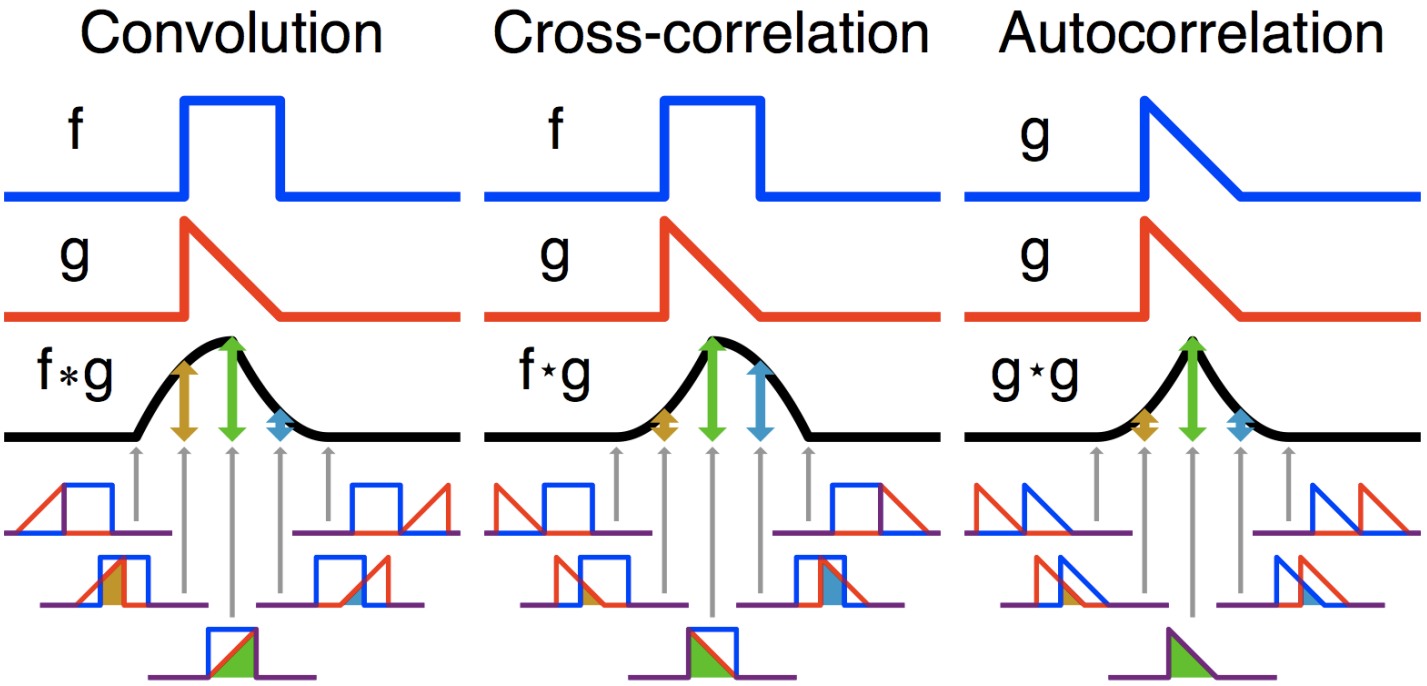
Prof. Alexander I. Iliev, Ph.D.

Applied Machine Learning

Lecture 13 …

- *Recommendation Systems*
- *PCA, SVD*
- *CNN*

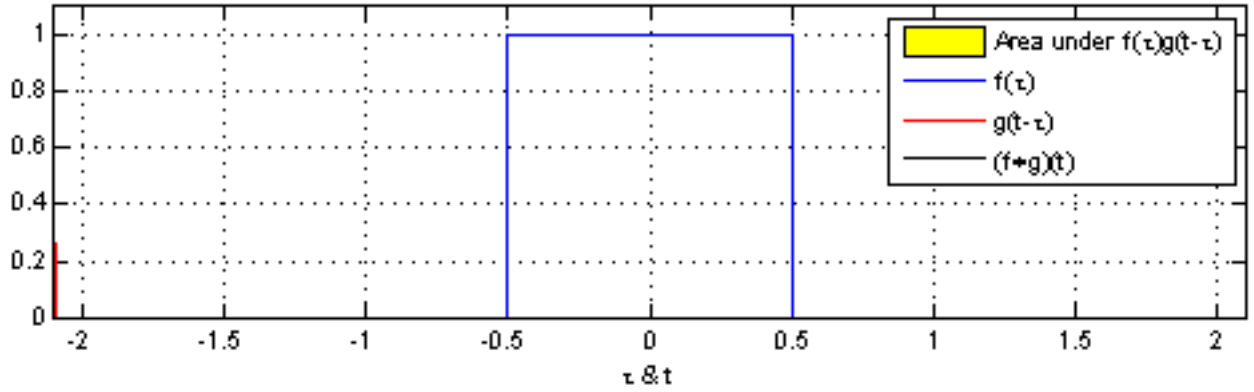
Signal Processing



Source Wikipedia

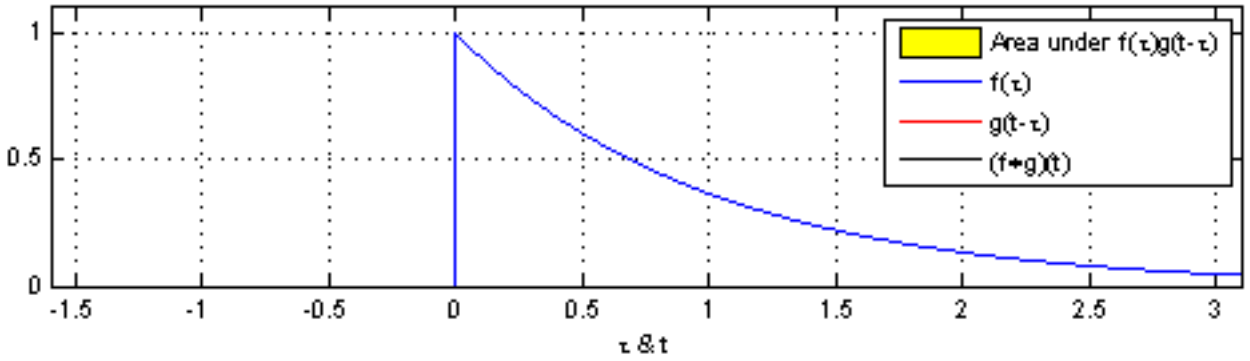
Signal Processing

- Convolving two signals:



box vs box

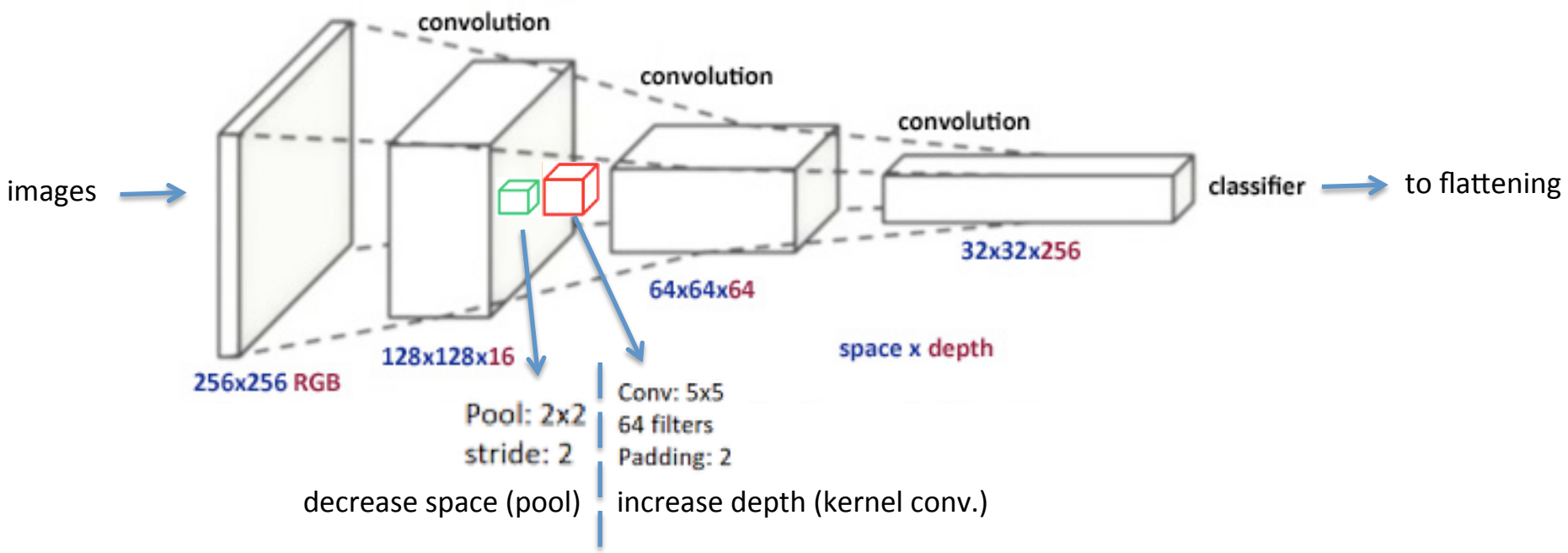
spiky vs box



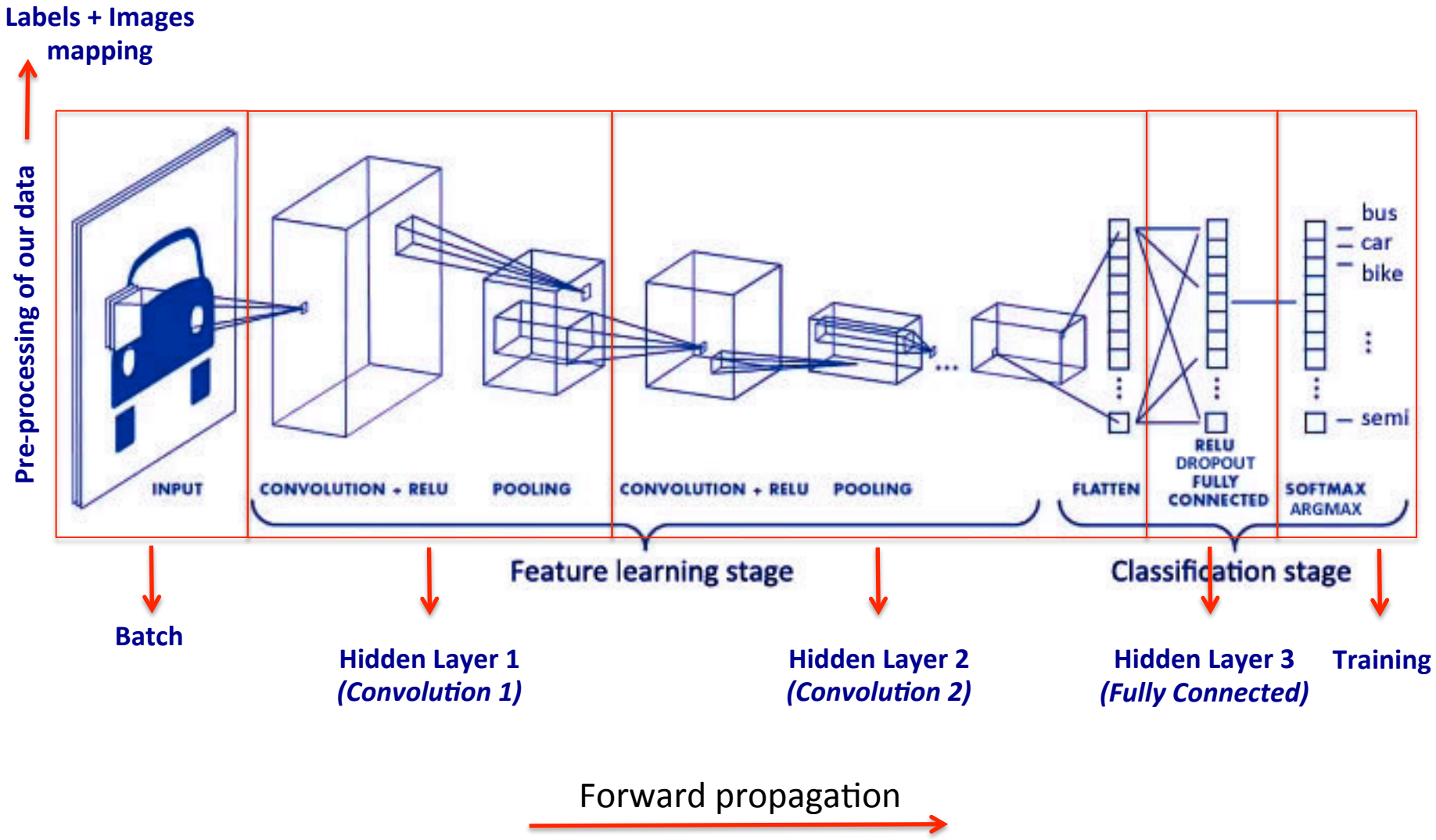
Source Wikipedia

Image recognition - CNN

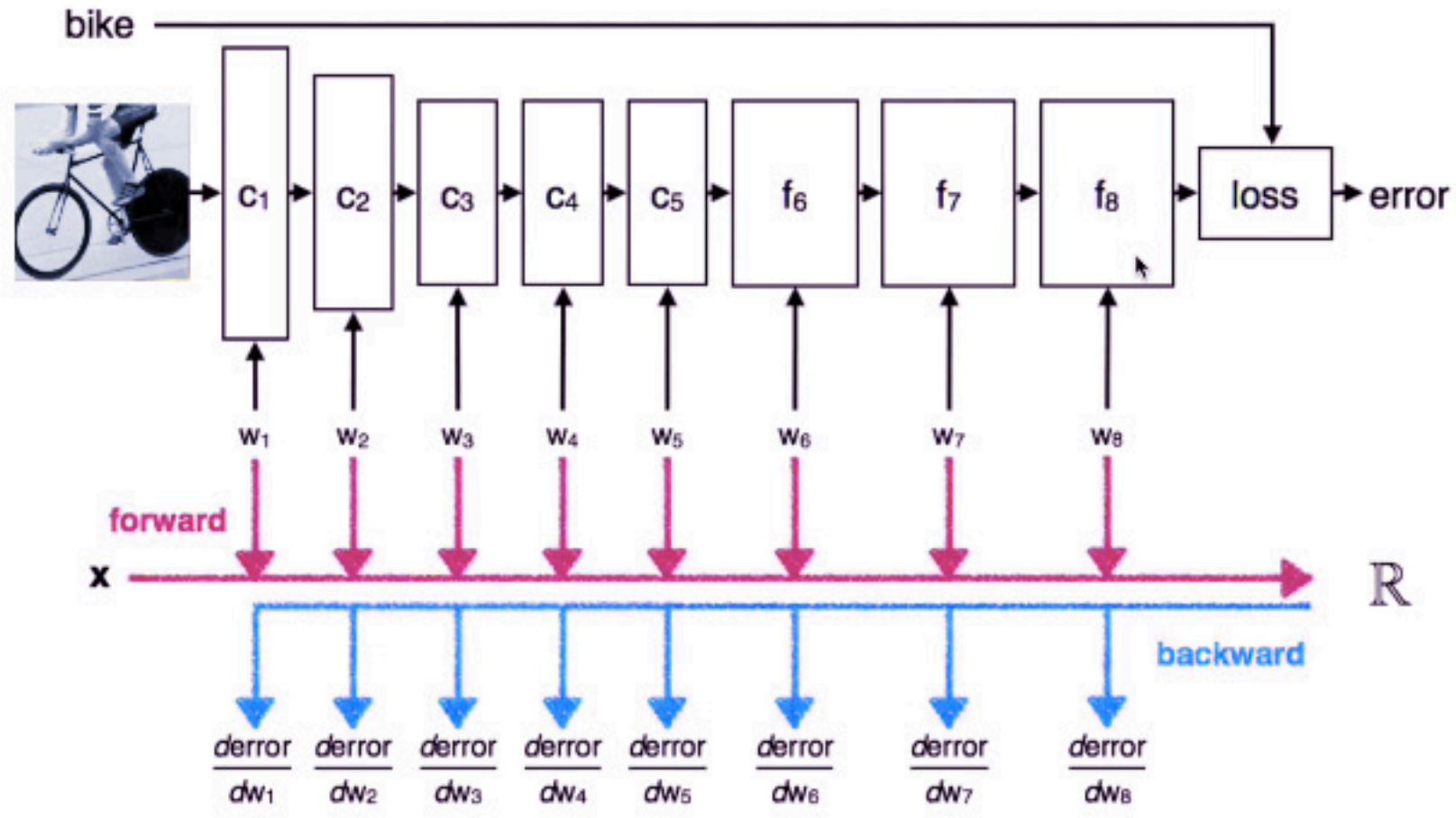
- In NN matrixes are multiplied (dot product) by other matrixes all the way through the network
- We start with bigger images (space) and after every layer they become more, but smaller (depth)



