Deep Learning Basics

Overview

Artificial Intelligence

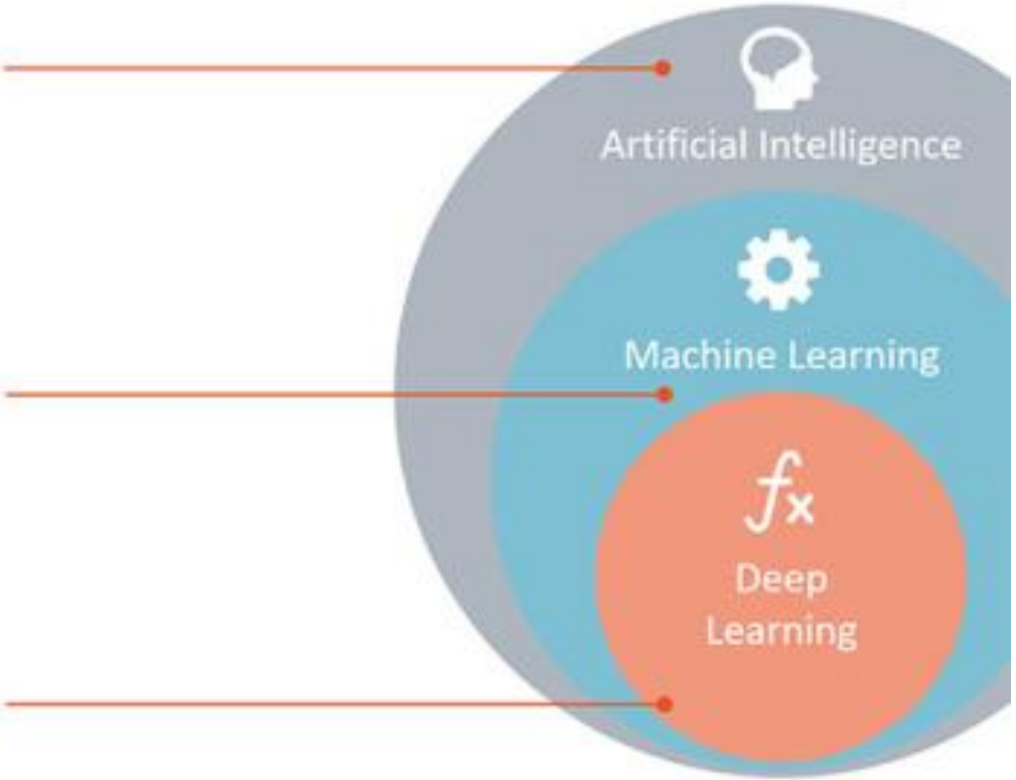
Any technique which enables computers to mimic human behavior.

Machine Learning

Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

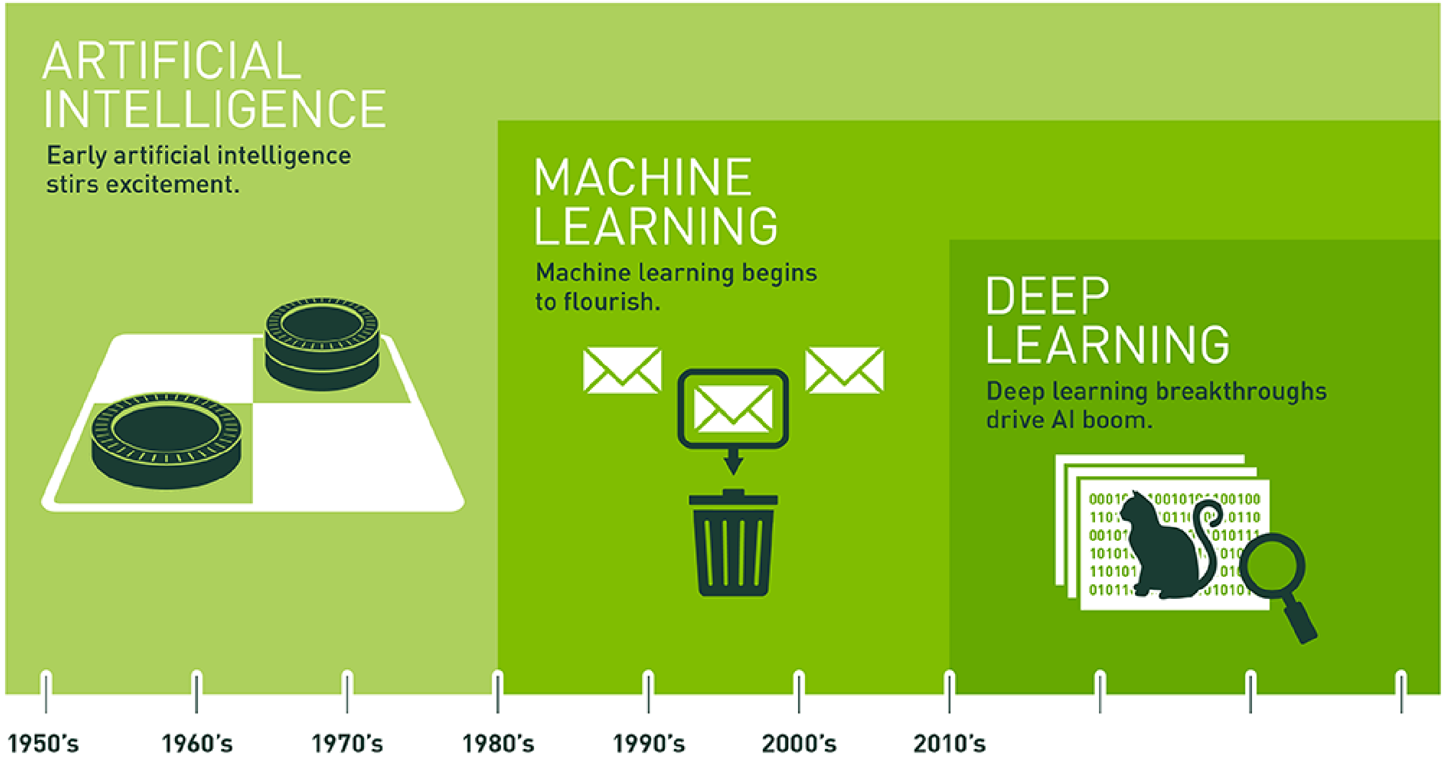
Deep Learning

Subset of ML which make the computation of multi-layer neural networks feasible.



Artificial Intelligence

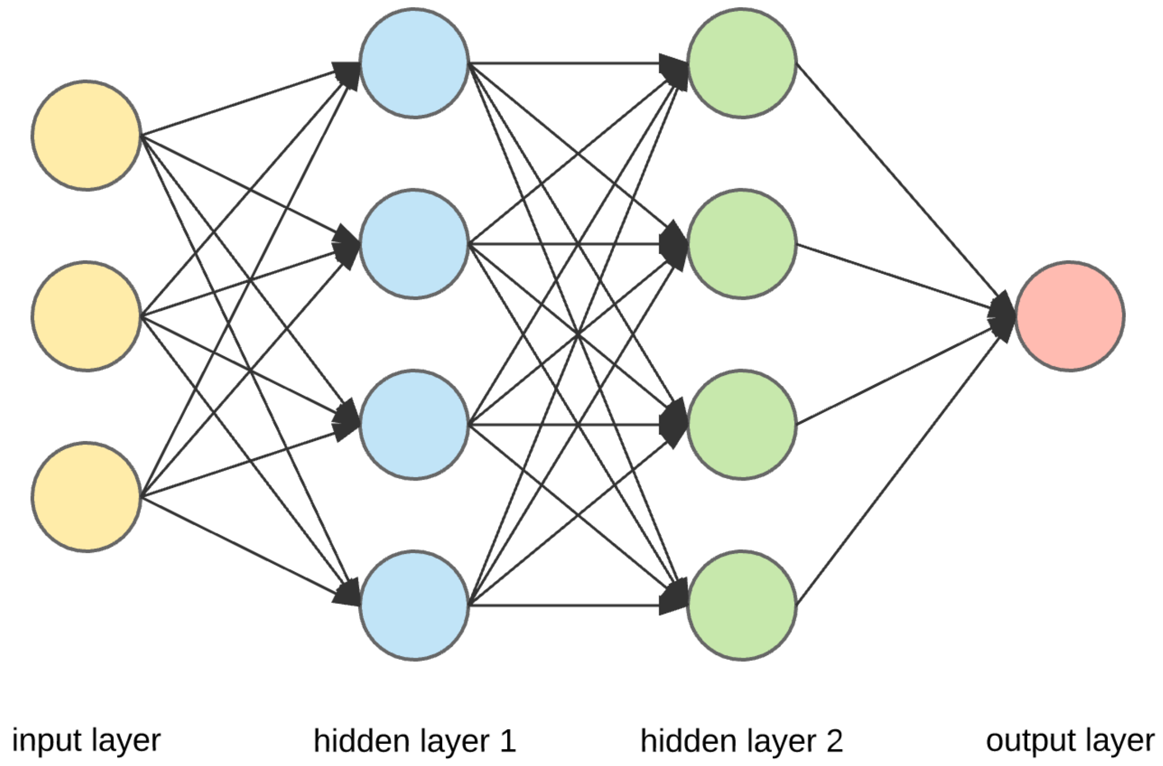
- Well-known example: computer beats chess world champion (DeepBlue, 1997)



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

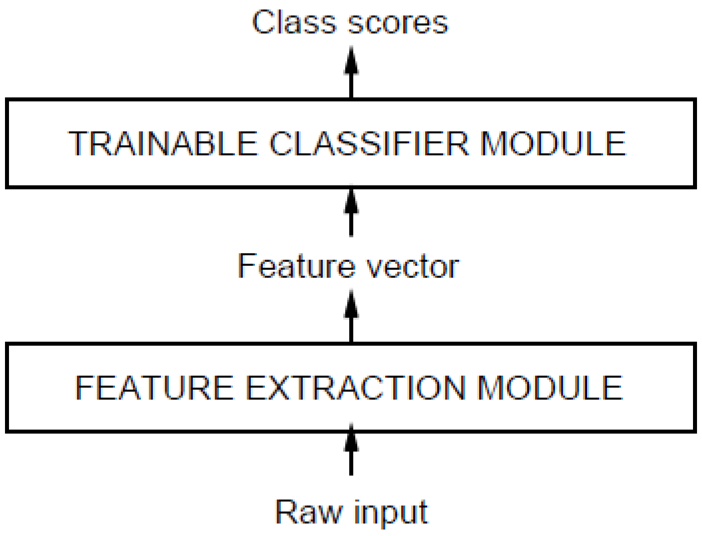
Machine learning

- Invention of neural networks
- Input: pattern represented by low-dimensional vectors or short strings of symbols



