



# MACHINE LEARNING

---

ISHITA GUPTA  
BS ECONOMICS  
IIT BOMBAY

# Introduction


According to Arthur Samuel, a pioneer in the field of artificial intelligence and computer gaming,

"Machine learning is a field of Computer Science that gives computers the ability to learn without being explicitly programmed" Now, let's get to how a Modern Computer Scientist thinks about a Machine Learning Algorithm: "A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ."

Of course, we are no computer scientists, so let's try to put it in a more layman terms. We can say that a machine learning algorithm is a computer program that allows a machine to do get better at doing a job by, if I may say, repeated practice without actually explicitly being programmed to that job. Of course, here the practice is the training set we provide the algorithm with (The training set is a collection of relevant data related to the job/task we wish to accomplish). This may sound something like teaching a machine and this is what excites many people.

The Big AI dream, though really far away from being realized, can one-day be achieved through Machine Learning techniques. Though we will not talk about AI here, I shall try to touch upon neural networks here, which were our first attempts at trying to emulate the human brain.

We come into contact with products based on machine learning every day in our life without even realizing it. From the famous google page rank algorithm to the applications in financial trading, machine learning is everywhere. Even the



algorithm that email service providers use for identifying spam is based on machine learning. Machine learning is almost everywhere. So, let's jump to the question of what is machine learning and what is a machine learning algorithm.

## Regression and Classification

### Regression

Regression is the task of predicting a continuous quantity.

A regression algorithm may predict a discrete value, but the discrete value in the form of an integer quantity.

Regression predictions can be evaluated using root mean squared error, whereas classification predictions cannot.

### Classification

Classification is the task of predicting a discrete class label.

Classification predictions can be evaluated using accuracy, whereas regression predictions cannot.

A classification algorithm may predict a continuous value, but the continuous value is in the form of a probability for a class label.

Now let's get to core machine learning. In machine learning there are two kinds of problems, namely, supervised learning problems and unsupervised learning problems.

Supervised Learning: Supervised learning is the machine learning task of inferring a function from labelled training data.

**Unsupervised Learning:** Unsupervised learning is the type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses.

The **supervised machine learning** problem where  $y$  is continuous is a Regression where as a supervised machine learning problem where  $y$  is discrete is a Classification.

The most common type of Regression is Linear Regression done by getting a best fit line calculated using Least Squares Method.

## Least-squares estimates

- For a simple linear regression equation:

$$y = a + bx$$

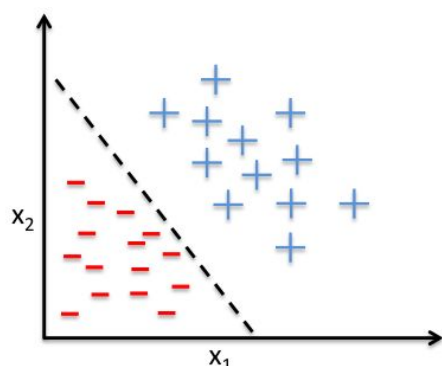
We have,

$$b = \frac{n\sum xy - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$a = \bar{y} - b\bar{x}$$

Where,  $\bar{y} = \sum y / n$  and  $\bar{x} = \sum x / n$

On the other hand, For the case of a Binary Linear classifier, the Hypothesis function has the following form:



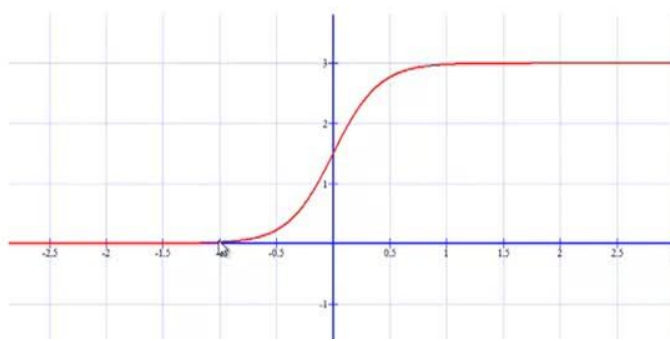
**Example of a linear decision boundary for binary classification.**

