Deep Learning

Vazgen Mikayelyan

YSU, Krisp

September 11, 2019

Outline

1. Course Schedule
2. References
3. What is Supervised Learning?
4. What is a neural network?
5. Gradient Descent

Course Schedule

1. Intro to supervised learning.
2. Overfitting, underfitting
3. What is a neural network?
4. Gradient Descent
5. Linear Regression, Logistic Regression
6. Activation functions
7. Softmax classifier
8. Data normalization
9. Back propagation
10. Random Initialization
11. Regularization
12. Dropout
13. Batch normalization
14. Data augmentation
15. Vanishing / Exploding gradients
16. Mini-batch gradient descent

Course Schedule

17. Gradient descent with momentum
18. RMSprop
19. Adam optimization algorithm
20. Learning rate tricks
21. Batch normalization
22. Introduction to TensorFlow
23. Convolutional neural networks
24. ResNets
25. Inceptions
26. VGG
27. Transfer learning
28. Autoencoders
29. GANs
30. Multitask learning
31. Basic Recurrent Neural Networks
32. GRU

Course Schedule

33. LSTM
34. Bidirectional RNN
35. Multicell RNNs
36. Attention models
37. Bayesian neural networks
38. Word2Vec
39. Sequence to sequence models
40. GPU optimisations for Neural Networks

Outline

1. Course Schedule
2. References
3. What is Supervised Learning?
4. What is a neural network?
5. Gradient Descent

References

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville - Deep Learning
2. Original Papers
3. Blog Posts

Outline

1. Course Schedule
2. References
3. What is Supervised Learning?
4. What is a neural network?
5. Gradient Descent

What is Supervised Learning?

Definition 1

In machine learning, ***supervised learning*** *refers to a class of systems and algorithms that determine a predictive model using data points with known outcome.*

What is Supervised Learning?

Definition 1

*In machine learning, **supervised learning** refers to a class of systems and algorithms that determine a predictive model using data points with known outcome.*

Supervised learning is often used to create machine learning models for two types of problems:

What is Supervised Learning?

Definition 1

*In machine learning, **supervised learning** refers to a class of systems and algorithms that determine a predictive model using data points with known outcome.*

Supervised learning is often used to create machine learning models for two types of problems:

1. Regression

What is Supervised Learning?

Definition 1

*In machine learning, **supervised learning** refers to a class of systems and algorithms that determine a predictive model using data points with known outcome.*

Supervised learning is often used to create machine learning models for two types of problems:

1. Regression
2. Classification

Example of regression problem

Example of regression problem

Size of House	Price of House
950	$123,325
1,535	$156,570
1,605	$158,895
1,905	$200,025
2,057	$230,384
2,227	$233,835
3,150	$261,420
3,620	$433,500

Example of classification problem

Example of classification problem

General case

Our data consist of pairs

$$(x_i, y_i)_{i=1}^{n}\,,\ \text{where } x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^m, i = 0, 1, \ldots, n.$$

General case

Our data consist of pairs

$$(x_i, y_i)_{i=1}^{n}, \text{ where } x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^m, i = 0, 1, \ldots, n.$$

We have to find a function f for which

$$f(x_i) \approx y_i$$

General case

Our data consist of pairs

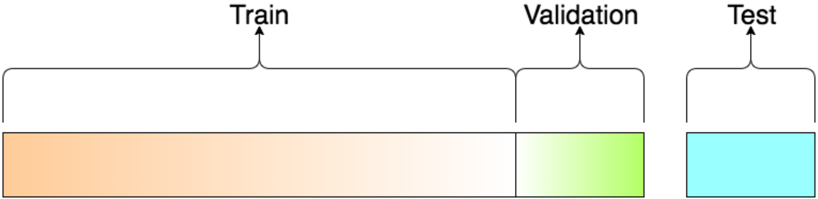
$$(x_i, y_i)_{i=1}^{n}, \text{ where } x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^m, i = 0, 1, \ldots, n.$$

We have to find a function f for which

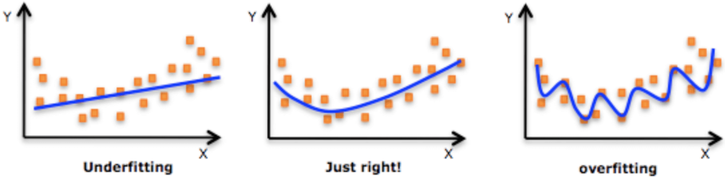
$$f(x_i) \approx y_i$$

and not only for pairs in our data.

