Research Paper on Artificial Neural Networks

Rohan Sinha

Abstract

An Artificial Neural Network is an information processing paradigm which was inspired from the human brain. Just like human brain, which is highly capable of understanding and perceiving new things in the environment, a machine would be designed that learns to interpret new information and process it on its own, instead of being programmed by a human. It is made up of highly functioning nodes called neurons which receive an input and compute a value and forward it to other neurons.

This paper talks about Artificial Neural Network, its working and learning procedure and its application in the current world. This paper also gives an example of a problem solved using neural network.

Introduction

Artificial Neural Network was originally motivated to achieve the goal of having machines that can mimic the brain. The human brain is one of the most complex organs of human body. It can learn to see, process images, process our sense of touch. Human brain has the ability to learn from the inputs it received. It seems that in order to create a machine that can mimic human brain, one needs to write a lot of different programs. However, there is only a single algorithm involved. Let us take an example. The auditory cortex is the part of brain responsible for the processing of auditory information in humans and other vertebrates. It is connected to the ear which routes the signals to the cortex. Some neuroscientists have conducted experiments on animals where they cut the wire connecting the ear to the cortex and rewiring it so that the signal form eyes to the optic nerve are routed to the cortex. The surprising result is that the animal develops completely functioning pathways in the auditory part of the brain. It sees the world with the brain tissue that was only assumed to be capable of hearing sounds. Another example is BrainPort which helps blind people to see. A grayscale camera is strapped to the forehead of the person which takes low resolution grayscale images of what is in front of him or her. A wire is connected to the electrodes, which are placed on the person's tongue so that each pixel is mapped to a location on the tongue. The brain eventually learns to interpret and use the information coming from tongue as if it were coming from eyes [1]. Implantation of third eye on a frog is another proof of brain's complexity. An experiment was conducted in which a third eye was implanted on a frog, and eventually the frog's brain learned how to use it as well. All these examples prove us that human brain works according to a single algorithm which makes it capable of learning through inputs and adapting to changes.

This led to the idea that if one is able to figure out the algorithm used in human brain and implement it on a computer, it will result in a real progress towards the Artificial Intelligence dream of building super intelligent machines.

History

The origin of neural networks goes back to 1943 when Warren McCulloch and Walter Pitts implemented a computational model based on mathematics and algorithms known as threshold logic. This model resulted for the research to split into two different approaches – Biological Processes in human brain and application of neural networks [2].

Neural Network became a trending topic in 1980s, however it slowly died off in the next decade. The reason being the fact that technology back then was not up to standard to facilitate Neural Network. In order to use neural networks, one needs two things- Data and Processing Power.

Figure 1

Figure 1 shows the evolution of storage capacity. The maximum storage in 1956 was 5 MB. A company had to pay $2500 to rent that hard drive. In 1980 the situation improved a little bit. The capacity increased to 10 MB. Today in 2017, we have got 256 GB SSD card for $150. In Figure we can see the decline in hard drive cost per gigabyte, dropping down to zero. Today we have Google Drive and Dropbox which are free of cost.

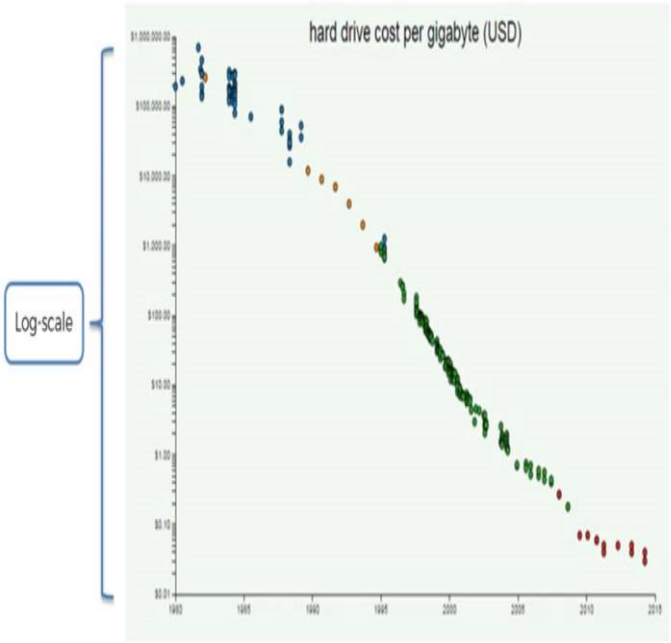


Figure 2

Working

In order to understand working of neural networks in machines, one will need to know the working mechanism in the actual brain.

