ML Project Report
(CS-361)

Learning of Perceptron Model from scratch

Group Number: 4

Mahnoor Awan
Manzoor Ul Haq
Noor Nabi
Syed Wajeh Kazmi

Course: Artificial Intelligence CS361

Instructor: Dr. Junaid Akhtar

University: Namal Institute, Mianwali

Date: December 27, 2020

Table of Contents

1. Learning Capacity 3
2. Theories of Learning 3
   2.1. Evolution of Human Intelligence 3
   2.2. Assessing Intelligence and Its Evolution 4
   2.3. Difference between Human and Non-human Intelligence 4
   2.4. Early Human and Modern Human Intelligence 4
3. ANNs 4
   3.1. What is ANN? 4
   3.2. History 5
      3.2.1. Convolutional Neural Networks 5
   3.3. Model 6
   3.4. Types of Artificial Neural Networks 6
   3.5. Components of ANN 6
      3.5.1. Neurons 6
      3.5.2. Connections and Weights 7
      3.5.3. Propagation Function 7
   3.6. Organization 7
      3.6.1. Hyper Parameter 7
   3.7. Activation Function 7
      3.7.1. Types of Activation Function 8
   3.8. Back Propagation Algorithm 9
   3.9. Practical Applications for Artificial Neural Networks 9
4. Classification 9
   4.1. Structure 10
   4.2. Given Data 10
   4.3. Functions 10
      4.3.1. showInputData() 10
      4.3.2. activationFunction() 10
      4.3.3. actualOutput() 11
      4.3.4. updateWeights() 11
      4.3.5. desiredOutput() 11
      4.3.6. learning() 11

4.3.7. graph() 12
4.3.8. main() 12
4.4. Output 12
5. References 13

1. Learning Capacity

Term learning is defined as gaining new skills, techniques or ideas regarding things happening around one's own environment. This phenomenon of learning doesn't only imply on Human beings but also is valid for other species. Taking birds in our observation, it can fly but it does not fly right after its birth but it learns flying similarly we have seen birds which fly between continents in search of a livable place, at some point in history they would have learnt that migration is needed for their survival. Birds learn to fly, they find place to live and find their food, their reflexes around other species are different like, a pigeon at some places acts aggressive to Humans but another pigeon at other part of the world it won't even fly. This behavior of a bird is developed by the behavior of Humans to that bird.

Pigeon learnt different reflexes in his own capacity a limit which implies to its brain. On the other hand we humans have bigger learning capacity, from the caveman to the modern human being of today we evolved and we also evolved things around us, consider houses, man lived in caves and now he lives in Heated Homes built with wood, cement etc to save himself from harsh cold weather and predators. This is what learning capacity of humans have made them capable of doing.

We look in the past we made progress in different ways by learning from past and experimenting with things today. Other species learn and implement the same way but are being limited by their mental and physical and abilities. There are things which humans cannot do so they want to define ways to make machines capable of doing such tasks and one such way in machine learning. In machine learning you define learning capacity and you define learning approach; it could be self driving cars or your phone's screen time notifier. All the application related to machines and defined by people.

2. Theories of Learning

2.1. Evolution of Human Intelligence

Human intelligence and language go far beyond science, mathematics, biological needs, abstract ideas of human and imaginary thoughts of human species. Many things in the world such as how dark turns into light. Human intelligence arose thoughts and maintain balance between trust and doubt. But this conflict lies at the heart of the scientific method. Many scientists try to understand how the universe really works. Actually, this world push human to use their hidden intelligence. Only human species demanded this evolution because this is particular adaption for them.

2.2. Assessing Intelligence and Its Evolution

Many methods are used to assess intelligence and its evolution such as behavioral measures (involves natural responses in laboratory), artificial measures (involves artificial tools) and neurological measures (involves the study of brain). These three pictures cover up the evolution. No living species are more closely related like dolphin and crows share some intellectual abilities with great apes and humans, but these abilities operate differently.

2.3. Difference between Human and Non-human Intelligence

Intellectual of human is far from other species because human can think and make creative things and explore the hidden gems of the world. Human intelligence is distinguishing by their abstract thinking that how to solve problem or communicate with the people etc.

2.4. Early Human and Modern Human Intelligence

The thinking criteria of human is same from the beginning of life on the world but ways of thinking changes time to time.

