Machine Learning

Prof. Sergei Gleyzer

Lecture

PH451, PH551

Mar 19, 2025

Announcements

- **Hackathon # 2 due on Friday**
- **Midterm on Thursday, April 10**
- **Extra Credit Opportunity:**
  - **5 minute videos**

Final Projects

- **Semester long activity (40% of grade)**
- **Team activity**
  - **Self-designed project in consultation with instructor**
  - **Has to include an ML component**
  - **Can pick any topic you feel passionate about**
  - **Graduate students – can pick something from your specific field or research area**

Final Projects

- **Pre-proposal 5% due 03/25**
- **Proposal 10% due 04/03**
- **Project Outline and Demo 25% due 04/17**
- **Presentation 10%**
- **Peer Evaluation 10%**
- **Final Project Submission (write-up, code) 40% due 05/01**

Pre-Proposals

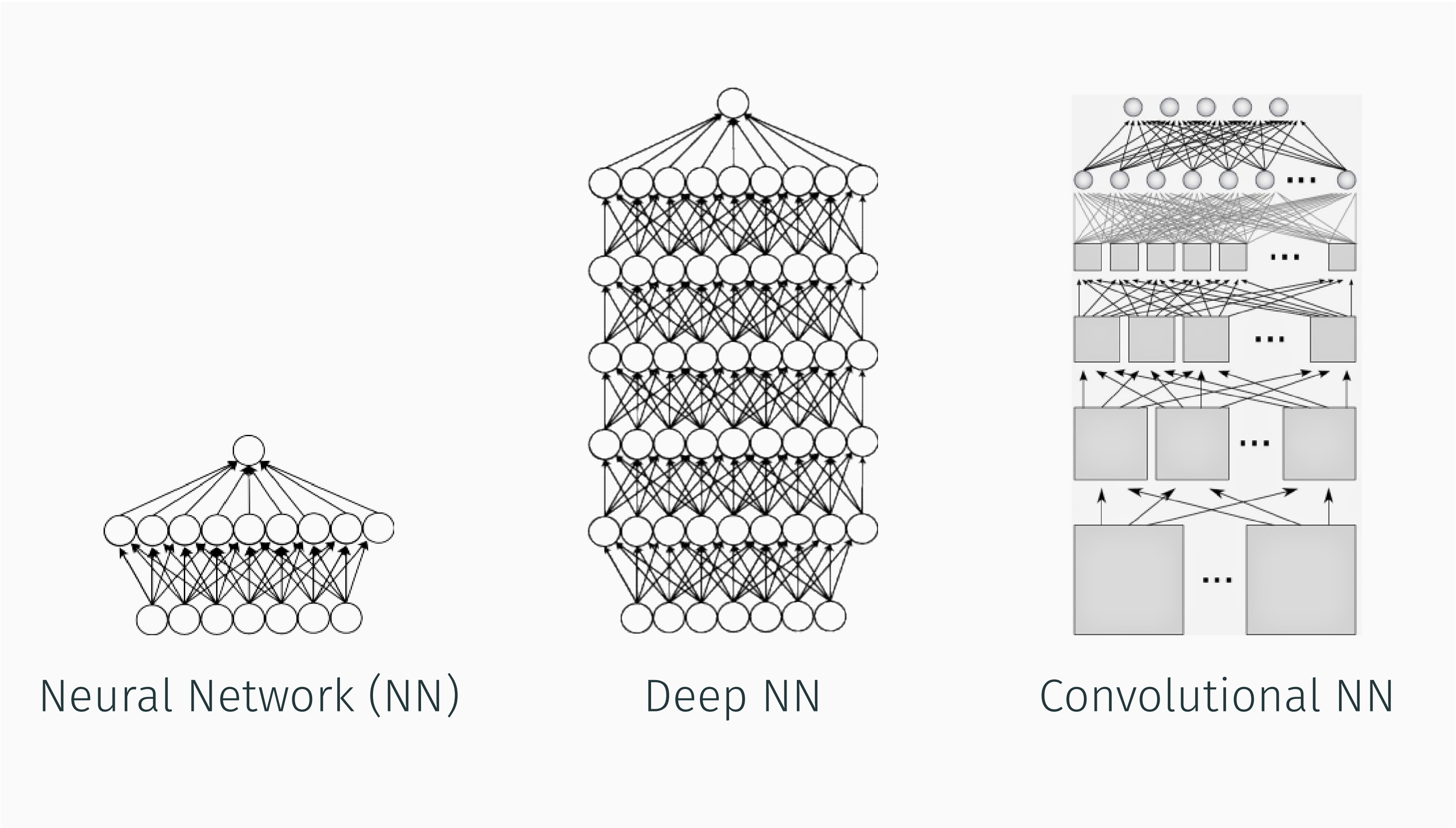
- **Should establish:**
  - **Your team**
  - **3 possible ideas for your project in your order of preference**
- **Pick potential problems that**
  - **You have an idea how to solve**
  - **Relevant to someone**
  - **Not yet solved (or provide a different solution)**

