Algorithm

Introduction To Artificial Intelligence

Chen Yishuai

yschen@bjtu.edu.cn

School of Electronic Information Engineering, Beijing Jiaotong University

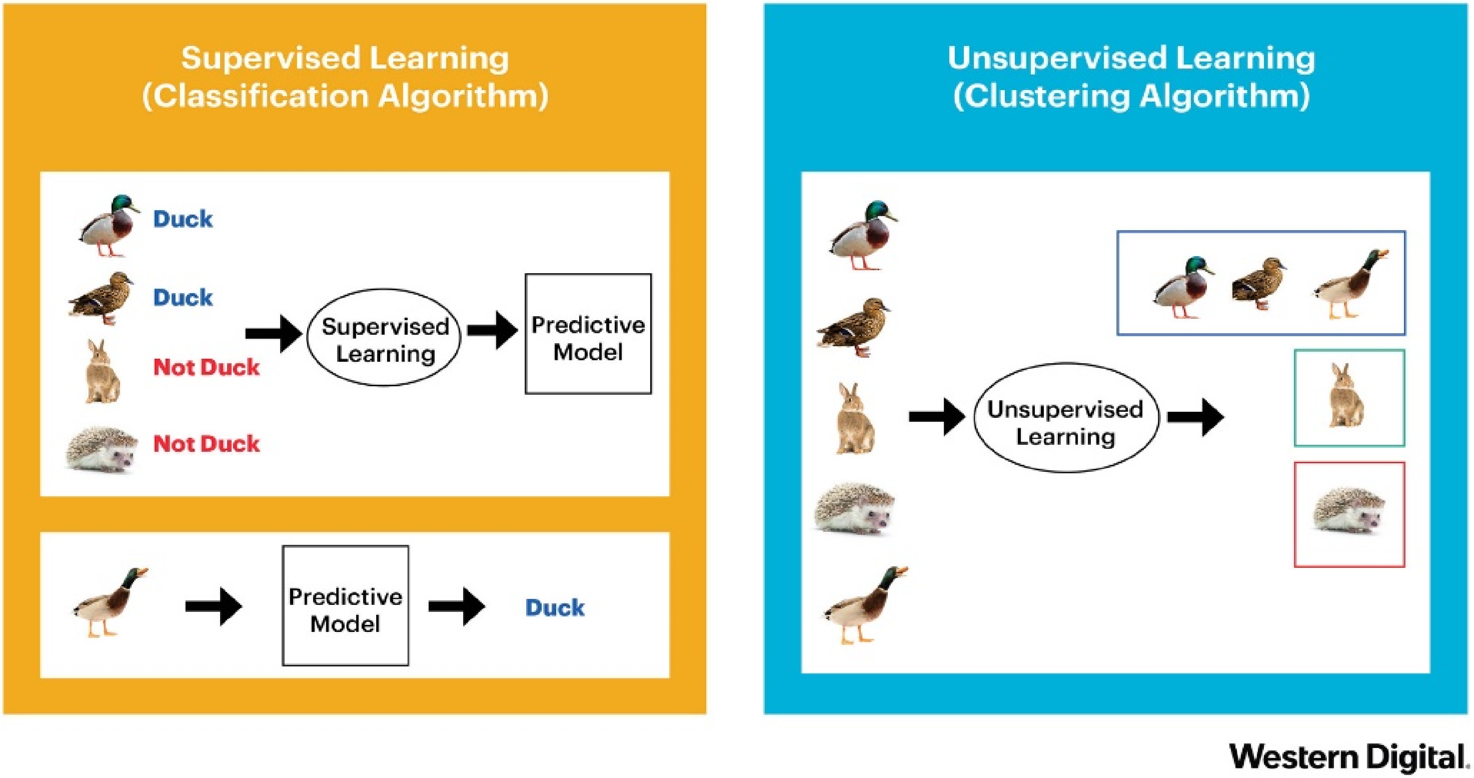
Content

- Introduction
- Machine learning model
- Deep learning model
- Model training
- Model selection

Algorithms

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement learning

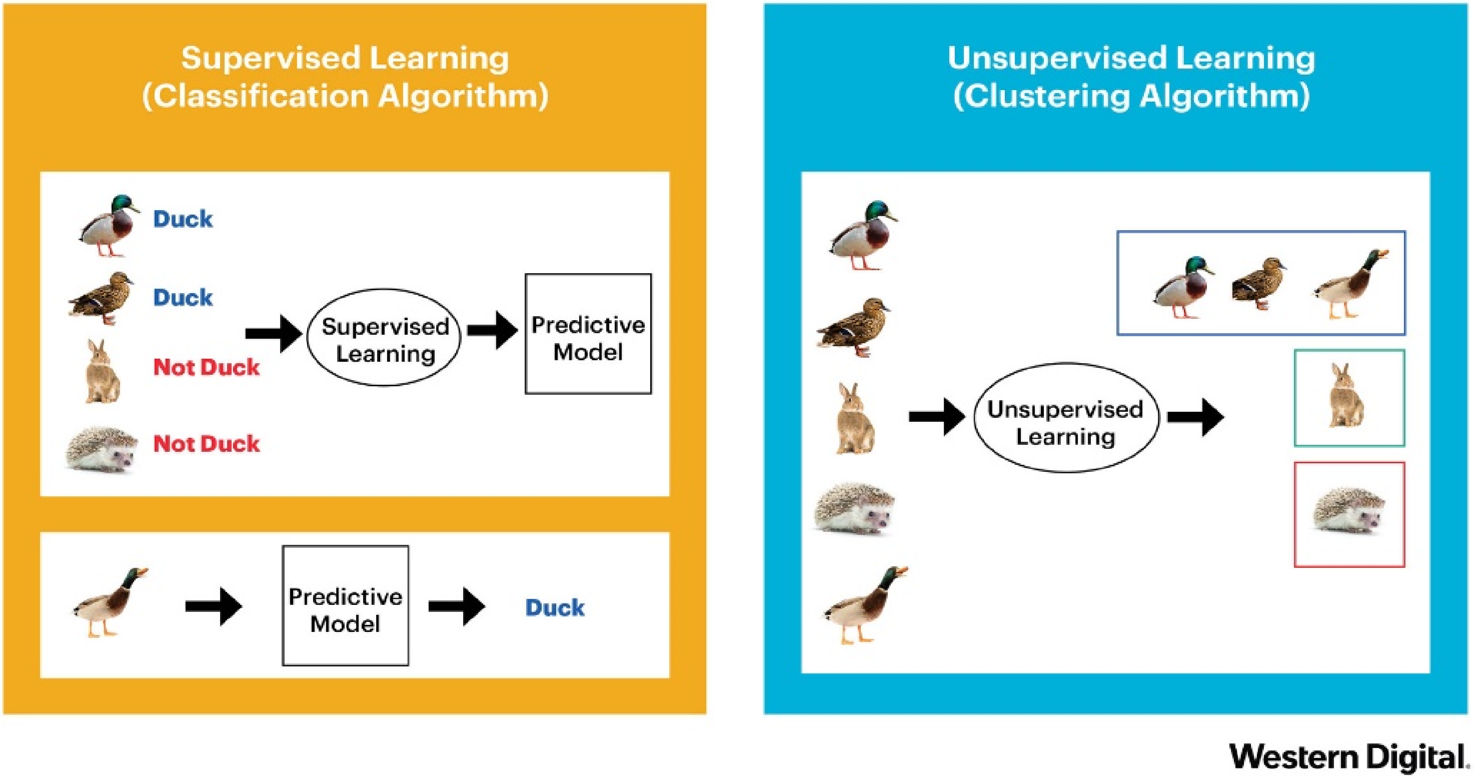
Supervised



Labeled correct answers, e.g., picture categories

Learn a model to obtain correct answer

Unsupervised



No labeled correct answer, e.g., only pictures

Use algorithm to learn the data pattern

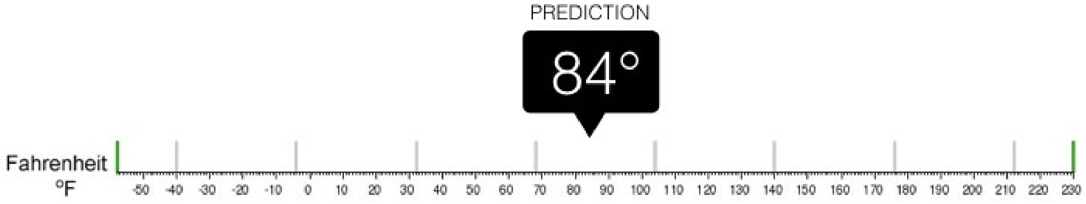
1) Supervised Learning

Known correct answer

Supervised Learning

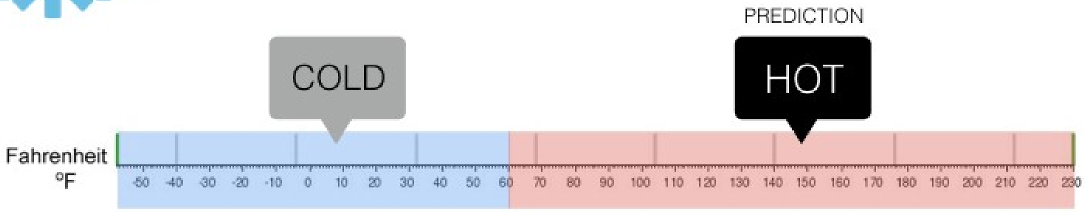
Regression

What is the temperature going to be tomorrow?



Classification

Will it be Cold or Hot tomorrow?



Steps

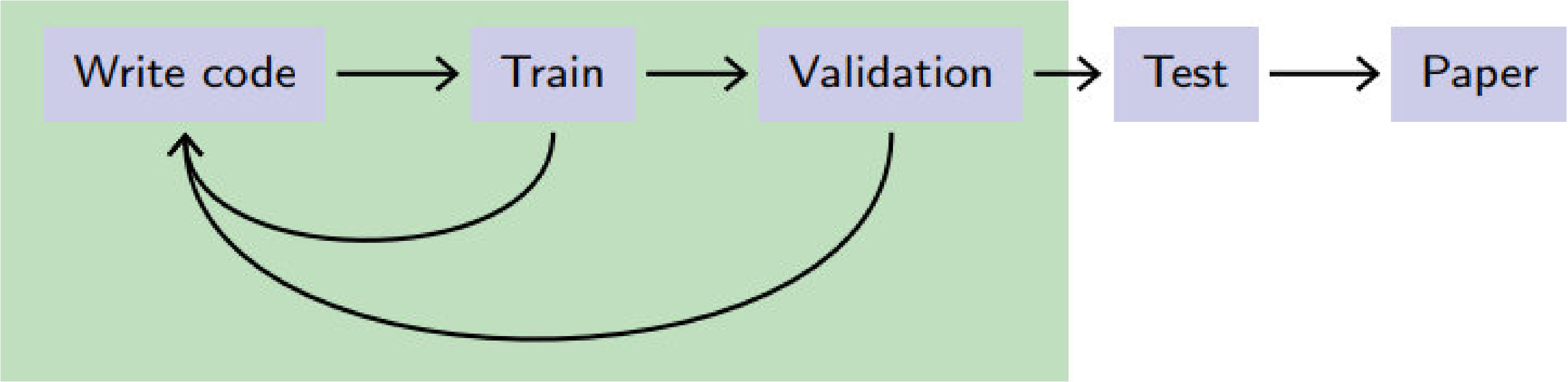
1. Labeling
2. Training
3. Testing

Preparing Data

1. Collecting data set
2. Labeling
    - Label the pictures: "Cat", "Dog"
3. Divide the data into three parts
    - Training set: training model
    - Validation set: selecting model parameters
    - Testing set: test model accuracy

Training Model

1. Training: training the model
2. Validation: selecting model parameters
3. Testing: evaluate the model on the test set



2) Unsupervised learning

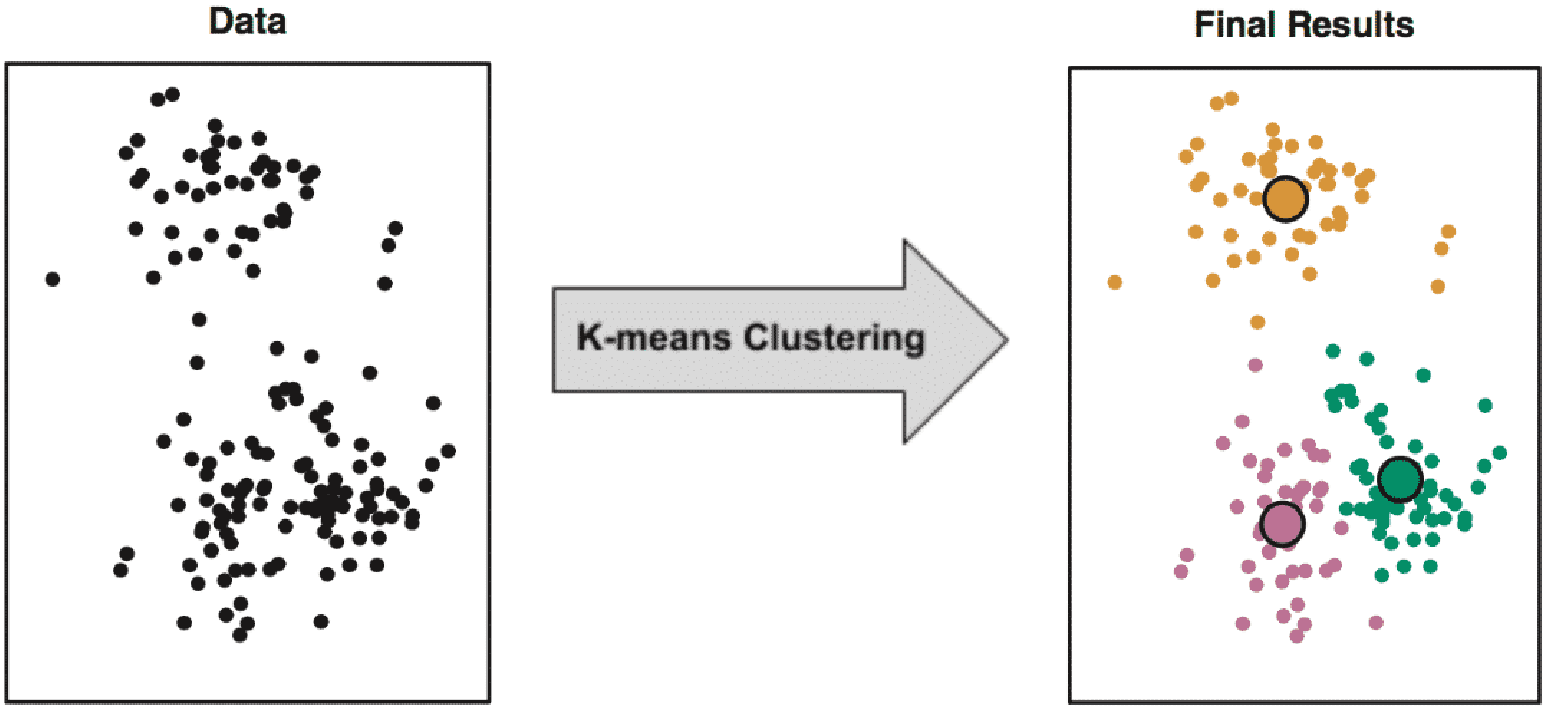
Without labels, look for patterns on the data

Unsupervised learning

- Clustering
- Outlier detection
- Auto-encoder
- Principal component analysis

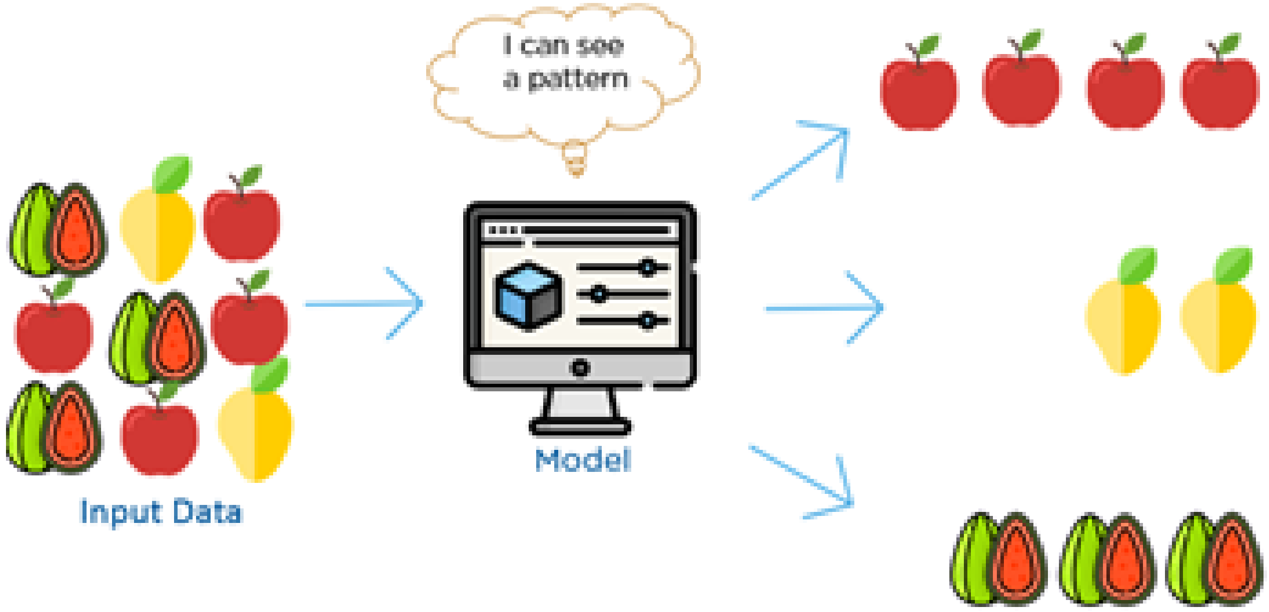
1) Clustering

Specify number of clusters: 3



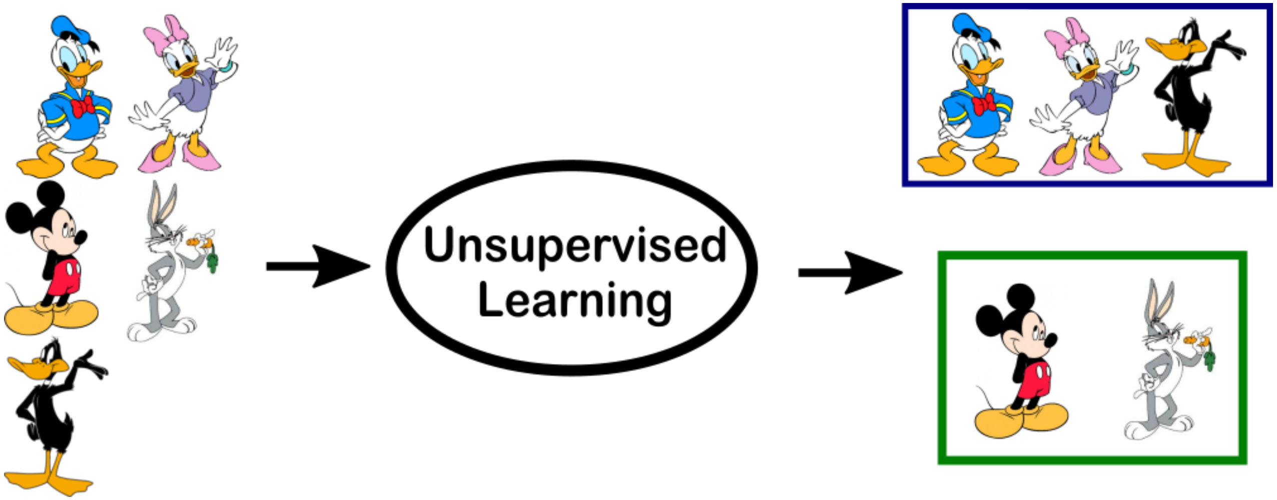
1) Clustering

- After clustering, observe each cluster to get its meaning
- The result might look like this:



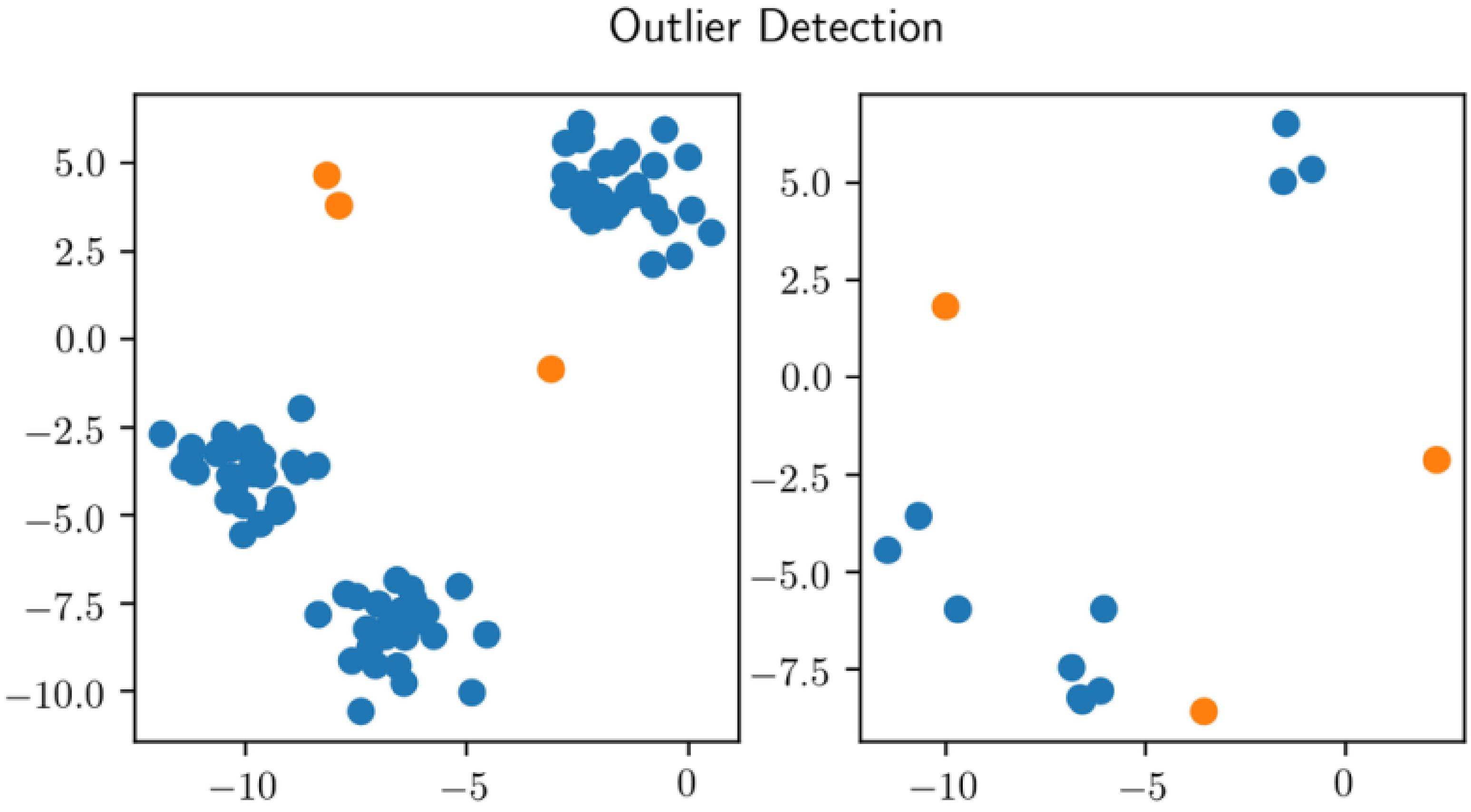
1) Clustering

- The result may also be like this



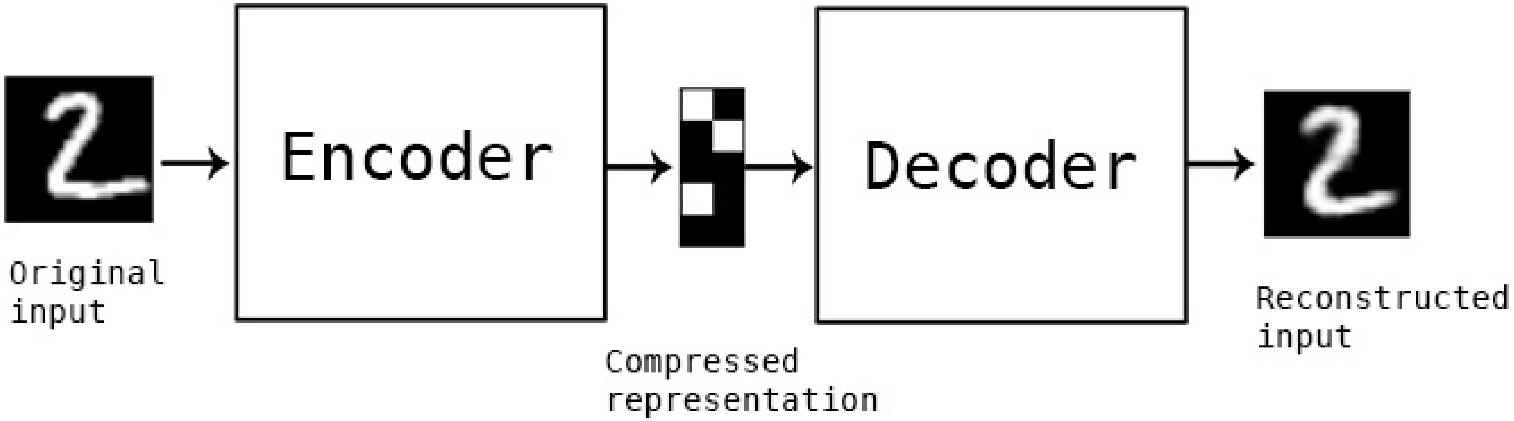
2) Outlier Detection

Find outliers, i.e., abnormal points



3) Auto-Encoder

- Encoding: get compressed representation of original image
- Decoding: restore original image based on compressed representation

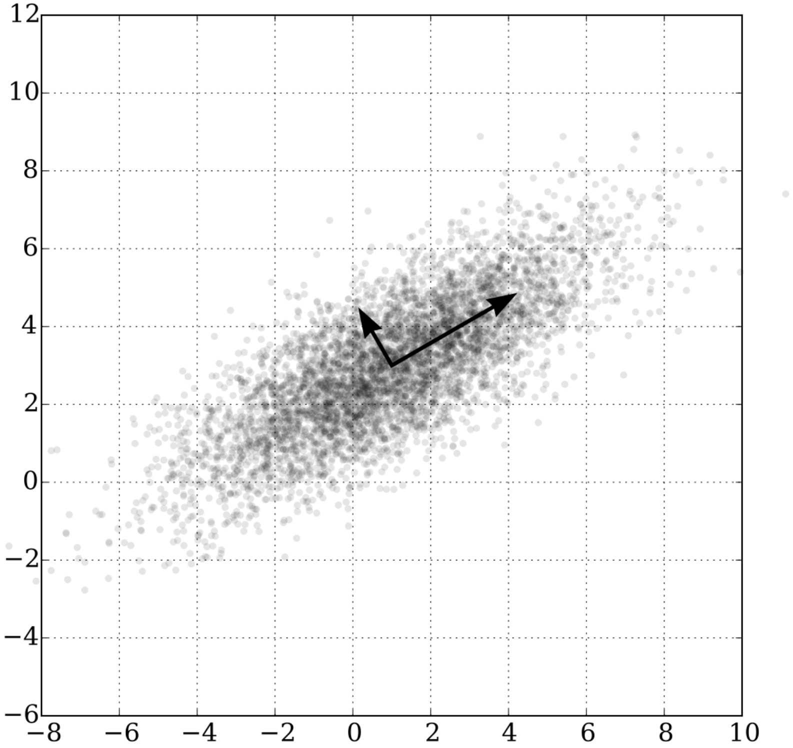


3) Auto-Encoder

- The result of compression is the code of the data obtained by auto-encoding
- Generally, deep neural network is used as encoder and decoder

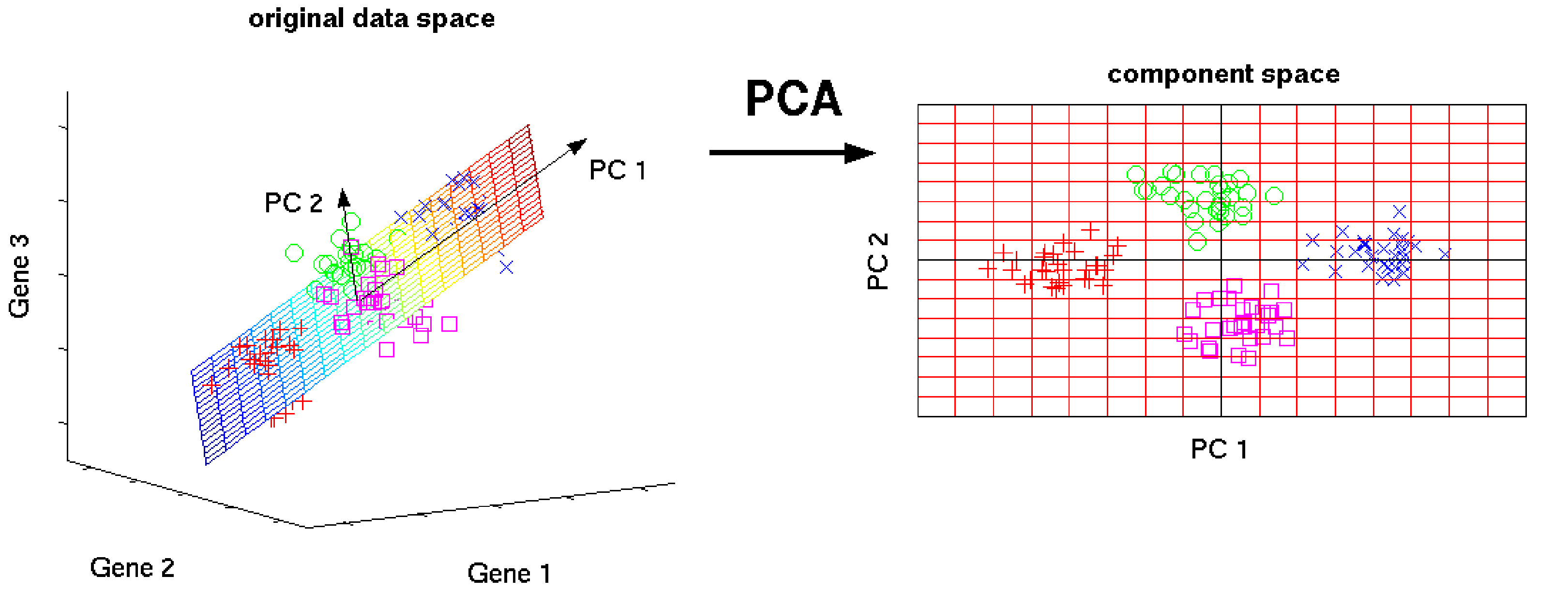
4) PCA: Principal Component Analysis

- The data information is mainly on its principal component vector



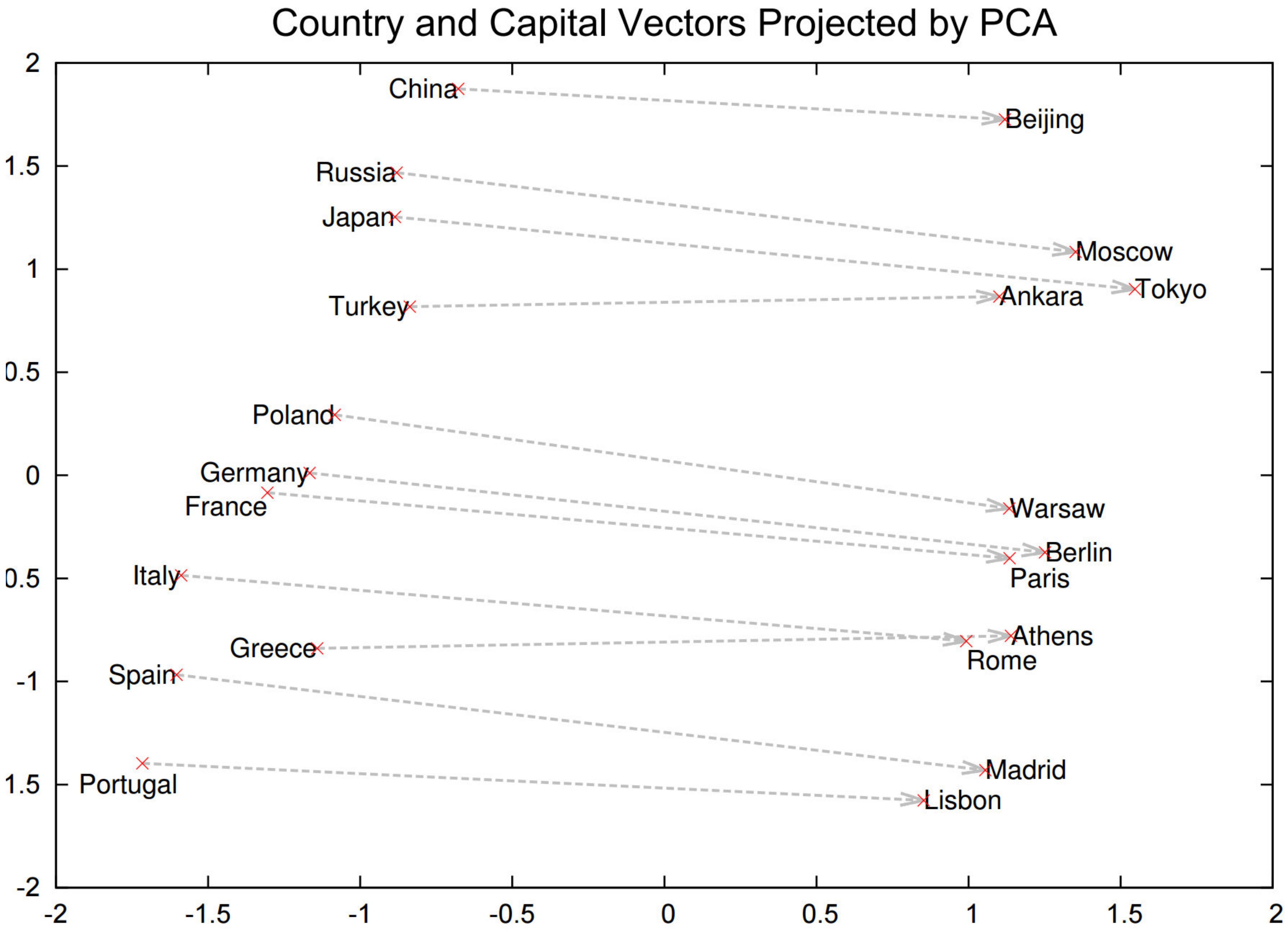
4) PCA

- Use PCA to represent 3D data in 2D
- Little information is lost, achieving dimensionality reduction



