Deep Learning and Its Application to Signal and Image Processing and Analysis

SPRING 2022

TAMMY RIKLIN RAVIV,
ELECTRICAL AND COMPUTER ENGINEERING
BEN-GURION UNIVERSITY

1

Information Processing & Neural Networks

- The Hubel and Wiesel Experiment 1959: They inserted a *microelectrode* into the *primary visual cortex* of an anesthetized cat, and projected patterns of light and dark on a screen in front of the cat.

- They found that some *neurons* fired rapidly when presented with lines at one angle, while others responded best to another angle. Some of these neurons responded to light patterns and dark patterns differently. Hubel and Wiesel called these neurons *simple cells*.

- Still other neurons, which they termed complex cells, detected edges regardless of where they were placed in the receptive field of the neuron and could preferentially detect motion in certain directions.

- These studies showed how the visual system constructs complex representations of visual information from simple stimulus features.

T. Wiesel (left) and D. Hubel (right) co-recipients of the 1981 Nobel Prize in Physiology for their discoveries concerning information processing in the visual system

2

Information Processing & Neural Networks

- The Hubel and Wiesel Experiment 1959:

T. Wiesel (left) and D. Hubel (right) co-recipients of the 1981 Nobel Prize in Physiology for their discoveries concerning information processing in the visual system

3

Hubel and Wiesel Experiments

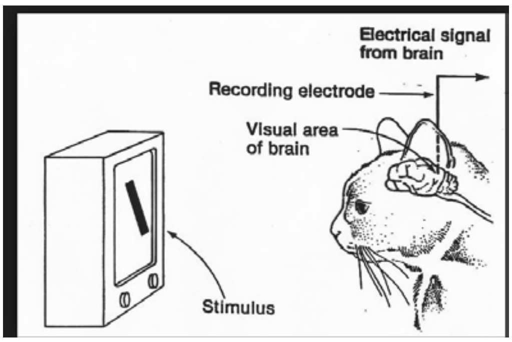
Some YouTube links:

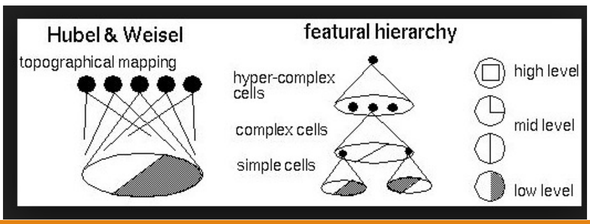
https://www.youtube.com/watch?v=IOHayh06LJ4

https://www.youtube.com/watch?v=8VdFf3egwfg

https://www.youtube.com/watch?v=y_l4kQ5wjiw

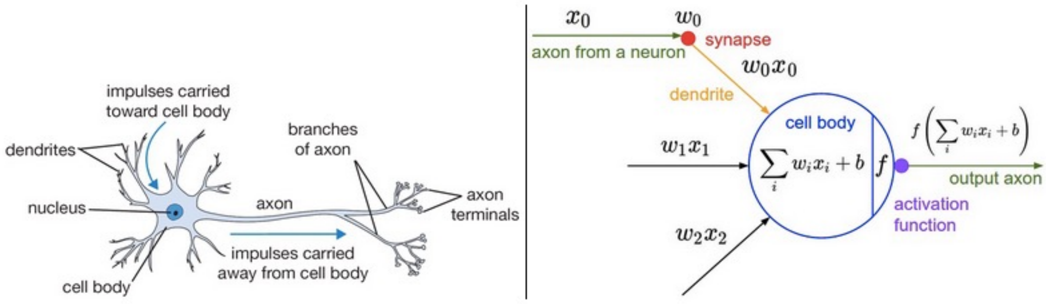
https://www.youtube.com/watch?v=UU2esxycMAw





4

Biological and Artificial Neurons



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

5

Rest of today's plan

Course overview and motivation
- The biological neuron
- The artificial neuron
- Neuron as linear classifier
- Logistic regression
- Feature representation
- Common architectures
- Some history

6

6

Deep Learning and Its Applications to Signal and Image Processing and Analysis

- Lecturer: Dr. Tammy Riklin Raviv
- No.: 361-21120
- Time: Wednesday 14:00-17:00
- Location: Building 34, room 16
- Graduate level course
- Course Web Site:

http://www.ee.bgu.ac.il/~rrtammy/DNN/DNN.html

7

7

Course Objectives

The primary objective of this course is to provide the students the necessary computational tools to:

1. Understand basic principles of artificial Neural Networks (NN) and deep learning and Machine Learning in general
2. Be familiar with a variety of NN architectures, training strategies, challenges and potential applications
3. Be familiar with up-to-date literature in ANN for signal processing/ image analysis
4. Implement, train and test DNN for particular applications

8

8

Course description

Convolutional Neural Network (CNN) – Classification, Segmentation, Object detection

Generative neural networks, GANs, Autoencoders

Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) networks, GRU, Transformers

Active learning

Reinforcement Learning

Graph neural networks

Network pruning, uncertainty, weak learning, unsupervised learning, data augmentation

Course Structure

1. Overview lectures: Basic introduction to ANN, Machine Learning, Image Processing and Analysis

2. In previous years we had a Lab. class, if needed we'll do it this year as well.

3. Guest lectures

4. Student lectures – each student will present a topic/paper to the class, followed by a discussion – a list will be distributed soon

5. Final project presentations

Course Resources

Ian Goodfellow, Yoshua Bengio, and Aaron Courville
Deep Learning. MIT Press. Online
http://www.deeplearningbook.org/

Mathematics of deep learning, Free online book

Tensor flow course: https://www.udacity.com/course/deep-learning--ud730

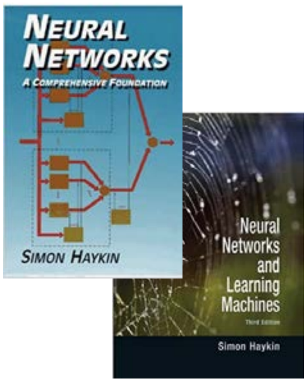
Convolutional Neural Networks for Visual Recognition – Stanford
http://cs231n.stanford.edu/ and a lecture series

Neural Networks for Machine Learning – Coursera by Jeff Hinton

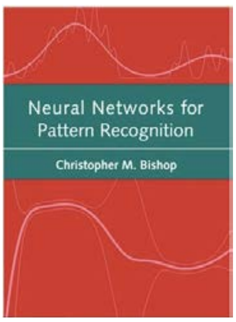
11

11

Further reading – old and new

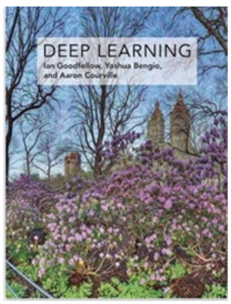


1994



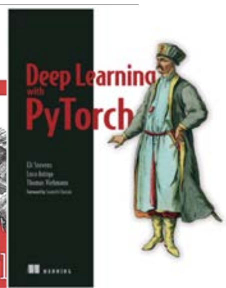
1995

..



2016

2017



2020

12

What should I do in order to succeed in the course?

