ACTIVATION FUNCTIONS AND TRAINING ALGORITHMS FOR DEEP NEURAL NETWORK

Gayatri D. Khanvilkar

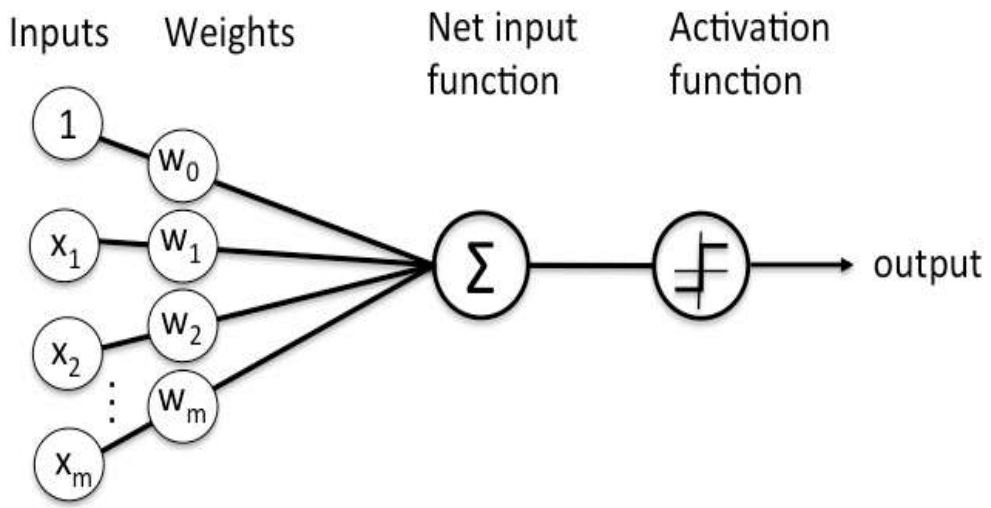
Email: khanvilkarg7@gmail.com

Content

- Neural Network
- Deep Neural Networks
- Applications
- Activation Functions
- Sigmoid or Logistic
- Tanh—Hyperbolic Tangent
- ReLu -Rectified Linear Units
- Vanishing Gradients
- Comparative study of Activation Functions.
- Training Algorithm
- Greedy Algorithm
- Dropout Algorithm
- Comparative study of Algorithms
- Summary
- References

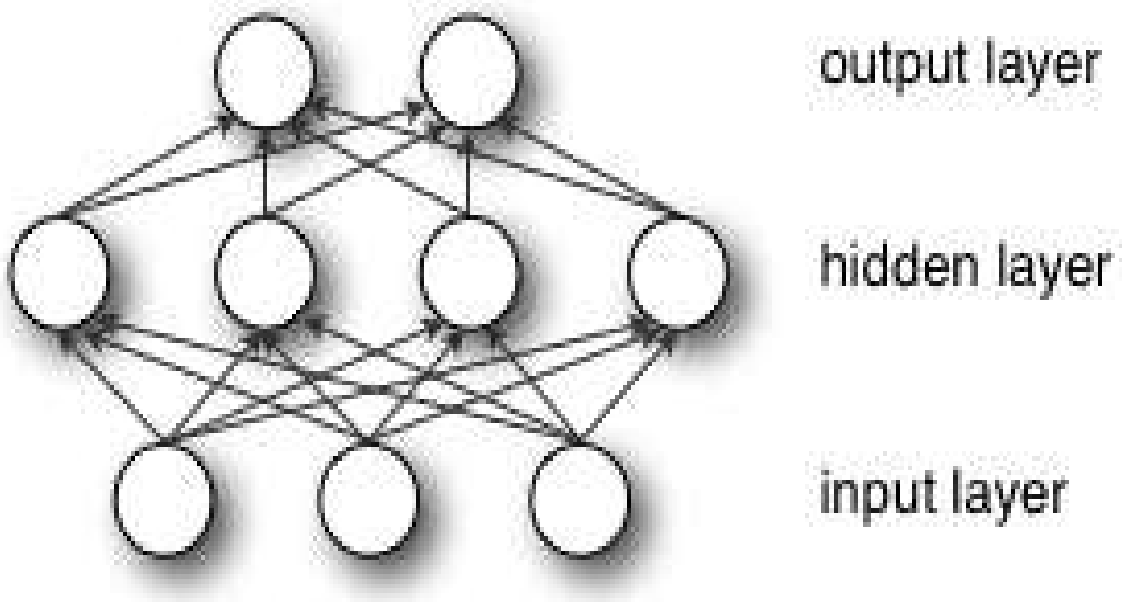
Neural Network

- A computer system modelled on the human brain and nervous system.
- Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'.
- Neurones works together to solve specific problems.



