# Technical Note: Supervised Machine Learning Techniques

Puja Kawale, Software Engineering, San Jose State University, Santa Clara, USA., Dt.10/12/2018, SJSU ID: 012506156, puja.kawale@sjsu.edu

Machine Learning is the discipline of Artificial Intelligence that makes use of statistics and analytics to learning power to the computers from data, without having the need to explicitly make a program to do that. Example, improving the performance of any task. Machine learning is broadly classified into three techniques – Supervised, Unsupervised and Reinforcement techniques.

Supervised Learning is a step involved in data mining which involves learning a function that maps input vector to an output value or the supervisory signal based on example input-output pairs. It usually infers the function from labeled data which come along with the set of training samples. Supervised machine learning mainly includes two types of algorithms, Classification and Regression. Some techniques in this domain consist of Decision Tree, k-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Naïve Bayes Classifier, Linear and Non-Linear Regression. Our intention is to discuss precisely more on supervised machine learning techniques for small farm factor devices like mobile phone.

Small Farm Factor devices: Let us start with SVM, SVM is a classification technique and its main purpose is to develop decision planes or hyper planes in order to classify and segregate all trained samples into two separate classes. This training sample consist of datapoints that form vector. The complete SVM learning process includes initially training a support vector machine following by cross validating the classifier. Mostly applied in gene expression, classification of the text, identifying an image. SVM provide better accuracy and are quick learners. Next is AdaBoost, which is again a classification technique Adaboost trains new model based on the experience with errors of previous model. Basically, it develops a predictor model with the help of prediction algorithm and collects all the errors, it then scans on these errors during the development of next model. This process caters to get the datapoints which are usually hidden and classifier then classifies the test sample accurately. AdaBoost is very popular for its speed, flexibility, effectiveness and versatility. Model is easy to build in this. Another technique in line is Artificial Neural Network (ANN). ANN is similar to the real neural network existing in the human brain. It provides methodology to learn real, discrete, vectorized functions from provided training sample. It is very robust. Applications that use ANN realize speech, handwritten characters and policy to control robots. Classical backpropagation method is most popular ANN method that uses gradient descent to tune network parameters in order to best fit a training set of input and output. Then we have, Gaussian process (GMM), which is a simple regression model. We can define it as a group of arbitrary variables, such that any finite number of them have a joint Gaussian distribution specified by its mean and covariance functions. Issues like predicting prices, weather, tracking can benefit from regression, as they involve estimating values that fall within a continuous range of outputs. Moving on to the next famous and most widely used classification cum regression technique – Decision Tree, lets try to understand its structure. It’s a hierarchical inverse tree like structure. Starts from root node at the top which is best feature to classify and then branches down with conditions to the end nodes or leaf nodes. This technique usually accompanies with problems like – representing the instances by attribute-value, the output function with discrete value, disjunctive descriptions accepted, the training data having errors or missing value. Lastly, we would list down Naïve Bayes Classifier. It classifies a sample based on probability. It first predicts the probability of the sample, then selects the class with highest. It is a generative model. The model with this technique is simple, fast. It handles discrete and streaming data. It is insensitive to nulls or outliers.

The paper that I have referred to understand this have described the application of these techniques in mobile applications like, Pedestrian help app, Fruit ripeness app, Application prediction app, Malware scanning app, house pricing app, Healthcare monitoring app. We conclude that machine learning techniques, especially supervised learning ones are useful in developing mobile apps.

Apart from above, we also have supervised mobile machine learning modelling offered by TensorFlow, TensorFlow lite, iOS Mlkit, Firebase Mlkit.

TensorFlow mobile is one of the best ML framework in today’s date. We can easily develop and train the models or neural networks that solve many big issues like image classification, speech recognition. It provides a library required to run machine learning models useful for mobile apps. [TensorFlow image recognition](https://data-flair.training/blogs/tensorflow-image-recognition/)provides a great variety of samples for detecting the types of objects inside an image. TensorFlow speech recognition is useful for application that are built with speech-driven interface. For them, TensorFlow have neural network that runs on the mobile for a word instead of listening to whole conversation. Gesture recognition in TensorFlow helps control apps using hands or other gestures while analyzing the sensor data. There are other examples like OCR – Optical Character Recognition, Translation, Text Classification, Voice Recognition.

