MBA USP ESALQ

Deep Learning

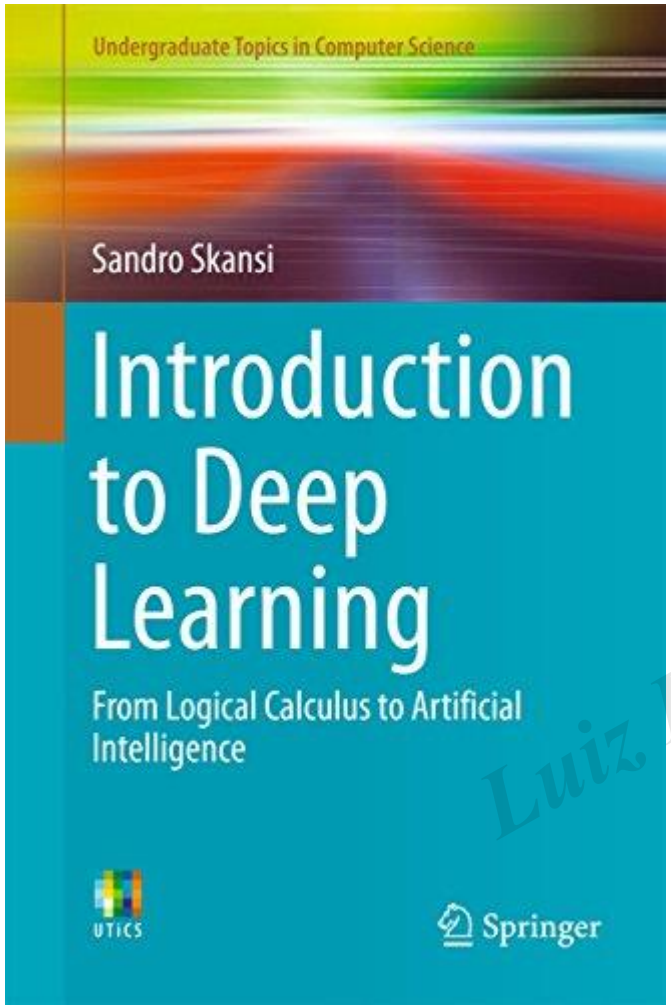
Prof. Dr. Jeronymo Marcondes

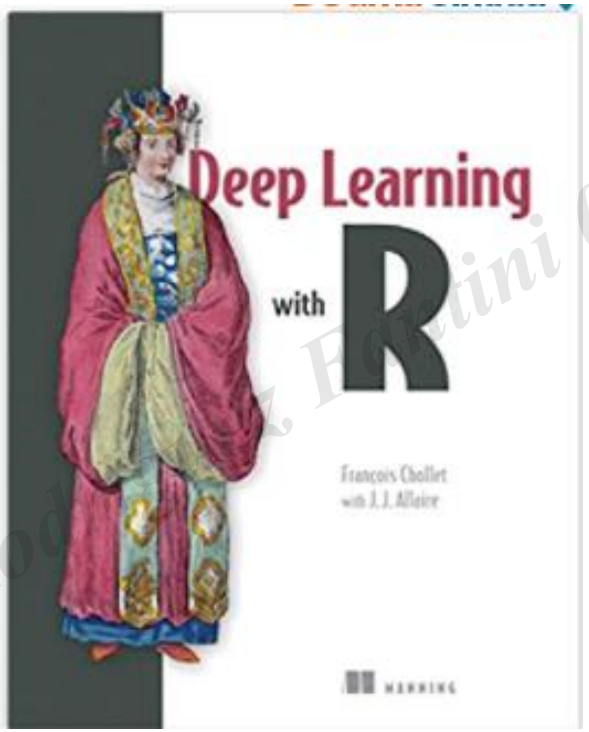
Introduction

- Plan of attack:

1. The role of multilayer neural networks
2. The problem of overfitting.
3. The problem of hyperparameters definition
4. The problem of finding optimal solution