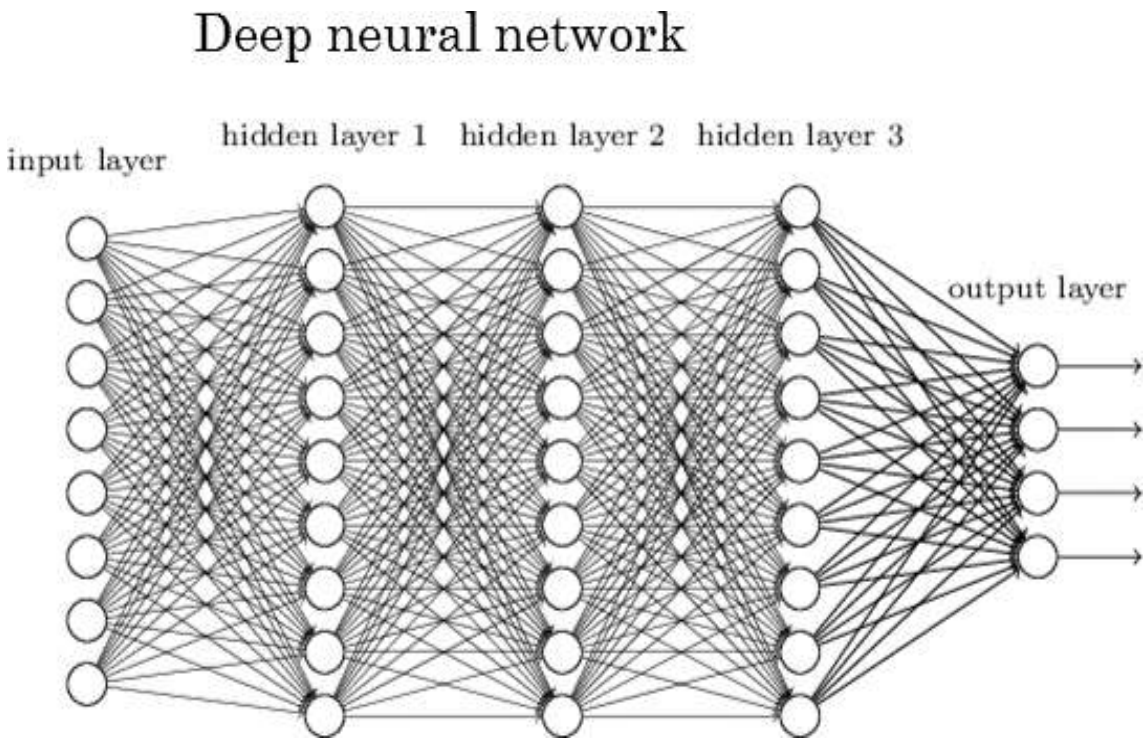
Neural Network (Cont.)

- Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.



Deep Neural Networks

- Deep-learning networks are different from the standard single-hidden-layer neural networks by their depth
- More than three layers (including input and output) qualifies as “deep” learning. So deep is a strictly defined, technical term that means more than one hidden layer.

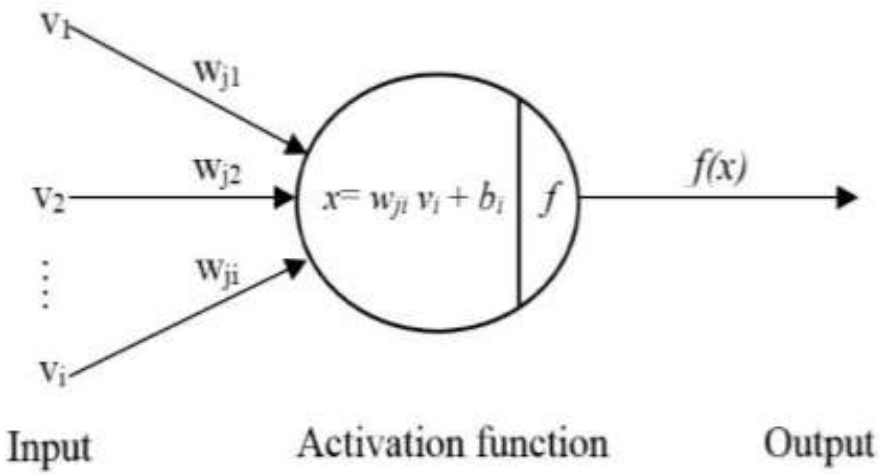


Applications

- Automatic speech recognition
- Image recognition
- Visual Art Processing
- Natural language processing
- Recommendation systems
- Bioinformatics
- Mobile Advertising

Activation Functions

- Their main purpose is to convert a input signal of a node in a A-NN to an output signal.
- Activation Function calculates a "weighted sum" of its input, adds a bias and then decides whether it should be "fired" or not. So, consider a neuron.
  - **X = ∑ (WEIGHT * INPUT) + BIAS**



Role and Importance of Activation Function.

