ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Recognition of Data Items using Tensor Flow

Kambampati Sowmya
Department of Information Technology,
Sreenidhi Institute of Science and Technology,
Hyderabad, Telangana-501301, India.

Sameera Turupu
Department of Computer Science & Engineering,
BVRIT College of Engineering for Women,
Hyderabad, Telangana-500090, India.

ABSTRACT
Nowadays, the interest around Machine Learning, is one of the fastest growing. So far from the side of the BSC Computer Science Department, that mainly uses his computational power for data mining and modelling analysis, the main purpose was to verify the difficulty to adapt his infrastructure "asterix", from the GPU Center of Excellence at BSC/UPC, the to Deep Learning. Instead, from the side of UPC IPG, there was the interest to test the environment developing a model for Object Tracking in Video that was suitable for the ILSVRC VID challenge. To achieve the first goal and analyze the workload on the machine, started to become an active user of the Tensor Flow community, learning from posts and blogs and I decided to implement a Virtual Environment that, led us to use different dependencies and different versions of the library software, depending on the model and purpose to reach. Till now, from the computer science point of view, this environment was the best choice and the most useful experience to work with, showing the easiness of use and implementation.

I. INTRODUCTION

Machine Learning gives "the ability to learn without being explicitly programmed to a computer." Machine learning is employed in the range of computing tasks where designing and programming explicit algorithms with good and high performance is very difficult or unfeasible, many example applications include email filtering, detection of intruders on network or malicious insiders working towards a data breach, optical character recognition (OCR) [1], learning to rank and computer vision. Inspired from the study of pattern recognition and computational learning theory in artificial intelligence, the study and construction of algorithms explored in machine learning that can learn from and make predictions on data as well as decisions on data, through building a model from sample inputs.

A core objective of a learner is to generalize from its existing behavior. Generalization in this context is the ability of a learning machine or a device which under programming to perform accurately on new, unseen examples or tasks after having experienced a learning data. The learner (i.e., computer) should be ableto build a general model which can produce enough and accurate predictions in new cases about this space. Machine learning is a method used to devise complex models and algorithms that lend themselves to prediction in commercial use, this is known as predictive analytics, and it is also a branch of advanced analytics [2]. These predictive analytical models allow researchers, data scientists, engineers and analysts to "produce reliable, repeatable decisions and results".
Machine learning tasks are typically classified into three categories, depending on the nature of the learning "signal" or "feedback" available to a learning machine. These are

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

Another categorization of machine learning tasks arises based on the desired output of the machine-learned system.

1. Classification
2. Regression
3. Clustering

Cite this article as: Kambampati Sowmya & Sameera Turupu, "Recognition of Data Items using Tensor Flow", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 6 Issue 7, 2019, Page 1-6.

SUPERVISED LEARNING:

A type of inductive learning which deals with theoretical results in machine learning is called as supervised learning. Supervised learning is the machine learning task of concluding from labeled training data. The training data consist of a set of training examples which are well known to the programmer. In supervised learning, each example is a pair comprises of an input object and a desired output value that demonstrates the intended relation of input and output values. Thenthe learner or machine (i.e., computer) is supposed to approximate the correct output, even for those examples that have not been shown during training.

Steps performed to solve the given problem of supervised learning:

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set i.e., the user or programmer should know about that data.
2. Gather a training set. The training set needs to be representative of the real-world use of the data items. Thus, a set of input objects are classified into respective categories is gathered and corresponding outputs are also gathered from measurements.
3. Determine the input feature representation of learned function or calculation. The accuracy of learned function depends strongly on how the input object or data item is represented. Typically, the input object is transformed into a feature vector, which contains several features that are descriptive of specific object.
4. Regulate the structure of the learned function and corresponding learning algorithm.
5. Run the learning algorithm i.e., the classifier on the gathered training set or training data. This creates different parameters relating to accuracy and performance of training.
6. Evaluate the accuracy of the learned function. The performance of the resulting functionshould be measured on a test set that is separate from the training set.