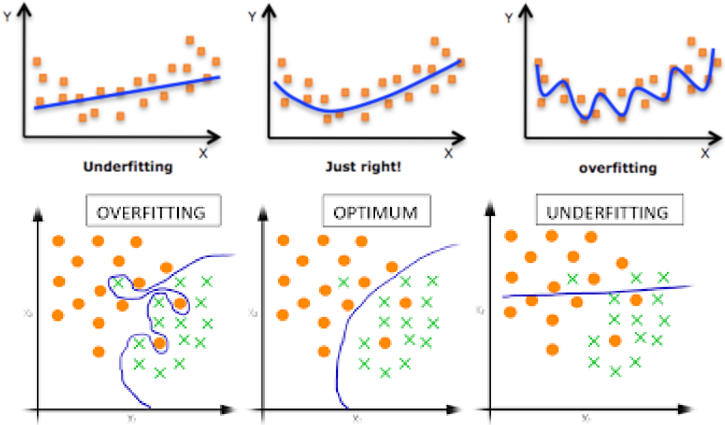
Train, Validation, Test



Overfitting, Underfitting



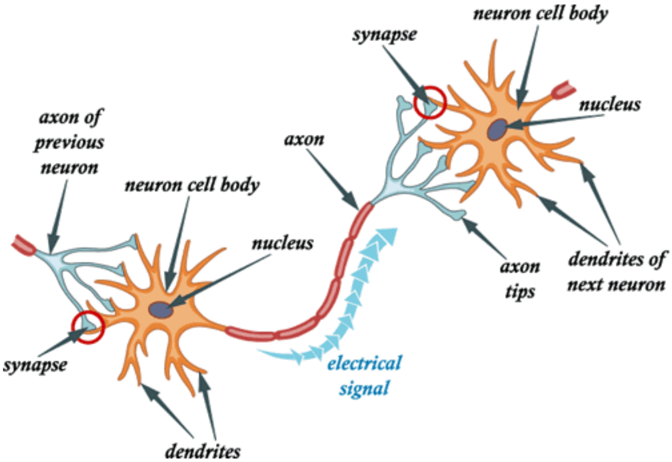
Overfitting, Underfitting



Outline

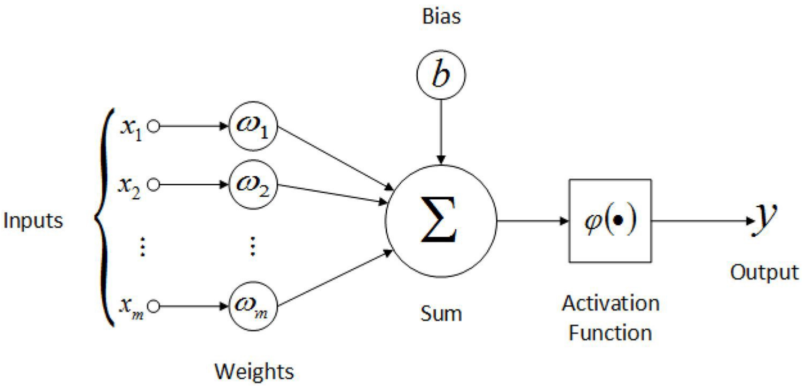
1. Course Schedule
2. References
3. What is Supervised Learning?
4. What is a neural network?
5. Gradient Descent

Human Brain

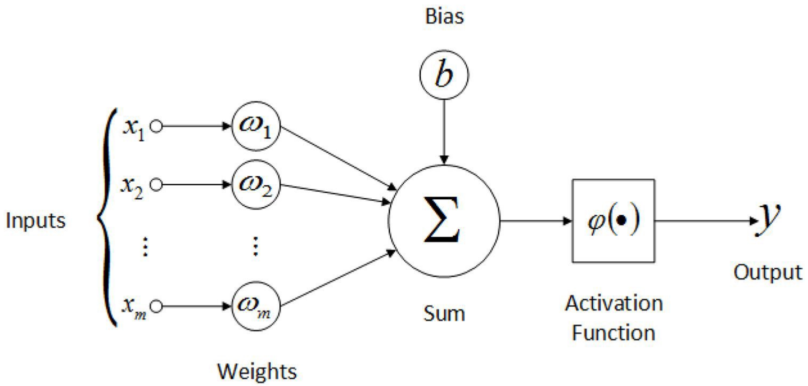


neuron cell body
synapse
nucleus
axon of
previous
neuron
axon
neuron cell body
nucleus
axon
tips
dendrites of
next neuron
synapse
electrical
signal
dendrites

