OUR INTELLIGENCE IS WHAT MAKES US HUMAN & ARTIFICIAL INTELLIGENCE IS AN EXTENSION OF THAT QUALITY

Yann Lecun

Great Learning

POWER AHEAD

NEURAL NETWORKS

INSIDE THIS HANDBOOK

This handbook comprises of all important details required for a learner to revise the concepts which were covered in depth as a part of the course offered by GreatLearning. This handbook should not be treated as a complete reference material for the topic.

ALGORITHM WHITEBOX

TOPIC OVERVIEW

Whitebox stands for keeping the algorithm transparent in terms of how it works on the backend to learn the patterns from the given data points and predict futuristic outputs. Each algorithm is a combination of well-programmed equations.

BRAIN TEASER

Use your linguistics skills to read the paragraph that is in French language. Try predicting a word that best fits the blank space

Nous menons actuellement un atelier sur la compréhension de l'AIML. Le virus Corona se propage rapidement en France. Il a été trouvé pour la première fois en China dans le district de _____.

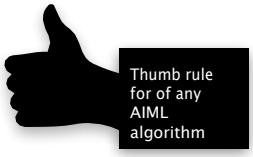
SOLUTION

The most probable answer should be "WUHAN". The strategy to predict is: you [brain] hearing, reading and discussing about COVID-19 since Dec 2019 till date. You have correlated certain words together i.e. when a discussion about COVID happens you would have always heard words like CORONA, VIRUS, CHINA and when the word DISTRICT is mentioned along with these words above the most probable answer should be WUHAN.

THINK

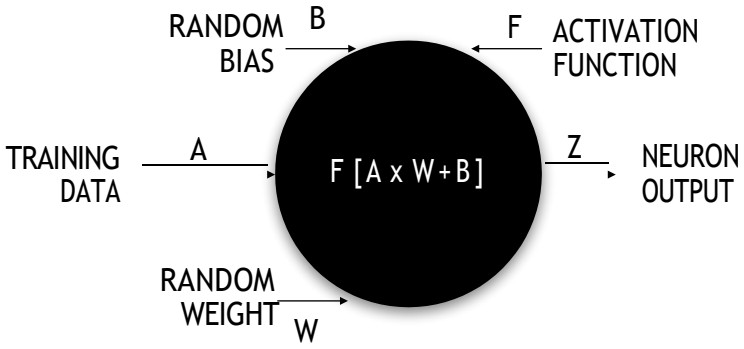
The thing to learn from the above example is that all predictions happen with experiences within similar situations where we learn the patterns that if certain combinations occur together it is a high probability for a particular even to be happening in future. Correlate the same as the backend of any AIML algorithm. These algorithms just like your brain are designed to identify PATTERNS from the historical events just as you identified the PATTERNS from the above brainteaser example.

Each algorithm has a different approach for PATTERN identification but the end goal is same to predict for future with best probability.



ARTIFICIAL NEURON

Inspired from theological neural networks of human brain. They conduct electric signals against various activities experienced by human sensor organs. Collection of connected units i.e. NEURONS which conducts/block signals depending the activity sensed by the human body.



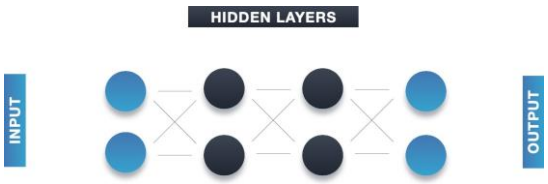
NEURON PARAMETERS

- TRAINING DATA: Input data. Input neurons = data features
- RANDOM WEIGHTS: randomly chosen number
- RANDOM BIAS: randomly chosen number
- ACTIVATION FUNCTION: mathematical equation
- NEURON OUTPUT: Z = F [A x W + B]

NEURAL NETWORKS

Collection of connected units i.e. NEURONS which conduct/block signals depending the activity sensed by the human body. Artificial Neural network layers are of 3 types:

- Input: Map predictors
- Hidden: Pattern learning
- Output: Regressor or classifier



BRING IN THE MATH

Let us use the neuron described above into simulating a neural network for a sample dataset. In forward propagation the input data is operated upon by weights, bias and activation function to produce an output. The produced output or "Predicted output" will be compared with "Actual output" to find the "Loss". Loss ideally should be 0 for a well-trained neural networks. Once loss is detected the weights will be adjusted [using corrective step functions] aiming for loss being 0 in the next forward propagation.

Preditor	Target
2	6.15
3	9.45
5	16.75
6	19.75

Above dataset consists of 2 columns. Column "Predictor" vs " Target" can be used to learn patterns.