Traditional pattern recognition

- hand-crafted feature extraction
- combined with automatic learning techniques

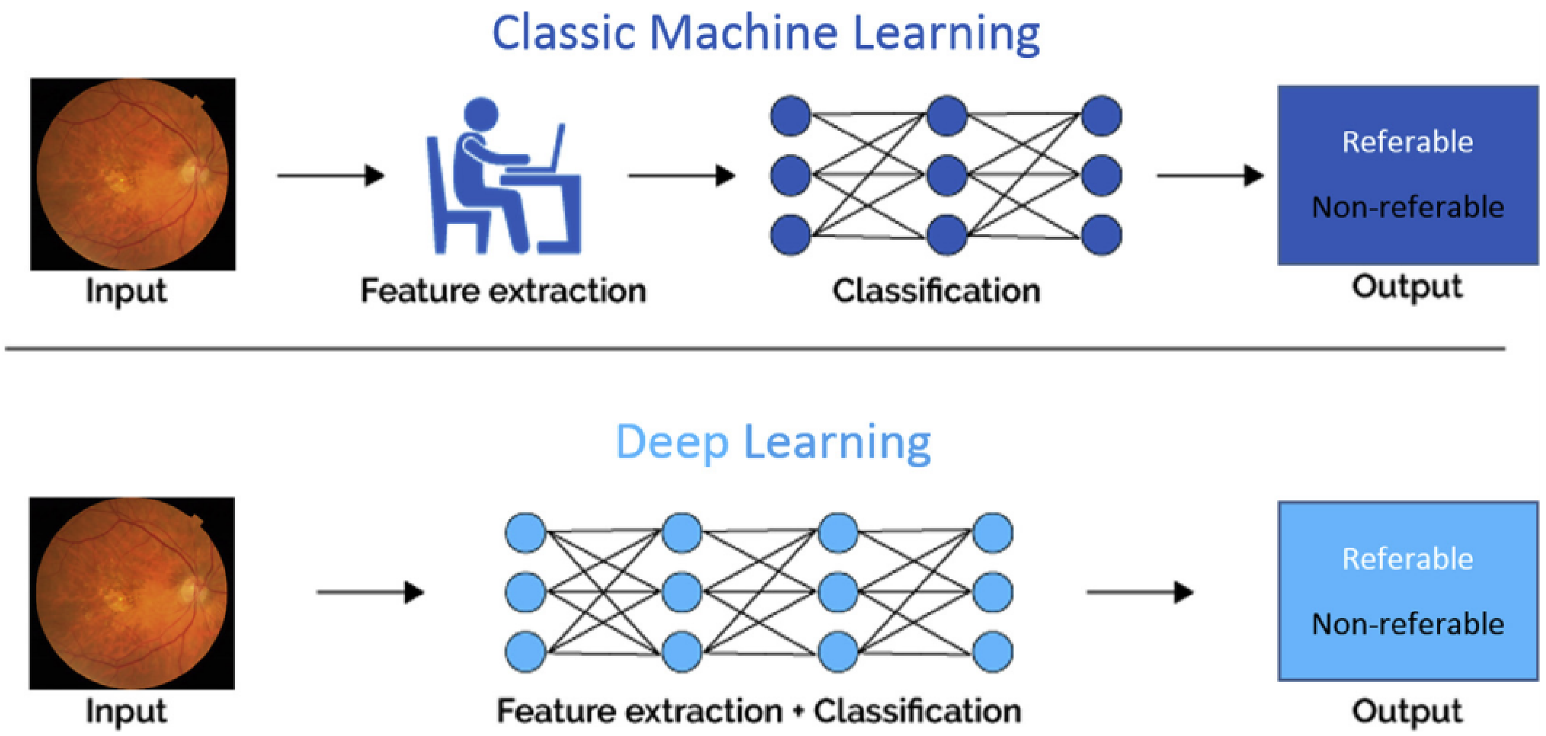


general-purpose

hand-crafted

Deep Learning

- No extraction of features
- Usually more hidden layers (bigger capacity)



Overview (cont.)

- Machine learning
  - Feed-forward neural networks
  - Activation functions
  - Backpropagation
- Convolutional neural networks
  - Fully connected layers
  - Convolutional layers
  - Sub-sampling layers
  - Architectures
    - LeNet5
    - AlexNet
  - Learning process
  - Overfitting
  - Examples (AlexNet)

Machine Learning

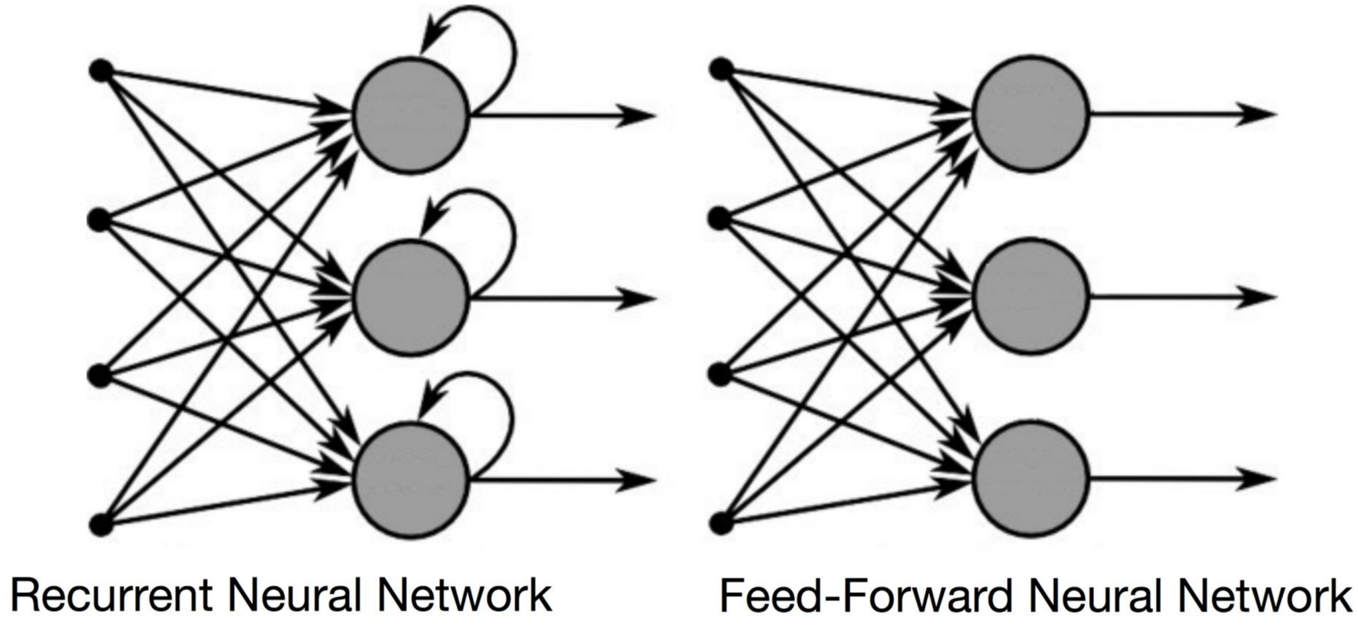
Neural Networks

Machine learning

- Feed-forward neural networks
- Activation functions
- Backpropagation

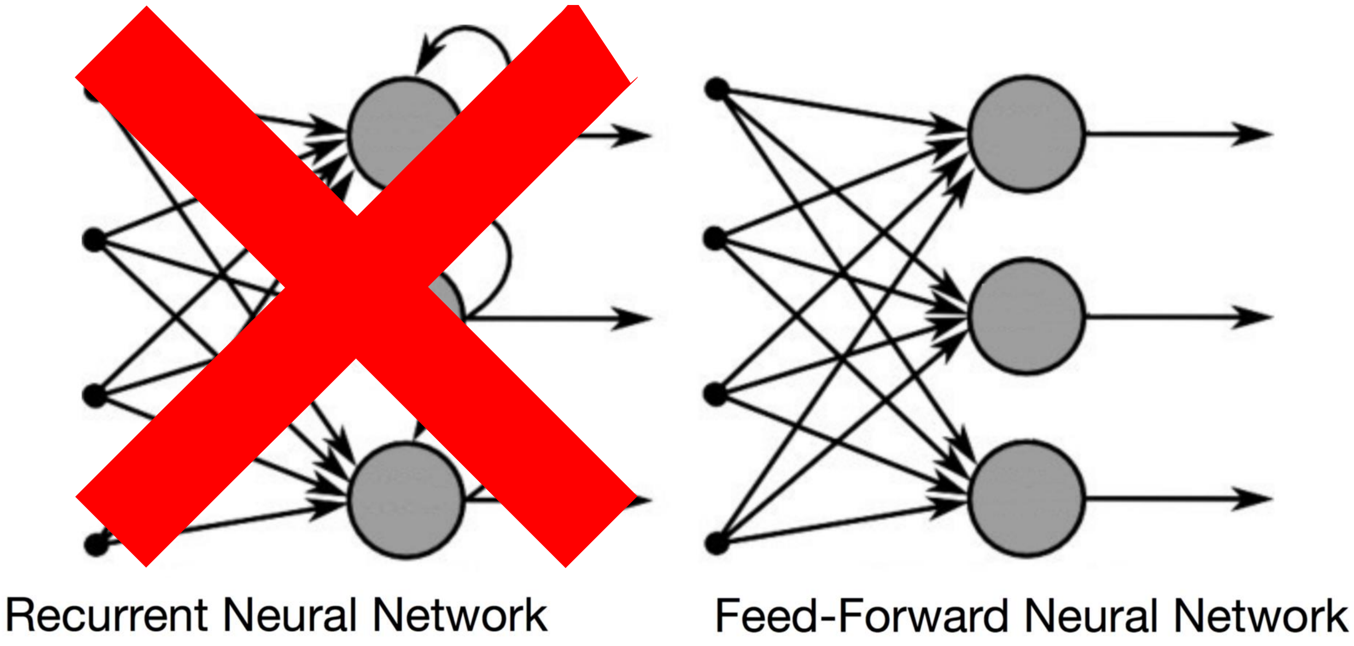
Feed-forward neural networks

- Neurons form an **acyclic** graph
- Opposite: Recurrent neural networks



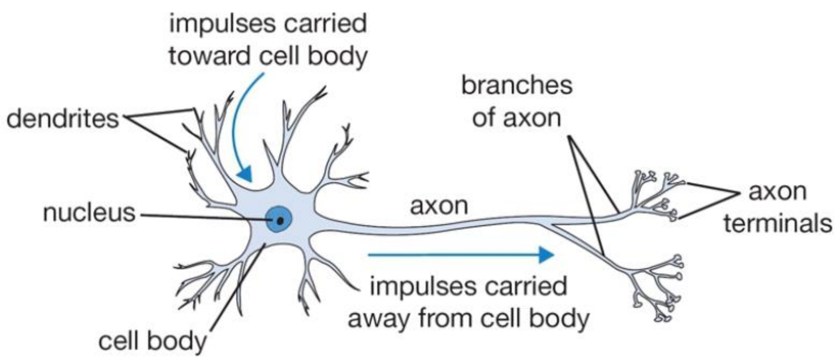
Feed-forward neural networks

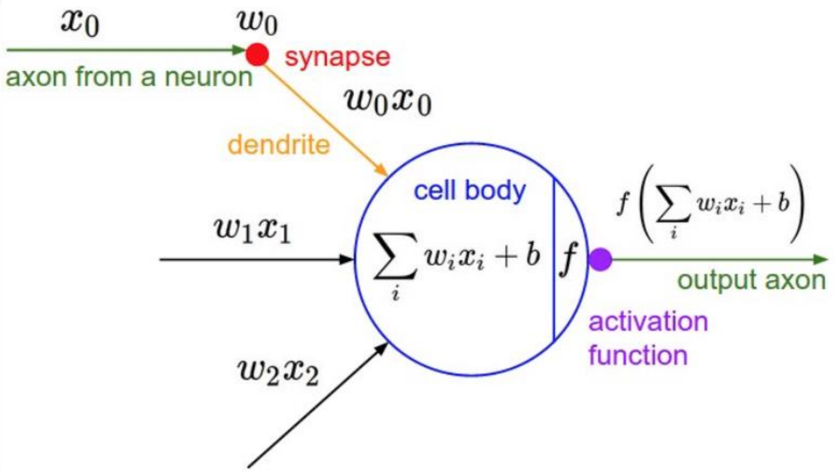
- Neurons form an **acyclic** graph
- Opposite: Recurrent neural networks



Activation function

- Function that computes the output of a neuron based on the inputs (neurons from previous layer * trainable weights + trainable bias)
- w_i : weights
- x_i : neurons from previous layer
- b : bias
- f: activation function