We are basically saying that it's a function of a linear combination of all the components  $Xu$  (hence the word "Linear"). The word 'Binary' refers to the fact that there are only two classes i.e.  $y$  can only take two discrete values in the problem we are dealing with. And the word 'classifier' refers to the fact that the above function is being used for classifying a given input into one of the two classes present. One of the classes is chosen as a positive class and the other class is chosen as negative class with  $y = 0$  corresponding to the negative class and  $y = 1$  corresponding to the positive class. The function  $f$  is chosen so that  $h(x)$  represents the probability of the input parameters corresponding to the positive class i.e. the corresponding to  $y = 1$ . One particular case of a Binary Linear Classifier that we are going to talk about is the Logistic classifier. For a logistic classifier the function  $f$  is the Logistic Function:

## Logistic Growth Models

Logistic growth model:

$$y = \frac{a}{1 + be^{-rx}}$$



# Neural Networks

A **neural network** is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. **Neural networks** can adapt to changing input; so, the **network** generates the best possible result without needing to redesign the output criteria.

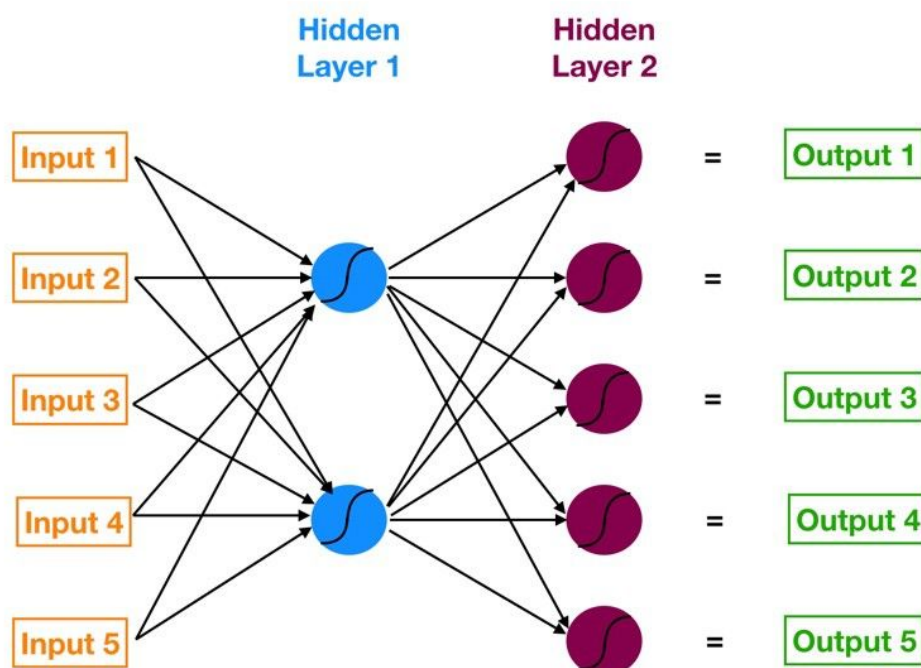
1) Neurons- A neural network is a graph of neurons. A neuron has inputs and outputs. Similarly, a neural network has inputs and outputs. The inputs and outputs of a neural network are represented by input neurons and output neurons. Input neurons have no predecessor neurons, but do have an output. Similarly, an output neuron has no successor neuron, but does have inputs.

2) Connections and Weights- A neural network consists of connections, each connection transferring the output of a neuron to the input of another neuron. Each connection is assigned a weight.

3) Propagation Function- The propagation function computes the input of a neuron from the outputs of predecessor neurons. The propagation function is leveraged during the forward propagation stage of training.

4) Learning Rule- The learning rule is a function that modifies the weights of the connections. This serves to produce a favoured output for a given


input for the neural network. The learning rule is leveraged during the backward propagation stage of training.



## Hyperparameter Tuning

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. ... These measures are called hyperparameters, and have to be tuned so that the model can optimally solve the machine learning problem.

While our model learns the parameters while training, there are some values for optimizing the rate and accuracy at which these parameters are learnt. Since these controls the parameters, they are called hyperparameters and thus this process of searching for the ideal model architecture is referred to as hyperparameter tuning. These hyperparameters might address model design questions such as:

- 
- I. What degree of polynomial features should I use for my linear model?
  - II. What should be the maximum depth allowed for my decision tree?
  - III. How many neurons should I have in my neural network layer? I
  - V. How many layers should I have in my neural network?
  - V. What should I set my learning rate for gradient descent?