- Active class participation (5 %)

Be Present in at least 10 classes out of the first 13

Last class (project presentation) is mandatory

- Homework Assignments (mandatory) 5% x 3 = 15%
- Class Presentation 10%
- Final Project 70%

13

13

The instructor

Tammy Riklin Raviv,
Research interests:
Signal processing: Biomedical Image Analysis, Computer Vision, Machine Learning
Contact info:
Telephone: 08-6428812
Fax: 08-647 2949
E-mail: rrtammy@ee.bgu.ac.il
Office: 212/33
Reception hours: please coordinate via email

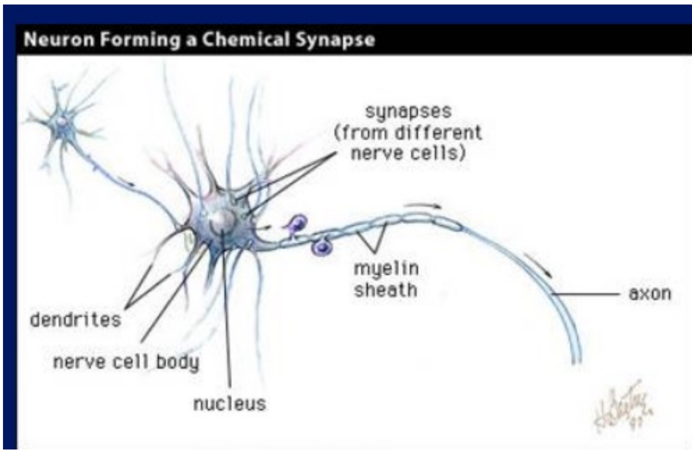
Personal web page:

http://www.ee.bgu.ac.il/~rrtammy/

14

14

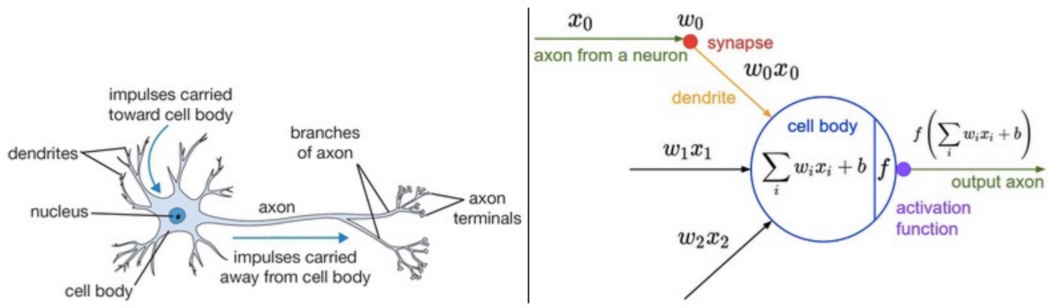
The biological neuron



https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

15

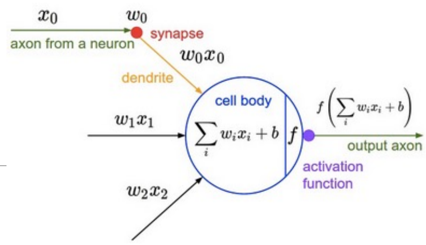
An Artificial Neuron



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

18

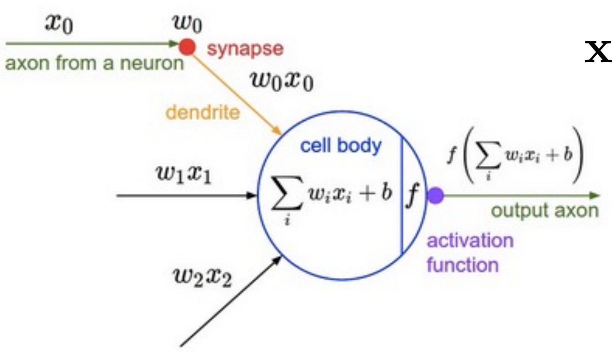
An Artificial Neuron



- Neural networks are made up of many artificial neurons.
- Artificial neurons - simplified models of biological neurons.
- Each input into the neuron is associated with weight
- A weight is simply a floating point number, which can be positive (*excitatory*) or negative (*inhibitory*) adjusted during training.
- The weighted sum of the inputs gives us the *activation.*
- The neuron's output is determined by an *activation function*.

19

An Artificial Neuron



input

$$\mathbf{x} = x_0, x_1, \ldots x_{D-1}$$

neuron parameters

$$w_0, w_1, \ldots w_{D-1}, b$$

output

$$f\left(\sum_i w_i x_i + b\right)$$

20

Learning to Classify

What can a single neuron do?

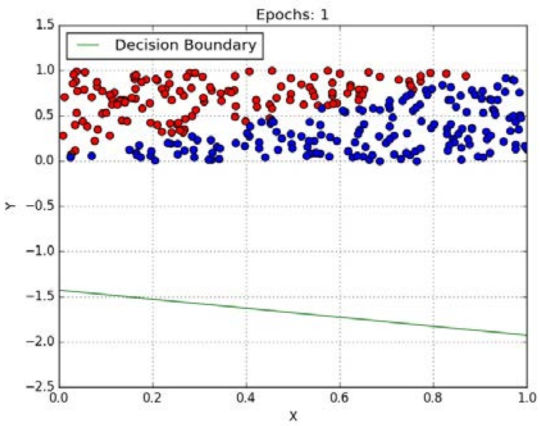
Supervised Learning problem:

input $\mathbf{x}^\mathbf{j} = \{x_0, x_1, \ldots x_{D-1}\}$

$$\{\mathbf{x}^0, L(\mathbf{x}^0)\}, \{\mathbf{x}^1, L(\mathbf{x}^1)\} \ldots \{\mathbf{x}^N, L(\mathbf{x}^N)\}$$

$$L(\mathbf{x}^j) \in \{ \bullet , \bullet \}$$

Learned parameters $\{w_i\}$, b



21

Learning to Classify

$$y = mx + k$$

$$0 = w_0x_0 + w_1x_1 + \ldots + w_{D-1}x_{D-1} + b$$

What can a single neuron do?

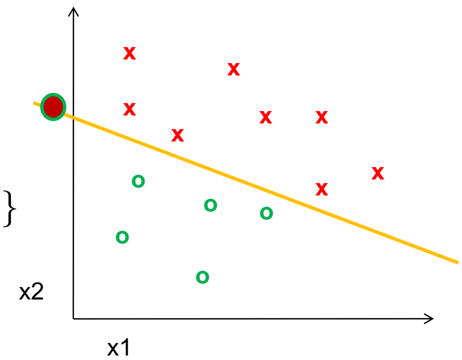
Supervised Learning problem:

input

$$\{\mathbf{x}^0, L(\mathbf{x}^0)\}, \{\mathbf{x}^1, L(\mathbf{x}^1)\} \ldots \{\mathbf{x}^N, L(\mathbf{x}^N)\}$$

$$L(\mathbf{x}^j) \in \{ \text{x} , \circ \}$$

Learned parameters $\{w_i\}$, b



22

Learning to Classify

$$y = mx + k$$

$$0 = \mathbf{wx} + b$$

What can a single neuron do?

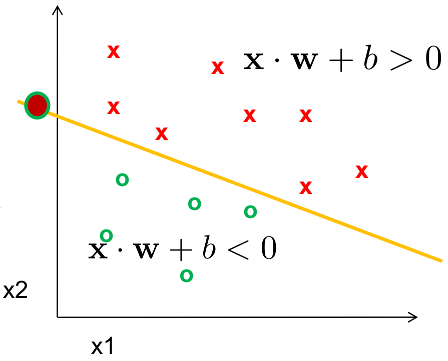
Supervised Learning problem:

input

$$\{\mathbf{x}^0, L(\mathbf{x}^0)\}, \{\mathbf{x}^1, L(\mathbf{x}^1)\} \ldots \{\mathbf{x}^N, L(\mathbf{x}^N)\}$$

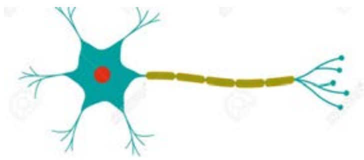
$$L(\mathbf{x}^j) \in \{\, \text{x} \,, \text{o} \,\}$$

Learned parameters $\{w_i\}$, b



23

Learning to Classify



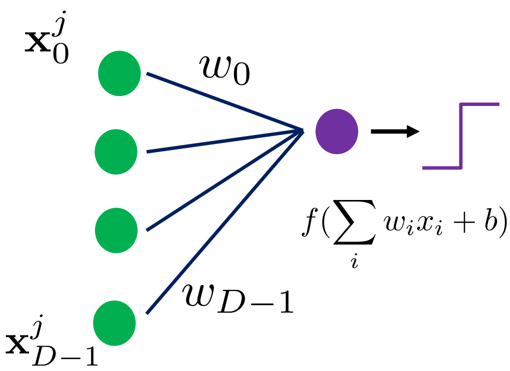
What can a single neuron do?