4) PCA

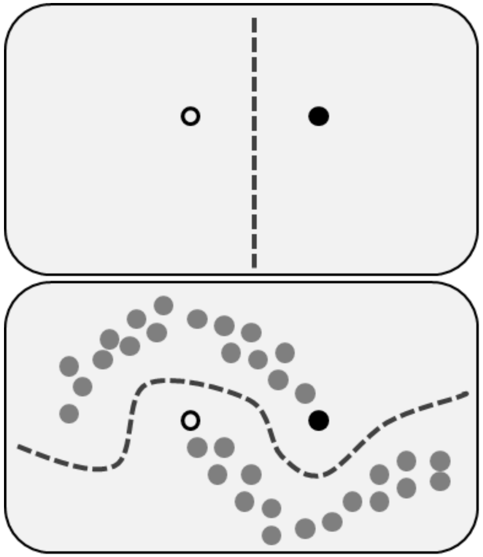
Word representation using PCA



3) Semi-Supervised Learning

Semi-Supervised Learning

- Labeling is time-consuming and labor-intensive
- Uses a large amount of data without labeling
- Combines a small amount of labeling data to improve performance

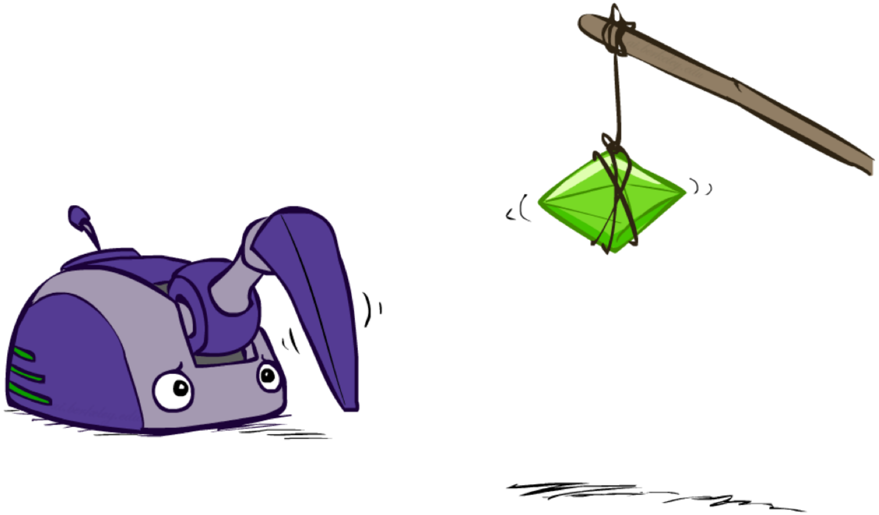


4) Reinforcement Learning

Learning based on the rewards received

Reward-Based Learning

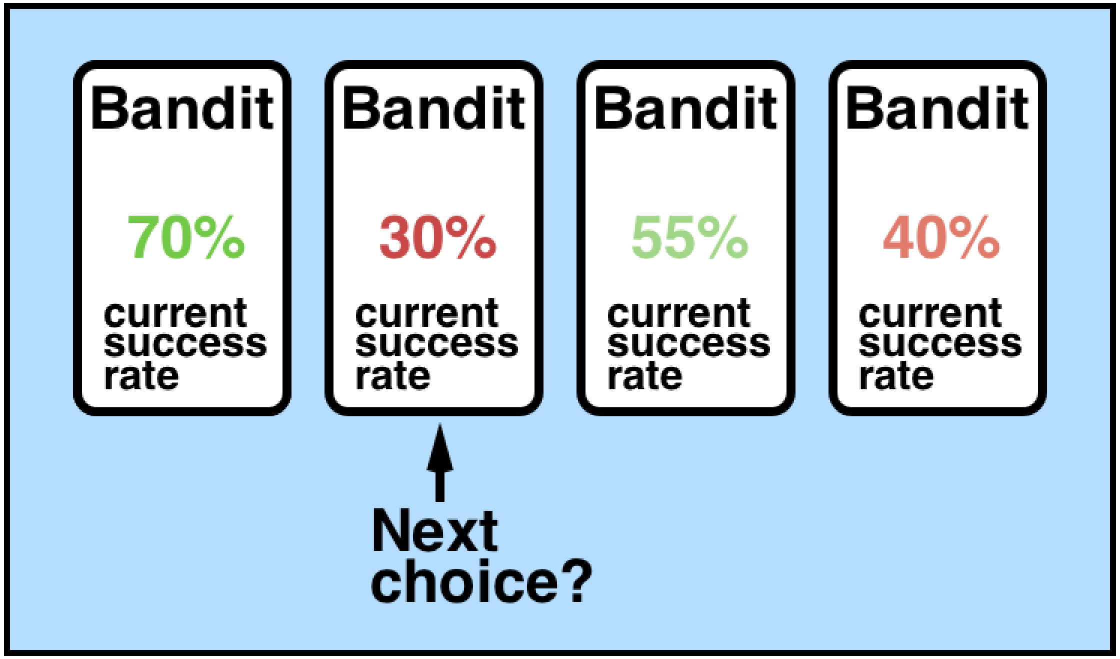
- No labeled data set
- There is a reward
- Learning based on the rewards received
- Goal: maximize reward



Multi-Arm Bandit

Which machine to choose?

Problems



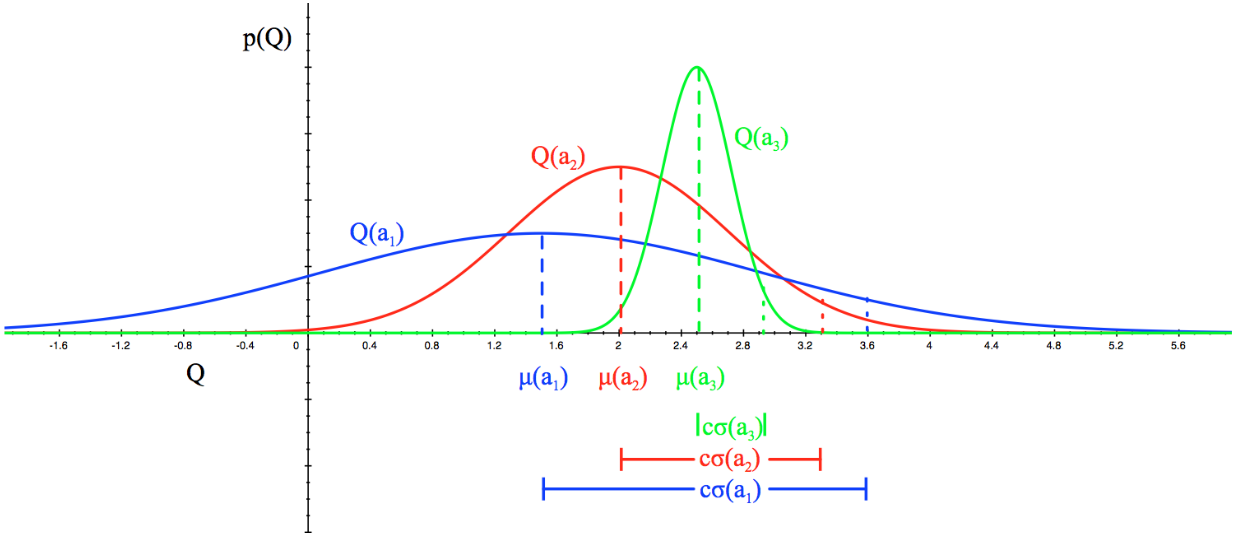
- "Utilization": Play the highest win rate machine ever found
- "Exploration": Play on machines that have not been fully explored

Key

Balance "utilization" & "exploration"

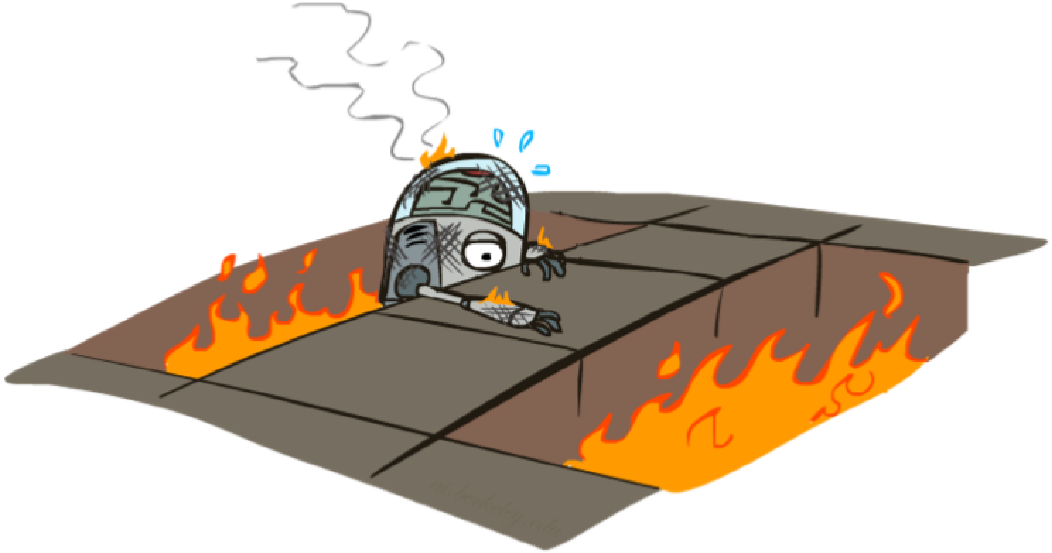
UCB Algorithm

- Upper Confidence Bounds: Upper bound of confidence interval
- Includes average win rate (mean) and exploration space (standard deviation)
- Balance "utilization" and "exploration"



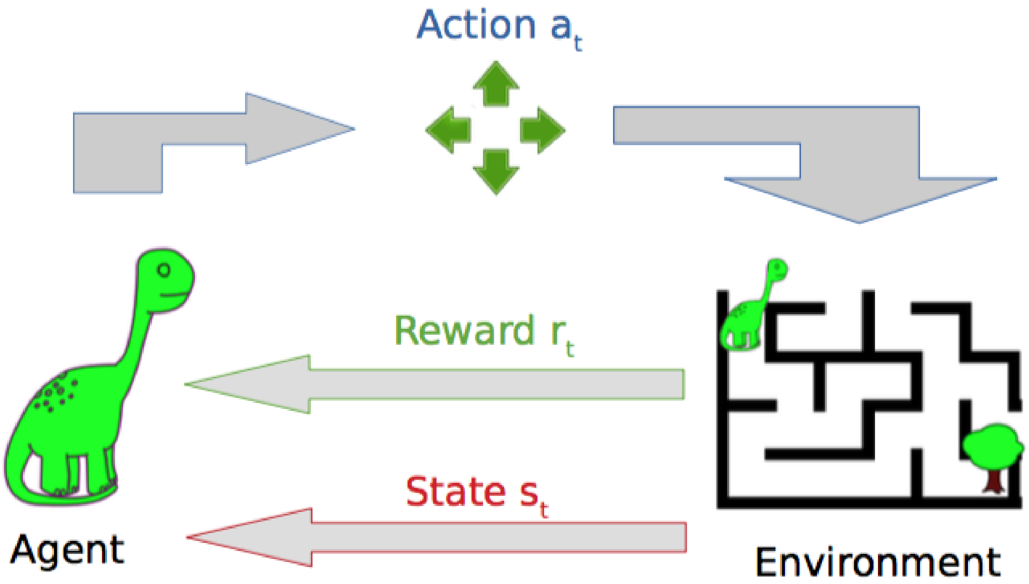
Reinforcement Learning

- Make a lot of experiments
- Don't be afraid to jump into the fire pit
- Replay



Reinforcement Learning

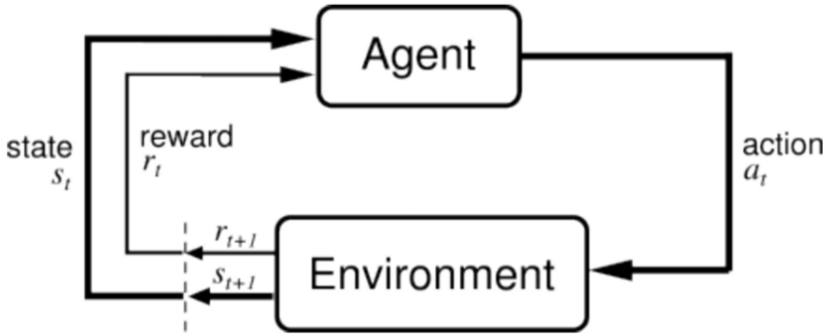
- Keep trying
- Get the "value" of each position
- Or get the best action in every position



MDP: Markov Decision Process

An MDP is defined by:

- Set of states S
- Set of actions A
- Transition function $P(s' \mid s, a)$
- Reward function $R(s, a, s')$
- Start state s_0
- Discount factor γ
- Horizon H



Application

- Robot
- Game
- Automatic control

Challenge

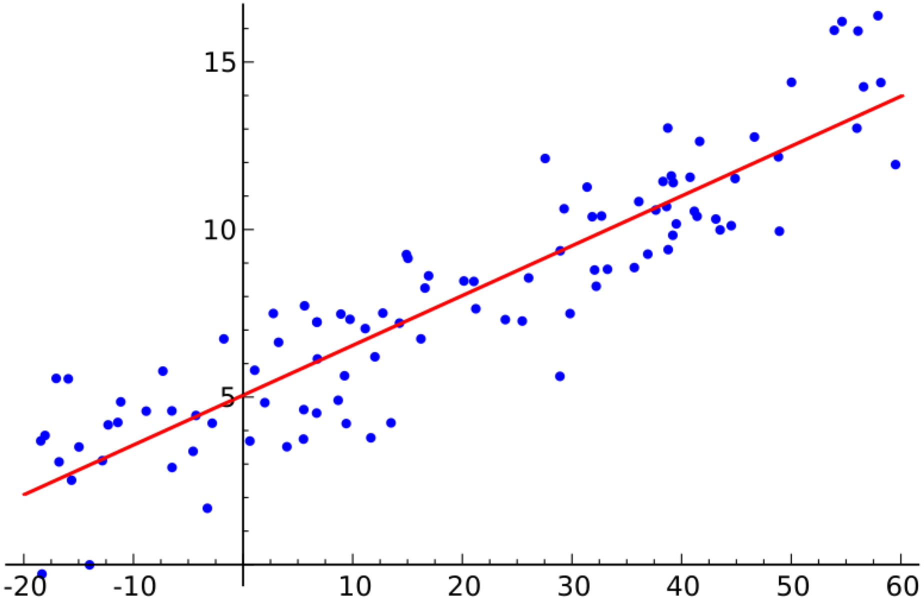
- Reward is delayed: examination results will not be known until the end of the semester
- Sparse reward feedback: only one final exam per semester

Summary

1. Supervised learning
   - Known correct answer (label)
2. Unsupervised learning
   - Discovering patterns from data
3. Semi-supervised learning
   - Leverage large amounts of data without labeling
4. Reinforcement Learning
   - Learn by trying

Model

Linear Regression

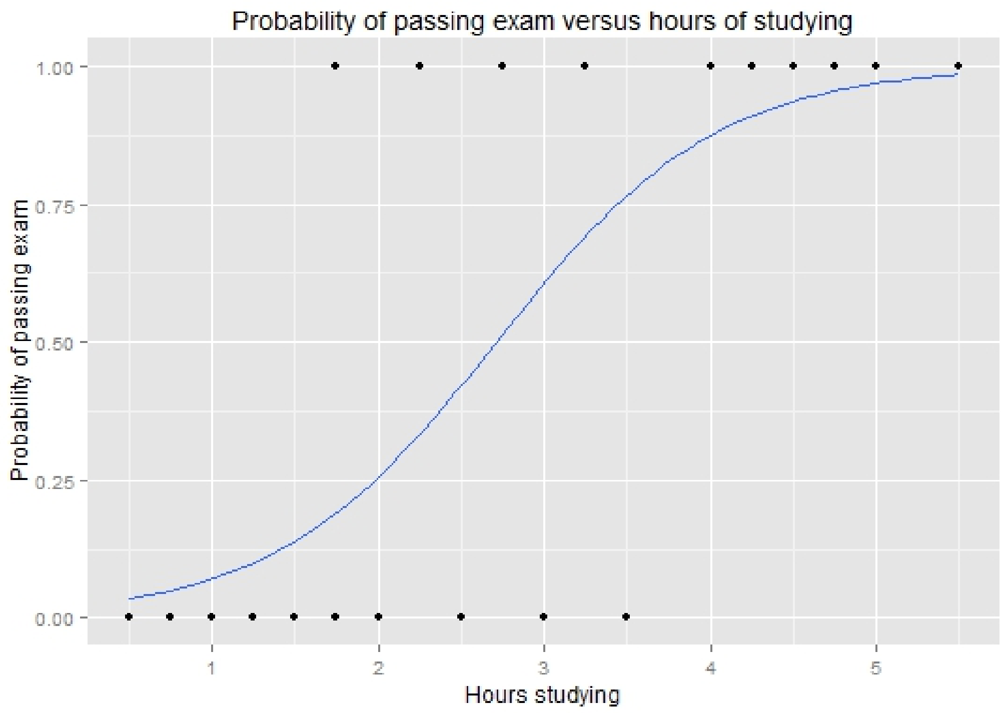


Straight Line

Logistic Regression

Logistic Regression

- Classification model
- Relationship between exam passing probability and study time



S Curve

Perceptron

Model human brain neurons

Neuron Model

- Neurons (brain cells) are connected through synapses
- The brain constantly creates, strengthens, and weakens these connections

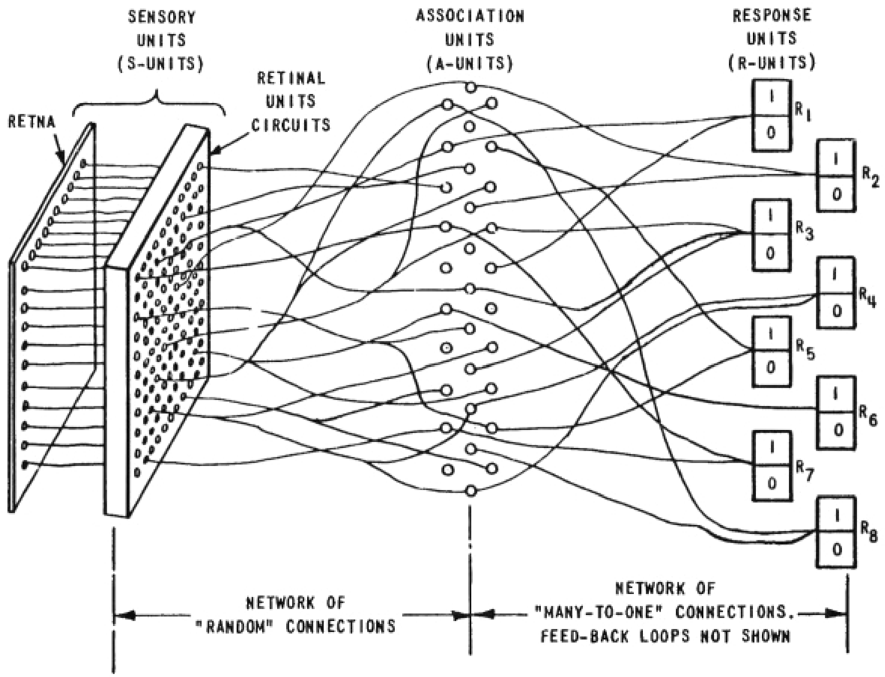

SENSORY UNITS (S-UNITS)
RETNA
RETINAL UNITS CIRCUITS
ASSOCIATION UNITS (A-UNITS)
RESPONSE UNITS (R-UNITS)
R1
R2
R3
R4
R5
R6
R7
R8
1
0
NETWORK OF "RANDOM" CONNECTIONS
NETWORK OF "MANY-TO-ONE" CONNECTIONS. FEED-BACK LOOPS NOT SHOWN

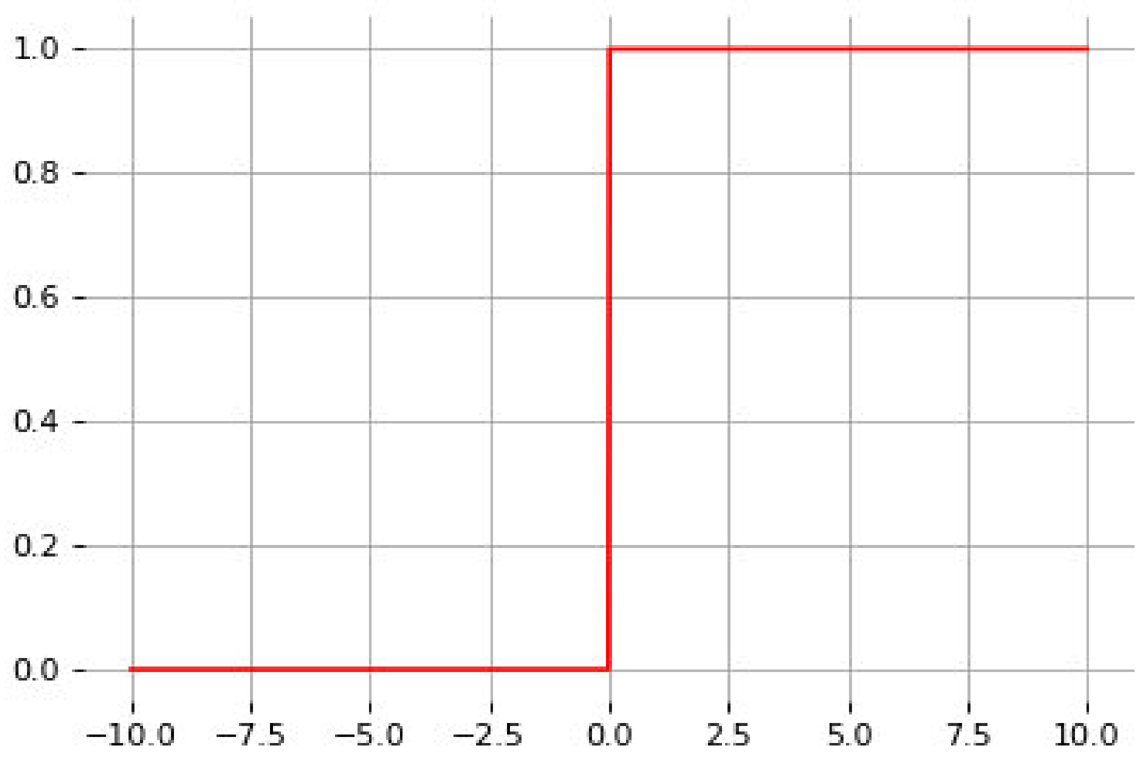
Figure I ORGANIZATION OF THE MARK I PERCEPTRON

Perceptron Model

- Linear weighted sum of inputs
    - Neuron input:
    - Connection weight:
    - Sum:

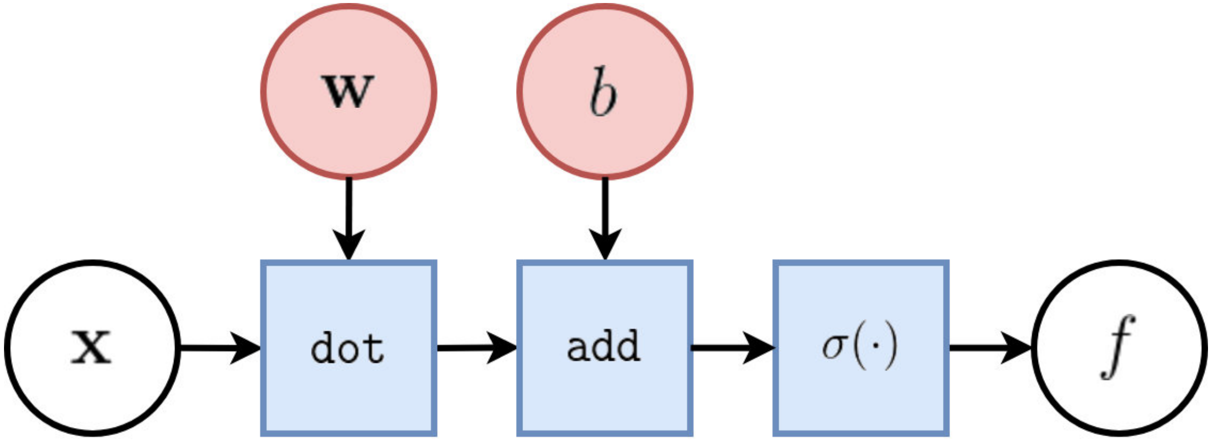
Perceptron Model

- Nonlinear activation function

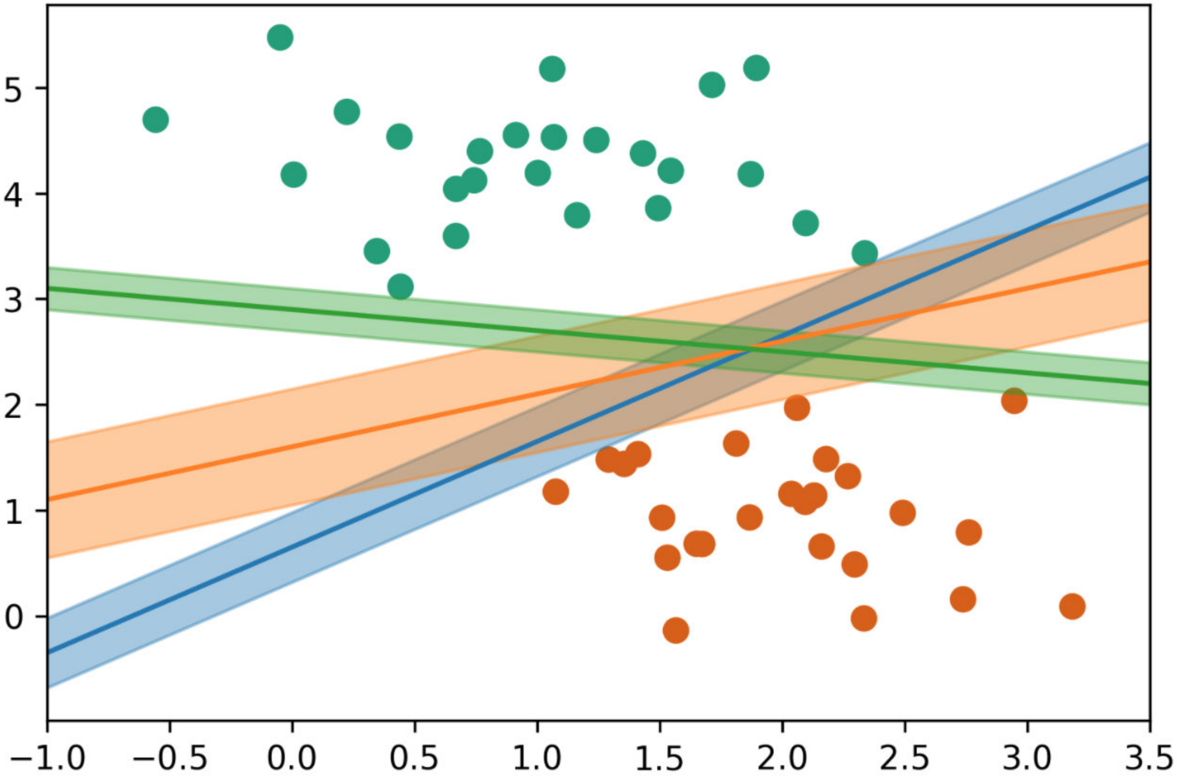


Perceptron Model

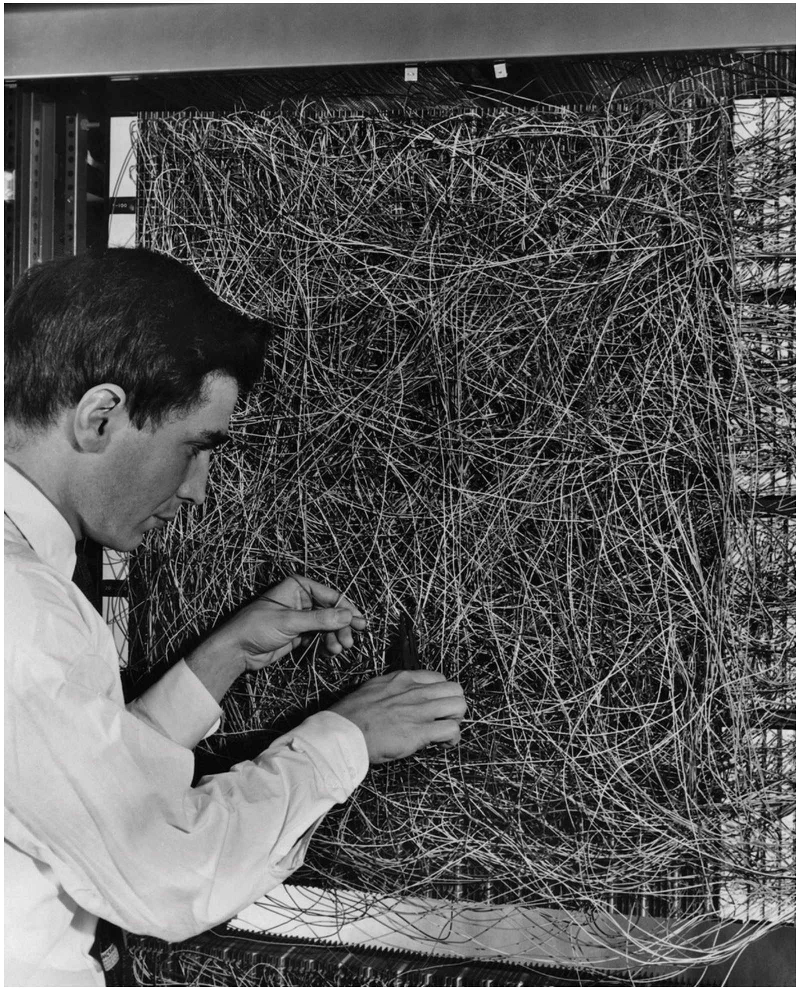
- Input linear weighting sum
- Non-linear activation function



Perceptron Model



Implementation



Model Training Method

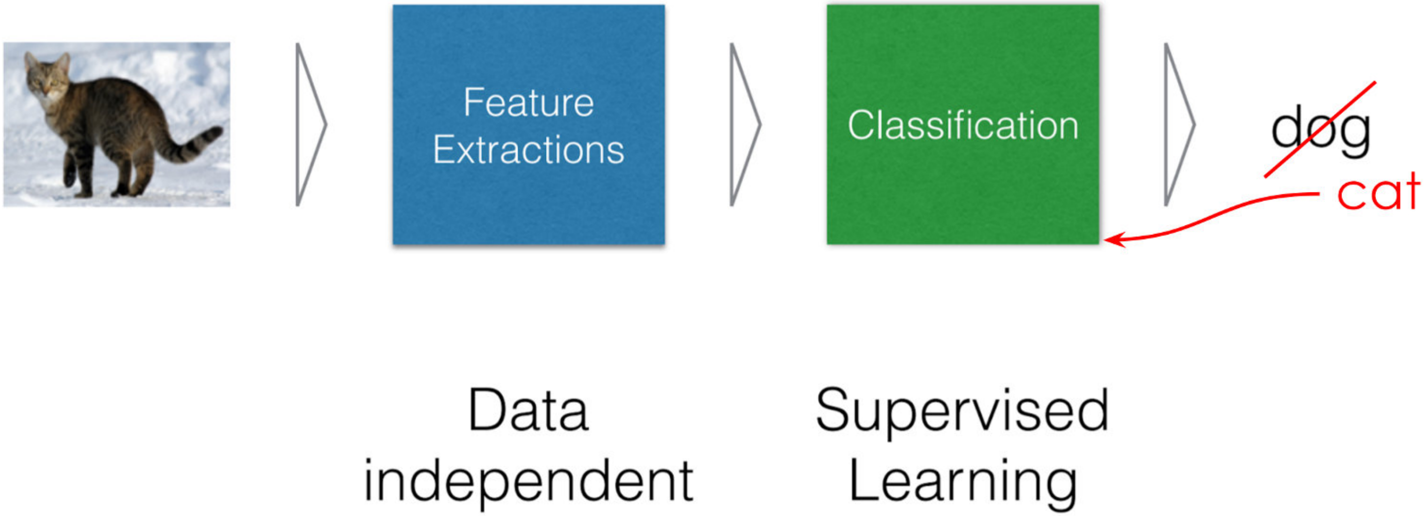
Learn from mistakes

Brain Learning Process

- Continuously create, strengthen, and weaken connections between neurons based on experimental results
- i.e., adjust the weight of the connection:

Machine Learning Process

- An error occurred, adjusting model parameters backwards



Feature Extractions

Classification

~~dog~~

cat

Data independent

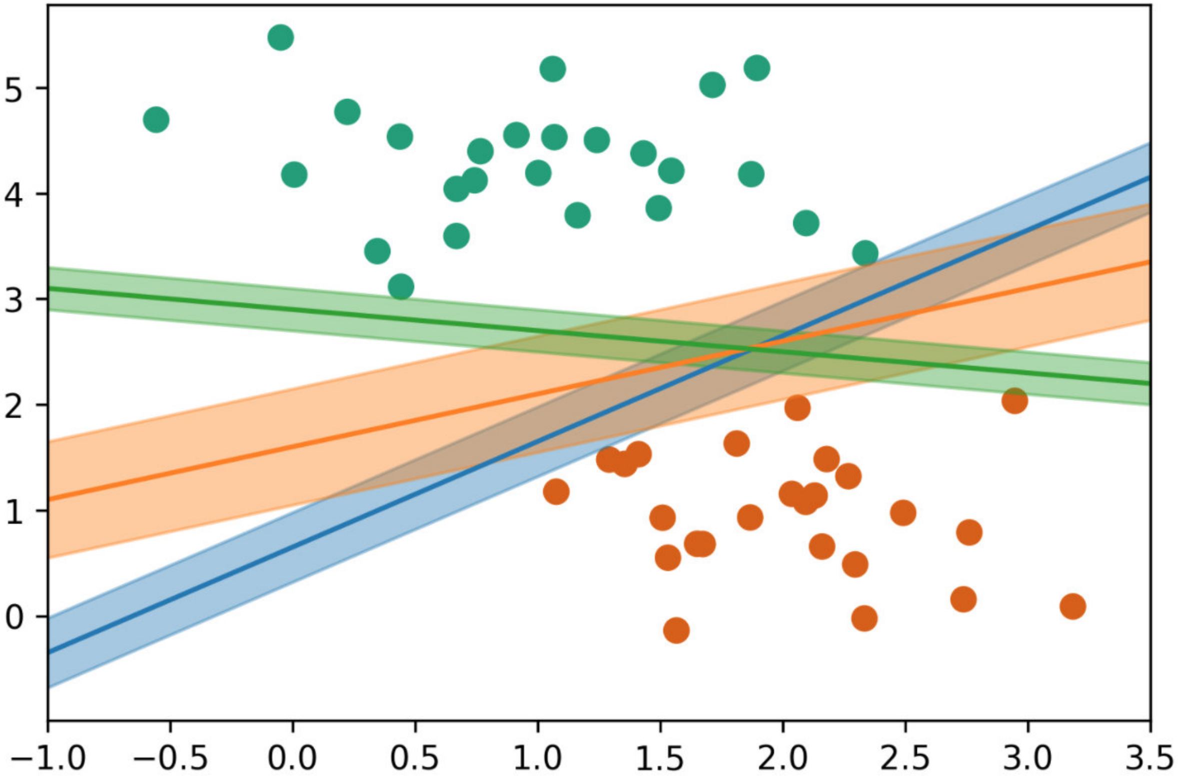
Supervised Learning

Perceptron Learning Process

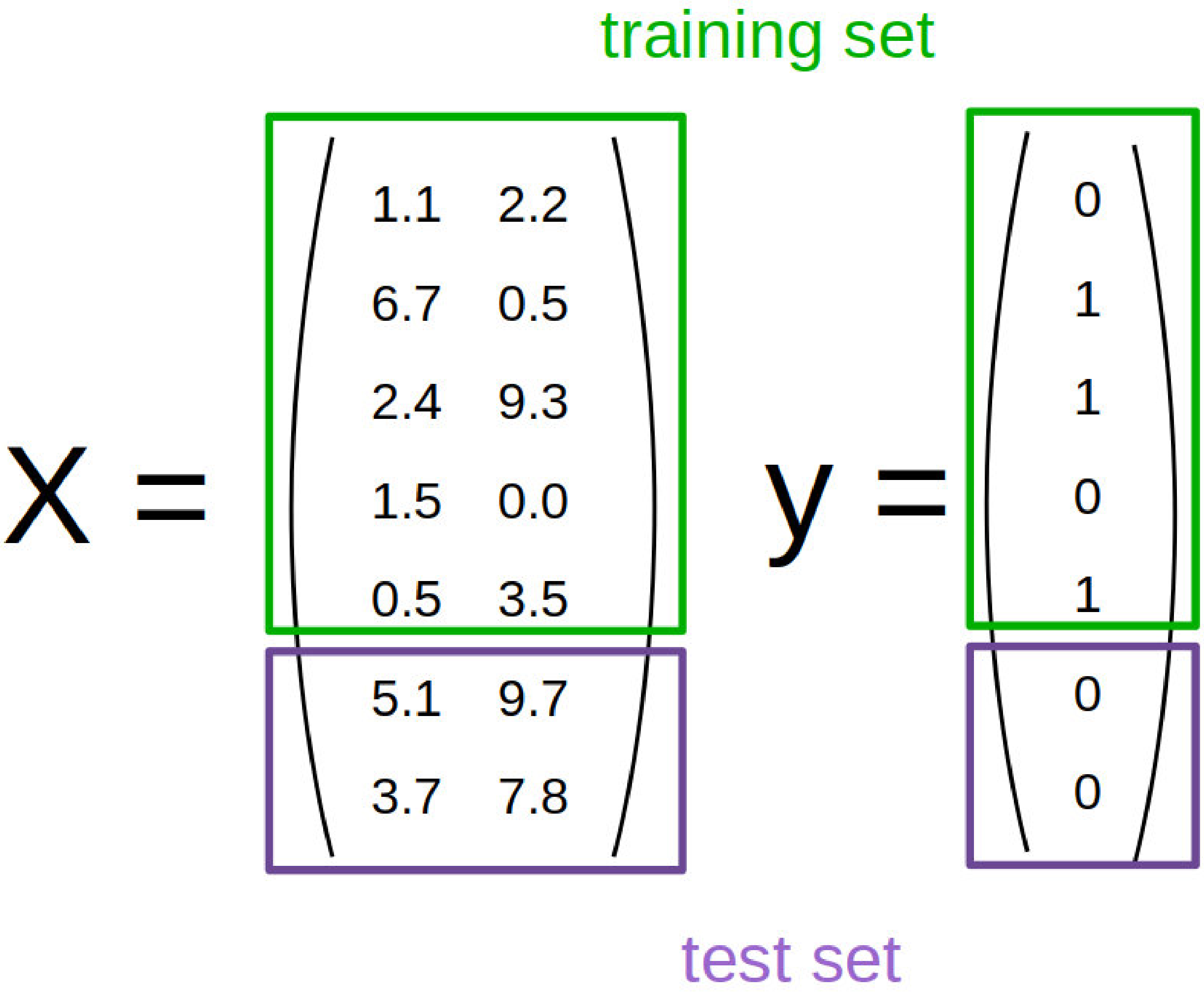
- Find errors, adjust weight to reduce errors

Perceptron Learning Process

- Find errors, adjust , adjust decision boundaries



Learning Process



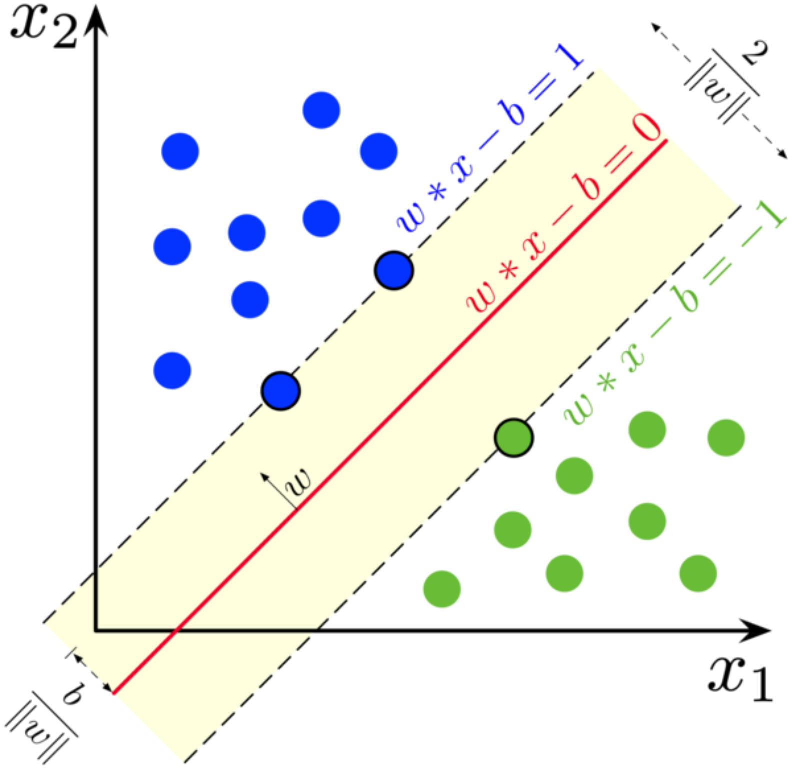
training set

$$X = \begin{pmatrix} 1.1 & 2.2 \\ 6.7 & 0.5 \\ 2.4 & 9.3 \\ 1.5 & 0.0 \\ 0.5 & 3.5 \\ 5.1 & 9.7 \\ 3.7 & 7.8 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

test set

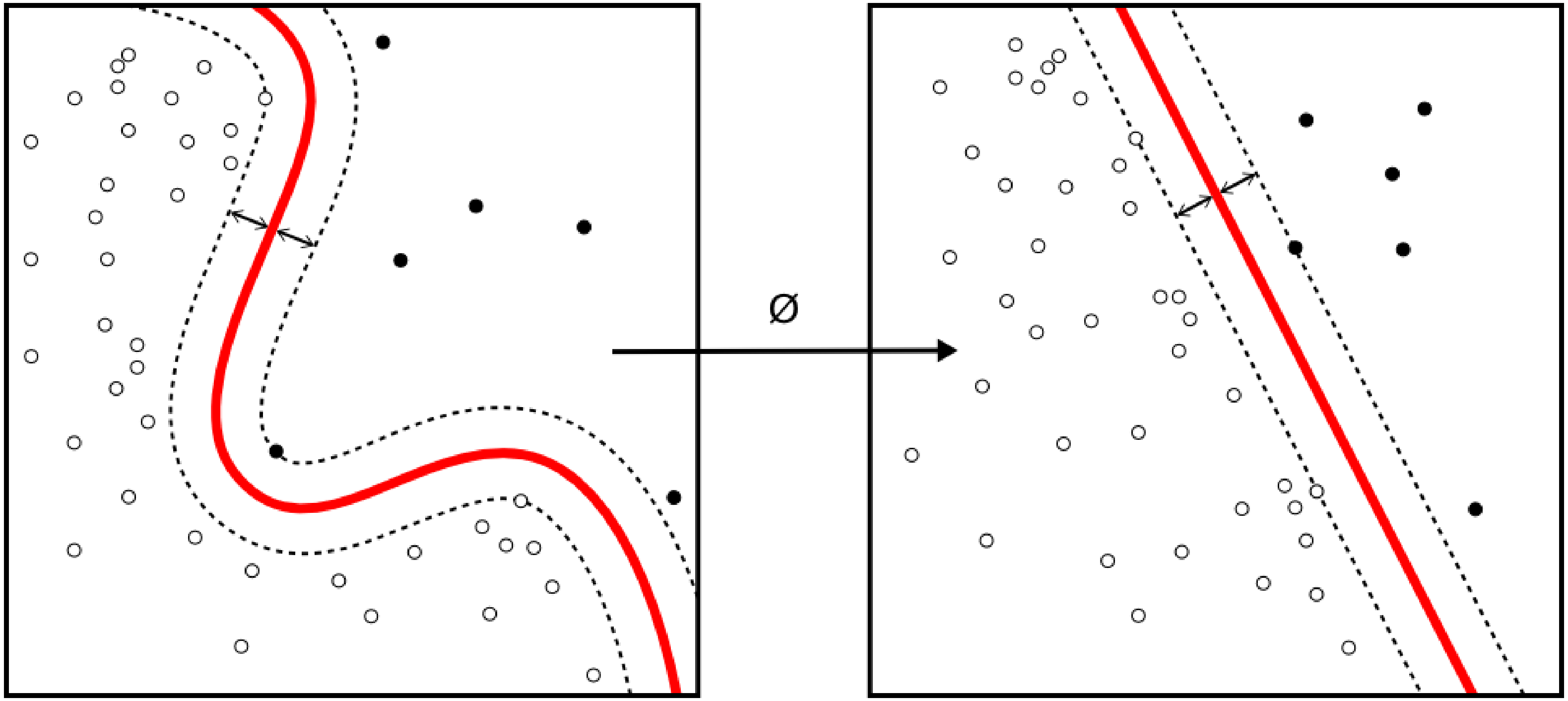
SVM

- Support Vector Machines
- Not only avoid mistakes, the farther the two sides are, the better



Kernel

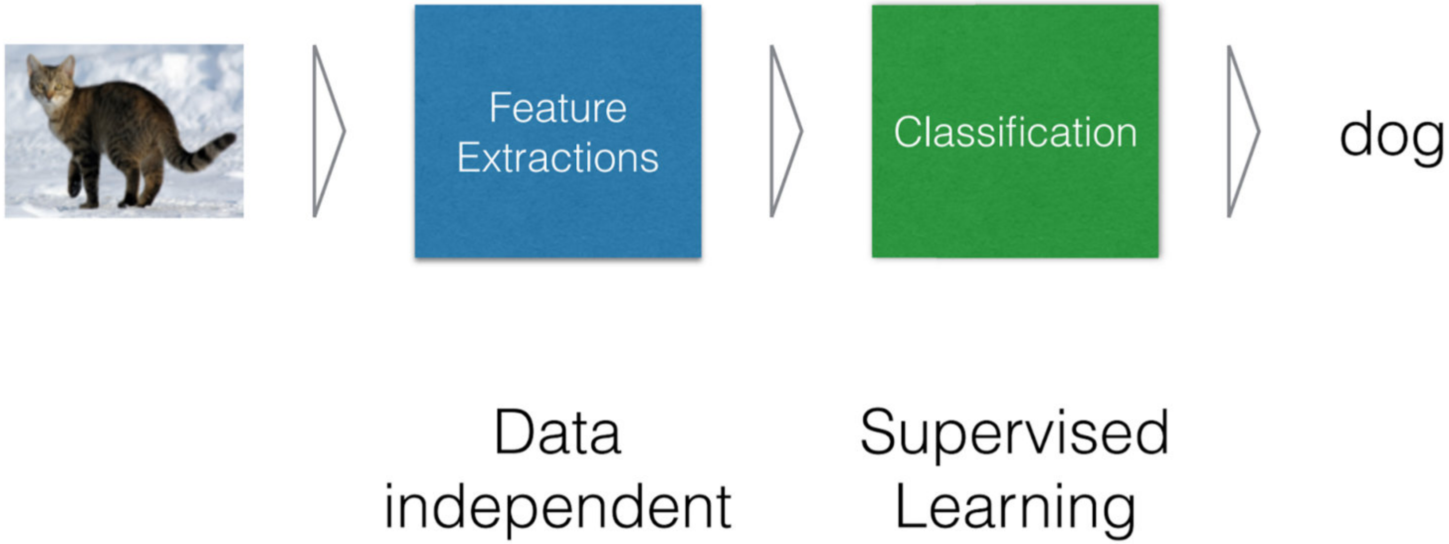
Use a non-linear kernel function instead of a vector dot product to support curve boundaries



Deep Learning

Machine Learning

- First extract image features
- Then learn based on these features



Feature Extractions

Classification

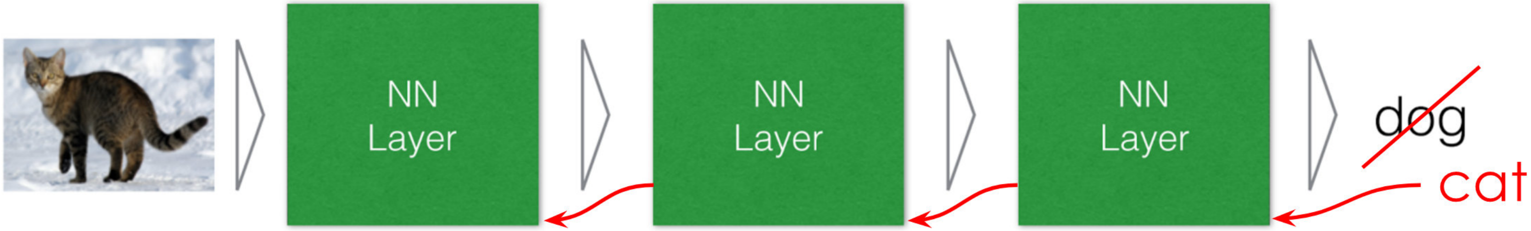
dog

Data independent

Supervised Learning

Deep Learning

- No specific feature extraction step
- Send the raw data directly to the multilayer neural network for learning
- Error occurred, adjust the model parameters



Supervised Learning

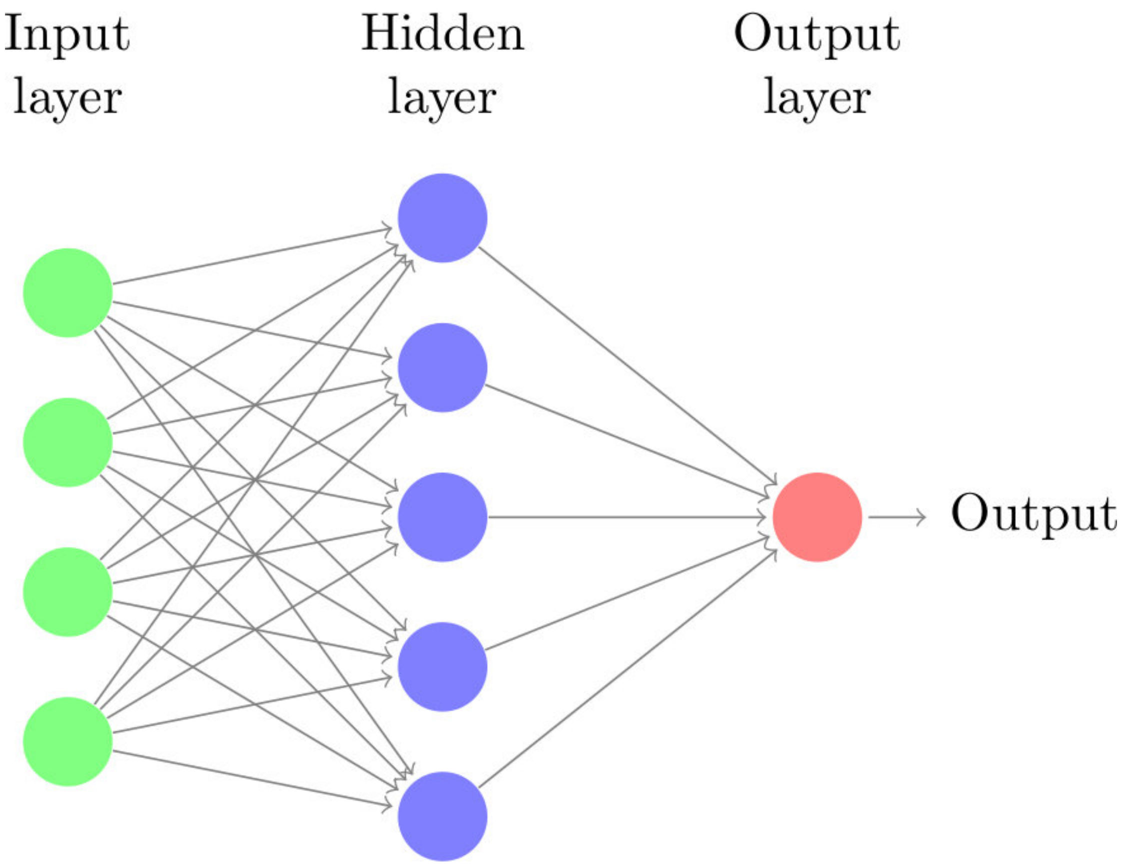
Typical Neural Network Structures

FFN、CNN、RNN

Forward Neural Network

FFN

Forward Neural Network

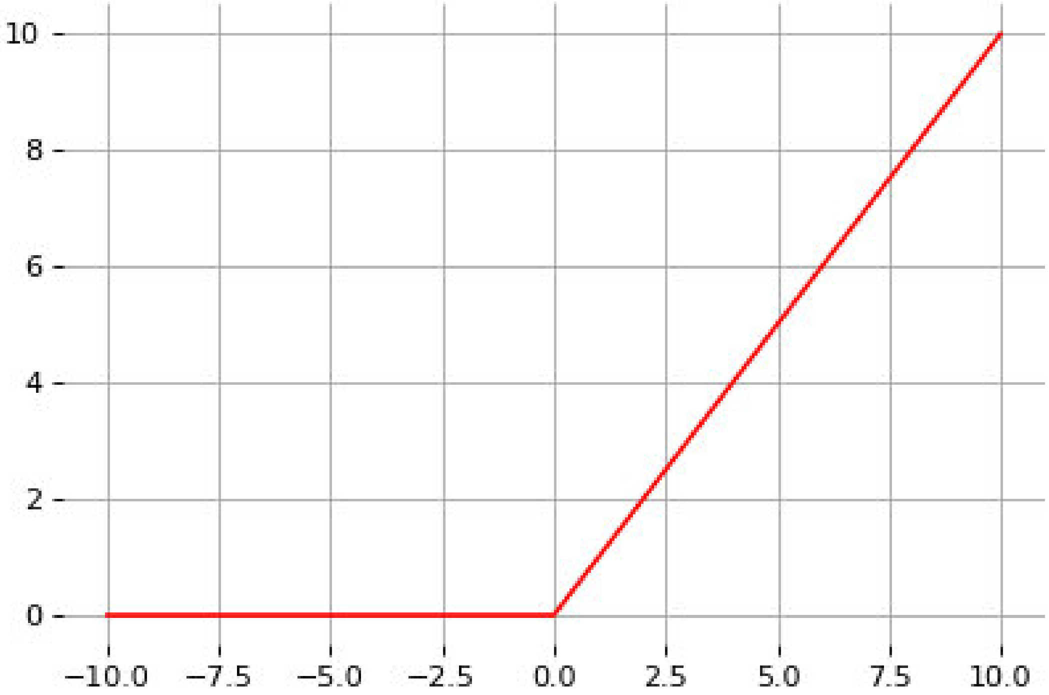


Hidden and output layer units: perceptron

Activation Function

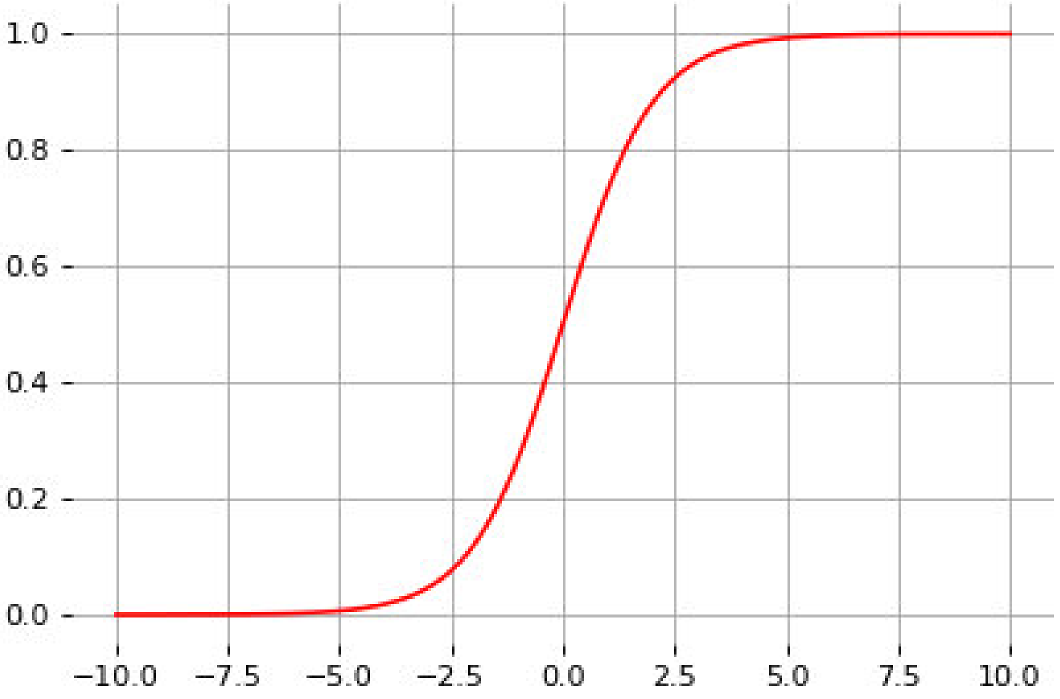
- ReLU: Rectified Linear Unit
- Sigmoid
- Tanh: Hyperboilic Tangent

ReLU



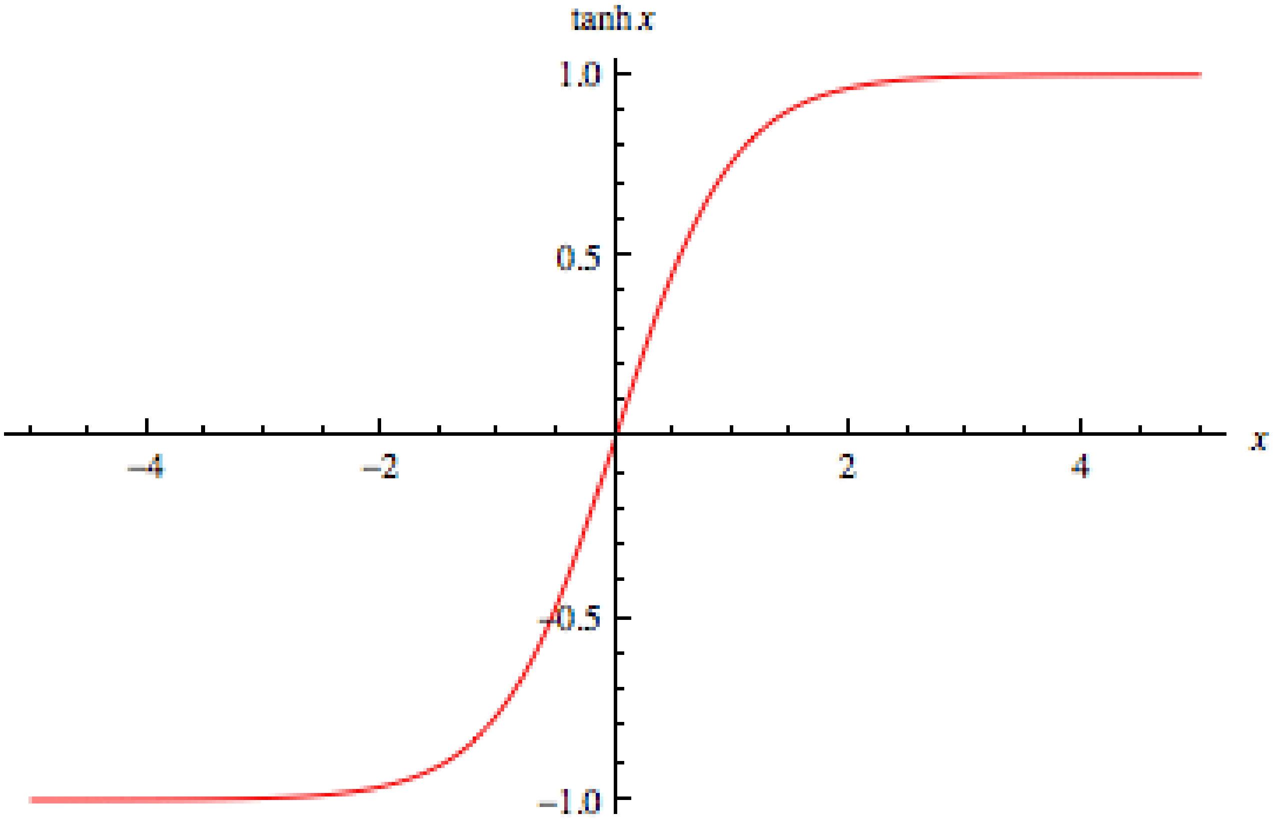
: Rectified Linear Unit

Sigmoid



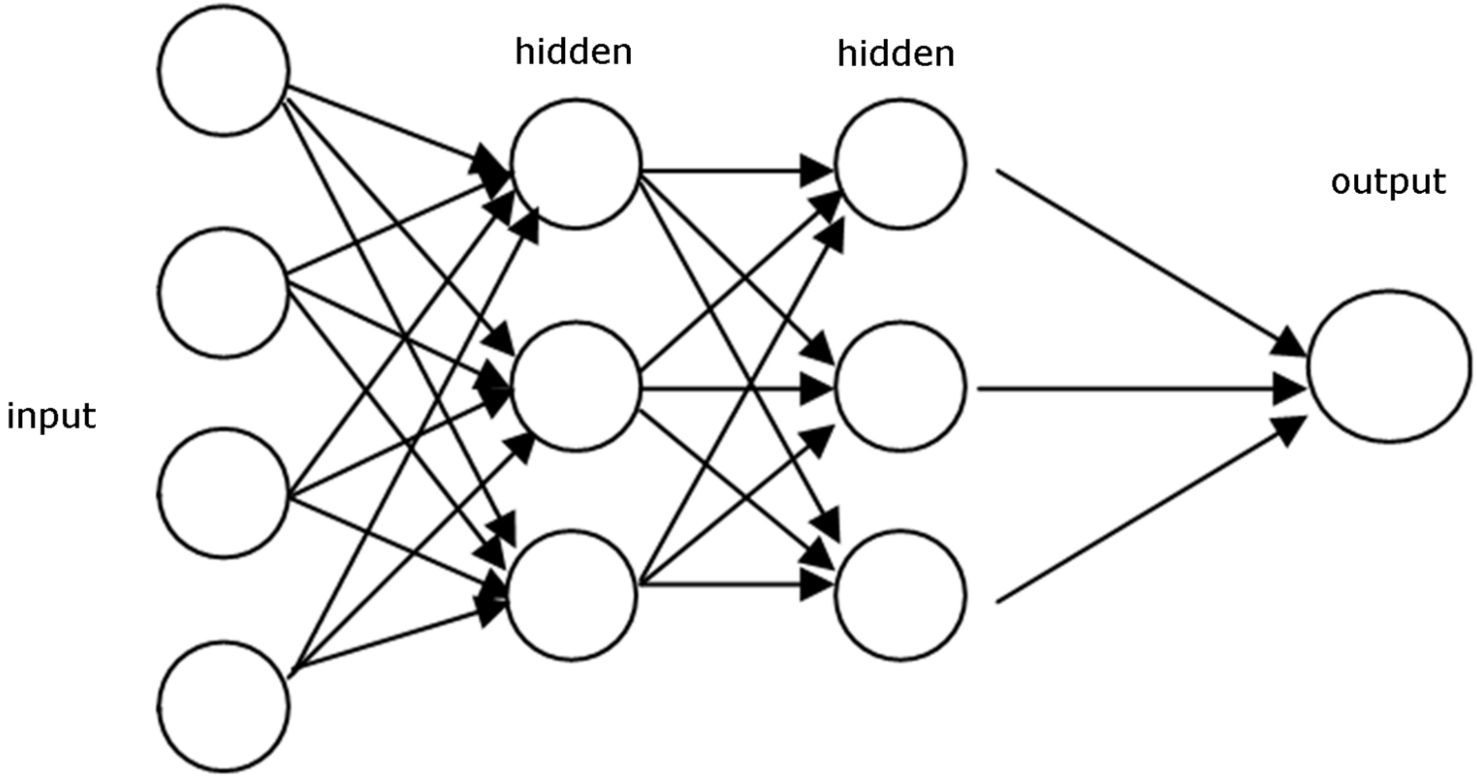
: S曲线

Tanh



: Hyperboilic Tangent

Deep Neural Network



Multiple hidden layers

Benefits of Depth

Generally, the deeper, the stronger the model

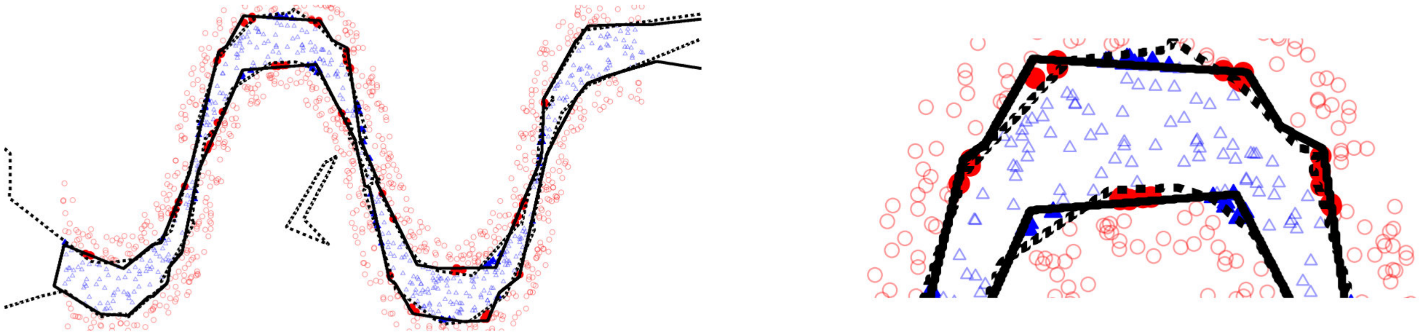
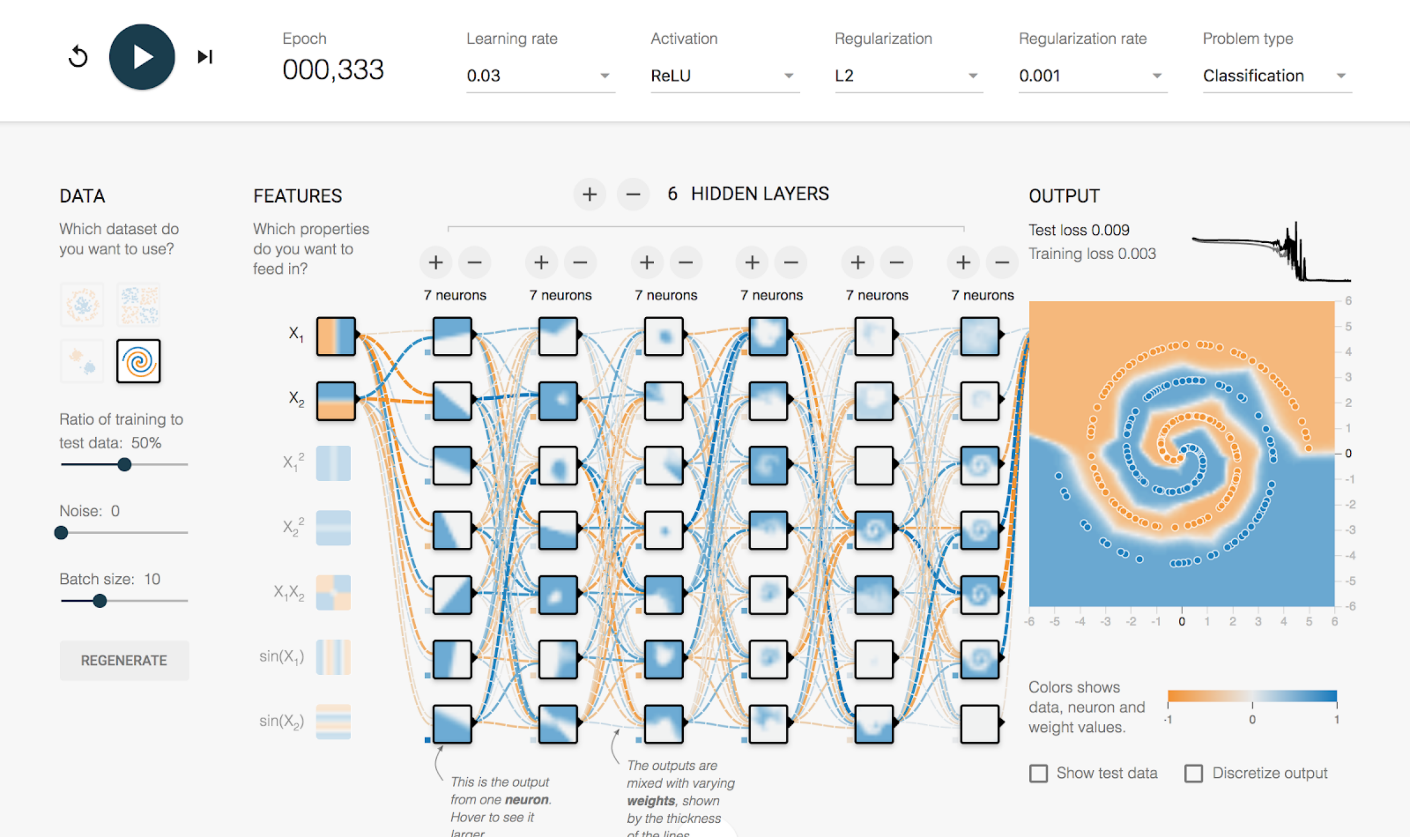


Figure 1: Binary classification using a shallow model with 20 hidden units (solid line) and a deep model with two layers of 10 units each (dashed line). The right panel shows a close-up of the left panel. Filled markers indicate errors made by the shallow model.

