Deep Learning

Vazgen Mikayelyan

YSU, Krisp

September 16, 2020

WELCOME TO

Deep Learning

Outline

Course Details

- Lectures: **Wednesday 20:00-21:20, Saturday 12:00-14:00**

Course Details

- Lectures: **Wednesday 20:00-21:20, Saturday 12:00-14:00**
- Instructor: **Vazgen Mikayelyan**

Course Details

- Lectures: **Wednesday 20:00-21:20, Saturday 12:00-14:00**
- Instructor: **Vazgen Mikayelyan**
- TA: **Saten Harutyunyan**

Course Details

- Lectures: **Wednesday 20:00-21:20, Saturday 12:00-14:00**
- Instructor: **Vazgen Mikayelyan**
- TA: **Saten Harutyunyan**
- Grading: **Homeworks, Final Exam**

Outline

References

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

Outline

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

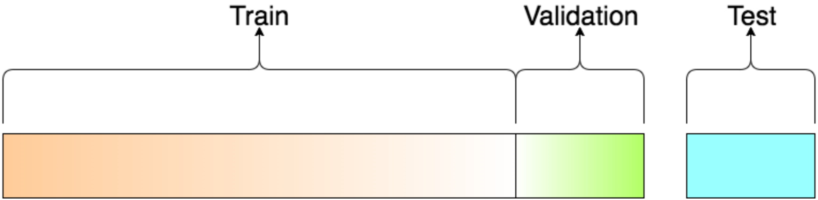
$$(x_i, y_i)_{i=1}^{n}, \text{ where } x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^m, i = 0, 1, \ldots, n.$$

We have to find a function f for which

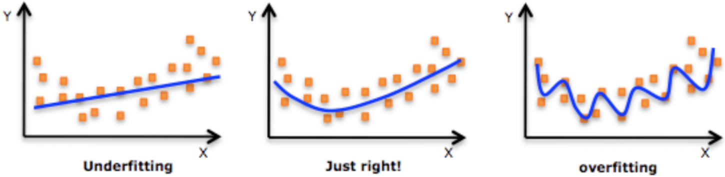
$$f(x_i) \approx y_i$$

and not only for pairs in our data.