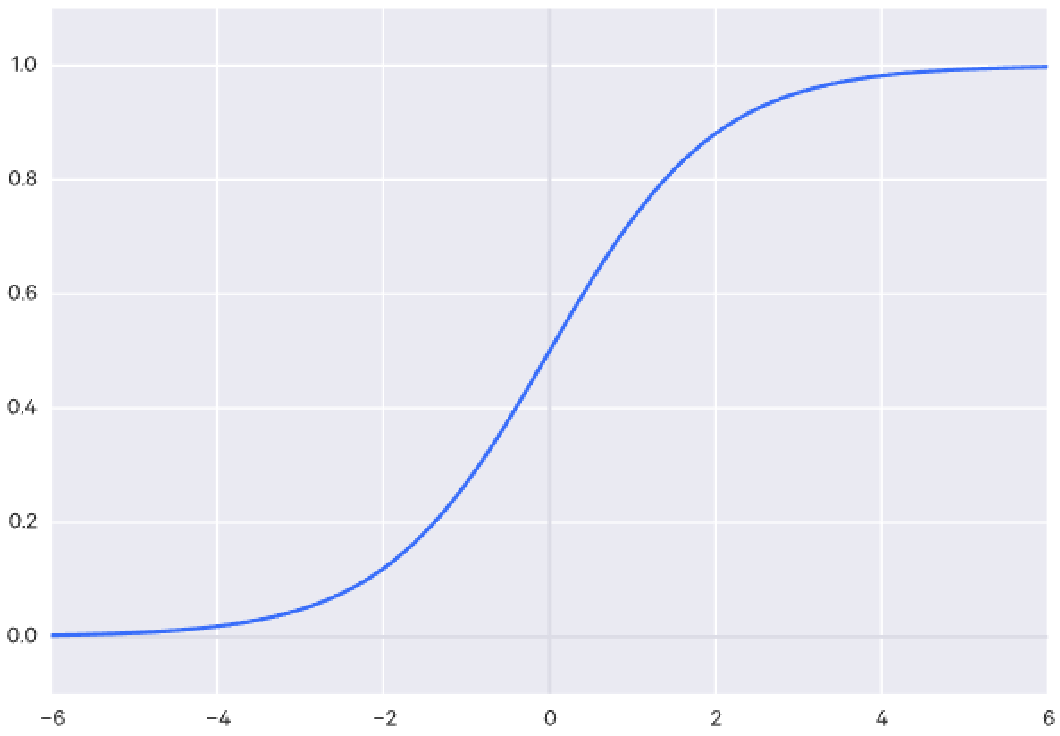
Common activation functions

- Logistic function (Sigmoid)
- Hyperbolic tangent (tanh)
- Rectified Linear Unit (ReLU)

Logistic function (Sigmoid)

- S-shaped function
- Range [0;1]

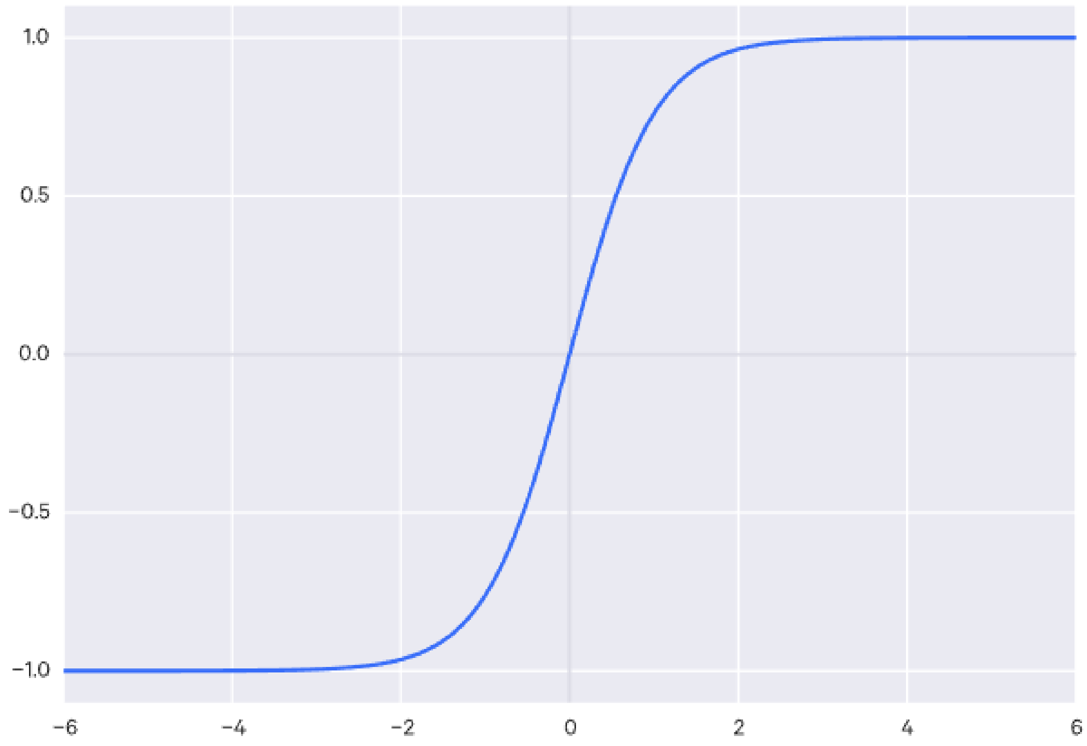
$$f(x) = \frac{1}{1+e^{-x}}$$



Hyperbolic tangent (tanh)

- Reshaped Sigmoid function
- Range [-1;1]

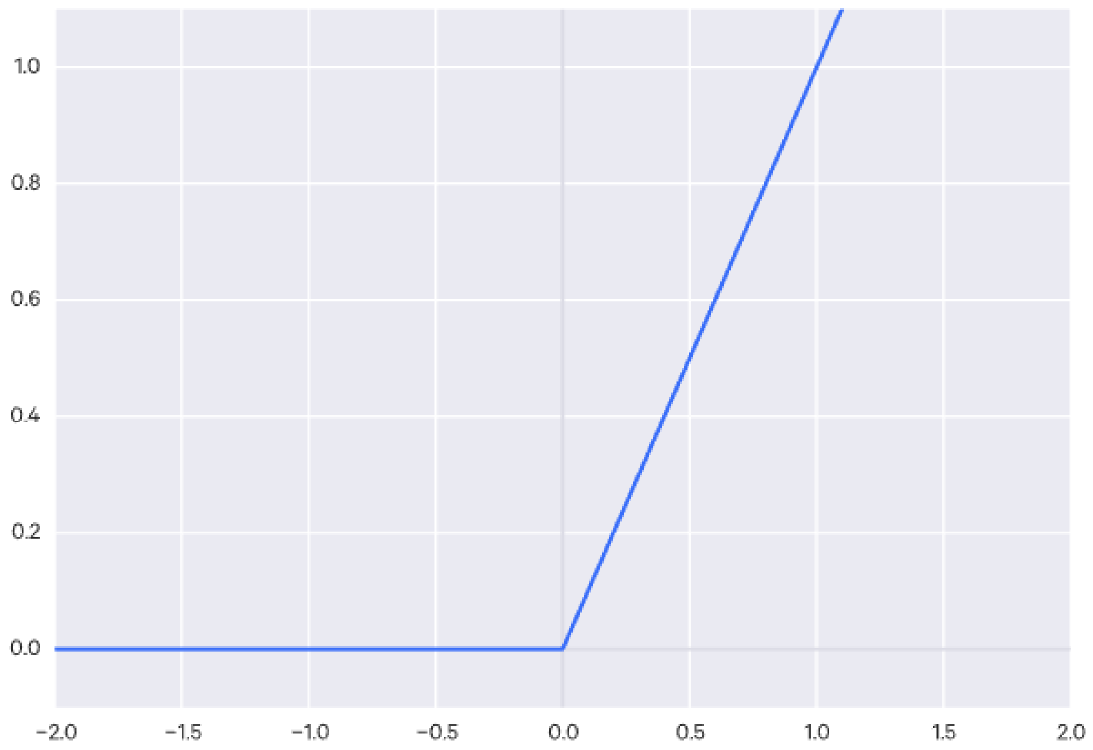
$$f(x) = \tanh(x)$$



Rectified Linear Unit (ReLU)

- Range [0;∞)
- Very efficient for large neural networks

$$f(x) = \max(0,x)$$



Backpropagation

- Algorithm to train a neural net
- Supervised learning (a target output vector must exist)
- Idea: Calculate the difference between the actual out and the target output and try to minimize it by adjusting the weights of the neurons

Backpropagation algorithm

1. Initialize weight vector (w_0) with random values
2. Calculate the output vector (y_i) by using the activation function:

$$y_i = f(w_i, x_i)$$

1. Computer the difference between the actual output vector (y_i) and the target output vector (y'_i) by using the error/ loss/ cost function, usually squared Euclidian distance:

$$E(y_i, y'_i) = \frac{1}{2} ||y_i - y'_i||^2$$

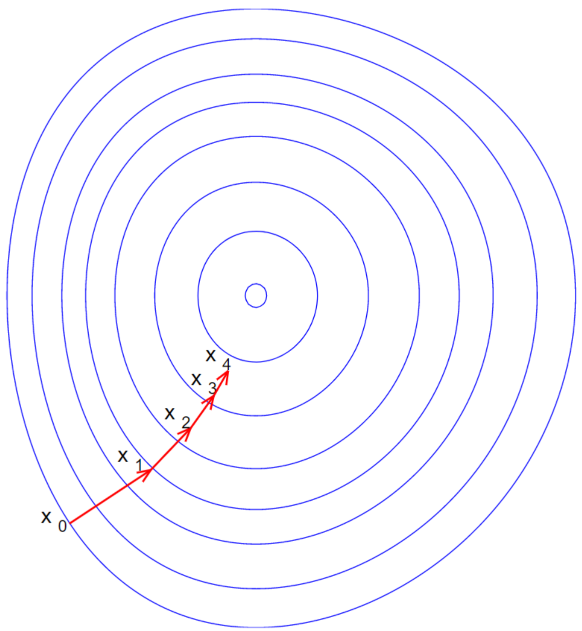
2. Update the weights and biases by applying gradient descent

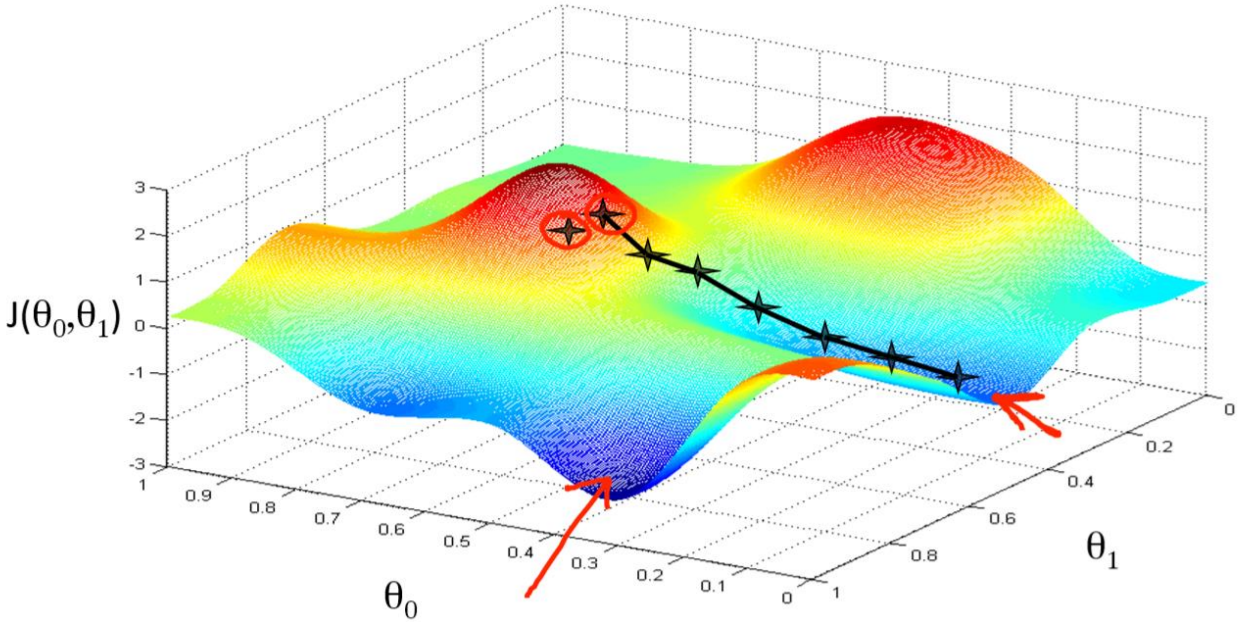
$$w_i = w_{i-1} - \eta \nabla E(y_i, y'_i)$$

3. Repeat steps 2-4, starting with $i = 1$

Gradient descent

- Algorithm for finding the minimum of a function by “following” the negative gradient of the current point.
- Analogy: Finding the valley by following the steepest descent
- Problem: Possible to get “stuck” in local minima





Convolutional Neural Networks (CNN)