FNN Experiments

- Browser-based TensorFlow experiments
- http://playground.tensorflow.org



Epoch 000,333

Learning rate 0.03

Activation ReLU

Regularization L2

Regularization rate 0.001

Problem type Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

FEATURES

Which properties do you want to feed in?

6 HIDDEN LAYERS

OUTPUT

Test loss 0.009

Training loss 0.003

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

This is the output from one **neuron**. Hover to see it larger

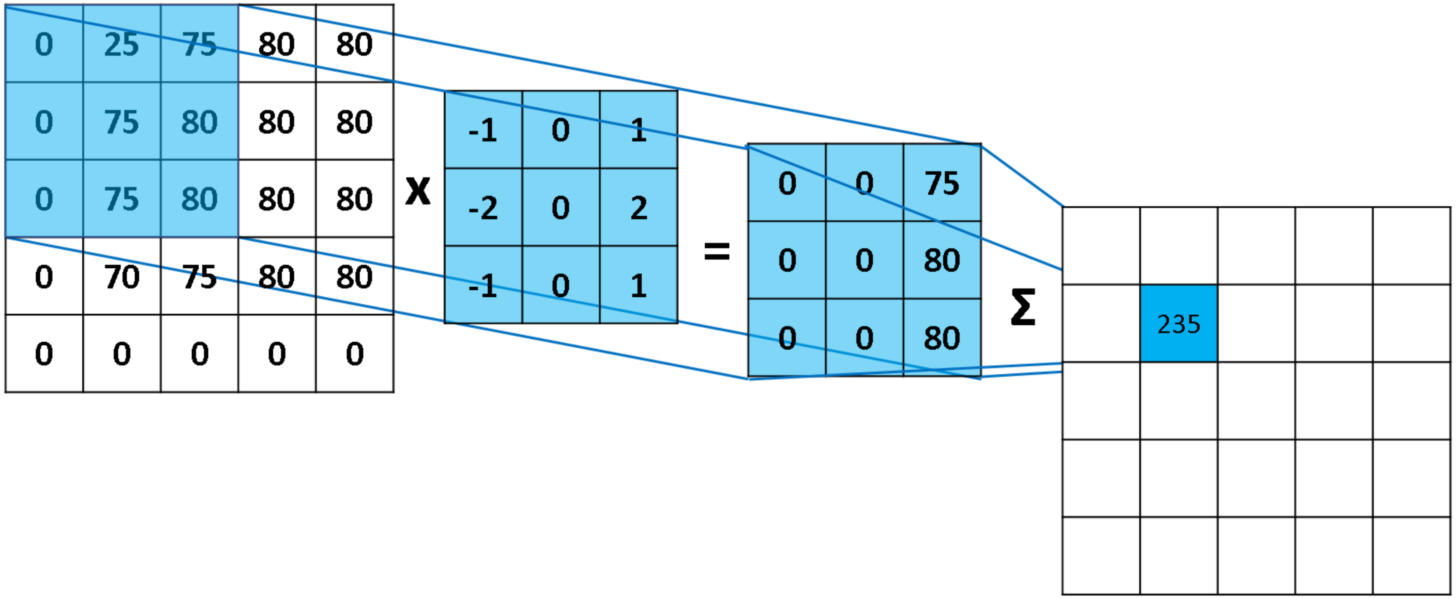
The outputs are mixed with varying **weights**, shown by the thickness of the lines

CNN

Convolutional Neural Network

2D Convolution

Multiply corresponding positions, then add



0	25	75	80	80
0	75	80	80	80
0	75	80	80	80
0	70	75	80	80
0	0	0	0	0

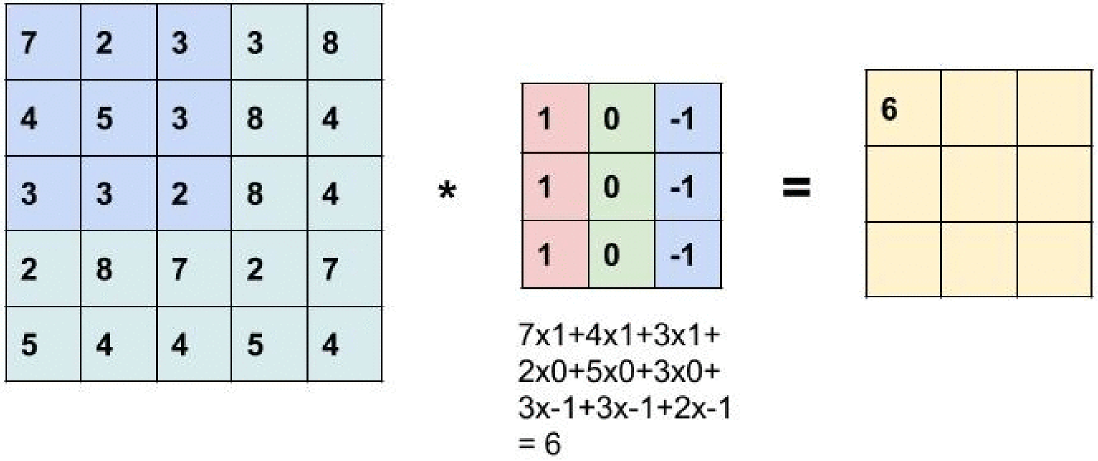
-1	0	1
-2	0	2
-1	0	1

0	0	75
0	0	80
0	0	80

	235			

Image Convolution

The filter slides on the picture for convolution.



7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

1	0	-1
1	0	-1
1	0	-1

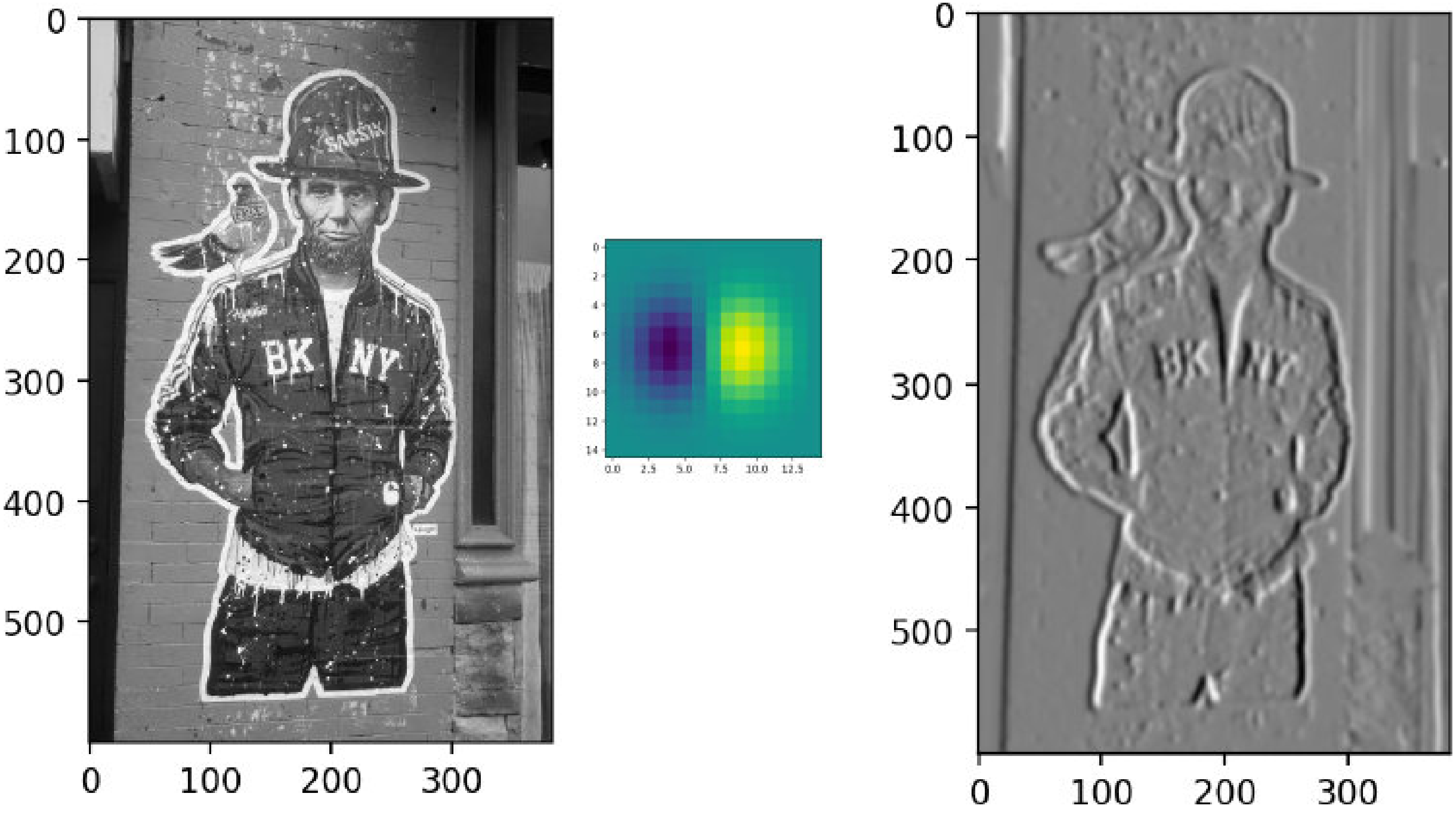
=

6		

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

Convolution Pixel Gradient

Select appropriate convolution kernel (filter) to calculate the pixel gradient of the image

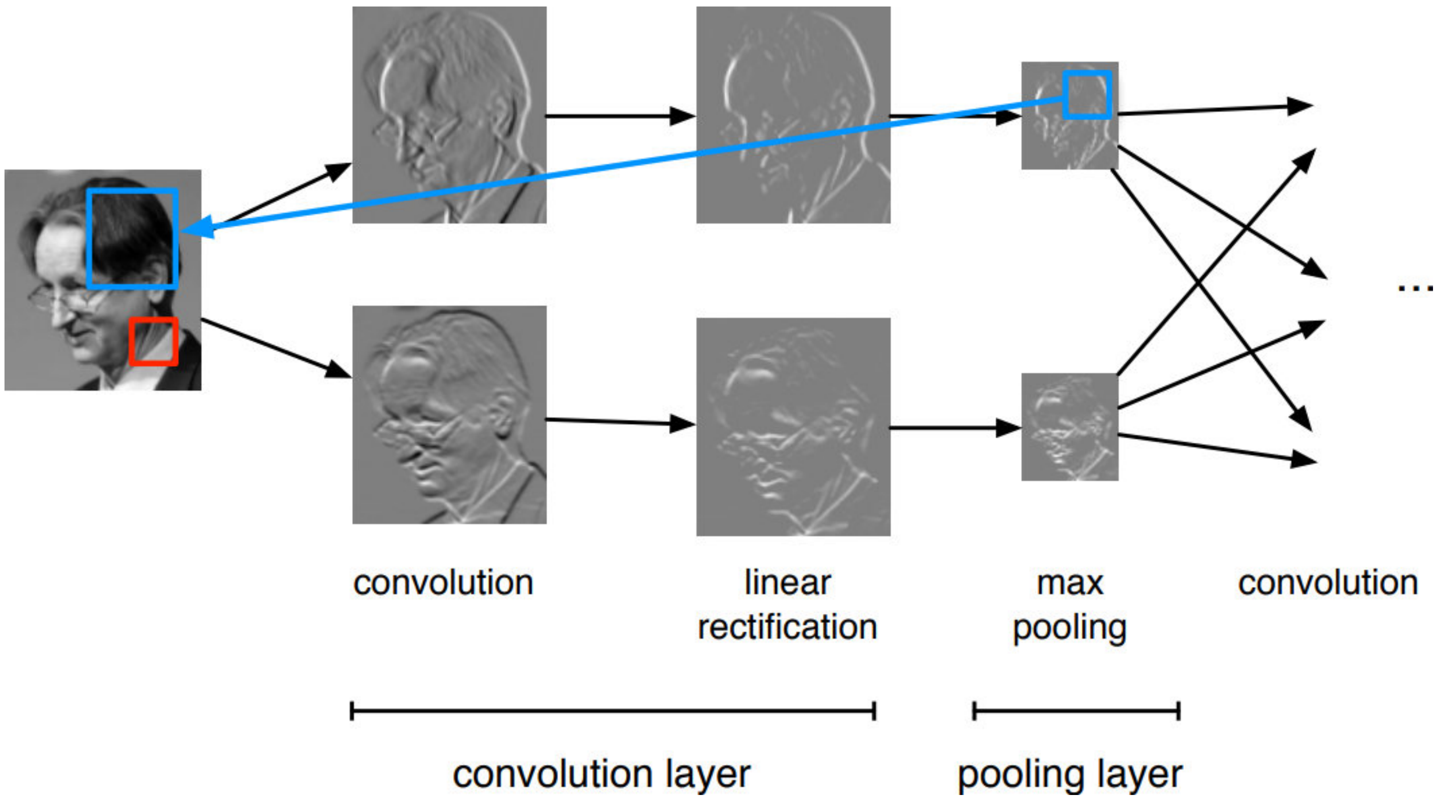


Convolutional Neural Network

- A special multilayer forward neuron network
- Origin: Handwriting Recognition
- Commonly used in image and vision applications, text processing

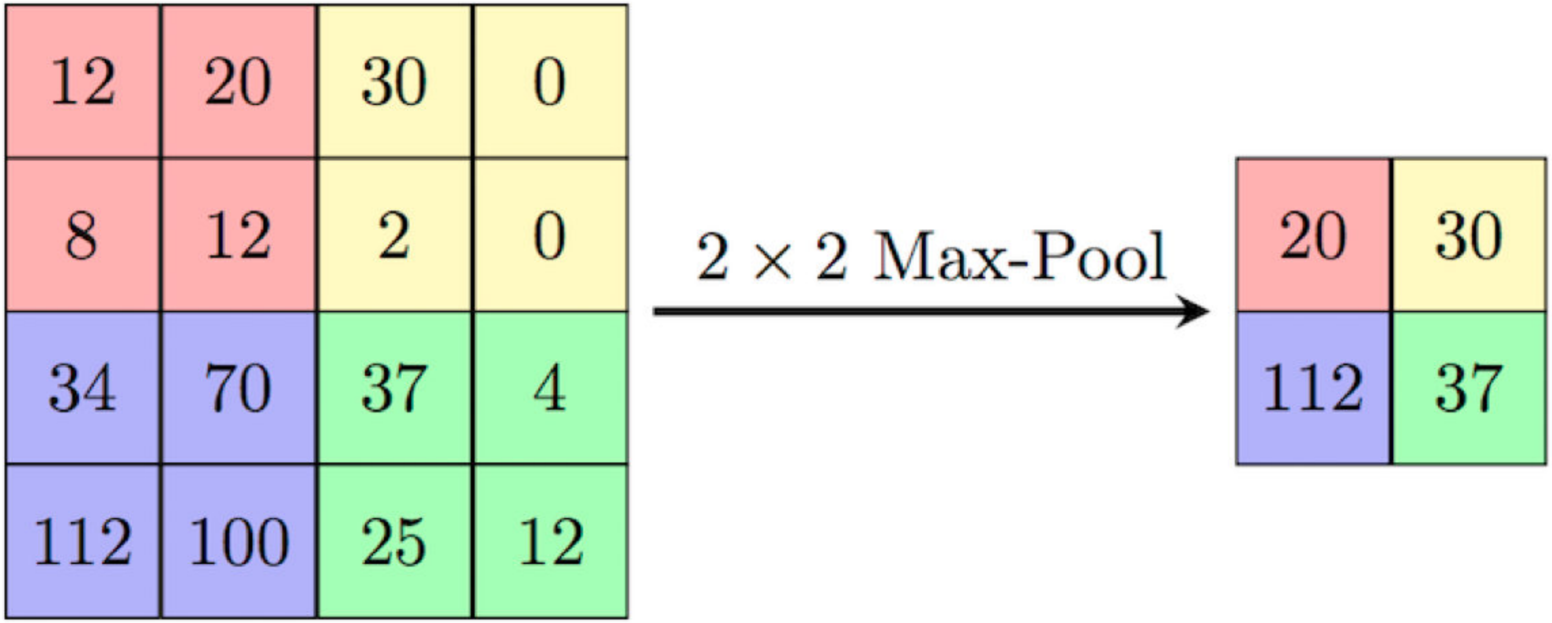
Architecture

- Convolutional layer
  - Convolution + non-linear activation function (such as ReLU)
- Pooling layer



Pooling Layer

Sampling reduces the amount of data



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

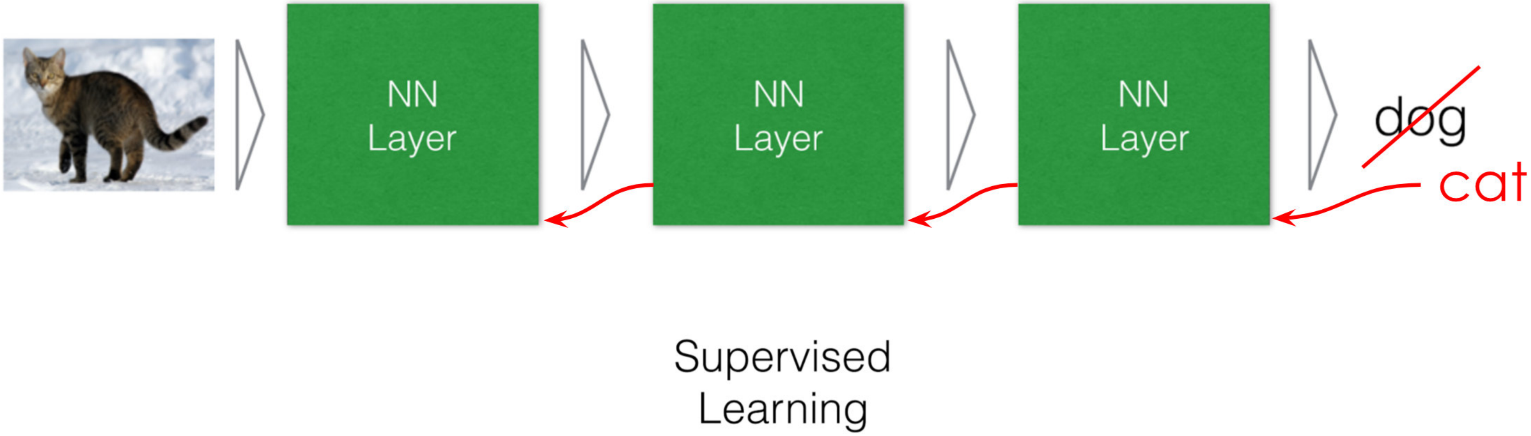
2 × 2 Max-Pool

20	30
112	37

Max Pooling

Deep CNN

- Send the raw data directly to the multilayer neural network for learning
- Multiple convolution and pooling layers
- An error occurred, adjusting the convolution kernel all the way



LeNet

- Handwriting recognition
- 1988, LeCun

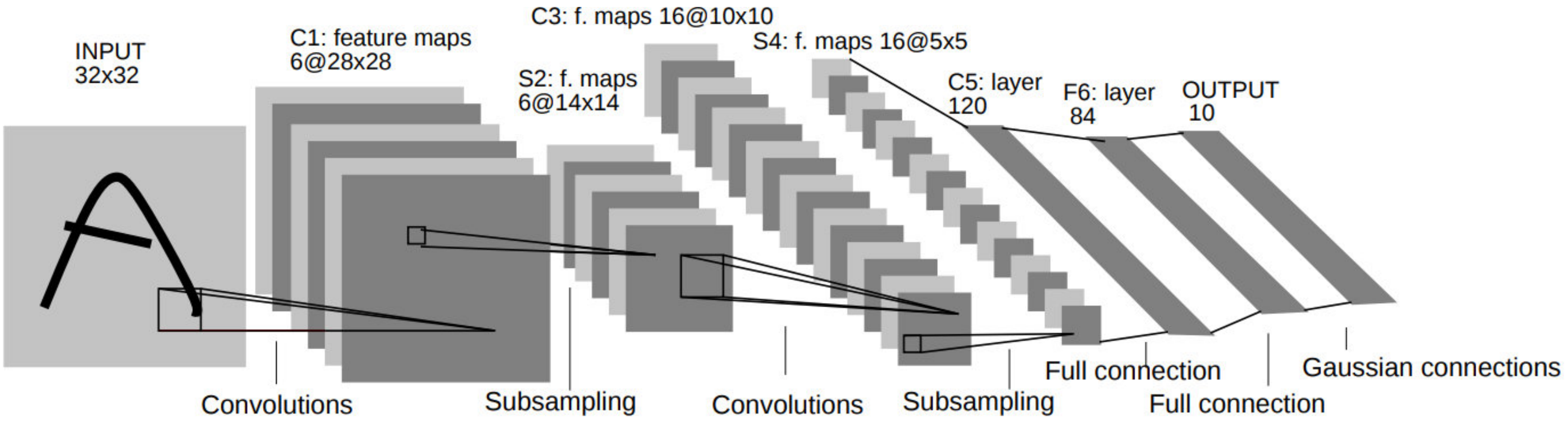


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Image Processing Result

After the first layer of convolution and pooling

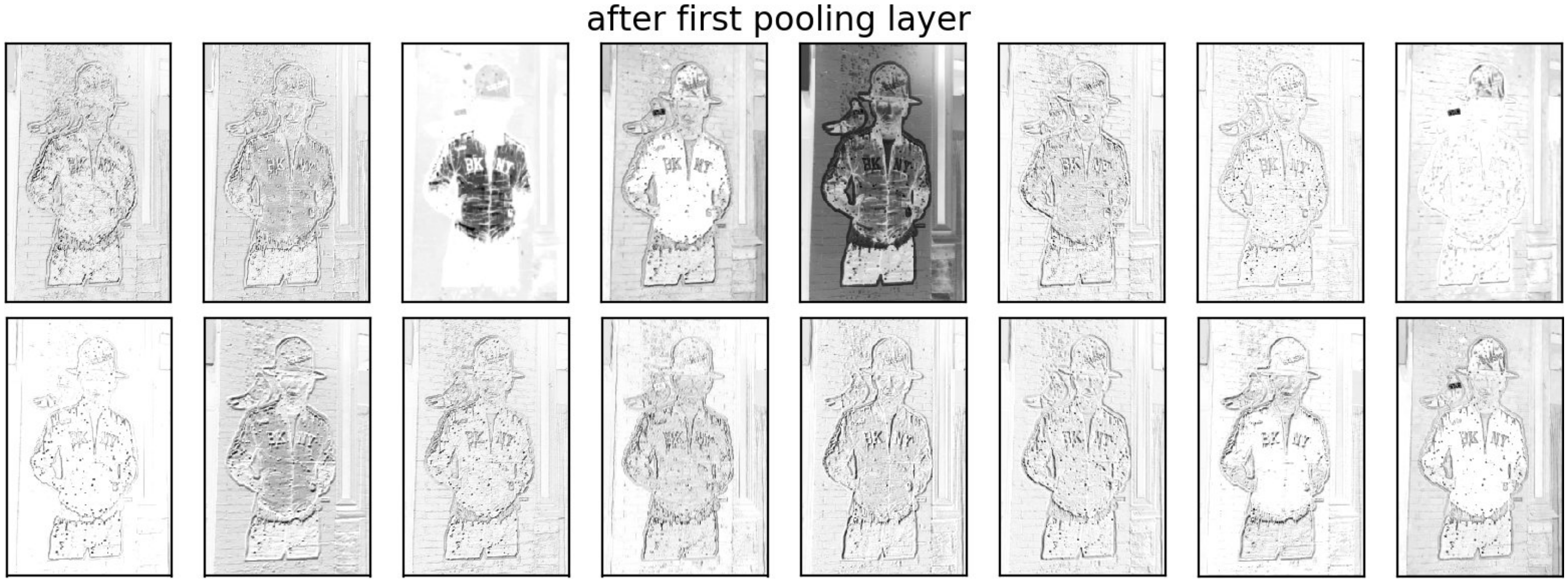
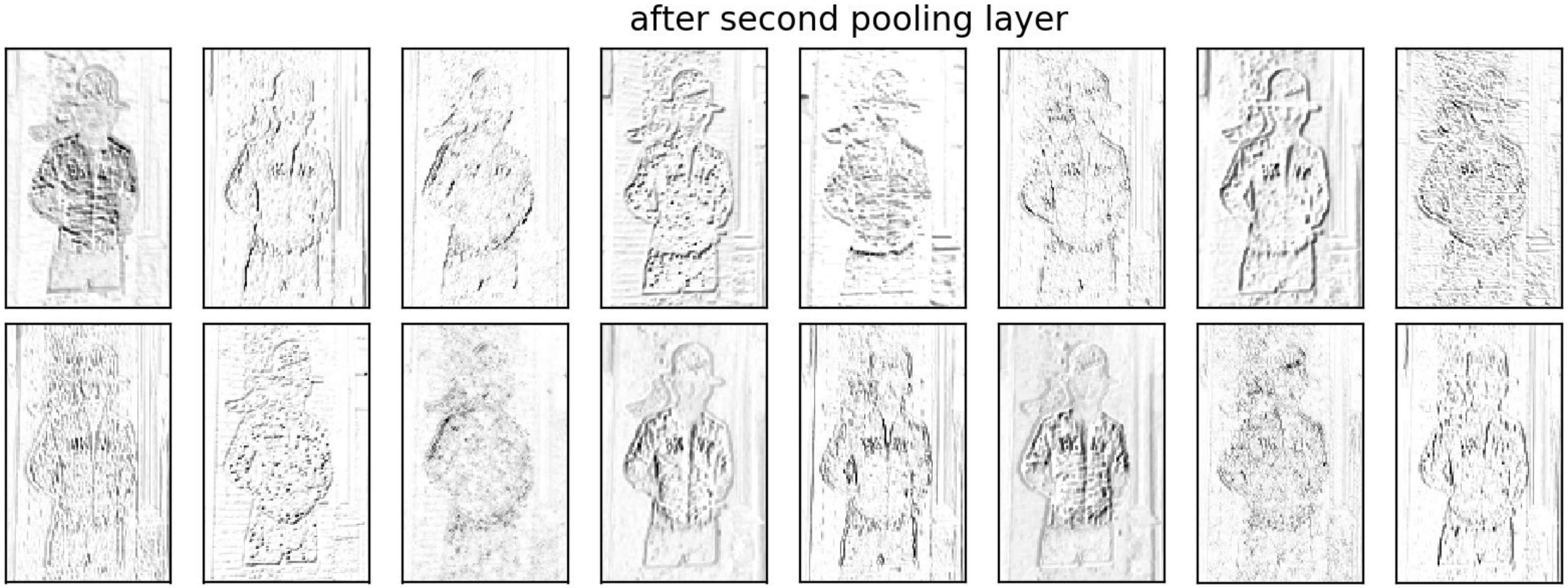


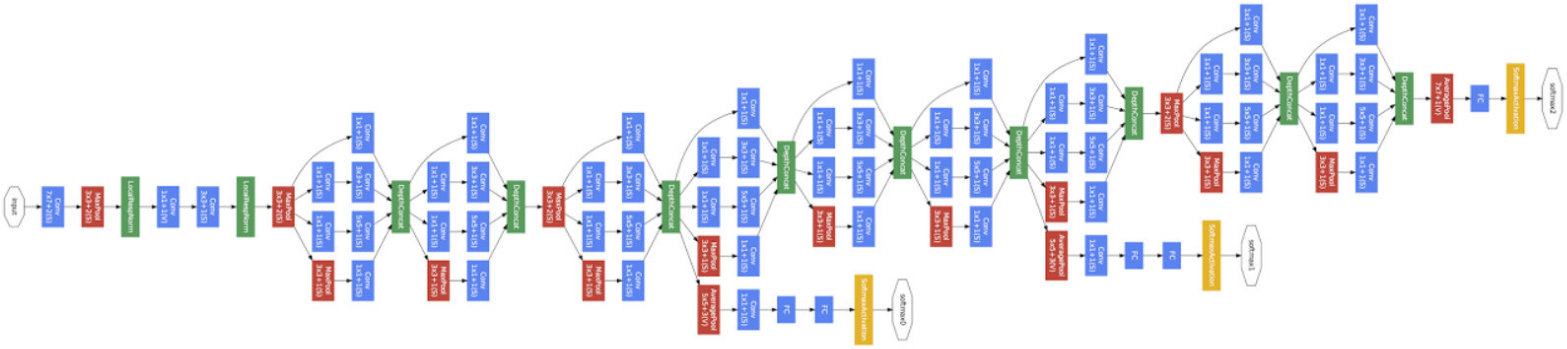
Image Processing Result

After the second layer of convolution and pooling



Deep CNN

- Many layers
- Tens of millions of pixels
- Tens of millions of parameters need to be calculated and adjusted