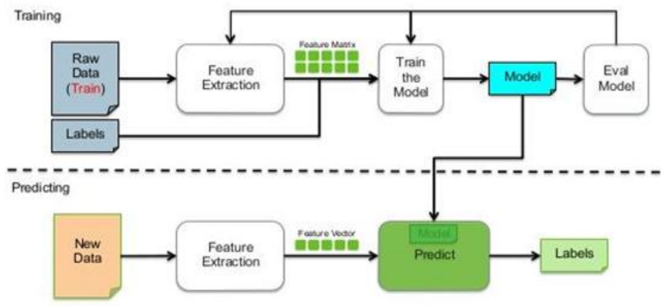

Fig. 1: Supervised Learning Workflow

DEEP LEARNING:

Feature extraction in machine learning requires a programmer to tell the computer or machine learner what kinds of things it should be looking for and that will be formative in making decisions, which can be a time-consuming process. Deep learning is a class of machine learning algorithms [3] that does not require manualintervention and use a cascade of many layers for feature extraction. Each successive layer uses the output from the previous layer as input and proceeds further. In a simple case, there might be two sets of neurons, one set that receives an input signal and one that sends an output signal. As input layer receives an input it passes on a modified version of the input to the next layer. In a deep network, there are many layers between the input and the output (and the layers are not made of neurons but it can help to think of it that way), allowing the algorithm to use multiple processing layers.

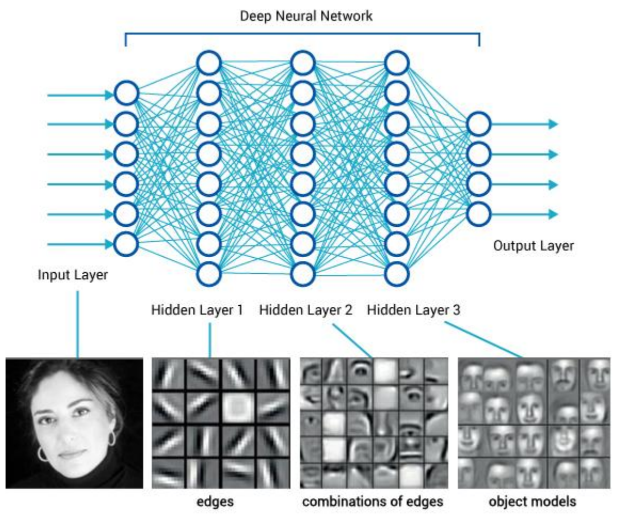

Fig. 2: Deep Learning

In deep learning, an observation or an output (e.g., an image) can be represented in many ways such as vector of intensity values as per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc.,For supervised learning tasks, deep learning techniques forestall include designing, by making an interpretation of the information into smaller middle of the road portrayals much the same as main parts, and infer layered structures which evacuate repetition in portrayal. When performing supervised learning on a multiclass classification problem, normal decisions for the enactment capacity and cost capacity are the softmax capacity and cross entropy work, individually.
Computational deep learning is firmly identified with class of speculations of mental health (particularly neocortical improvement) proposed by intellectual neuroscientists in the mid 1990's.

II. CONVOLUTIONAL NEURAL NETWORK

In machine learning, a convolutional neural network(CNN) [3] is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortexand are made up of neurons that have learnable weights and biases. Convolutional networks were inspired by biological processes and are variations of multilayer perceptron or nodes designed to use minimal amounts of pre-processing.

IMAGE RECOGNITION:

Convolution neural networks are mainly used in image recognition systems, recommender systems and in the fields of natural language processing. When applied to facial recognition, they could contribute to a large decrease in error rate and gave high success rate in using this as well as exploring in this fields. Convolutional Neural Networks (CNNs) consists of multiple layers of receptive fields. These are small neuron collections which process portions of the input image to learning function. The outputs of these collections are then attached or tiled so that their input regions overlap on each other, to obtain a higher-resolution representation of the original image; this is repeated for every such layer. Slating allows CNNs to tolerate translation of the input image.

Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters and they also consist of various combinations of convolutional and fully connected layers. A convolution operation on small regions of input is introduced to reduce the number of free parameters and improve generalization. One major advantage of convolutional networks reduces both memory footprint and improves performance by using shared weights in convolutional layers, which means that the same filter is used for each pixel in the layer.

