**CHAPTER 2**

**PROJECT THEORY**

**2.1 Transfer Learning:-**

The definition of transfer learning is given in terms of domain and task. The domain ***D*** consists of: a feature space ***X*** and a marginal probability distribution ***P(X)***, where . Given a specific domain, ***D={X, P(X)}***, a task consists of two components: a label space ***Y*** and an objective predictive function ***f(.)***(denoted by ***T={Y , f(.)}*** ***T={Y ,f(. )}***), which is learned from the training data consisting of pairs , which consist of pairs **{xi , yi},** where **xi** ϵ ***X*** and**yi** ϵ ***Y***. The function ***f(.)*** can be used to predict the corresponding label, ***f(x)*** of a new instance ***x.***

Given a source domain ***DS***  and learning task ***TS***  target domain ***DT*** and learning task ***TT*** transfer learning aims to help improve the learning of the target predictive function ***fT(.)*** in ***DT*** using the knowledge in ***DS*** and ***TS***  where ***DS=DT***  or ***TS=TT***  .

The concept of Transfer Learning is used for Developing the Deep Learning Model. Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks. This area of research bears some relation to the long history of psychological literature on transfer of learning, although formal ties between the two fields are limited.

Modern image recognition models have millions of parameters. Training them from scratch requires a lot of labelled training data and a lot of computing power (hundreds of GPU-hours or more). Transfer learning is a technique that shortcuts much of this by taking a piece of a model that has already been trained on a related task and reusing it in a new model. In this Project we will reuse the feature extraction capabilities from powerful image classifiers trained on Image Net and simply train a new classification layer on top.

We will be using transfer learning, which means we are starting with a model that has been already trained on another problem. We will then retrain it on a similar problem. Deep learning from scratch can take days, but transfer learning can be done in short order.Though it's not as good as training the full model, this is surprisingly effective for many applications, works with moderate amounts of training data (thousands, not millions of labelled images, and can be run in as little as thirty minutes on a laptop without a GPU.

**2.2 Convolution Neural Networks:-**

Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on it(say: multiply it with a variable w and adds another variable b ) to produce a value (say; z= wx+b). This value is passed to a non-linear function called activation function (f) to produce the final output (activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid, which is:

The neuron which uses sigmoid function as an activation function will be called Sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH .One neuron can be connected to multiple neurons, like this:

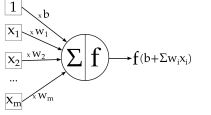


Fig-2

**Connection of a Neuron**

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

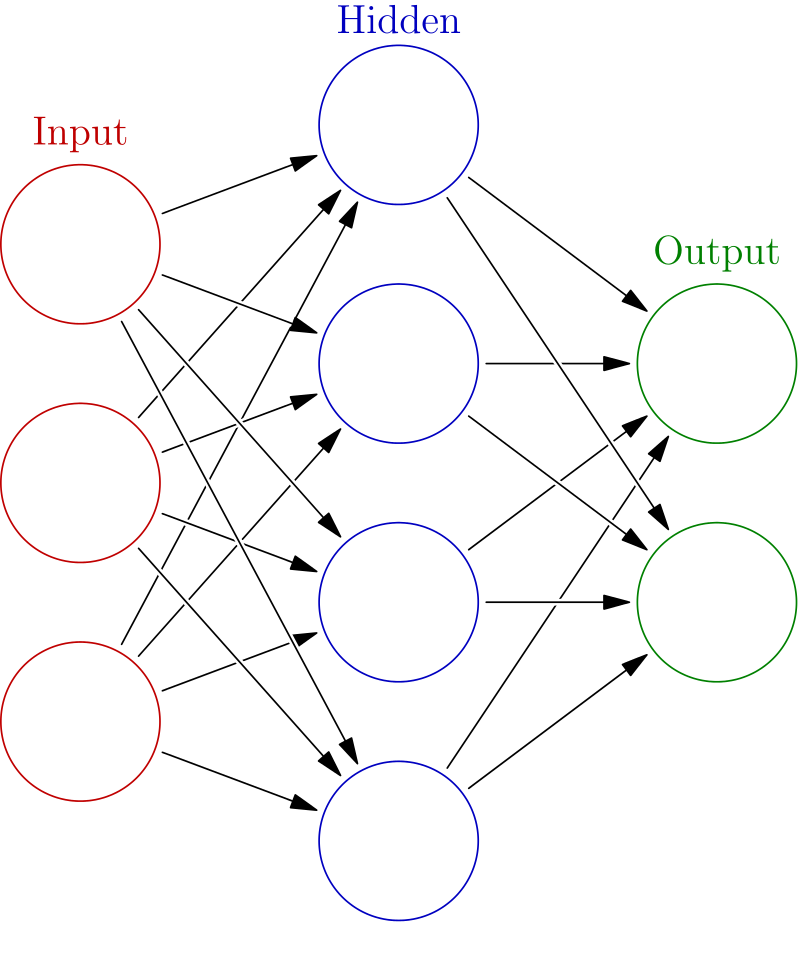


Fig-3

**Schematic Diagram of a Neural Network**

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptrons designed to require minimal pre-processing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.



Fig-4

**Convolutional Neural Network**

Convolutional networks are at the core of most state-of-the-art computer vision solutions for a wide variety of tasks. Since 2014 very deep convolutional networks started to become mainstream, yielding substantial gains in various benchmarks. Although increased model size and computational cost tend to translate to immediate quality gains for most tasks (as long as enough labeled data is provided for training), computational efficiency and low parameter count are still enabling factors for various use cases such as mobile vision and big-data scenarios.