Artificial neuron



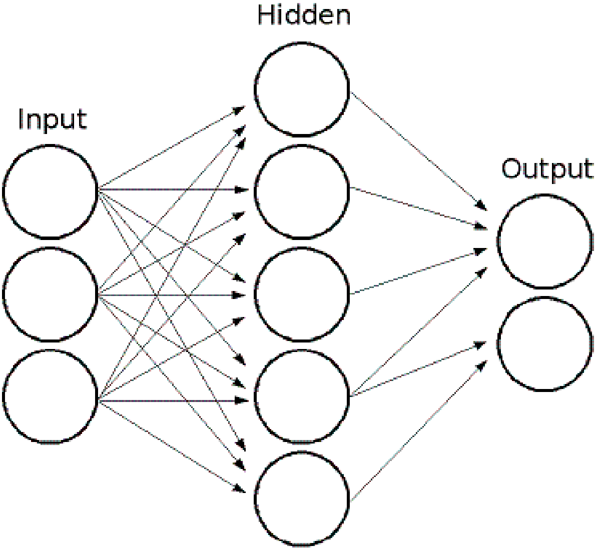
Artificial neuron



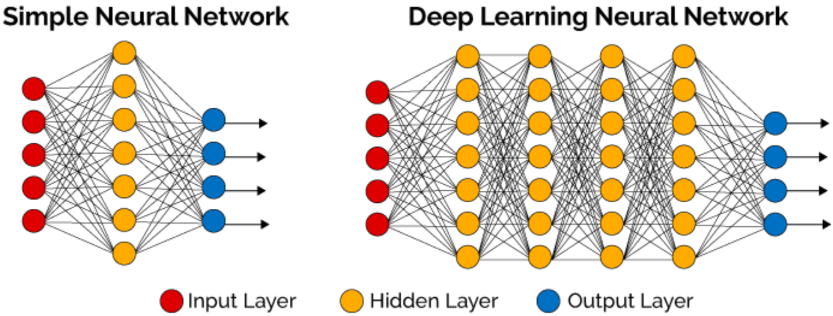
So artificial neuron is a function from $\mathbb{R}^m$ to $\mathbb{R}$, where m is the dimension of input:

$$f_{w,b}(x_1, x_2, \ldots, x_m) = \varphi(w_1 x_1 + w_2 x_2 + \ldots + w_m x_m + b) = \varphi\left(w^T x + b\right)$$

