Question Answering System

Chanchal Suman & Himani Srivastava

14/04/2017

1 Abstract of the project

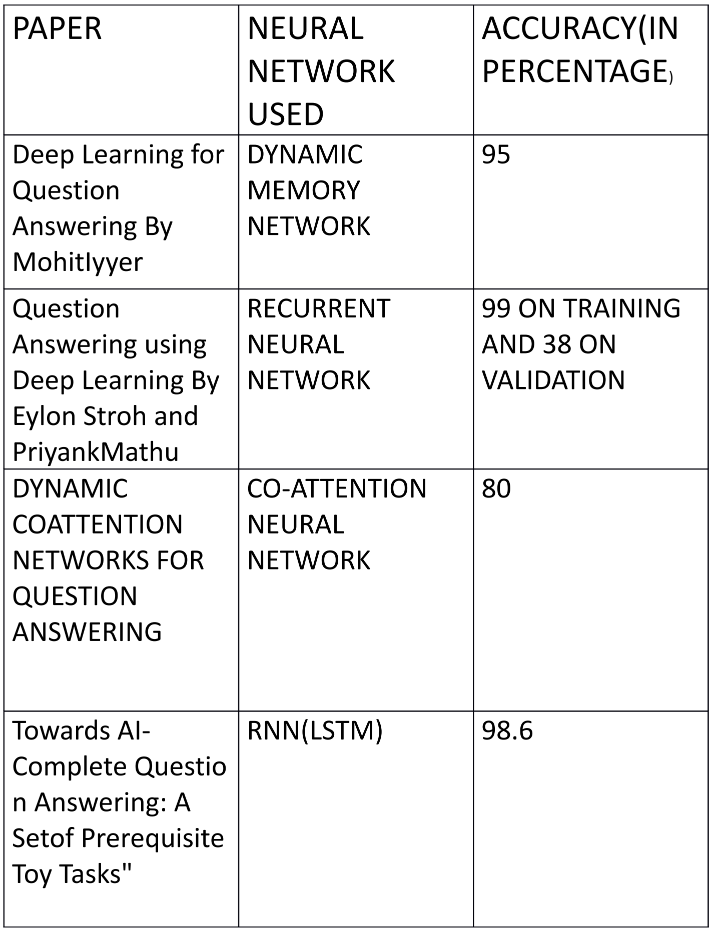
This project is based on the question answering system . we have implemented a text based Question Answering System , We have trained our network on a given paragraph , questions are given in the last of the paragraph , answer is just beside the question and then , the supporting facts. neural network is trained on the story and then tested on it . the system we developed has 93 percent accuracy. we have used sigmoid activation function and recurrent neural network (LSTM).

2 Introduction

We have implemented an Closed Domain Question Answering System , i.e Questions from the given paragraph will be answered only , We have used BABI dataset that is used by facebook , and trained and tested our neural network on this dataset only.

2.1 Literature survey

Papers that we have read for this projects are based on different different approaches and dataset but one common approach all papers have followed that they train network on a paragraph and retrieve answers to the questions.. Papers are:



PAPER	NEURAL NETWORK USED	ACCURACY(IN PERCENTAGE)
Deep Learning for Question Answering By MohitIyyer	DYNAMIC MEMORY NETWORK	95
Question Answering using Deep Learning By Eylon Stroh and PriyankMathu	RECURRENT NEURAL NETWORK	99 ON TRAINING AND 38 ON VALIDATION
DYNAMIC COATTENTION NETWORKS FOR QUESTION ANSWERING	CO-ATTENTION NEURAL NETWORK	80
Towards AI-Complete Questio n Answering: A Setof Prerequisite Toy Tasks"	RNN(LSTM)	98.6

fig:literature survey

We have followed paper 4 , the accuracy that they got are 98.6 accuracy on task 'single supporting fact 10k' at 700 epochs, using RELU as a activation function.

3 Resources

We have used bABi dataset for our training and testing, It is a standard dataset that is used by facebook for Question And answering System, Format for bAbi dataset are:

1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary? bathroom 1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel? hallway 4

We have used only Single supporting fact for our System from bAbi dataset , we have trained our network on 10000 training dataset and tested on 1000 test data set.