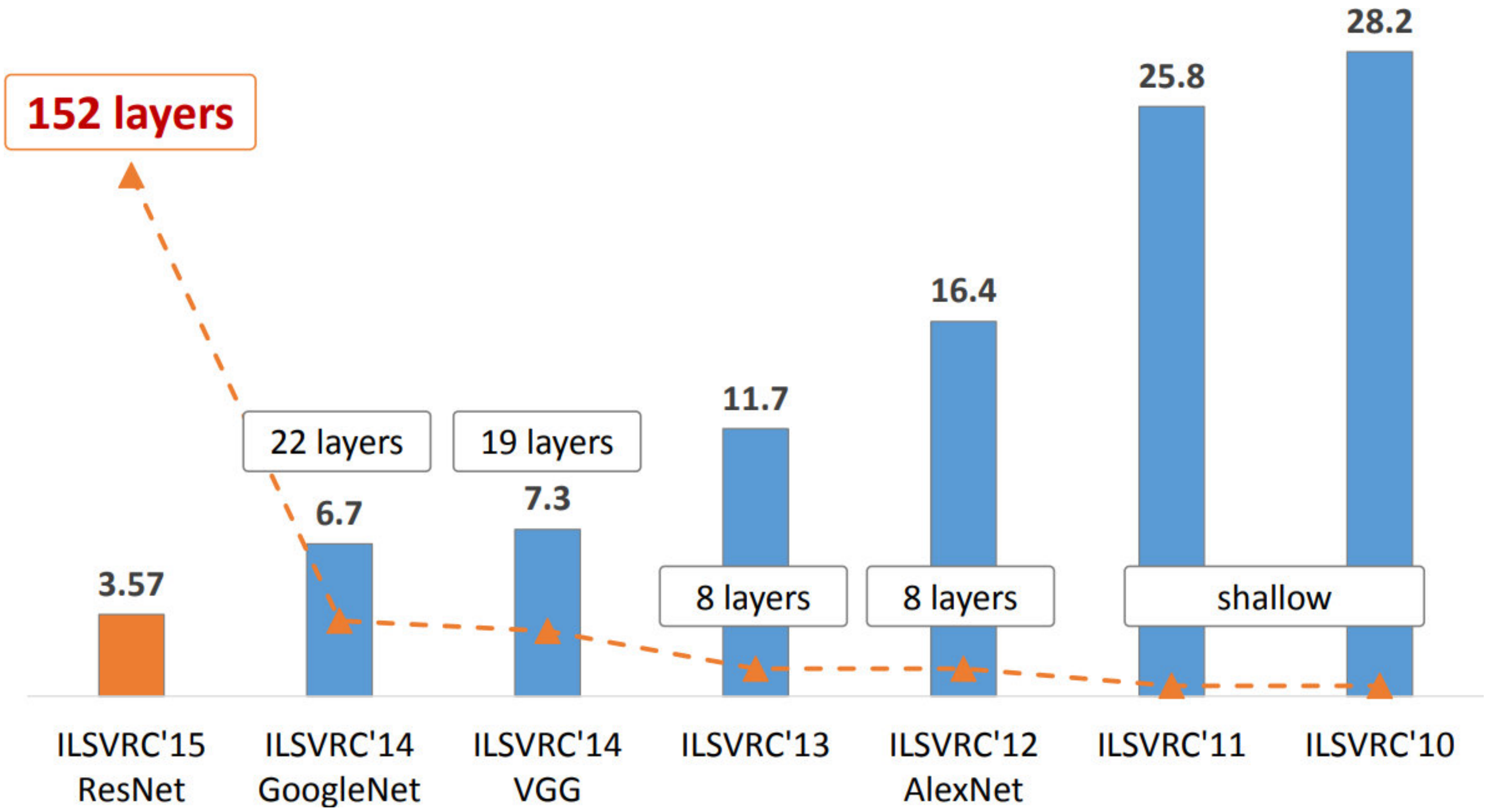
GoogleNet

GPU

- Parallel computing with thousands of computing units in GPU

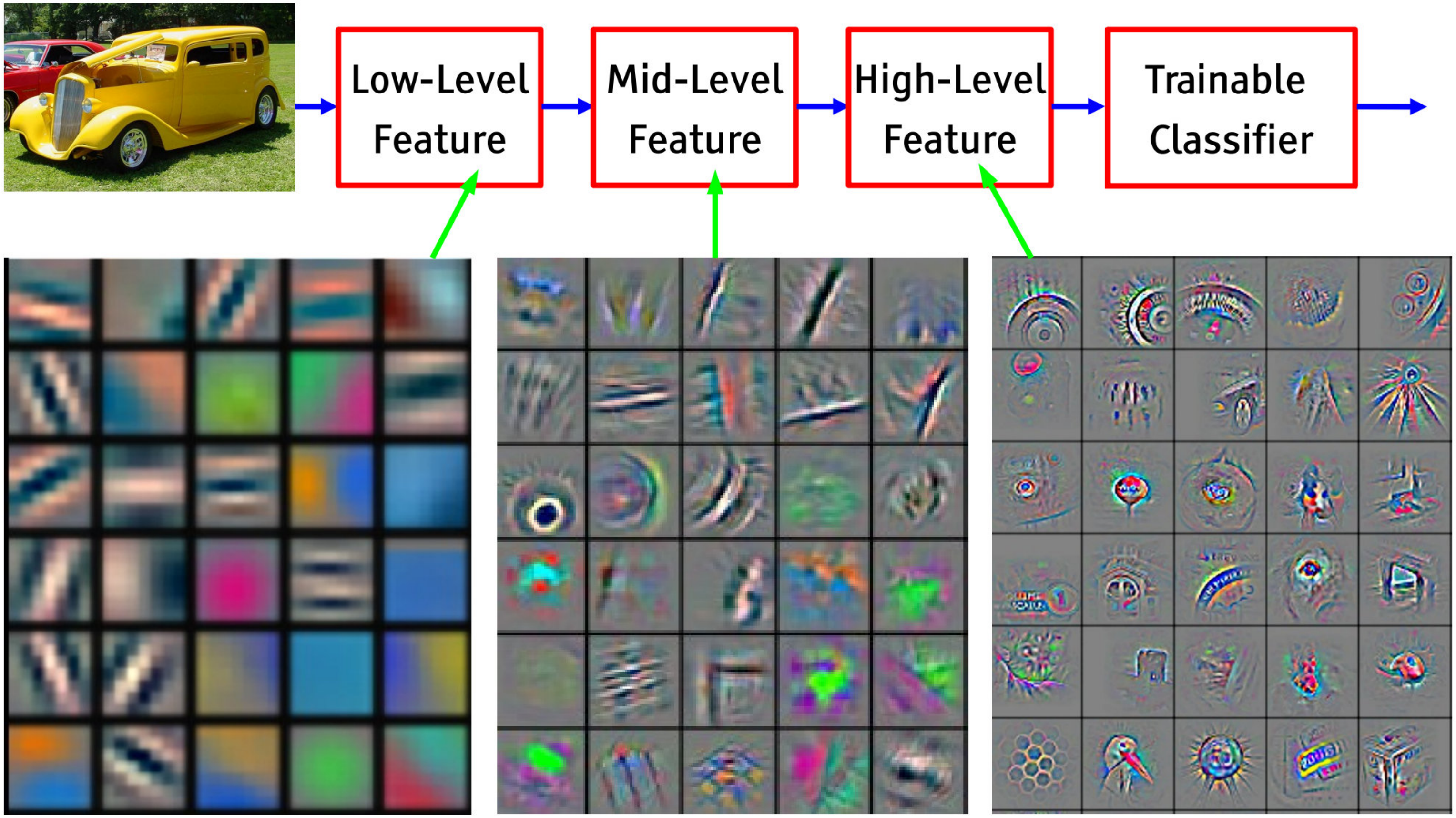
Great Performance Gain

ImageNet object recognition image dataset



Understanding of CNN

- Extract simple features at the bottom and complex features at the high level



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

CNN Demo

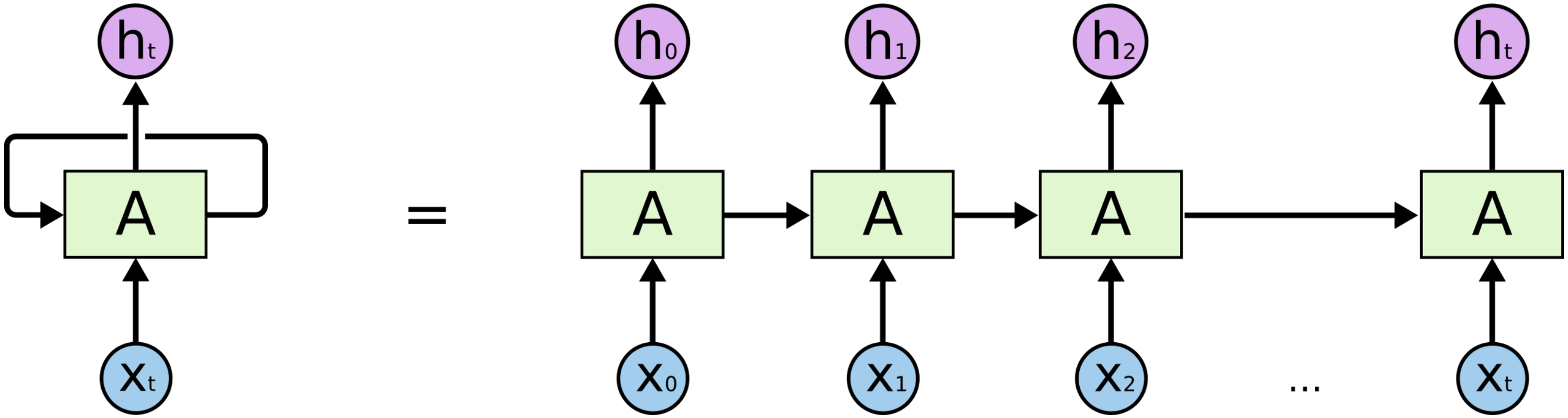
- Andrej Karpathy ConvNetJS
- https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html
- Train CNN in browser, experiment with MNIST handwriting recognition task

RNN

Recurrent Neural Network

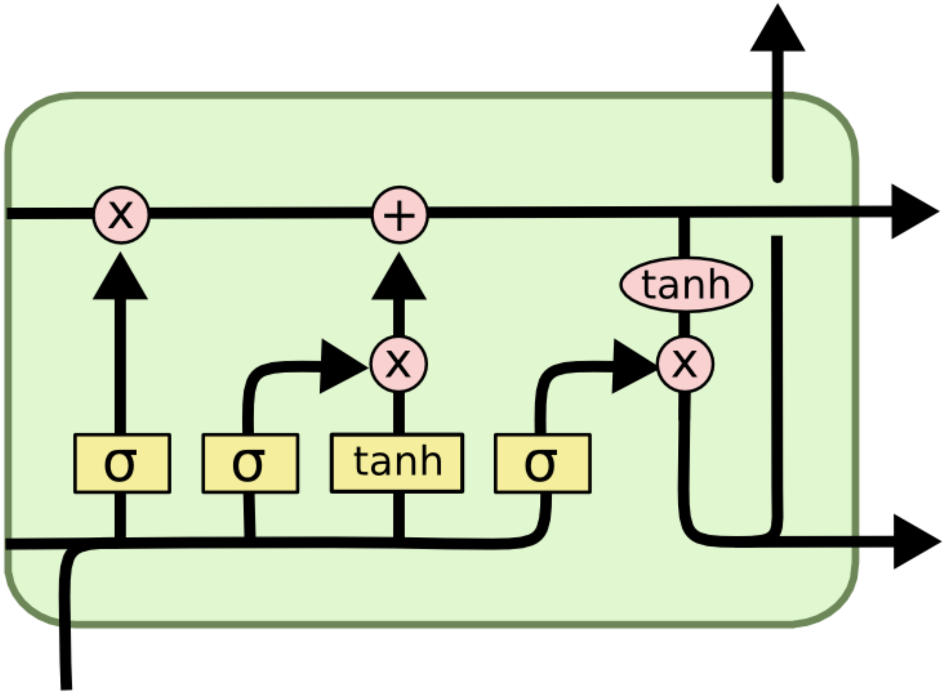
RNN

- “Memory unit”
- Suitable for processing time series data and natural language processing (NLP) tasks
- Sequence input



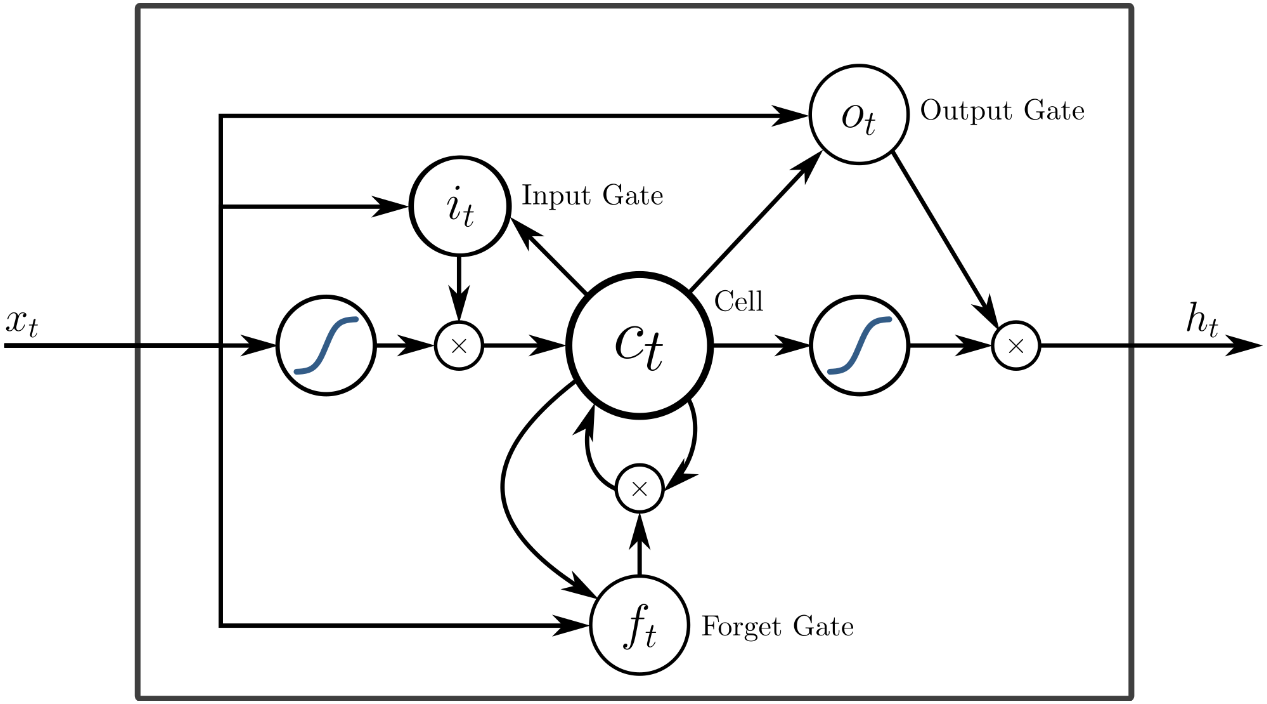
LSTM

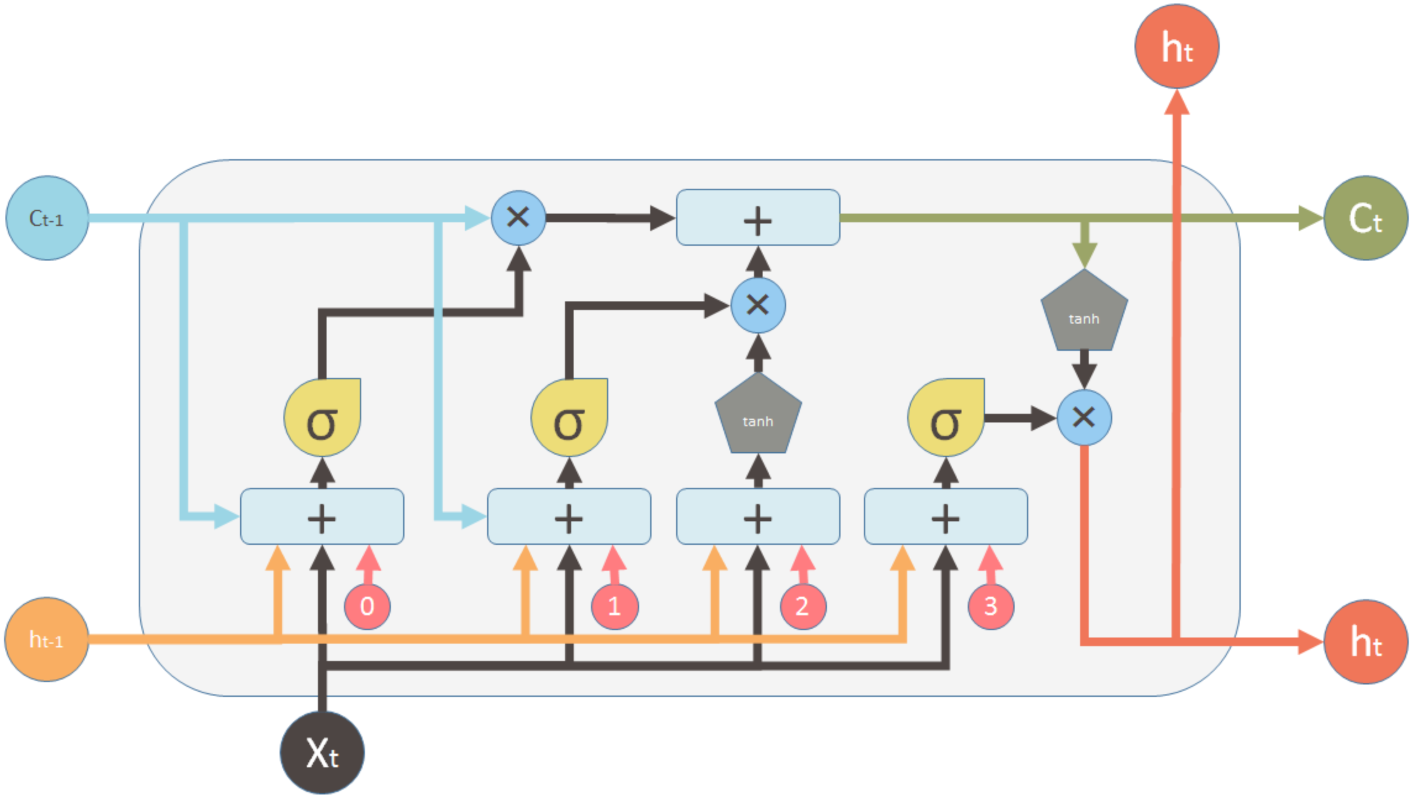
Long short-term memory unit



LSTM

- The human brain forgets
- Input gate, output gate, forget gate





Inputs:

X_t Input vector

C_{t-1} Memory from previous block

h_{t-1} Output of previous block

outputs:

C_t Memory from current block

h_t Output of current block

Nonlinearities:

σ Sigmoid

tanh Hyperbolic tangent

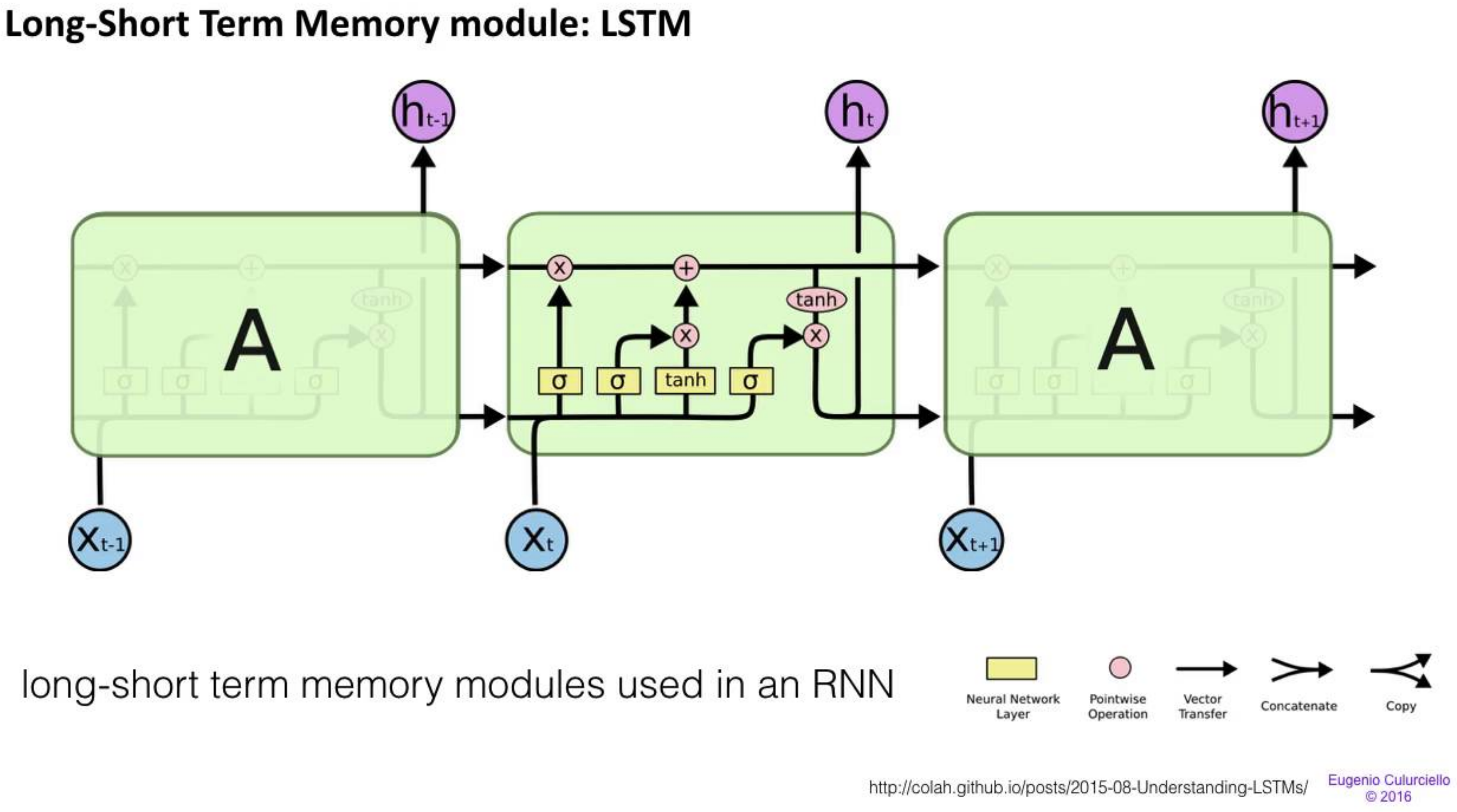
Bias: 0

Vector operations:

× Element-wise multiplication

+ Element-wise Summation / Concatenation

LSTM-based RNN



Long-Short Term Memory module: LSTM

long-short term memory modules used in an RNN

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Eugenio Culurciello
© 2016

Wide Application of RNN

1. Speech recognition
2. Machine translation
3. Text generation
4. Recommendation system
5. Time series prediction

Summary: Deep Learning Models

1. Forward neural network（FFN)
2. Convolutional neural network（CNN)
3. Recurrent neuron network（RNN)

Progress

Overview

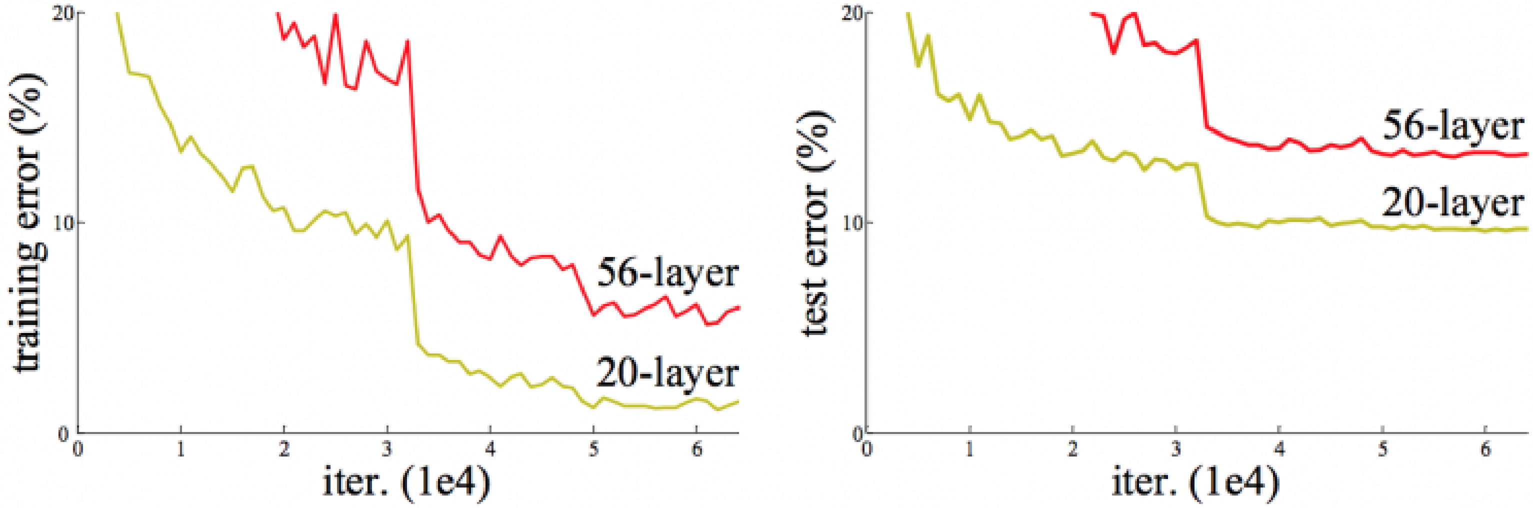
- There are many different types of neural networks
- Each neural network can be used to solve specific AI problems
- This field is growing rapidly
    - Ian Goodfellow invented GAN in 2014
    - Capsule network

ResNet

Residual network

ResNet

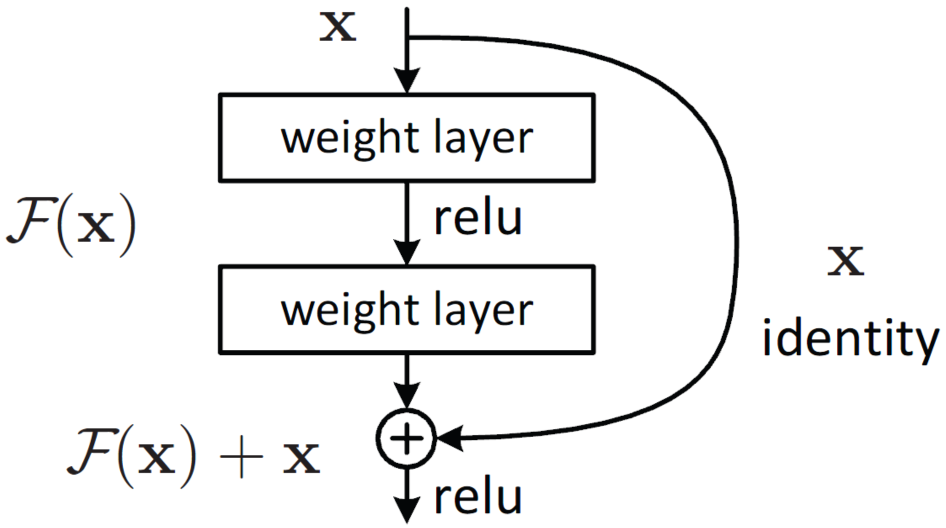
In general, for deep neural networks, after the number of layers exceeds a certain value, the more layers, the more difficult it is to optimize, and the performance becomes worse.



CIFA-10 dataset

ResNet

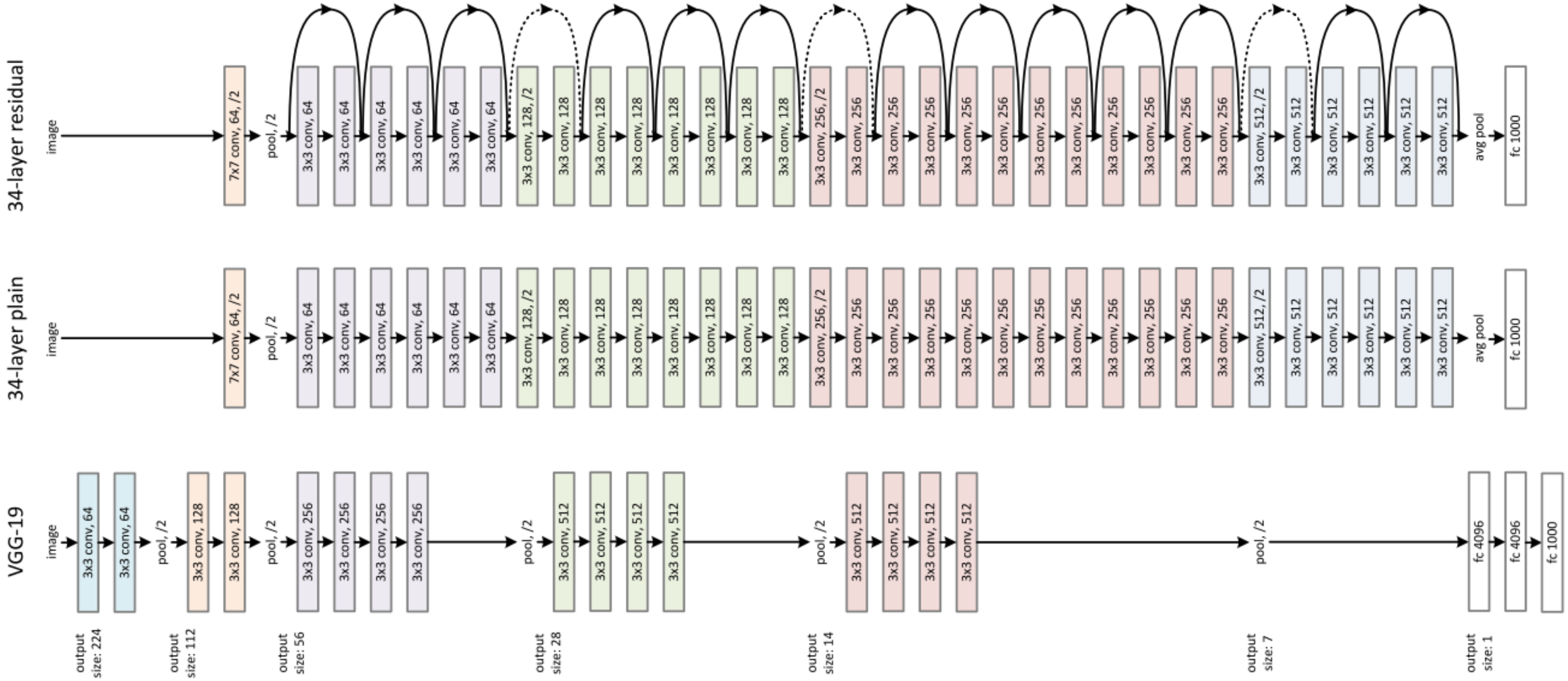
- Residual network
- Add direct link



Residual Network

ResNet

Support deeper networks for better performance



Attention

Attention mechanism

Attention

- Human's attention is not average
- Give different elements different attention to improve performance

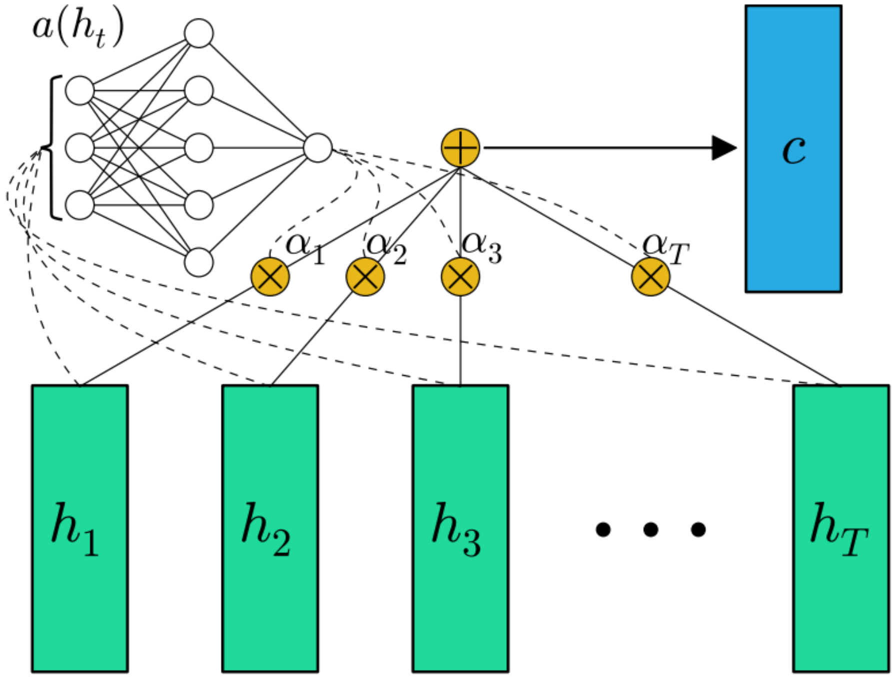
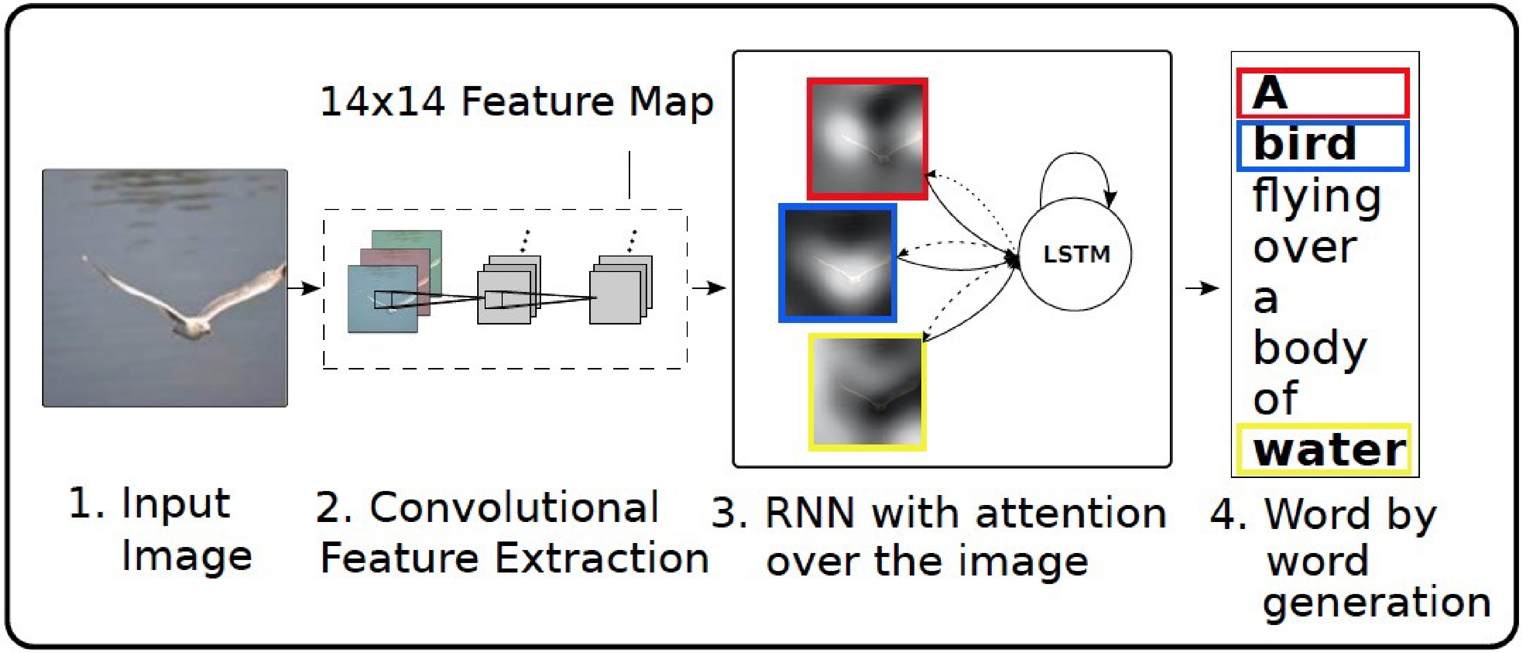


Figure 1: Schematic of our proposed "feed-forward" attention mechanism (cf. (Cho, 2015) Figure 1). Vectors in the hidden state sequence h_t are fed into the learnable function $a(h_t)$ to produce a probability vector α. The vector c is computed as a weighted average of h_t, with weighting given by α.