Hackathon #1 results

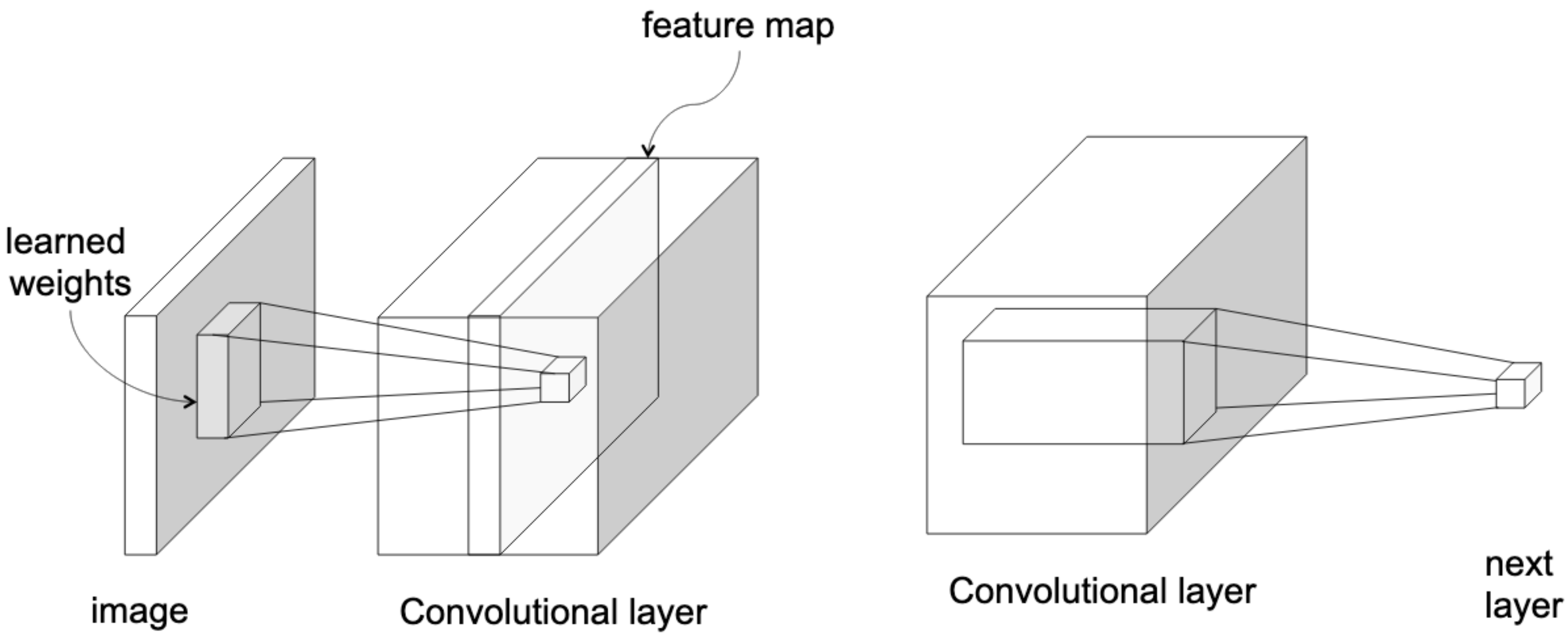
Group	AUC	Score
16	0.897	3
12	0.865	3
4	0.854	3
7	0.847	3
15	0.845	3
3	0.841	3
8	0.838	2.5
2	0.837	2.5
10	0.837	2.5
6	0.829	2.5
9	0.812	2.5
14	0.809	2.5
13	0.787	2
1	0.786	2
5	0.742	1.5
11	0.695	1.5