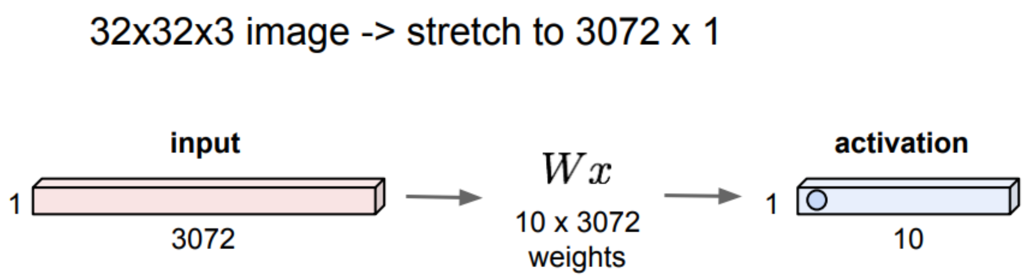
Deep Learning in Image Recognition

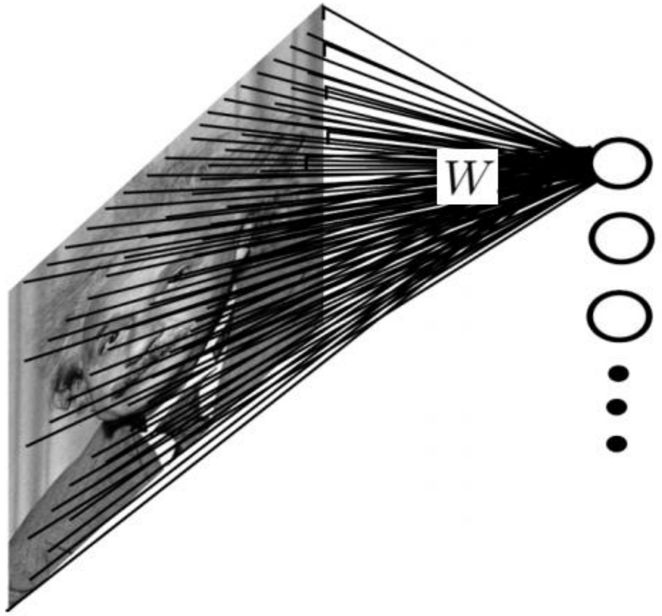
Convolutional neural networks

- Fully connected layers
- Convolutional layers
- Sub-sampling layers
- Architectures
  - LeNet5
  - AlexNet
- Learning process
- Overfitting
- Examples (AlexNet)

Fully connected layers

- Every neuron is connected to every neuron from the previous layer.





Problems of fully connected layers

- Image size
- Ignoring topology

Image size

- 1000x1000 image -> 1 million input units for each neuron
- Increased capacity requires a larger training set
- Memory requirement -> hardware limitation
- No invariance with respect to translation/ local distortions (i.e. the neural net has to learn how to recognize distinctive features at very possible location)

Ignoring topology

- Images have a strong local structure
- Pixels that are spatially nearby are highly correlated
- Extract and combine local features before recognizing objects

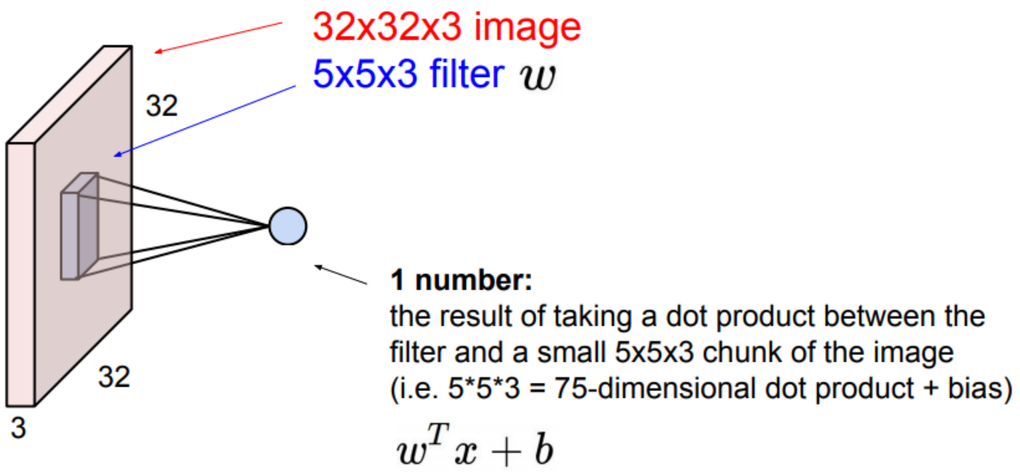
Problems of fully connected layers

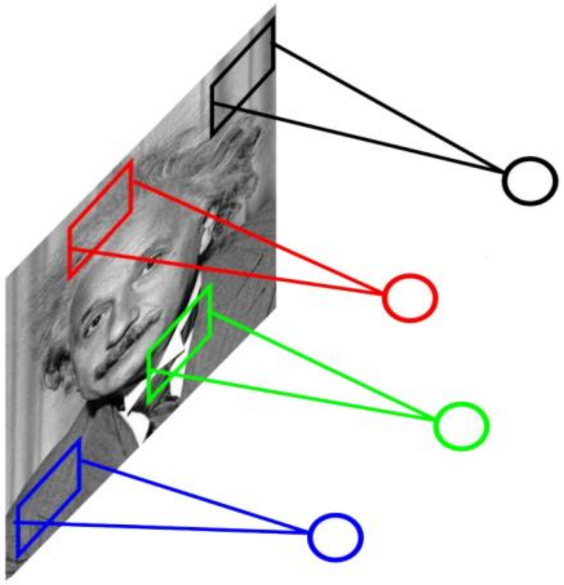
- Image size
- Ignoring topology

-> Convolutional layers

Convolutional layers

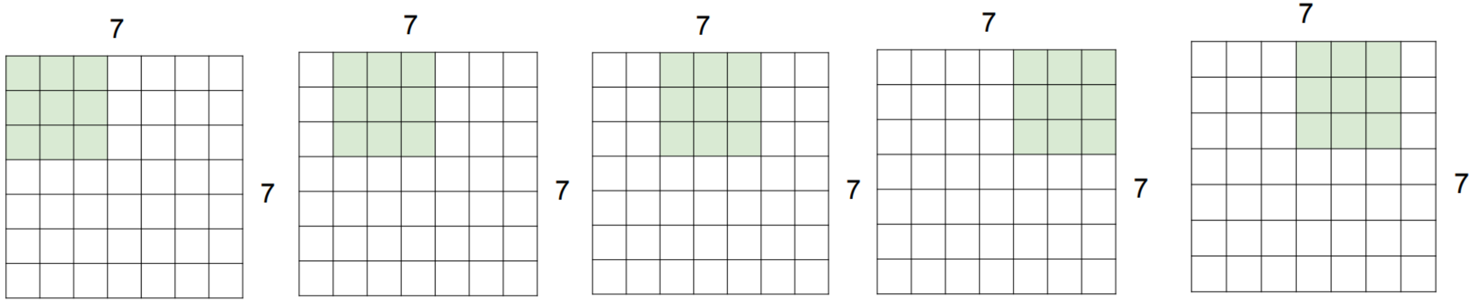
- Each neuron receives input only from a set of units located in a small neighborhood (**receptive field**) in the previous layer
- The **same** filter/ kernel is applied to every location (**shared weights**)



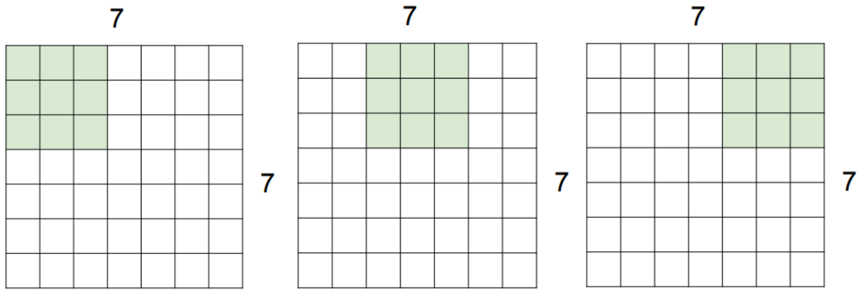


Convolutional layers (cont.)

- A 3x3 filter applied to a 7x7 input (**5x5 output**)

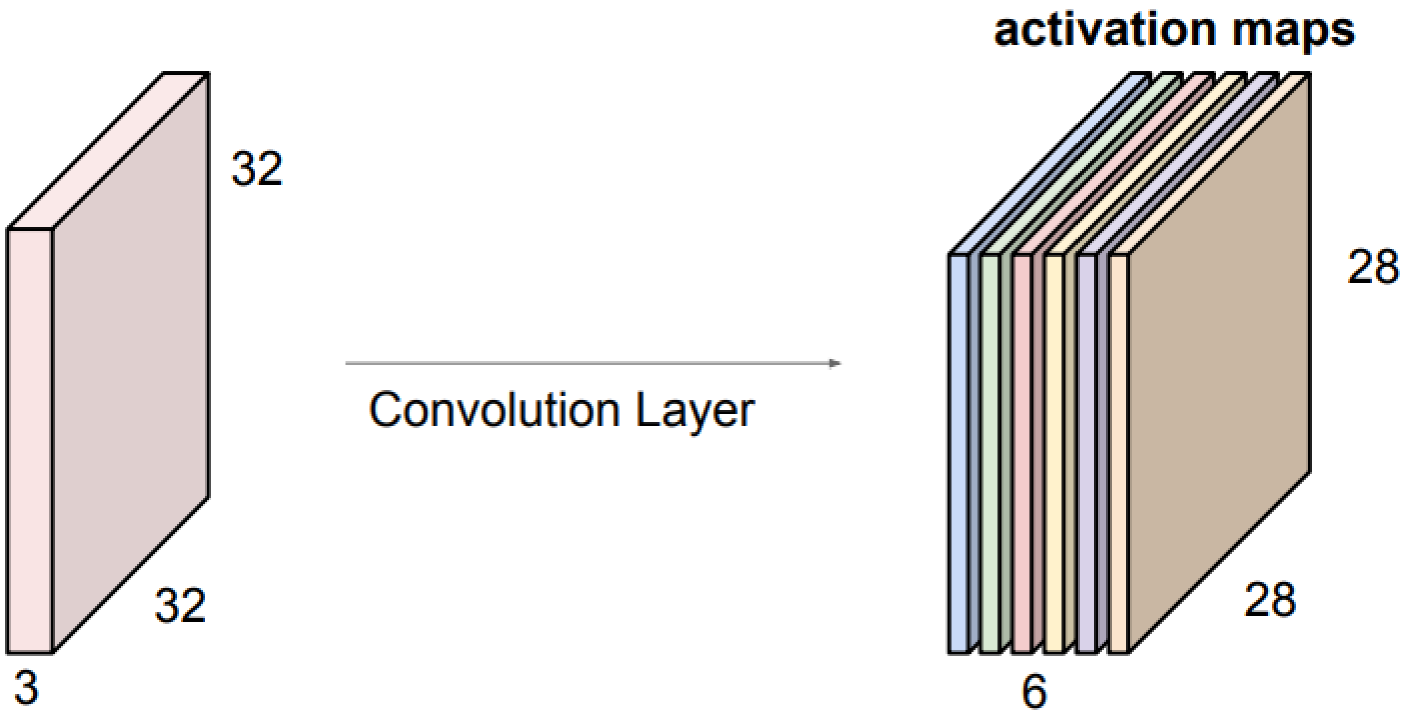


- A 3x3 filter applied to a 7x7 input with **stride 2** (**3x3 output**)