"Necessity is the mother of invention"

Now modern human tries to understand and explore the conventions. This period is a revolution in behavioral modernity.

Verily God will not change the condition of men, till they change what is in themselves (13:11)

The human changes its life. If he does not take change about change then there will be no change in the world and then there is no difference between human and non-human. This human does not cease their needs but doing work on this hiddenprecious world. Thinking and make creativity is in the nature of the human. Advance things happened in world because of Quran. Quran opens human's eyes and make great fact on the whole universe. There is a no doubt this is a divine knowledge.

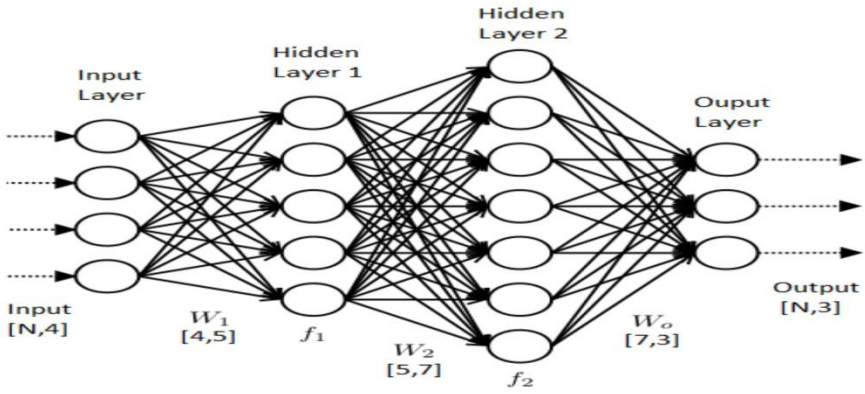
3. ANNs

ANN stands for artificial neural network. What is ANN will tell in detail in the below section.

3.1. What is ANN?

ANN is also called as Neural networks are inspired by the biological neural networks and consider as computer systems vaguely. ANN's are actually based on collection of connected units or nodes same as the neurons in the human mind. These connected units are nodes are called the artificial neurons. Like in human brain each neuron transmits a signal to other same is in the case of these artificial neurons. They transmit signals into one another. The artificial neuron transmit signal to then and then process it. The signal at a connection is a real number and the output of neuron is computed by some nonlinear function of inputs

sum. These connections are called edges. Each edge has some weight at it. The strength of a signal is increased or decreased by weights of its edges.



Artificial Neural Network

ANN initially goes through a training phase where it learns to recognize patterns in data, either visually, naturally, or text. In this monitored phase, the network compares its actual product to what it should have been produced - the desired output. The difference between the two outcomes is adjusted using regression. This means that the network is running backwards, from the output unit to the input units to adjust the weight of its connection between the units until the difference between the actual and desired result produces the lowest error.

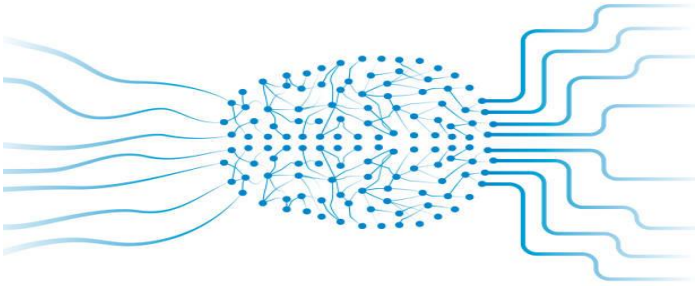
Will discuss some history of the ANN and how they are created and how it all started.

3.2. History

The work on artificial neural network was started by the two scientists Warren McCulloch and Walter Pitts in 1943. They created a computational model for artificial neural network the model was based on an algorithm called threshold logic. The proposed model splits the research on two ways. One focused on the biological neural networks and other in the application of neural networks in artificial intelligence. This work led to the research in the nerve networks and finite automata. In 1975 the research was started on the back propagation this algorithm leads to the training of the multi layered networks.

3.2.1. Convolutional Neural Networks

In 2011 the work started on the deep learning feed forward networks alternated between convolutional layers and max-pooling layers topped by several fully or sparsely connected layers followed by a final classification layer. These layers include such filters which submerged in the input.