Convolutional Networks

Convolutional Networks

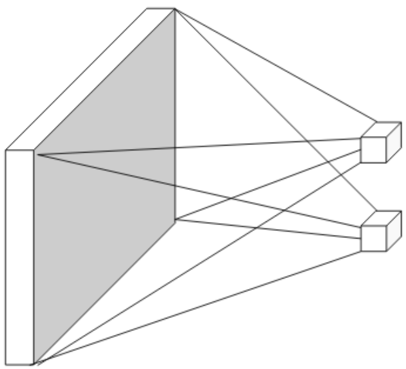


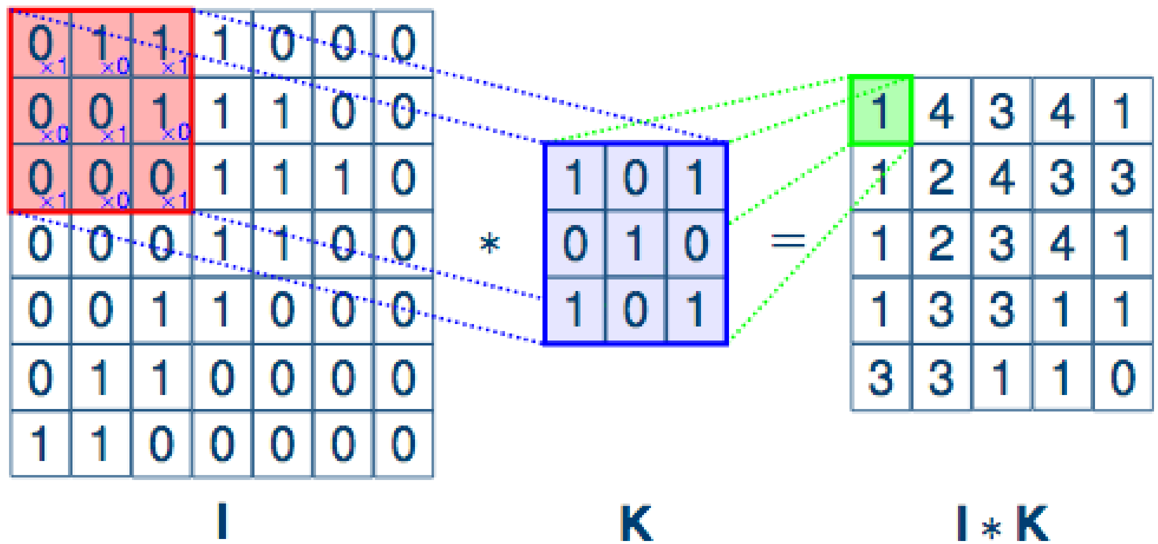
Convolutional NN



Input (Image) → Convolution → Activation (Non-Linear) → Spatial Pooling → Feature Maps

Convolution Example





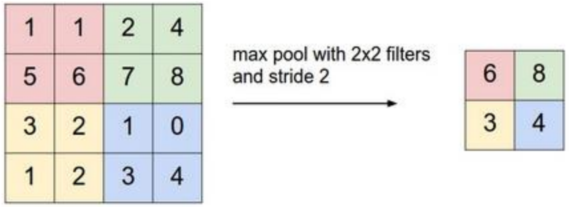
Same convolutional filter (set of weights) is applied to each element

- An element in a single 2-D location can only receive input from elements in similar location from previous layers (locality)
- Same weights for each feature map (and different across maps)
- Exploit structure, neighboring pixel dependence

Pooling

Down-sampling: shrink the size of the feature map

- Lower resolution that still contains important information

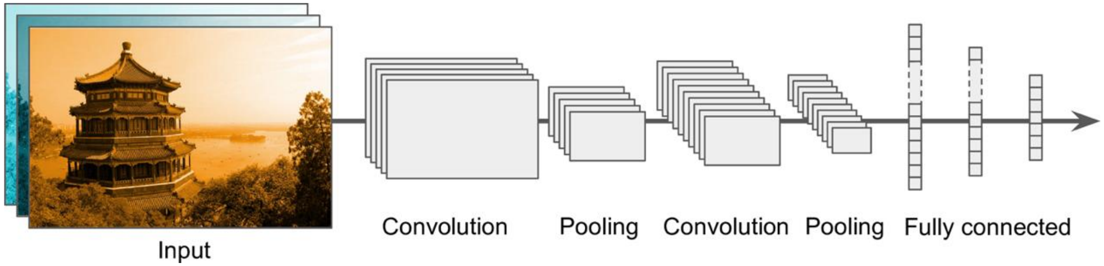


- Usually added after **convolution** and non-linearity (i.e. activation like ReLU) have been applied
- take the **average** (average-pooling) or **maximum activity** (max pooling) to represent the whole area
- Filter size is smaller than original feature map
- Helps model invariance to small local translations

Training

CNNs compute the stacked sequence of layers

- usually ending with a Fully-Connected Layer
- The FCN is the same as a regular neural networks