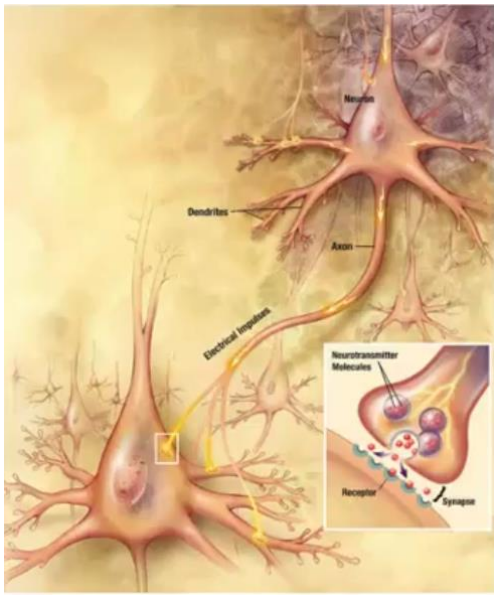


Figure 3

Figure 3 shows an illustration on neurons in the brain. A neuron has an input wire called Dendrites which receive inputs from other locations. It has an output wire called Axon. Neurons communicate with each other through pulses of electricity. A neuron sends a message in the form of an electric pulse through Axon to another neuron. The dendrites of the second neuron are connected to the axon of the first neuron. It performs some computation and sends the signal to other neurons.

The very same principle is applied by Artificial Neural Networks. Figure 4 shows an example of neural networks in which price of a house is determined based on the factors- Area, No of Bedrooms, Distance from the City and Age.

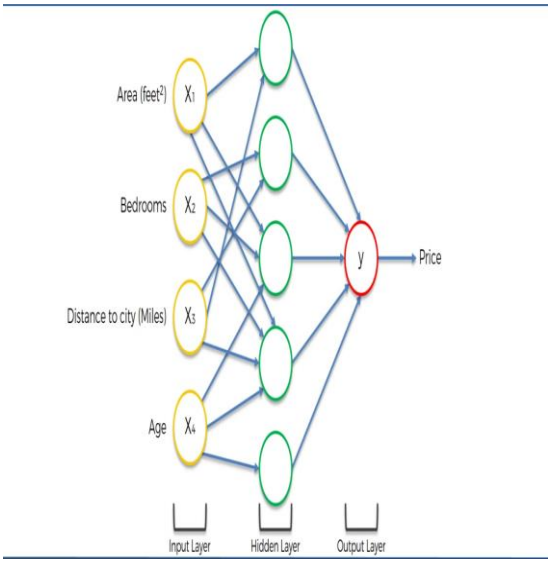


Figure 4

Generally, a neural network contains a dozen to hundreds to thousands of neurons called units, arranged in different layers. The first layer called the input layer contains input units (usually other neurons). The next layer, called the hidden layer, contains the neuron which computes a value using the inputs received. The third layer is the output layer.

Every input unit has a value, called weight, associated with it. Weights are crucial towards Artificial Neural Network. Weights are responsible to determine which signal gets passed along and which doesn't or to what extent a signal is passed along. Weights are adjusted according to the data and problem given.

The interesting part is what happens in the neuron. First step is calculation of weighted sum of input values. The next step is

applying of **Activation Function** to this sum, which is decided according to the inputs and their respective outputs. Different functions that can be used as Activation Function include Threshold Function, Sigmoid Function etc.

Learning

There are two fundamental ways to getting a computer or program do what we want to do. First is coding where we tell the program specific rules and what outcomes we want and we just guide it through the whole way

Second approach is where we create a facility for the program to understand what it has to do with the given set of inputs and outputs. Neural Networks use the second approach to train the program.

Let us represent the output value as Y' and the actual value as Y. If we want to build an efficient machine, we need minimum difference between the actual value (Y) and the output value (Y'). This is determined by **cost function.**

Cost Function is used to determine the accuracy of the program. Generally Cost function is defined as follows:-

$C = ½*(Y-Y')^2$

However, many different neural networks use different cost functions depending on the functionality of the program.

The first step is to calculate the output value for every record. This is done by applying the activation function to the weighted sum of the input values received. After this, cost function is calculated. When we take multiple records we simply add the sum of value of cost function for all inputs.