- If we do not apply a Activation function then the output signal would simply be a simple **linear function**.
- Also without activation function our Neural network would not be able to learn and model other complicated kinds of data such as images, videos, audio, speech etc.
- From a non linear Activation we are able to generate non-linear mappings from inputs to outputs

Activation Functions(Cont.)

Most popular types of Activation functions -

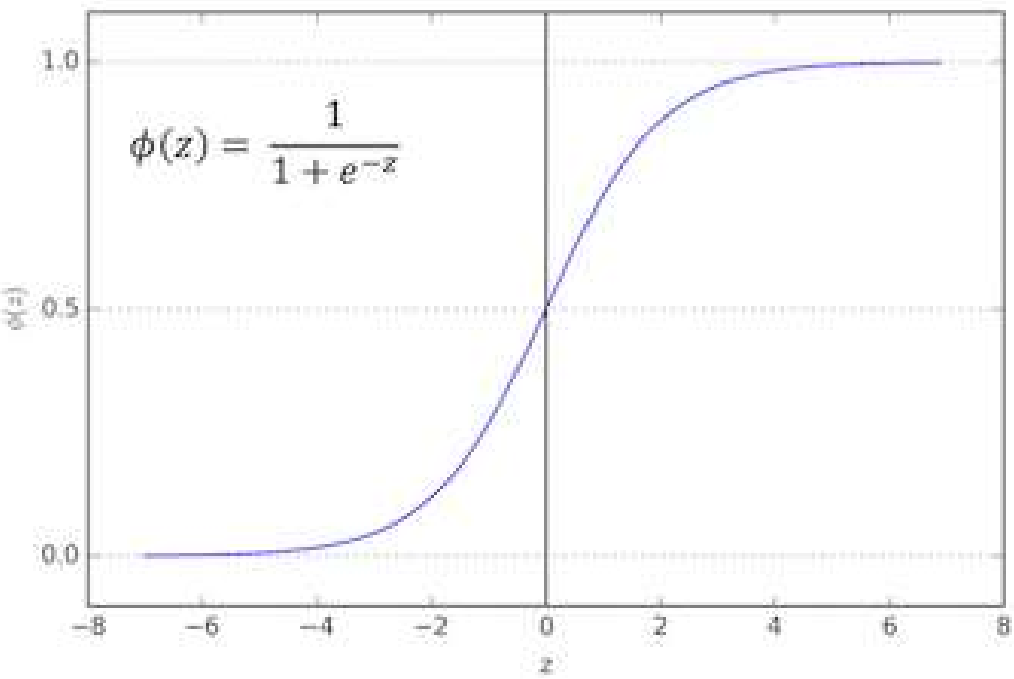
- Sigmoid(Logistic)
- Tanh—Hyperbolic tangent
- ReLu -Rectified linear units

Sigmoid or Logistic

- Mathematical Form:

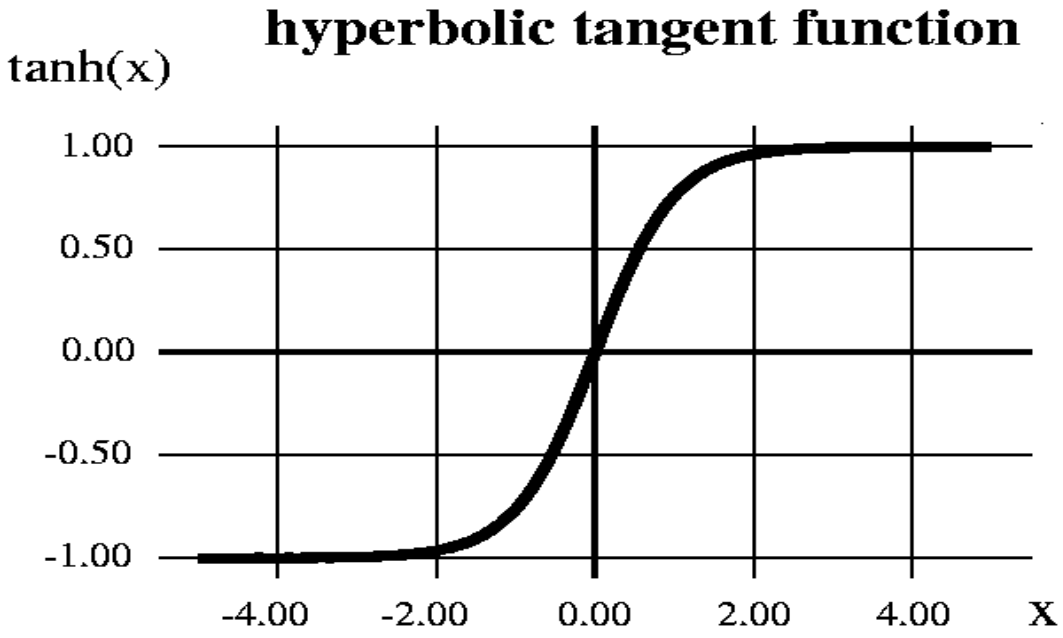
$$A = \frac{1}{1+e^{-x}}$$

- Advantage
  - Good for classifier.
  - It is easy to understand and apply
- Disadvantage
  - Vanishing gradients problem



Tanh—Hyperbolic Tangent

- Mathematical formula
- $\mathbf{f(x) = Tanh(x) = \frac{2}{1+e^{-2x}} - 1}$
- Gradient is stronger for tanh than sigmoid. Deciding between the sigmoid or tanh will depend on your requirement of gradient strength.
- Advantages
  - Easy to understand
  - Gradient is stronger than sigmoid
  - Give more optimized solution as compare to sigmoid
- Disadvantage
  - Vanishing gradient problem.