3.1 Work done

- Description of the data

  We have used bAbi dataset that contain different supporting facts like ,Single Supporting fact ,2 supporting facts.We have implemented our System on Single Supporting fact by following above papers.

- Exploration of different neural networks and observation from the same

  We have used Recurrent Neural network for our task , In one of the paper it was written that if we use Dynamic coattention network then accuracy will be higher but that project is under development and other paper that we have followed are based on Recurrent Neural Network.

  We have used Sigmoid, RELU ,TanH as a Activation function , Dropout as 0.7 and 0.3 and observed that using RELU we got around 98 PERCENT accuracy at Epoch 700 only but we didn't get any over fitting case with RELU , when we tried with Sigmoid we got an accuracy of 92PERCENT at Epoch 1000 , and after 1000 we got Over fitting case . TanH gives worst result among all three maximum accuracy we got with tanH is 52PERCENT only till Epoch 800.

- Error plot for validation set

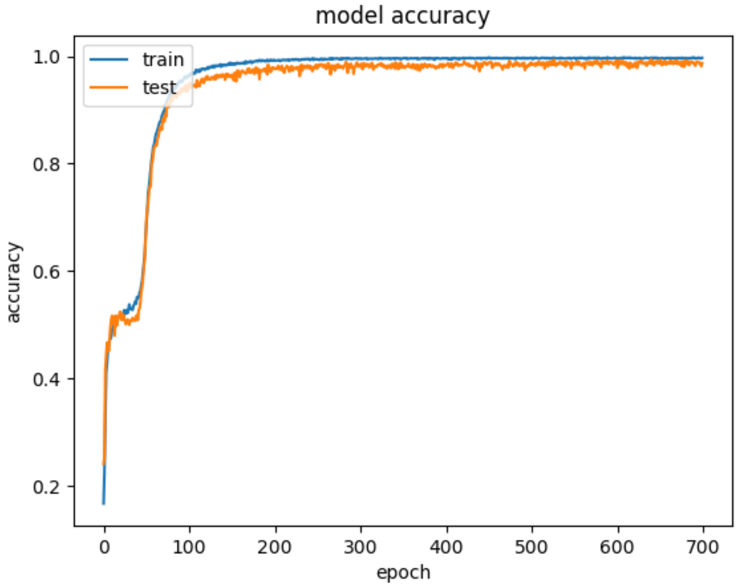


Fig : - Accuracy Vs Epoch With Relu , Dropout=0.7 , Epoch 700

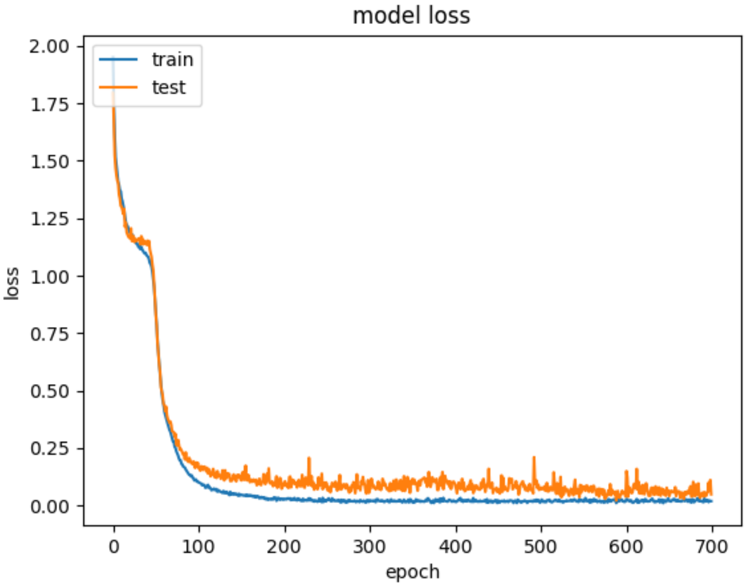


