

SRM INSTITUTE OF SCIENCE & TECHNOLOGY DEPARTMENT OF NETWORKING & COMMUNICATIONS **18CSC305J-ARTIFICIAL INTELLIGENCE**

SEMESTER – 6 BATCH-2

|  |  |
| --- | --- |
| **REGISTRATION NUMBER** | **RA1911003011043** |
| **NAME** | **RAMNATH** |

# **B.Tech- CSE / CC, Third Year (Section: H2)**

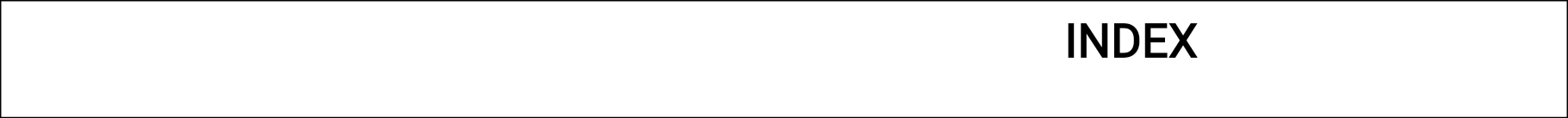
**Faculty In charge: Dr. S. Prabakeran, B.Tech, M.E, PhD**

**Assistant Professor**

**School of Computing - Department of**

**Networking and Communications**

**Year 2021-2022 / Even Semester**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ex No** | **DATE** | **Title** | **Page No** | **Marks** |
| 11 | 6/04/02 | **Deep Learning** |  |  |

## **Experiment No: 11**

**Date : 06-04-22**

**Deep Learning**

**AIM:** Implementation of Deep Learning

**CODE:**

**import** tensorflow **as** tf

**from** utils.DL\_utils **import** myCallback, build\_model, compile\_train\_model, plot\_loss\_acc

**from** itertools **import** product

accuracy\_desired **=** [0.85,0.9,0.95]

num\_neurons **=** [16,32,64,128]

cases **=** list(product(accuracy\_desired,num\_neurons))

print("So, the cases we are considering are as follows...\n")

**for** i,c **in** enumerate(cases):

print("Accuracy target {}, number of neurons: {}"**.**format(c[0],c[1]))

**if** (i**+**1)**%4**==0 and (i+1)!=len(cases):

print("-"**\***50)

**for** c **in** cases:

*# Create a mycallback class with the specific accuracy target*

callbacks **=** myCallback(c[0], print\_msg**=False**)

*# Build a model with a specific number of neurons*

model **=** build\_model(num\_layers**=**1,architecture**=**[c[1]])

*# Compile and train the model passing on the callback class,choose suitable batch size and a max epoch limit*

model **=** compile\_train\_model(model, x\_train,y\_train,callbacks**=**callbacks,

batch\_size**=**32,epochs**=**30)

*# Construct a suitable title string for displaying the results properly*

title **=** "Loss and accuracy over the epochs for\naccuracy threshold \

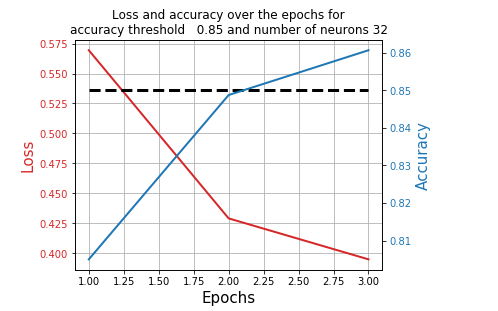
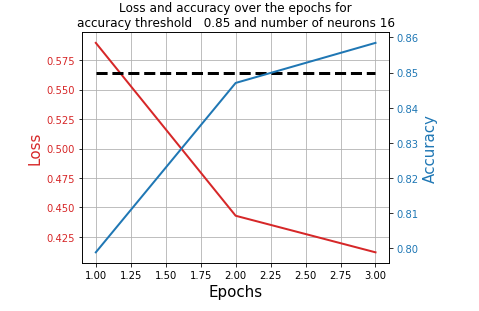
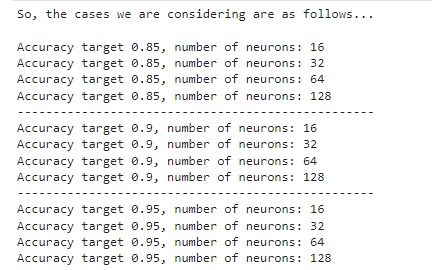
{} and number of neurons {}"**.**format(c[0],c[1])

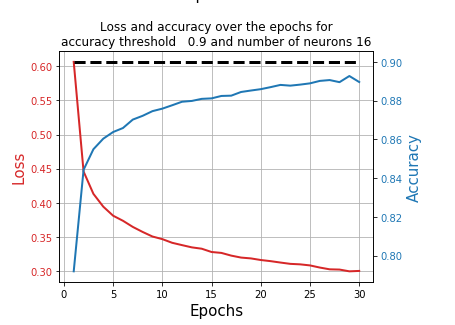
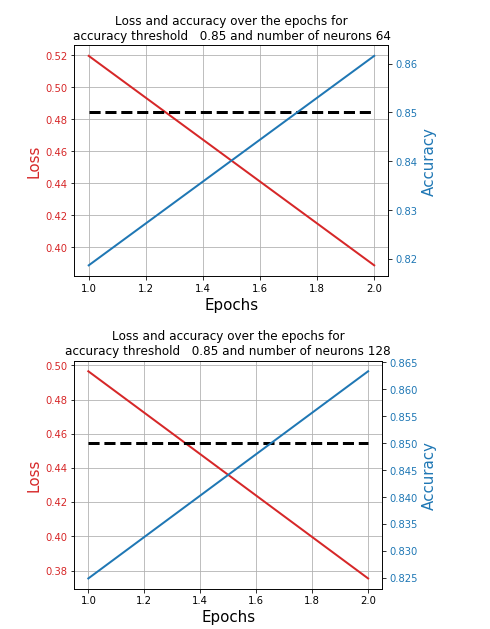
*# Use the plotting utility function, pass on the accuracy target,*

*# trained model, and the custom title string*

plot\_loss\_acc(model,target\_acc**=**c[0],title**=**title)

**OUTPUT:**

****

****

**RESULT:** Deep Learning Model Implemented