Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

An Introduction to Deep Learning – A Empirical Study

Helaria Maria x

Assistant Professor, New Horizon College, Bangalore, India

helariax@gmail.com

Article Info
Page Number: 10584-10590
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 September 2022
Revised: 30 October 2022
Accepted: 15 November 2022
Publication: 19 December 2022

Abstract:
Deep learning (DL), a subset of machine learning (ML) and artificial intelligence (AI) it is considered as a core technology in today's Industrial Revolution. Due to its learning capabilities from data, DL technology made from artificial neural network (ANN), it has become a one of the topics in the field of context of computing and is widely applied in various application areas like healthcare, visual recognition, text analytics, cybersecurity, and many more. In this pager we will be learning about the fundamental building blocks of deep learning which uses perceptron, stacking the perceptron together to solve more complex models by using mathematical optimize models using backpropagation and gradient descent. We will be learning practical challenges of training these models in the real life by applying best practices like adaptive learning, batching and regularization to struggle with overfitting.

Keywords: Deep learning (DL), Machine learning(ML), neural network architectures, backpropagation, gradient descent.

Introduction to Deep Learning:

Deep learning is a subset domain of machine learning. It has faster computational power and large data sets, deep learning algorithms can self-learn hidden patterns within data to make predictions. In this we can think of a branch of machine learning that's trained on huge amount of data and deals with a lot of computational units working in cyclic to perform predictions[1].

Deep Learning and Human Brain, we create systems that learn how humans learn, the architecture for deep learning is inspired by the structure of a human brain. A few fundamental expressions within deep learning can be mapped back to neurology. Neurons form the essential building blocks of the brain; deep learning architecture contains a computational unit that allows modelling of nonlinear functions called ***perceptron*.**

Aim of perceptron is to understand data representation by stacking together in many layers, where each layer should understand the input that is processed. A layer is a collection of computational units that learn to detects the value and repeats the occurrence of values.

Each layer of perceptron should interpret a specific pattern within the data. A network of the perceptron is how neurons in the brain form a network, so the architecture is called neural networks.

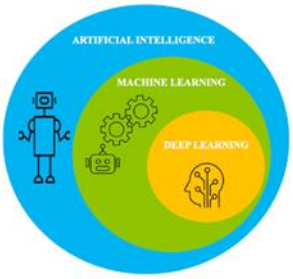


Fig 1: Architecture DL

The Basic fundamental block of Deep Learning is **Perceptron** which is a single neuron in a Neural Network[3]. Given a finite set of m inputs x1,x2,x3,….xm, we multiply each input by a weight Θ0, Θ1, Θ2… Θm then we sum ∑ up the weighted combination of inputs, add a bias ∫ and finally pass them through a non-linear activation function that process to the output Y.