Fig : - Loss Vs Epoch With Relu , Dropout=0.7 , Epoch 700

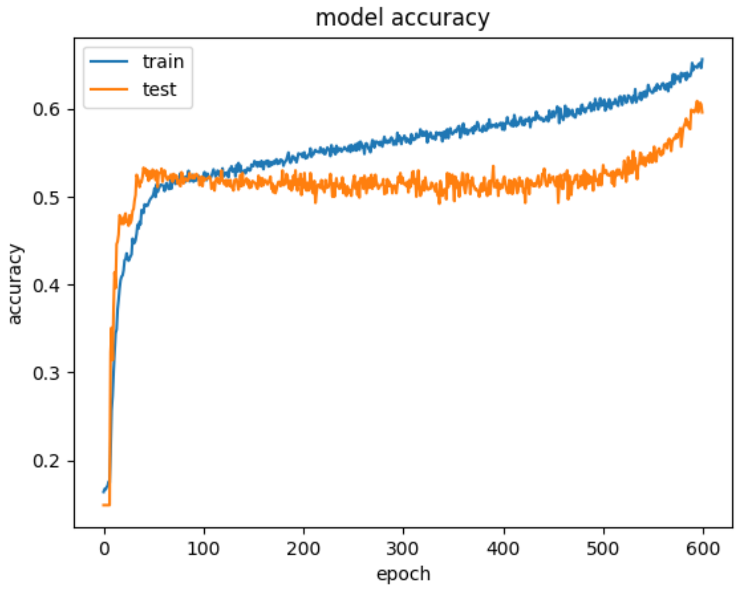


Fig : - Accuracy Vs Epoch With Sigmoid , Dropout=0.7 , Epoch 600

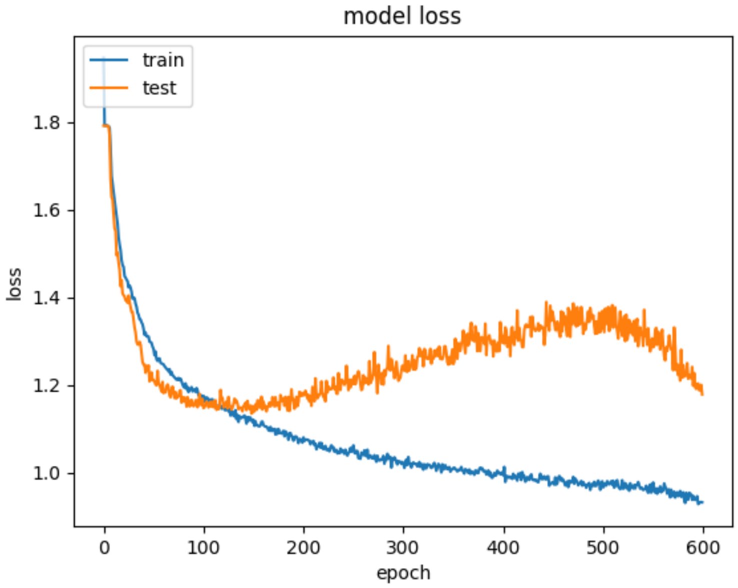


fig:loss vs epoch with sigmoid dropout 0.7 , epoch -600

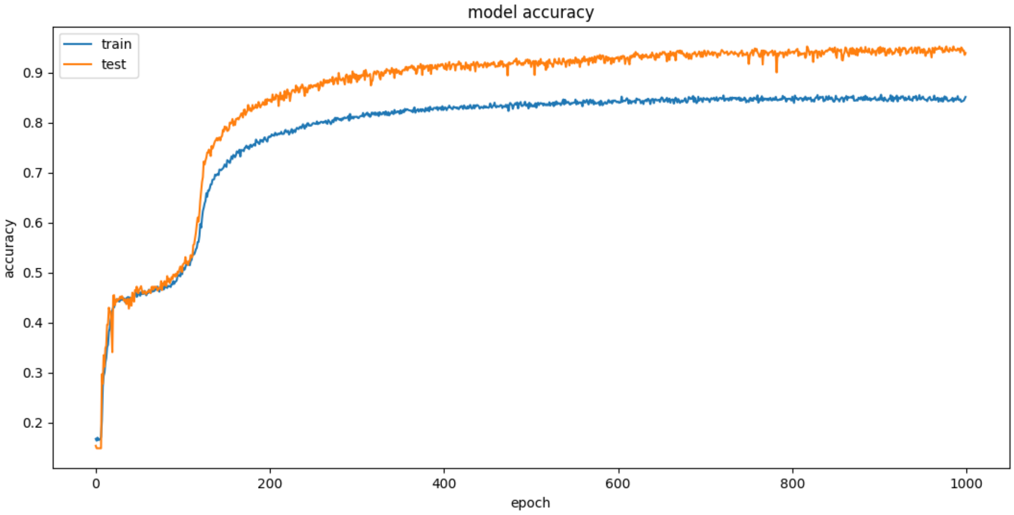


fig : - Accuracy Vs Epoch With Sigmoid , Dropout0.7 , Epoch1000

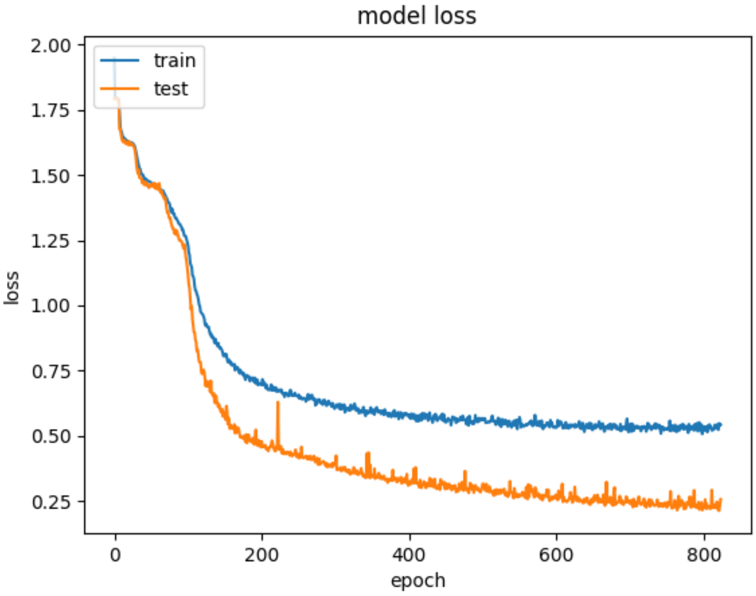


fig: - Loss Vs Epoch With Sigmoid , Dropout0.7 , Epoch1000

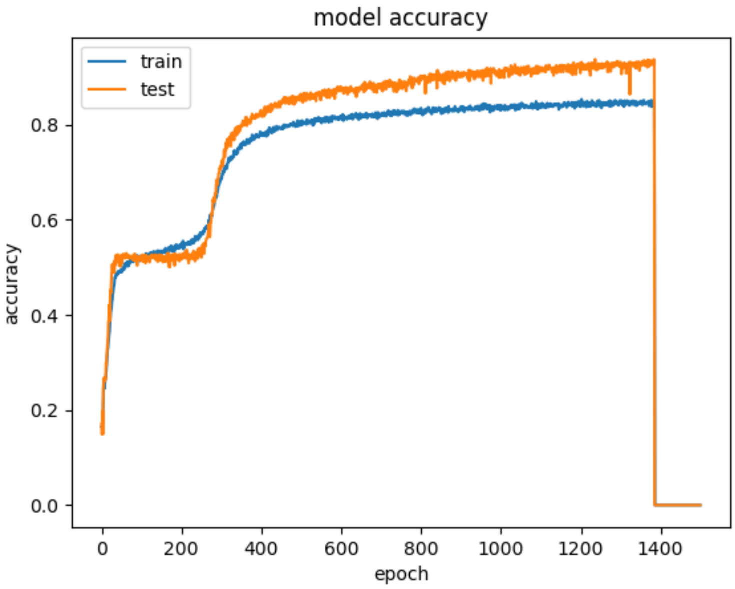


fig :- Accuracy vs epoch at epoch 1500 sigmoid dropout0.3(overfitting from 1381 epoch accuracy : 93.30 percent)

