IMPLEMENTING THE MEDICAL RESEARCH PAPER BY USING ARTIFICIAL NEURAL NETWORK

Analysis of Cardiotocogram Data for Fetal Distress Determination

Although the research uses the decision tree technique in implementing the above mentioned we will be using deep learning model in predicting the outcome...

Sample Code For Study:

```
 library(keras)
## read the data by using file.choose ##
data <- read.csv(file.choose(),header=T)
str(data) ## to see the structure of the data
###############################################################################
#
## changing the data into matrix ##
data <- as.matrix(data)
dimnames(data) <- NULL # to remove any default names from the data

###############################################################################
#
###### NORMALIZATION OF THE DATA SET ######
## this is done in order to make data lie on between '0' and '1'##
data[,1:21] <- normalize(data[,1:21])
data[,22] <- as.numeric(data[,22])-1
###############################################################################
#
###### DATA PARTITIONING ######
set.seed(1234)
ind <- sample(2,nrow(data), replace=T,prob=c(0.7,0.3)) ## splitting the data into '1' and '2' with weights into 0.7 and 0.3 or we say 70% and 30% ##
training <- data[ind==1, 1:21]
testing  <- data[ind==2, 1:21]
trainingtarget <- data[ind==1, 22]
testingtarget  <- data[ind==2, 22]
###############################################################################
#
###### ONE HOT ENCODING ######
## classification of classes into 0 1 2 3 ......  using 0 and 1 which the class
## lying within that category is given 1 otherwise 0 ##
trainlabel <- to_categorical(trainingtarget)
testlabel  <- to_categorical(testingtarget)
###############################################################################
#
###### SEQUENTIAL MODELLING ######
```

```
model <- keras_model_sequential()
model %>% # piped function --- takes information from left side and passes it onto the right
side
  layer_dense(units=8, activation='relu',input_shape = c(21)) %>%
  layer_dense(units=8, activation='relu')%>%
  layer_dense(units=8, activation='relu')%>%
  layer_dense(units=3, activation='softmax')
summary(model)
##################################################################################
#
###### COMPILING THE MODEL ######
model %>%
        compile(loss='categorical_crossentropy',
                optimizer='adam',
                metrics='accuracy')
##################################################################################
#
###### FITTING THE MODEL FOR LEARNING ######
history <- model%>%
              fit(training,
                  trainlabel,
                   epoch=10,
                   batch_size=32,
                   validation_split=0.2)

##################################################################################
#
##MODEL EVALUATION USING TEST DATA##
test_model <- model %>% evaluate(testing,testlabel)
##################################################################################
#
##CONFUSION MATRIX ANALYSIS OF TEST DATA##
prob <- model %>% predict_proba(testing)
pred <- model %>% predict_classes(testing)

confusion_matrix <- table(Predicted = pred , Actual = testingtarget)

cbind(prob , pred , testingtarget)

##################################################################################
##
##FINE TUNE THE MODEL##
test_model
confusion_matrix
```

Required Outputs

1) Structure of the dataset :

   Out of these 22 observational variables we have 21 numerical data and the last one (22^{nd}) as the NSP or categorical data which will be crucial for classifying the test data

```
'data.frame':  2126 obs. of  22 variables:
 $ LB      : int  120 132 133 134 132 134 134 122 122 122 ...
 $ AC      : num  0 0.00638 0.00332 0.00256 0.00651 ...
 $ FM      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ UC      : num  0 0.00638 0.00831 0.00768 0.00814 ...
 $ DL      : num  0 0.00319 0.00332 0.00256 0 ...
 $ DS      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DP      : num  0 0 0 0 0 ...
 $ ASTV    : int  73 17 16 16 16 26 29 83 84 86 ...
 $ MSTV    : num  0.5 2.1 2.1 2.4 2.4 5.9 6.3 0.5 0.5 0.3 ...
 $ ALTV    : int  43 0 0 0 0 0 0 6 5 6 ...
 $ MLTV    : num  2.4 10.4 13.4 23 19.9 0 0 15.6 13.6 10.6 ...
 $ Width   : int  64 130 130 117 117 150 150 68 68 68 ...
 $ Min     : int  62 68 68 53 53 50 50 62 62 62 ...
 $ Max     : int  126 198 198 170 170 200 200 130 130 130 ...
 $ Nmax    : int  2 6 5 11 9 5 6 0 0 1 ...
 $ Nzeros  : int  0 1 1 0 0 3 3 0 0 0 ...
 $ Mode    : int  120 141 141 137 137 76 71 122 122 122 ...
 $ Mean    : int  137 136 135 134 136 107 107 122 122 122 ...
 $ Median  : int  121 140 138 137 138 107 106 123 123 123 ...
 $ Variance: int  73 12 13 13 11 170 215 3 3 1 ...
 $ Tendency: int  1 0 0 1 1 0 0 1 1 1 ...
 $ NSP     : int  2 1 1 1 1 3 3 3 3 3 ...
```