Hidden Layer



Deep neural network

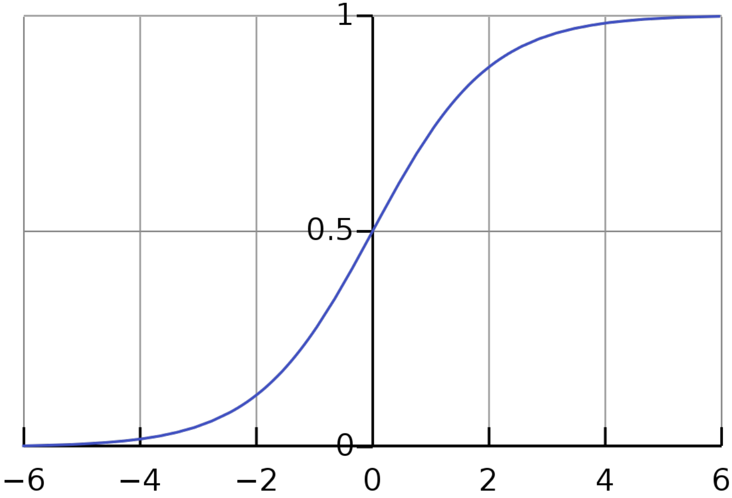


Activation functions

1. Sigmoid: $\sigma\left(x\right) = \dfrac{1}{1 + e^{-x}}$

Activation functions

1. Sigmoid: $\sigma(x) = \dfrac{1}{1+e^{-x}}$

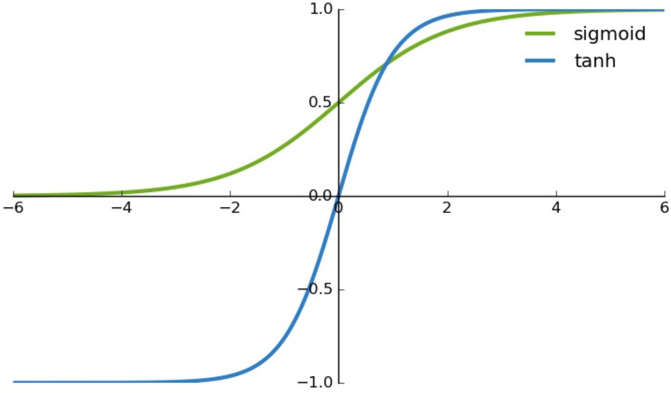


Activation functions

2. Tanh: $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

Activation functions

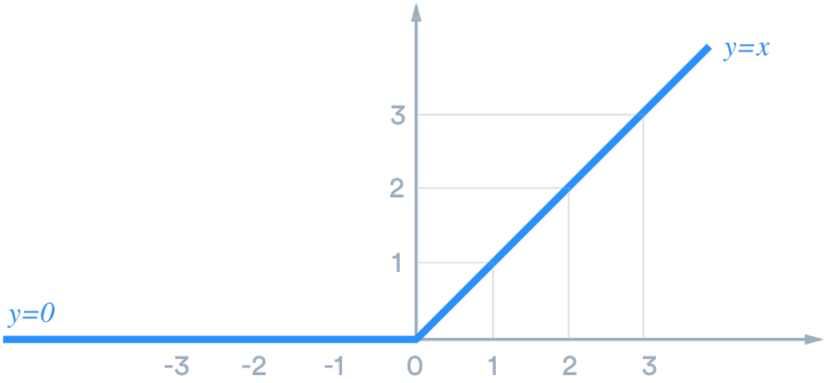
2. Tanh: $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$



Activation functions

3. Rectified linear unit: $ReLU(x) = \max(0, x)$

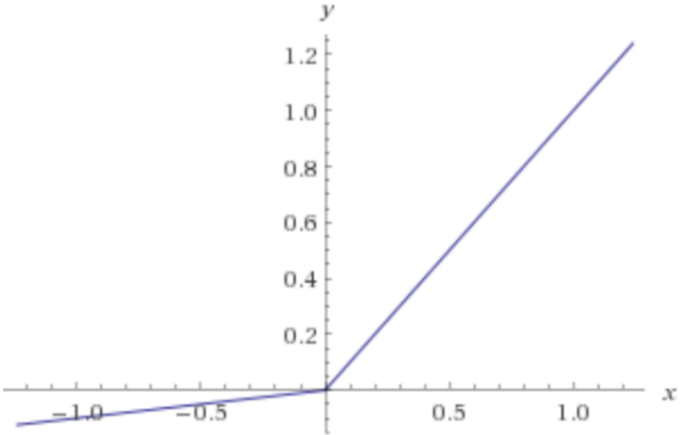
3. Rectified linear unit: $ReLU(x) = \max(0, x)$