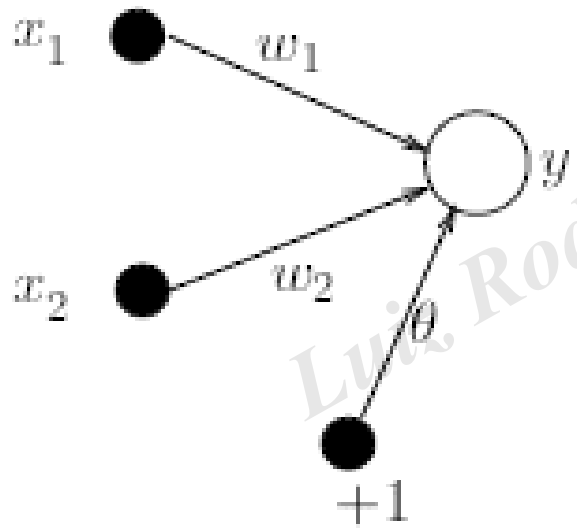
Introduction



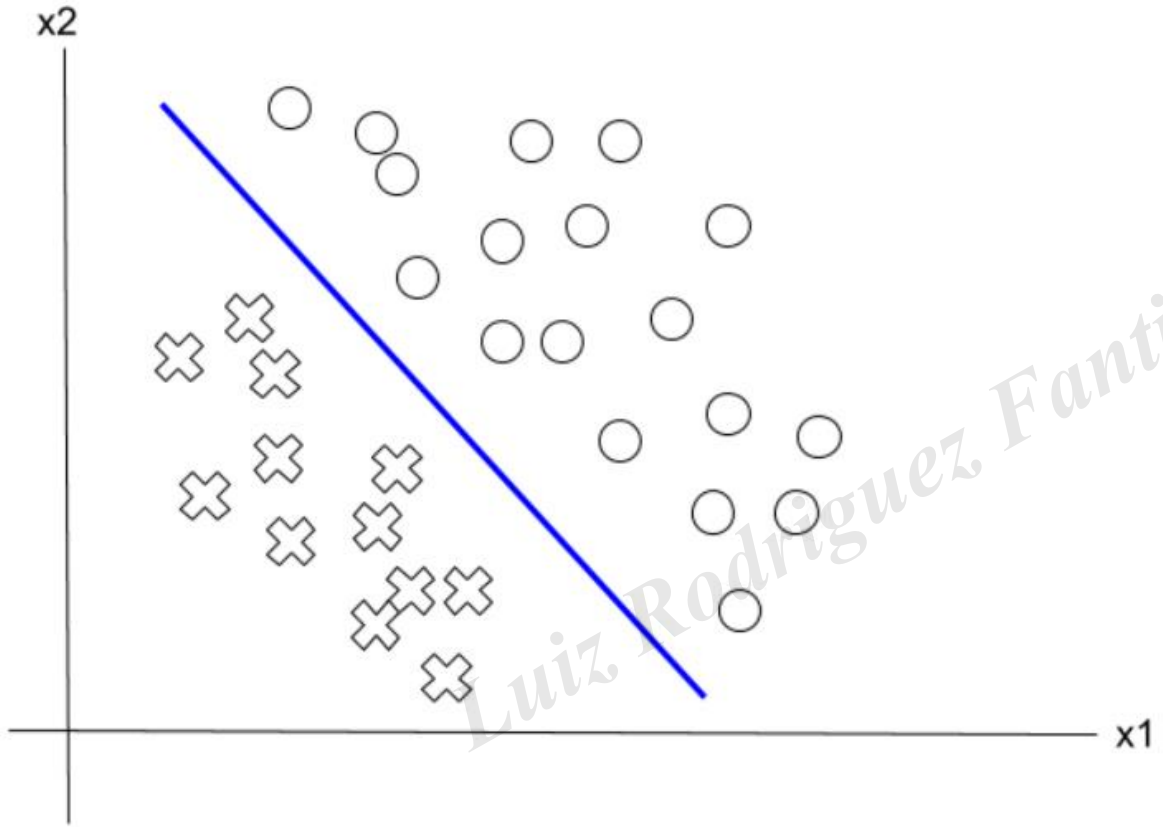


Introduction

- Dark age of artificial neural networks



Introduction

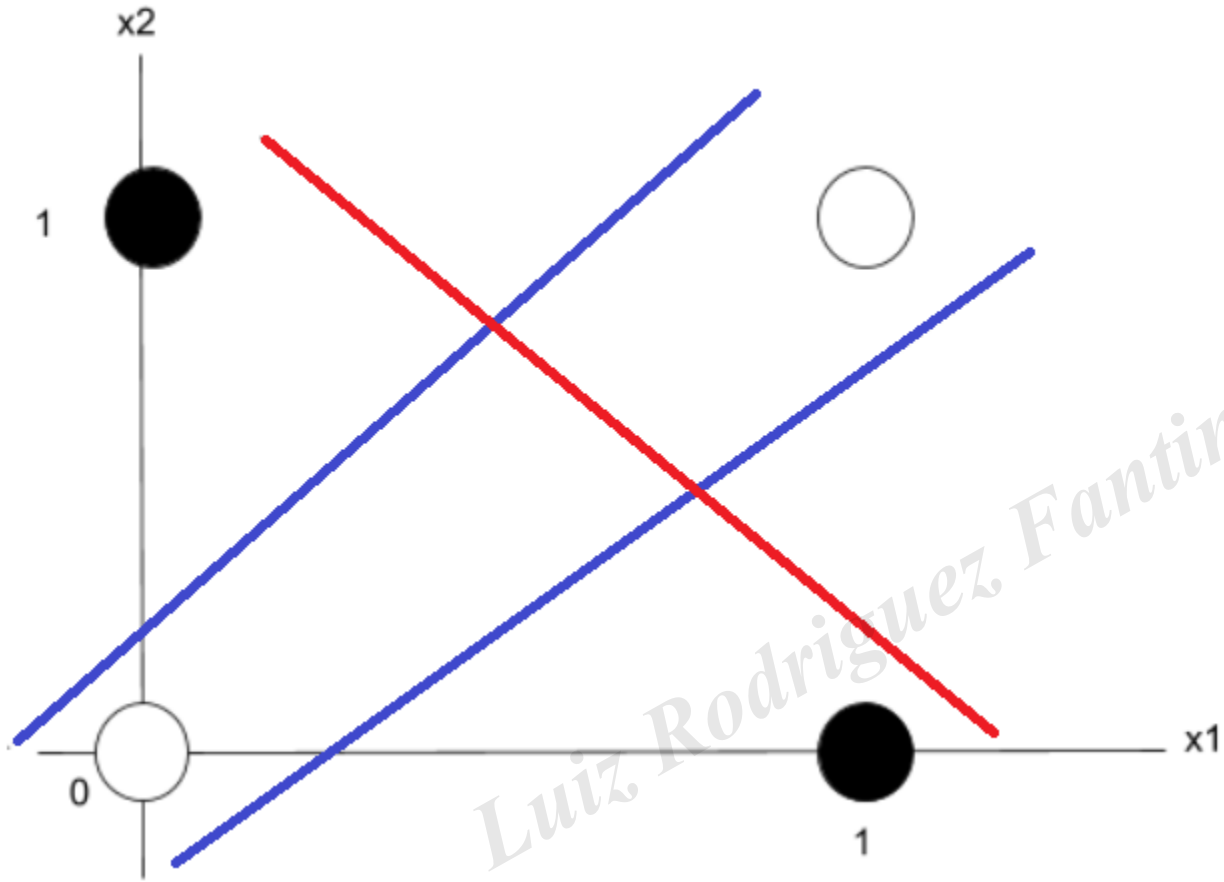


https://automaticaddison.com/linear-separability-and-the-xor-problem/

XoR

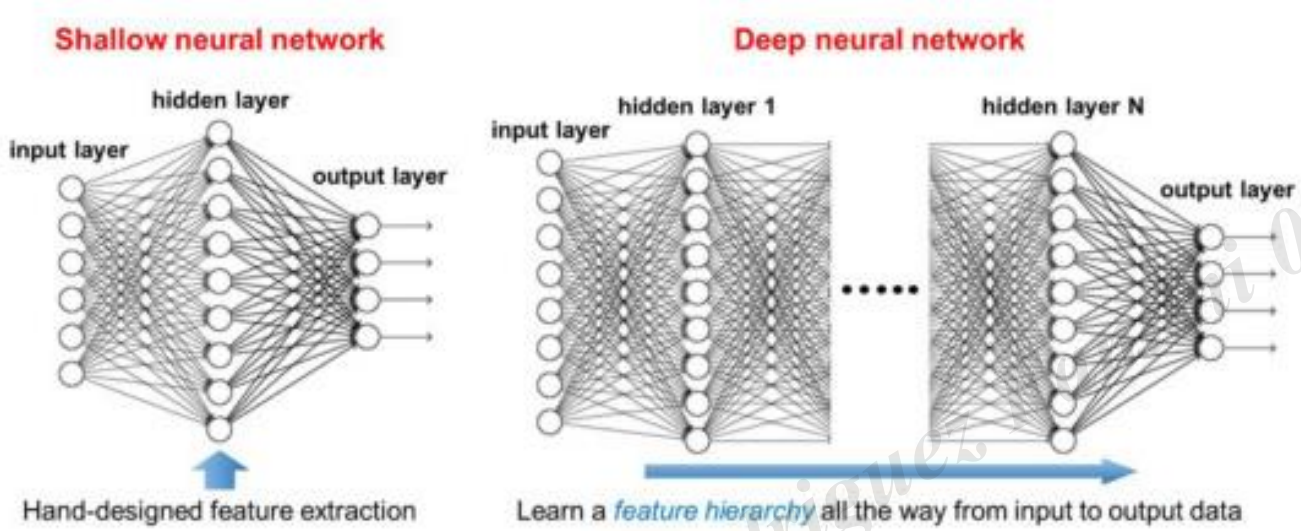
- Exclusive Or

x_0	x_1	d
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1



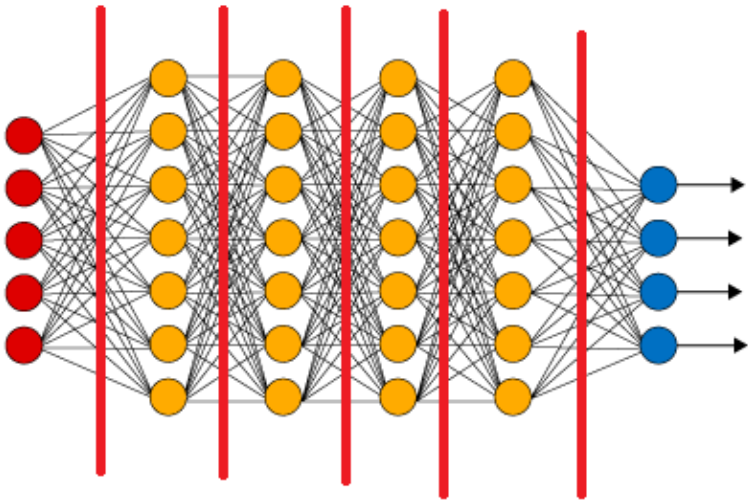
https://automaticaddison.com/linear-separability-and-the-xor-problem/