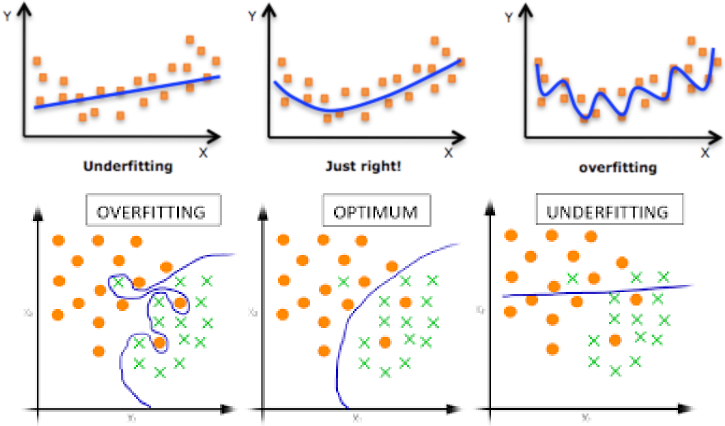
Train, Validation, Test



Overfitting, Underfitting

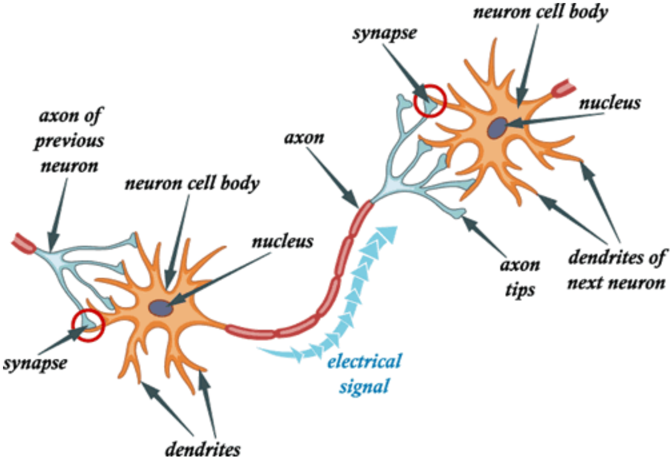


Overfitting, Underfitting



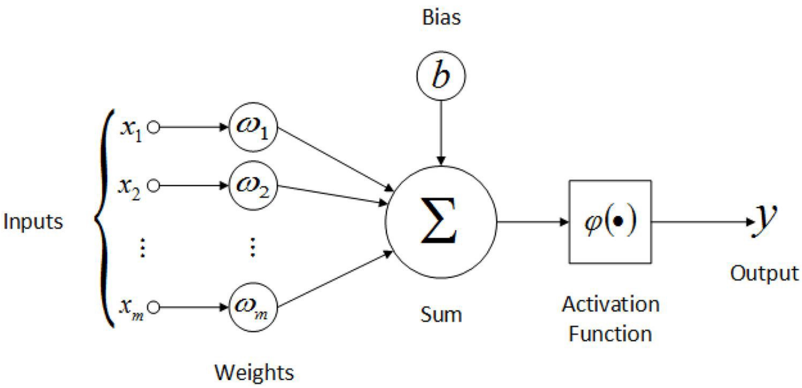
Outline

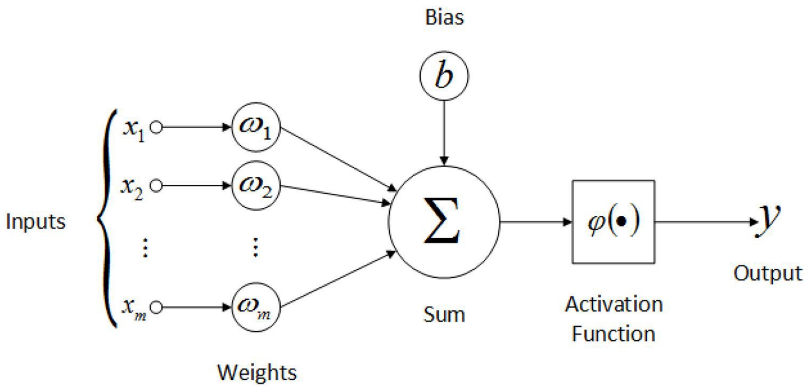
Human Brain



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

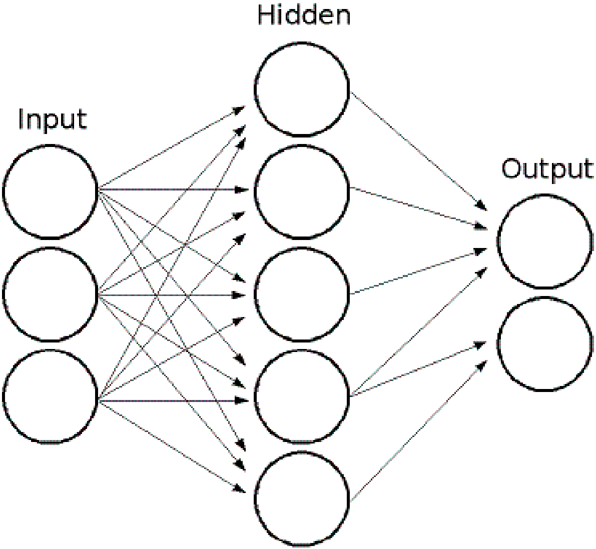




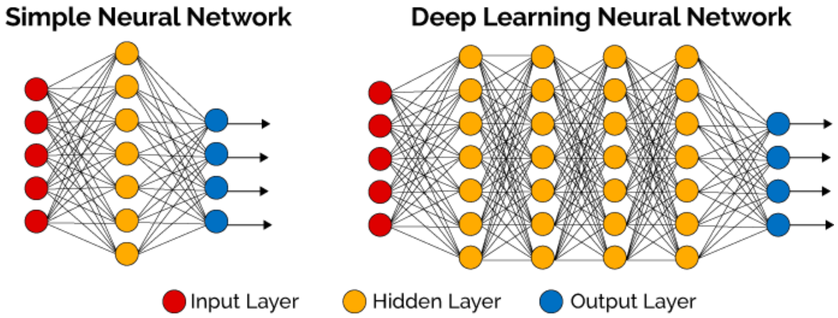
So artificial neuron is a function from $\mathbb{R}^m$ to $\mathbb{R}$, where m is the dimension of input:

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

Hidden Layer



Deep neural network

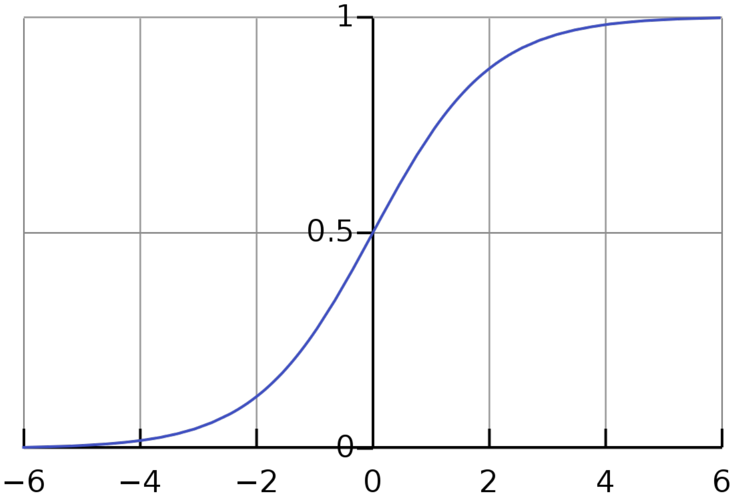


Activation functions

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

Activation functions

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

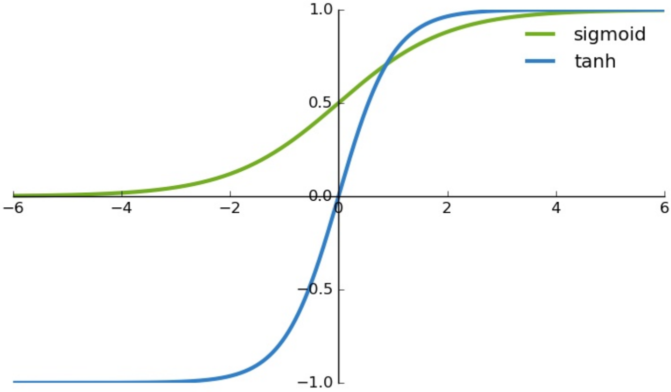


Activation functions

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

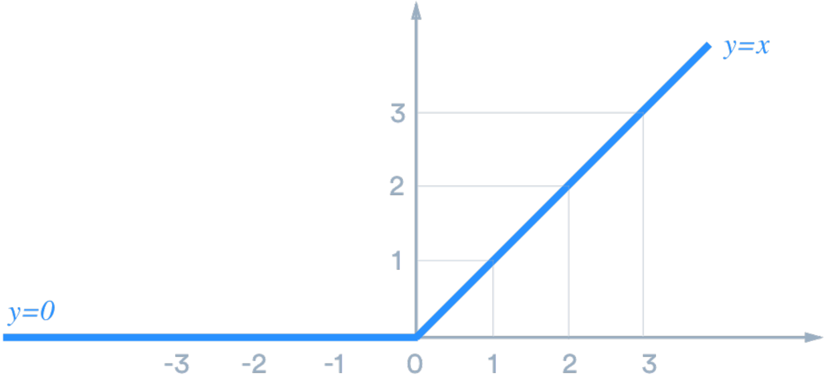
Activation functions

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



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

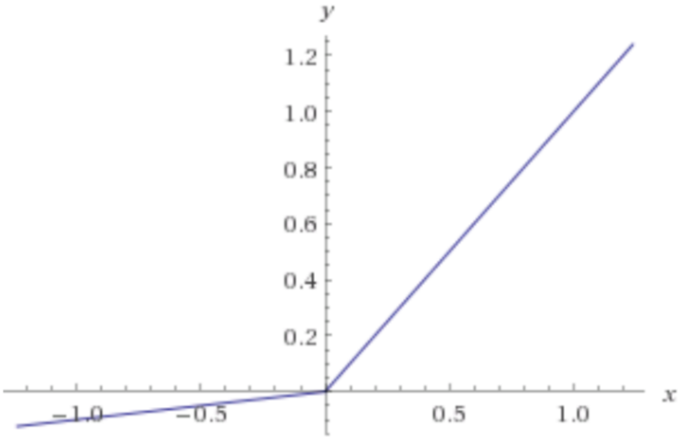
3. Rectified linear unit: $ReLU(x) = \max(0, x)$