TensorFlow lite of for the mobile phones and embedded devices that have low latency and small binary size. The trained TensorFlow model on the disk will be converted into TensorFlow Lite file format (.tflite) using the TensorFlow Lite converter. Then this model can be used on iOS or Android phones.

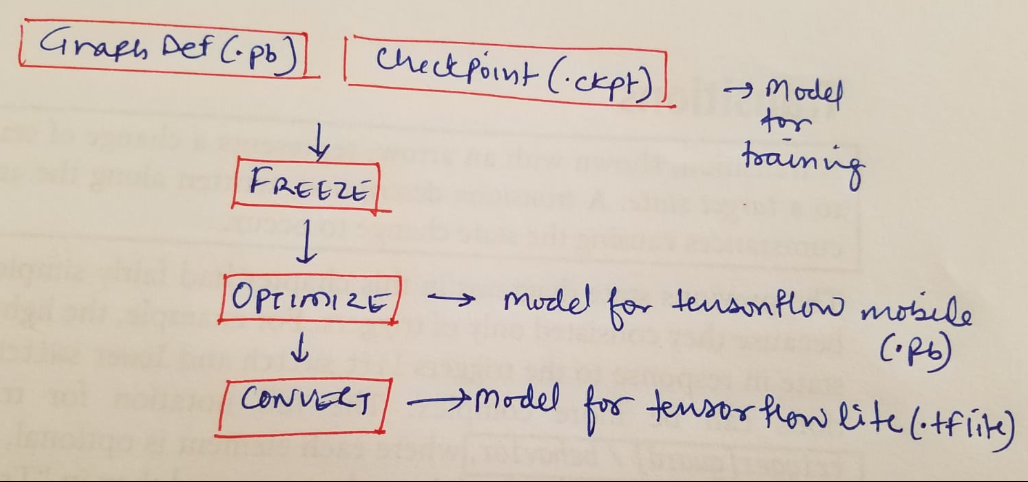


Fig.: TensorFlow Mobile Models

Not to forget ML Kit for Firebase that is making it very simple to apply ML supervised techniques to the mobile applications by expanding the google technologies like Google Cloud version API, Android Neural network API and TensorFlow Lite all inside single sdk. Features available are – Text recognition, face detection, scanning of barcode, image labelling, location detection, model inference.