Multilayer Neural Network



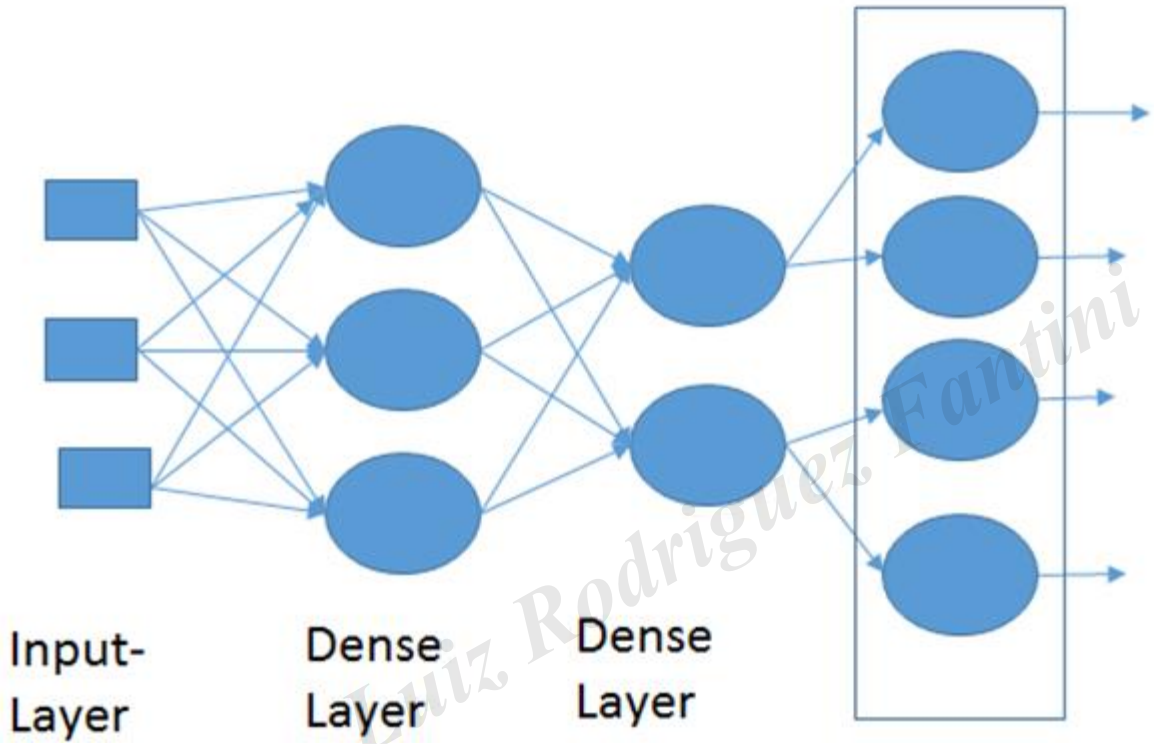
- Many layers allow to identify non-linear relationships.
- In the case of several intermediate layers, the deep learning is obtained.

Network Structure



- Layers – group of neurons in a process phase.
- Dense Layer - It connects each neuron of a layer to every neuron of its preceding layer. For example, if the current layer has 5 neurons and the previous layer has 3, the total of connections is 15.