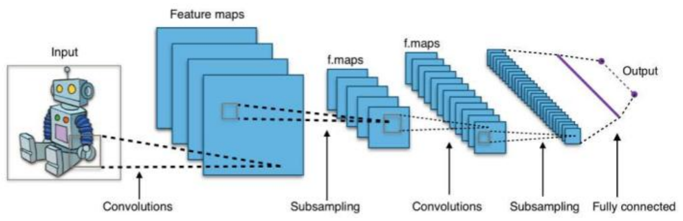

Fig. 3: Typical CNN architecture

POOLING LAYER:

Convolutional layers are often inter-weaved with pooling layers. There is a kind of layer called a max-pooling layer that is extremely popular. A max-pooling layer takes the maximum of features over small blocks or subsets of a previous layer. It partitions the input image is set onto non-overlapping rectangles and, for each such sub-region, it outputs the maximum in specific sub-region. The output tells us if a feature was present in a region of theprevious layer, but not precisely what and where. The instinct is that the correct area of an element is less vital than its unpleasant area in respect to different elements.

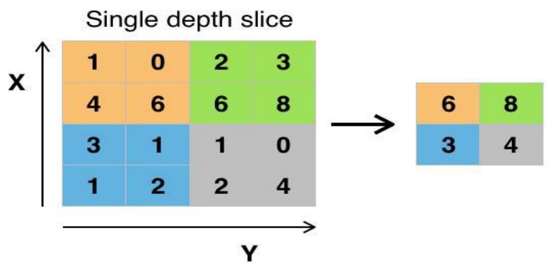

Fig. 4: Max Pooling with a 2x2 filter and stride = 2

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The pooling layer operates independently on every depth slice of the input and resizes it spatially to maximum function. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples at every depth slice in the input by 2 along width and height, discarding 75% of the activations. Here, every max operation is over 4 numbers.

STRUCTURE OF CONVOLUTIONAL NEURAL NETWORK:

When programming, we write function once and use it in many places – not writing the same code a hundred times in different places makes it faster to program, and results in fewer bugs. Similarly, a convolutional neural network can learn a neuron once and use it in many places, making it easier to learn the model and reducing error (advantage of using neural networks). Assume you want a neural network to look at audio samples and predict whether a human is speaking or not. Maybe you want to do more analysis if someone is speaking.You get audio samples at different points in time. The samples are evenly spaced.The easiest way to try and classify them with a neural network is to just connect them all to a fully-connected layer. There are a bunch of different neurons, and every input connects to every neuron.

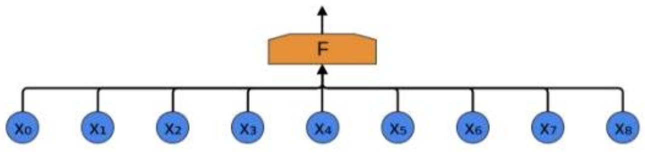

Fig. 5: Fully-Connected Layer

A more modern approach sees a sort of symmetry in the properties helpful to search for in the information. We think a great deal about local properties of the information. What frequency of sound is there around a given time? Are they increasing or decreasing? And so on. It's useful to know the frequencies at the beginning, it's useful to know the frequencies in the middle, and it's useful to know the frequencies at the end. Again, note that these are local properties, in that we only need to look at a small window of the audio sample in order to determine them. So, we can create a group of neurons, **A**, that look at small time segments of our data. **A**looks at all such segments computing certain features. Then the output of this convolutional layer is fed into a fully-connected layer **F**.

