Exploring neural networks from outside to inside

Brief history & Basic theories

Congcong Yuan

10/24/2019

Outline

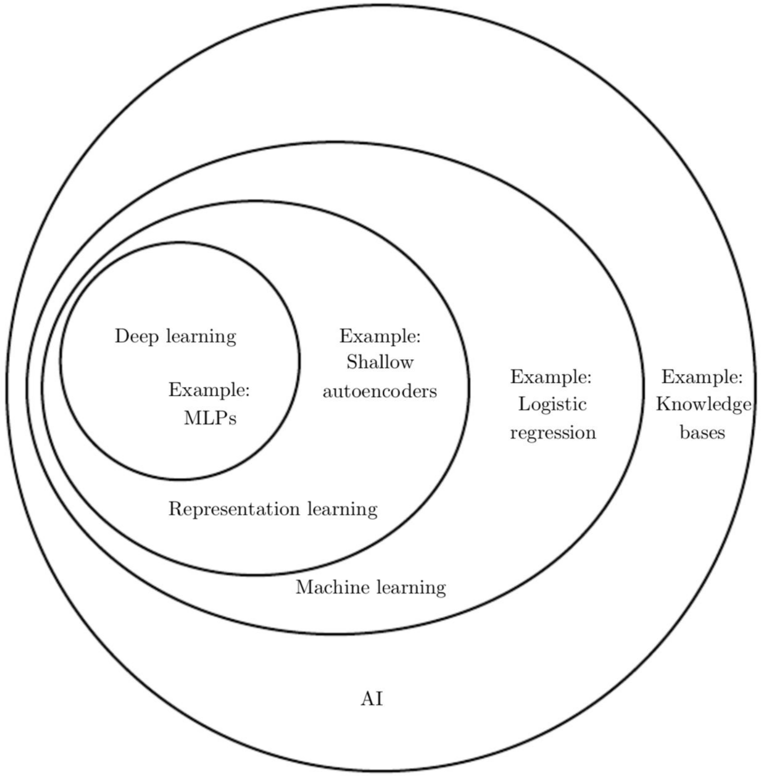
1. Deep learning brief history

2. Basic units of neural networks

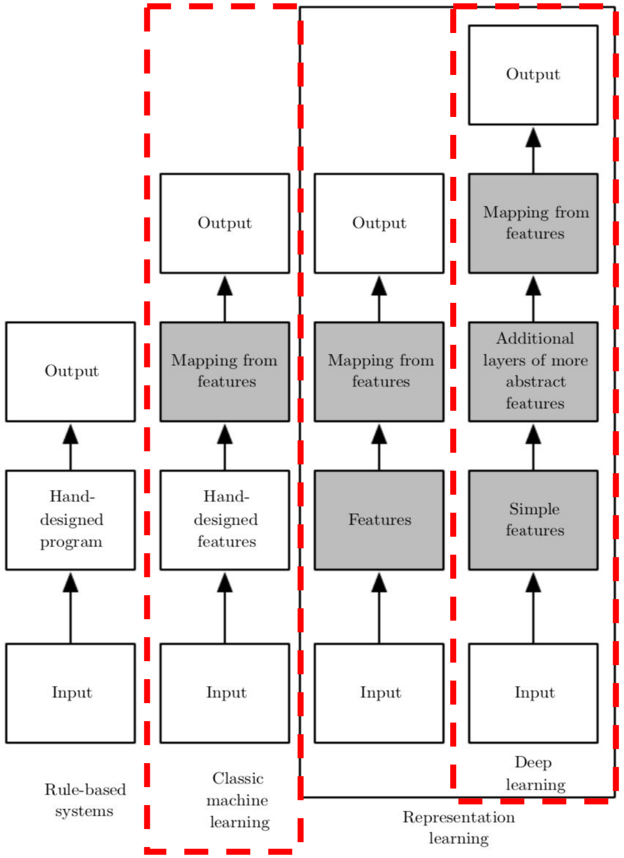
3. popular architectures of neural networks

01 Brief history

AI & Machine Learning & Deep learning



Venn diagram shows inclusion relation among AI, Machine learning and deep learning (Goodfellow et., 2017)

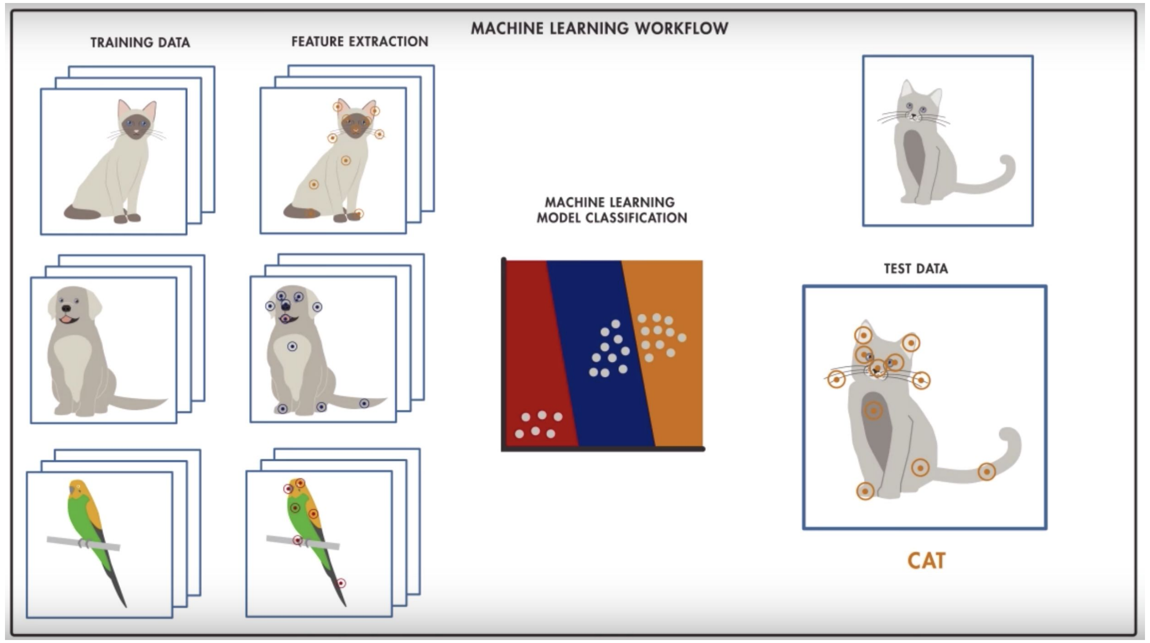


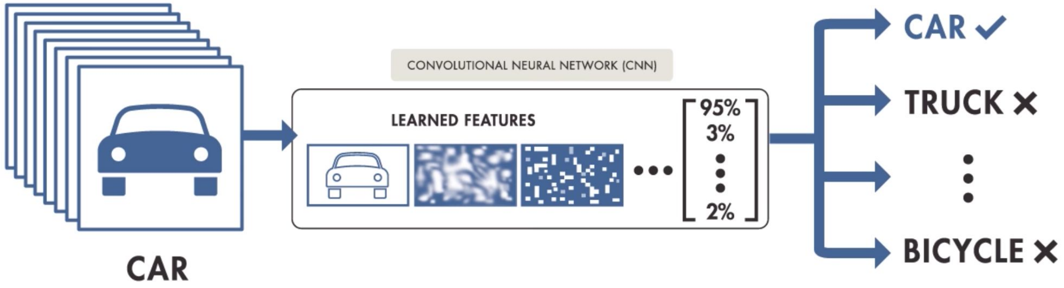
Flowchart shows how different parts of an AI system (Goodfellow et., 2017)

01 Brief history

Machine Learning & Deep learning

One Example

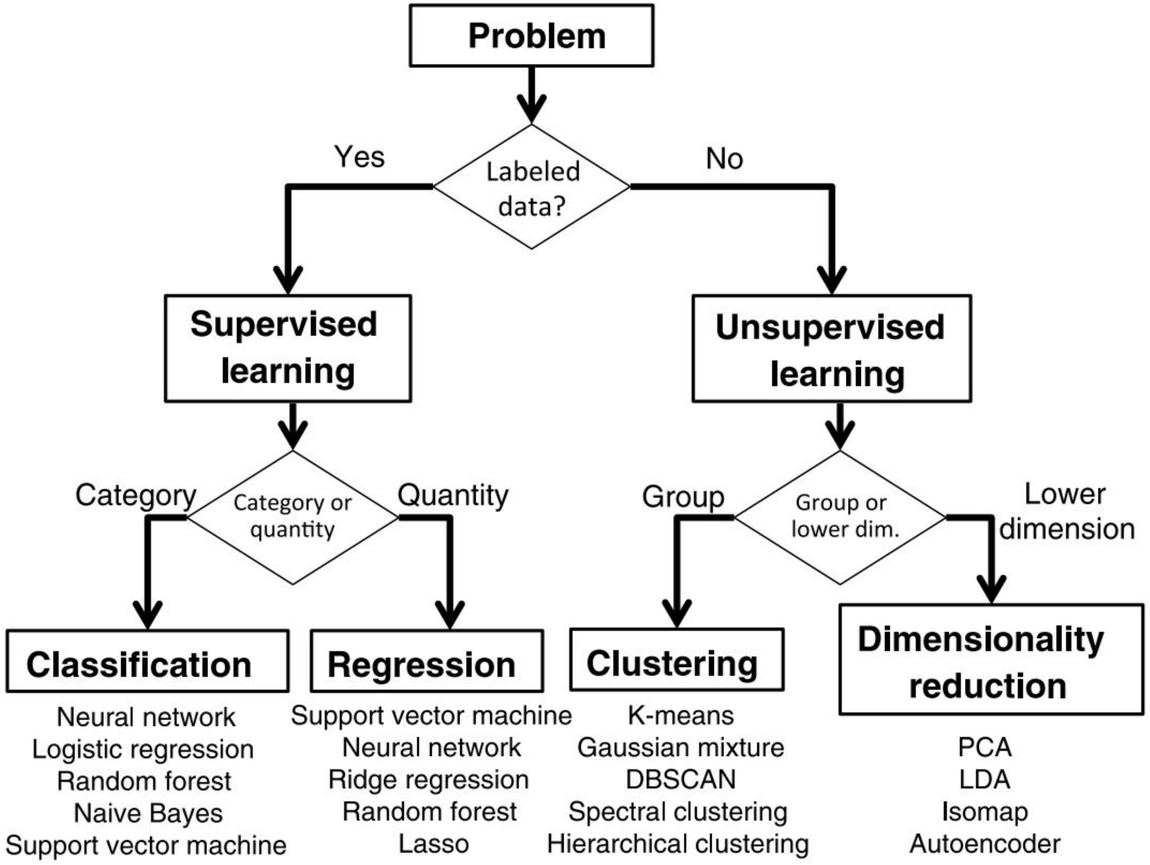




Matlab.mathworks.com

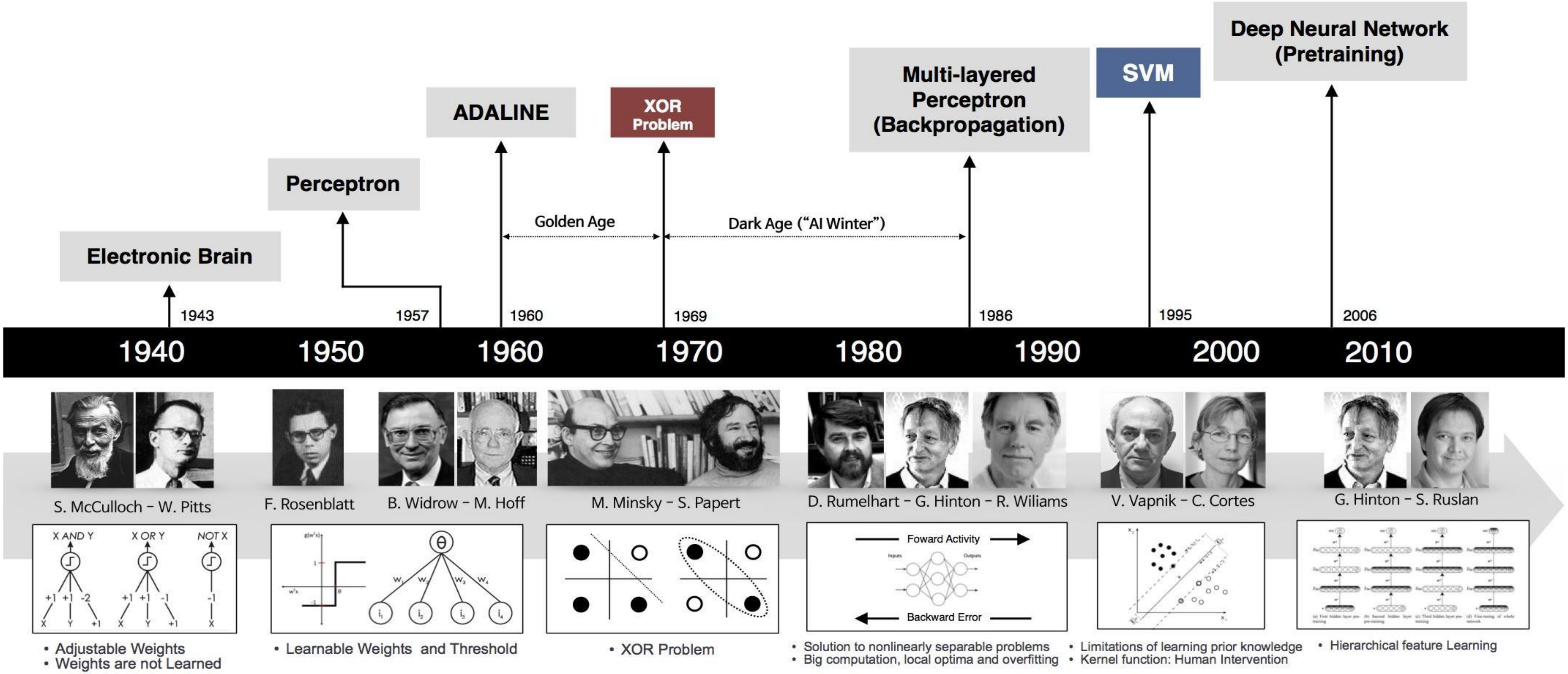
01 Brief history

Machine Learning & Deep learning



Types of machine learning algorithms (Kong et al., 2018)