Input	Weights	Bias	Function	Z	Actual	Loss
2	1.5	-1	Absolute	2	6	4.15
3	1.5	-1	Absolute	3.5	9	5.95
5	1.5	-1	Absolute	6.5	15	10.25
6	1.5	-1	Absolute	8	18	11.75

We have setup a neural network that will have:
- The"Predictor" columns as Input.
- Weights: Random weights
- Bias: Randomly chosen fixed bias
- Z = F [A x W + B] . Predicted
- Actual: Target column from the dataset
- Loss: Difference between Predicted vs actual value

Input	Weights	Bias	Function	Z	Actual	Loss
2	2	-1	Absolute	3	6	3.15
3	2	-1	Absolute	5	9	4.45
5	2	-1	Absolute	9	15	7.75
6	2	-1	Absolute	11	18	8.75

We have setup a neural network that will have:
- Weights: Change the value of weights
- Loss: The loss has reduced

Input	Weights	Bias	Function	Z	Actual	Loss
2	3.5	-1	Absolute	6	6	0.15
3	3.5	-1	Absolute	9.5	9	0.05
5	3.5	-1	Absolute	16.5	15	0.25
6	3.5	-1	Absolute	20	18	0.25

We have setup a neural network that will have:
- Weights: Change the value of weights
- Loss: The loss has reduced and has almost near to 0

Weights "W" are the learnings out of neural networks training.

*Bias is also updated in back propagation. For keeping it simple II have kept it constant in the above example.

ALGORITHM SIMULATION

TOPIC OVERVIEW

MS excel is used to understand how a simple artificial neural network performs forward and backward propagation on a simple numeric dataset. Please use the simulation MS excel file for actual hands on.

INPUT DATA	
Alphabets	Number
A	0.0000
B	0.0385
C	0.0770
D	0.1155
E	0.1540
F	0.1925
G	0.2310
H	0.2695
I	0.3080
J	0.3465
K	0.3851
L	0.4236
M	0.4621
N	0.5006
O	0.5391
P	0.5776
Q	0.6161
R	0.6546
S	0.6931
T	0.7316
U	0.7702
V	0.8087
W	0.8472
X	0.8857
Y	0.9242
Z	0.9627

INPUT DATA

Alphabets from A to Z are equally divided between 0 to 1. For example, the alphabet A is equal to 0 and alphabet Z in equal to 0.96. You may considerthe alphabets as different features in a dataset with the numbers corresponding to it as a single record of the dataset. The column "Number" is used as an input for a simple neural.

NEURAL NETWORKS

The designed network consists of:

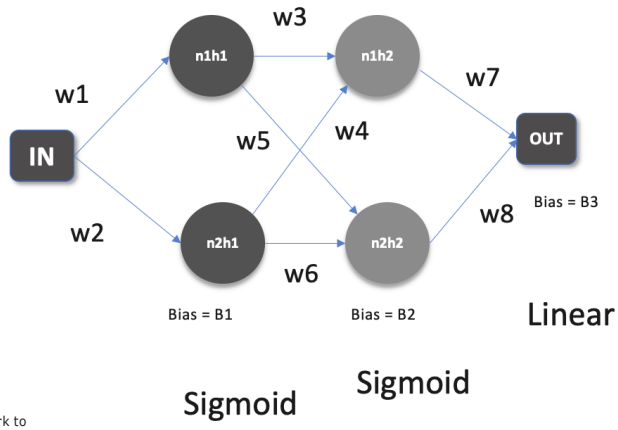
- Input layer: 1 in number. Input data flows in here one at a time.
- Hidden layer: 2 in number
  - Hidden layer 1: Weights [WI, W2], Bias [B1], Activation function: Sigmoid
  - Hidden layer 2: Weights [W3, W4,W5,W6], Bias [B2], Activation function: Sigmoid
- Input layer: 1 in number. Regressor output. Weights [W7, W8]. Activation function: Linear

TRAINING NEURAL NETWORK

We have randomly chosen the weights and bias for the given network.

- FORWARD PROPAGATION: We will use the input to propagate through the network to produce Output. Output is compared with actual Number from the dataset to get the value for Loss. Here we have calculated Loss using RMSE.
- BACKWARD PROPAGATION: We will use the SOLVER function from MS Excel to get the best combination of Weights and Bias to get the least value of RMSE.

NEURAL NETWORK



TRAINING

Input		Hidden layer 1		Hidden layer 2		Output	RMSE
Value	Actual	n1h1	n2h1	n1h2	n2h2		
W	0.84717	0.00063	0.97678	0.00026	0.11992	0.85056	0.07160
H	0.26952	0.00059	0.01978	0.00024	0.00028	0.00196	
Y	0.92419	0.00064	0.99149	0.00026	0.13036	0.92462	

TESTING

Input		Hidden layer 1		Hidden layer 2		Output	Error
Value	Actual	n1h1	n2h2	n1h2	n2h2		
S	0.69313	0.00062	0.52128	0.00025	0.00708	0.05019	0.643

WEIGHTS & BIAS

W1	W2	B1
0.12	13.23	7.47

W3	W4	B2
0.60	0.08	8.33

W5	W6
20.18	6.48

W7	W8	B3
0.00	7.09	0.00

WEIGHTS ARE ADJUSTED WHILE BACK PROPAGATION

ALGORITHM INGREDIENTS

TOPIC OVERVIEW

There are various parameter that define the optimal performance and training of a neural network. Understanding the functionality of each parameter and adjusting them appropriately will help achieve optimal neural networks performance.