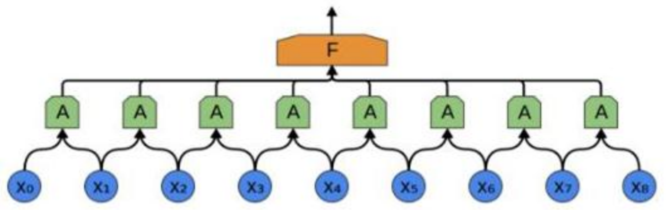

Fig. 6: Adding neurons to Fully-Connected Layer

In the above example, **A** only looked at segments consisting of two points. This isn't realistic. Usually, a convolutional layer's window would be much larger.In the following example, **A** looks at 3 points. That isn't realistic either, it's tricky to visualize **A** connecting to lots of points. In the real-worldexamples, it may look different number of points and they may or may not the same for all **A.**

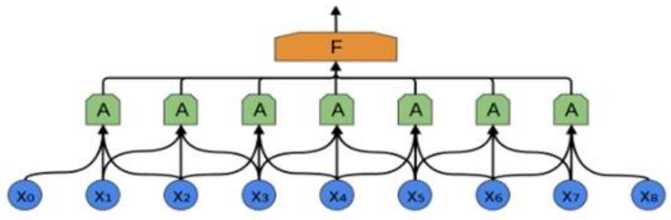

Fig. 7: Adding neurons to Fully-Connected Layer (more realistic way)

One extremely decent property of convolutional layers is that they're composable. You can feed the output of one convolutional layer into another. With each layer, the network can detect higher level, more conceptual elements.In the following example, we have a new group of neurons **B**. **B** is used to create another convolutional layer stacked on top of the previous one.

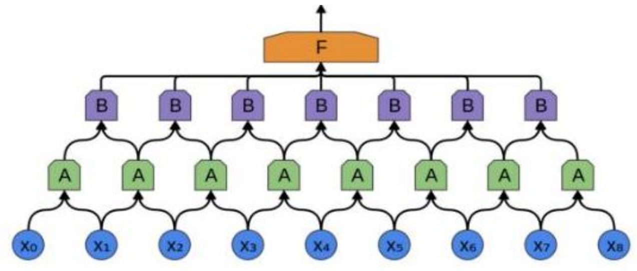

Fig. 8: Adding neurons to Fully-Connected Layer from the lower layers

Convolutional layers are often inter-weaved with pooling layers. In particular, max-pooling layer which is extremely popular. Max-pooling layers kind of "zoom out". They allow later convolutional layers to work on bigger segments of the data, because a small patch after the pooling layer corresponds to a much larger patch before it.They additionally make us invariant to some little changes of the information.

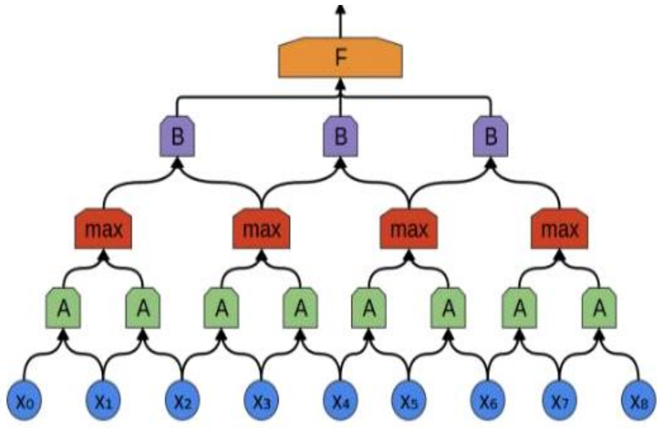

Fig. 9: Adding neurons to Fully-Connected Layer from the lower layers by max-pooling

In our previous examples, we've utilized1-dimensional convolutional layers. However, convolutional layers can work on higher-dimensional data too. Truth be told, the most well known accomplishment of convolutional neural systems is applying 2-dimensional convolutional neural networks to recognize images.In a 2-dimensional convolutional layer, instead of looking at segments, ***A*** will now look at patches. For each patch **A** will compute features. For example, it might learn to detect the presence of an edge or it might learn to detect a texture or perhaps a contrast between two colors.

CLASSIFIERS:

In machine learning, **classification** is the problem of identifying to which of set of categories in a new observation belongs, on the premiseof training set of data containing observations whose category membership is known.A case would allot a given email into "spam" or "non-spam" classes or doling out analysis to a given patient as depicted by observed characteristics of the patient.