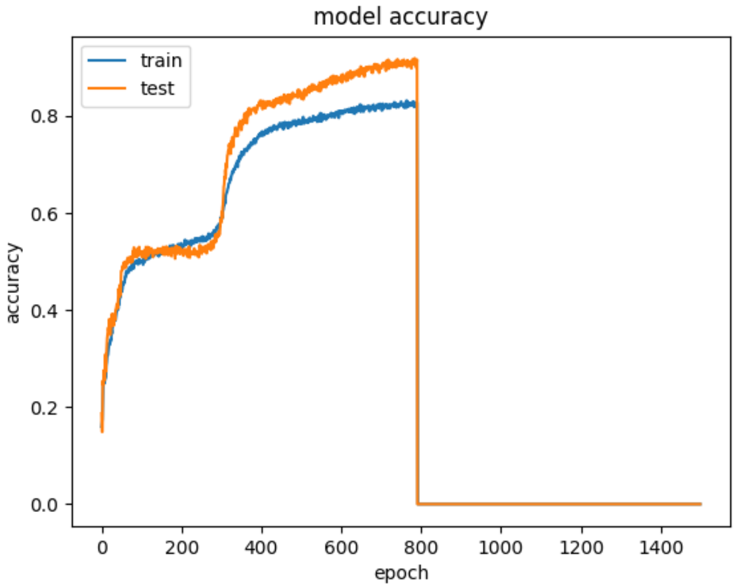


fig : Accuracy vs epoch at epoch 1500 sigmoid dropout0.7

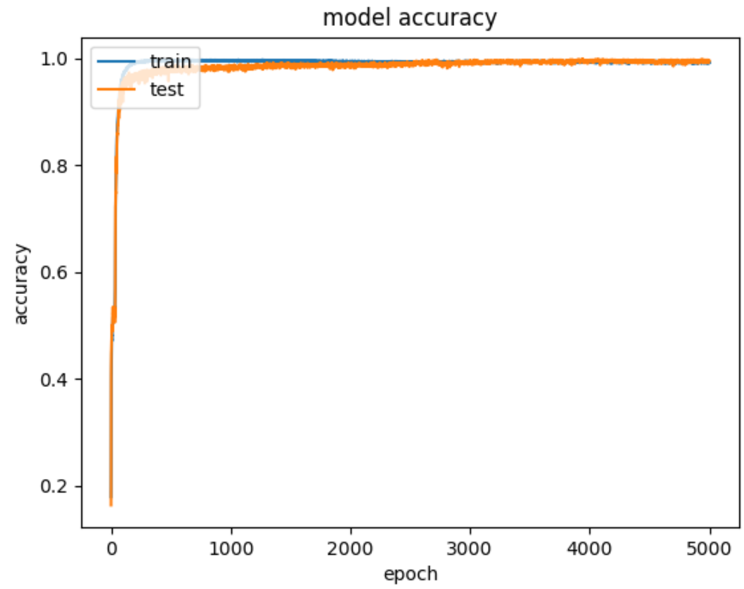


fig: Accuracy vs epoch graph of activation as Relu epoch 5000 dropout 0.7

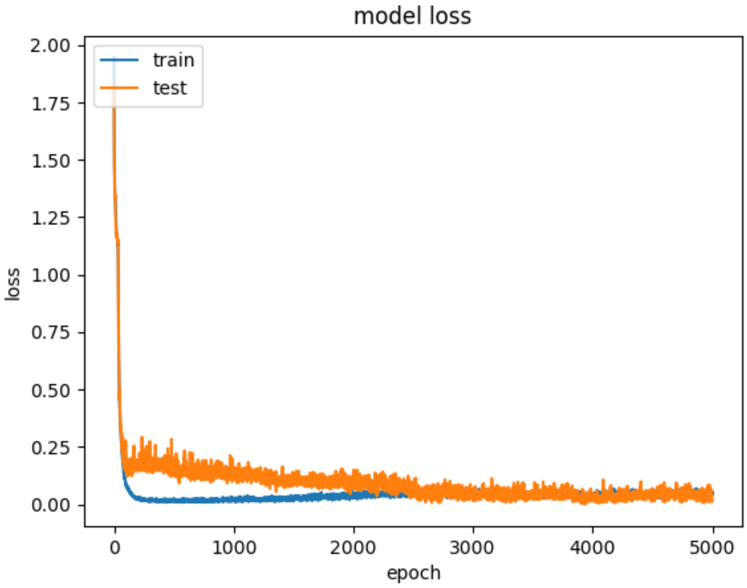
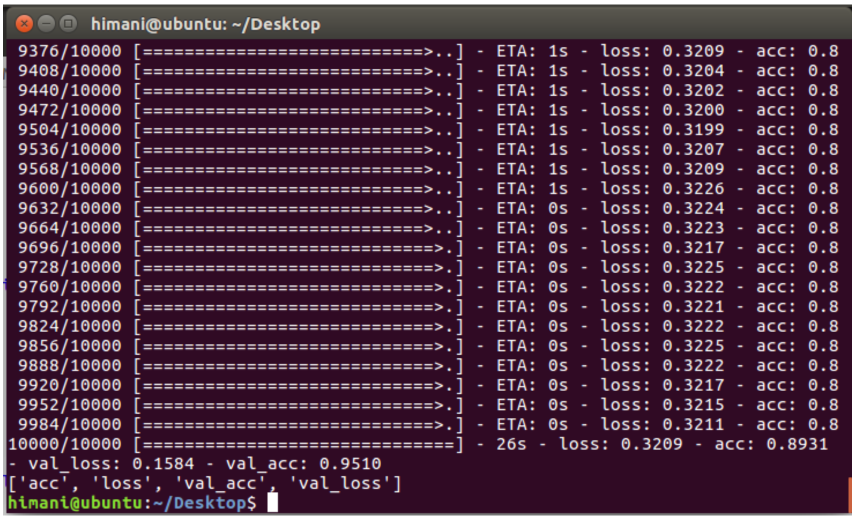


fig:Loss vs epoch graph of activation function as Relu , epoch 5000 , dropout 0.7

- Final architecture

Final Accuracy that we got is at Sigmoid Activation function, rms prop optimizer, Dropout=0.7 and Epoch 1000 , Accuracy and Loss Values at these Values are :



```
 9376/10000 [===========================>..] - ETA: 1s - loss: 0.3209 - acc: 0.8
 9408/10000 [===========================>..] - ETA: 1s - loss: 0.3204 - acc: 0.8
 9440/10000 [===========================>..] - ETA: 1s - loss: 0.3202 - acc: 0.8
 9472/10000 [===========================>..] - ETA: 1s - loss: 0.3200 - acc: 0.8
 9504/10000 [===========================>..] - ETA: 1s - loss: 0.3199 - acc: 0.8
 9536/10000 [===========================>..] - ETA: 1s - loss: 0.3207 - acc: 0.8
 9568/10000 [===========================>..] - ETA: 1s - loss: 0.3209 - acc: 0.8
 9600/10000 [===========================>..] - ETA: 1s - loss: 0.3226 - acc: 0.8