Dense Layer



Loss Function

- Function that allows to verify how correct is a certain prediction.
- Predicted x Actual
- They are different for continuous and categorical variables

Loss Function

- The categorical variables contain a finite number of different categories or groups. Categorical data may not have a logical order. For example, categorical predictors include gender, type of material, and payment method.
- Continuous variables are numerical variables that have infinite number of values between any two values. A continuous variable can be numerical or of date/time. For example, the length of a piece or the date and time in which a payment is received.
- Discrete variables are numerical variables that have a countable number of values between any two values. A discrete variable is always numerical. For example, the number of complaints of customers or the number of failures or defects.

https://support.minitab.com/

Some error functions

- Mean squared error:

$$EQM = \sum_{n=1}^{k} (Actual - Predicted)^2$$

	Nota	
	Actual	**Predicted**
A	7	8
B	10	9
C	5	10
D	8	8

Some error functions

- Mean Absolute Error

$$EQM = \sum_{n=1}^{k} |Actual - Predicted|$$

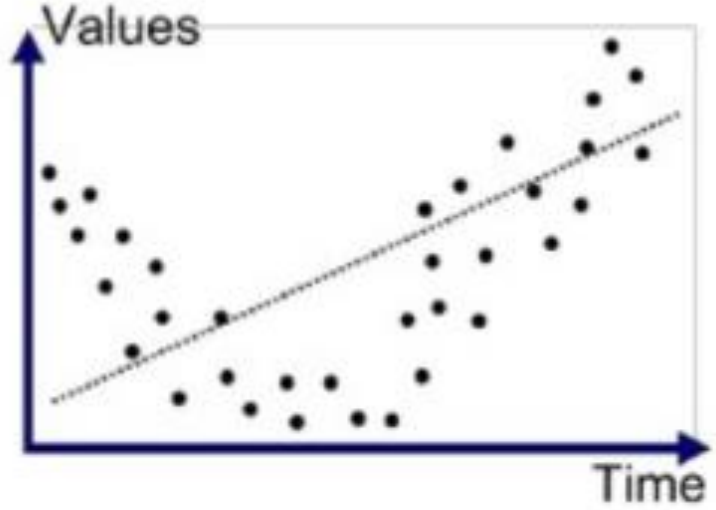
	Nota	
	Actual	**Predicted**
A	7	8
B	10	9
C	5	10
D	8	8

Some error functions

- Functions for continuous output.
- It's necessary other metrics for classifying, such as 1 and 0.
- Binary cross-entropy
- Categorical cross-entropy

Bias x Variance

- What are we looking for?
- What can happen?
- Importance of Generalization – map theory.



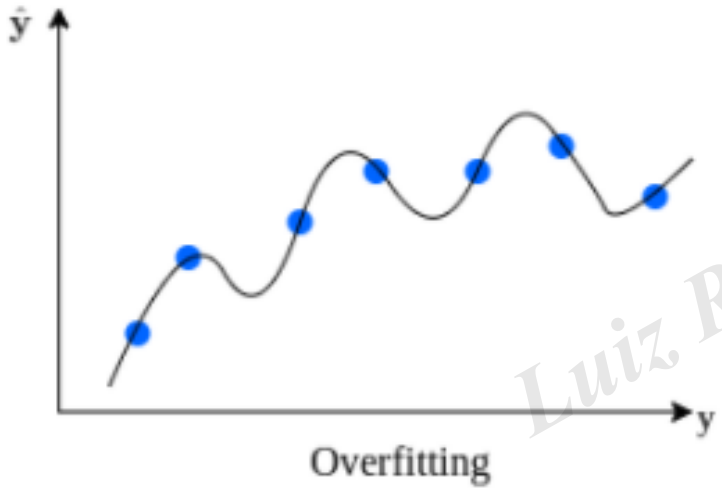
Underfitted

Bias x Variance

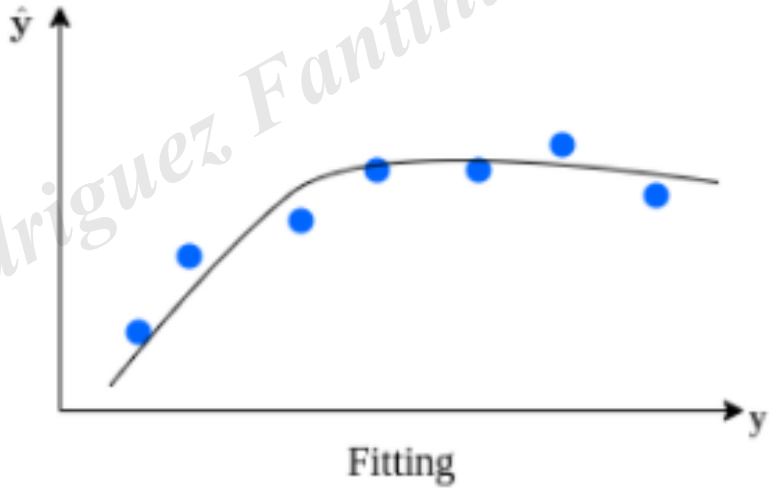
- Absence of Bias - you are right in the average part.
- Variance Reduction – target shooting.

Bias x Variance

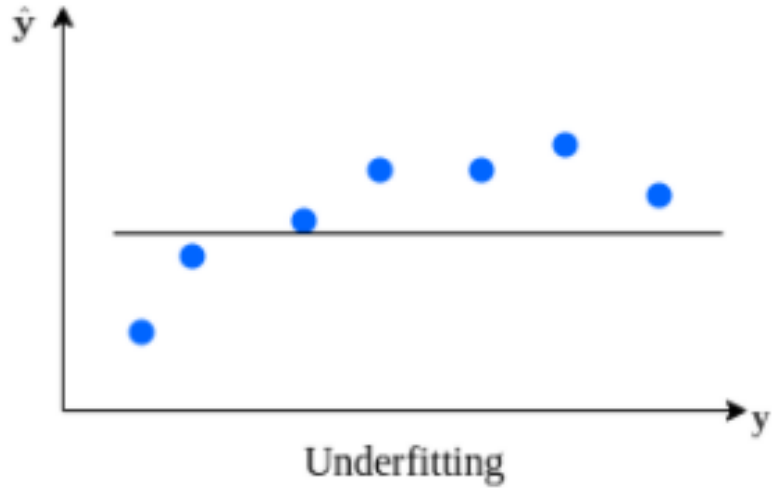
- Concern about the variance.