Activation functions

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

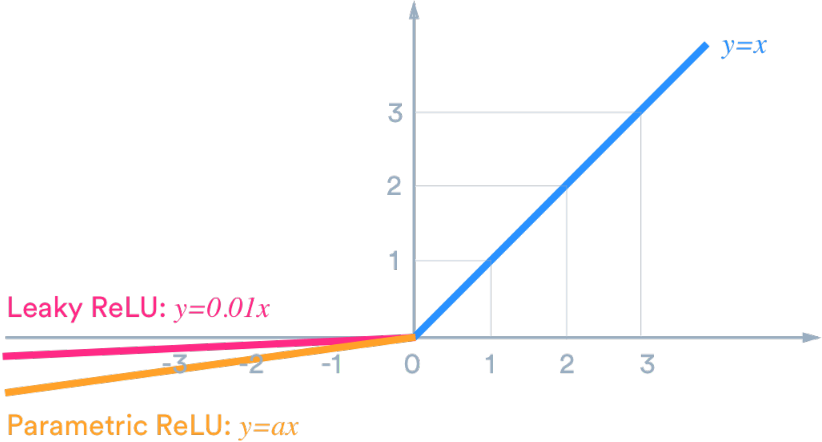
4. Leaky ReLU: $LR(x) = \begin{cases} 0.01x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$



Activation functions

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

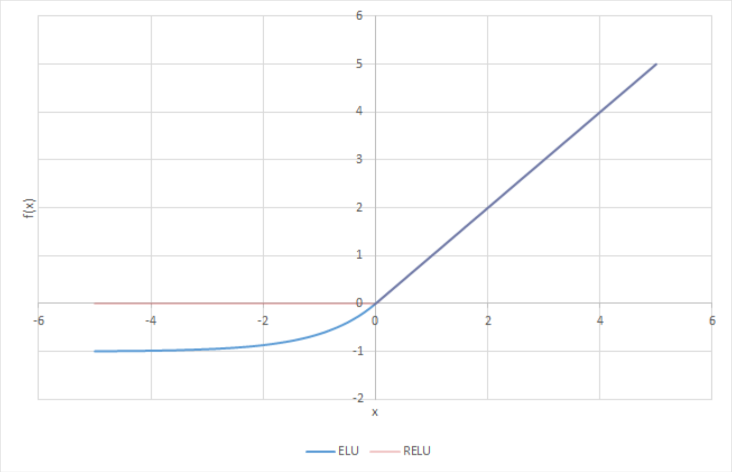
5. Parametric ReLU: $PR(x) = \begin{cases} ax, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$



Activation functions

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

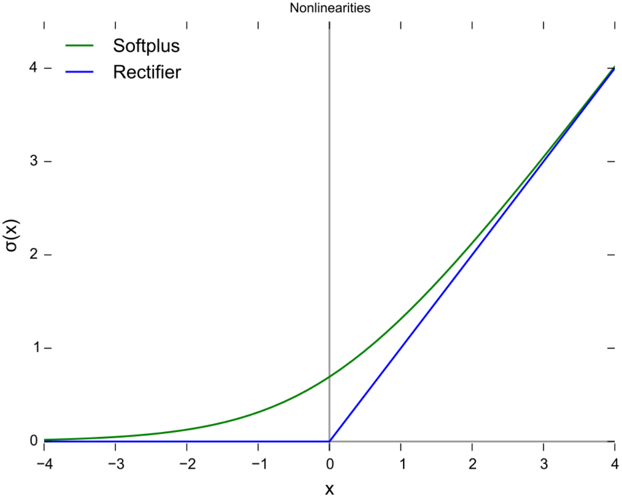
6. Exponential linear unit: $ELU(x) = \begin{cases} a\left(e^x - 1\right), & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$



Activation functions

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

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



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?