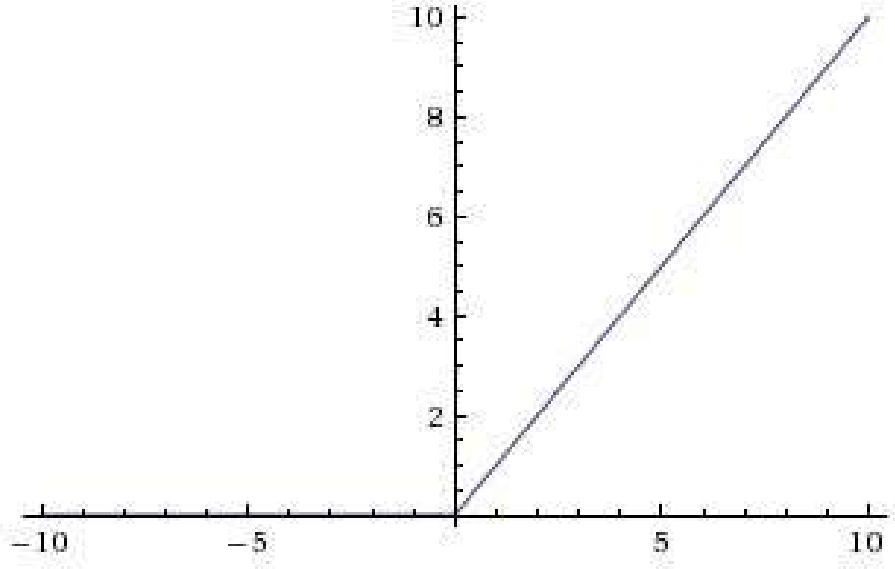
ReLu -Rectified Linear Units

R(x) = max (0, x) i.e.

if x < 0, R(x) = 0 and if

x >= 0, R(x) = x.

- Gives an output x if x is positive and 0 otherwise.
- Advantages
  - Making the activations sparse and efficient.
  - No problem of vanishing gradient
- Disadvantages
  - The gradient can go towards 0.Neurons which go into that state will stop responding to variations in error/ input. It is known as problem of dying neurons.
- To fix this problem, modification was introduced called **Leaky ReLu.**



Vanishing Gradients

- Generally, adding more hidden layers tends to make the network able to learn more complex functions, and thus do a better job in predicting future outcomes.
- While backpropagation, the ‘gradient values’, which dictate how much each neuron should change(i.e. weight) become smaller and smaller.
- This means that the neurons in the Earlier layers learn very slowly as compared to the neurons in the later layers.
- Because of this, the training process takes too long and the Prediction, Accuracy of the Model will decrease.

Comparative Study of Activation Function

Activation Functions/ Parameters	Sigmoid	Tanh	ReLu
Nature	Non-Linear	Non-Linear	Non-Linear
Mathematical Representation	$A=\frac{1}{1+e^{-x}}$	$Tanh(x)=\frac{2}{1+e^{-2x}}-1$	R(x) = max (0, x)
Complexity	Easy to understand and apply	Easy to understand and apply	Simple and efficient
Output Range	0 to 2	-1 to 1	O to ∞
Optimization	This function isn't zero centred so optimizations is harder	This function is zero centred so optimizations is easier.	Give optimized solution.

Activation Functions/ Parameters	Sigmoid	Tanh	ReLu
Computation Time	Requires more computation time	Requires more computation time than ReLu	Requires less computation time than sigmoid and Tanh
Sparsity Property	No	No	Yes
Problem of Dead Neuron	No	No	Yes Can be solved by LeakyReLu
Vanishing Gradient Problem	Yes	Yes	No
Layers	Can apply on any layer of NN	Can apply on any layer of NN	Can apply on only Hidden layer of NN
Classification and regression	Good for classification	Good in Classification	For classification need to use softmax and for regression need to use linear function.

Training Algorithm

- It is step-by-step procedure for adjusting the connection weights of a neural network.
- There are two types of training algorithms.
- In supervised training algorithm desired output is present to the neural network. Desired output and current output of same input vectors get compared and accordingly weights get adjusted.
- In unsupervised training algorithm weights get adjusted without knowing desired output.