Overfitting



Fitting



Underfitting

https://www.baeldung.com/cs/epoch-neural-networks

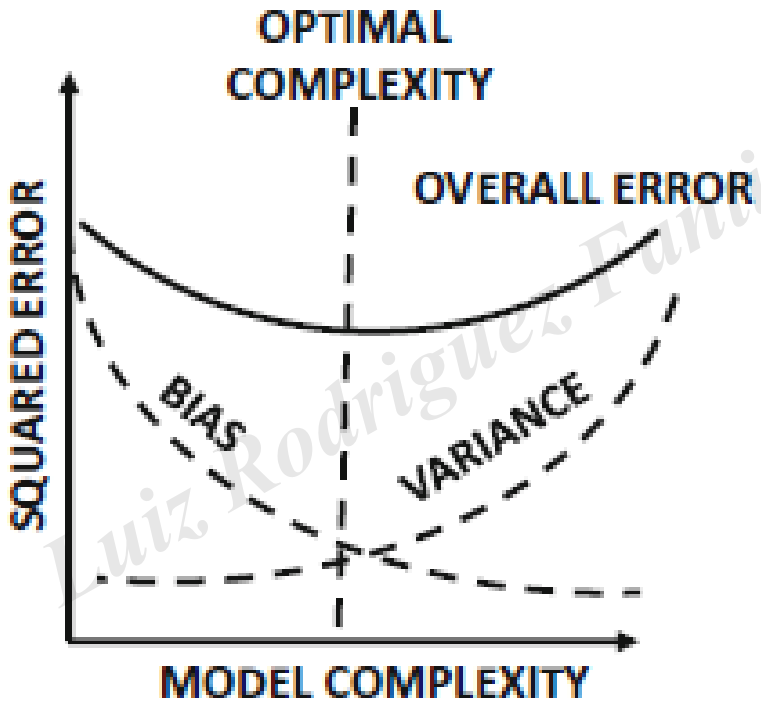
Fitting

- UnderFitting - Your model is subjugating training data when the model has unsatisfactory performance on training data. This occurs because the model is incapable of capturing the relationship between input examples (usually called X) and output values (usually called Y).

Fitting

- OverFitting - Your model is adjusting your training data when you observe that the model has a good performance in the training data but not in the evaluation data. This occurs because the model is memorizing the data that it noticed and is incapable of generalizing for unseen examples. Very common in Neural Networks.

Fitting



Overfitting reduction

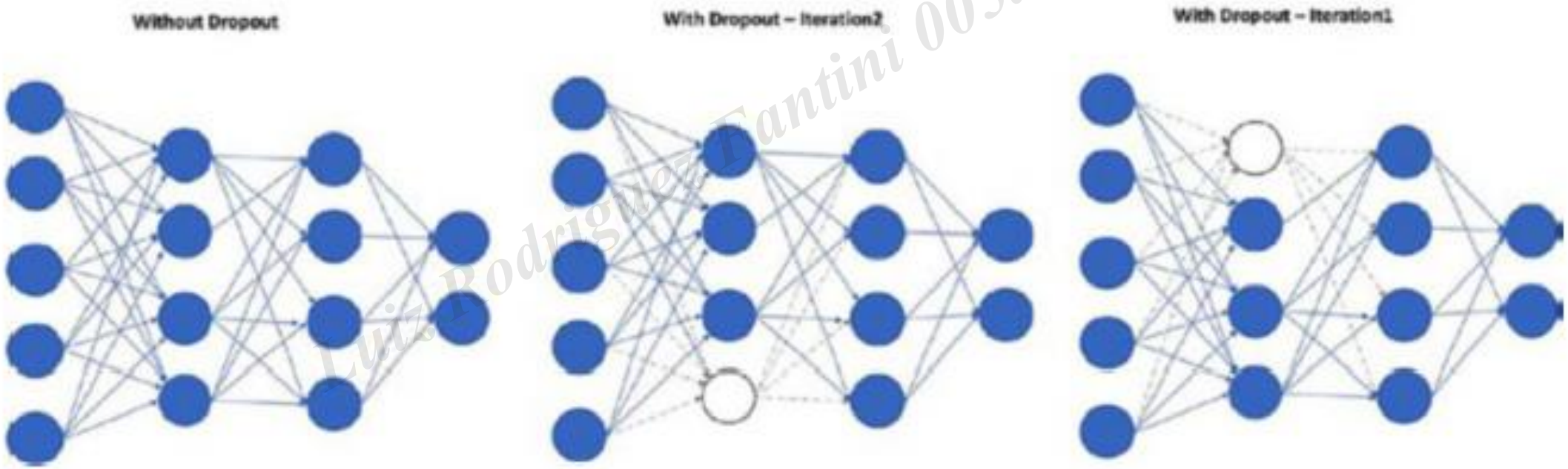
- How to reduce overfitting?
- Reduce number of layers
- More parameters = + memorize capacity
- Regularization

Regularization

- Introduction of “noise”.
- Dropout

The model dismiss or deactivate arbitrarily some neurons for a layer during each iteration. In each iteration, the model notices a slightly different structure of itself to optimize (as a couple of neurons and connections would be deactivated). Suppose that there are two successive layers, H1 and H2, with 15 and 20 neurons, respectively. The application of the elimination technique between these two layers would result to randomly dispose some neurons (based on a defined percentages) for H1, which, therefore, reduce connections between H1 and H2. This process is repeated for each iteration randomly, therefore, if the model needs to learn for a batch and it updated the weights, the next batch can have a very different set of weights and connections to train.

Dropout



L1

- Adjustment of weights made through regularization.
- The focus is no longer on neuron, but on the weight value.
- The weight is adjusted up to zero value depending on its importance.

L2

- Adjustment of weights made through regularization.
- The focus is no longer on neuron, but on the weight value.
- The weight is adjusted up to value close to zero depending on its importance.