Grid search is arguably the most basic hyperparameter tuning method. With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results. Each model would be fit to the training data and evaluated on the validation data. Clearly, this is an exhaustive sampling of the hyperparameter space and can be quite inefficient.

Random Search differs from grid search in that we no longer provide a discrete set of values to explore for each hyperparameter; rather, we provide a statistical distribution for each hyperparameter from which values may be randomly sampled. One of the main theoretical backings to motivate the use of random search in place of grid search is the fact that for most cases, hyperparameters are not equally important.

Bayesian optimization belongs to a class of sequential model-based optimization (SMBO) algorithms that allow for one to use the results of our previous iteration to improve our sampling method of the next experiment. The previous two methods performed individual experiments building models with various hyperparameter values and recording the model performance for each. Because each experiment was performed in isolation, it's very easy to parallelize this process. However, because each



experiment was performed in isolation, we're not able to use the information from one experiment to improve the next experiment.

## TENSORFLOW

TensorFlow is a [free](#) and [open-source software library](#) for [dataflow](#) and [differentiable](#) programming across a range of tasks. It is a symbolic math library, and is also used for [machine learning](#) applications such as [neural networks](#).<sup>[5]</sup> It is used for both research and production at [Google](#)



# TensorFlow

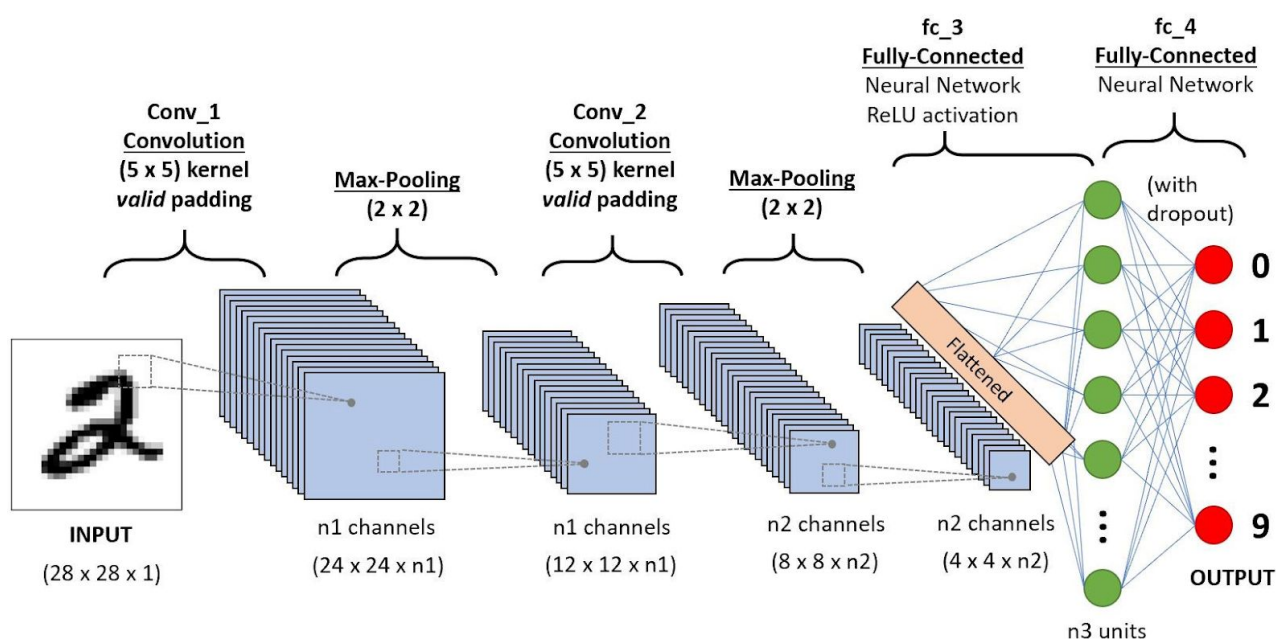
Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. It was first made public in late 2015, while the first stable version appeared in 2017.