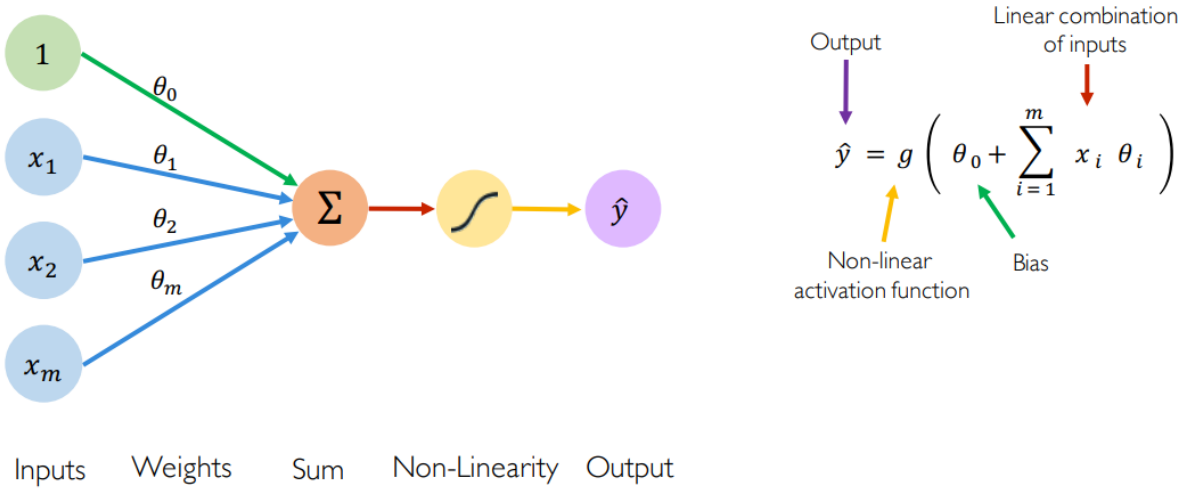


Fig 2: A single neuron in a Neural Network

Step 1 - Calculate weighted sum

- Inputs x1 through xm, which can also be denoted by a vector X. Xi represents the ith entry from the data set. Each entry from the data set contains n dependent variables.
- Weights Θ 1 through Θ m, which can be denoted as a matrix W
- A bias term b, which is a constant[5]

Step 2 - Activation function

The output from step 1 is passed through an **activation function**. The activation function g is a mathematical function, that transform the output to a non-linear before it is sent to next layer. It maps the calculation result to a range. This helps in identifying the neuron that need to be fired. For example, a sigmoid function maps values to the range [0,1], which is useful if you want your system to predict probabilities.

hL = ˆy = f(x)

Data: {xi, yi} N
i=1

Model:

yˆi = f(xi) = O(W3g(W2g(W1x + b1) + b2) + b3)

Parameters:
θ = W1, .., WL, b1, b2, ..., bL(L = 3)

Algorithm: Gradient Descent with Back-propagation

Objective/Loss/Error function: Say,
Min 1/N ∑ ∑ (ˆyij − yij)2
i=1 j=1
In general, min L (θ)

L (θ) is some function of the parameters
where L (θ) is some function of the parameters

Algorithm: gradient descent ()

t ← 0;
max iterations ← 1000;
Initialize w0, b0;
while t++ < max iterations do
wt+1 ← wt − η∇wt;
bt+1 ← bt − η∇bt;

end

Shallow neural network

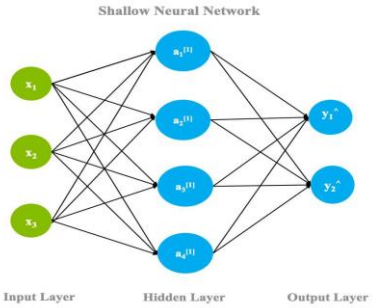
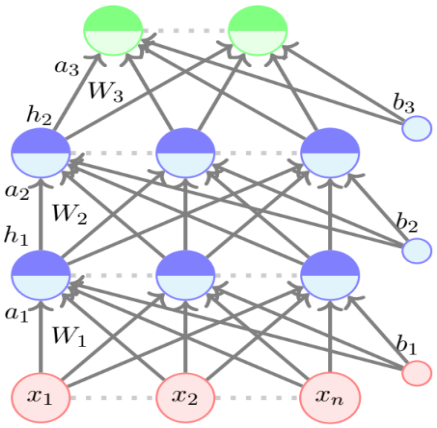


Fig 4: Gradient Descent



The perceptron fit inside the network and the flow is completed. a neural network contains three layers: *input layer hidden layer*, and *output layer*. As in the given figure, a network with just one hidden layer is termed a *shallow neural network.*

After one forward pass is completed, the output layer must compare its results to the actual ground truth labels and adjust the weights based on the differences between the ground truth and the predicted values. This process is a backward pass through the neural network and is known as *backpropagation*.

Deep Neural Networks

A deep neural network is a shallow neural network with more than one hidden layer. Each neuron in the hidden layer is connected to many others[2]. Each arrow has a weight property attached to it, which controls how much that neuron's activation affects the others attached to it.