01 Brief history
Deep learning



01 Brief history

Deep learning

- Google Brain to find Cat Video
- AlexNet: a deep neural network
- …

- VGG-16/19: a very deep neural network
- Adam: very efficient optimization method
- GAN: generative adversarial network
- …

- Tensorflow
- Real-time object detection
- …

- IBM Watson
  beats Humans in Jeopardy
- Feifei Li launched ImageNet
- …

- Deep RNN: Speech recognition
- Dropout: New regularization
- …

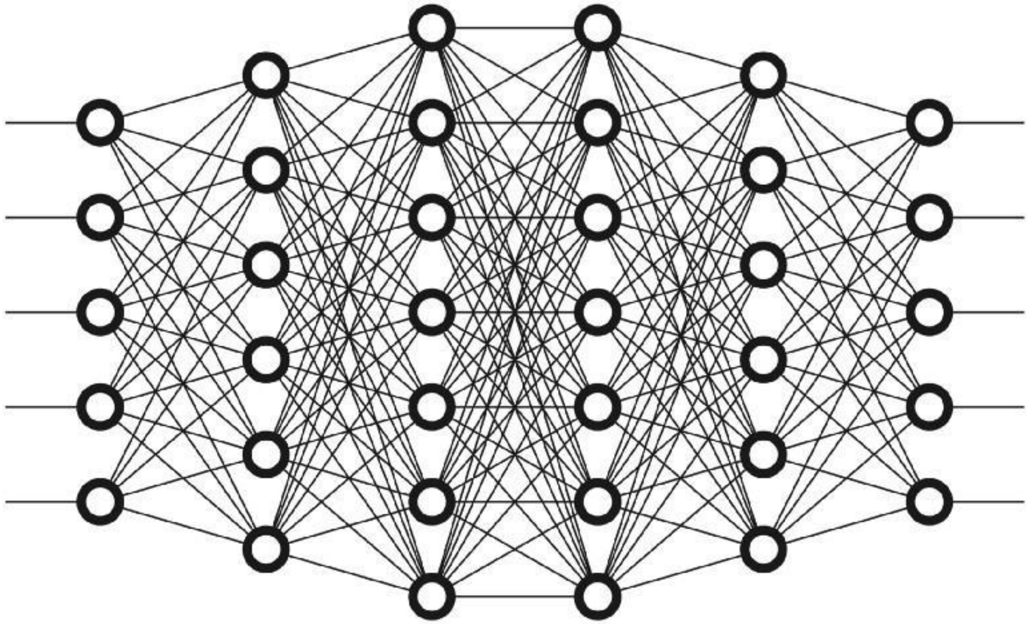
- Deep residual learning
- Batch Normalization
- Convolutional LSTM
- Unet
- Fast R-CNN
- …

- Alpha Go-zero
- Wasserstein GANs
- Cycle-consistent adversarial network
- …

02 Neural Network Basic Units

Different Housing materials

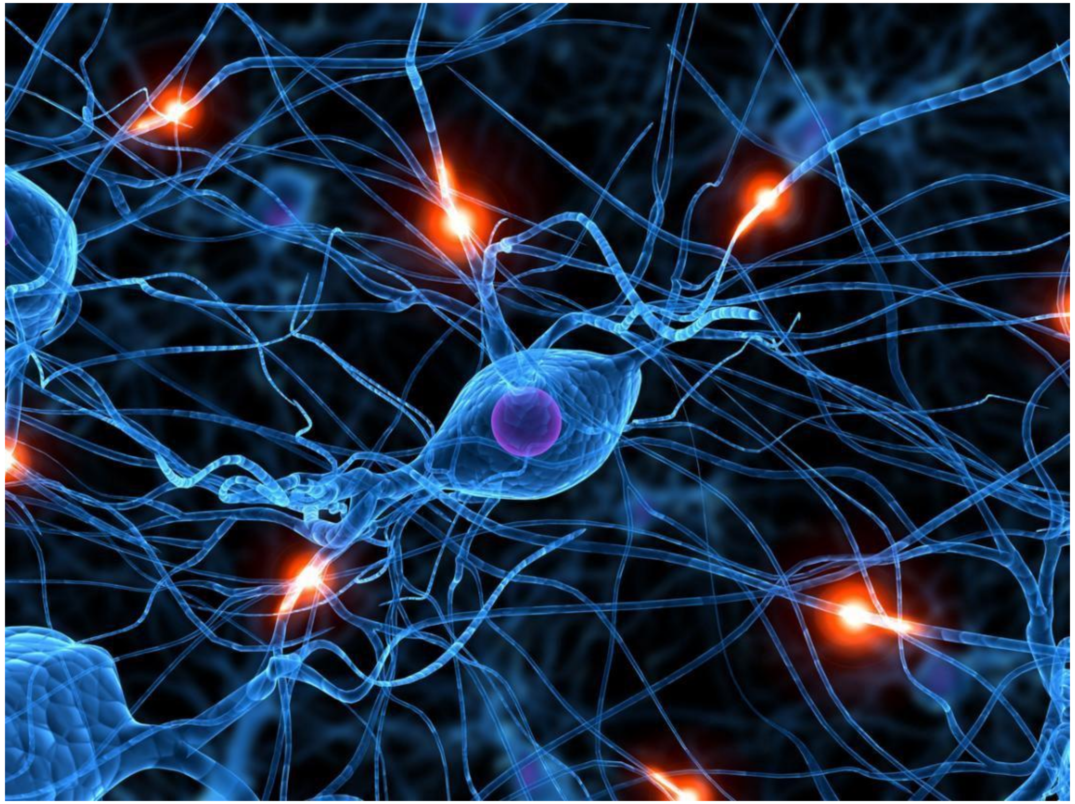
02 Neural Network Basic Units



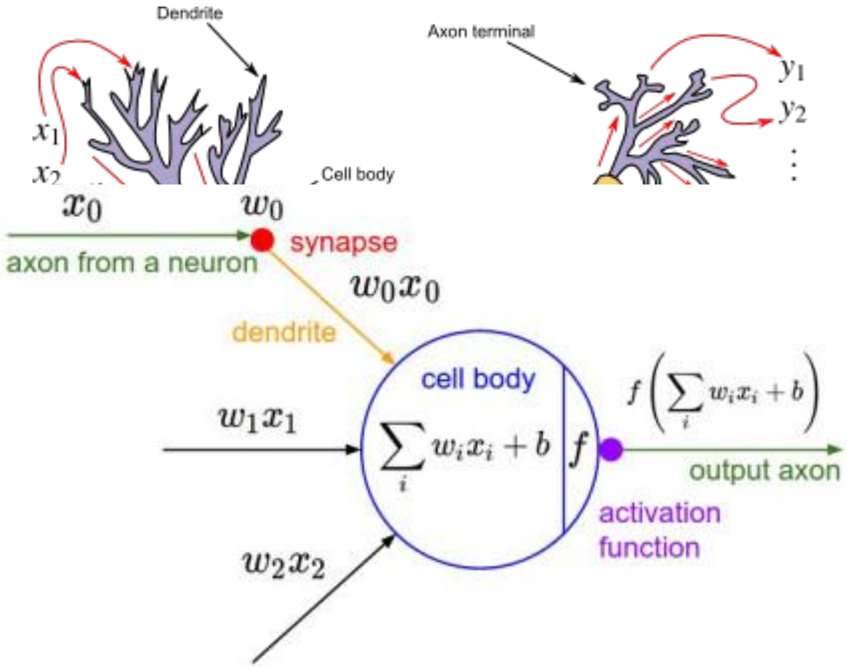
02 Neural Network Basic Units

02 Neural Network

Perceptron



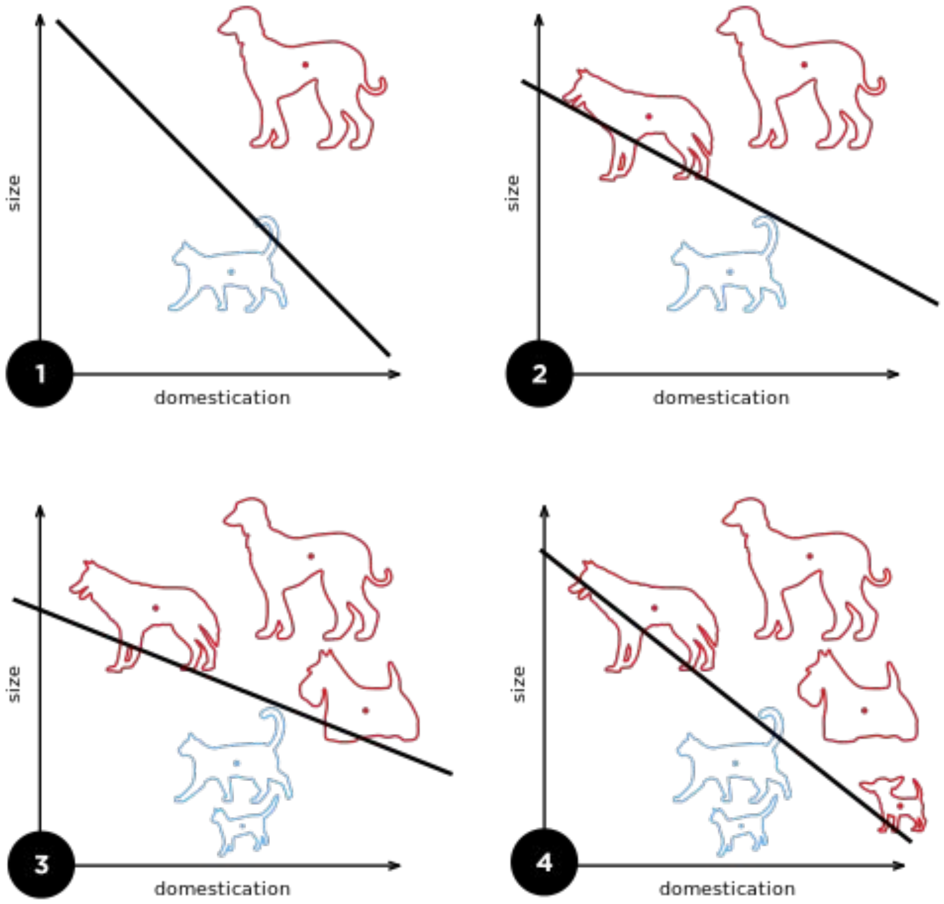
Neurons in brain



02 Neural Network

Perceptron

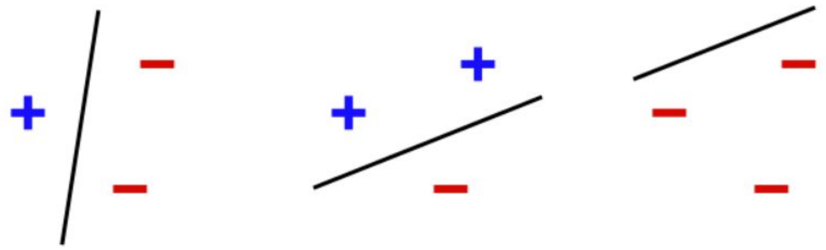
Linear classification



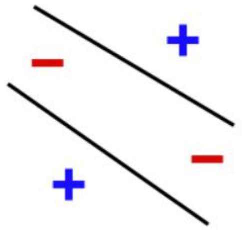
02 Neural Network

Perceptron

Linearly separable

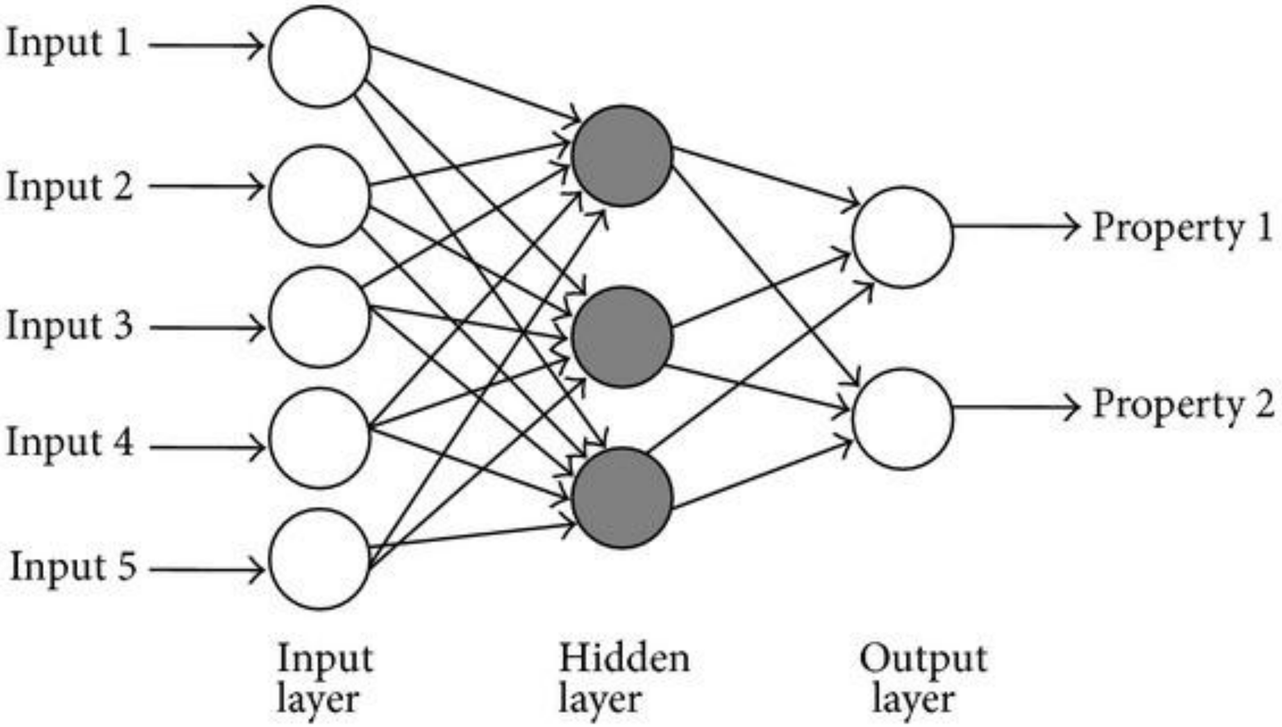


Non-linearly separable
(XOR problem)