 9632/10000 [===========================>..] - ETA: 0s - loss: 0.3224 - acc: 0.8
 9664/10000 [===========================>..] - ETA: 0s - loss: 0.3223 - acc: 0.8
 9696/10000 [============================>.] - ETA: 0s - loss: 0.3217 - acc: 0.8
 9728/10000 [============================>.] - ETA: 0s - loss: 0.3225 - acc: 0.8
 9760/10000 [============================>.] - ETA: 0s - loss: 0.3222 - acc: 0.8
 9792/10000 [============================>.] - ETA: 0s - loss: 0.3221 - acc: 0.8
 9824/10000 [============================>.] - ETA: 0s - loss: 0.3222 - acc: 0.8
 9856/10000 [============================>.] - ETA: 0s - loss: 0.3225 - acc: 0.8
 9888/10000 [============================>.] - ETA: 0s - loss: 0.3222 - acc: 0.8
 9920/10000 [============================>.] - ETA: 0s - loss: 0.3217 - acc: 0.8
 9952/10000 [============================>.] - ETA: 0s - loss: 0.3215 - acc: 0.8
 9984/10000 [============================>.] - ETA: 0s - loss: 0.3211 - acc: 0.8
10000/10000 [==============================] - 26s - loss: 0.3209 - acc: 0.8931
- val_loss: 0.1584 - val_acc: 0.9510
['acc', 'loss', 'val_acc', 'val_loss']
himani@ubuntu:~/Desktop$ 
```

fig :- output screenshot

- Results from different optimization techniques.