3.3. Model

The ANN is consisting if the collection of the simulated neurons. Each neuron here is actually a node which is connected to the other nodes by the links. Each node has its own weight which measures one influence on the other.

3.4. Types of Artificial Neural Networks

There are two Artificial Neural Network topologies

- Feedforward
- Feedback

Feedforward ANN

In this ANN, the information flow is unidirectional. A unit sends information to other unit from which it does not receive any information. There are no feedback loops. They are used in pattern generation/recognition/classification. They have fixed inputs and outputs.

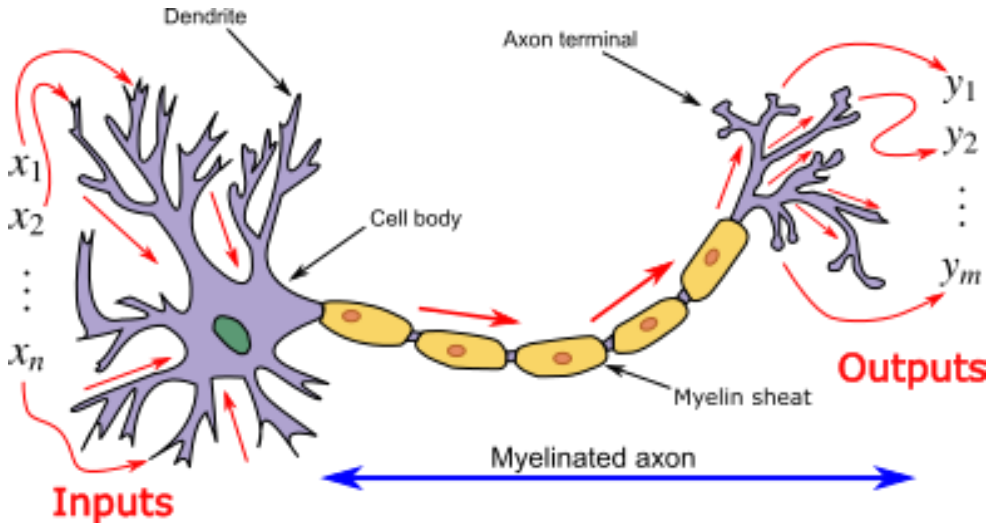
Feedback ANN

Feedback loops are allowed. They are used in content addressable memories.

3.5. Components of ANN

3.5.1. Neurons

The ANNS are composed of the artificial neurons that are derived from the biological neurons. Each artificial neuron has its own input and the output generated from it can be sent to the multiple other neurons.To detect a neuron release, we first take a weighted sum of all the inputs, weighed by the weights of the connection from the input to the neuron. We include the word bias in this amount. This weighty sum is sometimes called active. This weighted amount is then transferred to the activation function (usually non-linear) to produce the result. Initial external data input, such as images and text. The final results accomplish the task, such as recognizing the object in the illustration.



Biological Neural Network

(Figure illustrates the Neuron and militated axon, with signal flow from inputs at dendrites to outputs at axon terminals)

3.5.2. Connections and Weights

A network consists of connections, each of which provides the release of one neuron as an input to another neuron. Each connection is given a weight that represents its relative value. A given neuron can have many input and output signals.

3.5.3. Propagation Function

Involves input to a neuron from the effects of previous neurons and its connections as a weighted mass.

3.6. Organization

Neurons are usually organized into multiple layers, especially in deep learning. Single-layer neurons only connect to pre- and post-layer neurons immediately. The layer that receives the external data is the input layer. A layer that produces the effect of maintaining the output layer. Among them are zero or more hidden spaces. One layer and non-play networks are used. Between the two layers, multiple connection patterns are possible. They can be fully connected, with every neuron in one layer connecting to every neuron in the next layer.