02 Neural Network

Multi-layered Perceptron

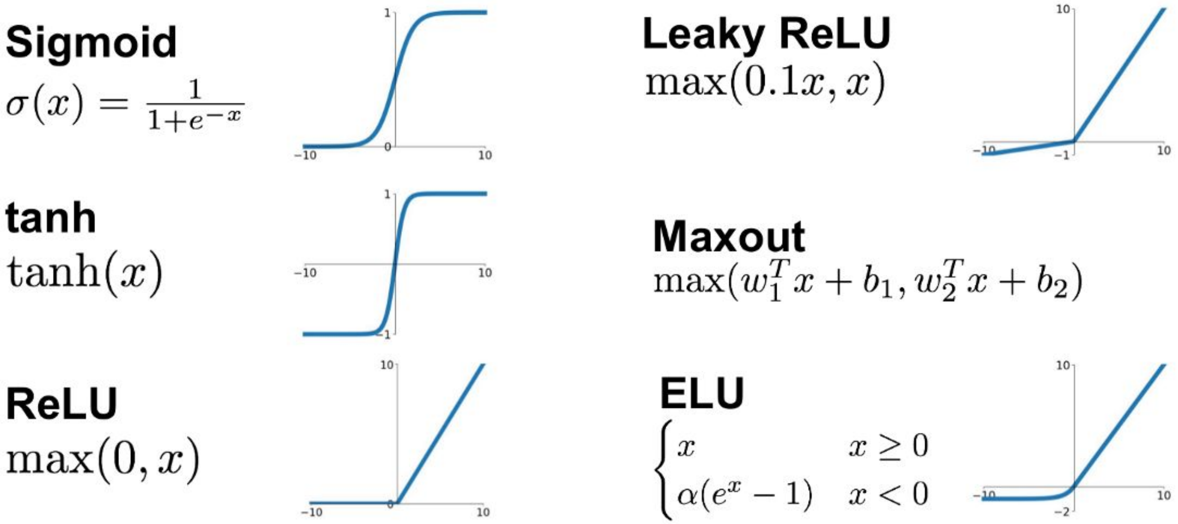


$$\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$$

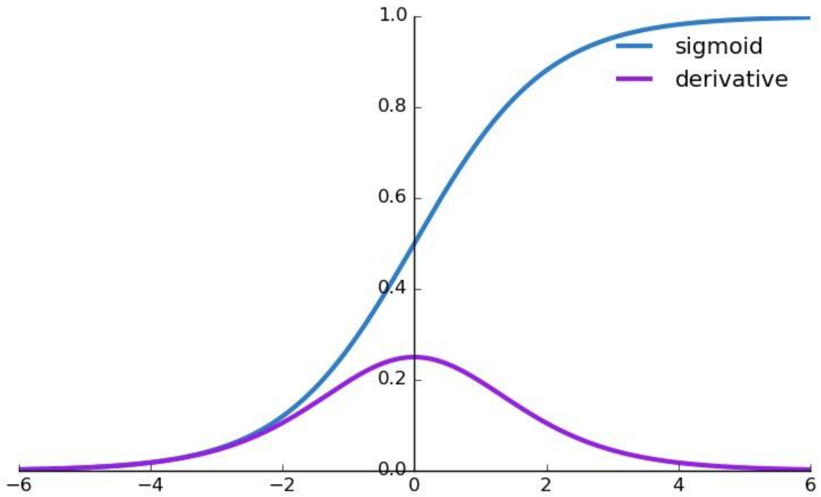
(Goodfellow et al., 2017)

02 Neural Network

Activation function



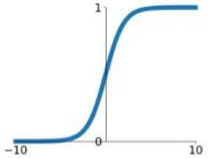
Sigmoid function



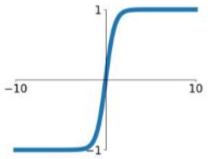
02 Neural Network

Activation function

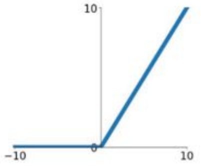
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



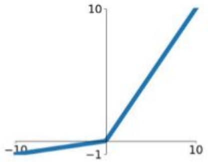
tanh
$\tanh(x)$



ReLU
$\max(0, x)$

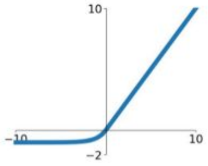


Leaky ReLU
$\max(0.1x, x)$

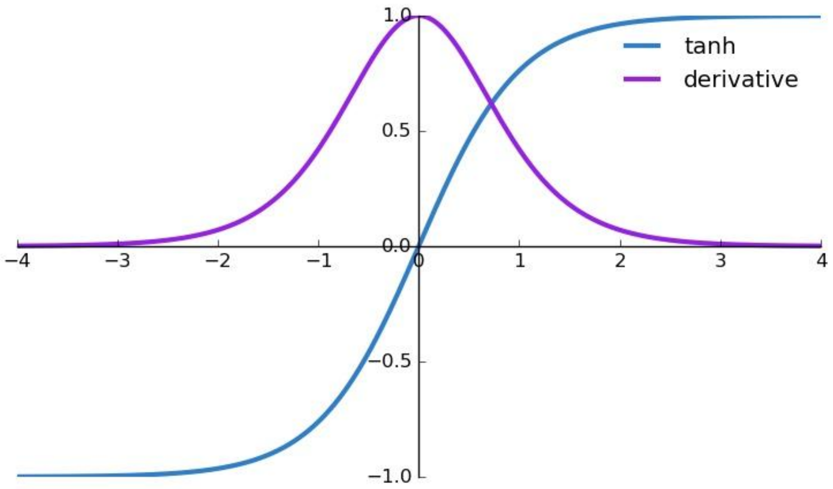


Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$



Tanh function

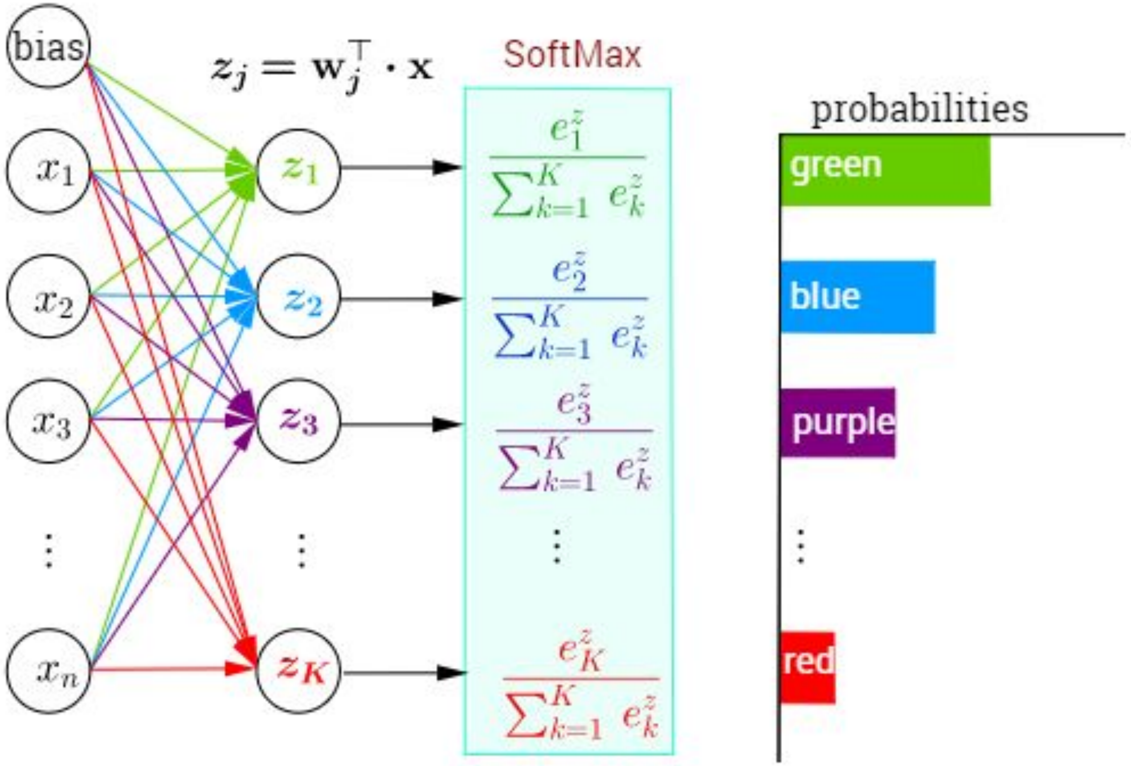


Only for 2-class classification !

02 Neural Network

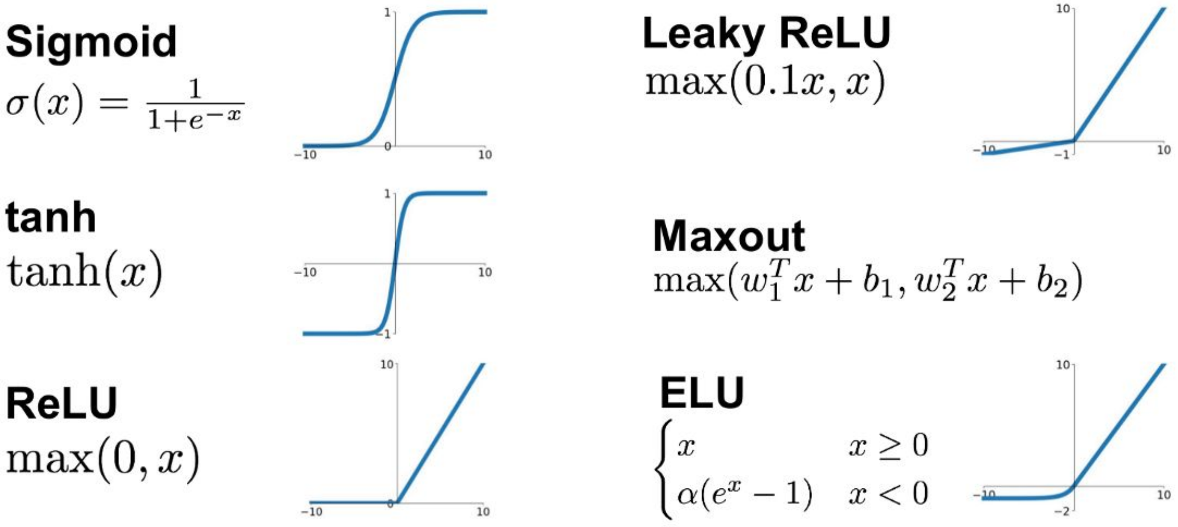
Activation function

Softmax function for multi-class classification

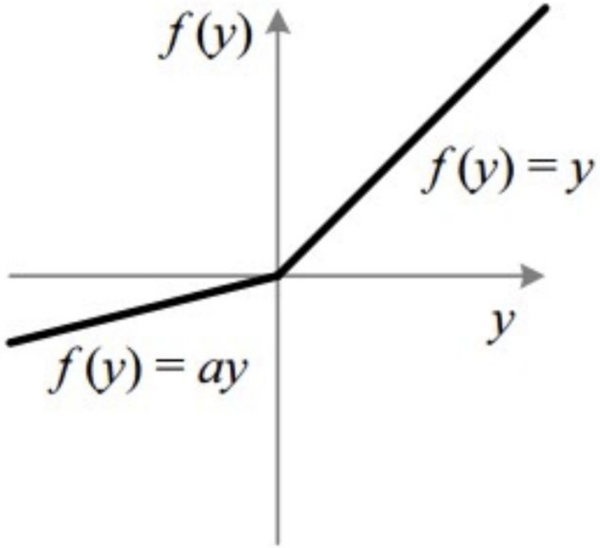


02 Neural Network

Activation function



LeakyReLU function
ReLU function



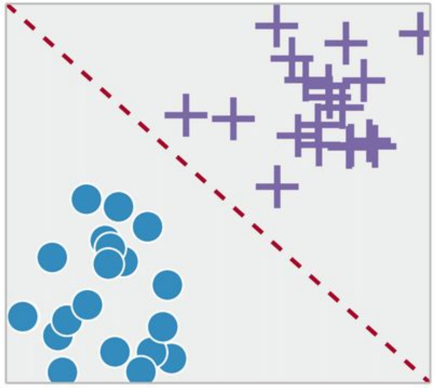
02 Neural Network

Cost function

Classification problems:

Cross entropy:

$$J(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(\boldsymbol{y} \mid \boldsymbol{x}).$$



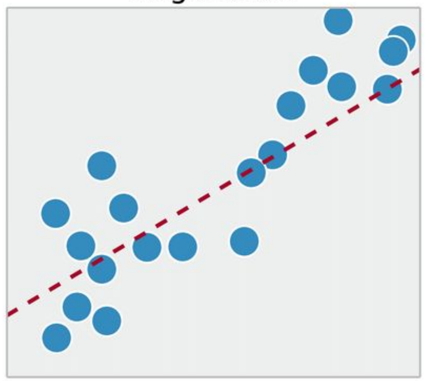
Regression problems:

Mean squared error:

$$f^* = \arg\min_f \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\text{data}}} ||\boldsymbol{y} - f(\boldsymbol{x})||^2$$

Mean absolute error:

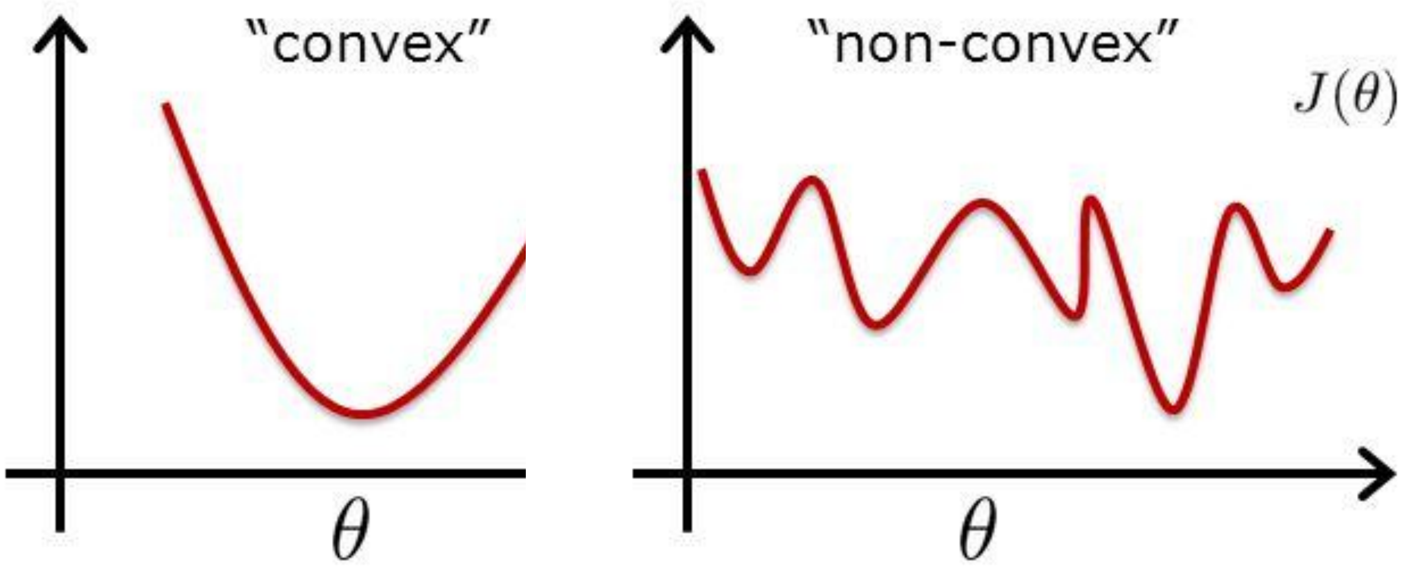
$$f^* = \arg\min_f \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\text{data}}} ||\boldsymbol{y} - f(\boldsymbol{x})||_1$$



02

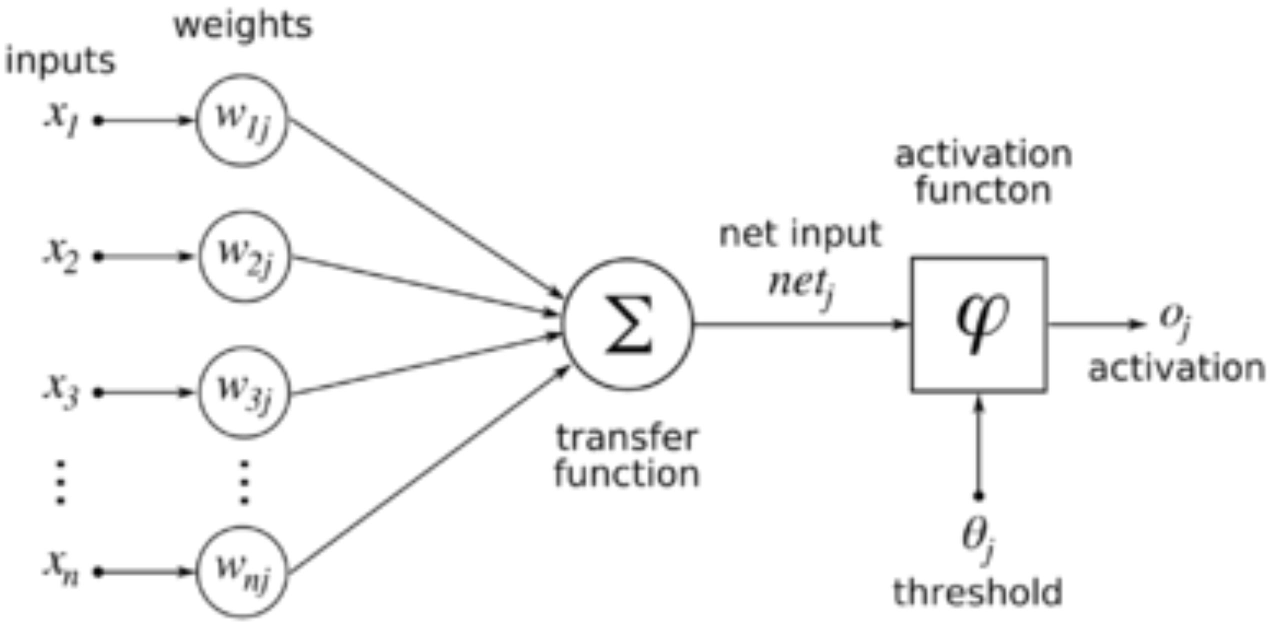
Neural Network

Cost function



02 Neural Network

Back propagation

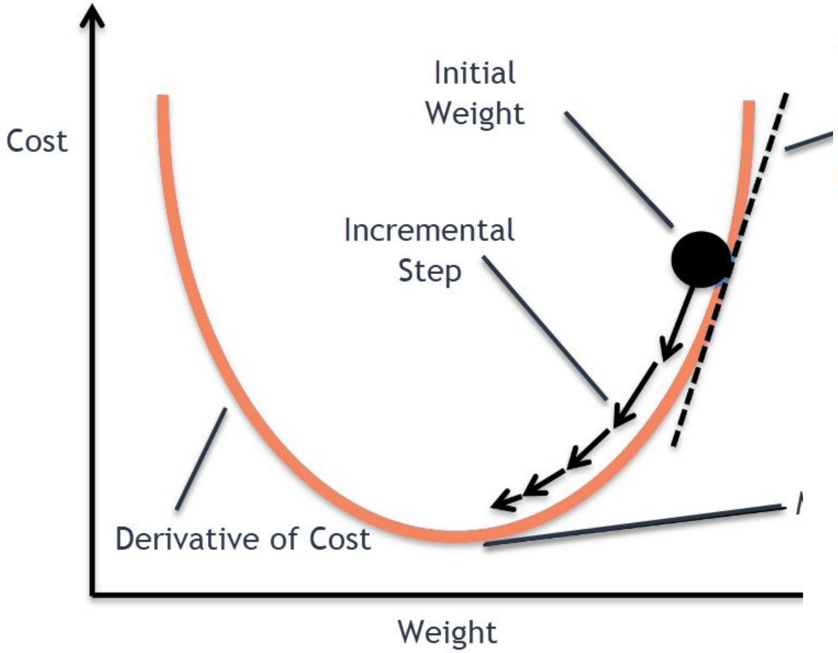


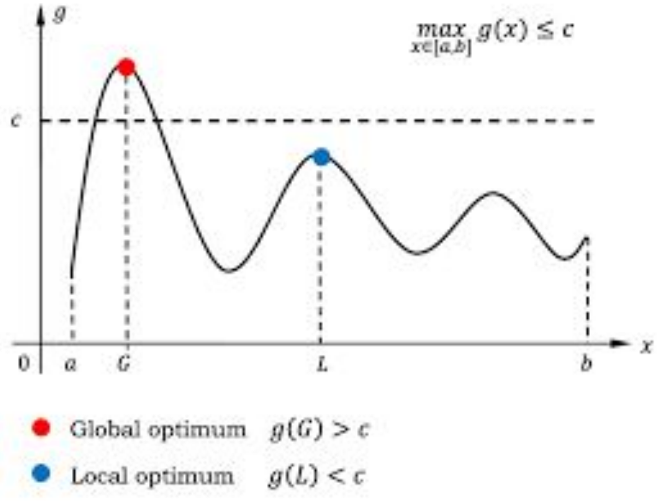
Chain rule:

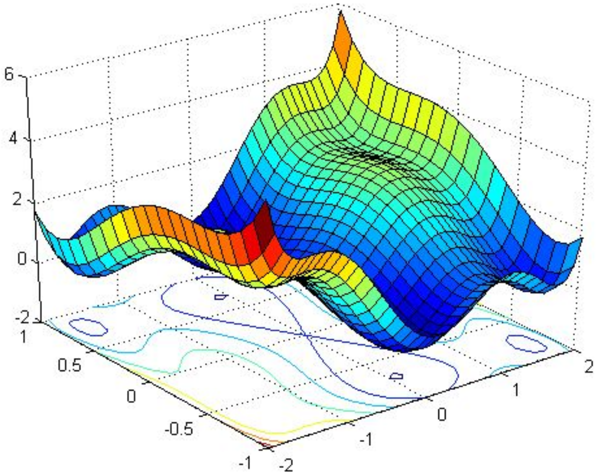
$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial \mathrm{net}_j}\frac{\partial \mathrm{net}_j}{\partial w_{ij}}$$

$$\frac{\partial \mathrm{net}_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left(\sum_{k=1}^{n} w_{kj} o_k\right) = \frac{\partial}{\partial w_{ij}} w_{ij} o_i = o_i.$$

02 Neural Network
Optimization method







02 Neural Network

Optimization method

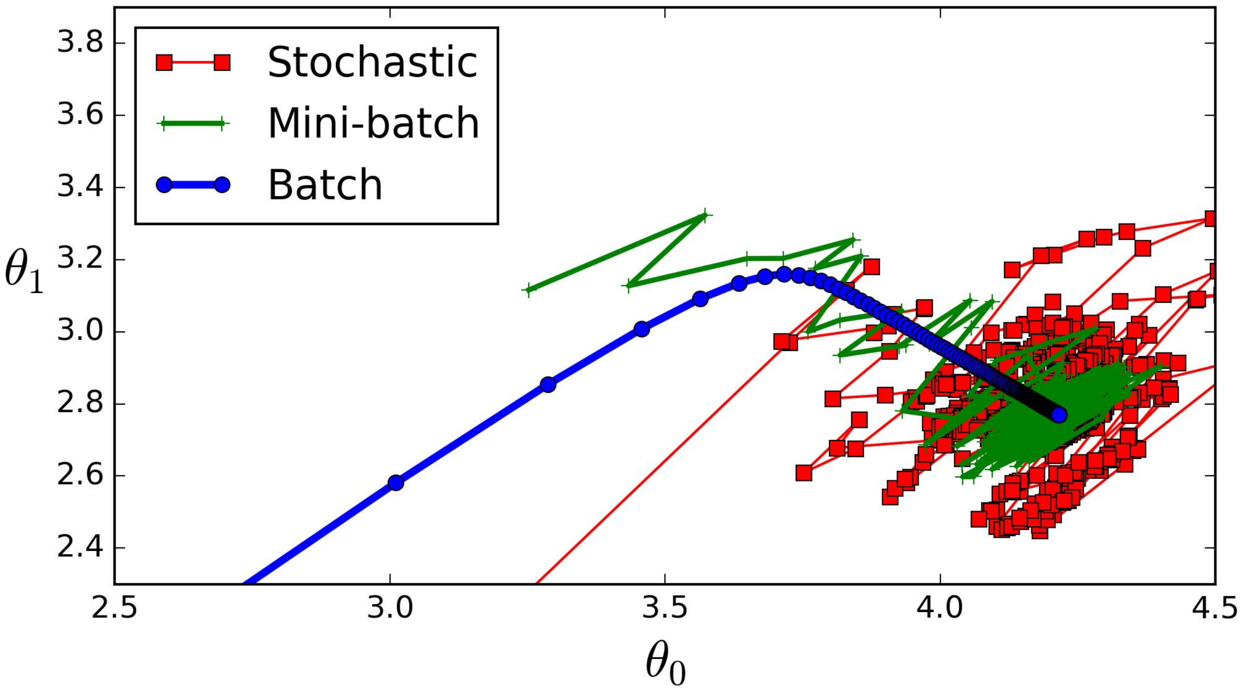
Stochastic gradient descent (SGD)

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\eta} \cdot \nabla J(\boldsymbol{\theta}; x(i); y(i))$$

Slow + unstable!

Batch/Minibatch SGD

Difficult to choose learning rate!



02 Neural Network

Optimization method

Momentum SGD

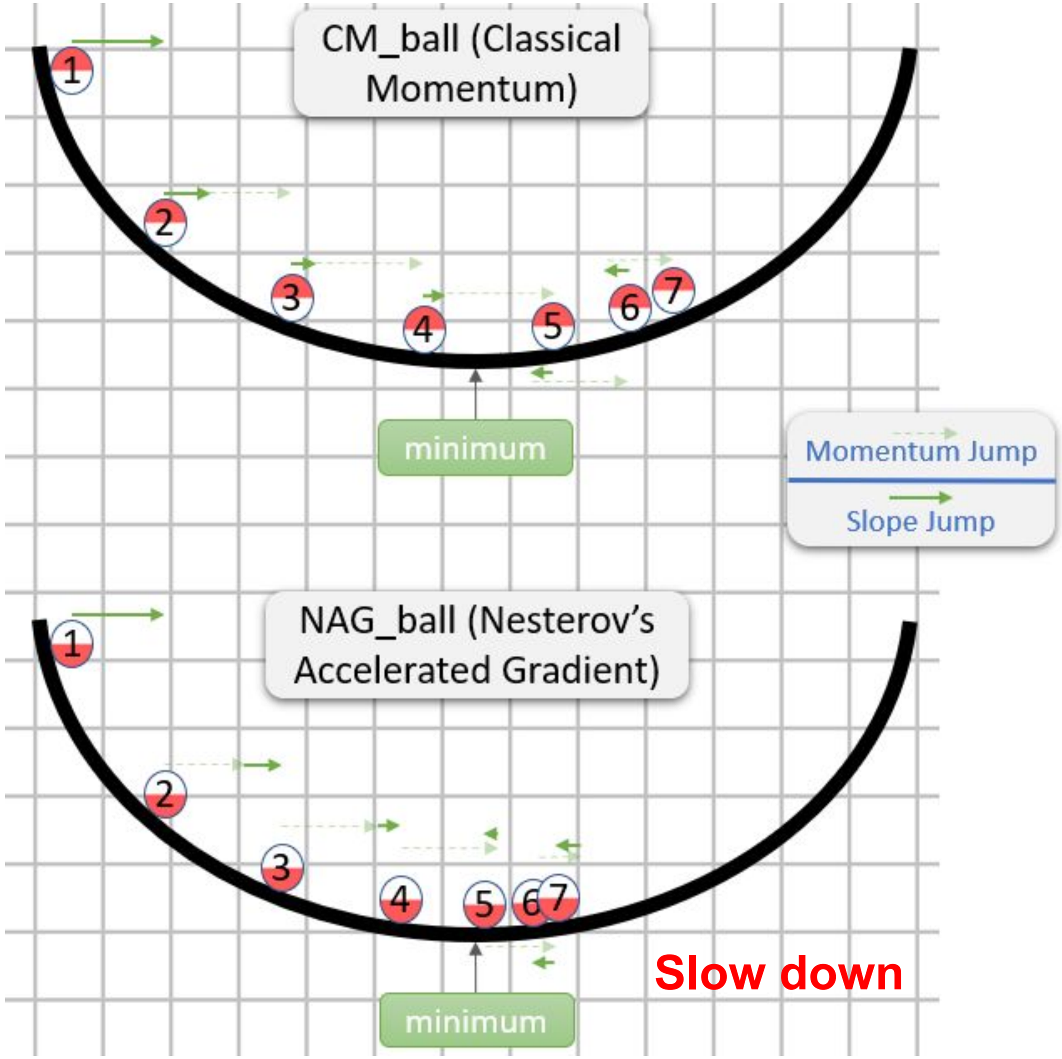
$$V(t)=\gamma V(t-1)+\eta\nabla J(\theta).$$

Blindly!

