What is Machine Learning?

In 1959, Arthur Samuel, a computer scientist who pioneered the study of artificial intelligence, described machine learning as “the study that gives computers the ability to learn without being explicitly programmed.”

Alan Turing’s seminal paper (Turing, 1950) introduced a benchmark standard for demonstrating machine intelligence, such that a machine has to be intelligent and responsive in a manner that cannot be differentiated from that of a human being.

Machine Learning is an application of artificial intelligence where a computer/machine learns from the past experiences (input data) and makes future predictions. The performance of such a system should be at least human level.

A more technical definition given by Tom M. Mitchell’s (1997) : “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.” Example:

A handwriting recognition learning problem:Task T: recognizing and classifying handwritten words within images
Performance measure P: percent of words correctly classified, accuracy
Training experience E: a data-set of handwritten words with given classifications

In order to perform the task T, the system learns from the data-set provided. A data-set is a collection of many examples. An example is a collection of features.

Machine Learning Categories

Machine Learning is generally categorized into three types: Supervised Learning, Unsupervised Learning, Reinforcement learning

Supervised Learning:

In supervised learning the machine experiences the examples along with the labels or targets for each example. The labels in the data help the algorithm to correlate the features.

Two of the most common supervised machine learning tasks are classification and regression.

In classification problems the machine must learn to predict discrete values. That is, the machine must predict the most probable category, class, or label for new examples. Applications of classification include predicting whether a stock's price will rise or fall, or deciding if a news article belongs to the politics or leisure section. In regression problems the machine must predict the value of a continuous response variable. Examples of regression problems include predicting the sales for a new product, or the salary for a job based on its description.

Unsupervised Learning:

When we have unclassified and unlabeled data, the system attempts to uncover patterns from the data . There is no label or target given for the examples. One common task is to group similar examples together called clustering.

Reinforcement Learning:

Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps. This method allows

machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal. For example, maximize the points won in a game over many moves.

Deep learning is also learned data like other machine learning algorithms but using neural networks called Artificial Neural Network (ANN). We can use deep learning libraries like TenserFlow(written by Google) , Keras(running on top of TensorFlow, Microsoft Cognitive Toolkit, etc), PyTorch(written by Facebook)

Neurons in an ANN are organized into layers. Following the type of layers, we can use in ANN on basis of the problem statement.

Dense Layer

Convolutional layers

Pooling layers

Normalization layers

Recurrent layers

Each type of layer is used for different types of task. Convolutional layers commonly used for Image classification and Dense layers is having each input to each output within its layer.

Each layer will pass information to the next layers from left to right(above in picture) in ANN. Each connection between nodes (neurons)will have weight-associated which is just a number. When any input pass to one given node then the input will be multiplied by weight and passed to next node. In the next node in next layer will compute passed input using activation function. This process will continue until out layer reached in ANN.

BERT, which is an acronym for Bi-
directional Encoder Representation from Transformer, is a state of the art language model which can be used for various natural language processing (NLP) tasks. My objective is to introduce BERT at a

high leveland enable you to create practical applications using BERT. You need to have basic knowle dge of Python as well as a basic idea of machine learning. After going through this post, you should b e able to use BERT for Binary or multiclass classification or Regression model or Question

Answering Application. BERT brings the transfer learning paradigm into the natural language proces sing area. Transfer learning means a model developed for a task can be reused as a starting point for another task. BERT is trained on the entirety of Wikipedia text (~2.5 billion words) as well as a book c orpus (800 million words). You don’t need to repeat this compute intensive process to make practica l use of BERT. For specific tasks such as classification or question answering you just need to do incre mental training on a much smaller dataset. This process is called fine tuning. This is like getting a per son who is proficient in English and providing them extra guidance on how to identify positive senti ment vs negative sentiment from movie reviews. This is a quick introduction on the BERT pre-

training process. For practical uses, you will get a BERT pretrained model and you do not need to per form this step. BERT takes 2 chunks of text, which may include multiple sentences, as inputs (shown i n the diagram above). These 2 chunks are referred to as Sentence 1 and Sentence 2 in the diagram a bove. Before feeding sentences to BERT, 15% of words are masked. Also, sentence 2 may not always be in continuation of sentence 1.

BERT's pre-
training process is like teaching the English language to the BERT model so that it can be used for var ious tasks which need knowledge of English. This is accomplished by 2 practice tasks given to BERT: 1 ) Predict masked (hidden) tokens: To illustrate, words "favorite" and "to" are masked in the diagram. BERT will try to predict these masked tokens as part of pretraining. This is like the "fill in the blanks" task we may give to a student learning English. While trying to fill the missing words, the student will learn the language. This is referred to as Masked Language Model (MLM) in the natural language processing area. 2) Next sentence prediction: Along with the technique \

mentioned above, BERT tries to predict if sentence 2 comes right after sentence 1 or not. This provi des deeper understanding on sentence dependencies. To use BERT for practical \

use, we need to fine-tune it for specific tasks. This process finetunes the pre-
trained model so that it can perform specific tasks such as text classification, sentiment \

analysis, question answering. In this process, BERT parameters get adjusted to do the specific task.