In the terminal issue of machine learning, classification is viewed as an instance of supervised learning. An algorithm that implements classification, especially in concrete implementation, is known as **Classifier.** The term "classifier" sometimes also refers to the mathematics function, implemented by a classification algorithm that maps input data to a category. Classifier performance depends greatly on the characteristics of the data to be classified.

Classification can be thought of two different problems one is binary classification and the other is multiclass classification. In binary classification, a better understood task, only two classes are involved that are known and unknown classes, whereas multiclass classification involves assigning an object to one of several classes.

Linear classifiers:

A Linear Classification can be achieved by making classification based on the value of linear combination of characteristics and large number of algorithms for classification can be phrased in terms of linear functions that assign a score to each possible category 'k' by joining the features vector of an instance with a weight vector, using a dot product. The predicted category is the one with the highest score. This type of score functions is known as a linear function and has the general form

$$score(X_i, k) = \beta_k . X_i,$$

Where $\mathbf{X_i}$is the vector of feature for instance i, $\boldsymbol{\beta_k}$is the weights vector corresponding to category k, and score $\mathbf{(X_i, k)}$ is the score associated with assigning instance i to category k.

III. DESIGN

Inception v3 model:

Our brains make vision seem easy and understand easily whether we are seeing a lion or a jaguar but these are hard problems to solve with a computer.In past years, the field of machine learning has made tremendous progress on addressing these difficult problems. We've found that a kind of model called a deep convolutional neural network can achieve reasonable performance on identifying visually -- matching or exceeding human performance in some domains.

Researchers have proved steady progress in computer vision by validating their work against ImageNet an academic benchmark for computer vision. Sequential models continue to show improvements, each time achieving a new state-of-the-art result: QuocNet, AlexNet, Inception (GoogLeNet), BN-Inception-v2. We're now taking the next step by releasing code for running image recognition on our latest model, Inception-v3. It is trained for the ImageNet Large Visual Recognition Challenge using the data from 2012. This is a standard task in computer vision, where models try to classify entire images into 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher". For example, here are the results from AlexNet classifying some images:

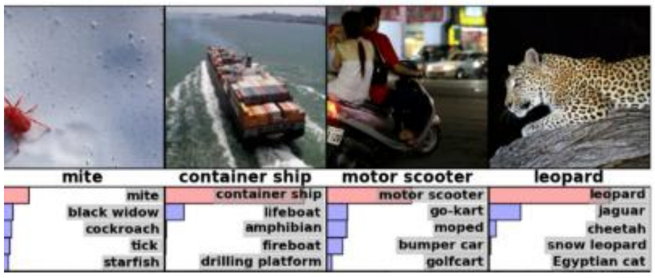

Fig. 10: AlexNet Classification

To compare models, we examine how often the model fails to predict the correct answer as one of their top 5 guesses -- termed "top-5 error rate". AlexNet achieved by setting a top-5 error rate of 15.3% on the 2012 validation data set; Inception (GoogLeNet) achieved 6.67%; BN-Inception-v2 achieved 4.9%; Inception-v3 reaches 3.46%.

IV. CONCLUSION

Deep convolutional neural network can achieve record breaking results on a highly challenging dataset using purely supervised learning. It is notable that our network’s performance degrades if a single convolutional layer is removed. Forexample removing any of the middle layer results in a loss of about 2% for the top-1 performance of the network. So the depth really is important for achieving results. Tensor Flow an open source deep learning library based on computational graphs. Its ability to perform fast automatic gradient computation, its inherent support for distributed computation and specialized hardware as well as its powerful visualization tools make it a very welcome addition to the field of machine learning. Its low-level programming interface gives fine-grained control for neural net construction.

V. SCOPE FOR FUTURE ENHANCEMENT

Machine Learning has a wide scope in future and they include

- Our understanding of neural networks will improve greatly.
- Natural Language Processing will begin to make sense.
- Machine learning pipelines will have increased levels of automation.
- Machine learning will be embedded everywhere.

REFERENCES

[1] Building Machine Learning Projects with Tensor Flow by Rodolfo Bonnin.

[2] https://codelabs.developers.google.com/codelabs/cpb102-txf-learning

[3] Getting Started with TensorFlow by Giancarlo Zaccone