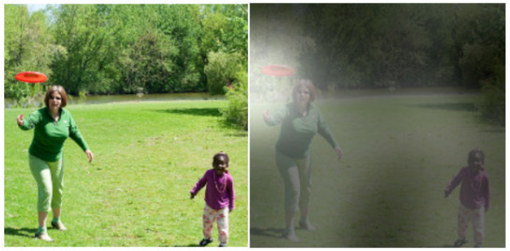
Application of Attention in Image Understanding

Generate a text description of the image

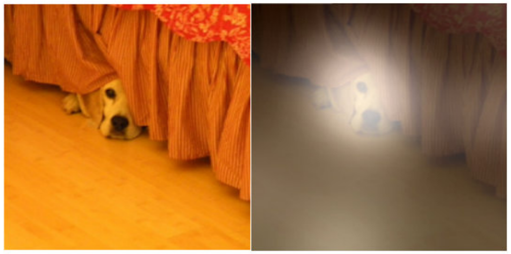

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Application of Attention in Image Understanding

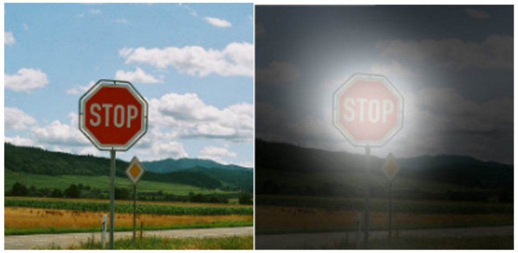
Match objects in text and images



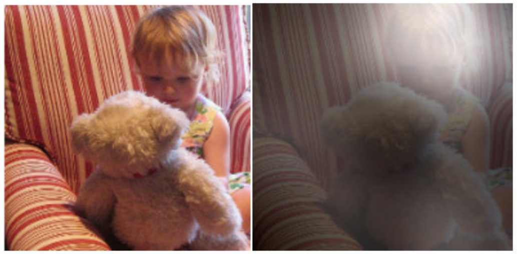
A woman is throwing a frisbee in a park.



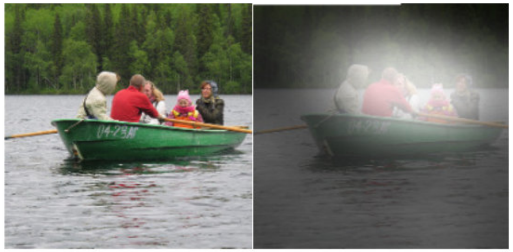
A dog is standing on a hardwood floor.



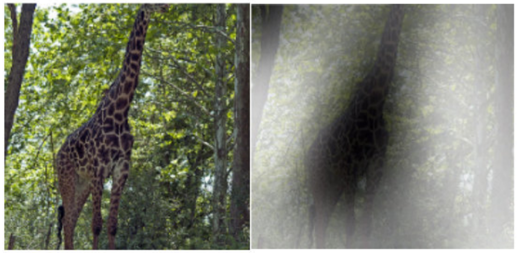
A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Transformer

Avoid RNN structure and use Attention

Transformer

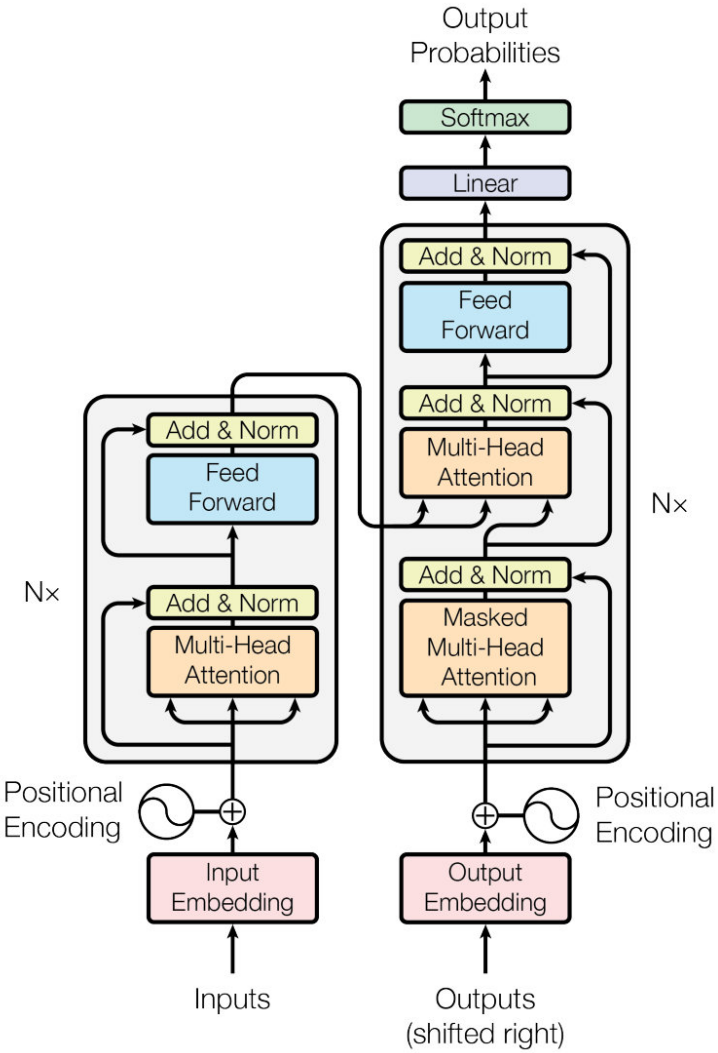


Figure 1: The Transformer - model architecture.

Performance

Performance

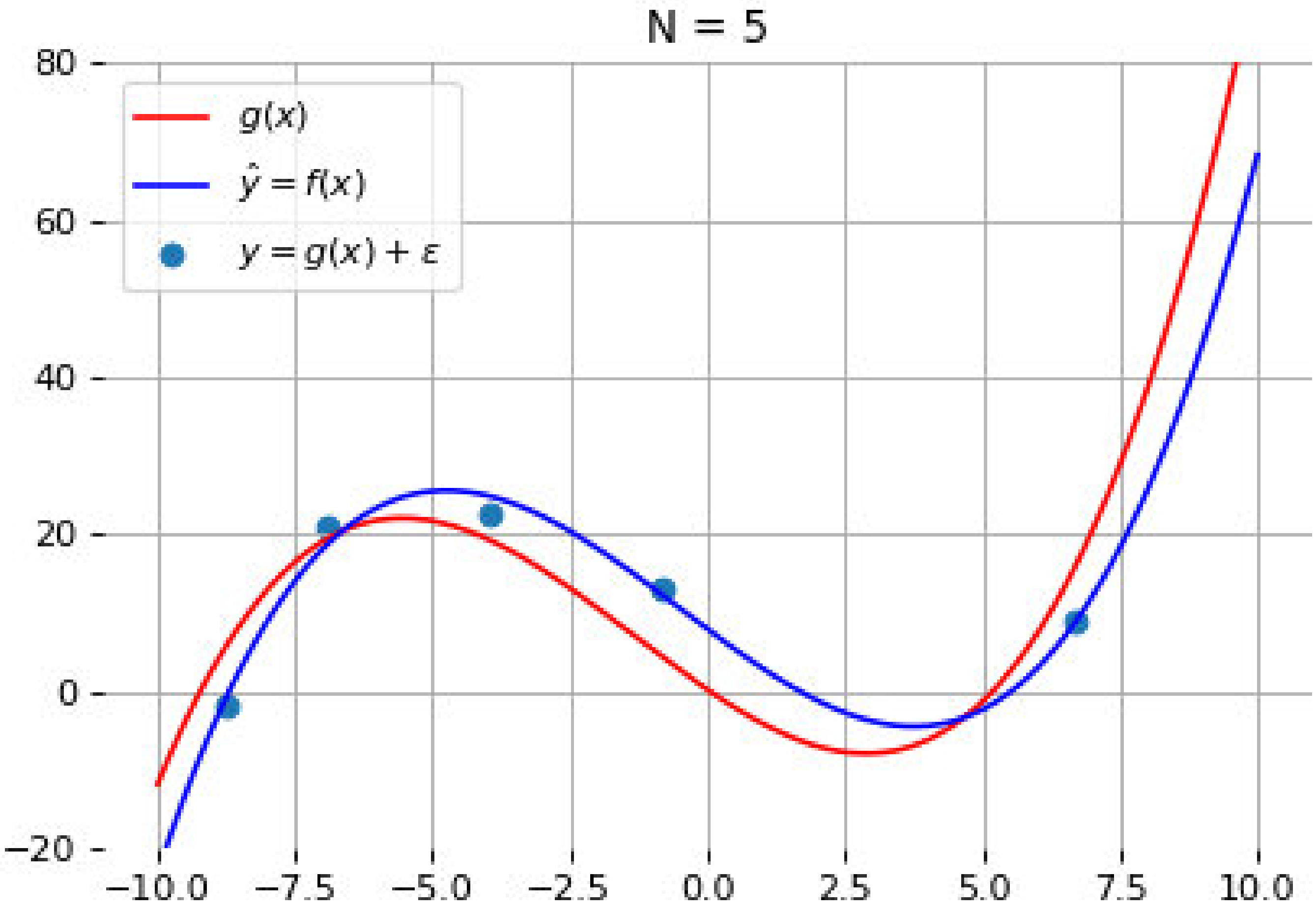
- Data
- Model
- Training method
- Optimization
- Parameter tuning