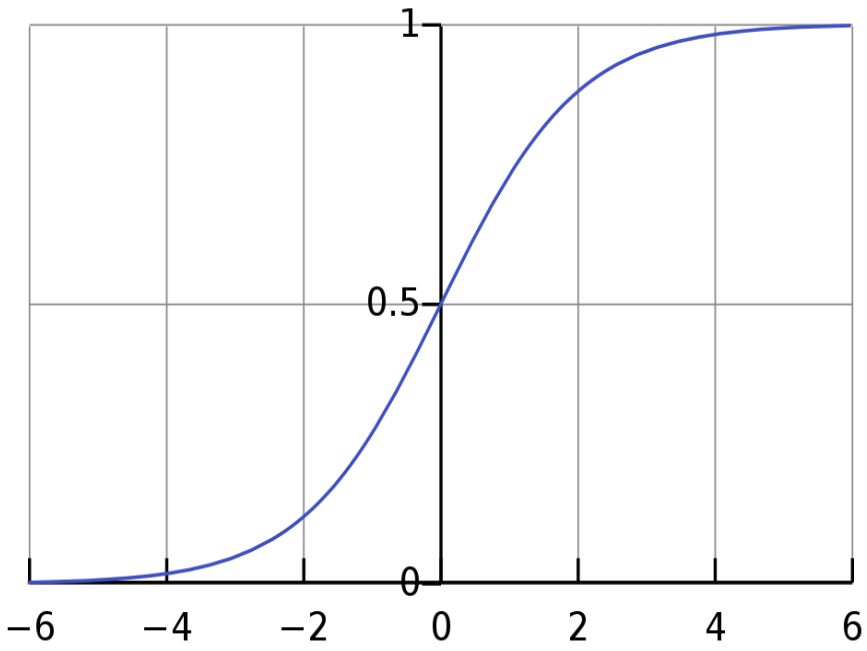
3.6.1. Hyper Parameter

A hyper parameter is a constant parameter whose value is set before the learning process begins. The values of parameters are derived via learning.

3.7. Activation Function

In artificial neural networks, a node activation function defines the output of that given node input or set input.

The standard integrated circuit can be identified as a digital network of startup operations either "ON" (1) or "OFF" (0), depending on the input. This is similar to the behavior of direct perceptron in neural networks. However, non-linear functions only allow those networks to calculate illegal problems using only a small number of nodes, and those activation functions are called nonlinearities.



Activation Function

3.7.1. Types of Activation Function

The activation functions have three types.

- Ridge Activation Functions
- Radial Activation Functions
- Folding Activation Functions

Ridge Activation Functions

Ridge functions are univariate functions acting on a linear combination of the input variables. Often used examples include:

- Linear Activation
- ReLUActivation
- Heaviside Activation
- Logistic Activation

Radial Activation Functions

A special class of activation functions known as radial basis functions (RBFs) are used in RBF networks, which are extremely efficient as universal function approximations.

Folding Activation Functions

Folding activation functions are extensively used in the pooling layers in convolutional neural networks, and in output layers of multiclass classification networks. These activations perform aggregation over the inputs, such as taking the mean, minimum or maximum.

3.8. Back Propagation Algorithm

It is training or learning algorithm. It reads by example. When you submit to the algorithm an example of what you want the network to do, it changes the weights of the network so that it can extract the desired inputs for certain completion of the training.

Back Propagation networks are ideal for simple pattern recognition and mapping tasks.

3.9. Practical Applications for Artificial Neural Networks

Artificial neural networks pave the way for life-changing applications designed for all sectors of the economy. Artificial intelligence platforms built on ANNs disrupt traditional ways of doing things. From translating web pages into other languages to having an online food order assistant to discuss problem-solving discussions, AI platforms make it easy to transact and make services accessible to all at minimal cost.

Artificial neural networks have been used in all workplaces.

- **Email Service**
  Providers use ANNs to detect and remove spam from a user's inbox.
- **Asset Managers**
  Use it to predict company stock management.
- **Credit Rating**
  Firms use them to improve their credit.
- **Commerce Platforms**
  They use to customize recommendations totheir audience.
- **NLP**
  Discussions are supplemented by ANNs for natural language processing.
- **In-depth Learning Algorithms**
  In-depth learning algorithms that use ANN to predict event opportunities.

4. Classification

We are classifying given set of points depending upon their output which is also given and the classifier here is a line dividing the set of points w.r.t output either 1 or -1 using ANN model that is perceptron model.

Our classifier that is a line which is being learned on every input point in each iteration and this will learn a best line on upgrading the weights – coefficients in equation of a line. Updating weights are responsible of rotation and translation of a line. Iterations will stop when the weights remain same for every input in an iteration and these weights are of that line which perfectly classify the set of points.

4.1. Structure

We have a made a class named **Perceptron** which contains all the functions in it which is to be described in section 4.3. Its constructor function is containing all the input data that will be used in our whole code.