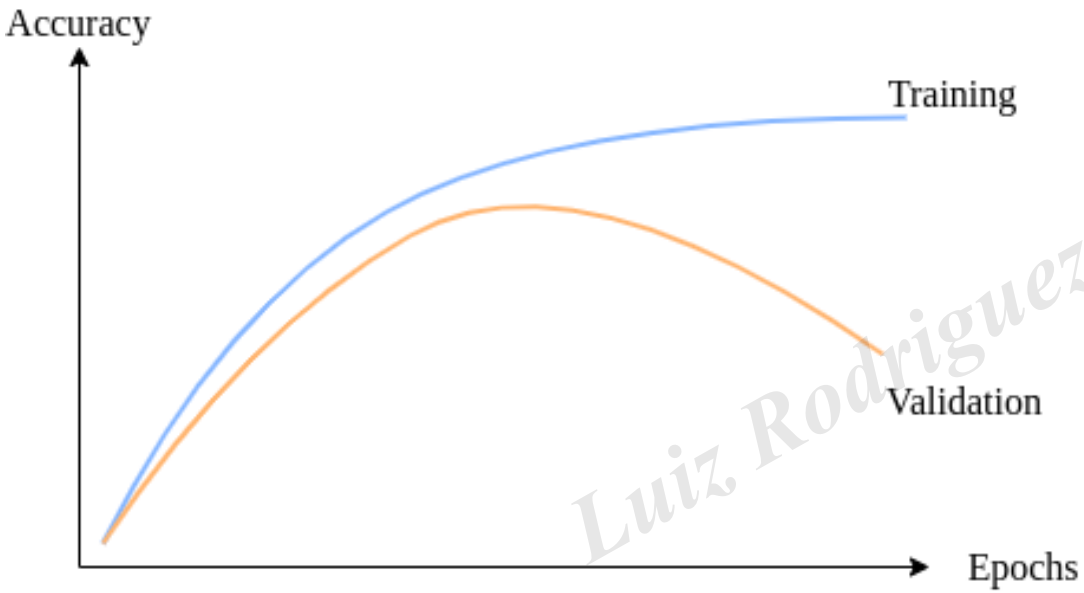
Early Stopping

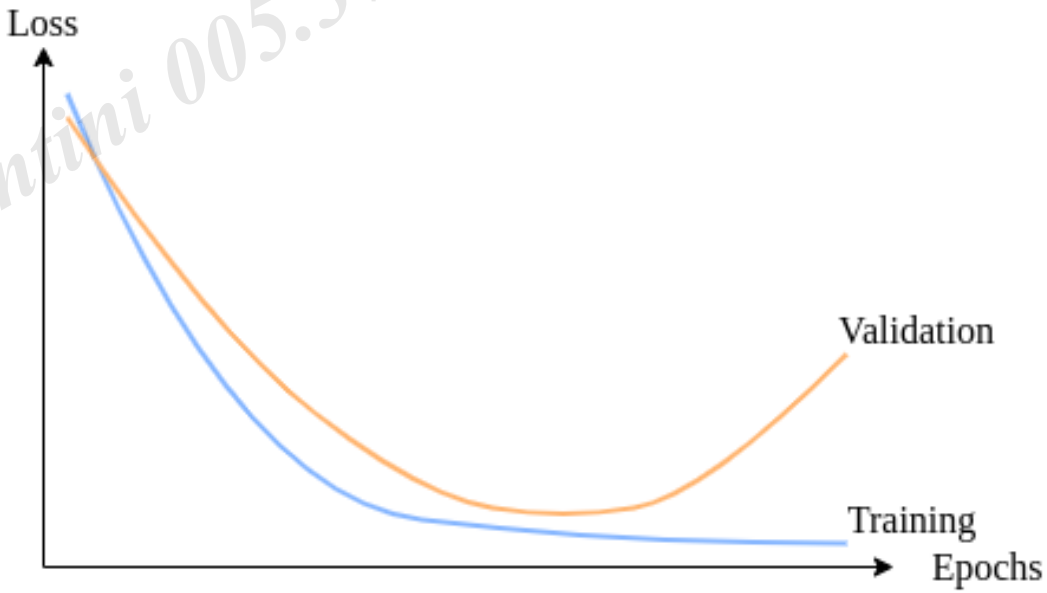
- End of each epoch – verify improvement.
- Isn't the improvement occurring anymore? Stop!
- This avoids overfitting.

Epochs

- How to verify if there is overfitting?
- How often data go through the network.
- Each epoch is a step toward the "optimal" result.
- Convergence.

Epochs





https://www.baeldung.com/cs/epoch-neural-networks

Batch

- How to perform training?
- Techniques that can help in the optimization process.
- Use of batch.

Batch

Normally, the training would be done in batches due to memory restrictions in the system. A batch is a collection of training samples of all input. The network updates its weights after processing all samples in a batch. This is called iteration (that is, a successful passage of all samples in a batch followed by a weight update on the network). The computing of all training samples provided in the input data with updates of batch weight by batch is called an epoch. In each iteration, the network makes use of the function of the optimizer to perform a small change for its weight parameters (that were randomly initialized at the beginning) to improve the final prediction, reducing the loss function. Step by step, with several iterations, and then, several epochs, the network updates its ponderance and learns to make a correct prediction for the the training of samples.

Hyperparameters

- What are they?
- Any number used by the network that is not learned.
- How to determine these values?
- Cross validation? Grid Search?

Hyperparameters

- Let's remember some of them.

1. Learning rate
2. Batch size
3. Epochs
4. Activation function

AUTOML

- Let's discuss a little about this.