1) Data

Good data is the key to success

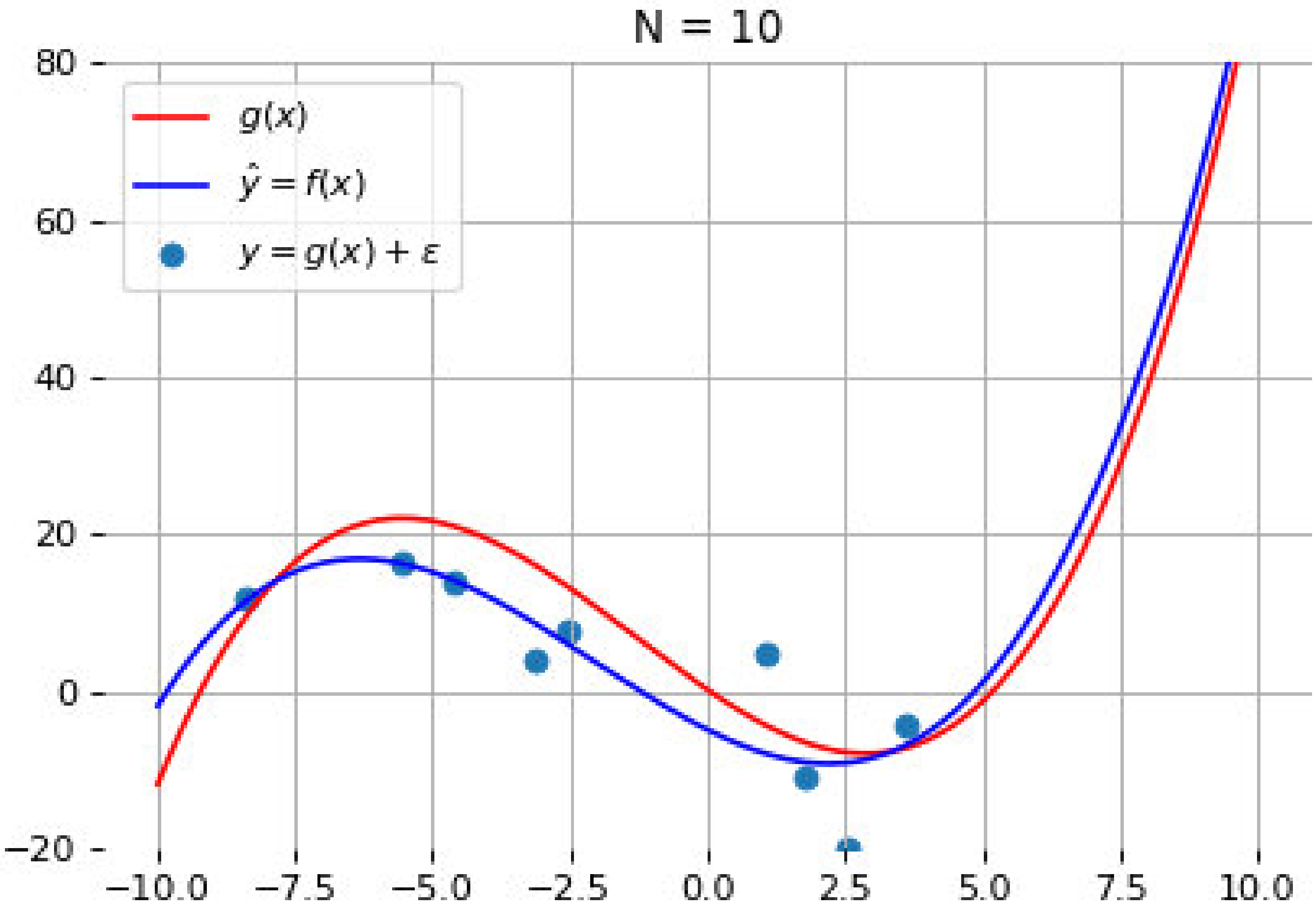
Small Amount of Data

Large errors



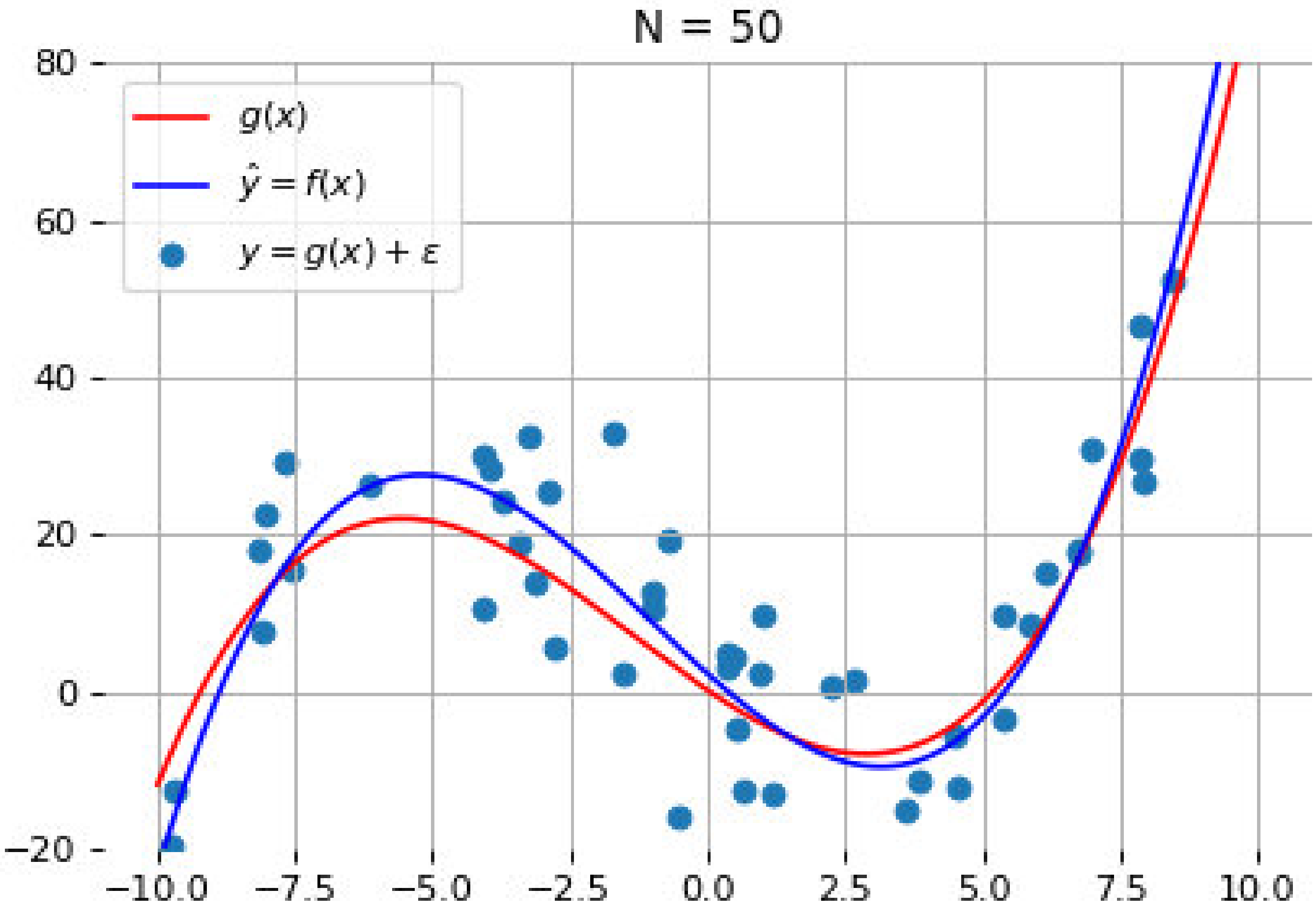
Data Volume Grows

Model errors decrease



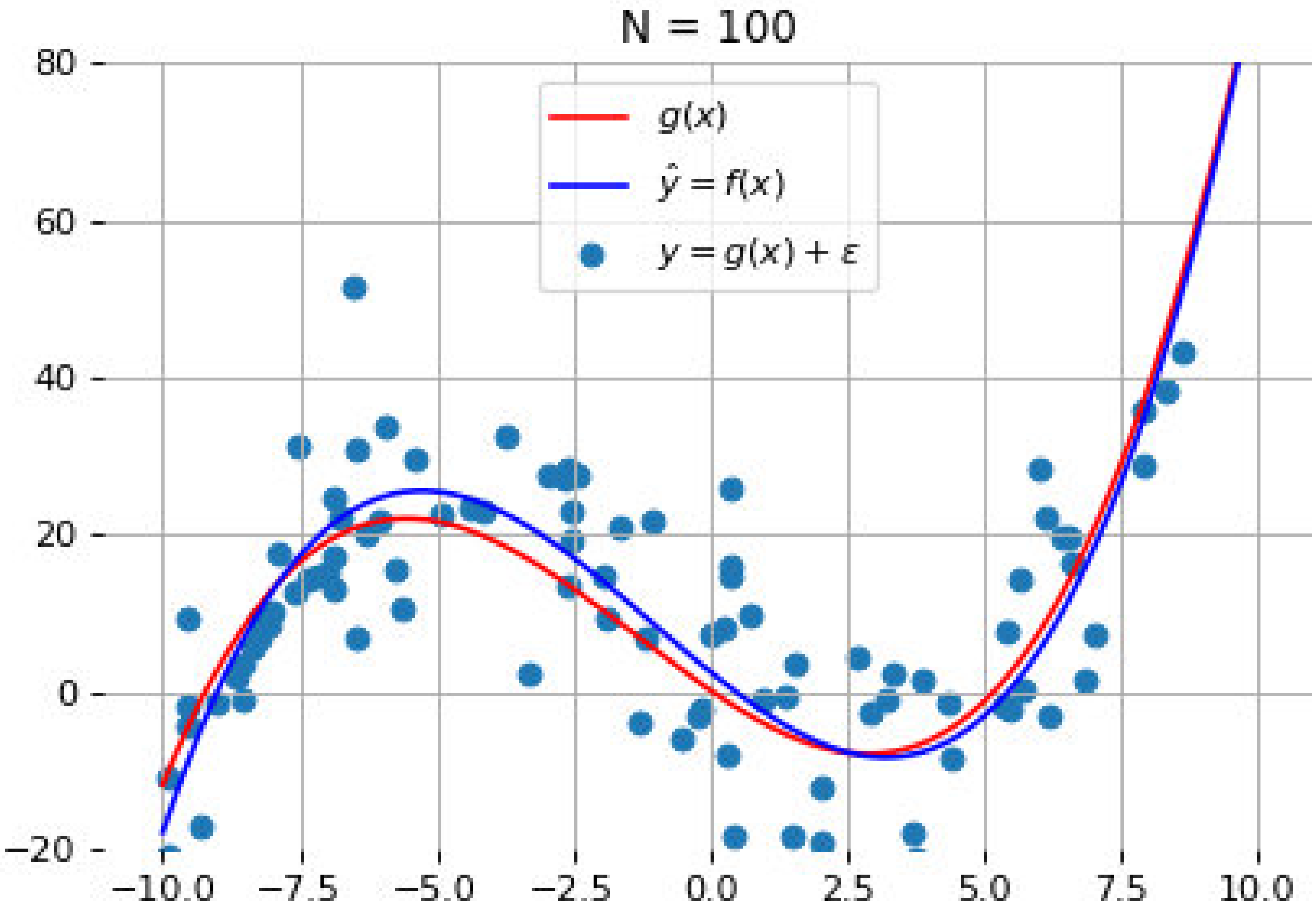
Data Volume Grows

Model errors decrease



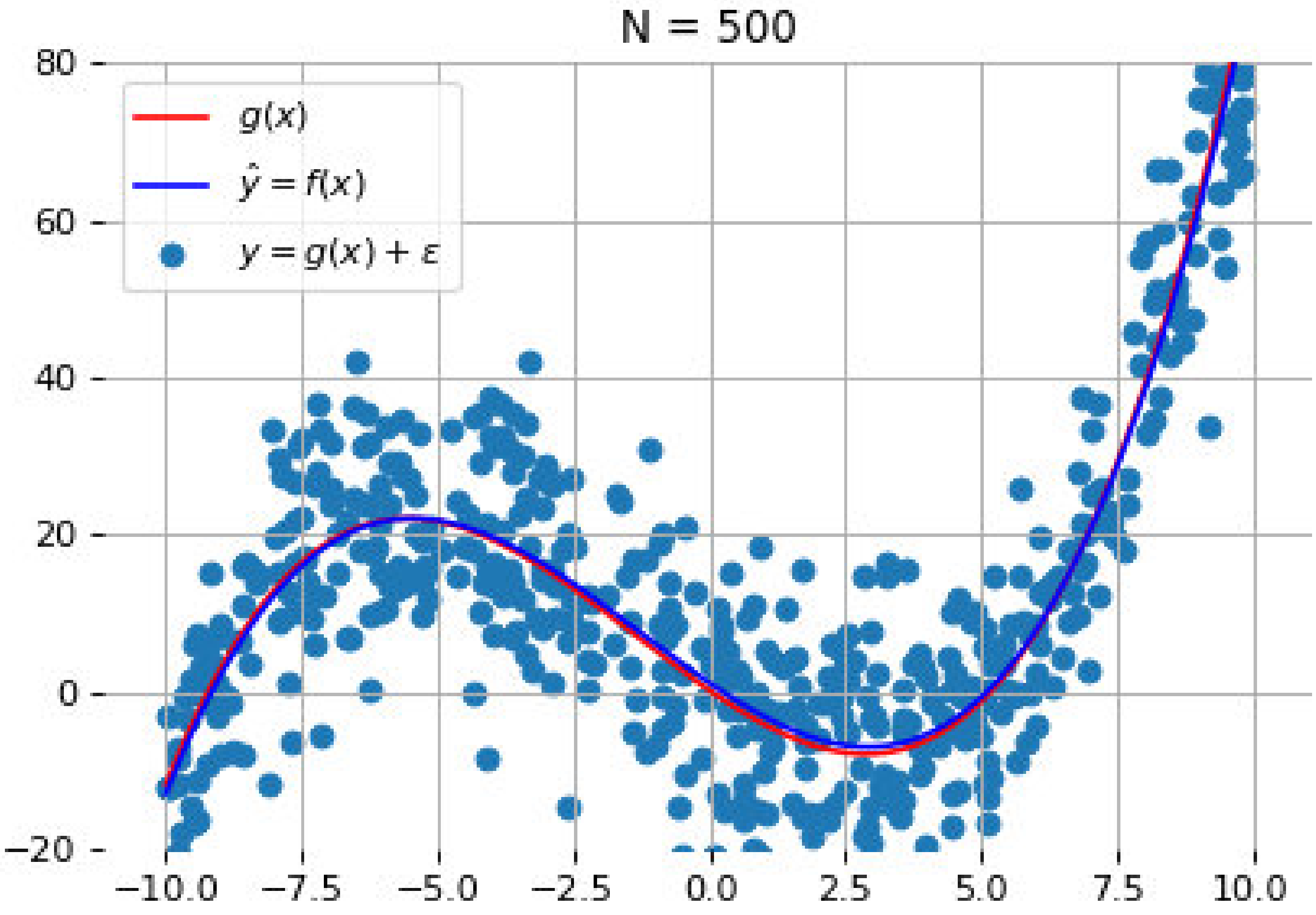
Data Volume Grows

Model errors decrease



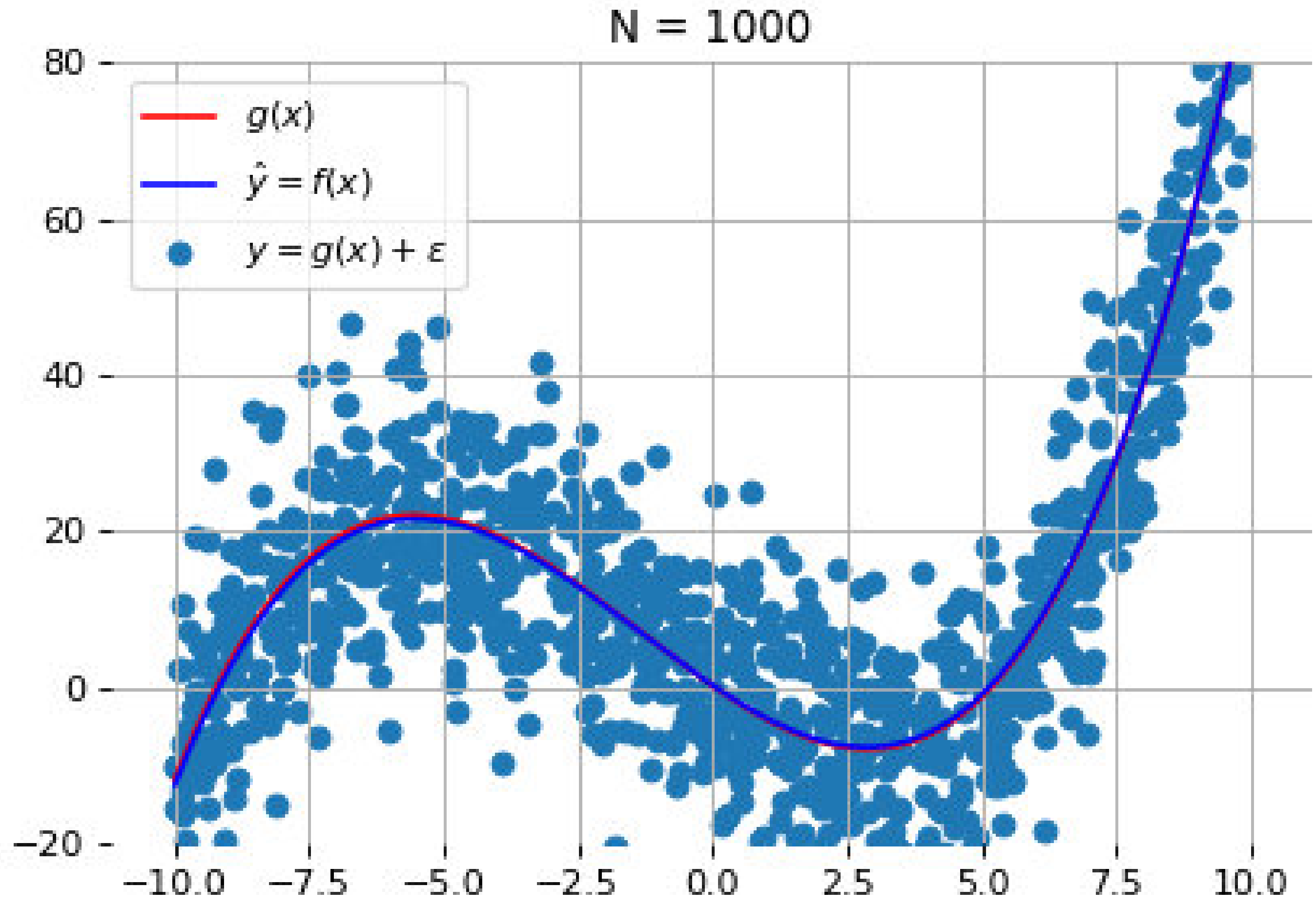
Data Volume Grows

Model errors decrease



Data Volume Grows

Model errors decrease

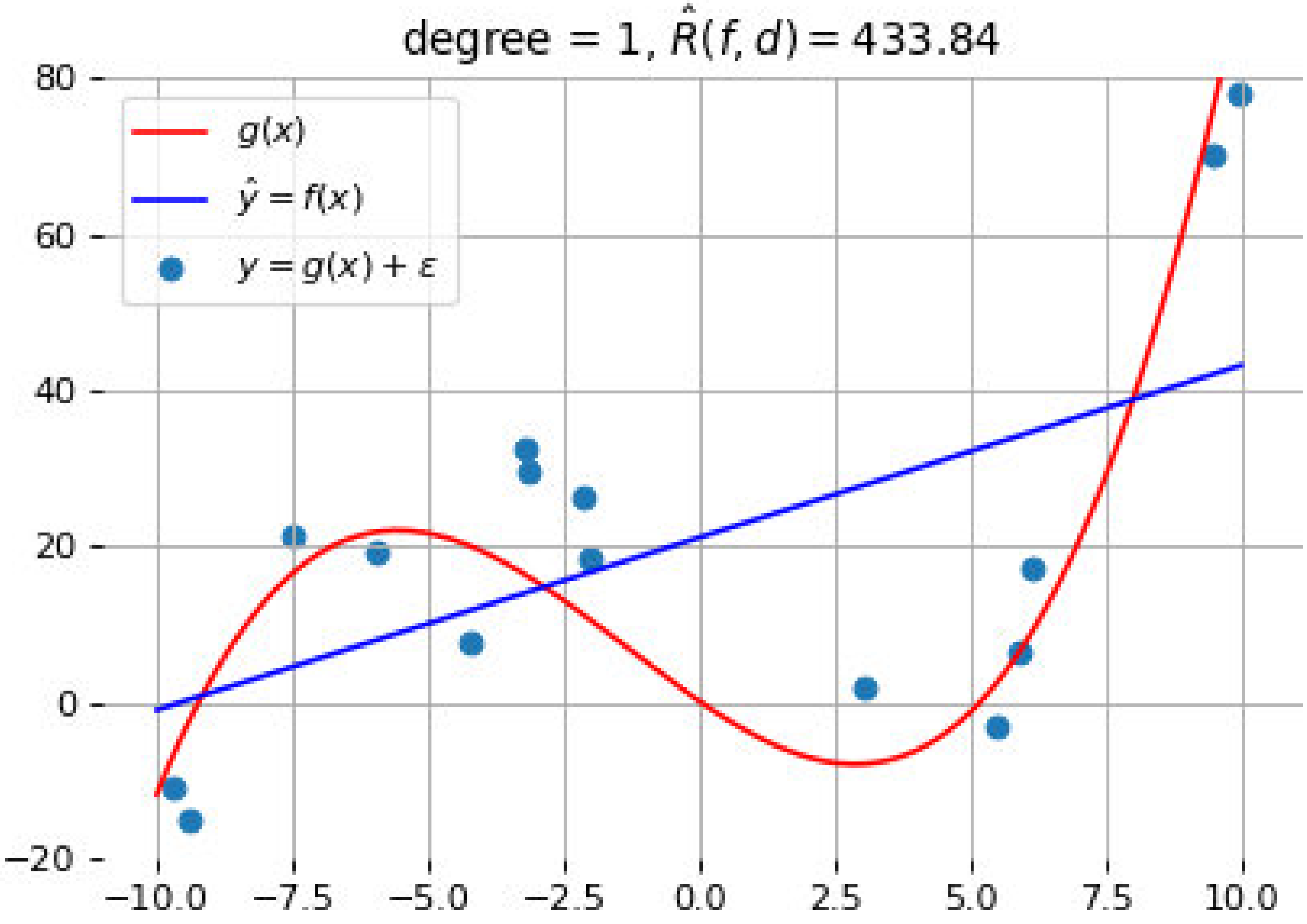


2) Model

Model selection is very important

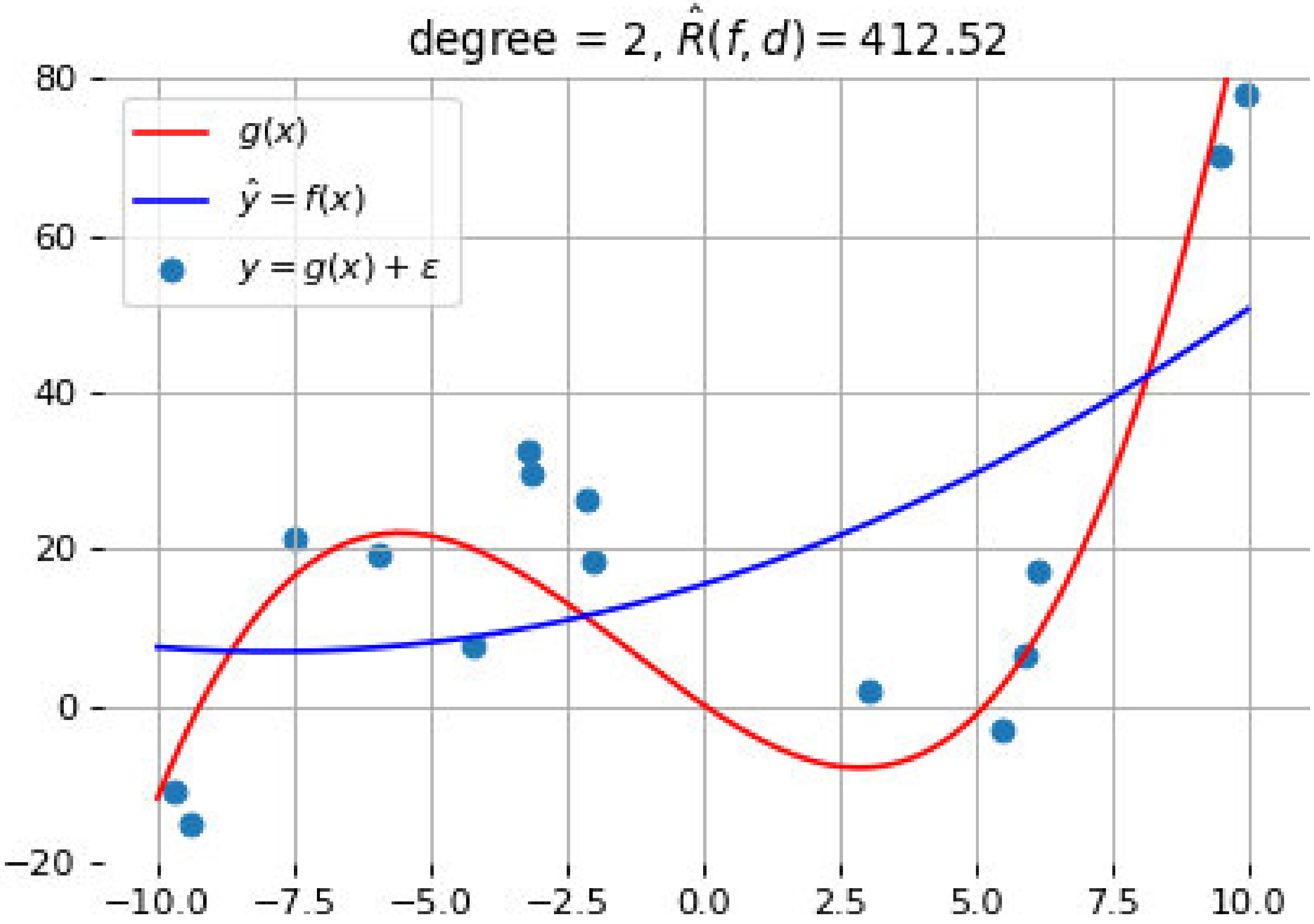
Underfitting

Too simple model, low capacity, underfitting



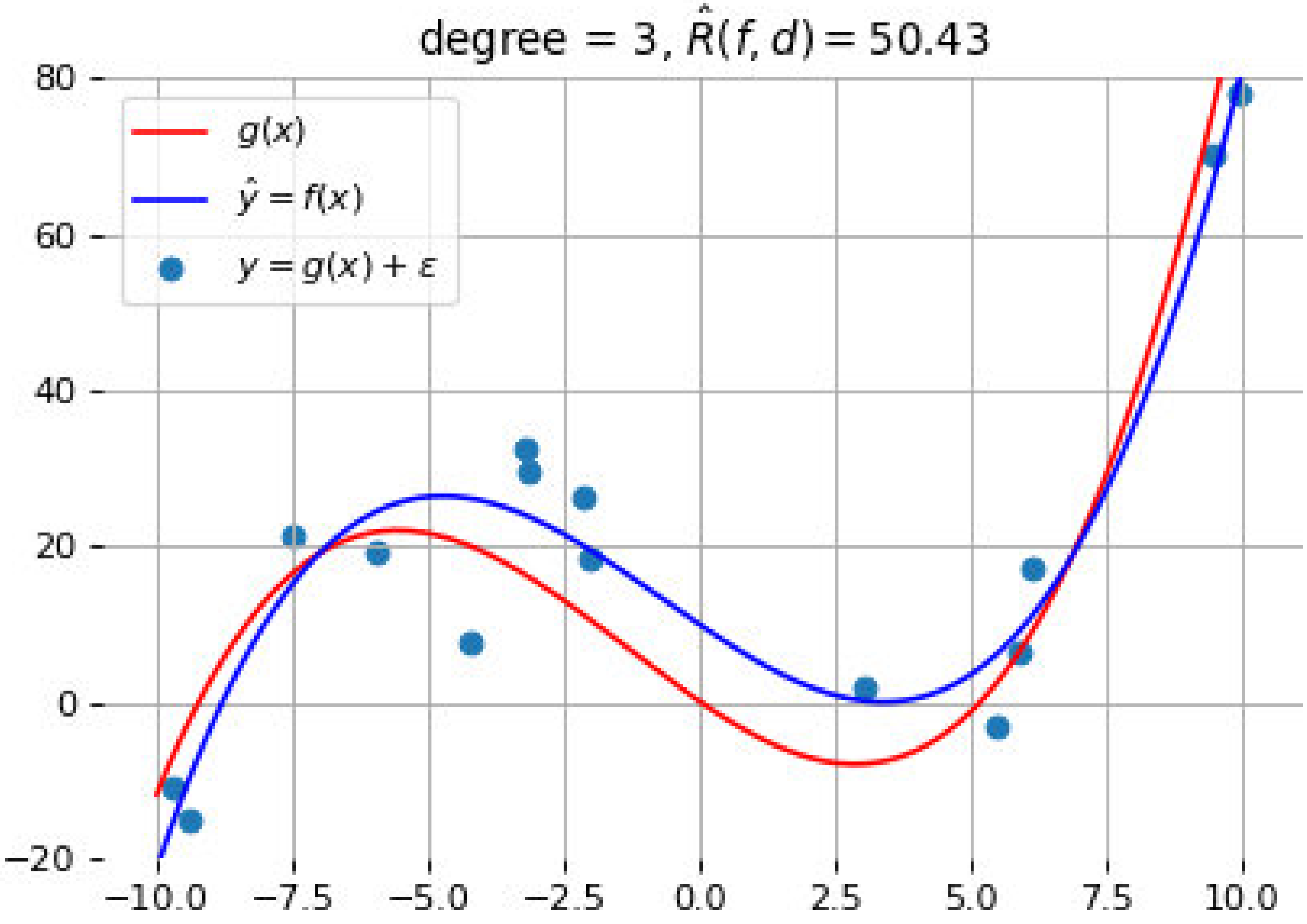
Underfitting

Too simple model, low capacity, underfitting



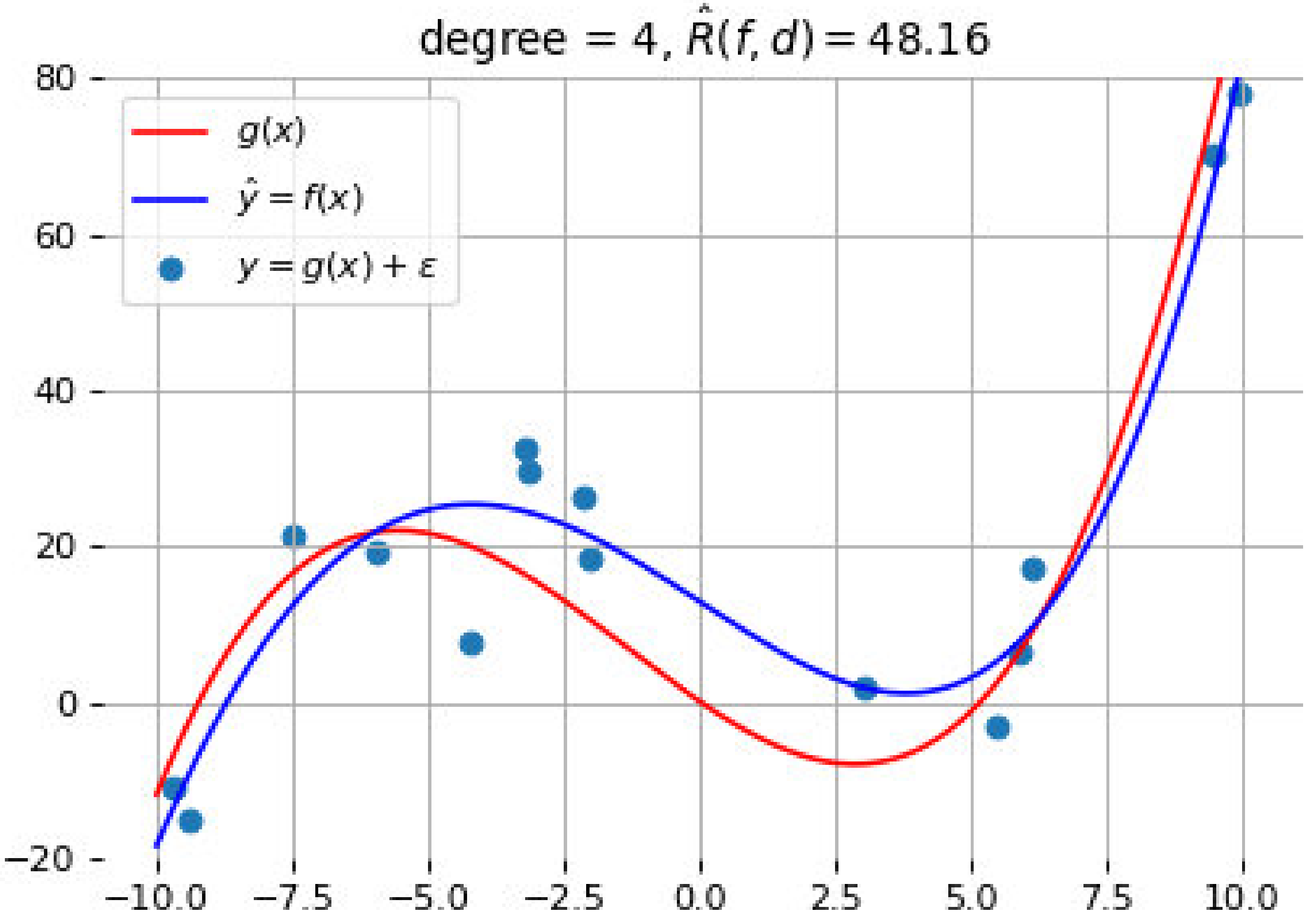
Model Capabilities

Moderate model



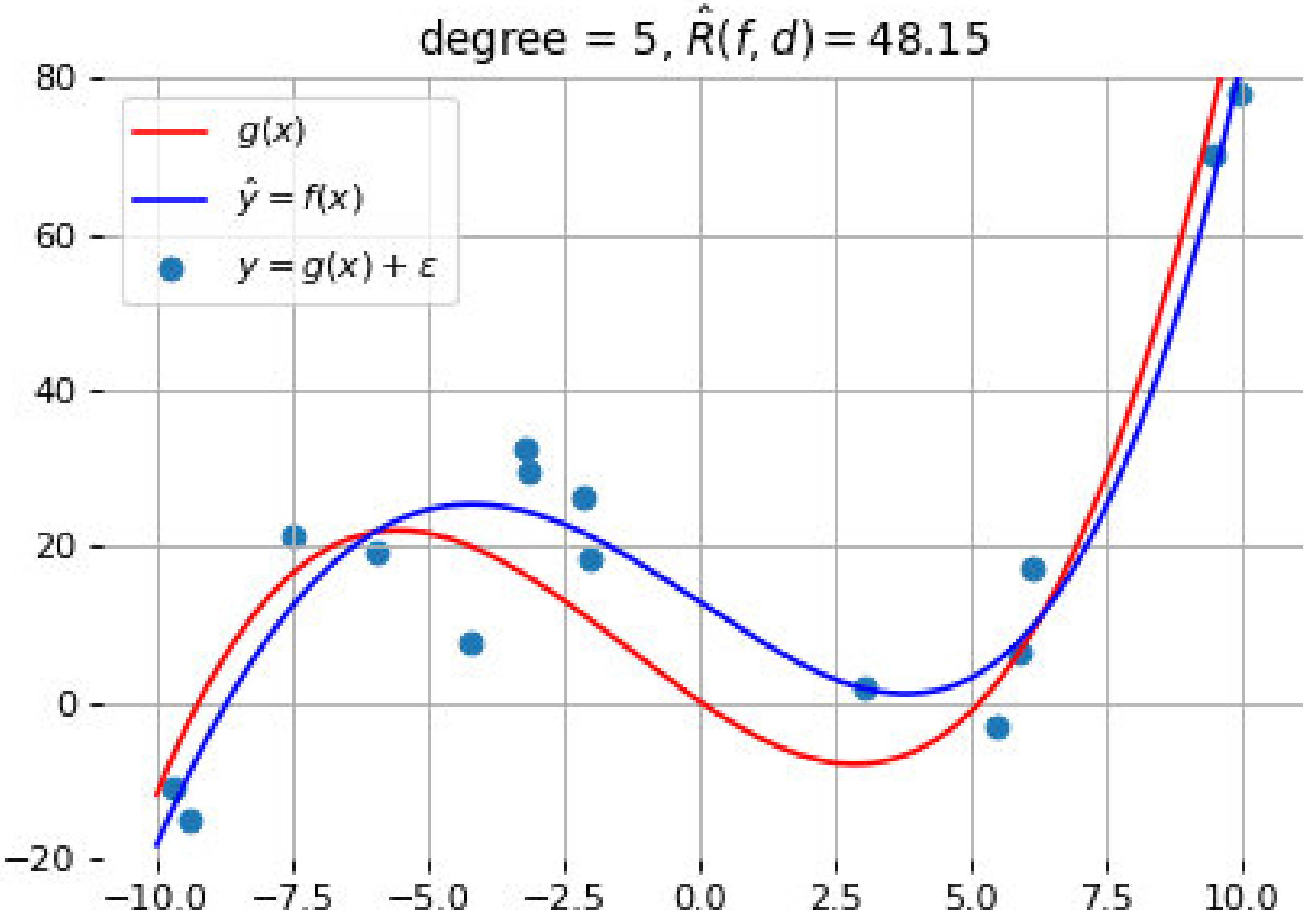
Model Capabilities

Moderate model



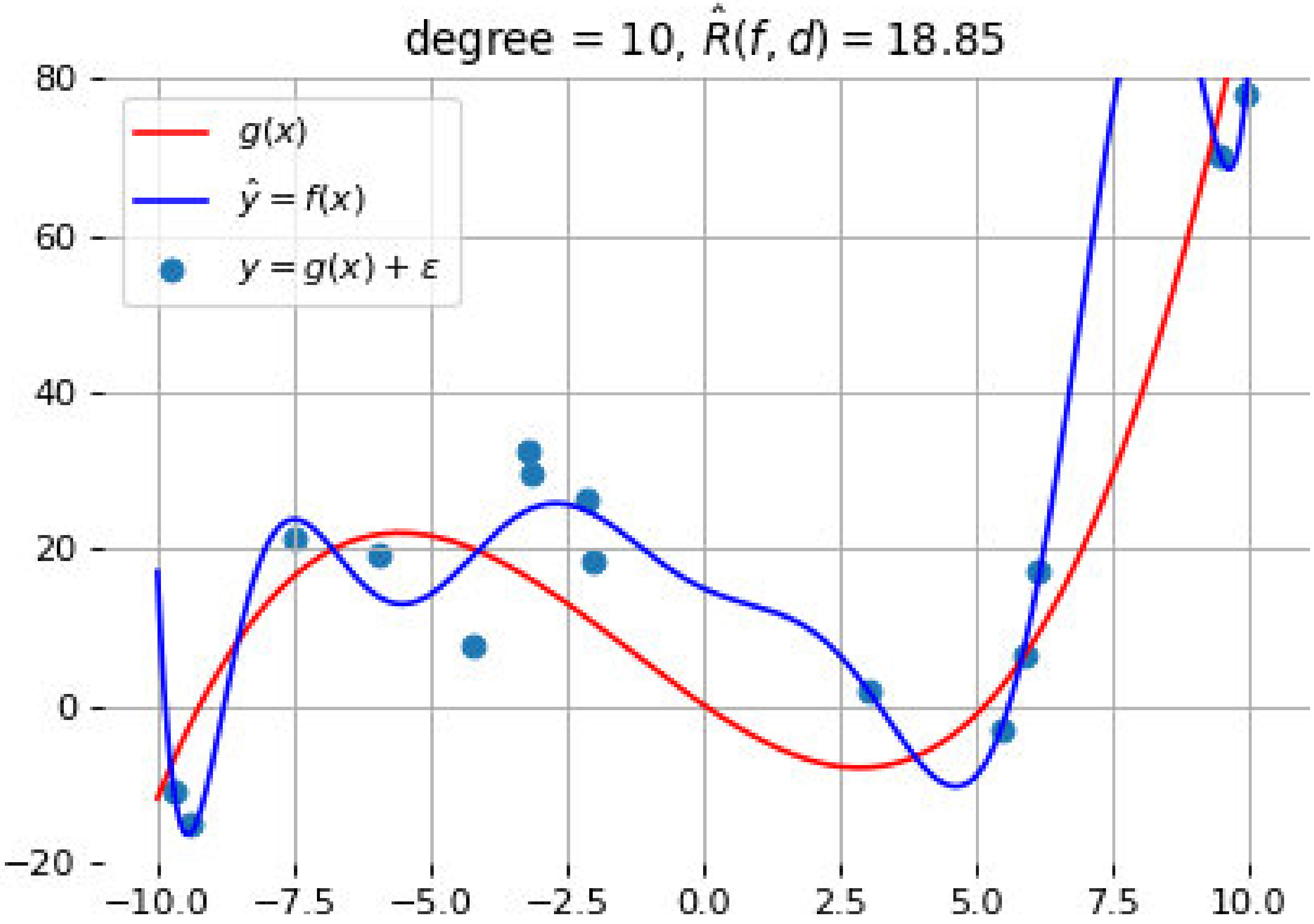
Model Capabilities

Moderate model



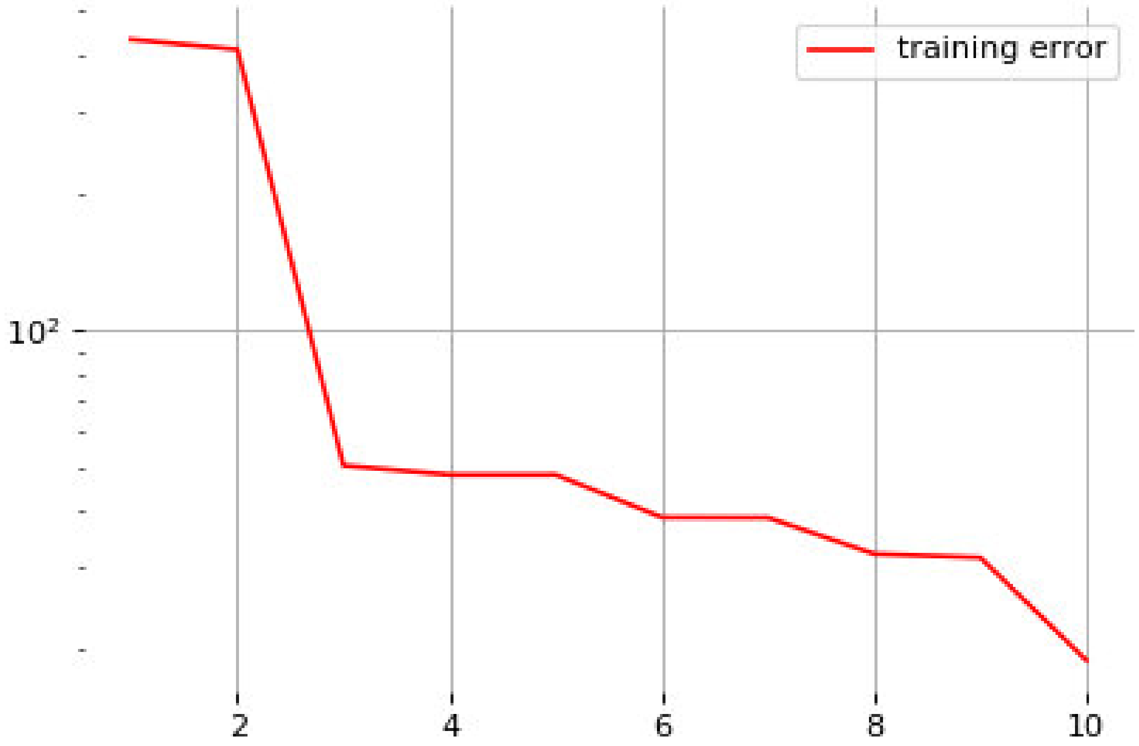
Overfitting

Too complex model, capability is too high



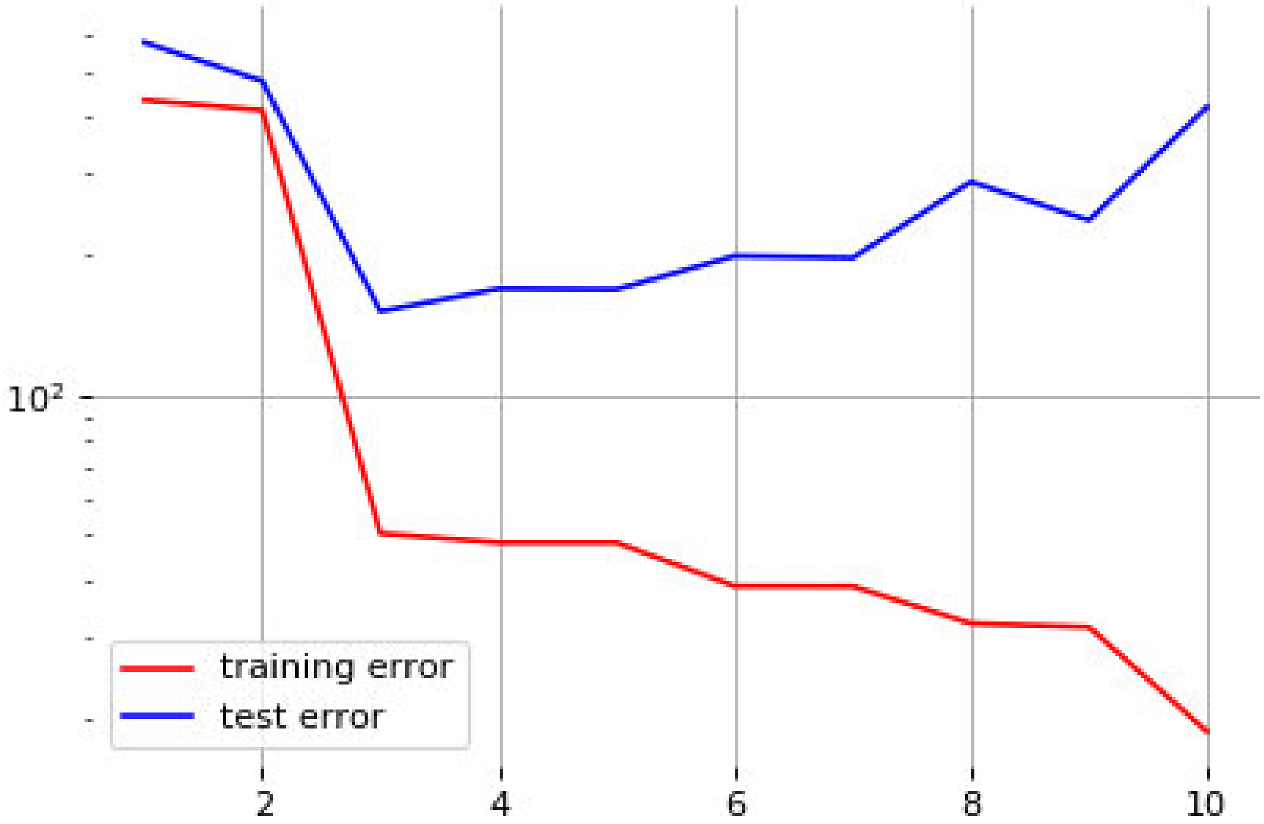
Overfitting

- On the training set, model errors continue to decline as model capability increases
- But the final drop is overfitting



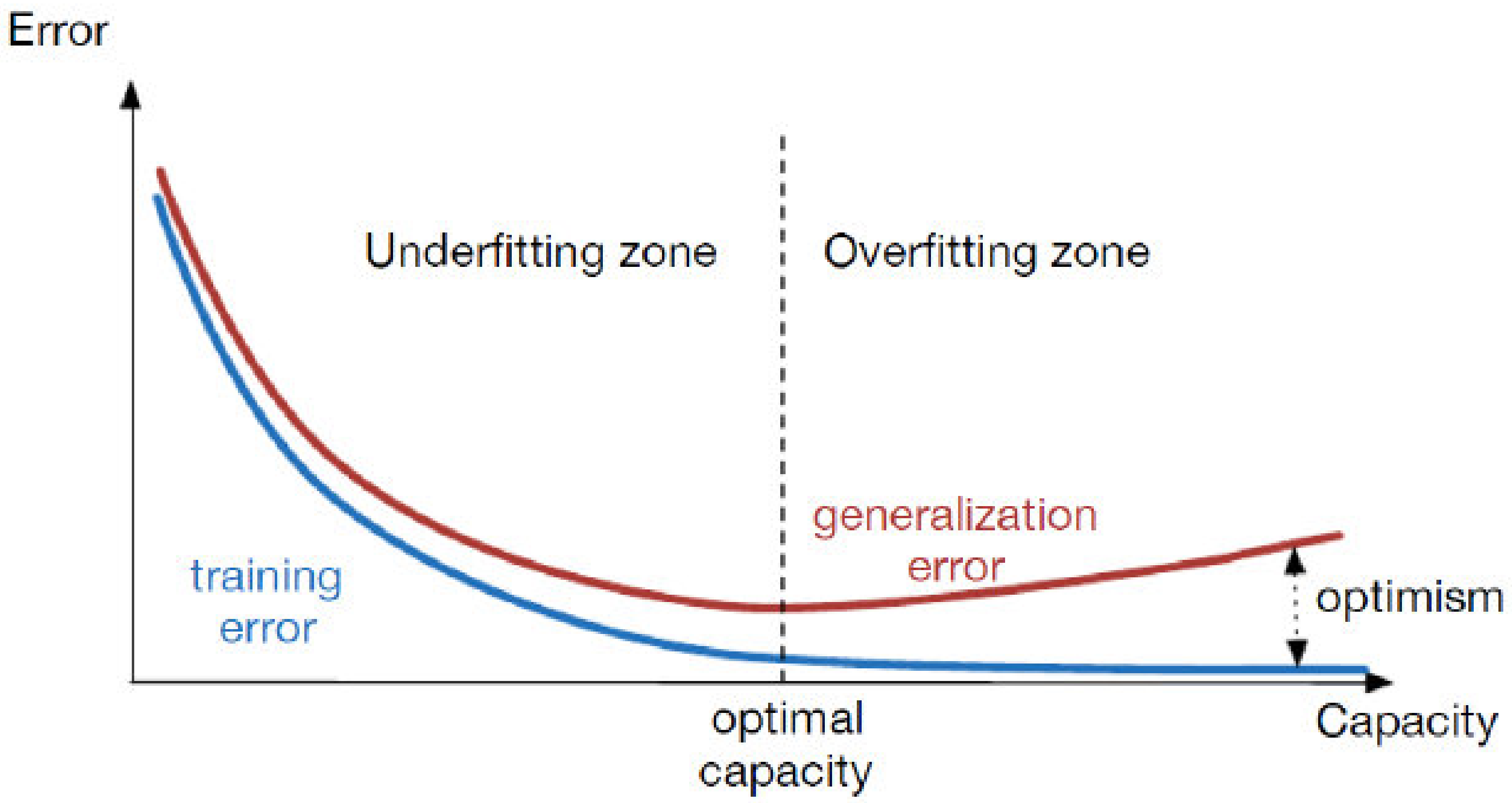
Overfitting

- Overfitting causes model errors to rise incorrectly on test set



Model Capabilities

Choosing the right model is very important



Model Selection

- Deep neural network is not the only machine learning algorithm
- You can solve the problem based on a clean data set and simpler algorithms (such as linear regression).
- Occam's Razor Guidelines

Occam's Razor Guidelines

Simplicity first

“The explanation requiring the fewest assumptions is most likely to be correct”

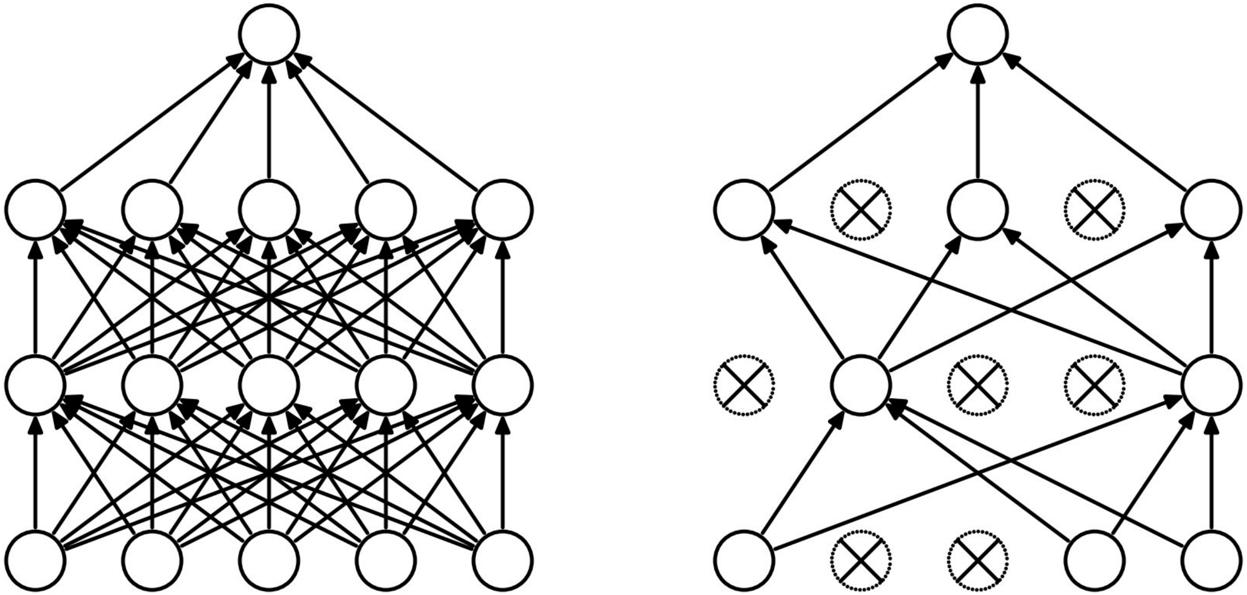
Occam's Razor

- Proposed by 14th-century logician, William of Occam
- "When presented with competing hypotheses that make the same predictions, one should select the solution with the fewest assumptions"

3) Training Method

Dropout

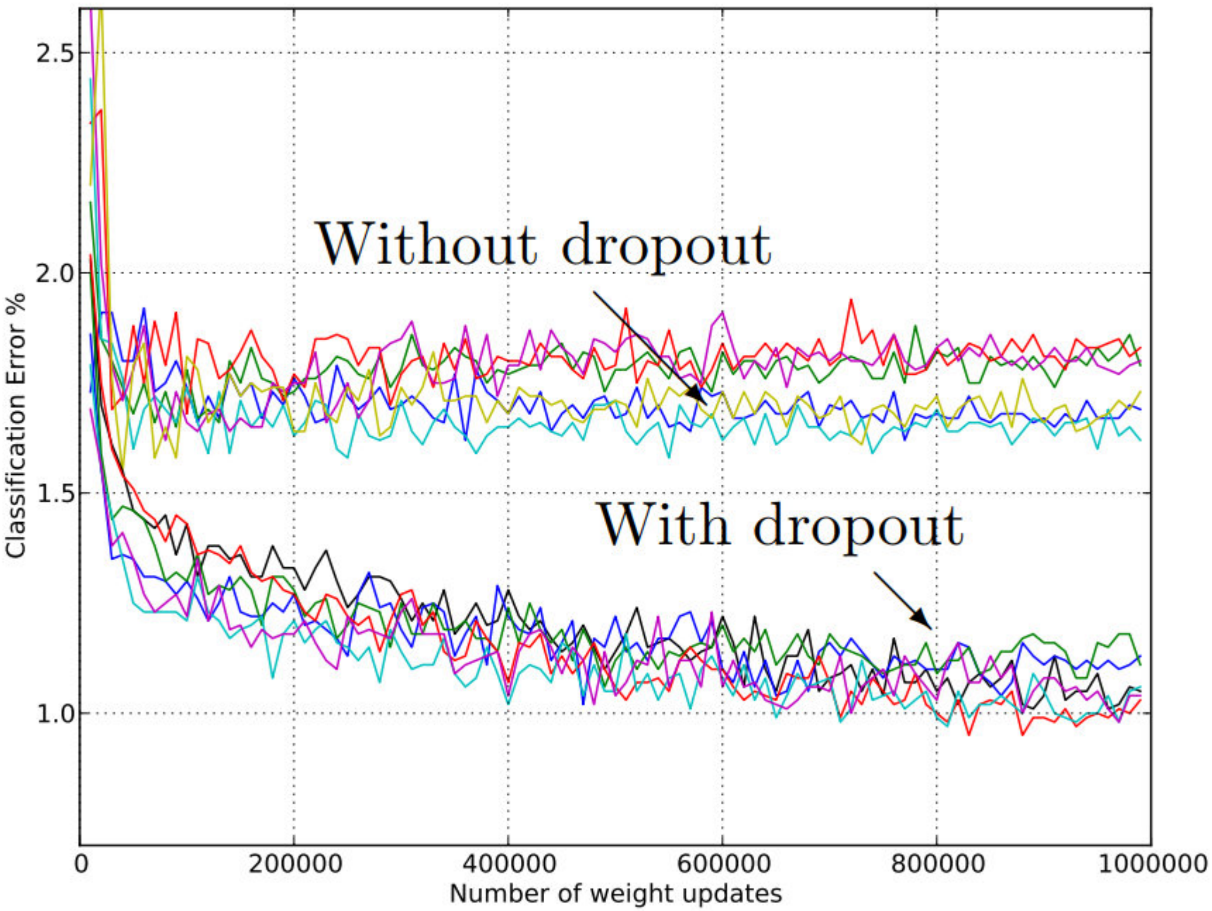
- In each round of optimization, some neurons are randomly selected and added to the calculation
- Prevent some neurons from being particularly powerful and dominate



(a) Standard Neural Net

(b) After applying dropout.