4.2. Given Data

Input data or given data is declared and initialized in the constructor of the class, when the class is called it will automatically be initialized by the training inputs. Given data contains a constant of line that will remain 1, matrix of weights, learning rate, number of iterations, training input and training output.

x1	x2	Desired
1	1	1
9.4	6.4	1
2.5	2.1	1
8	7.7	-1
0.5	2.2	-1
7.9	8.4	-1
7	7	-1
2.8	0.8	1
1.2	3	-1
7.8	6.1	1

4.3. Functions

4.3.1. showInputData()

This function is responsible for printing out the training input data and desired output in the form of a table as shown below.

```
def __init__(self):
    self.x0 = 1
    self.W = [-0.6, 0.75, 0.5] # initial weights
    self.l_r = 0.02 # learning rate
    self.epochs = 30 #total iterations over the training data

    self.trainingInput = [[1,1],[9.4,6.4],[2.5,2.1],[8,7.7],[0.5,2.2],[7.9,8.4],[7,7],[2.8,0.8],[1.2,3],[7.8,6.1]]
    self.trainingOutput = [1, 1 ,1, -1, -1, -1, -1, 1, -1, 1]  #Desired Output
```

4.3.2. activationFunction()

This function checks the actual output whether it is greater than zero or less than zero. If it is greater or equal to zero, then this function returns 1 otherwise it will return -1.

```
def activationFunction(self, Output):
    if Output >= 0:
        return 1
    else:
        return -1
```

4.3.3. actualOutput()

In this function, we are making equation of a line on the basis of the weights and apply the input point to get the actual output. Both are the weights and input point are given as input to this function.

```
def actualOutput(self, Weights, X):
    Output = (Weights[0] * self.x0) + (Weights[1] * X[0]) + (Weights[2] * X[1])
    return self.activationFunction(Output)
```

4.3.4. updateWeights()

This function is updating the weights based on the learning of a line. It takes actual output, desired output, input point, previous weights and learning rate as input and returns an updated weights matrix.

```
def updateWeights(self, actual_Output, desired_Output, X, Weights, learningRate):
    updatedWeights = [0, 0, 0]
    updatedWeights[0] = Weights[0] + (learningRate * (desired_Output - actual_Output) * self.x0)
    updatedWeights[1] = Weights[1] + (learningRate * (desired_Output - actual_Output) * X[0])
    updatedWeights[2] = Weights[2] + (learningRate * (desired_Output - actual_Output) * X[1])
    return updatedWeights
```

4.3.5. desiredOutput()

In this function, we are matching the input point with the training input data and returning the desired output of that input data point.

```
def desiredOutput(self, trainInput, trainOutput, X):
    for i in range(10):
        if trainInput[i] == X:
            index = i
            break
    return trainOutput[index]
```

4.3.6. learning()

This function is responsible of doing iterations and count number of iterations in which line is perfectly being learned. In each iteration, previous weights are saved in a variable and there is a inner loop too which checks all inputs points whether the weights are

learned on any input or it is same as the previous weight. If it is same, then the function returns the weights and the number of iterations.

```
def learning(self, Weights, trainInput, trainOutput, learningRate):
    count = 0
    while True:
        previousWeights = Weights
        count += 1
        for X in trainInput:
            actual_Output = self.actualOutput(Weights, X)
            desired_Output = self.desiredOutput(trainInput, trainOutput, X)
            Weights = self.updateWeights(actual_Output, desired_Output, X, Weights, learningRate)
        if Weights == previousWeights:
            return Weights, count
```

4.3.7. graph()

This function takes the weights, input training points, desired ouput and draw a graph showing points and a line learned based on these weights which classify the points w.r.t their outputs.

```
def graph(self, Weights, trainInput):
    x = np.linspace(0,10,100)
    y = -((Weights[1] * x) + (Weights[0] * 1)) / Weights[2]
    plt.scatter(*zip(*trainInput))
    plt.plot(x, y, '-r', label='Classifier')
    plt.title('Classifier')
    plt.xlabel('x', color='#1C2833')
    plt.ylabel('y', color='#1C2833')
    plt.legend(loc='upper left')
    plt.grid()
    plt.show()
```