Supervised Learning problem:

input $\mathbf{x^j} = \{x_0, x_1, \ldots x_{D-1}\}$

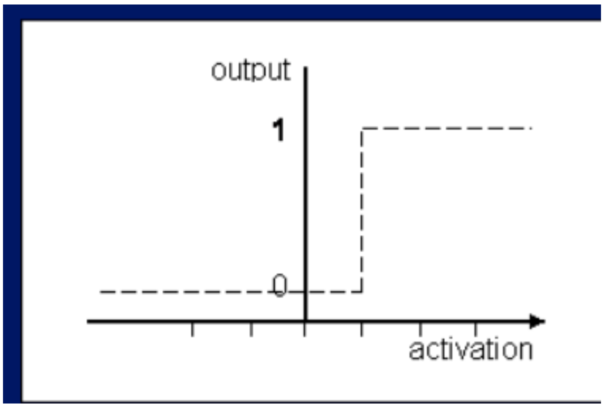
$$\{\mathbf{x}^0, L(\mathbf{x}^0)\}, \{\mathbf{x}^1, L(\mathbf{x}^1)\} \ldots \{\mathbf{x}^N, L(\mathbf{x}^N)\}$$

Learned parameters $\{w_i\}$, b



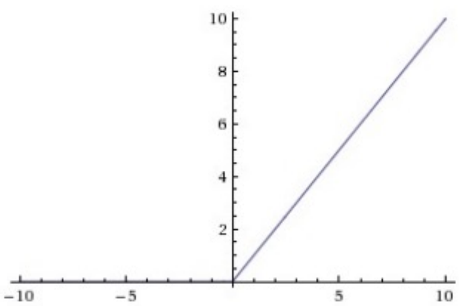
24

Simple activation function - threshold



25

Rectified linear unit (ReLU) activation function

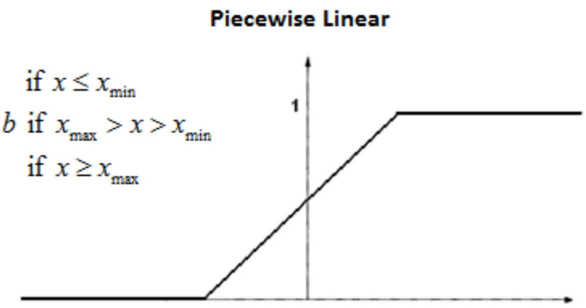


$$f(x) = max(0, x)$$

26

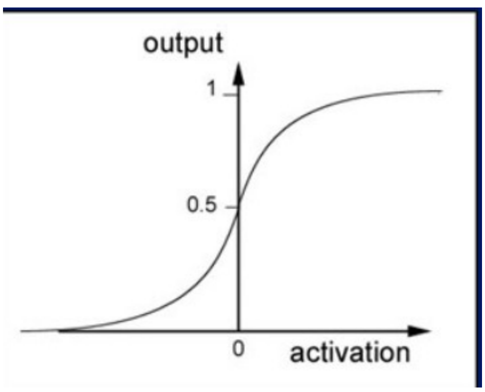
Piecewise linear activation function

$$f(x)=\begin{cases}0 & \text{if } x \le x_{\min} \\ mx+b & \text{if } x_{\max} > x > x_{\min} \\ 1 & \text{if } x \ge x_{\max}\end{cases}$$



27

Sigmoid activation function

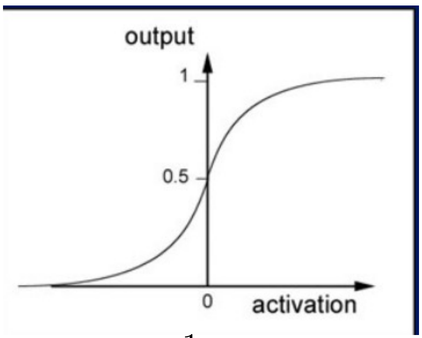


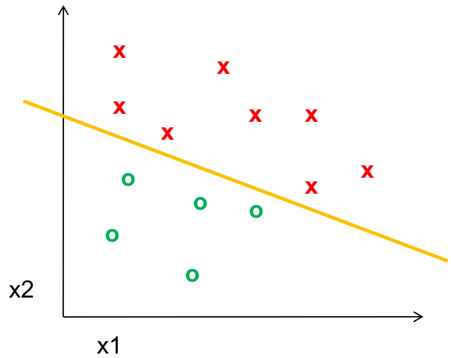
$$f = \frac{1}{1+\exp(-(\mathbf{w}\cdot\mathbf{x}+b))}$$

bias

28

Sigmoid activation function





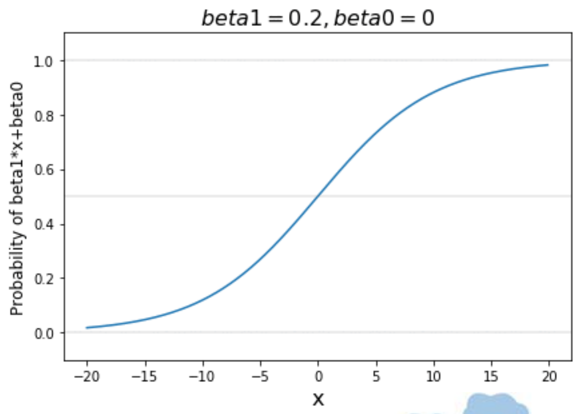
$$f = \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))}$$

29

Logistic regression function

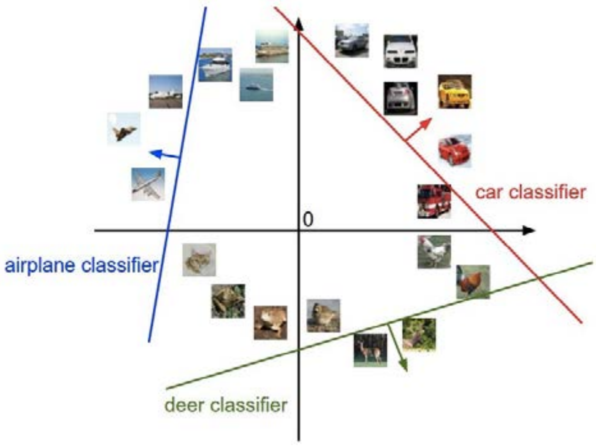
$$f = \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))}$$