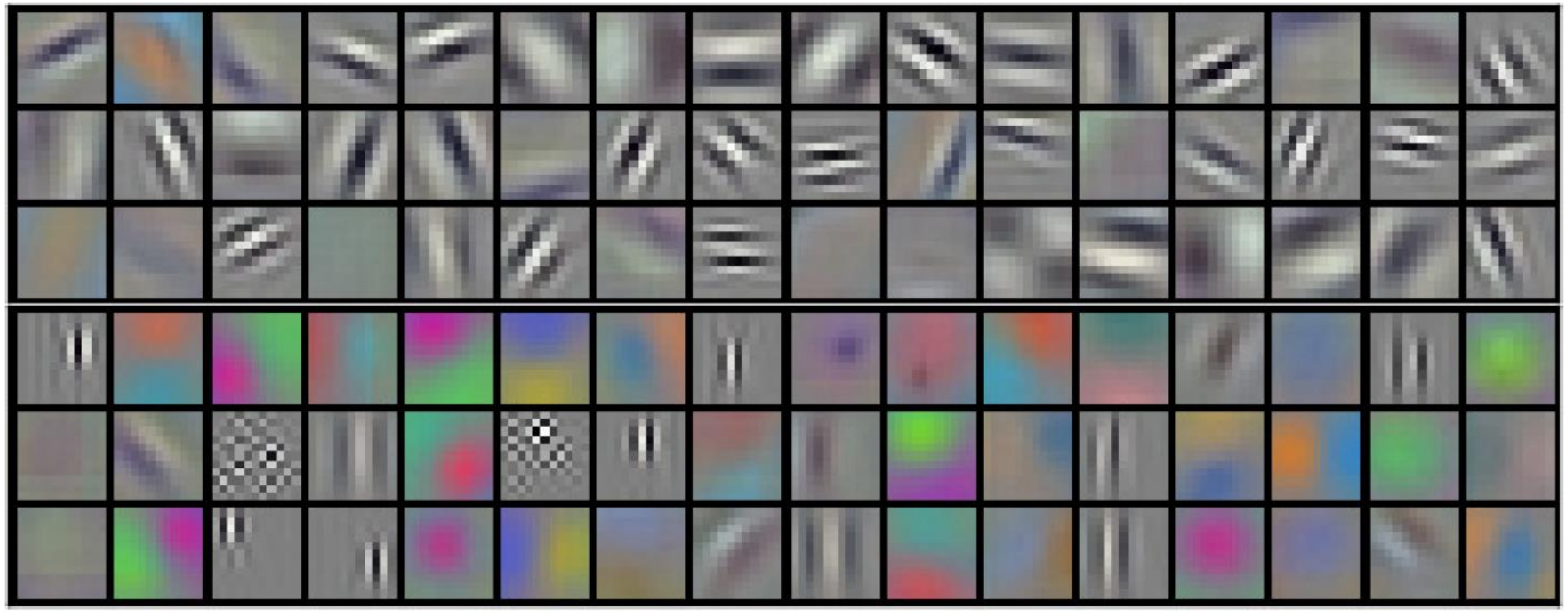
Convolutional layers (cont.)

- Convolutional layers are composed of several feature/activation maps, so that multiple features can be extracted at each location



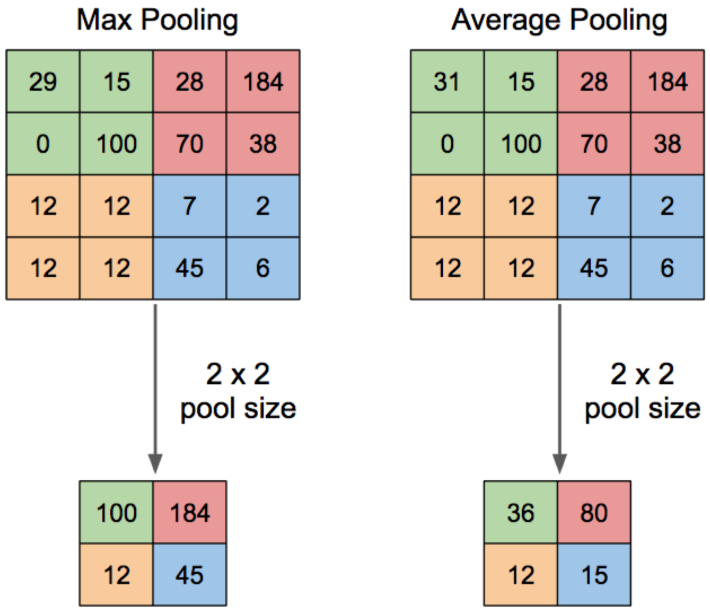
Convolutional layers (cont.)

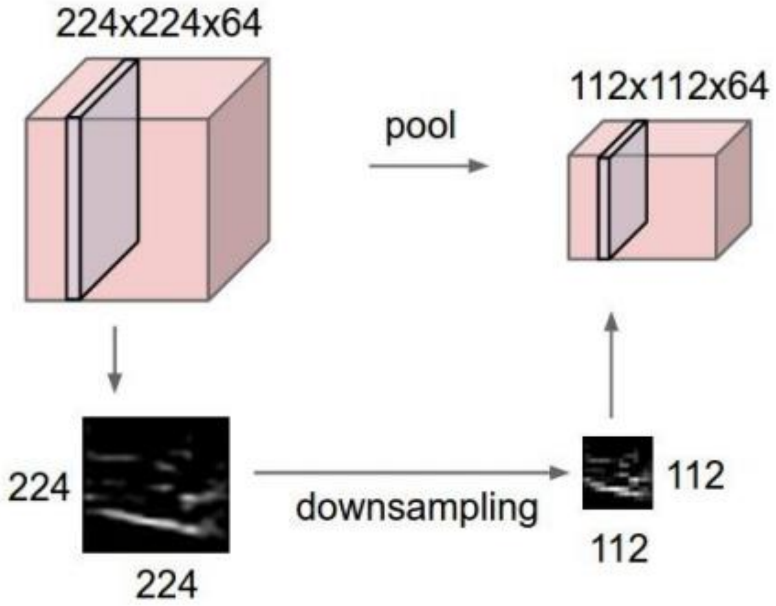
- **Example**: 96 convolutional kernels of size 11x11x3 learned by the first convolutional layer on the 224x224x3 input images.



Sub-sampling layers

- Perform local averaging
- Reduce the resolution of the feature map
- Reduce the sensitivity of the output to shifts/ distortions
- Pooling (+ multiply with coefficient + add bias + activation function)



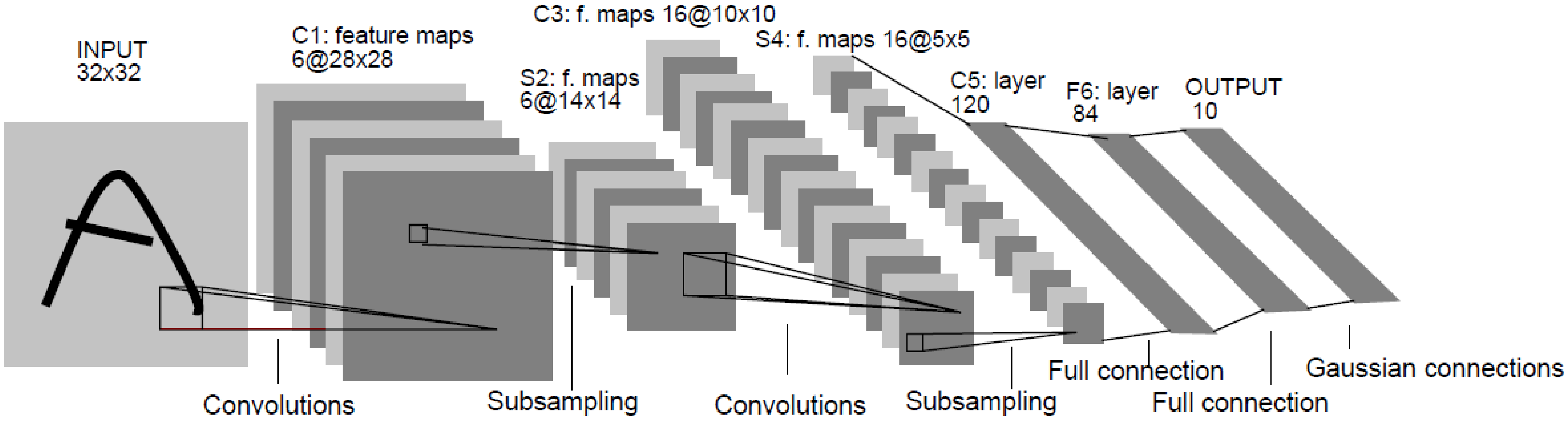


Architecture

- Yann LeCun (LeNet-5), 1998 (!)
- Alex Krizhevsky (AlexNet), 2012

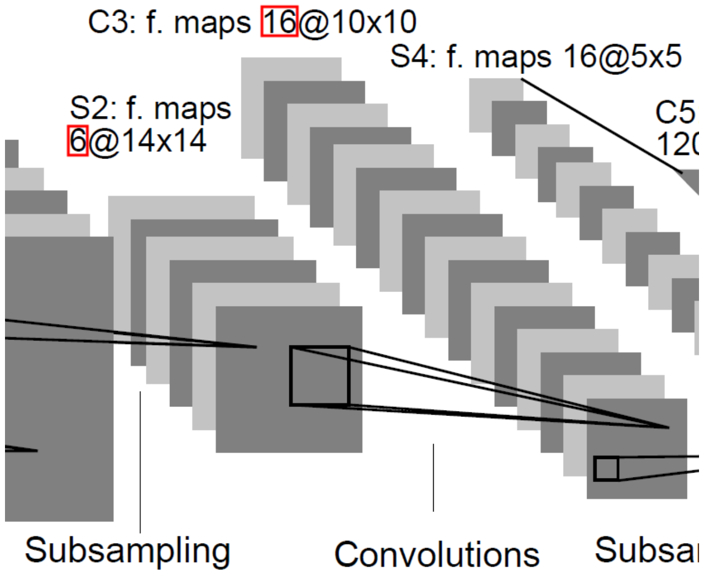
LeNet-5

- Designed for character recognition
- 7 layers (3 **c**onvolutional, 2 **s**ub-Sampling, 2 **f**ully-connected)
- Subsampling: average pooling with trainable coefficient and bias
- At each layer, the number of feature maps is increased as the spatial resolution is decreased



LeNet-5 (cont.)

- Not every S2 feature map is connected to every C3 feature map
- Reasons:
  - Keeps numbers of connections low
  - Break of symmetry (extract different features)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

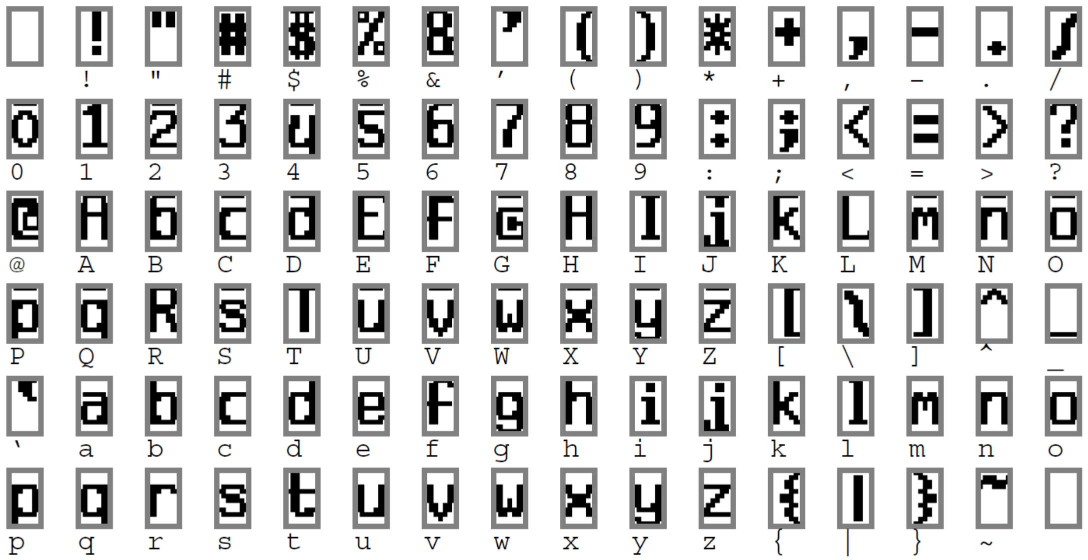
Each column indicates which feature map in S2 are combined by the units in a particular feature map of C3.

LeNet-5 (cont.)