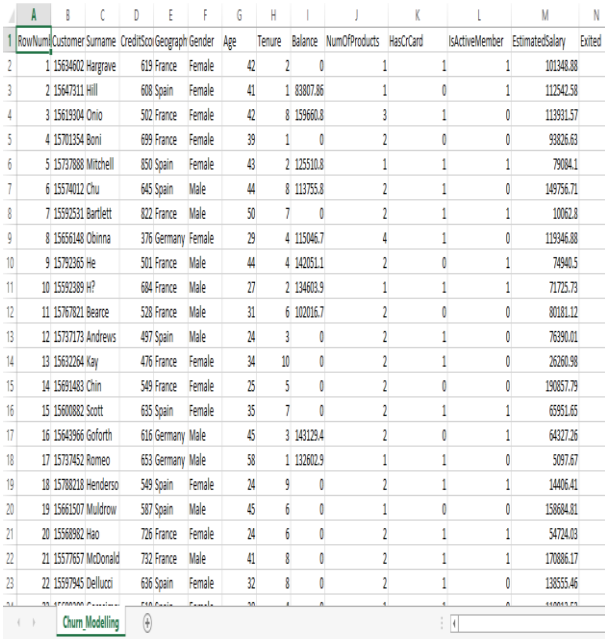
The next step is to minimize the cost function, which is done by adjusting the weights associated with each input unit. For every set of weights, cost function is calculated and the set is chosen which gives minimum cost function. This way the program learns to function. This technique is called backward error propagation.

This type of learning is called supervised learning, in which we provide the program with the set of inputs and outputs

Unsupervised learning is the one where the program is given the inputs but not the desired outputs. The program is expected to learn and identify what features it will use to group the data. Currently this concept is not completely understood. This concept can be used for robots that will learn to interact with environment as they experience new situations.

Example Problem

Let us take the Churn rate problem. We have taken a dataset of a bank (10000 customers). Every row contains CustomerID, surname, gender and other different measures like EstimatedSalary, IsActiveMember etc. One of the column represents Exited which has the value 1, if the person has left the bank and 0 if the customer hasn't. We have to make a program that can tell us which person is most likely to leave the bank using the given dataset. We use Python for this program.



RowNum	CustomerId	Surname	CreditSco	Geograph	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	
5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1	1	1	79084.1	
6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.71	
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	
8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.88	
9	15792365	He	501	France	Male	44	4	142051.1	2	0	1	74940.5	
10	15592389	H?	684	France	Male	27	2	134603.9	1	1	1	71725.73	
11	15767821	Bearce	528	France	Male	31	6	102016.7	2	0	0	80181.12	
12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	
13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	
14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	
15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	
16	15643966	Goforth	616	Germany	Male	45	3	143129.4	2	0	1	64327.26	
17	15737452	Romeo	653	Germany	Male	58	1	132602.9	1	1	0	5097.67	
18	15788218	Henderso	549	Spain	Female	24	9	0	2	1	1	14406.41	
19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81	
20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	
21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	
22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	

Churn_Modelling

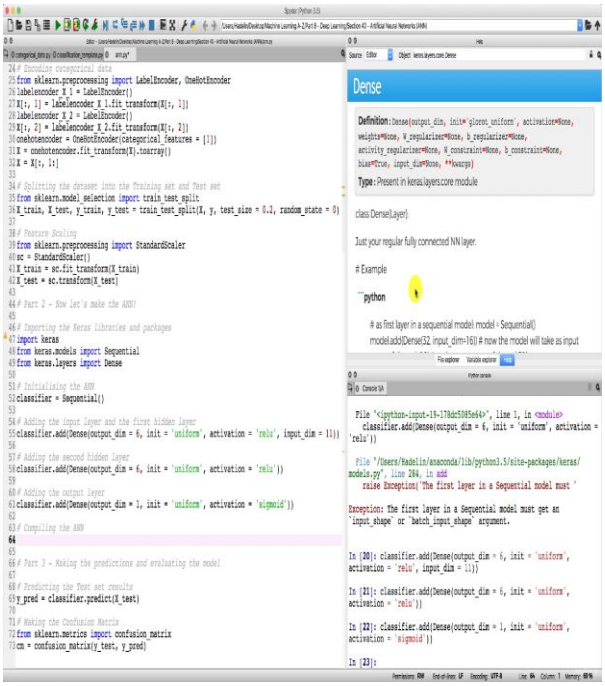
Figure 5

First we will consider the input features, which will be selected on the basis of how much they affect a person's chance of leaving the bank. Features in the dataset like Surname, RowNumber, CustomerID have no role in affecting one's decision in leaving the bank.