$$= \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$



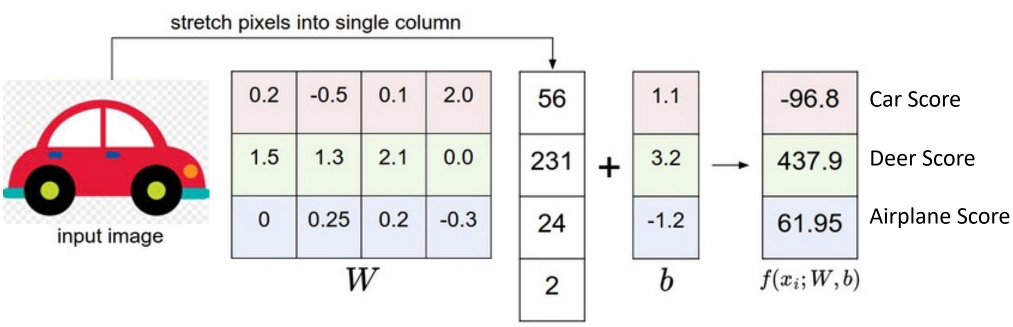
30

Categorical (multi-class) Classification



31

Categorical (multi-class) Classification

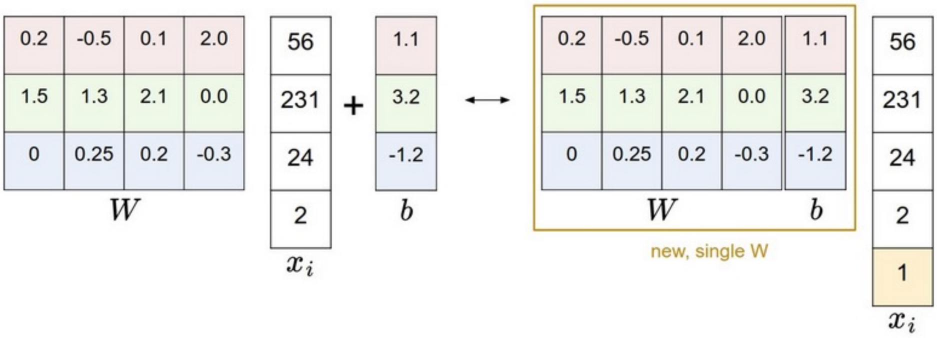


$$\mathbf{x^j} = \{x_0, x_1, \ldots x_{D-1}\}$$

32

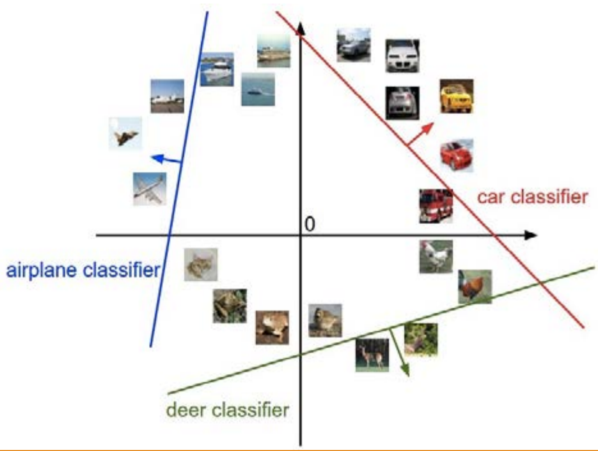
Bias trick

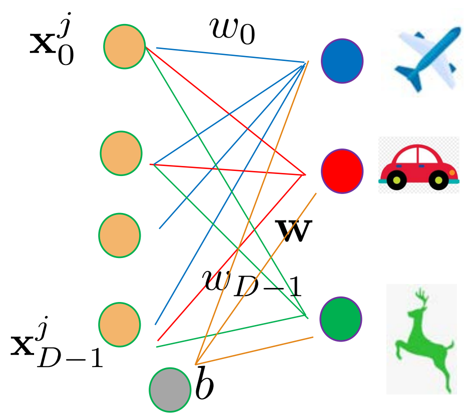
$f(x_i, W, b) = Wx_i + b \longrightarrow f(x_i, W) = Wx_i$



33

Categorical (multi-class) Classification





34

Scores and Loss

(1, 0, 0)

(0, 1, 0)

(0, 0, 1)

One-hot encoding

-96.8	Car Score
437.9	Deer Score
61.95	Airplane Score

$f(x_i; W, b)$

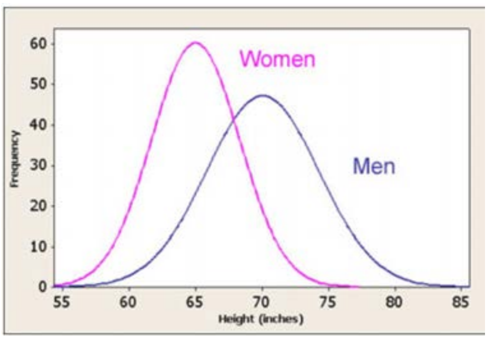
$$(\alpha, \beta, \gamma)$$

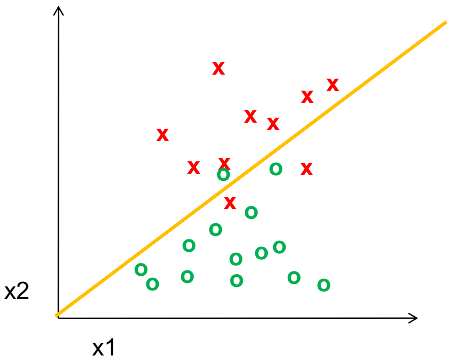
multinomial
logistic regression

35

Representation matters

How to select features?

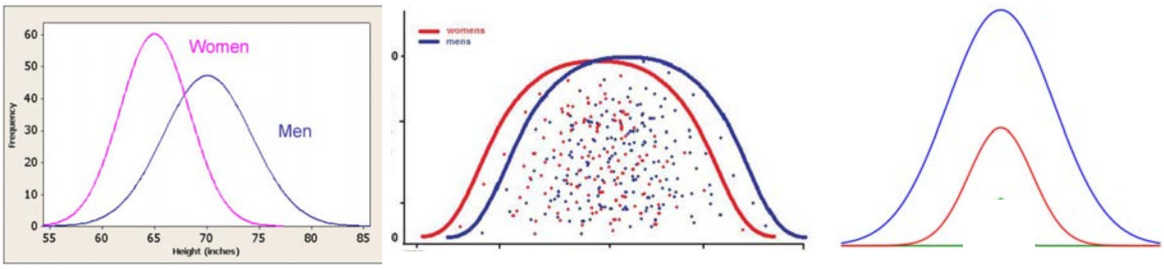




36

Representation matters

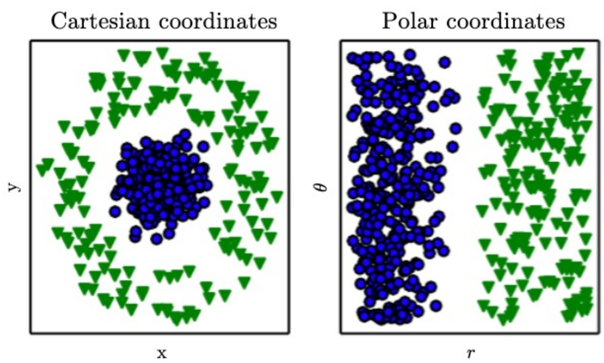
How to select features that discriminate between classes?



37

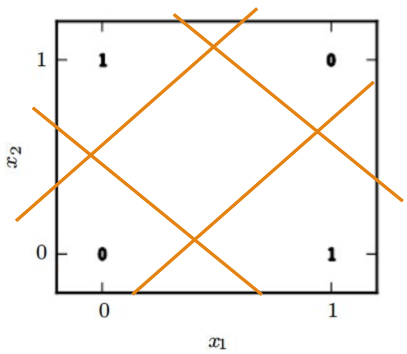
Representation matters

How to select linearly separable features?