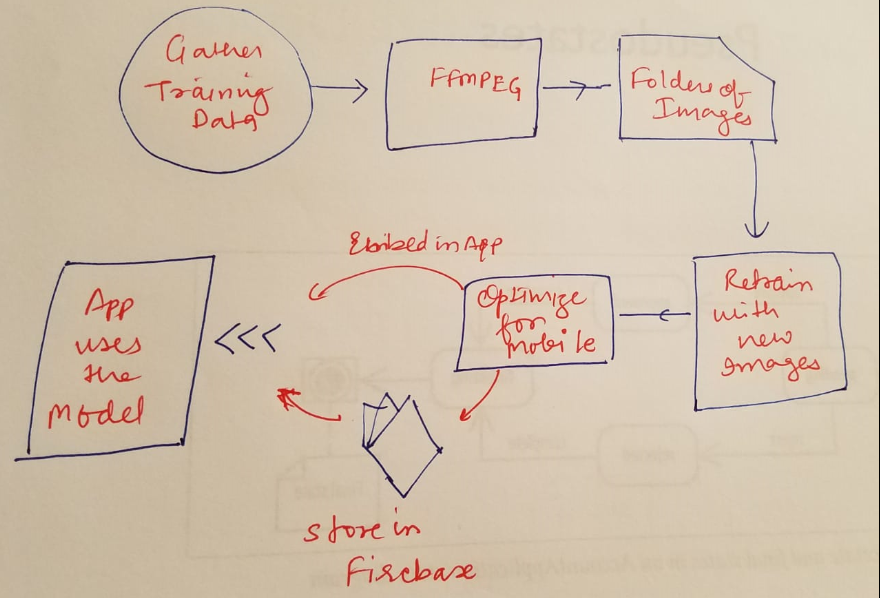


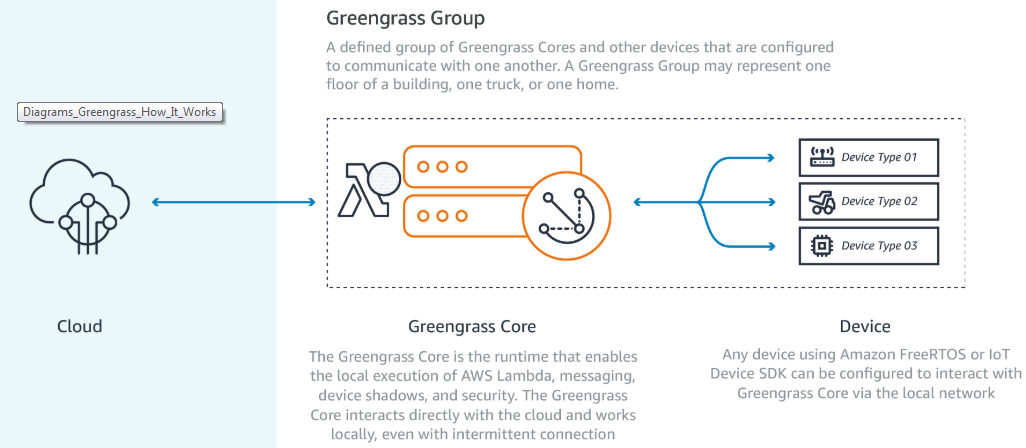
Fig.: Process of developing a model with MLKit

These supervised machine learning techniques can also be used for Embedded systems. There are many optimized libraries that support broad range of machine learning options for embedded system as well. For example, the CMSIS-NN software library provide below functions of neural network:

* Neural Network Convolution Functions
* Neural Network Activation Functions
* Fully-connected Layer Functions
* Neural Network Pooling Functions
* Softmax Functions
* Neural Network Support Functions

Using machine learning to embedded systems would add a level of intelligence in the system which may have been time consuming or difficult to achieve. Usage of ML in microcontroller have improved the overall architecture (Example – Usage of Cortex -M can be used to spot keyword in keyword spotting applications which would use less power as compared to the traditional approach).

Moreover, we have completely cloud supported solutions like AWS Greengrass that cater us with statistical models that can learn with existing data called as training data that is collected from several devices to make decisions regarding the new data called as inference. For example, we can develop a model in Amazon SageMaker for scene detection analysis and then run it locally on a Greengrass enabled security camera device where there is no cloud connectivity to predict and send an alert when an incoming visitor is detected.



*Fig: AWS Greengrass architecture;*

*Source:* [*https://aws.amazon.com/greengrass/*](https://aws.amazon.com/greengrass/)

References:

Books –

<https://www.sciencedirect.com/science/article/abs/pii/S030646031730446X>

Web references –

<https://firebase.google.com/docs/ml-kit/>

<https://developer.apple.com/documentation/coreml>

<https://aws.amazon.com/greengrass/>

<https://www.designnews.com/artificial-intelligence/5-reasons-embedded-developers-should-care-about-machine-learning/26759745459087>

Scholarly articles –

<https://scholar.google.com/scholar?q=smartphone+based+data+mining+for+human+activity+recognition&hl=en&as_sdt=0&as_vis=1&oi=scholart>

<https://pdfs.semanticscholar.org/6050/2b0a78d6d0a8af455af7dd58f37e171ea488.pdf>

Supervised Learning Techniques in

Mobile Device Apps for Androids by Priyanka Basavaraj and Aparna S. Varde\_

Below attached is the above reference:

