Convolution and Pooling as an Infinitely Strong Prior

Sargur Srihari
srihari@buffalo.edu

This is part of lecture slides on Deep Learning:
http://www.cedar.buffalo.edu/~srihari/CSE676

Topics in Convolutional Networks

- Overview
1. The Convolution Operation
2. Motivation
3. Pooling
4. Convolution and Pooling as an Infinitely Strong Prior
5. Variants of the Basic Convolution Function
6. Structured Outputs
7. Data Types
8. Efficient Convolution Algorithms
9. Random or Unsupervised Features
10. The Neuroscientific Basis for Convolutional Networks
11. Convolutional Networks and the History of Deep Learning

Topics in Infinitely Strong Prior

- Weak and Strong Priors
- Convolution as an infinitely strong prior
- Polling as an infinitely strong prior
- Underfitting with convolution and pooling
- Permutation invariance

Prior parameter distribution

- Role of a prior probability distribution over the parameters of a model is:
  - Encode our belief as to what models are reasonable before seeing the data

Weak and Strong Priors

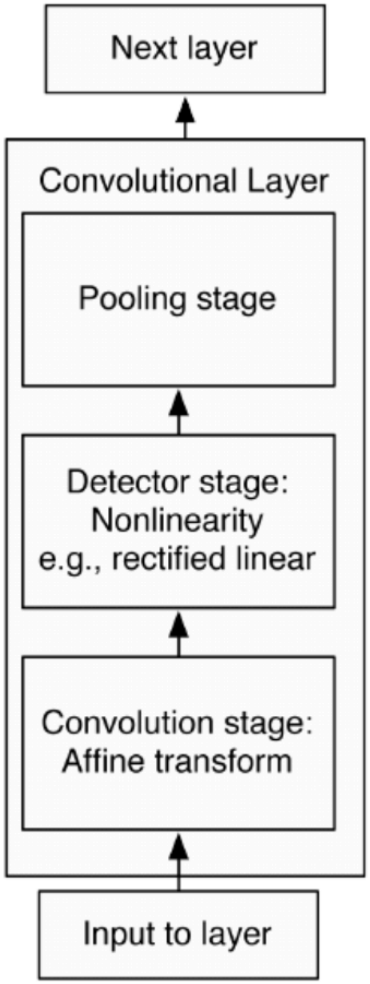
- A weak prior
  - It is a distribution that has high entropy
    - e.g., Gaussian with high variance
  - It allows data to move the parameters freely
- A strong prior
  - It has very low entropy
    - E.g., a Gaussian with low variance
  - Such a prior plays a more active role in determining where the parameters end up

Infinitely Strong Prior

- An infinitely strong prior places zero probability on some parameters
- It says that some parameter values are forbidden regardless of support from data

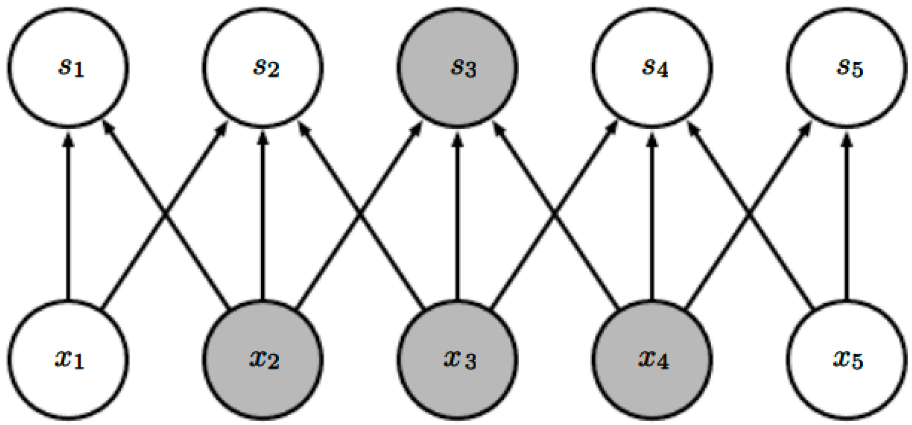
Convolutional Network

- Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers



Convolution as infinitely strong prior

- Convolutional net is similar to a fully connected net but with an infinitely strong prior over its weights
  - It says that the weights for one hidden unit must be identical to the weights of its neighbor, but shifted in space
  - Prior also says that the weights must be zero, except for in the small spatially contiguous receptive field assigned to that hidden unit