Activation functions

4. Leaky ReLU: $LR(x) = \begin{cases} 0.01x, \text{ for } x < 0 \\ x, \text{ for } x \geq 0 \end{cases}$

4. Leaky ReLU: $LR(x) = \begin{cases} 0.01x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$

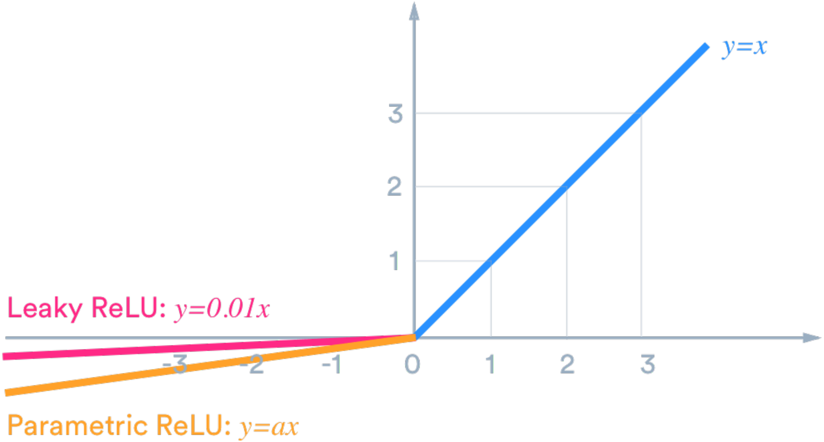


Activation functions

5. Parametric ReLU: $PR\left(x\right) = \begin{cases} ax, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$

Activation functions

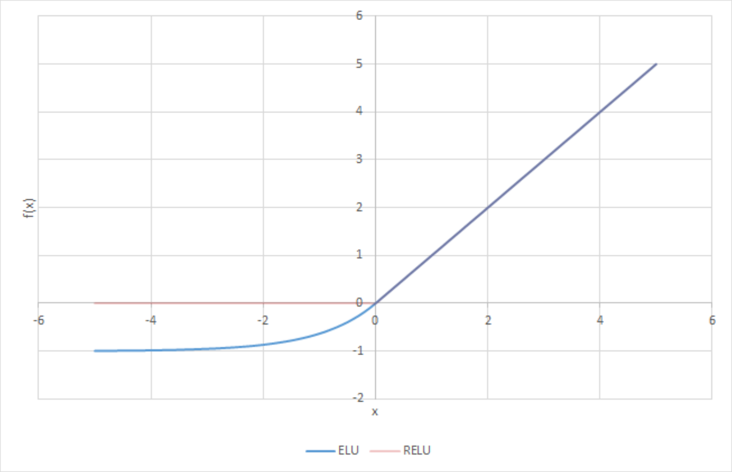
5. Parametric ReLU: $PR(x) = \begin{cases} ax, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$



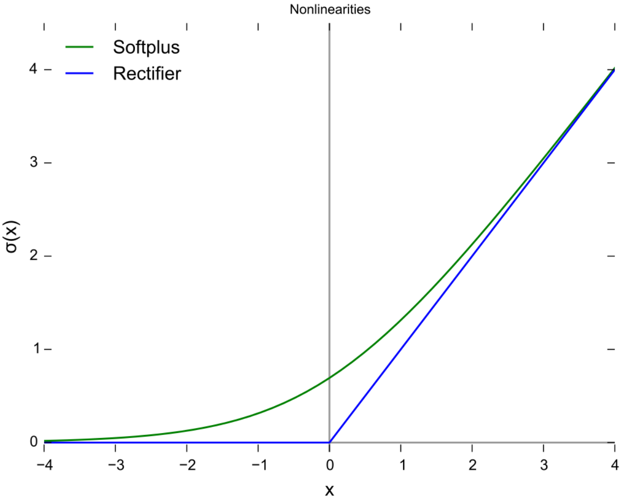
6. Exponential linear unit: $ELU(x) = \begin{cases} a\left(e^{x} - 1\right), & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$

Activation functions

6. Exponential linear unit: $ELU(x) = \begin{cases} a\left(e^x - 1\right), & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$



7. SoftPlus: $SP(x) = \log\left(1 + e^{x}\right)$



8. Softmax: $S\left(x_1, x_2, \ldots, x_n\right) = \left(\dfrac{e^{x_1}}{\sum\limits_{i=1}^{n} e^{x_i}}, \dfrac{e^{x_2}}{\sum\limits_{i=1}^{n} e^{x_i}}, \ldots, \dfrac{e^{x_n}}{\sum\limits_{i=1}^{n} e^{x_i}}\right)$

Questions

1. Why do we need activation functions?

Questions

1. Why do we need activation functions?
2. How should we define activation functions, for a layer or for a neuron?

Outline

1. Course Schedule
2. References
3. What is Supervised Learning?
4. What is a neural network?
5. Gradient Descent

Gradient Descent

Let $f : \mathbb{R}^k \to \mathbb{R}$ be a convex function and we want to find its global minimum.

Gradient Descent

Let $f : \mathbb{R}^k \to \mathbb{R}$ be a convex function and we want to find its global minimum. This optimization algorithm is based on the fact that the fastest decreasing direction of the function is the opposite direction of gradient:

$$x_{n+1} = x_n - \alpha \nabla f\left(x_n\right)$$

and $x_0 \in \mathbb{R}^k$ is a arbitrary point.