We include features like CreditScore, Geography, Gender, Age, Tenure, Balance, NoOfProducts, HasCrCard, IsActive and EstimatedSalary. Since we will be using numeric values for calculation, we have to encode the data stored in Geography and Gender (as they contain characters). Let us represent FRANCE with 0, GERMANY with 1 and SPAIN with 2. Similarly, FEMALE becomes 0 and MALE becomes 1. This part in Data Pre-processing.

Next step is to build the Artificial Neural Network. We have to decide no of nodes we want to add in the hidden layer. There is no rule to find an optimal number of nodes. The most basic idea is to use the average of nodes in input layer and output layer as the number of nodes. The average in this case is 6. Thus there will be 6 nodes in hidden layer. We will make this using Dense() function.

We will now randomize the initial weights using *uniform* function. We will be using Rectifier Function for the first hidden layer.



```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])
labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])
onehotencoder = OneHotEncoder(categorical_features = [1])
X = onehotencoder.fit_transform(X).toarray()
X = X[:, 1:]

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Part 2 - How let's make the ANN!

# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense

# Initialising the ANN
classifier = Sequential()

# Adding the input layer and the first hidden layer
classifier.add(Dense(output_dim = 6, init = 'uniform', activation = 'relu', input_dim = 11))

# Adding the second hidden layer
classifier.add(Dense(output_dim = 6, init = 'uniform', activation = 'relu'))

# Adding the output layer
classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))

# Compiling the ANN

# Part 3 - Making the predictions and evaluating the model

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

Figure 6

Now we will add second Hidden Layer, with 6 nodes, with the Rectifier Function as the activation function.

We will add output layer which has 1 node and Sigmoid Function as the activation function.

All these codes have been written is Python as it has default functions for Neural Network which makes it the best

programming language for Neural Networks. We can also use R programming language which is considered one of the most used languages in data science.

The next step is training of the program using the dataset.

We have two options- To update the weights after each observation, or to update the weights after a batch of observations. In this example I have taken the second option with a batch size of 10. Thus after calculating the output value for the first 10 data items, weights are adjusted and used for the next batch of data items.

As we can see in the diagram, our model has an accuracy of 86.50%. The next step is predicting the result of a new data item. We should note that the value calculated by the program is the probability whether a person will leave the bank or not and hence is between 0 and 1.

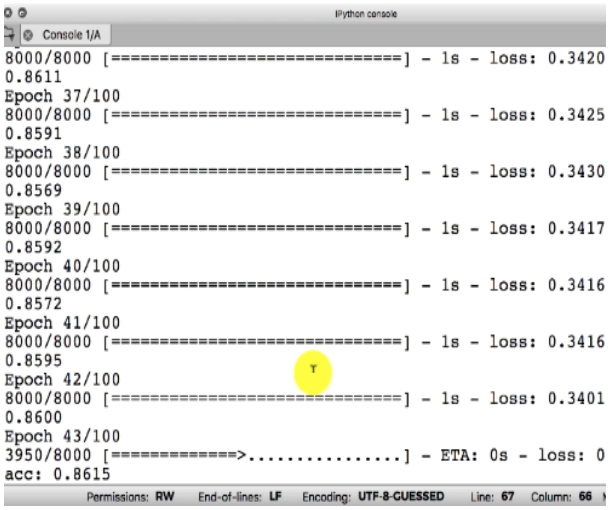


Figure 7

While predicting we have to choose a threshold value. In this case we will take 0.5 as threshold. Thus, the value predicted for output value greater than 0.5 is 1 (customer is leaving) or else 0 (customer is staying).

We predict the values for 2000 new observations and compare them to their actual values. As we can see out of 2000 observations we have 1725 correct predictions and 275 incorrect predictions which gives the accuracy of 86.25%