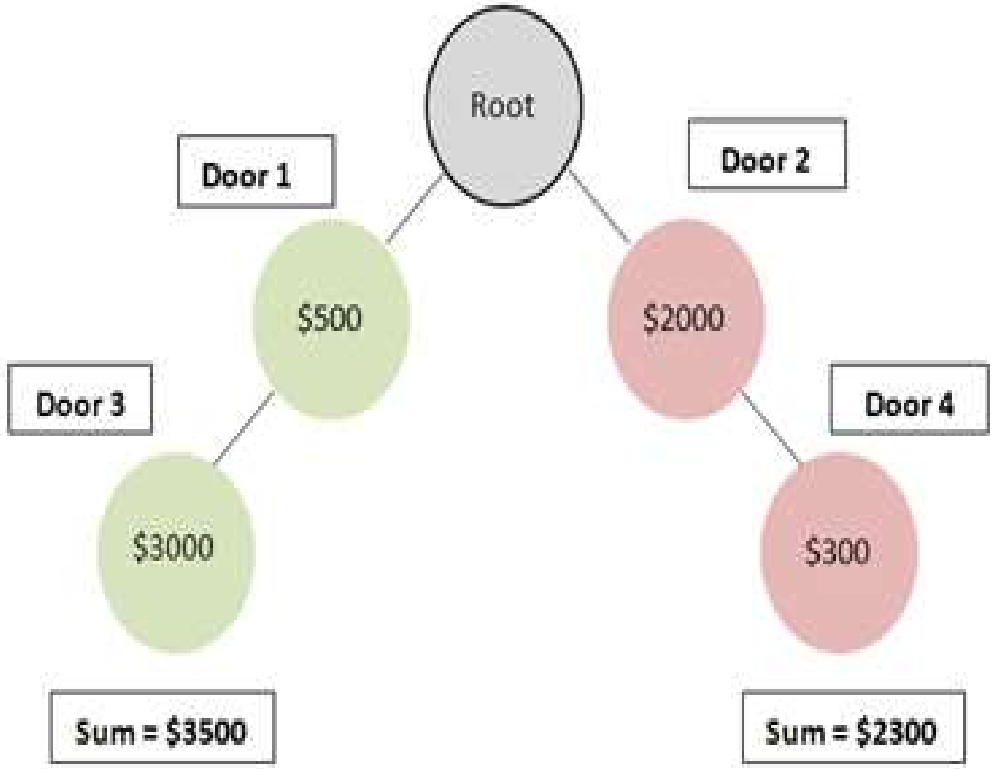
Greedy Algorithm

- Solve the problem by the method of step by step
- The original problem is simplified to a similar sub problem with smaller size after being greedy
- Solving the problem of every step to make some decisions
- Always makes the choice that seems to be the best at that moment.
- Relatively simple, the effective method to solve problem fast, but the solution is not necessarily the overall optimal solution.
- Greedy algorithms mostly (but not always) fail to find the globally optimal solution
- Most problem for which greedy algorithm work well, will have 2 properties:
  - Greedy choice property
  - Optimal substructure

A failure of the greedy algorithm

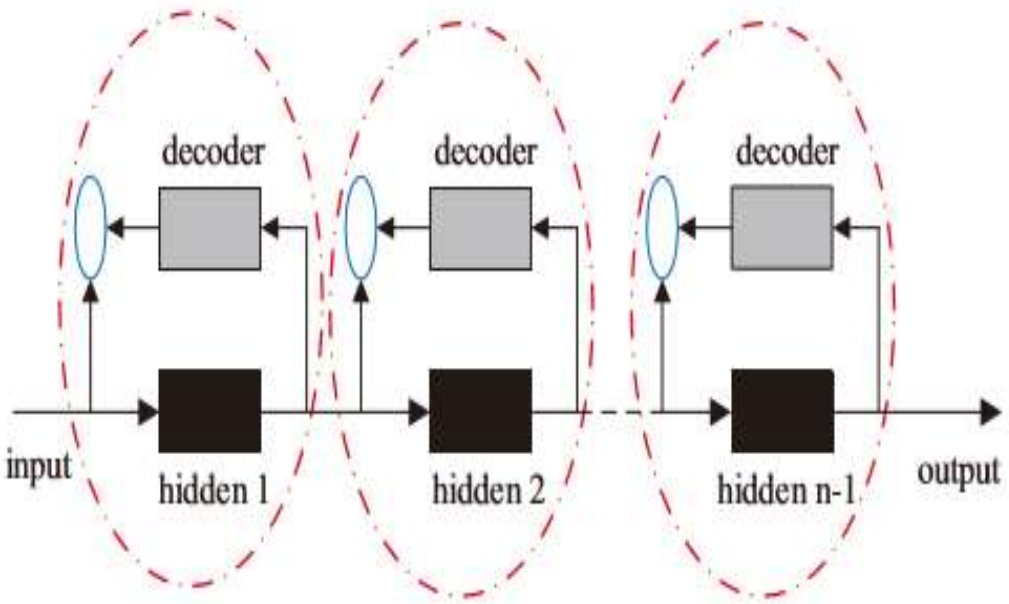
Example:

- In this we need to collect maximum money with the help of greedy algorithm.
- Choosing Door 1 and Door 3 is the best route to maximize gain. But when there are only two options Door 1 and Door 2 and we don't know the next Door value, the greedy algorithm will choose Door 2.
- This is the short-sighted nature of this class of algorithms and may not always lead to maximum profit.



