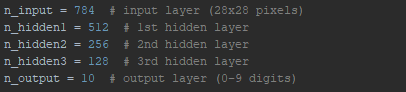
Introduction

Artificial intelligence is a branch of computer science that aims to create intelligent machines (machines learning) that work and react like humans. It has become an essential part of the technology industry. Some of the activities computers with artificial intelligence are designed for include: Speech recognition, Handwrite recognition, Learning, Planning, Problem-solving, self-driving cars. There are two types in machines learning are unsupervised type and supervise type. The first type is Unsupervised learning problems can be further grouped into clustering and association problems. Some popular examples of unsupervised learning algorithms are:  
 - k-means for clustering problems.  
 - Apriori algorithm for association rule learning problems.  
The other type is Supervised Machine Learning It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance. [1] The majority of practical machine learning uses supervised learning. There are many tools used in Machine learning such as artificial neural networks. So what is artificial neural networks (ANN) and what's used for and how does it work, how do we train it and what are the tools used for train.

Artificial Neural Networks:   
Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, the inspired by biological neural networks which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

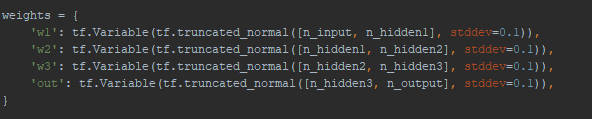
Our work:   
In this report will write a computer program implementing a neural network that learns to recognize handwritten digits. We're focusing on handwriting recognition because it's a simple prototype problem for learning about neural networks in general and it's also the first step to entering to a machine learning. In our work, we will use open source library the MINST training dataset which contains tens of thousands of scanned images of handwritten digits, together with their correct classifications. The images are greyscale and 28 by 28 pixels in size. The MNIST data comes in two parts. The first part contains 60,000 images to be used as training data. The second part of the MNIST data set is 10,000 images to be used as test data. We'll use the test data to evaluate how well our neural network has learned to recognize digits. And TensorFlow is an open-source Python library developed by the Google Brain labs for deep learning research, you will take hand-drawn images of the numbers 0-9 and build and train a neural network to recognize and predict the correct label for the digit displayed.

The architecture of the neural network:   
The architecture of the neural network is a set of connected neurons organized in layers: input layer and hidden layers and output layer.   
The input layer is for inputting values and the output layer is for the output values hidden layer is used for all of the layers in between the input and output layers in our work we use three hidden layers.  
The network architecture show in the picture below:

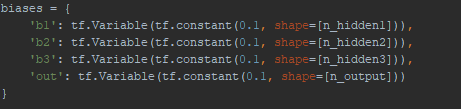


Build our network:   
To build our network, we will set up the network as a computational graph for TensorFlow to execute. The core concept of TensorFlow is the tensor, a data structure similar to an array or list. Initialized, manipulated as they are passed through the graph, and updated through the learning process. The parameters that the network will update in the training process are the weight and bias values, so for these we need to set an initial value rather than an empty placeholder. These values are essentially where the network does its learning, as they are used in the activation functions of the neurons, representing the strength of the connections between units.

We'll use random values from a truncated normal distribution for the weights. We want them to be close to zero

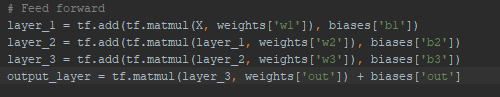


For the bias, we use a small constant value to ensure that the tensors activate in the initial stages and therefore contribute to the propagation.



Forward propagation:   
Forward propagation is the process of feeding the Neural Network with a set of inputs to get their dot product with their weights then feeding the latter to an activation function and comparing its numerical value to the actual output called “the ground truth”.

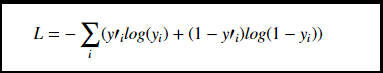
Now, to set up the layers of the network by defining the operations that will manipulate the tensors. Like in the picture below



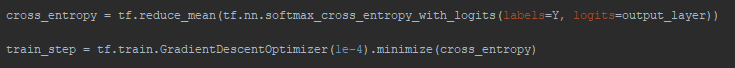
Each hidden layer will execute matrix multiplication on the previous layer’s outputs and the current layer’s weights, and add the bias to these values.

#### Cross entropy error

The final step in building the graph is to define the loss function that we want to optimize. A popular choice of loss function in TensorFlow programs is cross-entropy, also known as log-loss, which quantifies the difference between two probability distributions (the predictions and the labels). A perfect classification would result in a cross-entropy of 0, with the loss completely minimized. And it defines by relation below



We also need to choose the optimization algorithm which will be used to minimize the loss function. A process named gradient descent optimization is a common method for finding the (local) minimum of a function by taking iterative steps along the gradient in a negative (descending) direction. The gradient descent optimization algorithms already implemented in TensorFlow as:



Neural Network learning:  
The essence of the training process in deep learning is to optimize the loss function, by a way to minimize the difference between the predicted labels of the images, and the true labels of the images, this will be changed (update)in weights .The process involves four steps which are repeated for a set number of iterations:

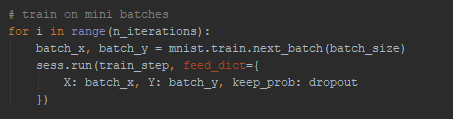
-Propagate values forward through the network

-Compute the loss

-Propagate values backwards through the network

-Update the parameters

At each training step, the parameters are adjusted slightly to try and reduce the loss for the next step. As the learning progresses, we should see a reduction in loss, and eventually, we can stop training and use the network as a model for testing our new data. The picture below show how to implement the process:



[1] <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>