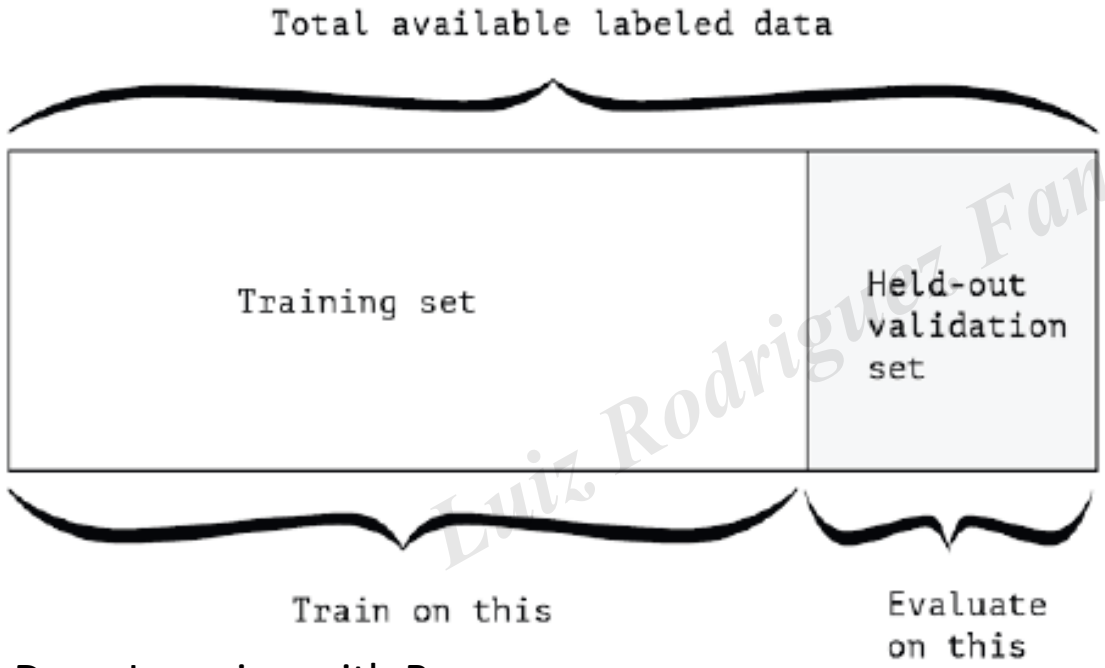
- AutoKeras, H20.

- Transfer Learning.

- Grid Search

Cross validation

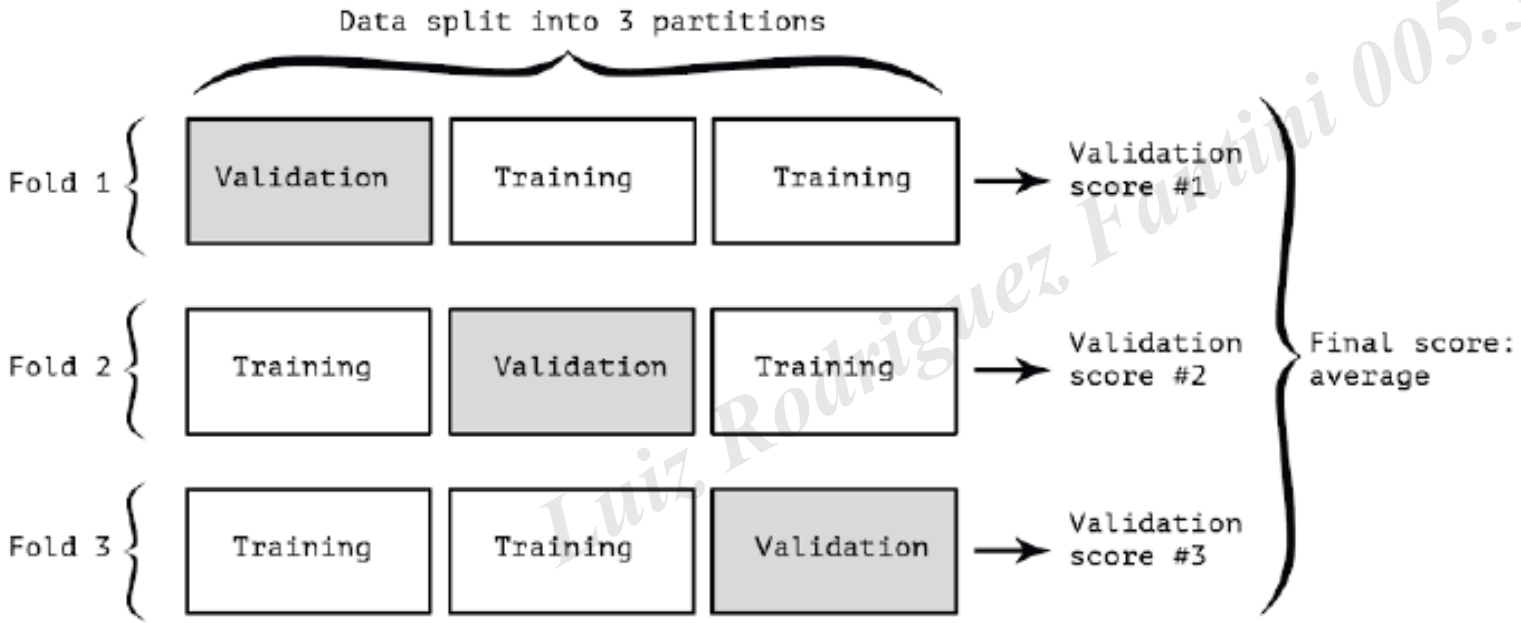
- How does it work?
- Training set, test and validation



