**PROJECT SYNOPSIS**

**Predicting Diseases With Edge Intelligence**

Kazaree Basak (504), Rajdeep Das (515), Rajarshi Bhattacharya (537)

M.Sc. Computer Science, Semester 3

Under Guidance of

Dr. Tapalina Bhattasali

St. Xavier's College

[Autonomous], Kolkata

**INTRODUCTION**

**OBJECTIVE**

Our objectives for this project will be-

* To detect and track the person with any kind of disease.
* Keep patient data records for analysis and decision making using edge intelligence.
* To apply edge intelligence in surveillance for faster detection.

So our main motive will be to create a generic DeepNet model which can predict any kind of diseases from the dataset provided. Then we will simulate this DeepNet model at the Edge part of the network using a Cloud simulator. The data will be provided locally and the model will predict the result based upon the input provided.

**SURVEY OF TECHNOLOGY**

An Edge device is any piece of hardware that controls data flow at the boundary between two networks. Edge Computing allows computing resources and application services to be distributed along the communication path, via decentralized computing infrastructure.

When edge computing is merged with machine learning, we get edge intelligence. Edge Intelligence deals with supporting the intelligence acquired through data at a local level.

**Merits of Edge Intelligence**

1. By processing the data closer to the source can greatly reduce latency and they can be quickly analysed and decisions can be made based on that faster.
2. Since more data is processed on local devices rather than sending it back to a central data center, the amount of data at risk at a time is reduced.
3. By combining colocation services with regional edge computing data centers, organizations can expand their edge network reach quickly and cost-effectively.
4. Scalability of edge computing makes it incredibly versatile. Edge data centers allow them to service end users efficiently with little physical distance or latency. This is especially valuable for content providers looking to deliver uninterrupted streaming services.
5. Even if a nearby data center is not available, IoT edge computing devices will continue to operate effectively on their own because they handle vital processing functions natively.

**Deep Learning** is a collection of algorithms inspired by the workings of the human brain in processing data and creating patterns for use in decision making, which are expanding and improving on the idea of a single model architecture called **Artificial Neural Network.**.

**Reasons for using Deep Learning in Edge Intelligence**

1. Deep Learning algorithms, having high resource demanding workload naturally suits for edge computing.
2. Low latency and low cost computing- Since DL services are deployed closed to the requesting user and cloud participates when required.

**SYSTEM DESIGN**

**RESEARCH METHODOLOGY**

**DeepNet Model**

def \_\_init\_\_(self, shape\_x, shape\_y, opt, l, neurons, mname):

self.model=keras.Sequential([keras.layers.Flatten(input\_shape=(shape\_x,shape\_y))])

for n in neurons:

self.model.add(keras.layers.Dense(n, activation='relu'))

self.model.add(keras.layers.Dense(shape\_y, activation='sigmoid'))

self.model.compile(optimizer=opt, loss=l, metrics=['accuracy'])

self.mname=mname

The neurons in input and the hidden layers are considered as per the features in the following dataset as the accuracy of the model depends on the number of features provided and as well as the number of neurons being considered. For example, for the Diabetes dataset we have used the following,

dlmod=dn.DeepNet(8, 1,'adam','binary\_crossentropy',[12,8],"Diabetes")

We have considered 80 percent of the dataset as training dataset and rest as test dataset.

We have considered the epoch as 250 and batch size as 10.

Epoch means one pass through all of the rows in the training dataset.

Batch Size means one or more samples considered by the model before weights are updated.

For optimizer we have taken ‘adam’ and loss function as ‘binary\_crossentrophy’.

The reason to choose ‘adam’ as an optimizer as this is a popular version of gradient descent because it automatically tunes itself and gives good results in a wide range of problems. and for loss function ‘binary\_crossentrophy’ is selected for its binary classification as the target values are in binary 0 and 1 with crossentrophy as loss argument.

**HARDWARE & SOFTWARE SPECIFICATION**

Software - Python including Tensorflow,matplotlib,pandas,numpy,sklearn packages and Cloud Simulator.

Hardware - Laptop/Desktop.