38

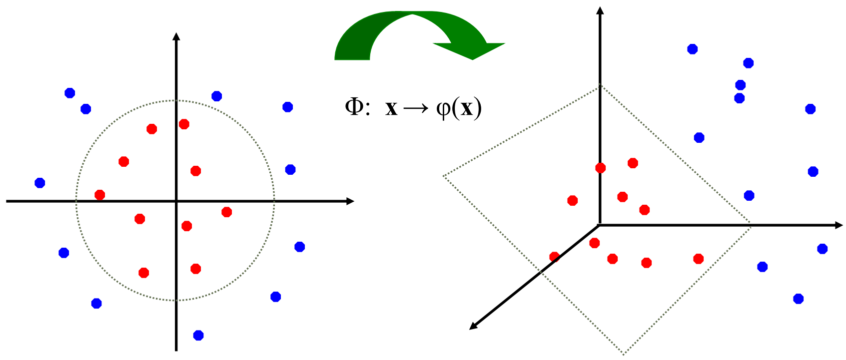
XOR Function



39

Classical Machine Learning

Mapping into higher dimension

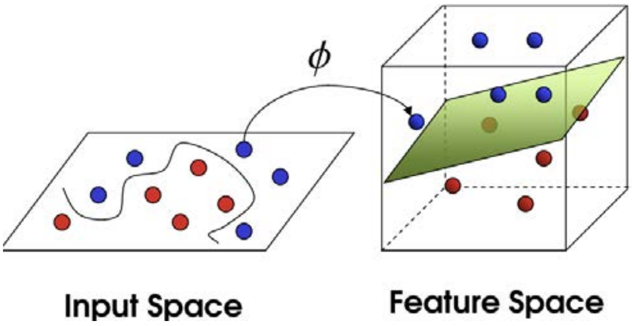


Slide credit: Andrew Moore

40

Classical Machine Learning

Mapping into higher dimension

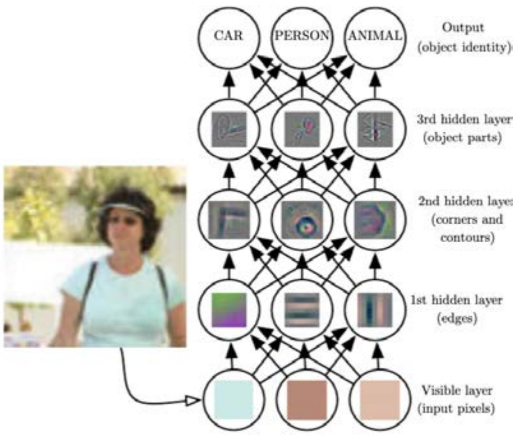


Slide credit: Andrew Moore

41

Deep learning model

From pixels to features
as we get deeper

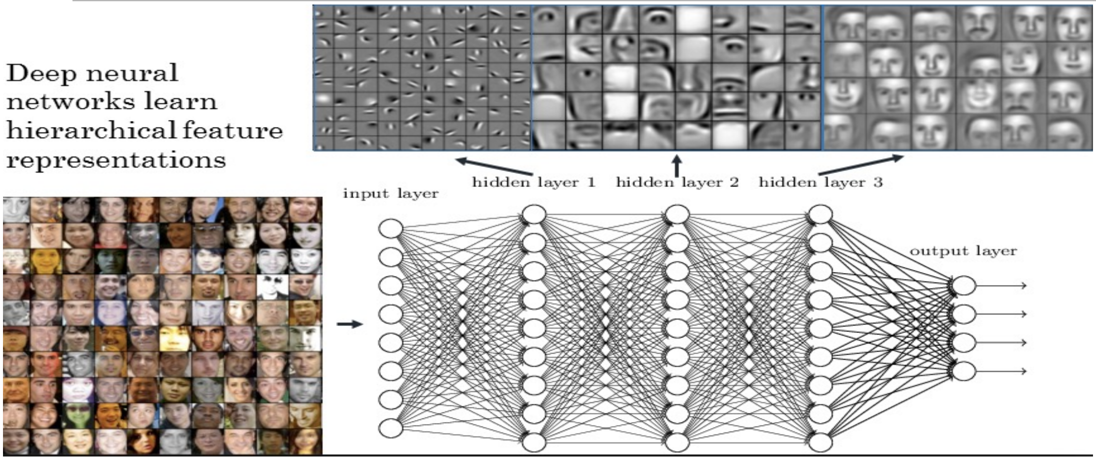


42

Hidden layers

The hidden layers can be seen as distorting the input in a non-linear way so that categories become linearly separable by the last layer.

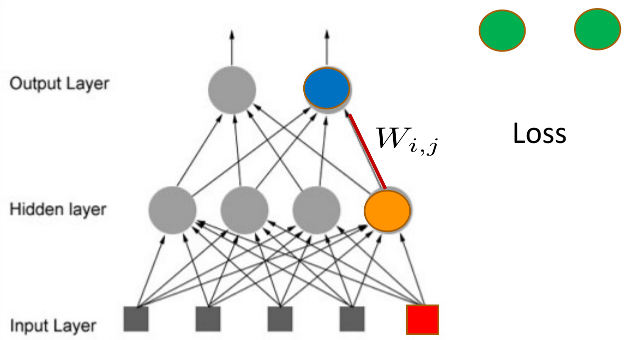
Deep neural networks learn hierarchical feature representations



43

Deep Neural Network

Feed forward neural network:



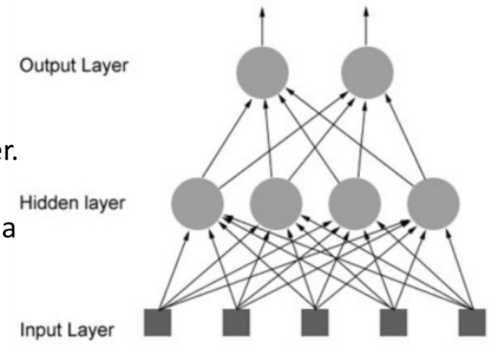
44

Artificial neural network architecture

Feed forward neural network:

- Each input is sent to every neuron in the *hidden layer* and then each hidden layer's neuron's output is connected to every neuron in the next layer.

- There can be any number of hidden layers within a feedforward network and any number of neurons.



45

Artificial neural network architecture

Feed forward neural network:

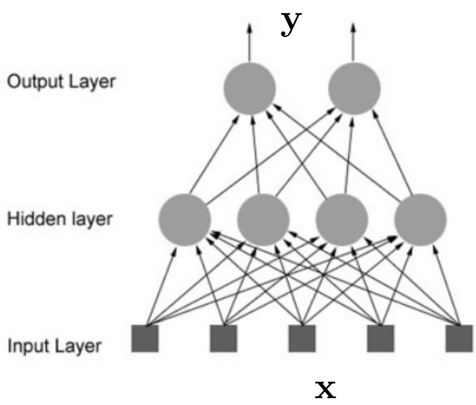
Goal: approximate some function f^*

Classifiers: $y = f^*(\mathbf{x})$

maps an input $\mathbf{x}$ to a category y

Enhancement/denoising/ semantic segmentation/transformation:

$$\mathbf{y} = f^*(\mathbf{x})$$

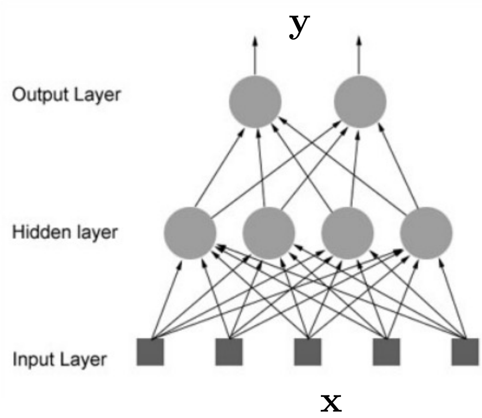


46

Artificial neural network architecture

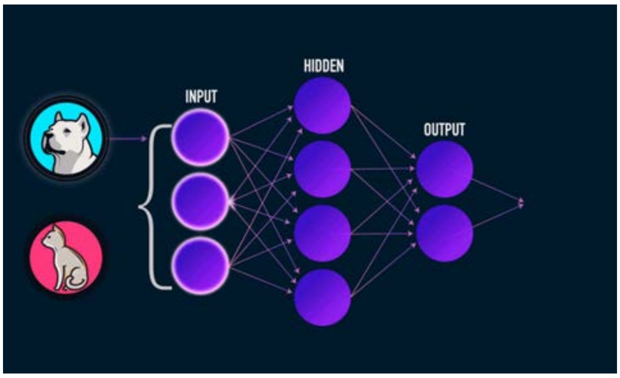
Feed forward neural network:

A feedforward network defines a mapping $\mathbf{y} = f^*(\mathbf{x}, \theta)$ and learns the value of the parameters θ that result in the best function approximation.