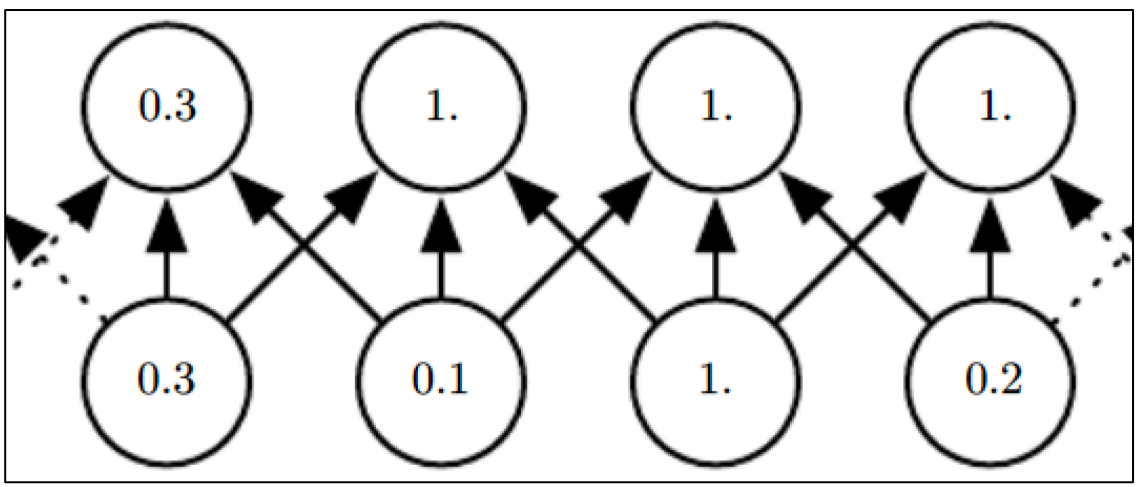


Convolution with a kernel of width 3
s_3 is a hidden unit. It has 3 weights which are the same as for s_4

  - Convolution introduces an infinitely strong prior probability distribution over the parameters of a layer
    - This prior says that the function the layer should learn contains only local interactions and is equivariant to translation

Pooling as an Infinitely strong prior

- The use of pooling is an infinitely strong prior that each unit should be invariant to small translations
- Maxpooling example:

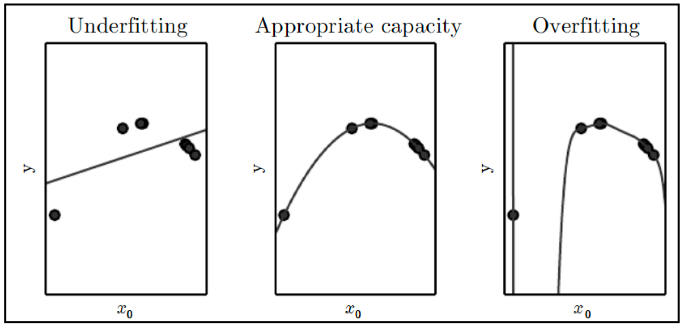


Implementing as a prior

- Implementing a convolutional net as a fully connected net with an infinitely strong prior would be extremely computationally wasteful
- But thinking of a convolutional net as a fully connected net with an infinitely strong prior can give us insights into how convolutional nets work

Key Insight: Underfitting

- Convolution and pooling can cause under-fitting
  - Under-fitting happens when model has high bias
- Convolution and pooling are only useful when the assumptions made by the prior are reasonably accurate
- Pooling may be inappropriate in some cases
  - If the task relies on preserving spatial information
    - Using pooling on all features can increase training error



High Bias/Underfit can be countered by:
1. Add hidden layers
2. Increase hidden units/layer
3. Decrease regular. parameter λ
4. Add features

When pooling may be inappropriate

- Some convolutional architectures are designed to use pooling on some channels but not on other channels
  - In order to get highly invariant features and features that will not under-fit when the translation invariance prior is incorrect
- When a task involves incorporating information from a distant location
  - In which case, prior imposed by convolution may be inappropriate

Comparing models with/without convolution

- Convolutional models have spatial relationships
- In benchmarks of statistical learning performance we should only compare convolutional models to other convolutional models – since they have knowledge of spatial relationships hard-coded
- Models without convolution will be able to learn even if we permuted all pixels in the image
- Permutation invariance: $f(x_1,x_2,x_3)=f(x_2,x_1,x_3)=f(x_3,x_1,x_2)$
- There are separate benchmarks for models that are permutation invariant