It is open source under Apache Open Source license. You can use it, modify it and redistribute the modified version for a fee without paying anything to Google. It makes setting up Neural Networks look like a child's play. While I sometimes even struggled to write the code for back-propagation, I just needed less than 10 lines to set up a whole Neural Networks with TensorFlow.

## Convolutional Neural Networks (CNN)

A convolutional neural network (**CNN**) is a specific type of artificial neural network that uses perceptron, a **machine learning** unit algorithm, for supervised **learning**, to analyse data. CNNs apply to image processing, natural language processing and other kinds of cognitive tasks.

CNNs are basically a class of deep neural networks, most commonly applied to analysing visual imagery. In a usual Neural Network, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss



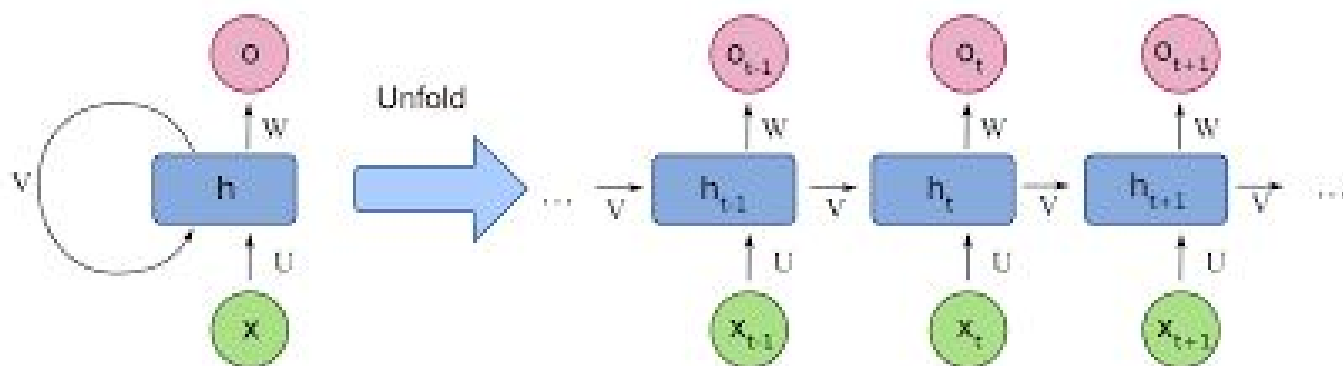


function. However, CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme. The role of the Convnet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

## RNN

A **recurrent neural network (RNN)** is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition[1] or speech recognition.

The ability to work with sequence data, like music lyrics, sentence translation, understanding reviews or building chatbots – all this is now possible thanks to sequence modelling, and RNNs are the first step to begin with. Recurrent Neural Network (RNN) are a type of Neural Network where the output from the previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