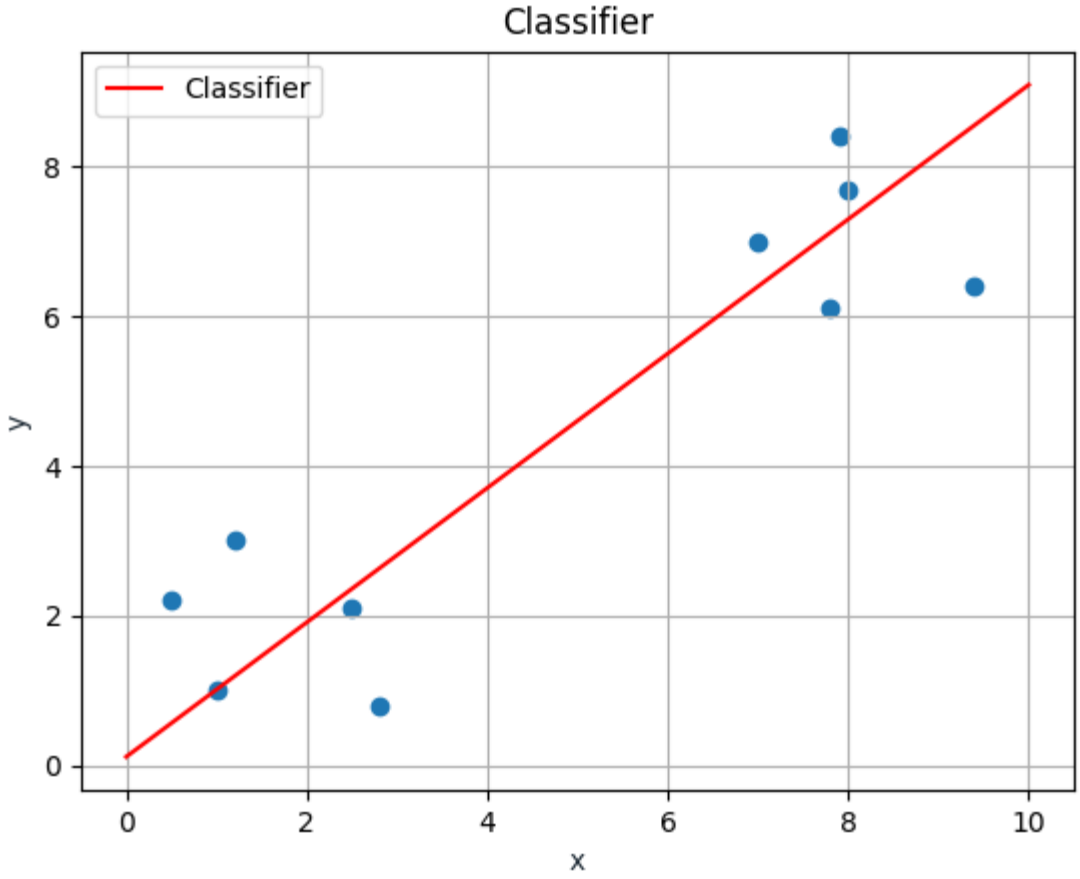
4.3.8. main()

This function first calls the showInputData() which displays the data. Secondly, it calls learning() function and stores the weights and number of iterations in variables returned by it. Lastly, it calls the graph() function to draw a graph to show a classified data.

```
def main(self):
    self.showInputData(self.trainingInput, self.trainingOutput)
    Weights, self.epochs = self.learning(self.W, self.trainingInput, self.trainingOutput, self.l_r)
    print("Epochs: ", self.epochs)
    self.graph(Weights, self.trainingInput)
```

4.4. Output

Output of the whole program is a graph showing that there a line being learned on the perceptron model to classify data w.r.t their desired output.



5. References

- https://en.wikipedia.org/wiki/History_of_artificial_neural_networks
- https://en.wikipedia.org/wiki/Artificial_neural_network
- https://www.investopedia.com/terms/a/artificial-neural-networks
- https://en.wikipedia.org/wiki/Activation_function
- https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.html
- Plotting a list of (x, y) coordinates in python matplotlib - Stack Overflow
- https://stackoverflow.com/questions/21519203/plotting-a-list-of-x-y-coordinates-in-python-matplotlib
- https://www.nature.com/articles/s41437-019-0214-2.pdf?origin=ppub
- https://arxiv.org/pdf/1308.5034.pdf