Nesterov accelerated gradient (NAG)

$$V(t)=\gamma V(t-1)+\eta\nabla J(\,\theta-\gamma V(t-1)\,)$$

Smarter!



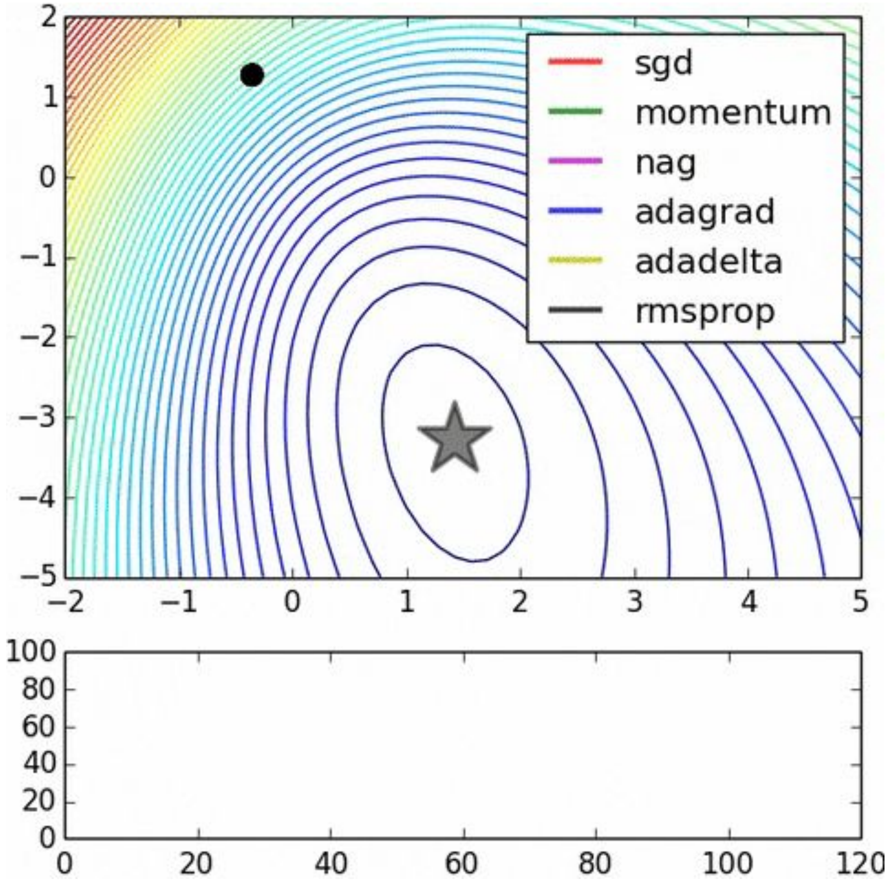
02 Neural Network

Optimization method

Adagrad

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\varepsilon I + diag(G_t)}} \cdot g_t, \quad G_t = \sum_{\tau=1}^{t} g_\tau g_\tau^\top.$$

Allows the learning rate to adapt on the parameters, but learning rate always decreases and decays



Adaptive Moment Estimation (Adam)

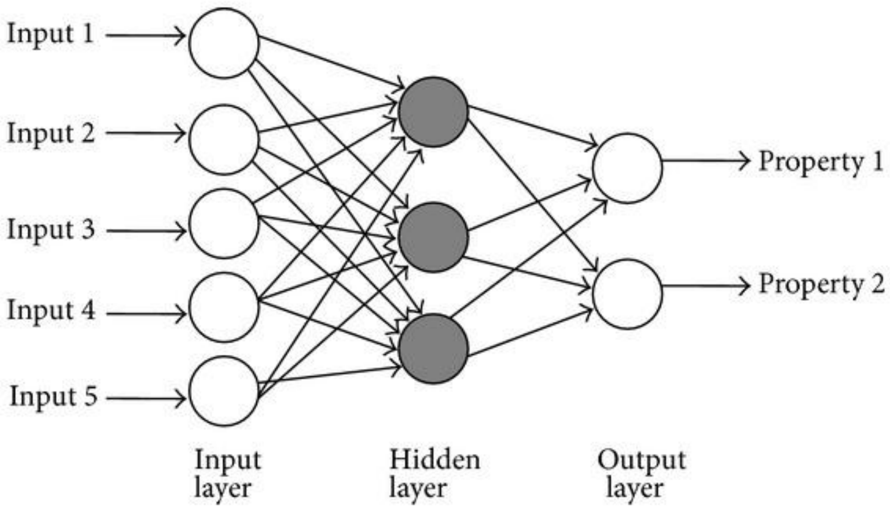
$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

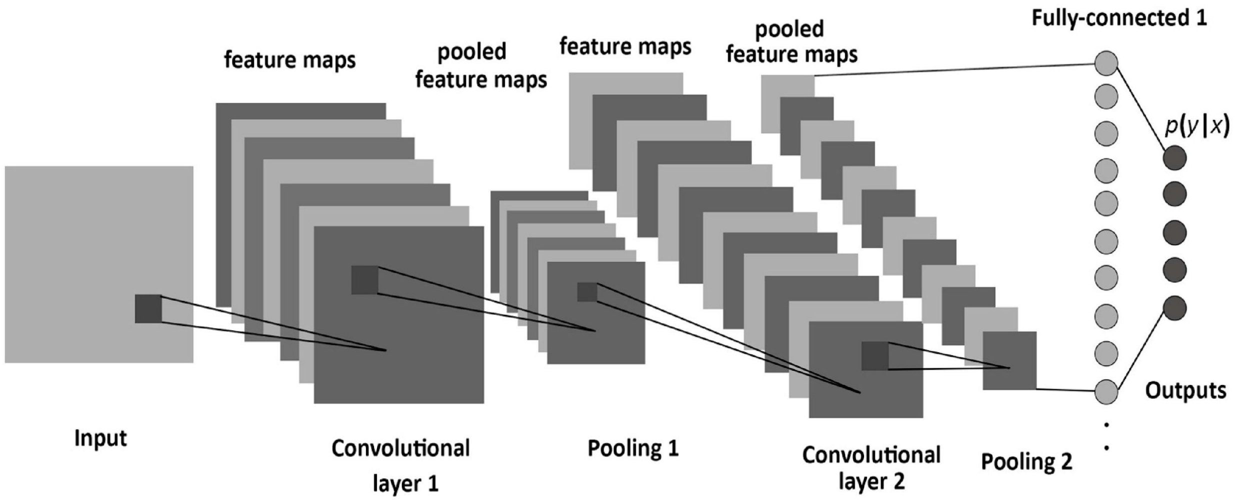
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$ Mean of gradients

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$ Uncentered variances of gradients (similar to momentum)

Straightforward to implement/computational efficient

02 Neural Network Convolution



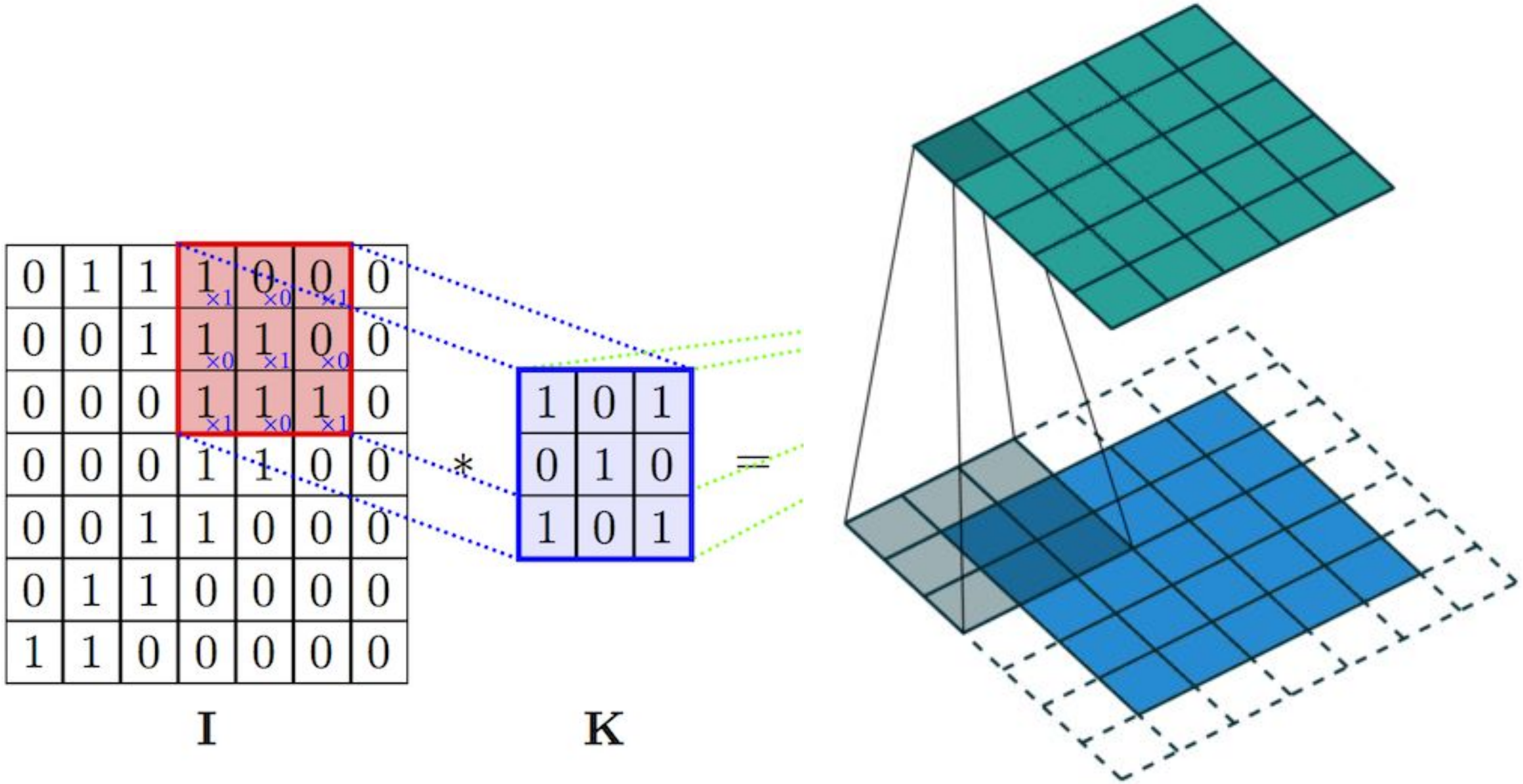


Question: What are differences between above networks?

(Goodfellow et al., 2017)