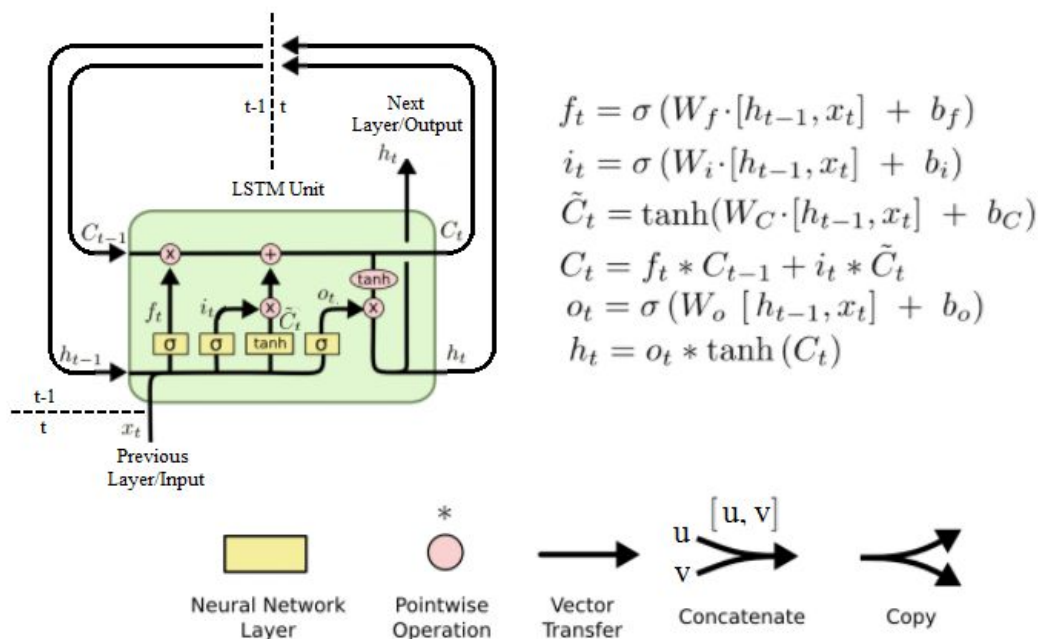


But RNNs themselves have a major problem; error propagate backwards from output to input layer propagating the input error gradient. With deeper neural networks issues can arise from back propagation like vanishing and exploding gradients. Vanishing Gradients: As we go back to the lower layers, gradient often gets smaller, eventually causing weights to never change at lower layers.

## LSTM

Long short-term memory is a deep learning system that avoids the vanishing gradient problem. LSTM is normally augmented by recurrent gates called "forget" gates. LSTM prevents back propagated errors from vanishing or exploding. Instead, errors can flow backwards through unlimited numbers of virtual layers unfolded in space. That is, LSTM can learn tasks that require memories of events that happened thousands or even millions of discrete time steps earlier. LSTM works even given long delays between significant events.

# Understanding LSTM Networks



## KEY PROJECT

I used an RNN or **recurrent neural network** trained on data of all of Shakespeare's previous writings, and then used it to output completely new text based on what it learned!

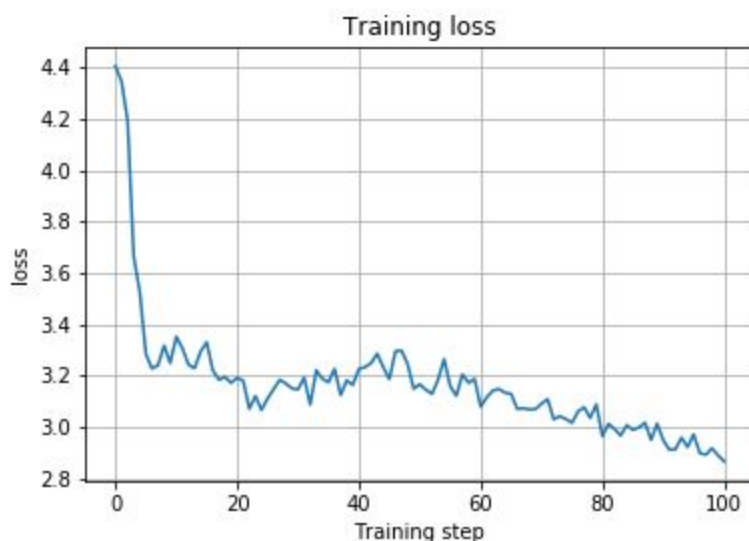
**1: Importing libraries and data pre-processing** – Importing libraries like NumPy and Porch

**2: Defining the Model** - implement dropout for regularization and also create character dictionaries within the network. We'll have 1 LSTM unit and also 1 fully connected layer.

**3: Training** - Define an optimizer (Adam) and loss (cross entropy loss). We then create the training and validation data and initialize the hidden state of the RNN. We'll loop over the training set, each time encoding the data into one-hot vectors, performing forward and backpropagation, and updating the gradients.

**4: Generating new Shakespeare text** - Define a sampling method that will use the previous method to generate an entire string of text, first using the characters in the first word (prime) and then using a loop to generate the next words using the topek function, which chooses the letter with the highest probability to be next.

## Graph of loss vs no. Of steps



## What I learnt in a nutshell:

1. Learnt basics of python, linear algebra got familiar with many more libraries of python,
2. Learnt basics of Machine Learning - Linear Regression, Classification & Cauterization.
3. Learnt about Hyper-parameter tuning and regularisation.
4. Learnt about TensorFlow and how it makes implementing neural networks such a bliss.
5. Learnt about Convolutional Neural Networks.
6. Learnt about Sequence Models: RNNs, Word Embeddings, Beam Search, Attention Models.
7. Implemented all of the above as assignments And assignments of Stanford University on machine learning and the CS231 assignments.