- Output layer is composed of Euclidian Radial Basis Function units (RBF):

$$y_i = \sum_j (x_j - w_{ij})^2$$

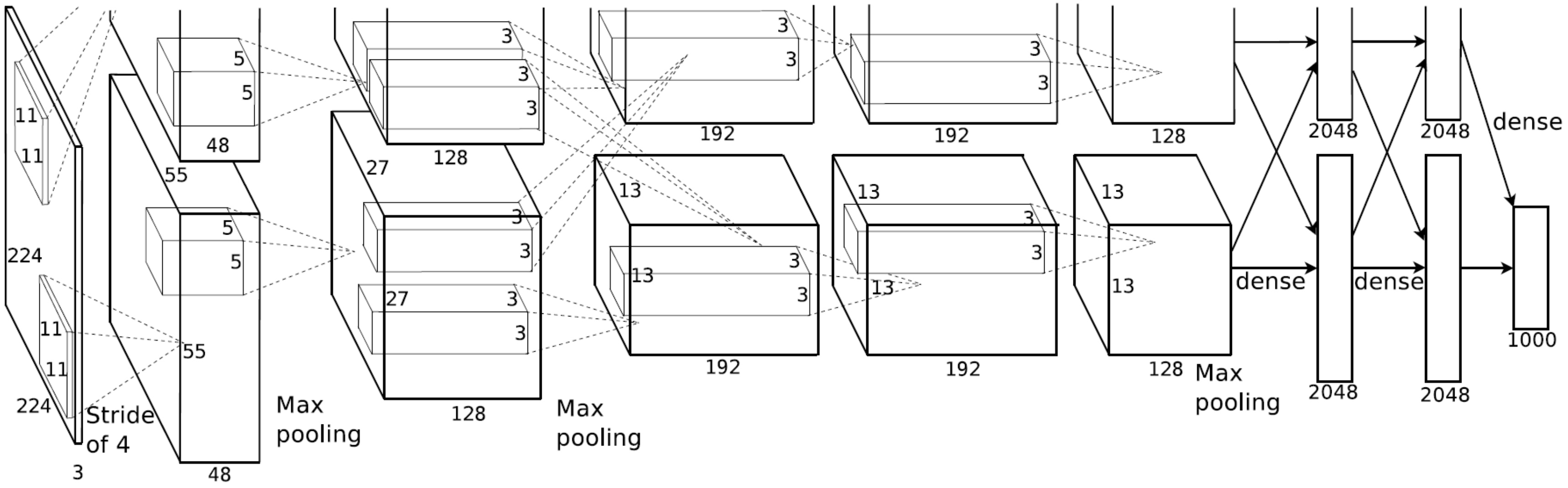
- Each RBF unit computes the distance between input and parameter vector (**penalty**)
- Especially useful for recognizing strings (0/O/o, 1/l/I)



LeNet-5 Misclassifications

AlexNet

- Designed for ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), 2012
- 11 layers (5 convolutional, 3 sub-sampling, 3 fully connected)
- Implementation on 2 GPUs
- Subsampling is done by overlapping max pooling
- Training set: 1.2 million labeled images



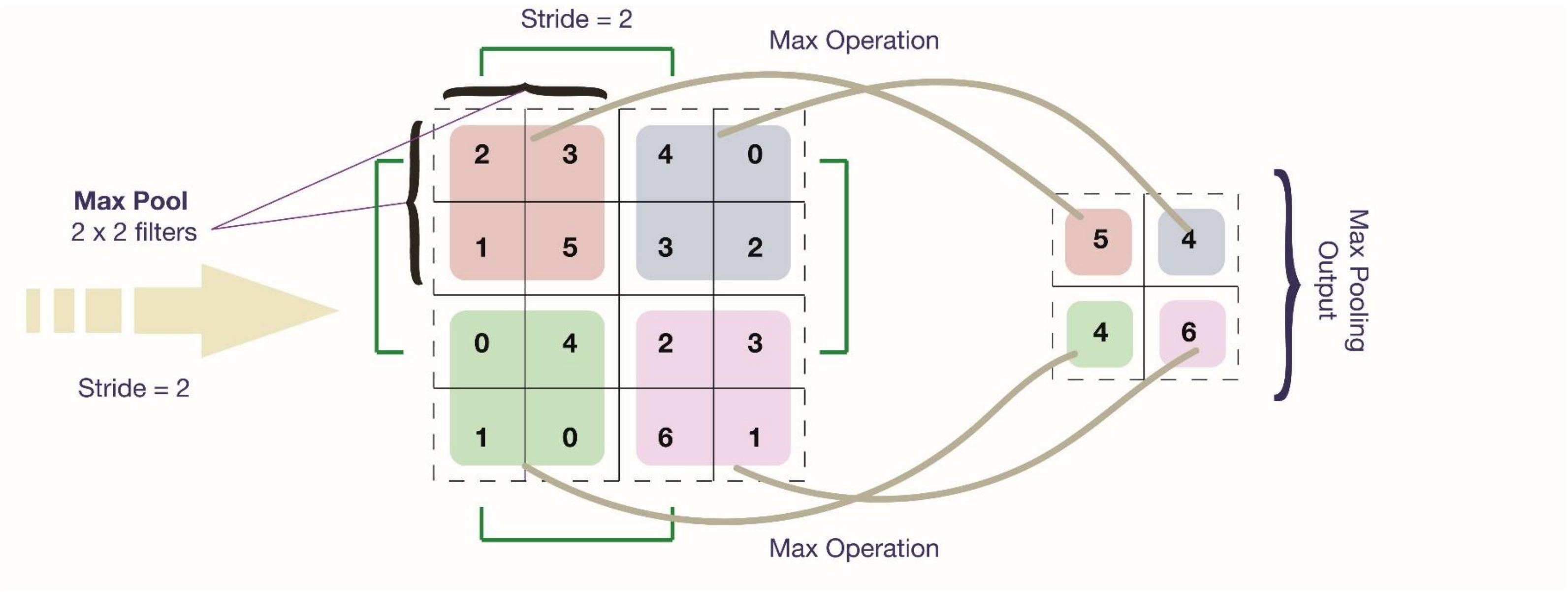
AlexNet: Local response normalization

- Activity of a neuron is normalized by running over n "adjacent" kernel maps at the same position
- $b^i_{x,y}$: response-normalized activity by applying kernel *i* at position *(x,y)*
- $a^i_{x,y}$: activity of a neuron by applying kernel *i* at position *(x,y)*
- k, n, α, β: hyper-parameters $(k=2, n=5, \alpha=10^{-4}, \beta=0.75)$

$$b^i_{x,y} = a^i_{x,y} \Big/ \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

AlexNet: Overlapping pooling

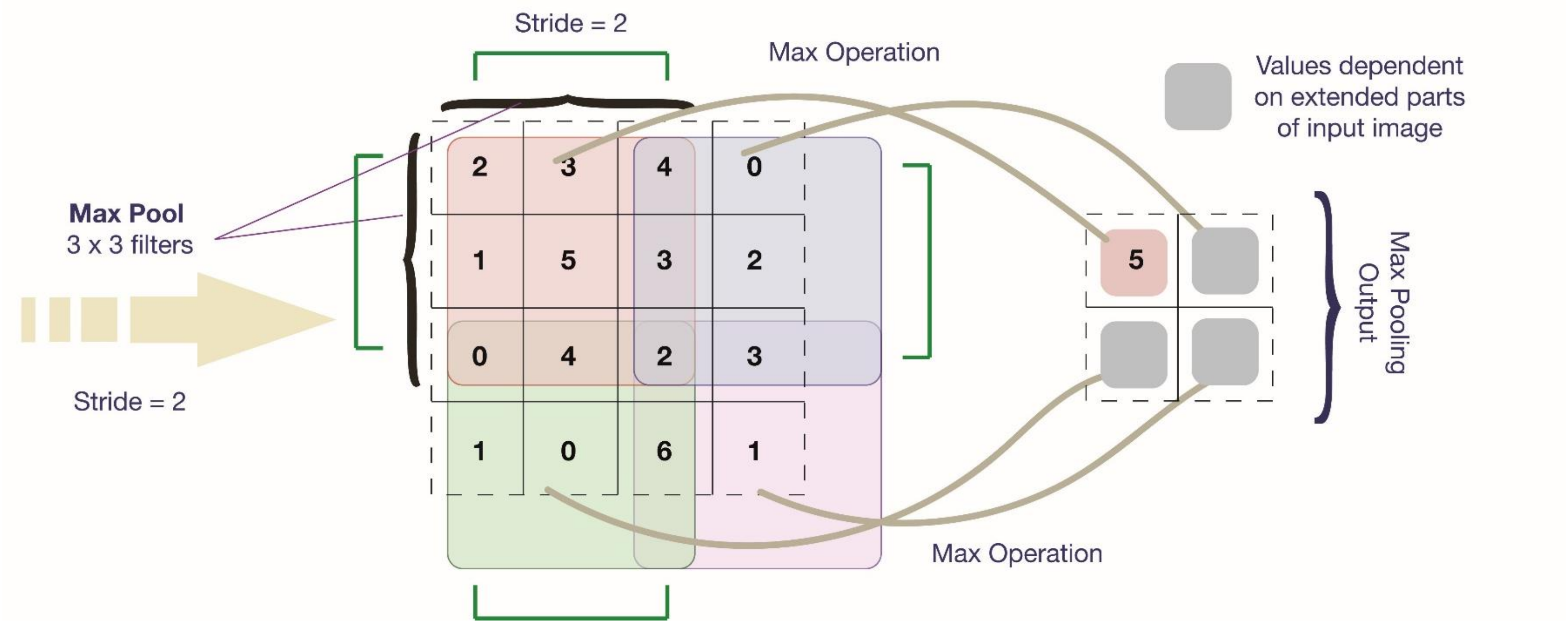
- Stride is smaller than the size of the pooling unit
- Increase of accuracy



Non-overlapping pooling

AlexNet: Overlapping pooling (cont.)

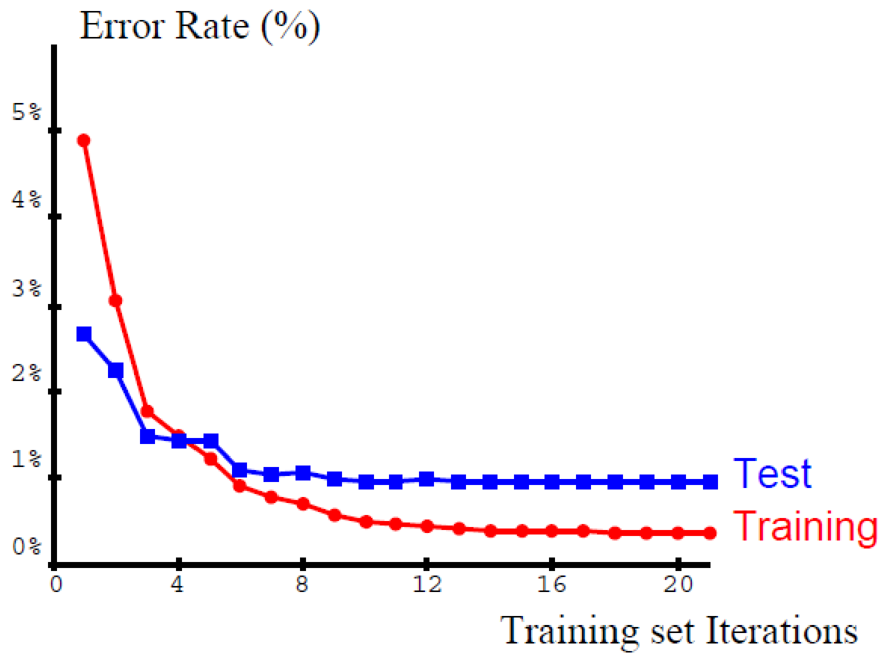
- Stride is smaller than the size of the pooling unit
- Increase of accuracy