02 Neural Network

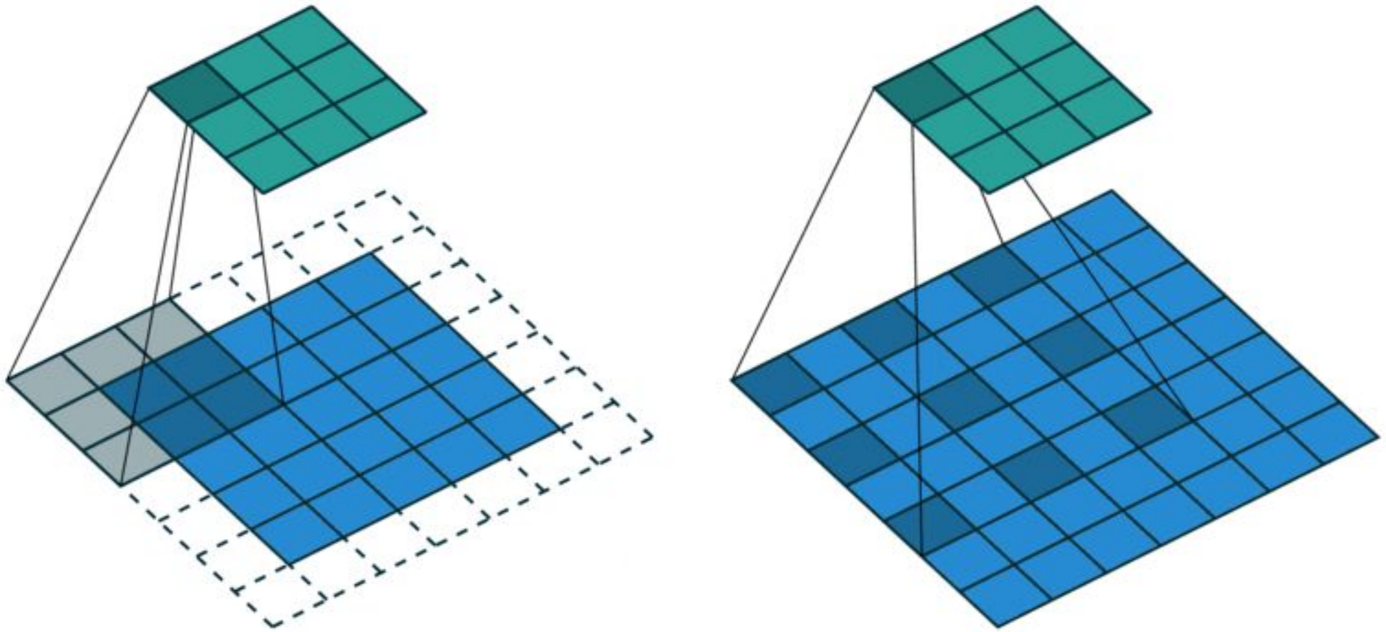
Convolution



02

Neural Network

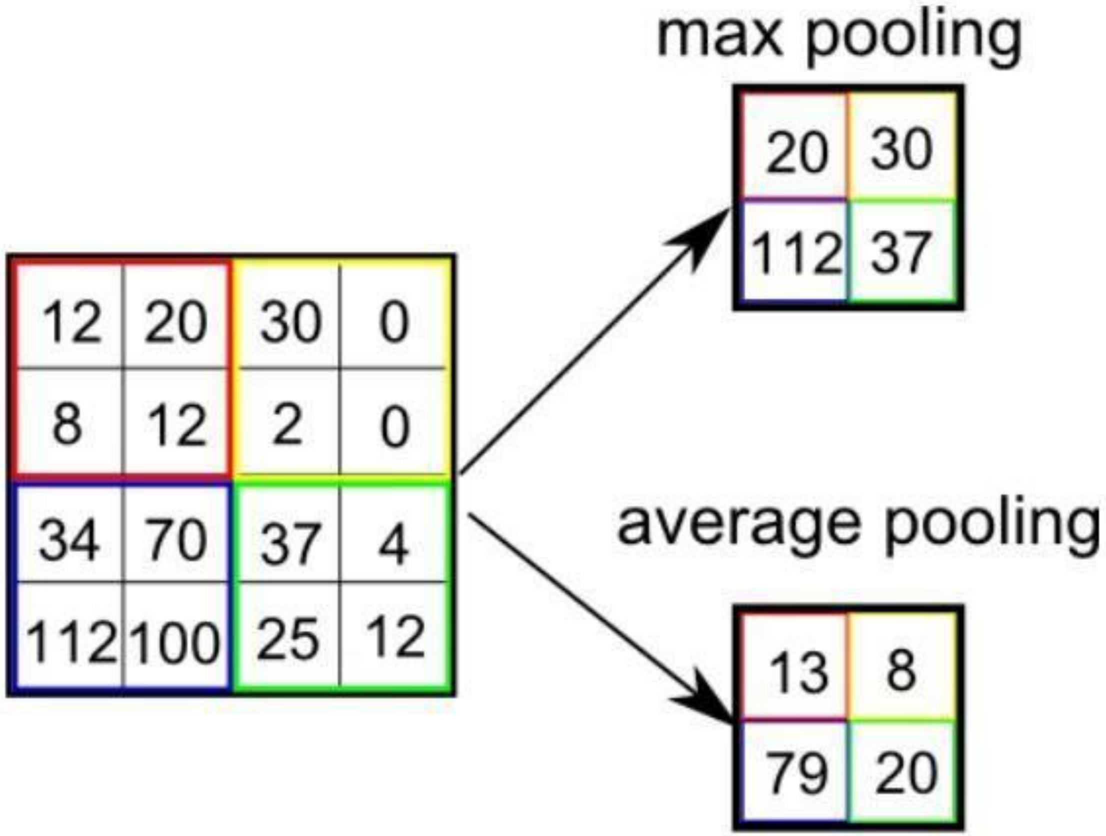
Dilated convolution



02

Neural Network

Pooling/down-sampling

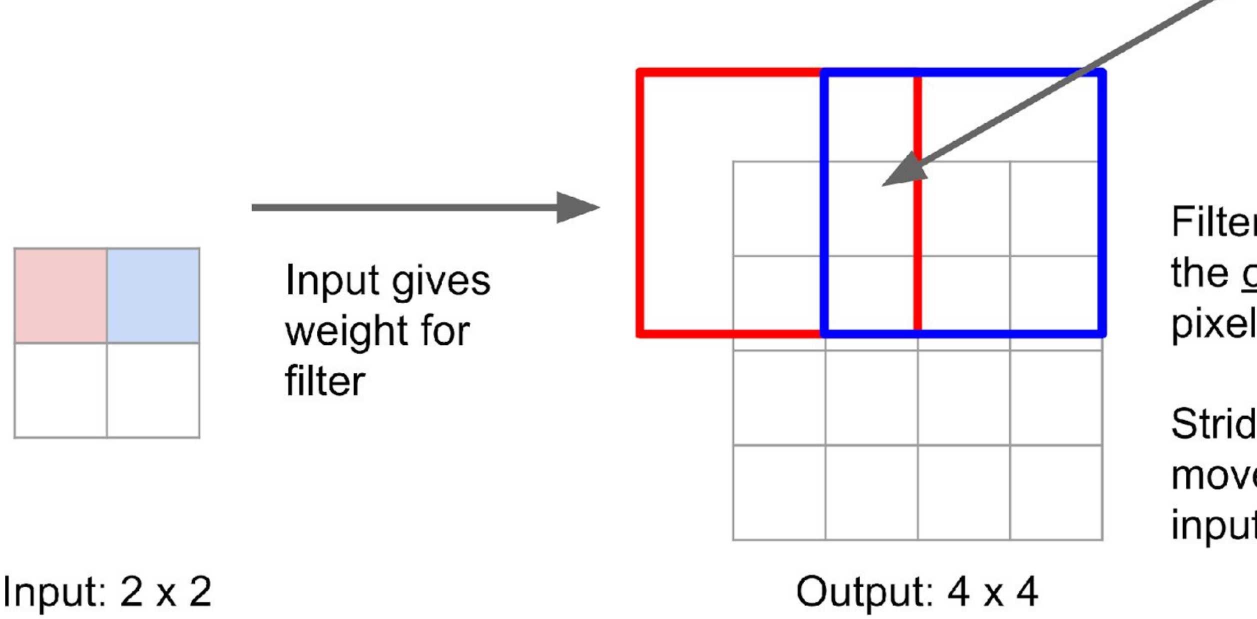


Up-sampling

Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

3 x 3 **transpose** convolution, stride 2 pad 1

Sum where output overlaps

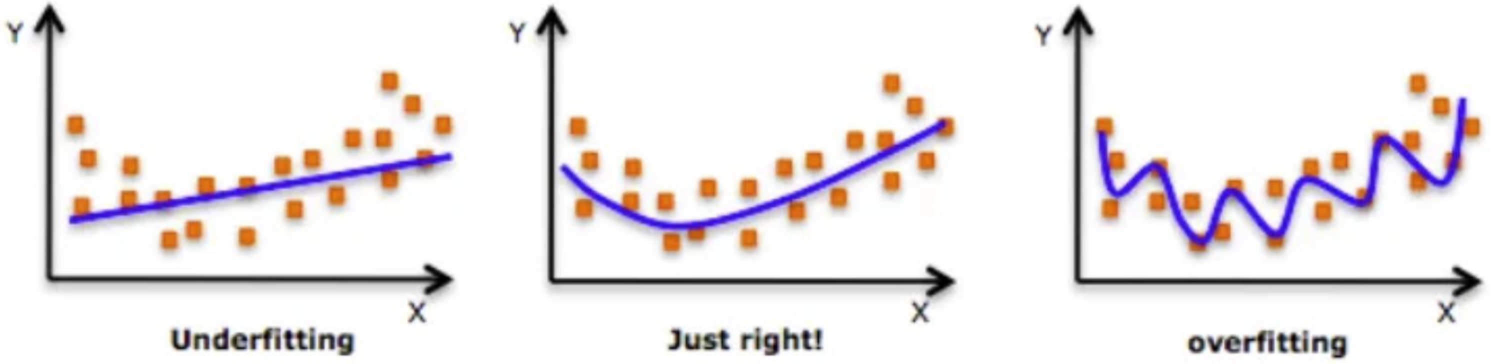


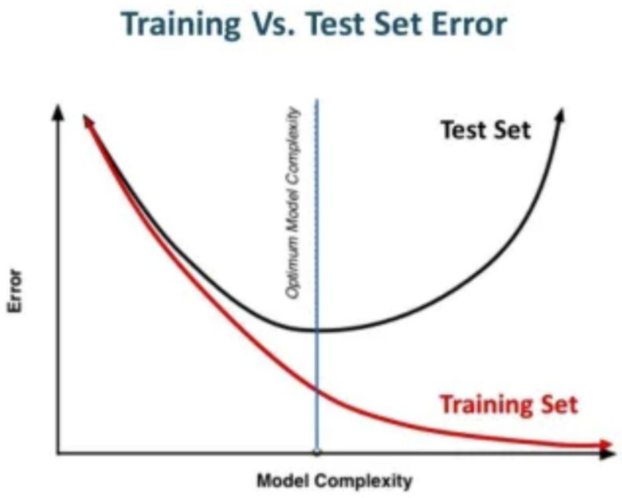
Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

02 Neural Network

Overfitting & regularization





(Sliderplayer)

02 Neural Network

Overfitting & regularization

L1 & L2 parameter regularizations

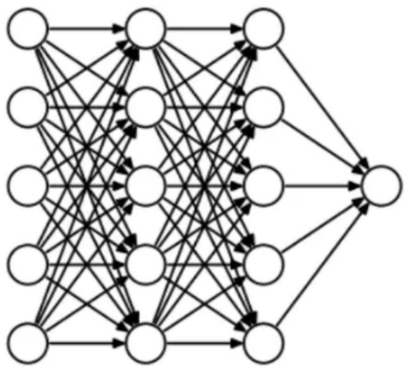
$$Cost\ function\ =\ Loss\ +\ \boxed{\frac{\lambda}{2m} * \sum \|w\|}$$

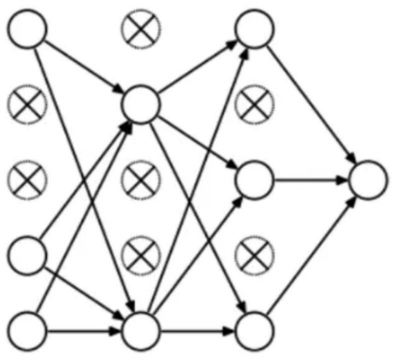
Decay weights to zero

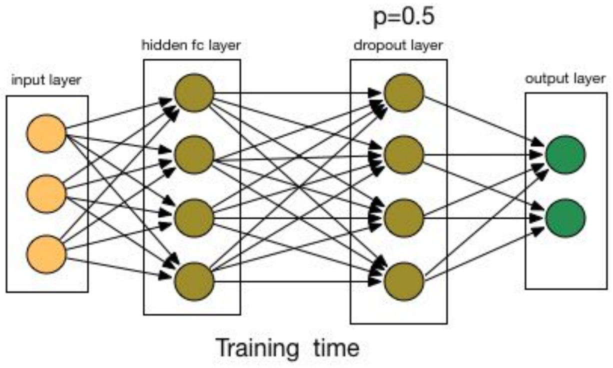
$$Cost\ function\ =\ Loss\ +\ \boxed{\frac{\lambda}{2m} * \sum \|w\|^2}$$