47

Feed Forward

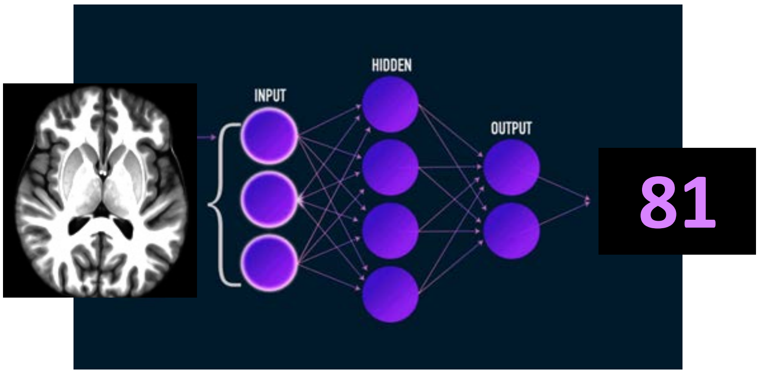


https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

48

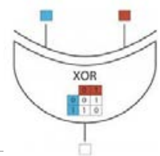
Prediction by Artificial Neural Network

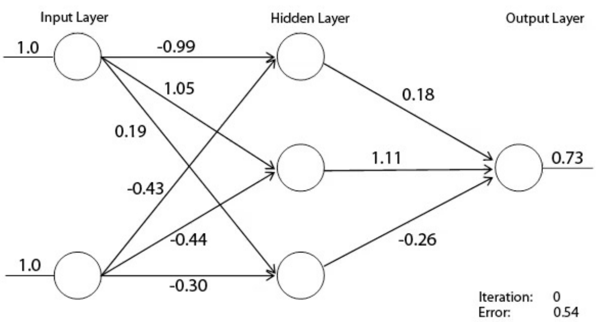
Regression



49

Training NN as a XOR function





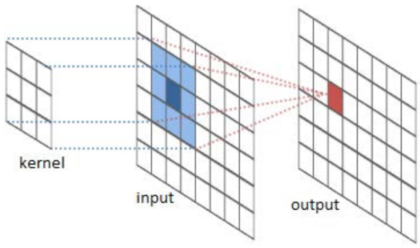
Truth table:

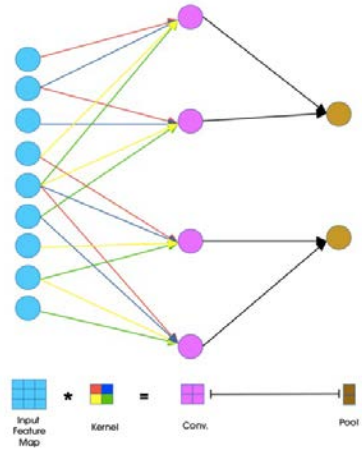
X_1	X_2	Result
0	0	0
0	1	1
1	0	1
1	1	0

50

Artificial neural network architecture

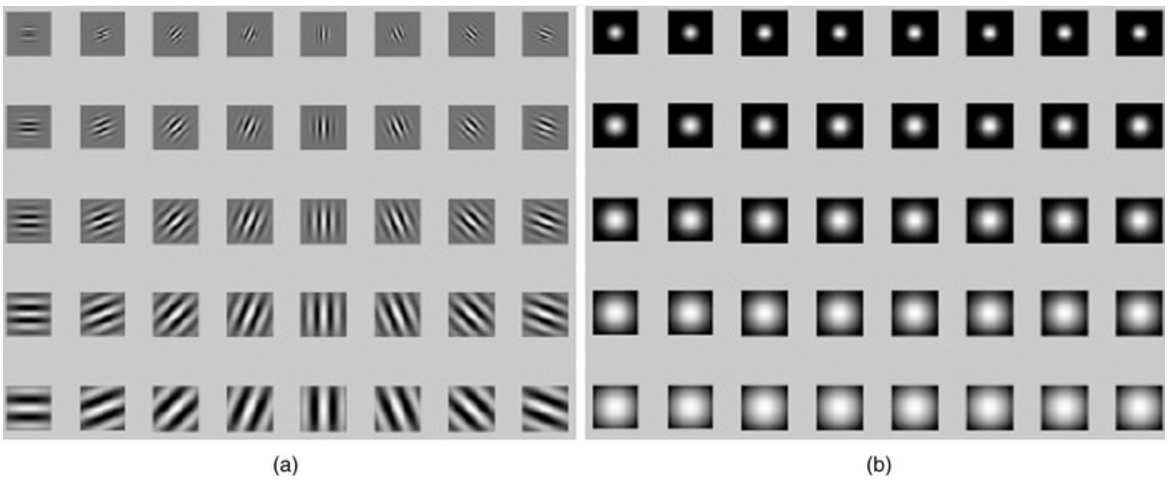
Convolutional neural network (CNN):





52

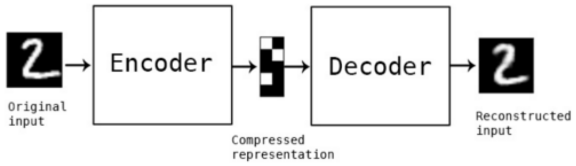
Feature extraction

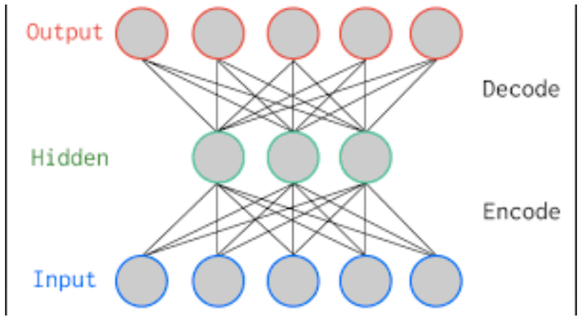


53

Artificial neural network architecture

Autoencoder:

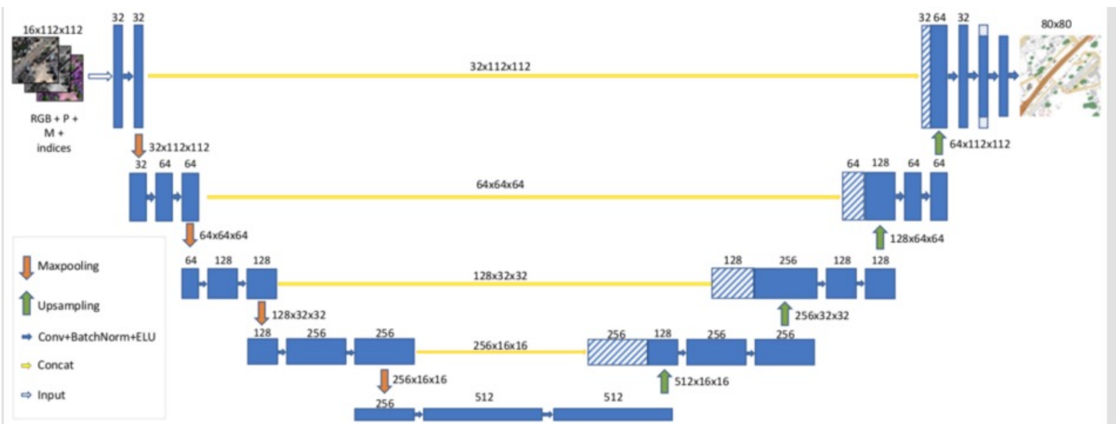




54

Artificial neural network architecture

U-Net:



55

Artificial neural network architecture

Residual neural network (Resnet):

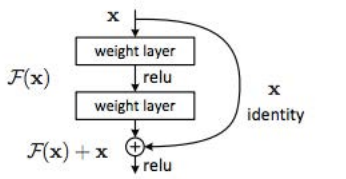
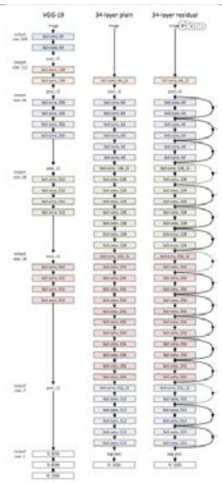


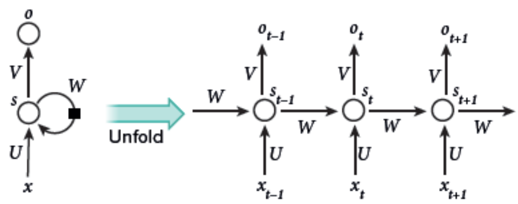
Figure 2. Residual learning: a building block.