We have tried with two optimization techniques Adam and rms prop and observed that using rmsprop is much beneficial and give better accuracy.

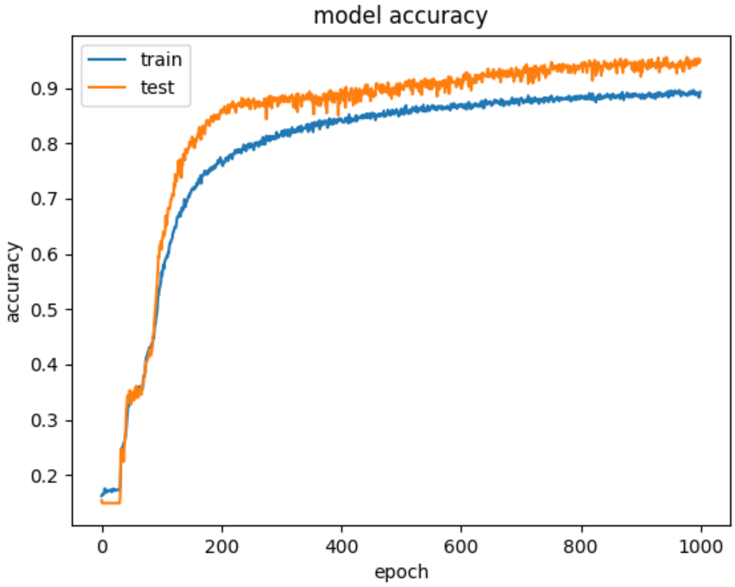


fig : - Accuracy vs epoch sigmoid optimizer - adam , epoch 1000 ,dropout 0.7

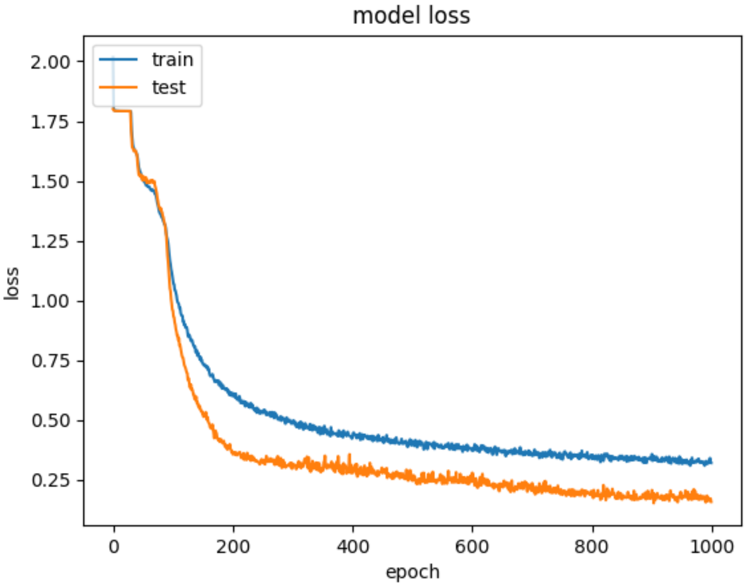
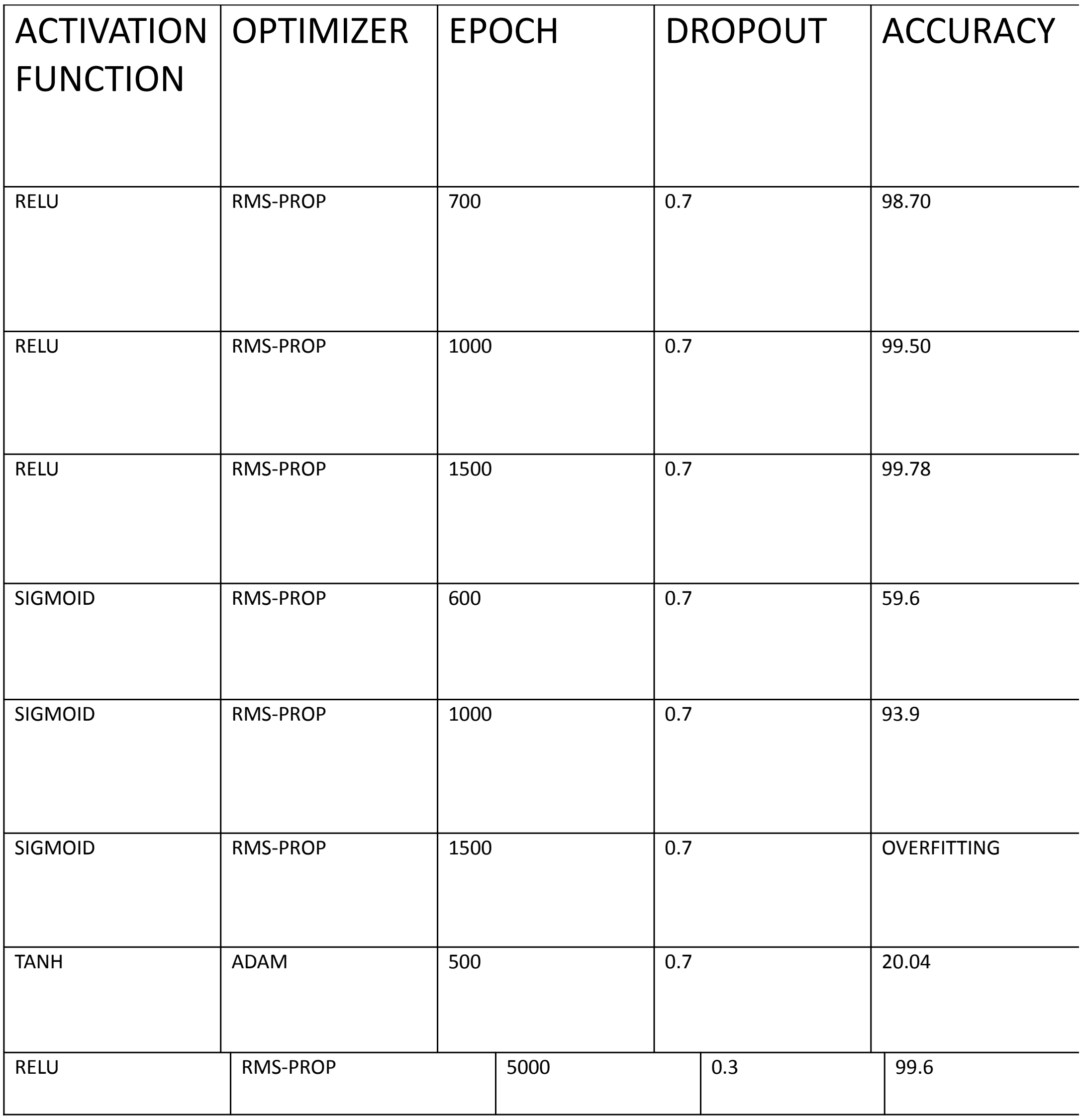


fig: loss vs epoch at epoch 1000 , optimizer - adam , dropout 0.7 , activation - sigmoid



ACTIVATION FUNCTION	OPTIMIZER	EPOCH	DROPOUT	ACCURACY
RELU	RMS-PROP	700	0.7	98.70
RELU	RMS-PROP	1000	0.7	99.50
RELU	RMS-PROP	1500	0.7	99.78
SIGMOID	RMS-PROP	600	0.7	59.6
SIGMOID	RMS-PROP	1000	0.7	93.9
SIGMOID	RMS-PROP	1500	0.7	OVERFITTING
TANH	ADAM	500	0.7	20.04
RELU	RMS-PROP	5000	0.3	99.6

Table :- observation

- github location for Our code *https : //github.com/HimaniChanchal/QuestionAnsweringBasedSystem*

3.2 Future work

In future we will try to implement our system with 2 supporting facts and will try to apply DYnamic Coattention network to get higher accuracy.