Decay weights close to zero

Dropout



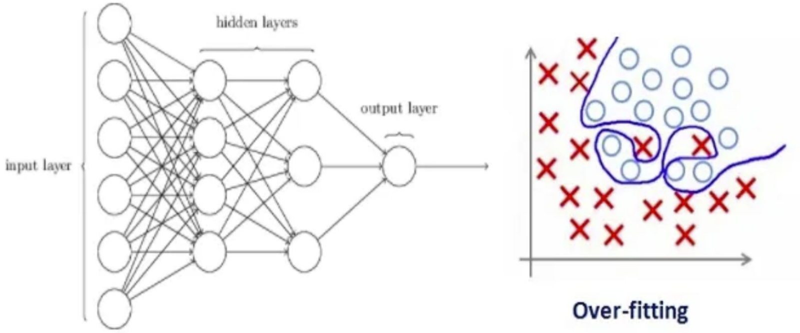




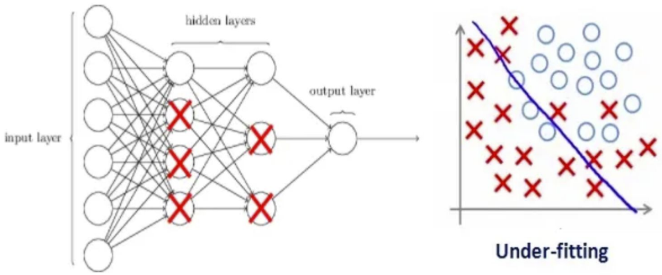
02 Neural Network

Overfitting & regularization

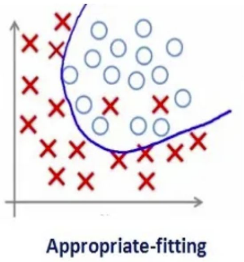
No regularization



Large regularization



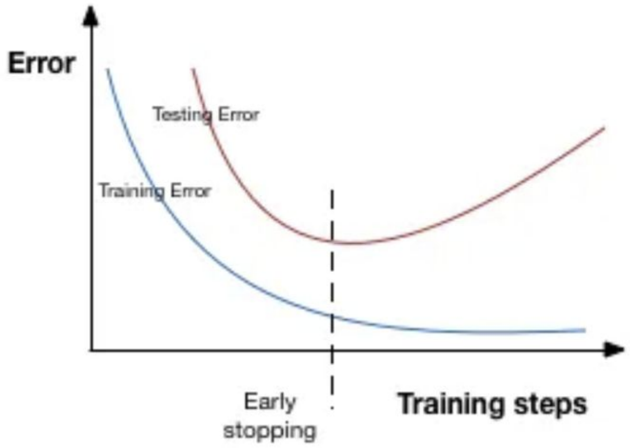
Appropriate regularization



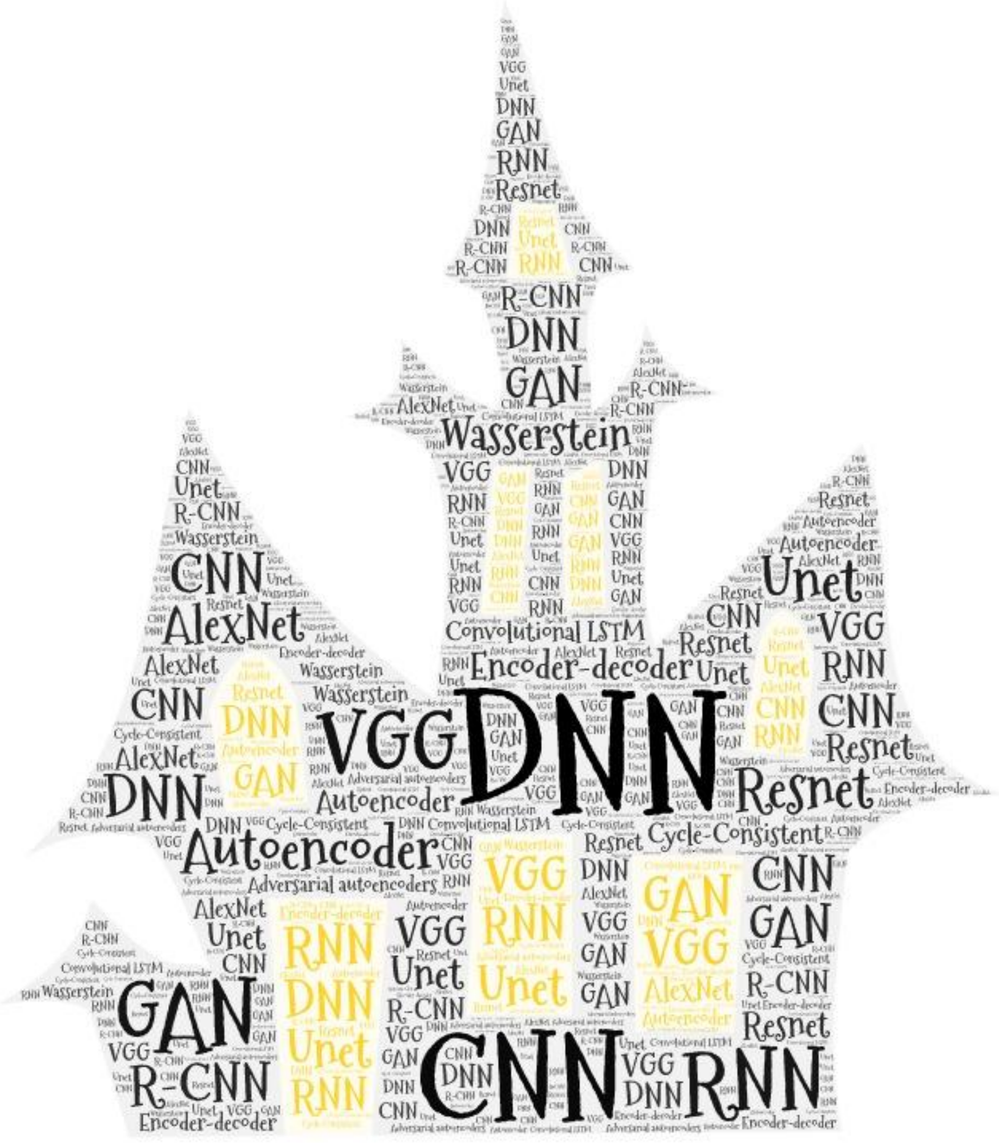
Overfitting & regularization

Data Augmentation

Early stopping

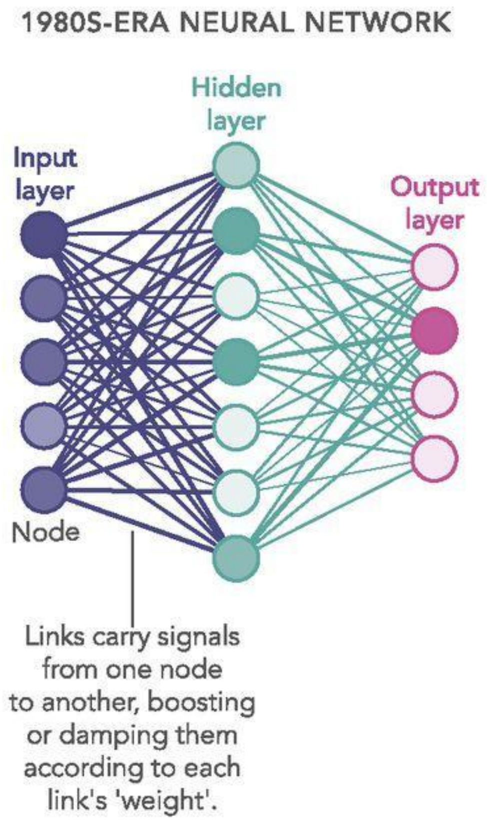


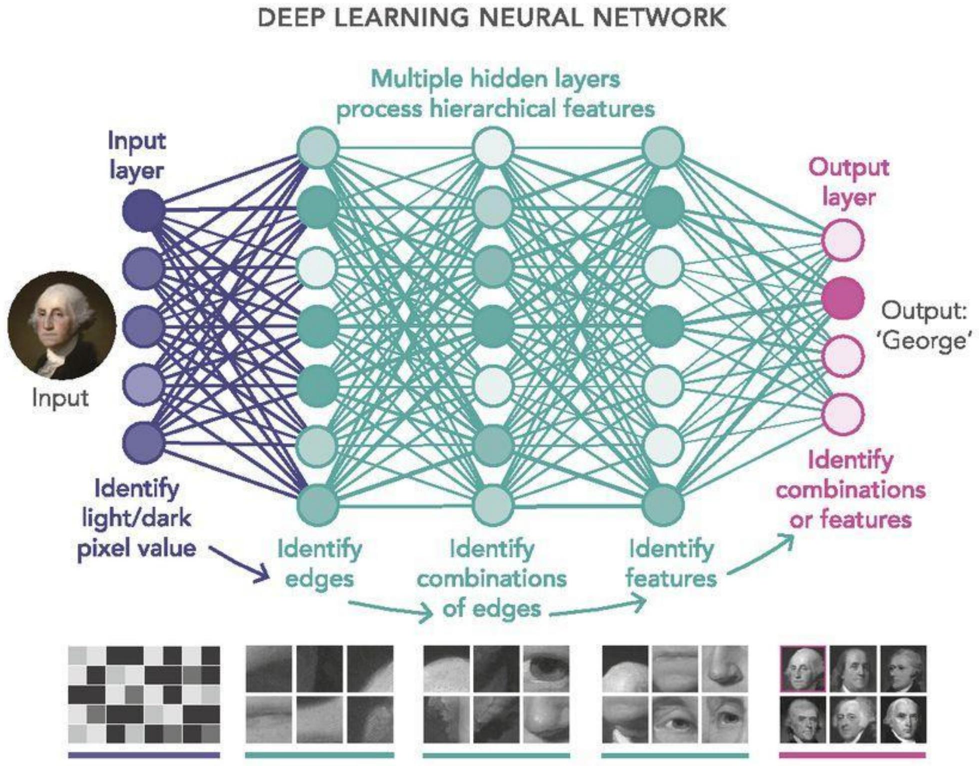
03 Neural Network Architecture



03 Neural Network

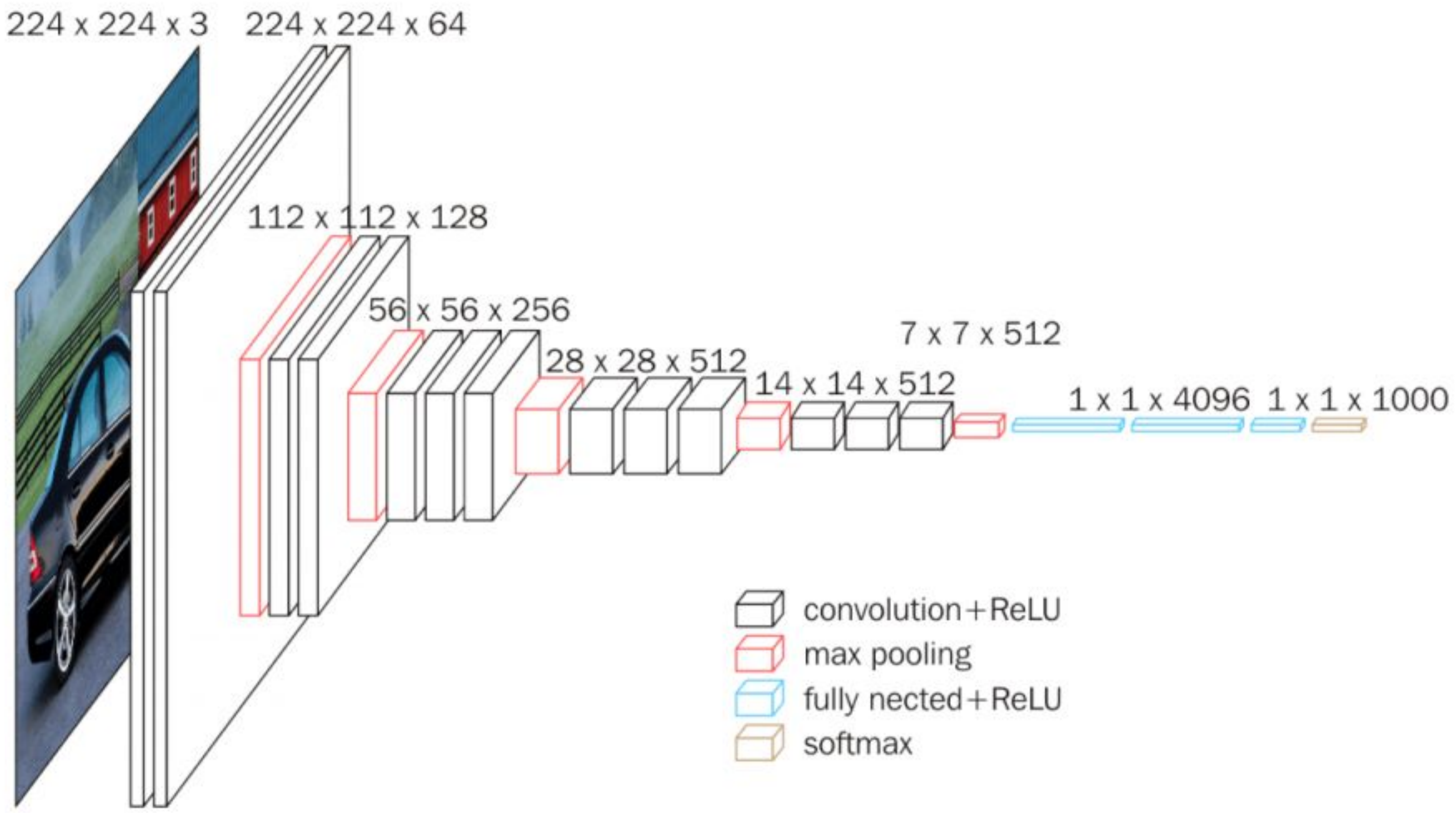
Architecture - DNN





03 Neural Network

Architecture - CNN



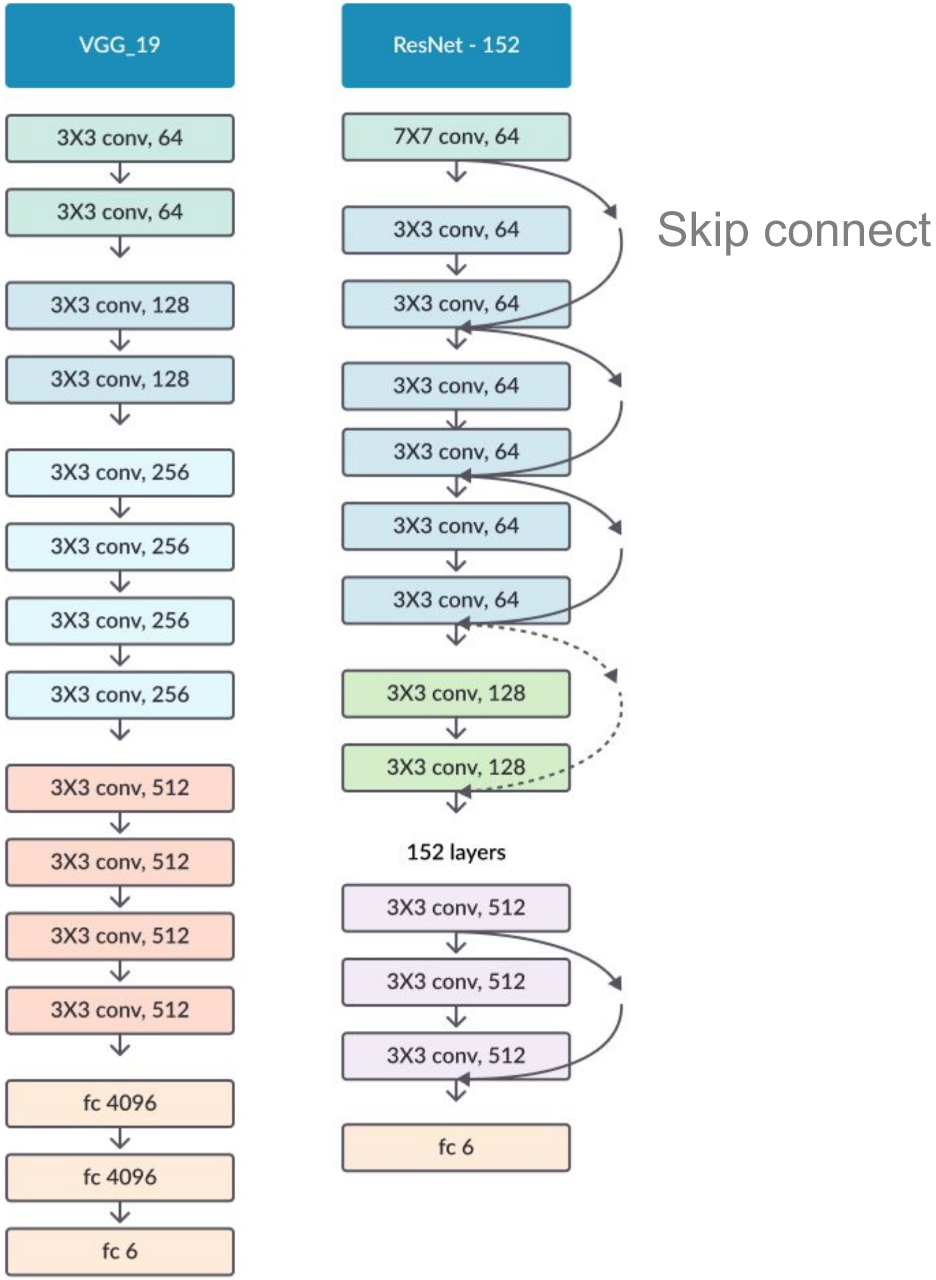
(Simonyan and Zisserman, 2014)