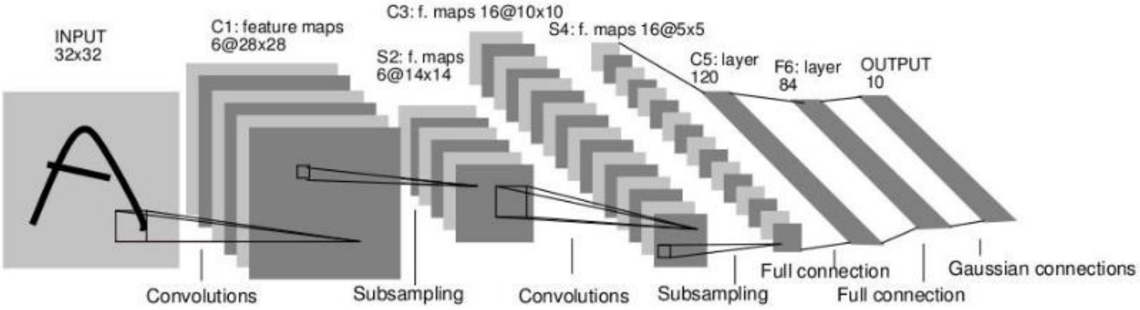


- Train with back-propagation

Some Well-known Architectures

- **LeNet5 (1990s)**
  - Early CNN used to read digits

LeNet 5

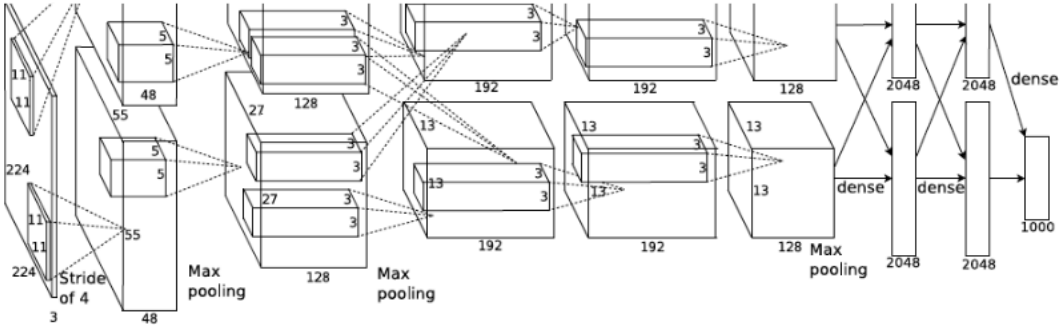


  - Y. LeCun et al., 1998
  - Average pooling, sigmoid, trained on MNIST

AlexNet

AlexNet (2012)

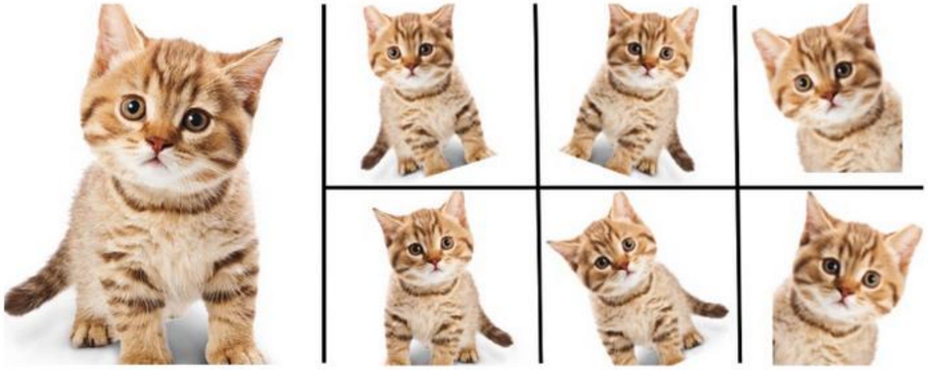
- Similar to LeNet but bigger and deeper model (8 layers, 60M params)



- ReLU activations, max pooling, dropout and data augmentation trained on GPUs on ImageNet
- Krizhevsky et al., 2012 (Imagenet 2012 winner)