Model


Labels + Images mapping
Pre-processing of our data
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
RELU
DROPOUT
FULLY CONNECTED
SOFTMAX
ARGMAX
bus
car
bike
semi
Feature learning stage
Classification stage
Batch
Hidden Layer 1
(Convolution 1)
Hidden Layer 2
(Convolution 2)
Hidden Layer 3
(Fully Connected)
Training
Forward propagation

Model



Backward propagation

CNN clarifications

- These terms mean the same thing in CNN:

  weights = weight matrix = feature matrix = feature map = kernel = filter

- We can specify filters by taking specific parts of images for example or we can use pre-specified filters

- We then convolve these filters with other images and search for suitable overlap, where parts that are not interesting will = 0 and those that are interesting will > 0. Higher values are better !!!

- For that we use ReLU and is called activation. It is applied to every single pixel of the image.

CNN clarifications

- The neurons in each layer perform the same mathematical operation
- We feed it with the data from the previous layer in the network.
- Dropout provides random neuron dropout to prevent overfitting
- Fully connected layer is using softmax for probability (last layer)
- Softmax will give us a probability of our findings [0:1] based on the classes that are more likely to occur.
- We pick the maximum probability class using np.argmax()

CNN clarifications

- We train the network using Gradient Descent = Backpropagation
- The number of hidden layers and neurons is determined experimentally
- Flip a convolutional network and you get a De-convolutional network where you can create an image out of text
- Use 2D and 3D CNN any time we have spatial data (where the space and order matter), such as: images, sounds, text, etc.
- Remember: for regression we use the MSE or RMSE and for classification we use the softmax function

CNN clarifications

- How to improve the performance of a CNN?
  - Vary all hyperparameters such as:
    - Number of inputs (decrease large number of pixels to all images)
    - Number of layers
    - Type, frequency and % of pooling
    - Droput to prevent overfitting
  - Image pre-processing techniques:
    - Use uniform size and aspect ratio for all images
    - Use scaling, cropping and padding of images to make them of equal size
    - Use mean pixel values across all training examples (for each pixel)
    - Image normalization: subtract the mean from each pixel and then dividing the result by the STD
    - Convert images to grayscale (only if it makes sense)
    - Determine stride step for the kernel (filter) at each step
  - To increase performance, consider holding on pooling for several steps:

    conv2d > activation > conv2d > activation > max_pool

    This way you use more intermediate features initially to maximize the effect of training at the cost of computational efficiency