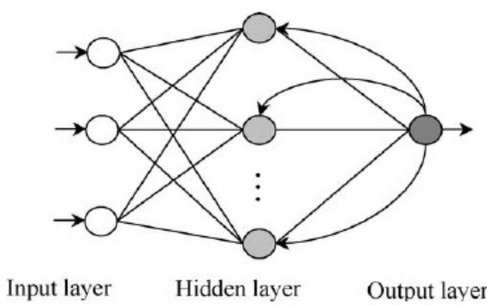


56

Artificial neural network architecture

Recurrent neural network (RNN):





57

Artificial neural network architecture

Long Short-term Memory (LSTM):

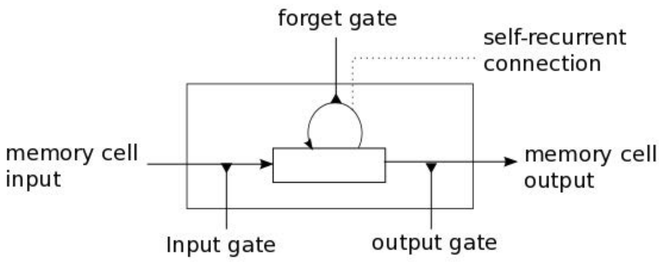
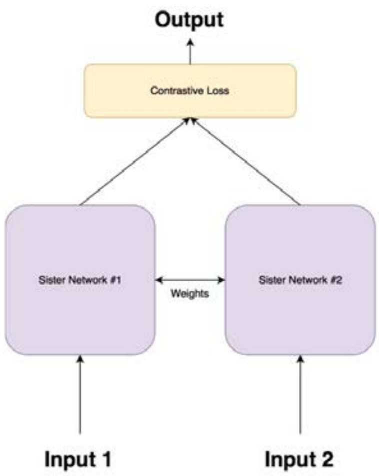


Figure 1 : Illustration of an LSTM memory cell.

58

Artificial neural networks architecture

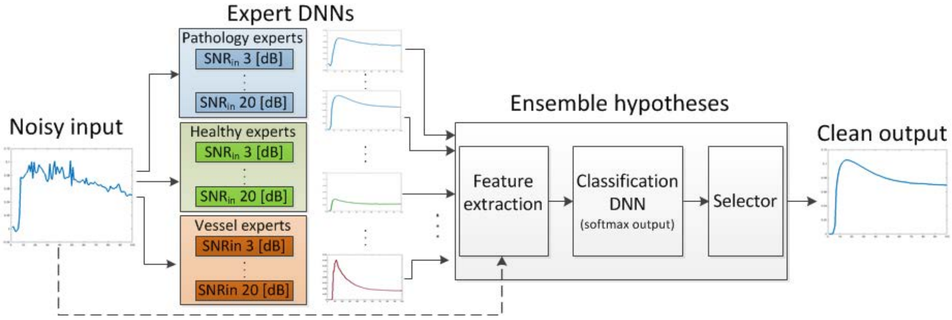
Siamese Neural Networks:



59

Artificial neural networks architecture

Expert DNN ensemble:

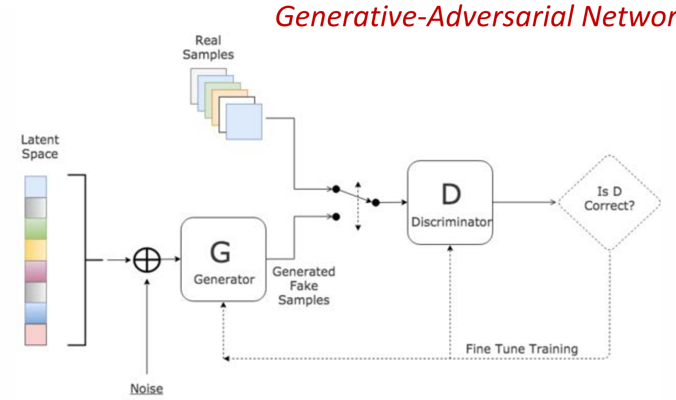


Benou et al.

60

Artificial neural networks architecture

Generative-Adversarial Networks (GANs):



https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

61

Artificial neural network

Simple (and classical) example
Character recognition:

Input: binary vector of length N X N
Requires N X N input neurons.

Output: a binary vector e.g.
(0,0,0,1, …0)

and the answer is '4'

62

Training and Test

Supervised neural networks:

The weights (and the biases) are adjusted by training using a *training set*.

A common way to train is by *back-propagation*.

When weights are adjusted – run *test examples.*

We will talk on both *supervised* and *unsupervised* networks.

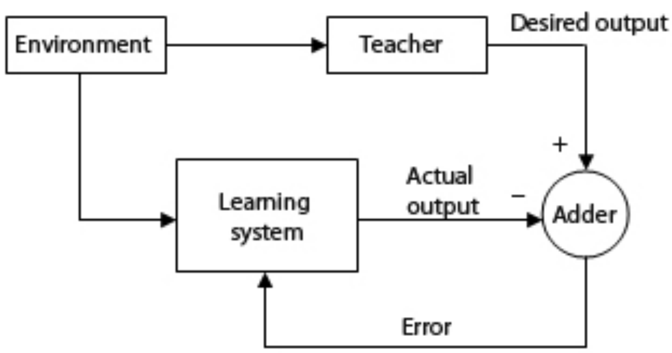
63

Back-propagation

Supervised

?

Most common



64

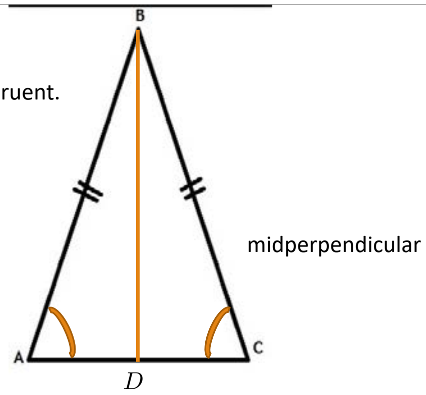
“Easy” and “difficult” tasks

68

Some high-school geometry

Theorem

If two sides of a triangle are congruent,
then the angles opposite those sides are congruent.



An isosceles triangle

69

"Difficult/Easy" task
Image classification

Assigning an input image one label from a fixed set of categories

Motivation: An important computer vision problem.

Has a large variety of practical applications.

Many other seemingly distinct Computer Vision tasks (such as object detection, segmentation) can be reduced to image classification.

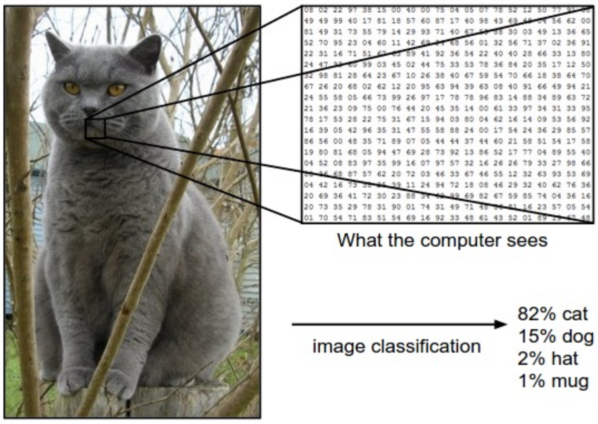
Example 1: Digit classification {0,1,2,3…8,9}

Example 2: Take a single image and assign probabilities to 4 labels, *{cat, dog, hat, mug}*.