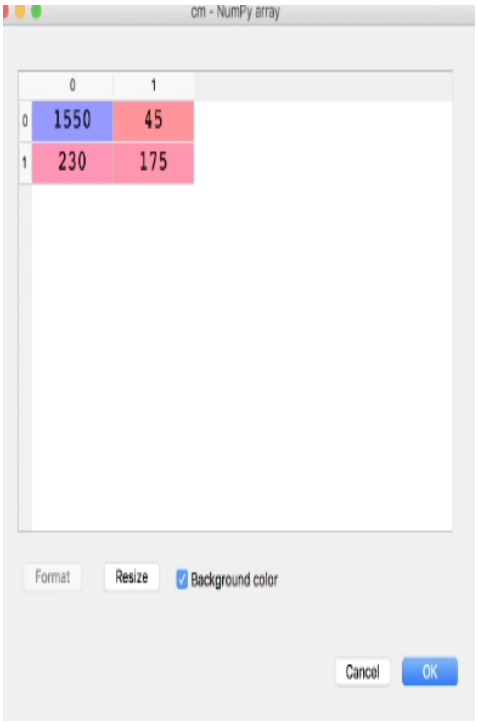


Figure 8

Applications

- Character Recognition

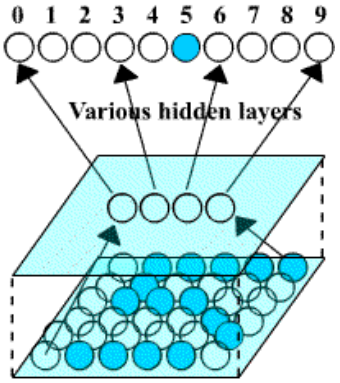


Figure 9[3]

The concept of using feedforward networks for identifying handwritten characters is rather straightforward. As in most supervised training, the input is the bitmap pattern of the handwritten character, with the desired output as the correct letter or digit. Generally such programs require the user to train the network by providing the program with their handwritten patterns.

- Stock Market Prediction

This is the most common type of application of neural networks. Businessmen often need to anticipate the value of stocks and how much will it rise to fall down, in order to take the next step for their business and to maximize profit and minimize loss.

Dean Barr and Walter Loick at LBS Capital Management have obtained decent results with their system that consisted of only 6 input nodes.

The system is fed stock market values for the past years, a decade maybe and trained to predict the future market values.

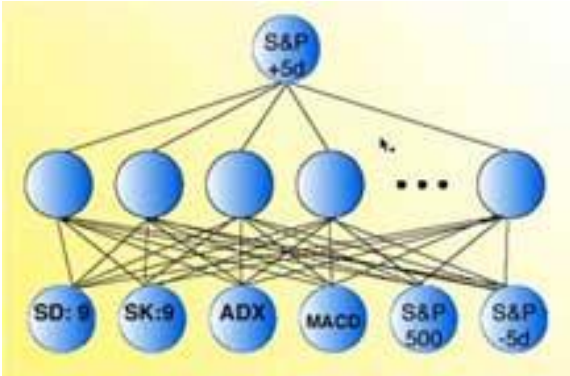


Figure 10[3]

- Medicine

One of the areas that uses machine learning is Cardiopulmonary Diagnostics. The way neural networks work in this field is by comparing many distinct models.

The data may consist of breathing rate, blood pressure, heart rate etc. to different models. The models may also include variations for age, gender, and amount of physical activity. Each person's physiological data is compared to previous physiological data and/or data of the various generic models. The deviations from the norm are compared to the known causes of deviations for each medical condition. The neural network learns by learning the different conditions and models, merging them to form a fully conceptual picture, and then diagnose a

patient's condition based on the models.

- Banks

  Every bank wants to earn as much money as possible. This is done by getting the interest from people who have taken loans from the bank. However, a bank cannot grant loan to anyone as they must ensure that the person or company is capable of paying the amount back. Neural networks comes in handy in such situation.

  A set of features like a person's Account Balance, Salary, Number of Jobs Changed, Family members etc is taken into account to estimate whether a person is eligible for loan.

  In fact, in some banks the failure rate of their traditional method is higher than that of the neural networks.

Conclusion

This paper gives an overview of the basics of Artificial Neural Network and explains the process of how it works and learns. It also talks about some applications of Neural Networks. There are many advantages of Machine Learning over conventional programs. It is able to deal with situations which were not expected while development, which a conventional program fails to do. Neural Networks is the next step for fulfilling the dream of having highly intelligent computers. However, its future is dependent in hardware development. Currently most of the neural networks development simply prove the working of the principle.

References

[1]

http://science.howstuffworks.com/brainport3.htm

[2]

https://en.wikipedia.org/wiki/Artificial_neural_network#History

[3]

https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Applications/index.html