The term 'deep' refer to deep learning is attributed to these deep hidden layers and derives its effectiveness from it. Selecting the number of hidden layers depends on the nature of the problem and the size of the data set. Figure6: shows a deep neural network with two hidden layers.

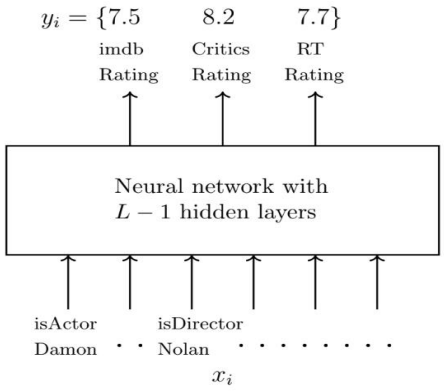
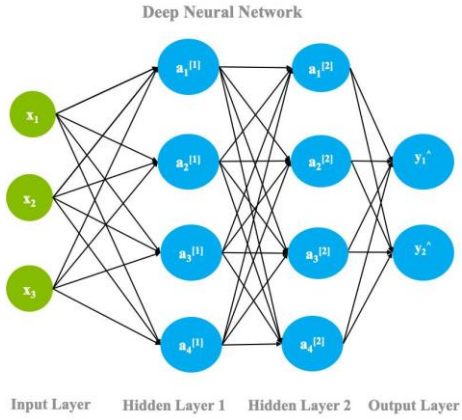


Fig 6: Deep Neural Network

Fig 3: Backpropagation



where ∇θt =∂L (θ) ∂wt,∂L (θ)∂bt T

Now, in this feedforward neural network, instead of θ = [w, b] we have
θ = [W1, W2, .., WL, b1, b2, .., b L]
We can still use the same algorithm for learning the parameters of our model

How to choose the loss function L (θ) ?

How to compute ∇θ which is composed of:
∇W1, ∇W2, ..., ∇WL−1 ∈ R^ n×n, ∇WL ∈ R^ n×k

∇b1, ∇b2, ..., ∇bL−1 ∈ R^n and ∇bL ∈ R^k ?

The choice of loss function depends on the problem, let illustrate this with the help of two examples. Consider our movie example again, but this time we are interested in, predicting ratings.
Here yi ∈ R3.

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3}(\hat{y}_{ij} - y_{ij})^2$$

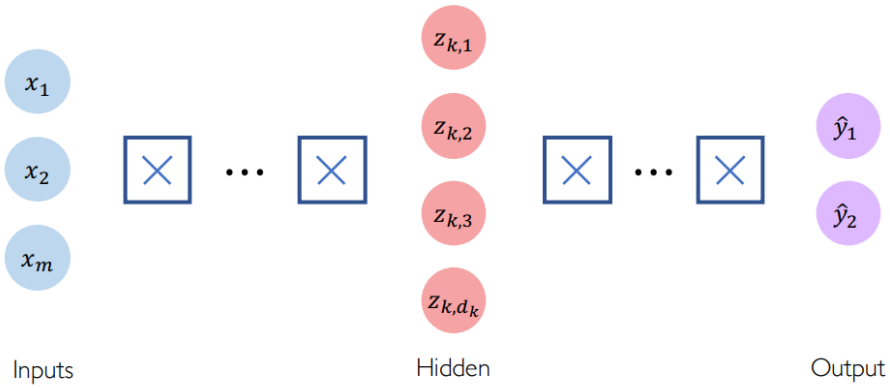
The squared error loss

The loss function should capture how much ˆyi deviates from yi

If yi ∈ R^n then the squared error loss, can capture this deviation.

Deep Neural Networks are no more than a **stacking** of multiple perceptron (hidden layers) to produce an output.