03 Neural Network

Architecture - ResNet

Advantages:

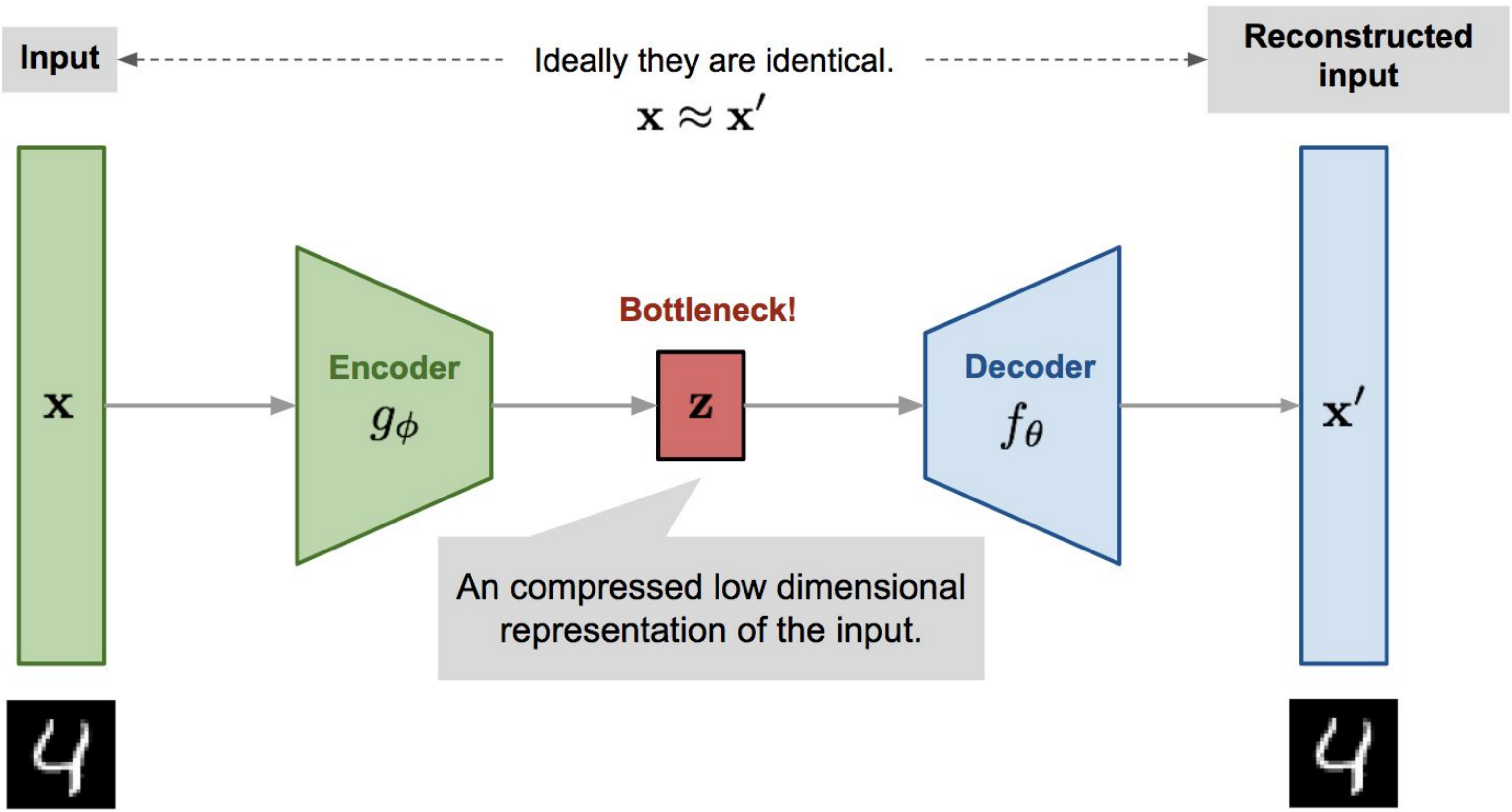
1. To avoid problem of vanishing gradients;
2. Effectively simplifies the network and speed learning;
3. Less vulnerable to perturbations in data.



03

Neural Network

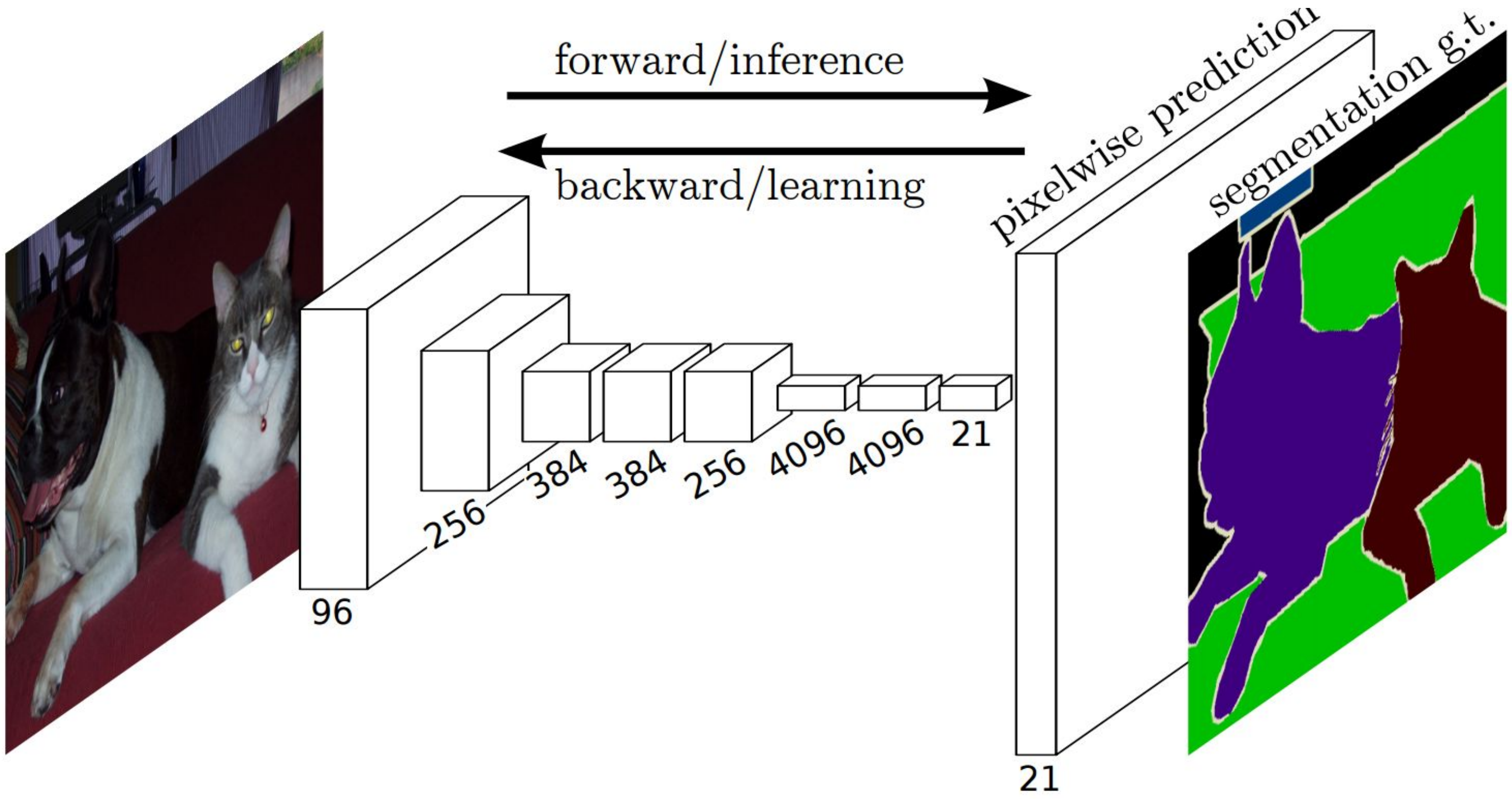
Architecture - autoencoder



03 Neural Network

Architecture – encoder-decoder

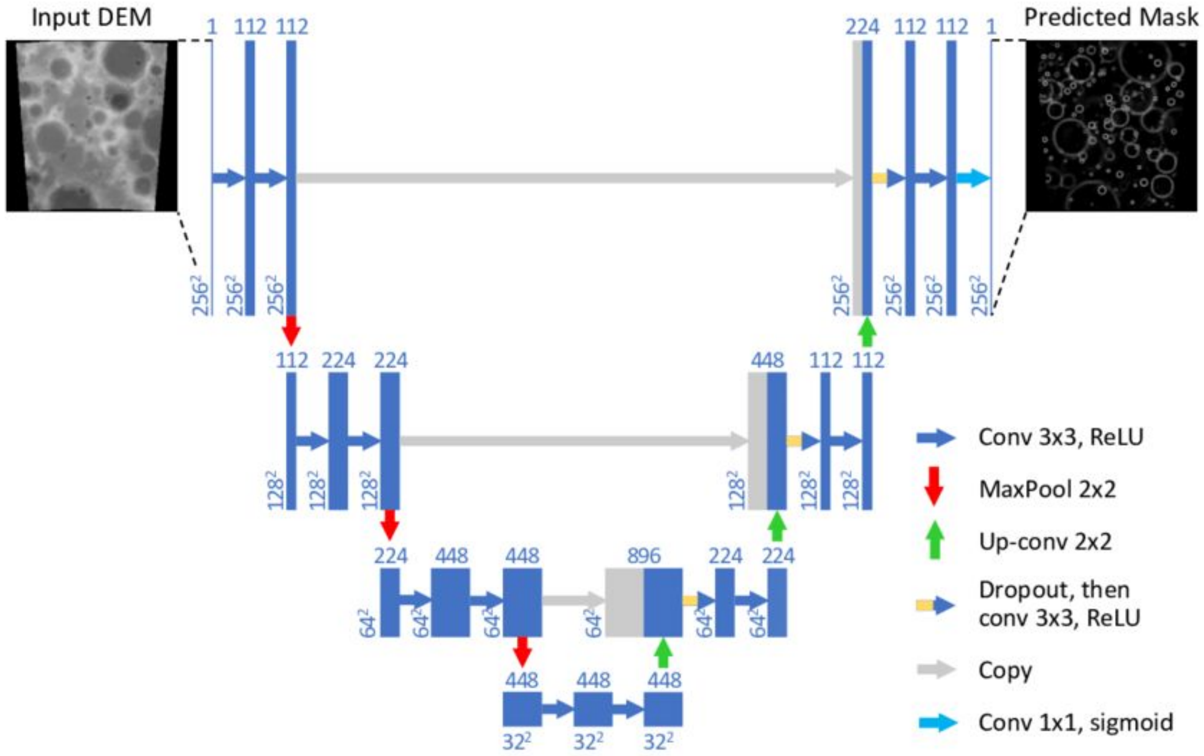
Fully convolutional neural network for Semantic Segmentation



03 Neural Network

Architecture – encoder-decoder

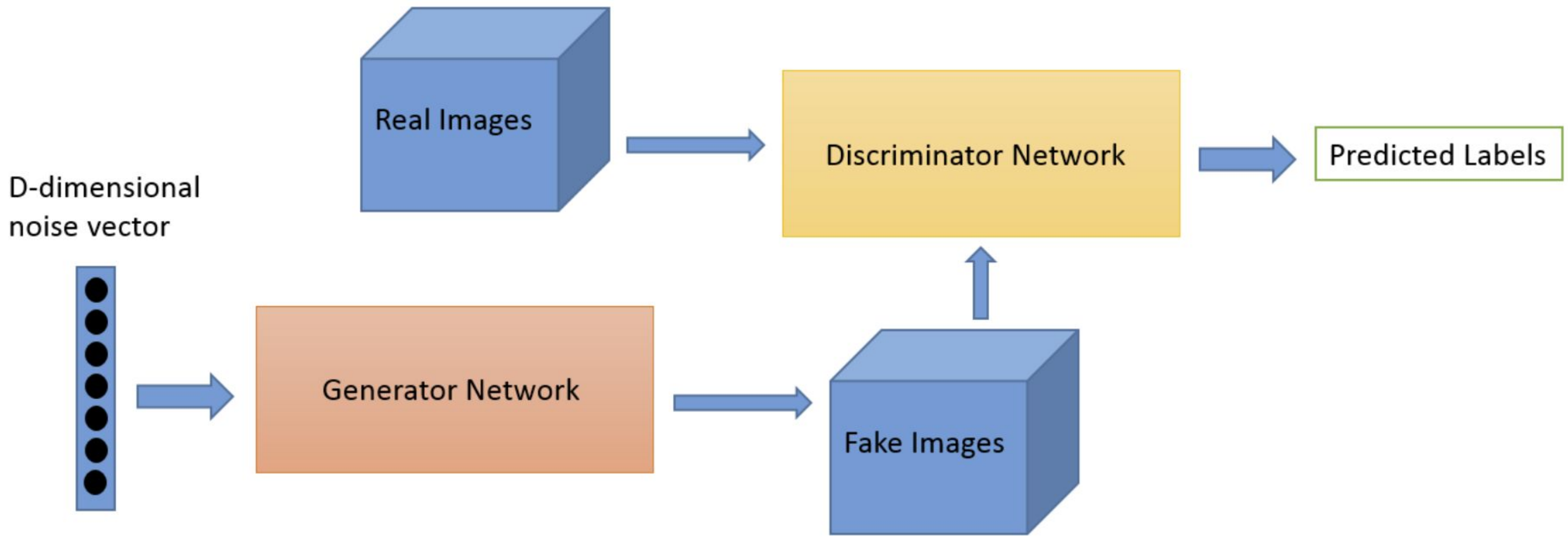
Unet (Fully convolutional neural network + Skip connection) for biomedical segmentation



03

Neural Network

Architecture – GAN



nature COMMUNICATIONS

PERSPECTIVE

https://doi.org/10.1038/s41467-019-11786-6 OPEN

A critique of pure learning and what artificial neural networks can learn from animal brains

Anthony M. Zador[1]

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**

Future

- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

(Lecun, *NIPS 2016)*

(WestWorld S1)