**2.3 Mobile Nets:-**

MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. This architecture uses depthwise separable convolutions which significantly reduces the number of parameters when compared to the network with normal convolutions with the same depth in the networks. This results in light weight deep neural networks.

The normal convolution is replaced by depth wise convolution followed by point wise convolution which is called as depth wise separable convolution.

In the normal convolution, if the input feature map is of **Hi,Wi,Ci** dimension and we want **Co** feature maps with convolution kernel size K then there are **Co** convolution kernels each with dimension **K,K,Ci**. This results in a feature map of **Hi,Wi,Co** dimension after convolution operation.

In the depth wise separable convolution, if the input feature map is of **Hi,Wi,Ci** dimension and we want **Co** feature maps in the resulting feature map and the convolution kernel size is **K** then there are **Ci** convolution kernels, one for each input channel, with dimension K,K,1. This results in a feature map of **Ho,Wo,Ci** after depth wise convolution. This is followed by point wise convolution [1x1 convolution]. This convolution kernel is of dimension **1,1,Ci** and there are **Co** different kernels which results in the feature map of **Ho,Wo,Co** dimension.

This results in the reduction of number of parameters significantly and thereby reduces the total number of floating point multiplication operations which is favourable in mobile and embedded vision applications with less compute power.

By using depth wise separable convolutions, there is some sacrifice of accuracy for low complexity deep neural network.

MobileNets, a family of mobile-first computer vision models for TensorFlow, designed to effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application. MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. They can be built upon for classification, detection, embeddings and segmentation similar to how other popular large scale models, such as Inception, are used.

The Mobile Net is configurable in two ways:

* Input image resolution: 128,160,192, or 224px. Unsurprisingly, feeding in a higher resolution image takes more processing time, but results in better classification accuracy.
* The relative size of the model as a fraction of the largest Mobile Net: 1.0, 0.75, 0.50, or 0.25.

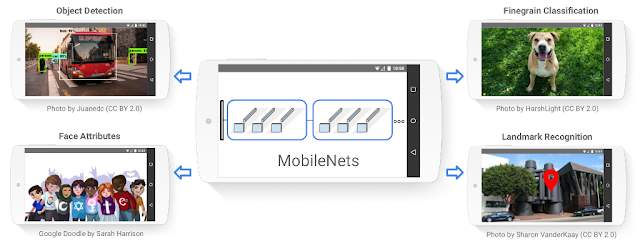
****

Fig-5

**MobileNets Feautures**

The ImageNet project is a large visual database designed for use in visual object recognition software research. More than 14 million images have been hand-annotated by the project to indicate what objects are pictured and in at least one million of the images, bounding boxes are also provided. ImageNet contains more than 20,000 categories with a typical category, such as "balloon" or "strawberry", consisting of several hundred images.

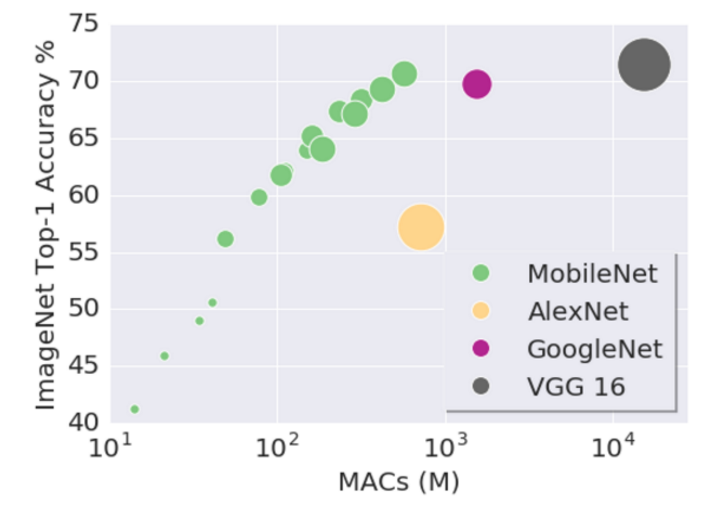
ImageNet models are networks with millions of parameters that can differentiate a large number of classes.

Fig-6

**Accuracy % of different Models**

The graph above shows the first-choice-accuracies of these configurations (y-axis), vs the number of calculations required (x-axis), and the size of the model (circle area).

16 points are shown for MobileNet. For each of the 4 model sizes (circle area in the figure) there is one point for each image resolution setting. The 128px image size models are represented by the lower-left point in each set, while the 224px models are in the upper right.

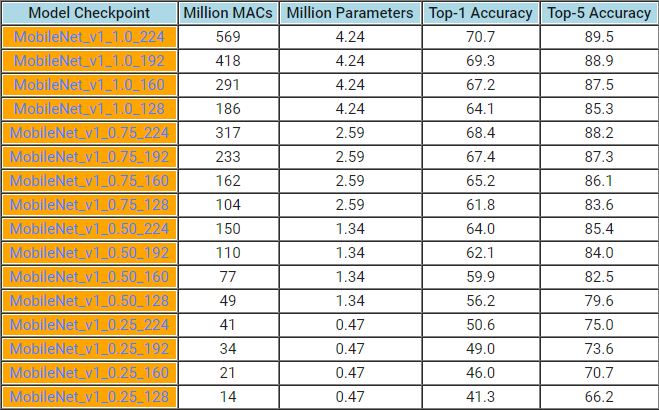


Table-2

**MobileNets Version Analysis**