WEIGHTS & BIAS

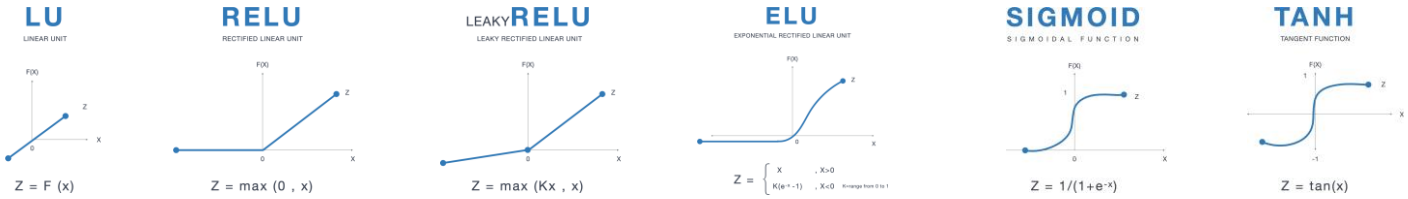
We can assign random weights to each layer of a neural network in many ways. Please usethe link given on the right to refer to the list of weights and bias initializers available in keras.

https://keras.io/api/layers/initializers/

ACTIVATION FUNCTIONS

Activations functions bring in non-linearity within the neural networks. Activations functions are used to regulate the functioning of a neuron i.e. if the neuron is activated or deactivated. There are many activation functions available for a neural network. Please use the link given on the right to refer to the list of activation functions available in keras.

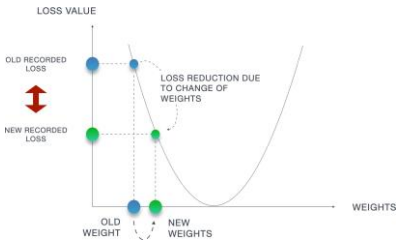
https://keras.io/api/layers/activations/



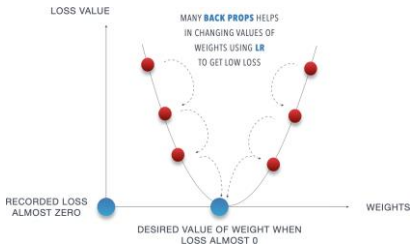
GRADIENT DESCENT

Gradient descent refers to technique used to find the best weight for a neuron that produces a loss closeto 0 i.e. the predicted value from neural network output and the actual value from the learning dataset are very close to each other making sure the best learning is obtained out of the neural networks.

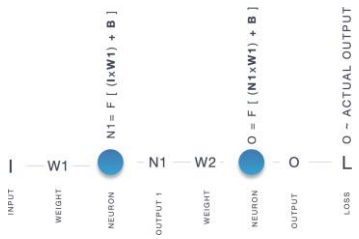
https://keras.io/api/optimizers/



Refer the above plot for LOSS vs WEIGHTS. The above plot displays the gradient descent curve that can be used to observe the rate of change of loss when the weights are increased or decreased.



Refer the above plot for LOSS vs WEIGHTS. The above plot displays the gradient descent curve that can be used to understand that when LEARNING RATE CONCEPT is used to change the value of weights. In this exercise, there is one location where LOSS becomes 0 at a particular weight value.



Refer the above simple artificial neural network. Please pay attention on the network attributes used. We will use the same below to build the chain rule derivative used for adjusting the weight for LOSS —> 0 while performing the back propagation.

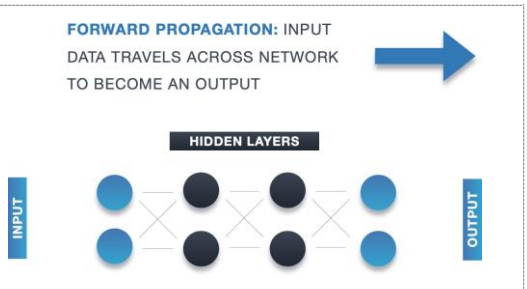
$$\frac{\partial N1}{\partial W1} \times \frac{\partial W2}{\partial N1} \times \frac{\partial O}{\partial W2} \times \frac{\partial L}{\partial O} \longrightarrow \frac{\partial L}{\partial W1} = \frac{\partial N1}{\partial W1} \times \frac{\partial W2}{\partial N1} \times \frac{\partial O}{\partial W2} \times \frac{\partial L}{\partial O}$$

DERIVATIVE CHAIN RULE

Rate of change of LOSS with respect to change in WEIGHT 1

$$\frac{\partial L}{\partial W2} = \frac{\partial O}{\partial W2} \times \frac{\partial L}{\partial O}$$

Rate of change of LOSS with respect to change in WEIGHT 2



SECRET SAUCE

Neural networks consists of 2 propagation techniques.

- FORWARD PROPAGATION
- BACKWARD PROPAGATION

KEEP

LEARNING !

Great Learning

POWER AHEAD

NEURAL NETWORKS