Dropout

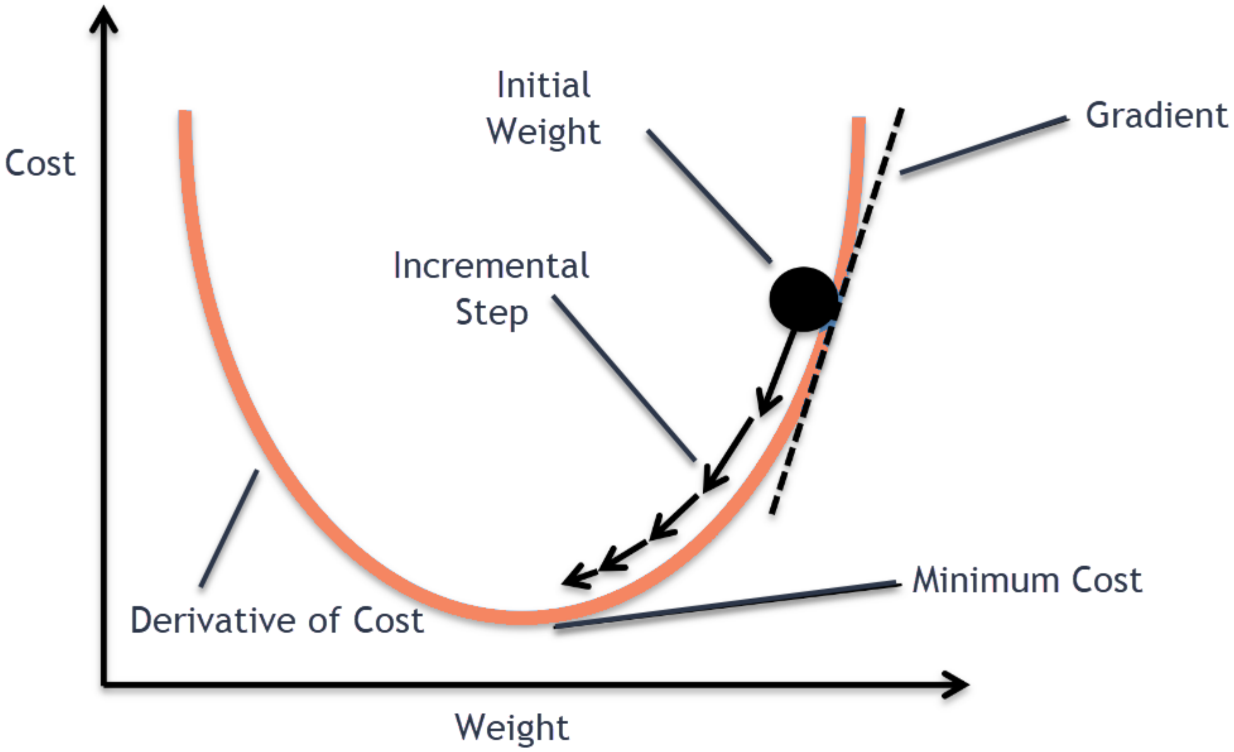

Without dropout
With dropout
Classification Error %
2.5
2.0
1.5
1.0
0
200000
400000
600000
800000
1000000
Number of weight updates

4) Optimization

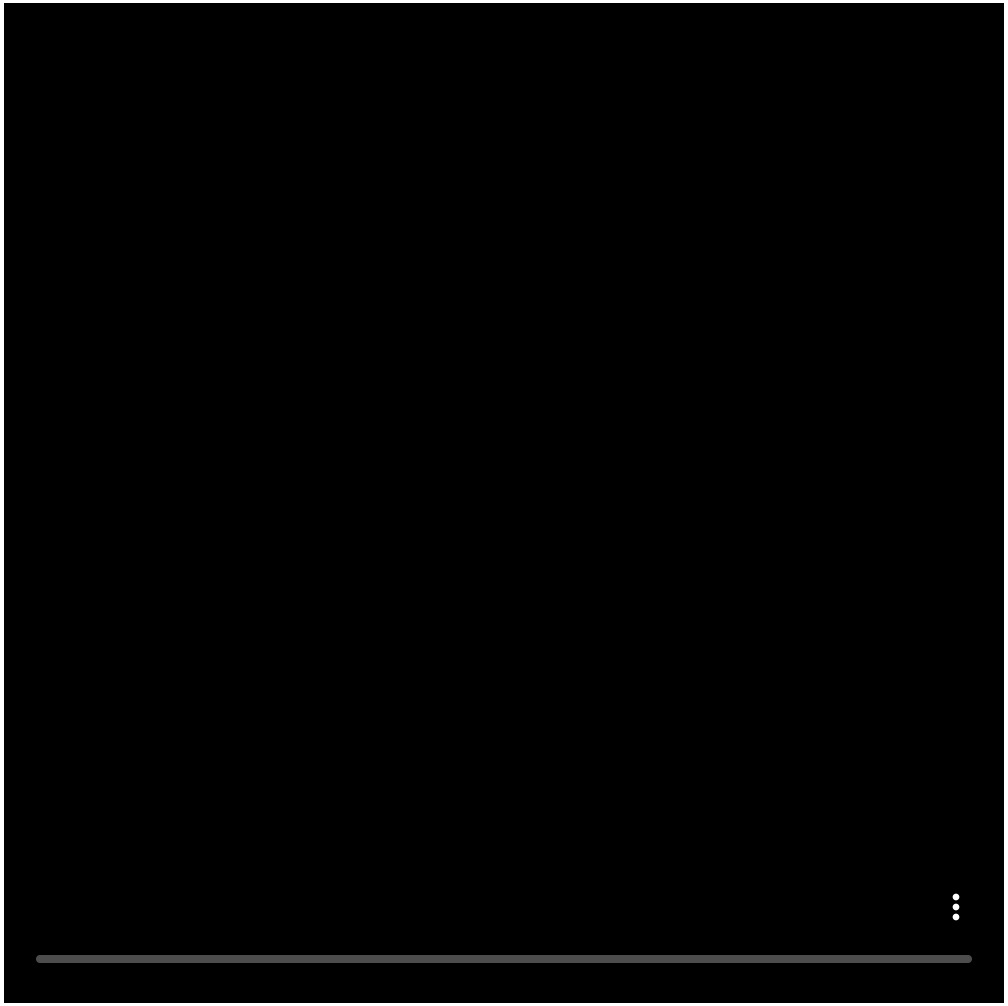
Find the right to minimize model errors

Gradient Descent

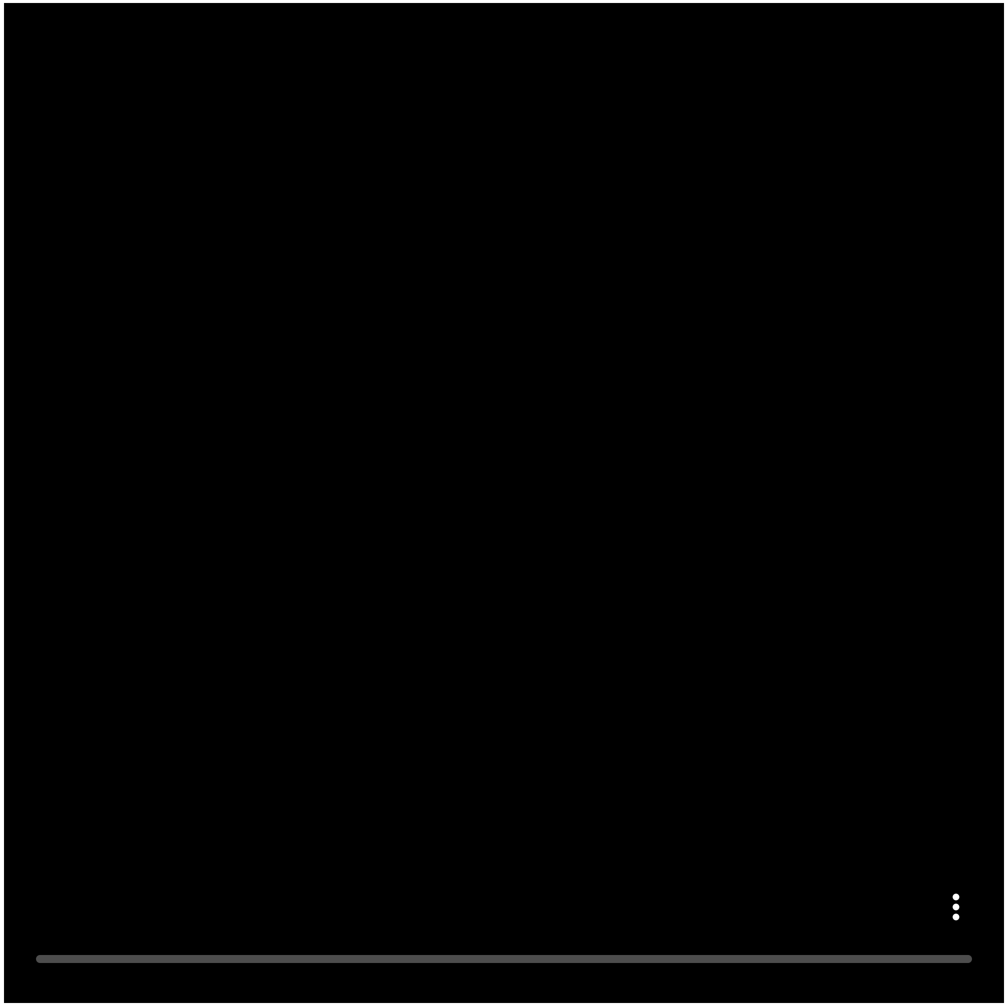
- Find model parameter with the smallest error
- Positive slope, reduce
- Negative slope, increase



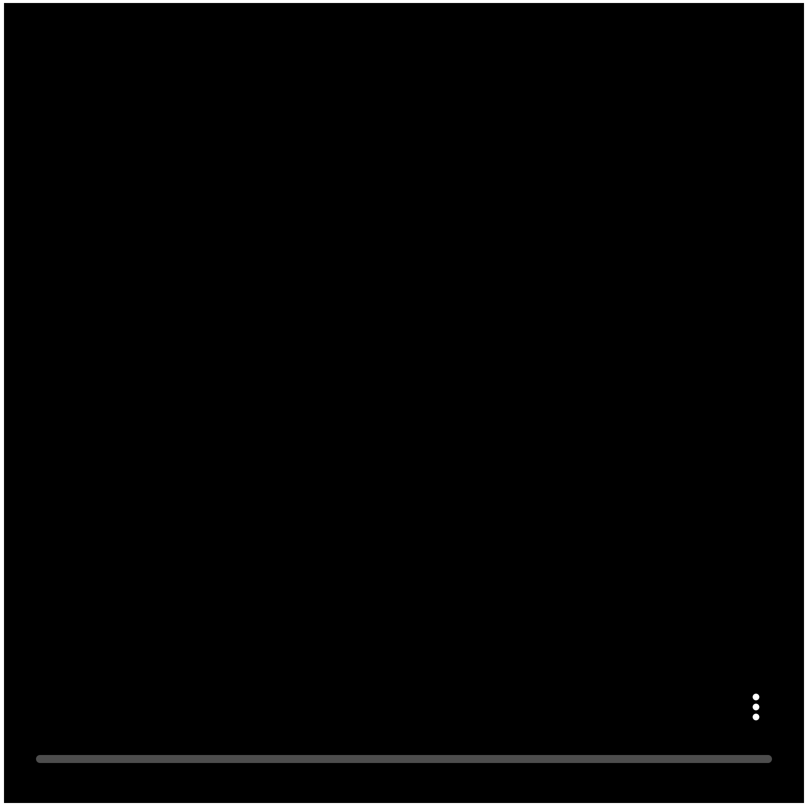
Gradient Descent



Gradient Descent



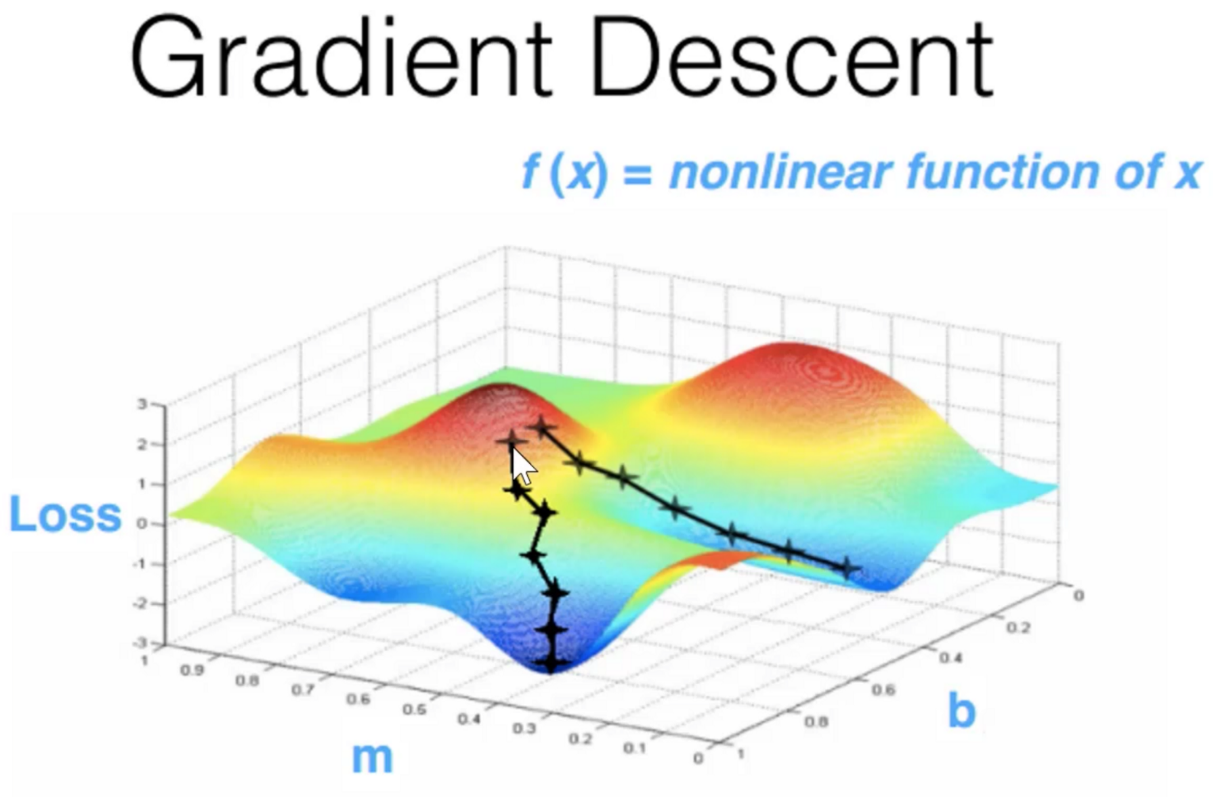
Adaptive Step Size Selection



5) Parameter Tuning

Parameters affect model performance

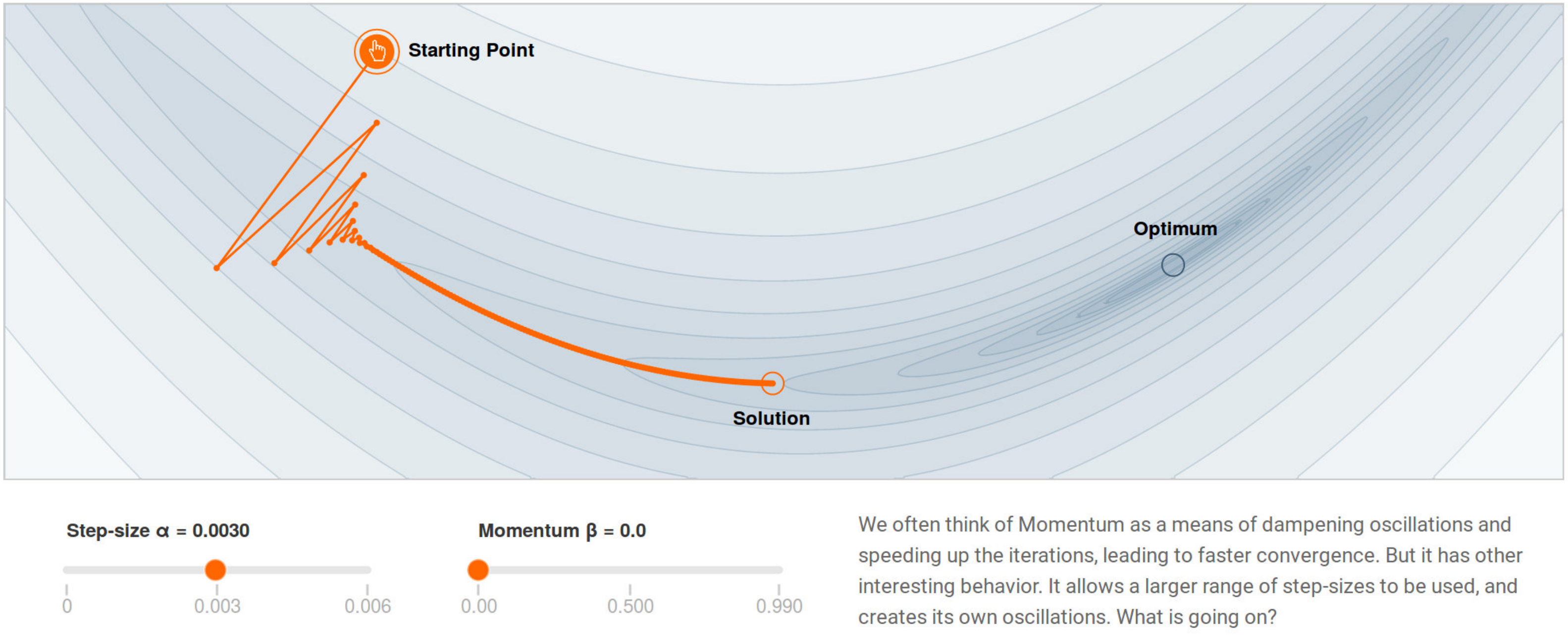
Situation is Complex in High Dimensions



Gradient Descent

f (x) = nonlinear function of x

Adaptive Step Size Selection



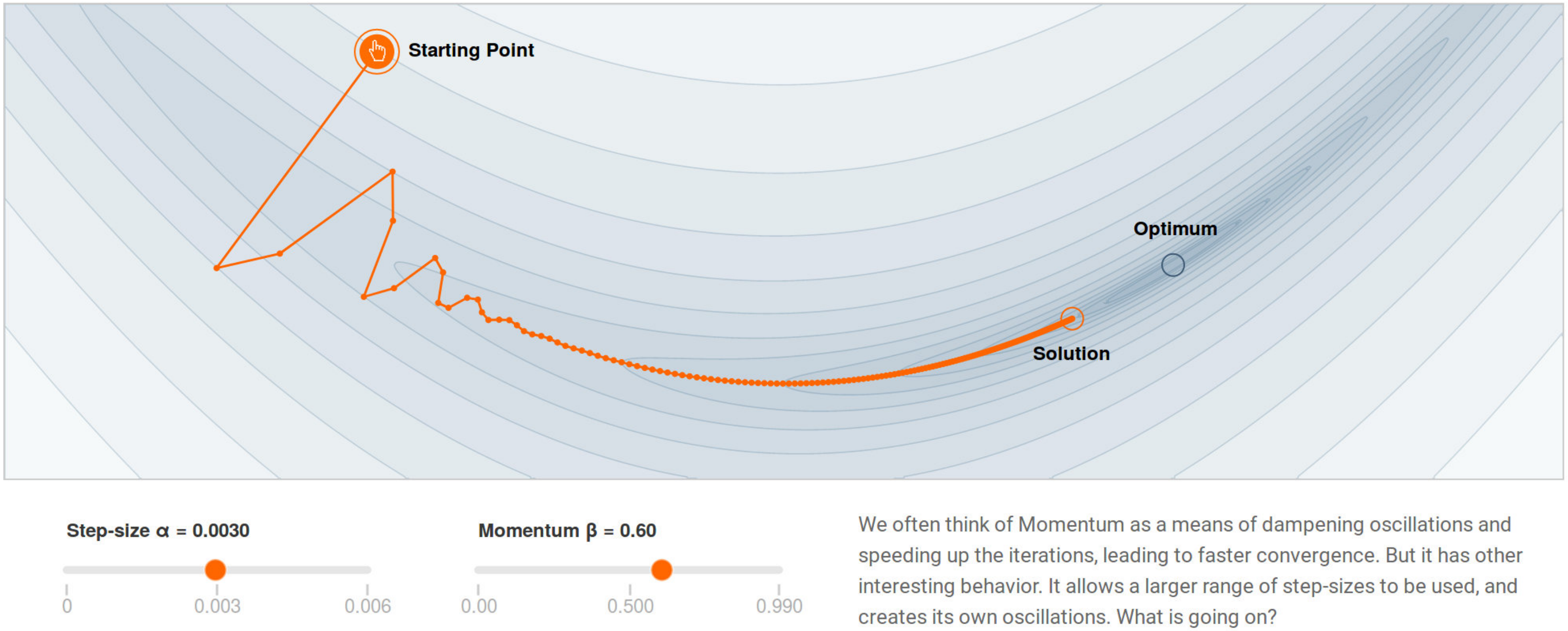
Step-size α = 0.0030

Momentum β = 0.0

We often think of Momentum as a means of dampening oscillations and speeding up the iterations, leading to faster convergence. But it has other interesting behavior. It allows a larger range of step-sizes to be used, and creates its own oscillations. What is going on?

Miss the best

Adaptive Step Size Selection



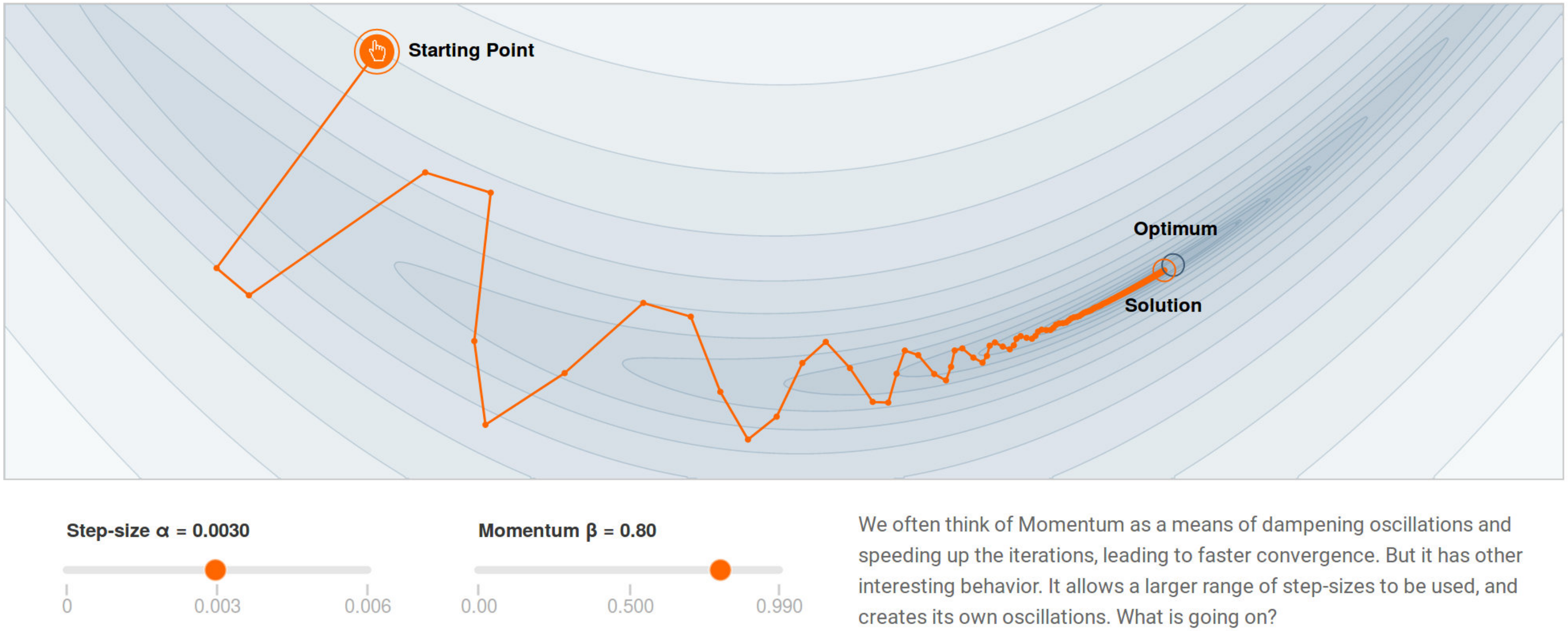
Step-size α = 0.0030

Momentum β = 0.60

We often think of Momentum as a means of dampening oscillations and speeding up the iterations, leading to faster convergence. But it has other interesting behavior. It allows a larger range of step-sizes to be used, and creates its own oscillations. What is going on?

Miss the best

Adaptive Step Size Selection



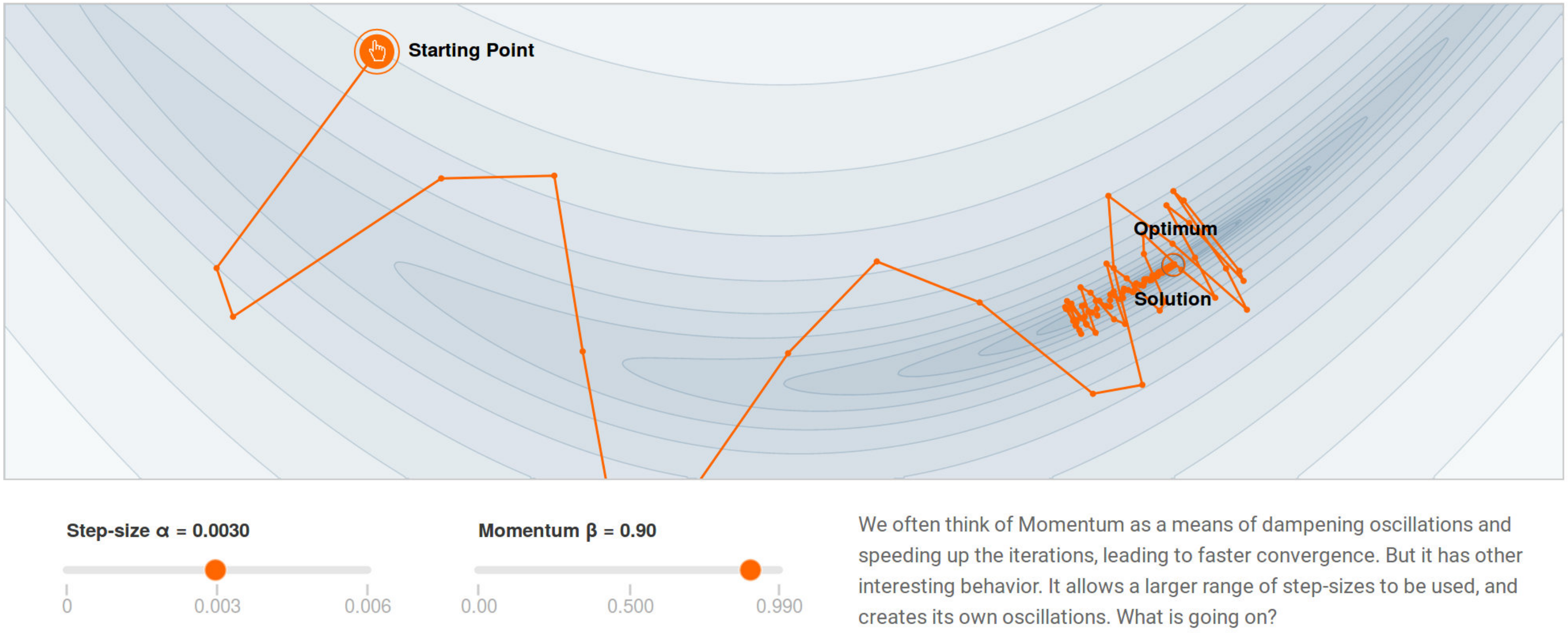
Step-size α = 0.0030

Momentum β = 0.80

We often think of Momentum as a means of dampening oscillations and speeding up the iterations, leading to faster convergence. But it has other interesting behavior. It allows a larger range of step-sizes to be used, and creates its own oscillations. What is going on?

Reach the best

Adaptive Step Size Selection



Step-size α = 0.0030

Momentum β = 0.90

We often think of Momentum as a means of dampening oscillations and speeding up the iterations, leading to faster convergence. But it has other interesting behavior. It allows a larger range of step-sizes to be used, and creates its own oscillations. What is going on?

Reached the best, but oscillate

Parameter Tuning

1. Randomly initialize model
2. Use this model to make predictions
3. Compare predictions with real results: if wrong, adjust model
4. Repeat steps 2-3 until performance cannot be improved
5. Validate on the validation set and choose the best model parameters

Quiz I

- What is supervised learning? What is unsupervised learning?
- Is image classification supervised or unsupervised?
- Is clustering supervised or unsupervised?
- Is the linear regression model a straight line or an S-curve?
- Is the logistic regression model a straight line or an S-curve?
- What are the two parts of a perceptron?

Quiz II

- What are the three most typical types of deep neural networks?
- The model is not capable enough. Will it overfit or underfit?
- The model is too powerful. Will it overfit or underfit?
- What is Occam's Razor Principle?