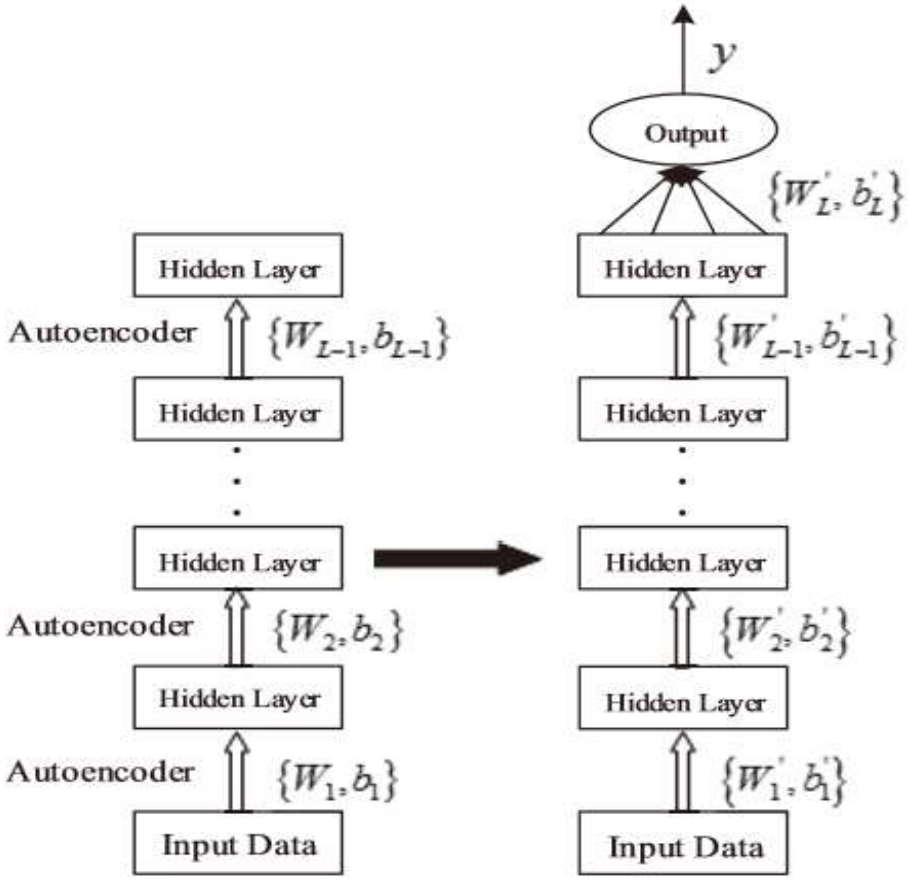
Layer-wise training of deep networks

- First an autoencoder neural network is consisting of input layer and the first hidden layer, after this autoencoder is trained the first hidden layer and the second hidden layer will form another autoencoder neural network, until the training of the last hidden layer is completed.
- The outputs of each hidden layer are given as inputs to the next hidden layer, and the encoding step for the greedy layer-wise training is given by running the autoencoder training of each hidden layer. At last the weights of each hidden layer will be used for initializing the weights of the deep network.



Training process of deep neural network

- Trained weights $\{W_l, b_l\}$ will be used for initializing the weights of each hidden layer
- $\{W'_L, b'_L\}$ is the weight of output layer



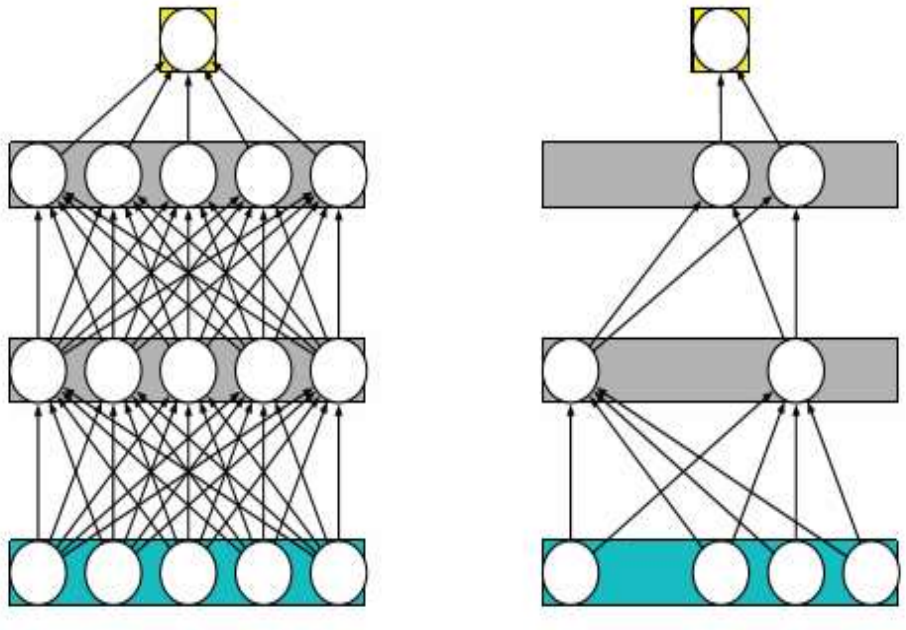
Greedy Algorithm (Cont.)