Fig 5: predicting Movie ratings



$$z_{k,i} = \theta_{0,i}^{(k)} + \sum_{j=1}^{d_{k-1}} g(z_{k-1,j})\,\theta_{j,i}^{(k)}$$

Training a Neural Network

A set of X-ray images, the model to automatically distinguish that are related to a sick patient from the others. For this we need machine learning models, like humans, need to learn to differentiate between the two categories of images by **observing** some images of both sick and healthy individuals. Automatically understand patterns that better describe each category. This is called as **training phase**.

A pattern is a weighted combination of inputs like images, parts of images or other patterns. **The training phase is more than the phase during which the estimate the weights also called parameters of the model.**

The objective function we must optimize, function shall be constructed on reflect the performance of the training phase. The prediction task objective function is usually called **loss function** and measures the cost incurred from incorrect predictions. When the model predicts something that is very close to the true output then the loss function is very low, and vice-versa.

Applications of Deep Learning

Health care: huge amounts of data, health care use cases have been a perfect fit for applying deep learning. Using image recognition, cancer detection from MRI imaging and x-rays has been surpassing human levels of accuracy. Drug discovery, clinical trial matching, and genomics have been other popular health care-based applications.

Autonomous vehicles: Though self-driving cars is a risky field to automate, it has recently taken a turn towards becoming a reality. From recognizing a stop sign to seeing a pedestrian on the road, deep learning-based models are trained and tried under simulated environments to monitor progress.

e-commerce: Product recommendations has been one of the most popular and profitable applications of deep learning. With more personalized and accurate recommendations, customers can easily shop for the items they are looking for and are able to view all the options that they can choose from. This also accelerates sales and thus, benefits sellers.

Personal assistant: advancements in the field of deep learning, having a personal assistant is devices like Alexa or Google Assistant. These smart assistants use deep learning in various aspects such as personalized voice and accent recognition, personalized recommendations, and text generation.

Challenges in deep learning

Advancements in this field in recent years are primarily because of the increase in computing power and high-performance graphical processing units (GPUs), coupled with the large increase in the wealth of data these models have at their disposal for learning, as well as interest and funding from the community for continued research. Though deep learning has taken off in the last few years, it does come with its own set of challenges that the community is working hard to resolve.

Conclusion:

This paper we forced on the fundamental building block of Deep Learning, which is the Perceptron, learned about stacking this perceptron together to compose more complex hierarchical models and how to mathematically optimize these models' using backpropagation and gradient descent. Finally, practical challenges of training these models in real life and some best practices like adaptive learning, batching and regularization to combat overfitting.

Deep learning algorithms have proven to beat human-level accuracy, there is no clear way to backtrack and provide the reasoning behind each prediction that's made. This makes it difficult to use in applications such as finance where there are mandates to provide the reasoning behind every loan that is approved or rejected.

References

1. Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In Large-Scale Kernel Machines. 2007.
2. Y. Bengio, O. Delalleau, and N. Le Roux. The curse of highly variable functions for local kernel machines. In NIPS, 2005.
3. Y. Bengio, P. Lamblin, V. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In NIPS, 2007.
4. G. E. Hinton, S. Osindero, and Yee-Whye Teh. A fast-learning algorithm for deep belief nets. Neur. Comput., 18:1527–1554, 2006.
5. P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In Parallel Distributed Processing, pages 194–281. 1986.
6. D. H . Ackley, G. E . Hinton, and T. J . Sejnowski. A learning algorithm for Boltzmann machines. Cognitive Science, 9:147–169, 1985.

7. Z. Ghahramani. Unsupervised learning. In Adv. Lect. Mach. Learn., pages 72–112. 2004.
8. M. Ranzato, Y.-L. Boureau, S. Chopra, and Y. LeCun. A unified energy-based framework for unsupervised learning. In AISTATS, 2007.
9. V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In ICML, 2010.
10. A. Krizhevsky. Convolutional deep belief networks on CIFAR-10. Technical report, Univ. Toronto, 2010