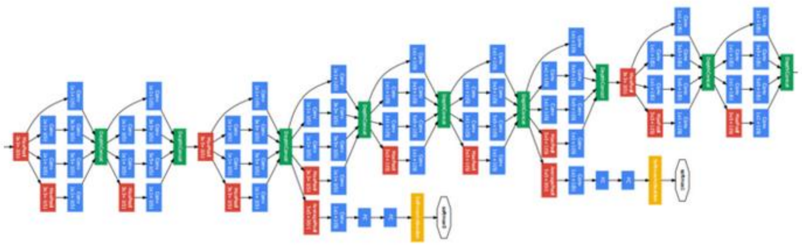
Data Augmentation

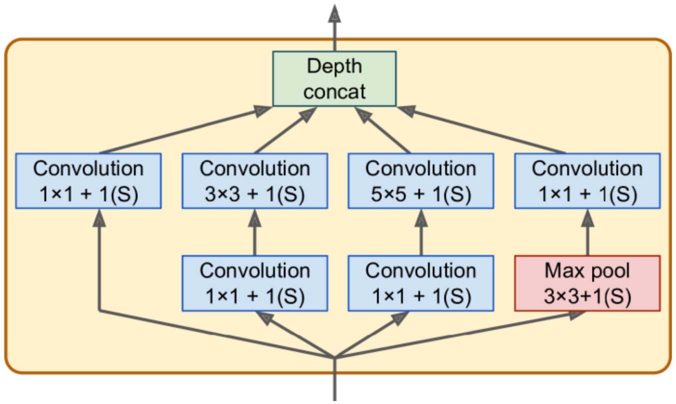
- **Useful technique for increasing training dataset size**
  - Apply rotations, shifts and re-sizing to make as many realistic training images as possible
  - Helps in training and to reduce overfitting



GoogleNet

- Szegedy et al., 2014
  - Much deeper than previous CNNs
  - **Inception modules**

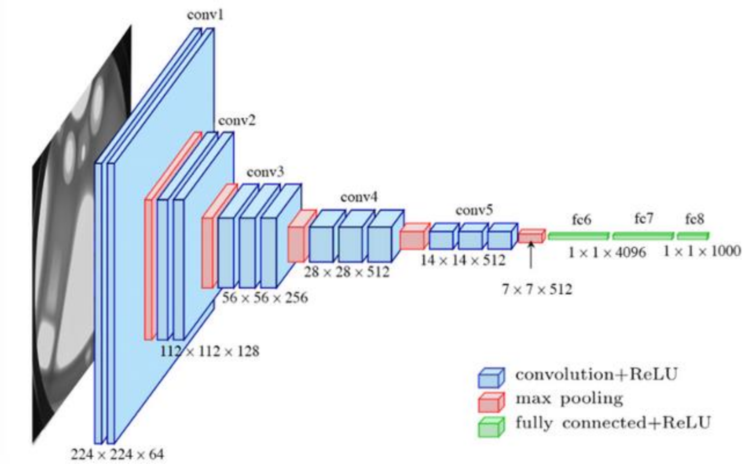




  - Multiple kernels stacked at same level
    - Concatenated along the depth dimension
    - Serve to capture information along the depth dimension across scales, bottlenecks to reduce dimensionality and behave like multi-dimensional layers

VGGNet

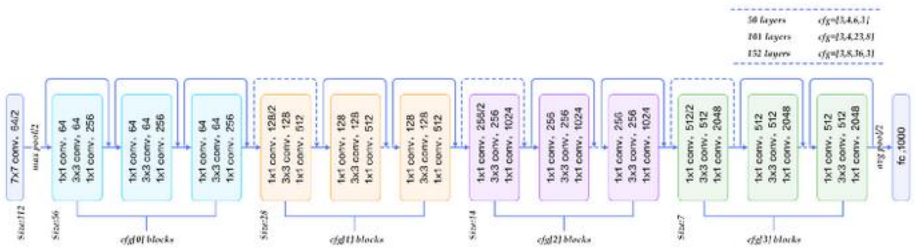
- Simonyan and Zisserman, 2014
  - Stacked smaller kernel-sized filters (3x3)

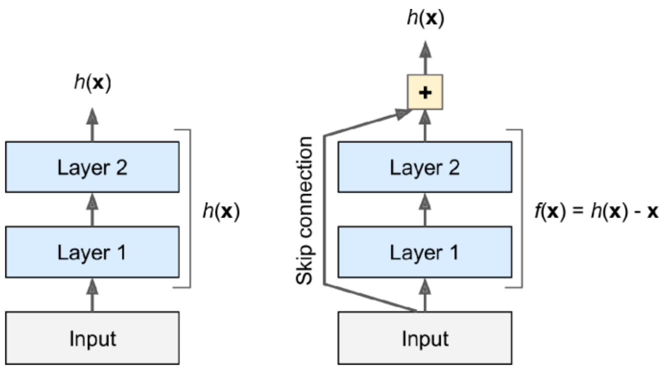


  - 16 layers: 2/3 convolutional, 1 max pool and repeat

ResNet

- He et al., 2015 (Imagenet 2015 winner)
  - Residual network with skip connections





  - 152 layers similar to VGG with skip connections (gated units) and batch normalization
  - Residual learning: h(x) – x
  - Helps propagate your signal across the whole network