- **Advantages**
  - Simple and easy to understand
  - It can train neural network in supervised and unsupervised manner.
  - It is an effective method to solve the problems fast

- **Disadvantages**
  - No guarantee of optimized solution.
  - Can give optimised solution to some problems but can’t for all problems.

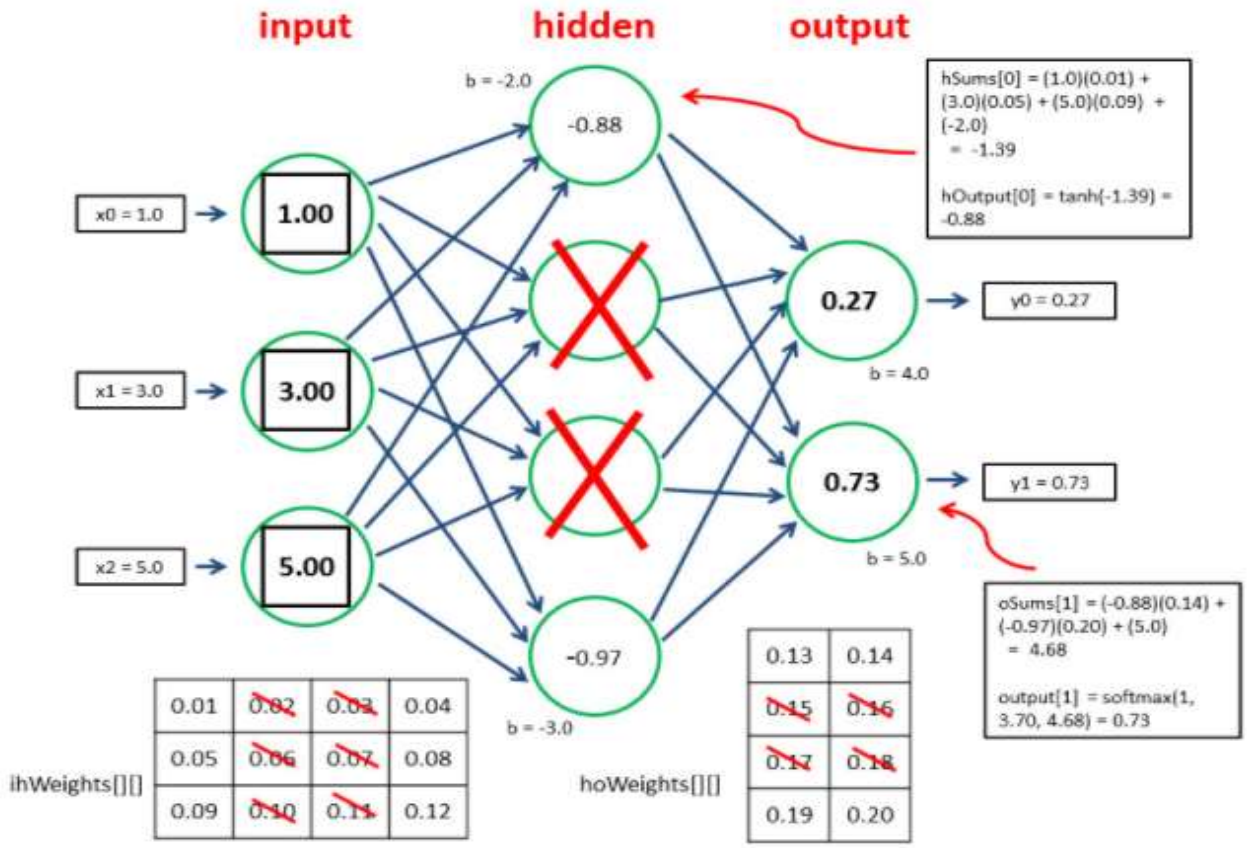
Dropout Algorithm

- The dropout method is an alternative and more efficient option for addressing DNN overfitting.
- In a DNN developed with the dropout method, hidden units as well as their incoming and outgoing connections are temporarily excluded in the network during the training
- Hidden units to be excluded are randomly selected with a probability p.