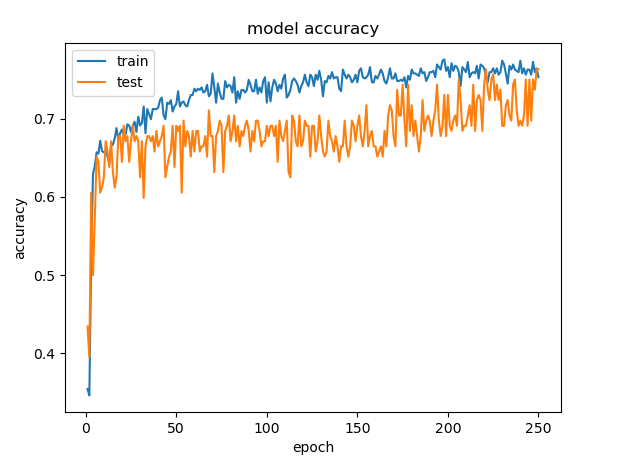
**RESULTS & DISCUSSION**

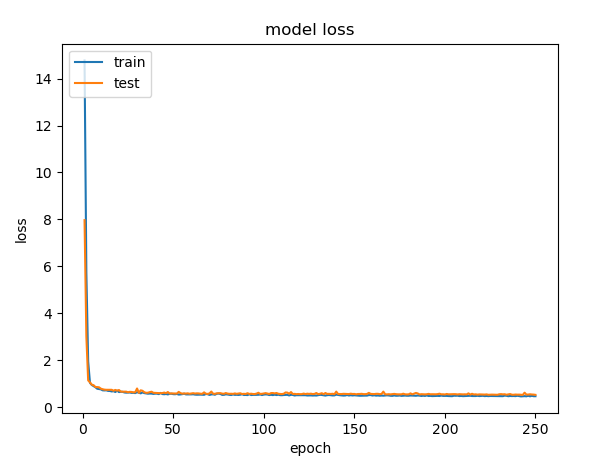
The Results we have got using the DeepNet model using Heart Dataset is approx. 80% and Diabetes Dataset is approx 75%.

Test Results for Diabetes

=========================

5/5 [==============================] - 0s 1ms/step - loss: 0.5352 - accuracy: 0.7632

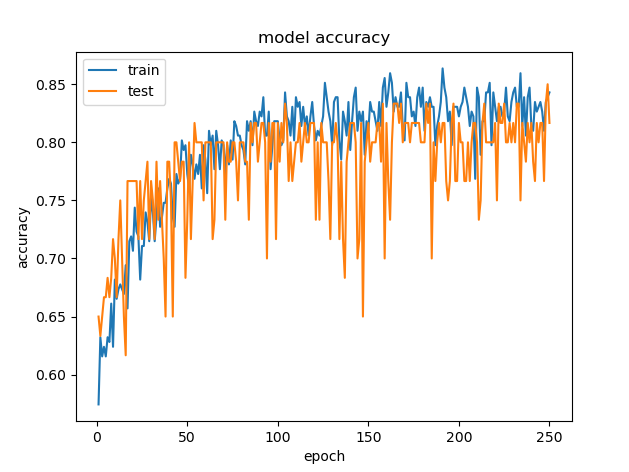


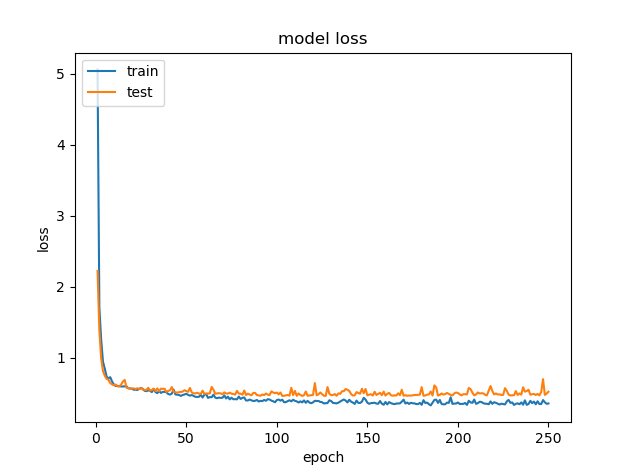


Test Results for Heart

=========================

2/2 [==============================] - 0s 0s/step - loss: 0.5233 - accuracy: 0.8167





The DeepNet model predicts the results for the Heart dataset in 0.060 secs for the testing data and in 0.125secs for the Diabetes dataset’s testing data.

**CURRENT STATUS OF DEVELOPMENT**

The phase involving creating a DeepNet model is complete. The next phase involving simulating the model on the edge needs to be done.

**INNOVATIVENESS & USEFULNESS**

This project aims to generate the idea of a generic deep learning model which can predict any disease based upon the dataset provided. The further usefulness that we did incorporate is that the whole computation is done within the edge which can be more faster and effective than the traditional cloud computing based systems.

**FUTURE PROPOSED WORK**

We have so far implemented a generic DeepNet model which we can modify to work with every kind of data be it with diabetes or heart dataset etc. Further, we will simulate this model at the edge part of the network using CloudSim simulator.