Deep Learning with R

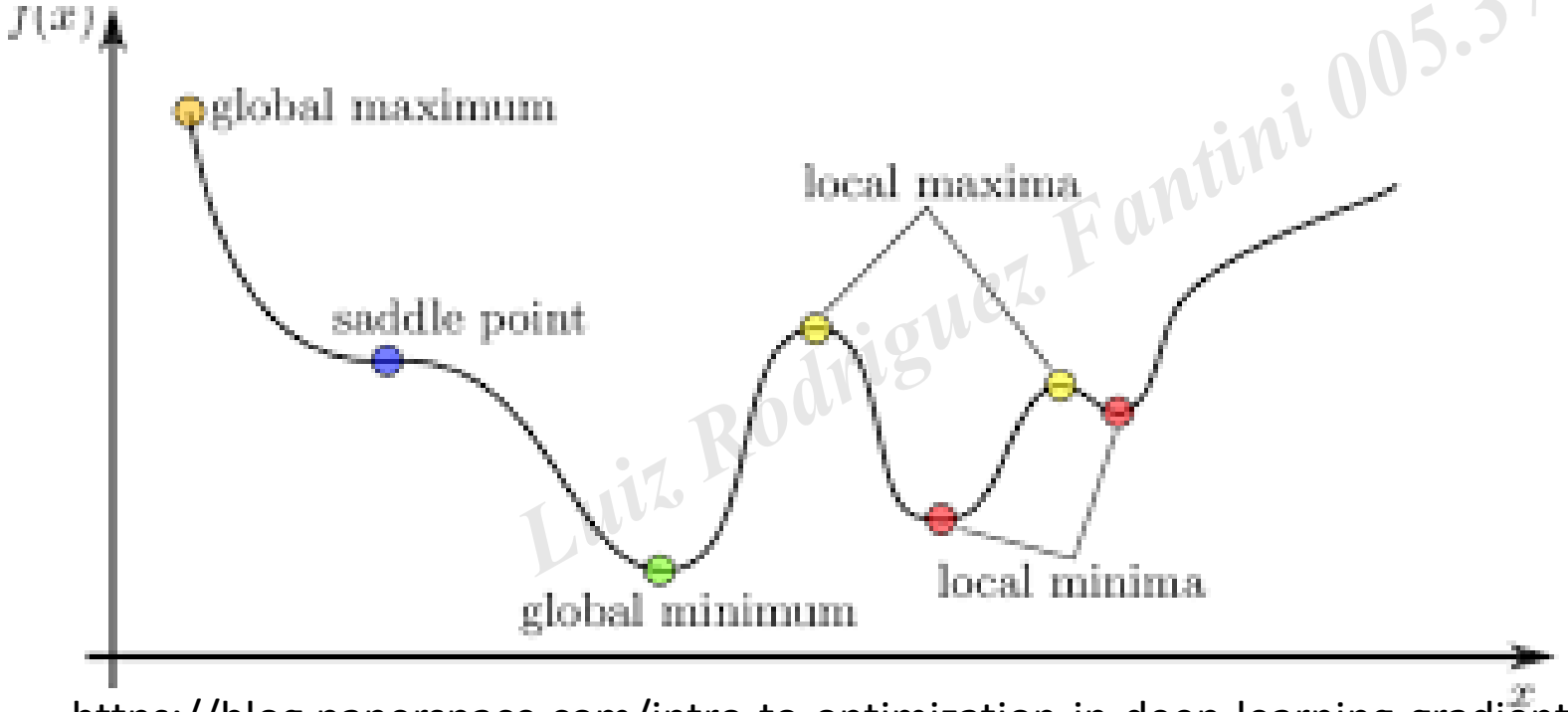
Cross validation

- K fold cross validation



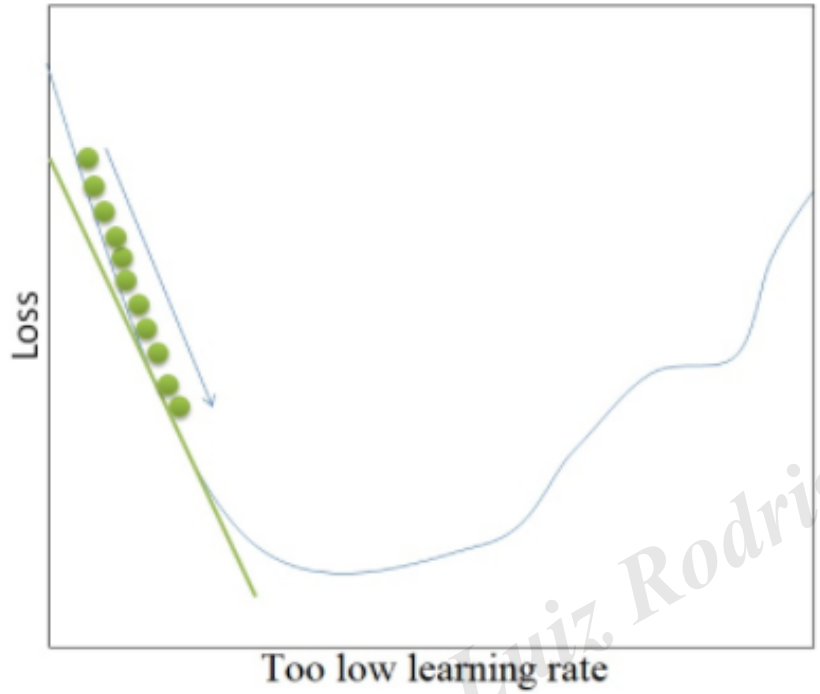
Deep Learning with R

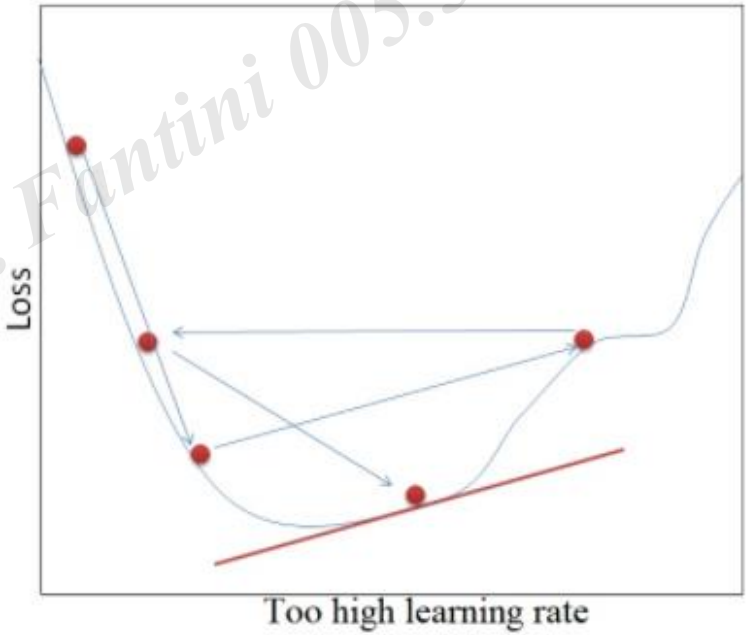
Local minimum and saddle point



https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/

Local minimum and saddle point





https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/

Optimizers

- That is, using the loss function, how to update the weights?
- What is the rule?
- We use the classic model of gradient descent, but there are others.

Optimizers

- ADAGRAD
- ADAM
- RMSPROP

Momentun

- It is also called inertia
- Concept close to the inertia = friction
- Figure above – the ball with speed could pass the saddle point and reach the minimum.

Momentun

$$w_i^{new} = w_i^{old} - \eta \frac{\partial E}{\partial w_i^{old}} + \mu(|w_i^{old} - w_i^{older}|)$$

Introduction to deep Learning

https://www.linkedin.com/in/jeronymo-marcondes-585a26186