Working of Dropout algorithm for feedforward network

- The fig. represents a neural network that has three inputs, four hidden nodes and two outputs.



Dropout Algorithm (Cont.)

- **Advantages**
  - Simple and easy to understand
  - Solves overfitting problem of neural network
  - Improves performance
  - Dropout algorithm is more effective than classical data-driven algorithms.
  - Make neural network more robust.
  - It can train neural network in supervised and unsupervised manner.

- **Disadvantages**
  - Sparse and noisy input to dropout simply reduce the amount of data available for learning.

Comparison of Algorithms

Algorithm / parameters	Greedy Algorithm	Dropout Algorithm
Key concept	Divide large problem to subproblem.	Generate different virtual subnets of original neural network.
Solution	No guarantee of optimized solution.	Can give optimized solution.
Optimization	Optimization depends on nature of problem.	No such problem.
Drop nodes	Does not dropout nodes	It dropout nodes.
Advantages	Faster than other algorithms in optimization	Solves problem of overfitting and improve performance
Applications	Task Scheduling, Network routing, Graph colouring, etc.	Image Classification, Automatic prediction of heart rejection, etc.

Summary

- Artificial Neural Network train with the help of Activation Function and Training Algorithms. ReLu is most widely used activation function but it Faces problem of dead neuron.
- Selection of activation function is depends on type of problem. eg. sigmoid can solve this classification problem more efficiently and in faster way than ReLU
- Selection of algorithms also need to check type of problem, if problem can be divided into similar small problem then can use Greedy Algorithm. If there is problem of overfitting then Dropout Algorithm is best choice.

References

1. Wikipedia contributors. "Machine learning." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 24 Oct. 2017. Web. 29 Oct. 2017
2. “NEURAL NETWORKS by Christos Stergiou and Dimitrios Siganos ”, https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html, September 2017.
3. Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org
4. Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
5. Wikipedia contributors. "Deep learning." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 23 Oct. 2017. Web.
6. “opening up deep learning for everyone”, http://www.jtoy.net/2016/02/14/opening-up-deep-learning-for-everyone.html, October 2017.
7. Lau, Mian Mian, and King Hann Lim. "Investigation of activation functions in deep belief network." Control and Robotics Engineering (ICCRE), 2017 2nd International Conference on. IEEE, 2017.
8. “Understanding Activation Functions in Neural Networks”, https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0, September 2017.

References(Cont.)

9. “Activation functions and it’s types-Which is better?”, https://medium.com/towards-data-science/activation-functions-and-its-types-which-is-better-a9a5310cc8f, September 2017.
10. Qian, Sheng, et al. "Adaptive activation functions in convolutional neural networks." Neurocomputing (2017).
11. “What is a piecewise linear function?”, https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.2/ilog.odms.cplex.help/CPLEX/UsrMan/topics/discr_optim/pwl/02_pwl_defn.html, October 2017.
12. “ReLU and Softmax Activation Functions”, https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions, October 2017.
13. “The Vanishing Gradient Problem”, https://medium.com/@anishsingh20/the-vanishing-gradient-problem-48ae7f501257, September 2017.
14. “What is the vanishing gradient problem?”,https://www.quora.com/What-is-the-vanishing-gradient-problem, September 2017.
15. “Neural Network Dropout Training”, https://visualstudiomagazine.com/articles/2014/05/01/neural-network-dropout-training.aspx , September 2017.

References(Cont.)

16. Liu, Jun, Chuan-Cheng Zhao, and Zhi-Guo Ren. "The Application of Greedy Algorithm in Real Life." DEStech Transactions on Engineering and Technology Research mcee (2016).
17. “Intro to Greedy Algorithms feat. Uncle Scrooge”, https://medium.com/the-graph/uncle-scrooge-meets-greedy-algorithms-dfc80c33d7ac, October 2017.
18. Wang, Jian-Guo, et al. "A mothed of improving identification accuracy via deep learning algorithm under condition of deficient labeled data." Control Conference (CCC), 2017 36th Chinese. IEEE, 2017.
19. Wikipedia contributors. "Greedy algorithm." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 19 Apr. 2017. Web. 28 Oct. 2017.
20. Tong, Li, et al. "Predicting heart rejection using histopathological whole-slide imaging and deep neural network with dropout." Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on. IEEE, 2017.
21. Wang, Long, et al. "Wind turbine gearbox failure identification with deep neural networks." IEEE Transactions on Industrial Informatics 13.3 (2017): 1360-1368.
22. Ko, ByungSoo, et al. "Controlled dropout: A different approach to using dropout on deep neural network." Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on. IEEE, 2017.
23. “Regularizing neural networks with dropout and with DropConnect”, http://fastml.com/regularizing-neural-networks-with-dropout-and-with-dropconnect/ , October 2017.

Thank You