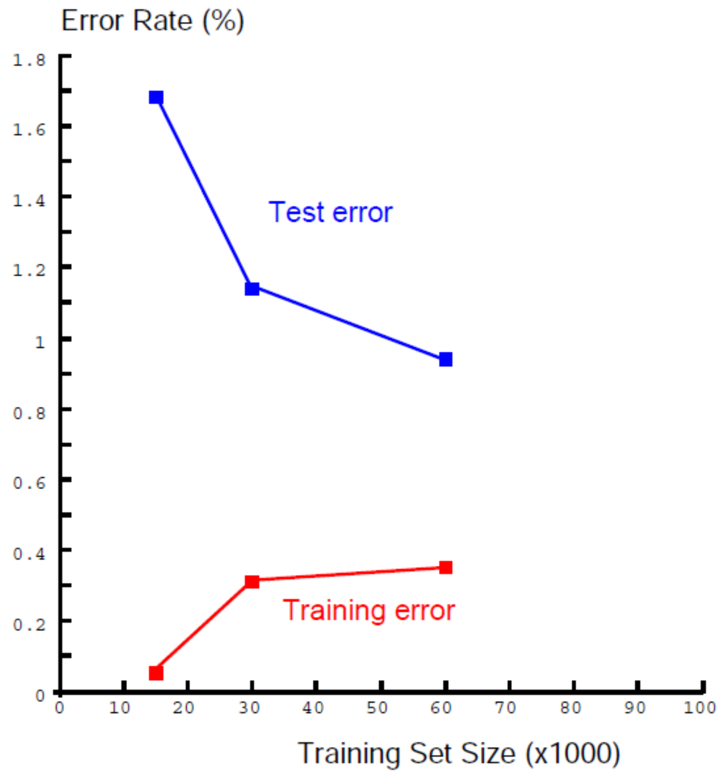


Overlapping pooling

Learning process

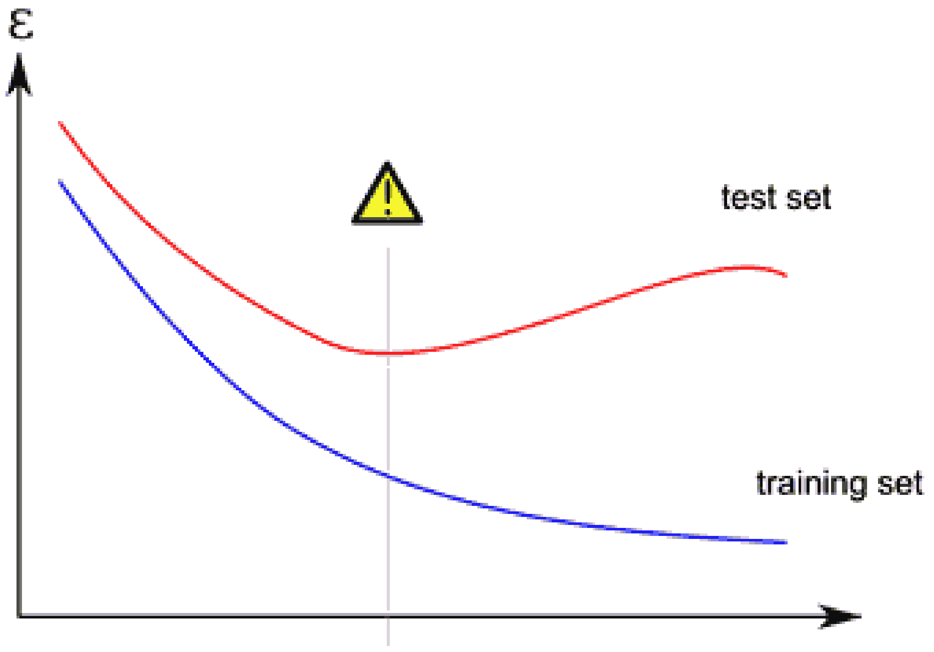
- Perform multiple iterations on the training set
- More training data improves the accuracy
- In this case: 60,000 training examples

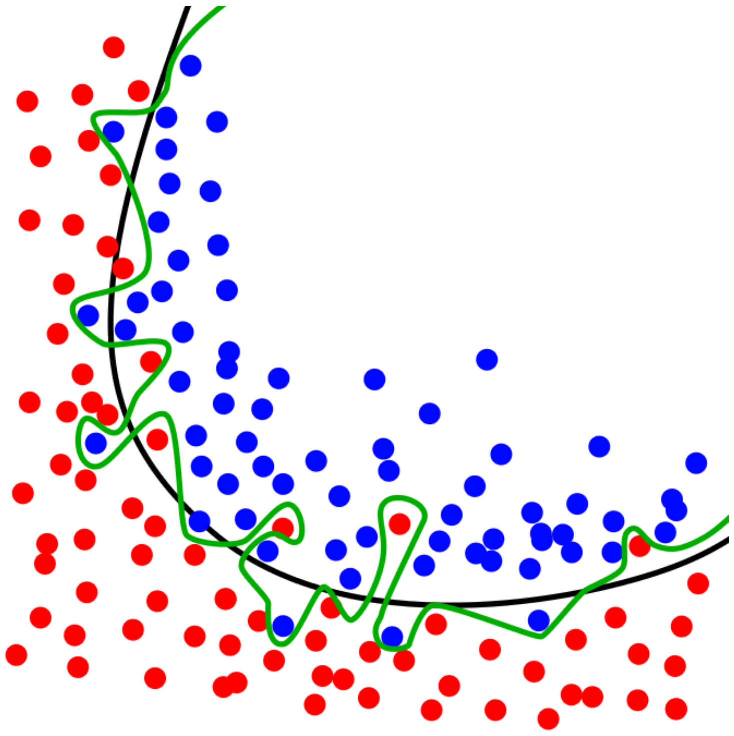




Overfitting

- Test error increases (while the training error decreases) because the neural net adapts too closely to the training set.



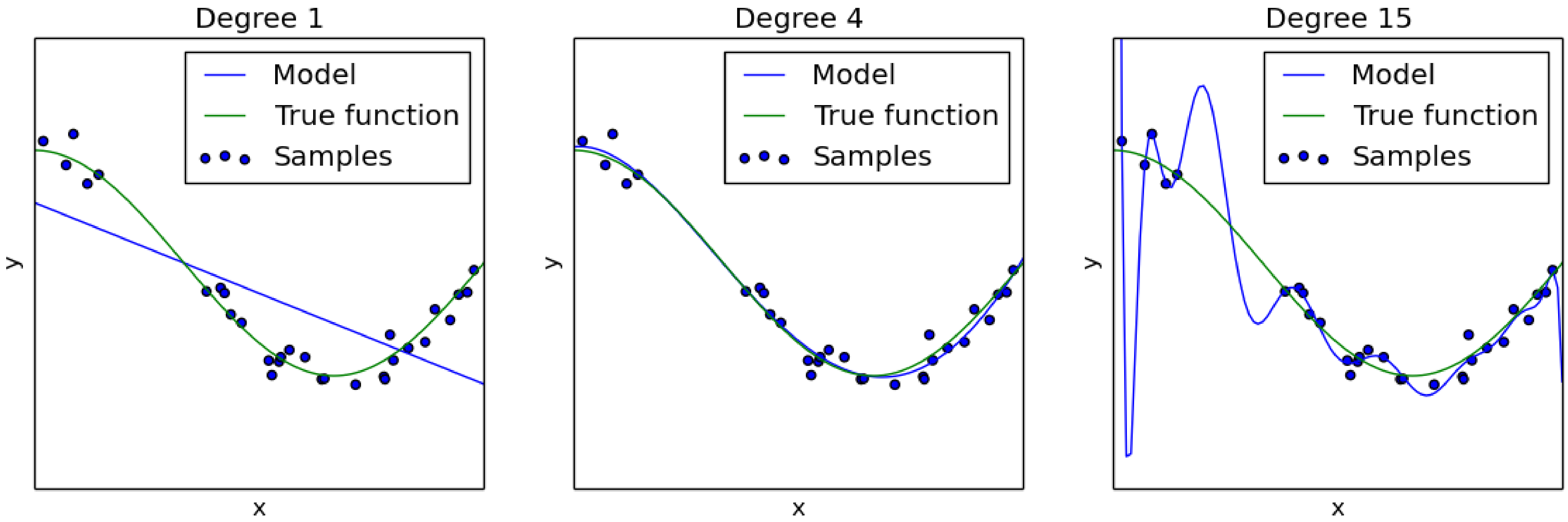


Reducing overfitting

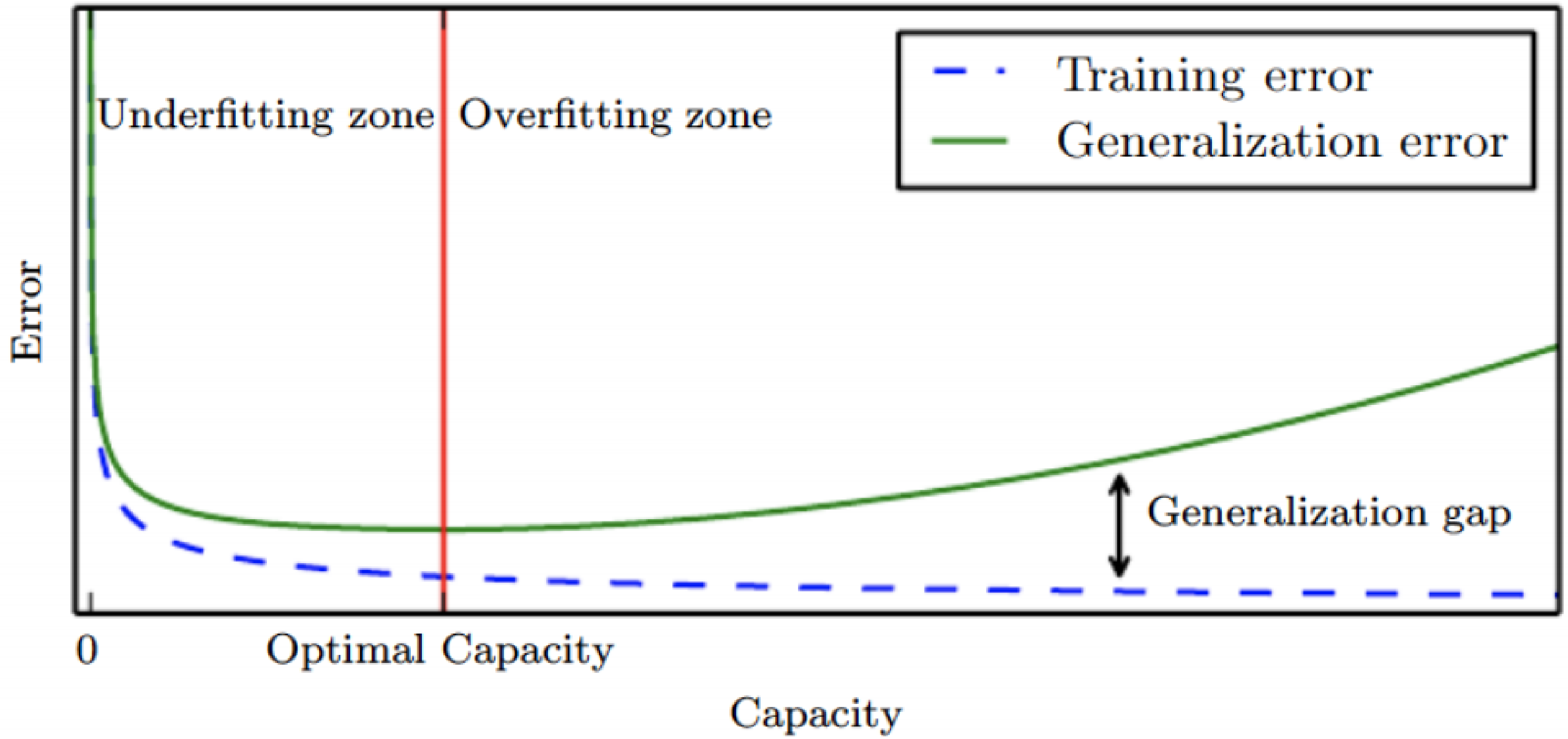
- Reducing capacity
- Data augmentation
- Dropout

Reducing capacity

- Reduces the degree of the function
- Problem: Risk of underfitting

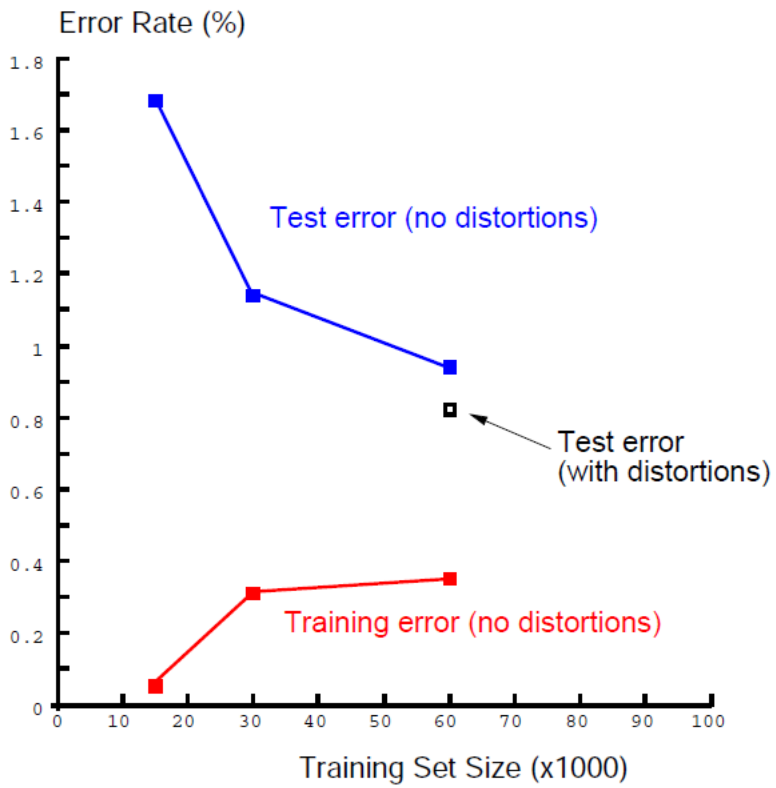


Reducing capacity (cont.)