70

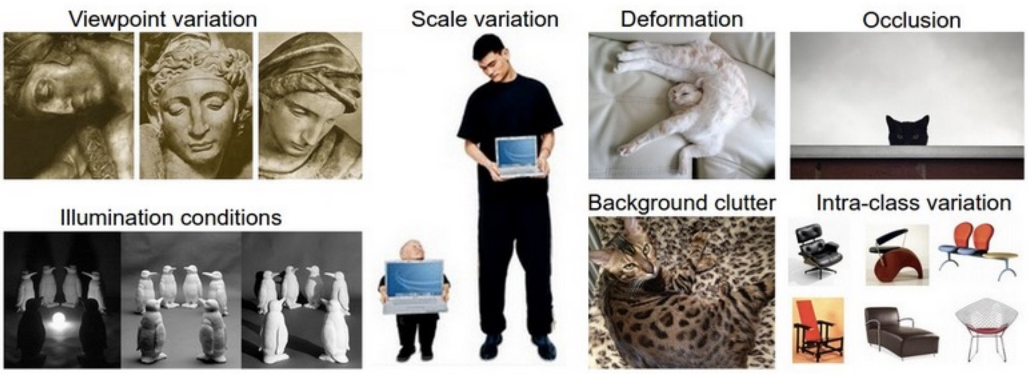
Image classification – an example

The task in Image Classification is to predict a single label (or a distribution over labels as shown here to indicate our confidence) for a given image. Images are 3-dimensional arrays of integers from 0 to 255, of size Width x Height x 3.
The 3 represents the three color channels Red, Green, Blue.



71

Image classification - challenge



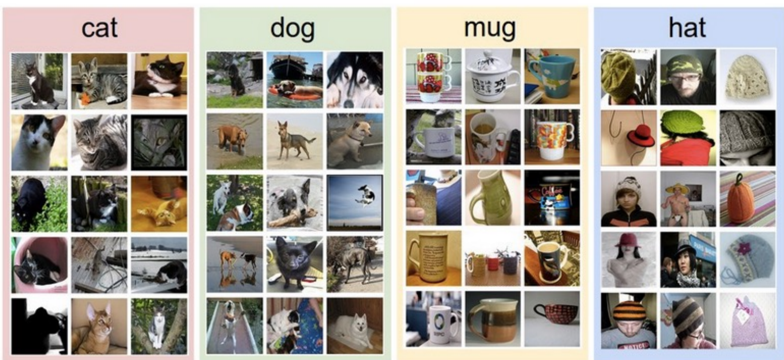
72

Data driven approach

Instead of trying to specify what every one of the categories of interest look like directly in code, provide the computer with many examples of each class and then develop learning algorithms that look at these examples and learn about the visual appearance of each class.

73

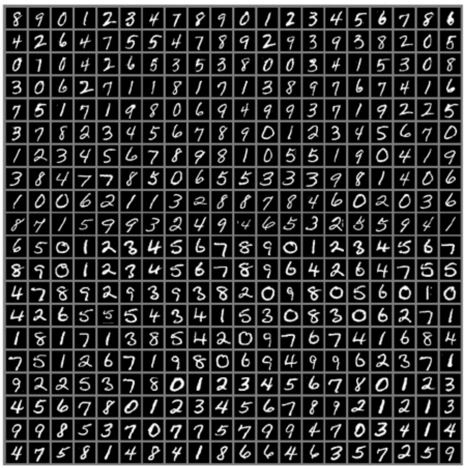
Labeled database



An example training set for four visual categories. In practice we may have thousands of categories and hundreds of thousands of images for each category.

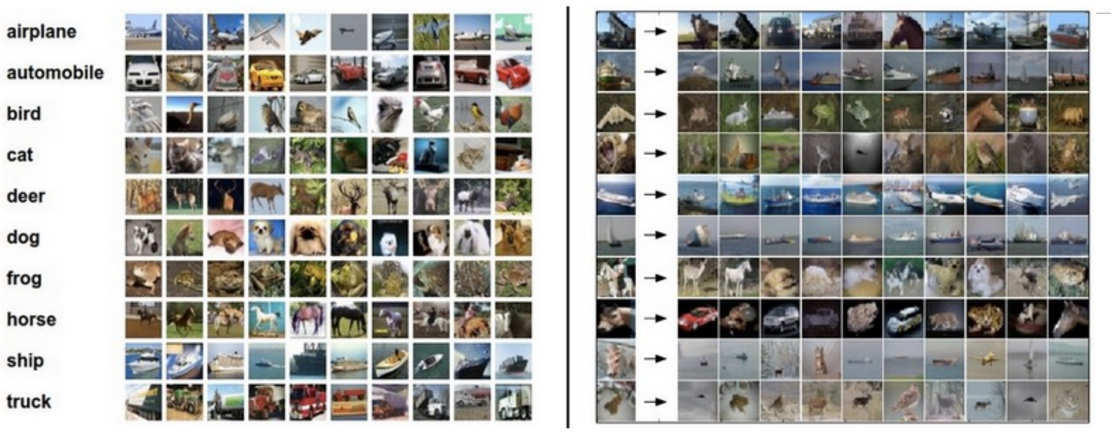
74

MNIST database



75

image classification dataset: CIFAR-10.

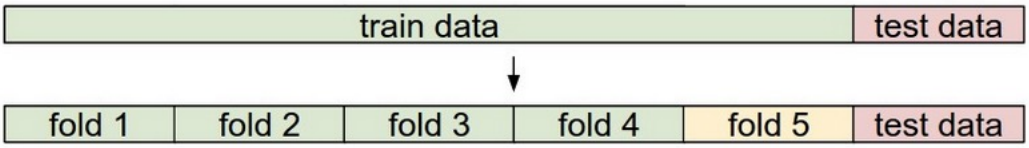


Left: Example images from the CIFAR-10 dataset. Right: first column shows a few test images and next to each we show the top 10 nearest neighbors in the training set according to pixel-wise difference.

76

Cross validation

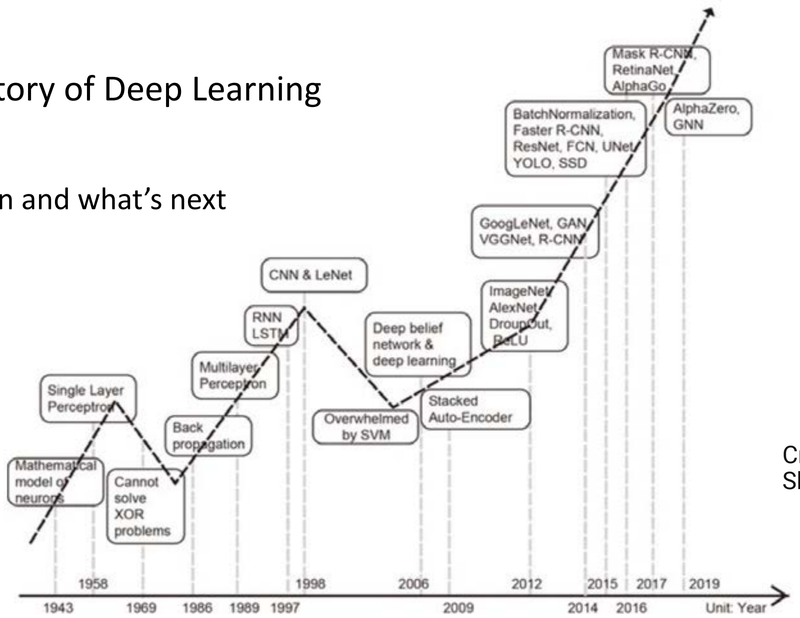
Insufficient data ?
Split your training set into training set and a validation set.
Use validation set to tune all hyperparameters.
At the end run a single time on the test set and report performance.



77

The History of Deep Learning

Conclusion and what's next



Credit
Shichao Jin

78

Next class

More on neural networks

Loss functions

Stochastic gradient descent

Back Propagation

Regularization

Optimization

Capacity

Receptive field

Overfitting

Underfitting

79