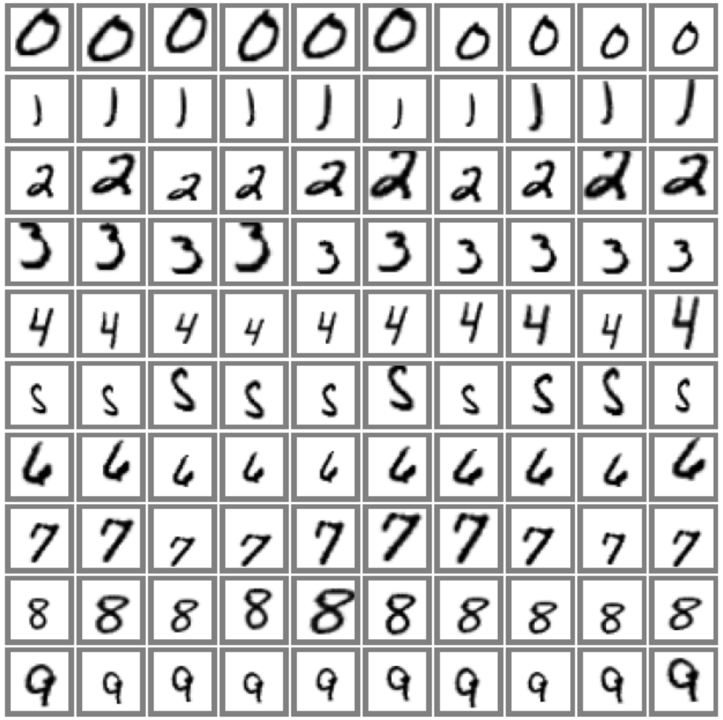
Data augmentation

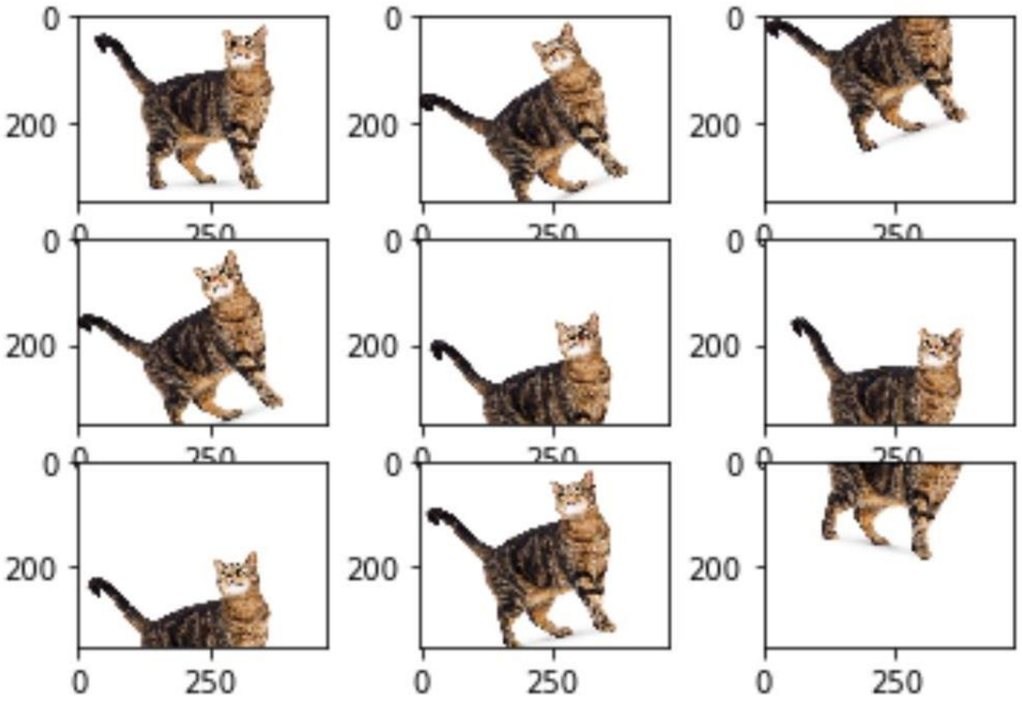
- Idea: Artificially generate more training examples by randomly **distorting** the original training images
- Also improves overall accuracy



Data augmentation (cont.)

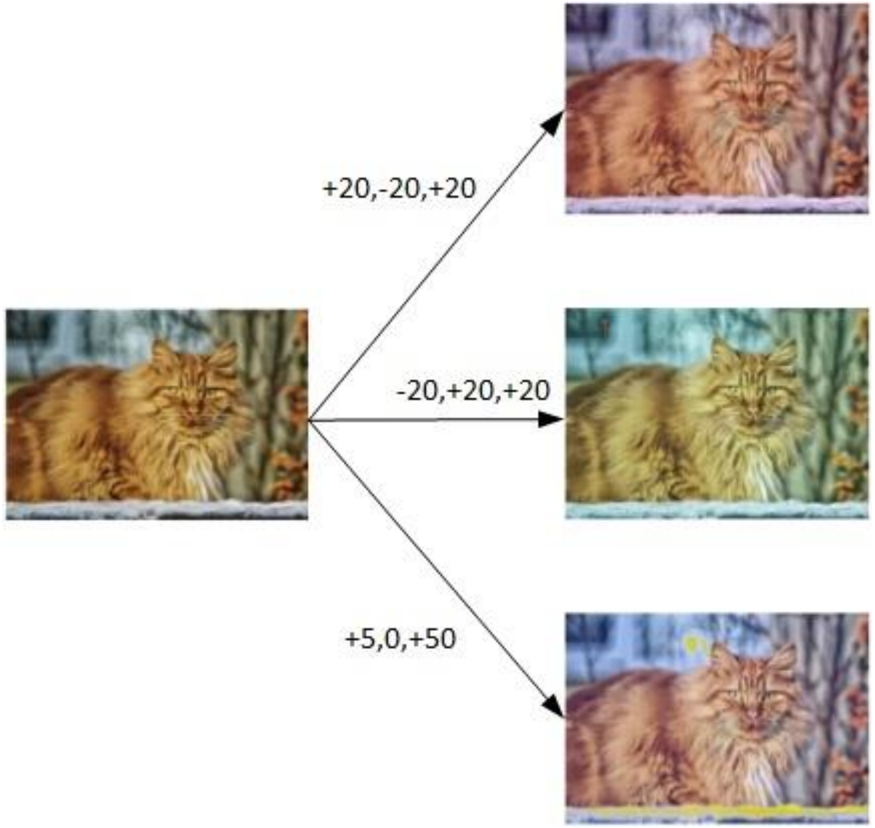
- E.g. translation, scaling, squeezing, horizontal shearing

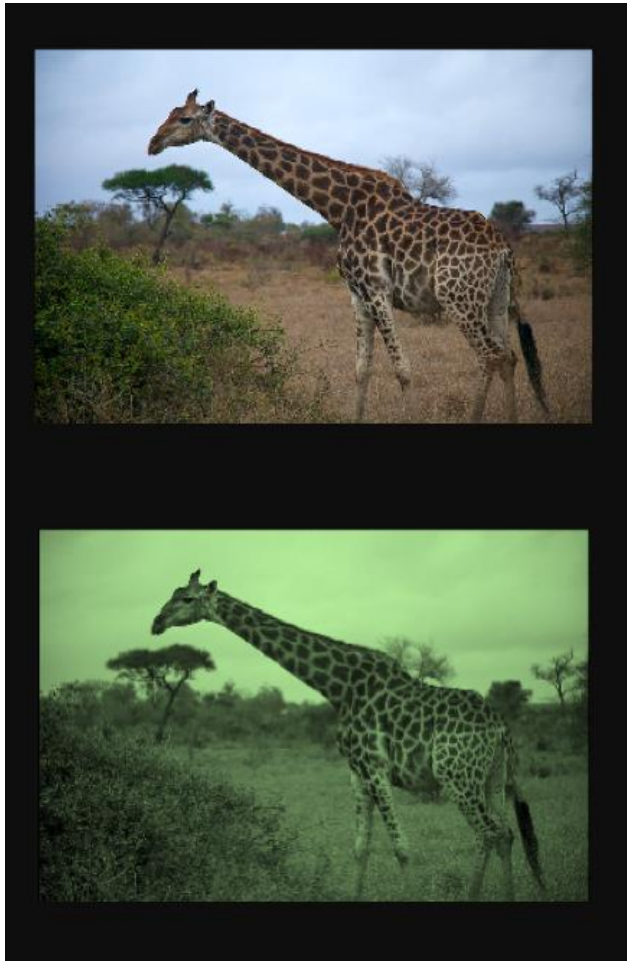




Data augmentation (cont.)

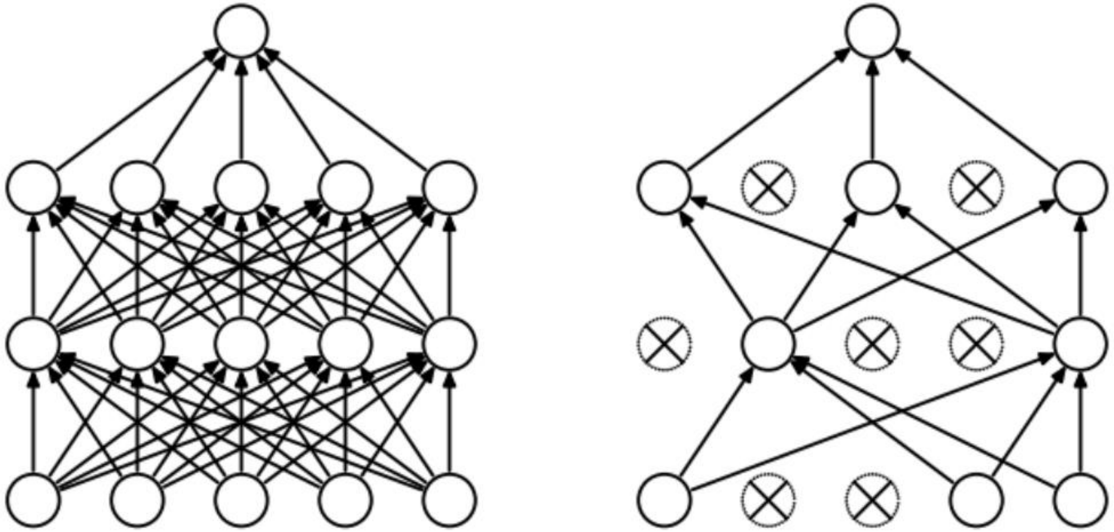
- For RGB images: change in color





Dropout

- Set the output of each hidden neuron to zero with probability 0.5
- Alternating architecture reduce co-adaptions of neurons
- More robust features that are useful in conjunction with many different random subsets



(a) Standard Neural Net (b) After applying dropout.

Examples (AlexNet)