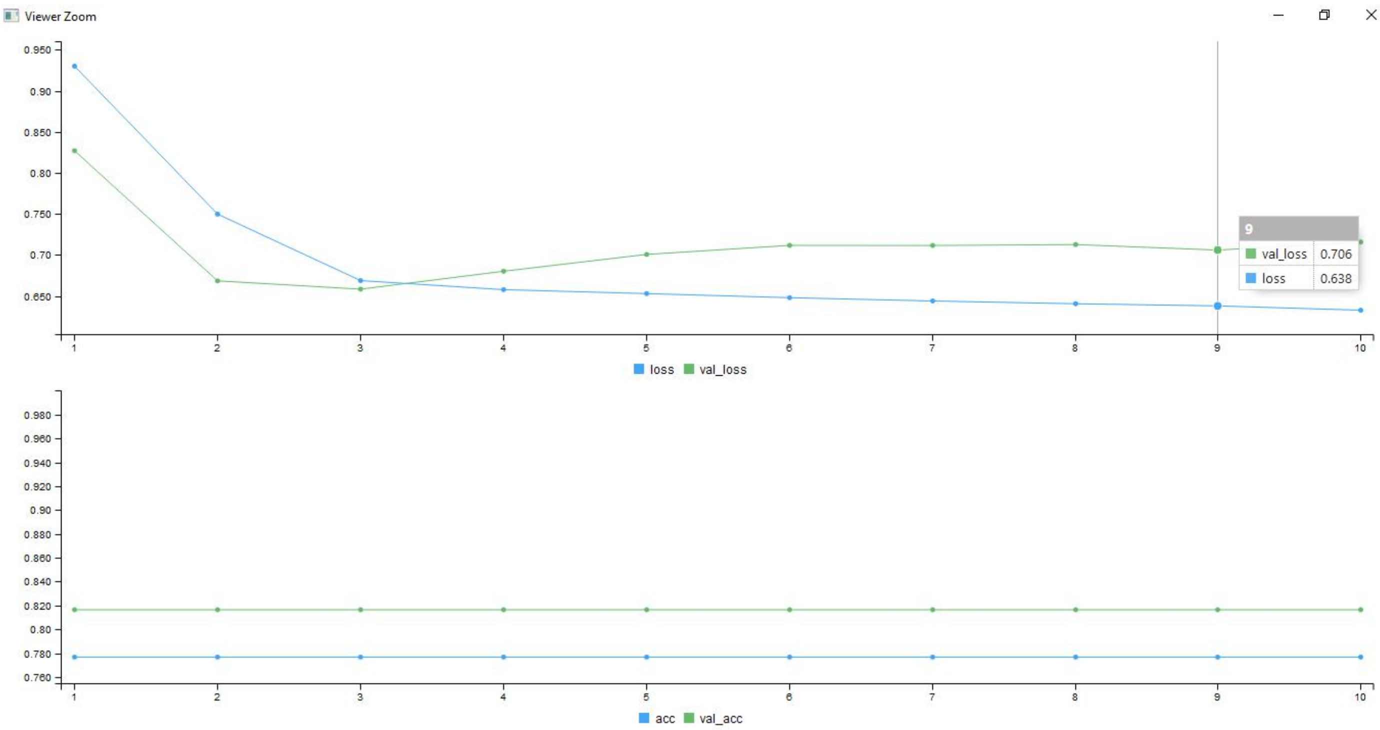
2) Structure of the neural network model

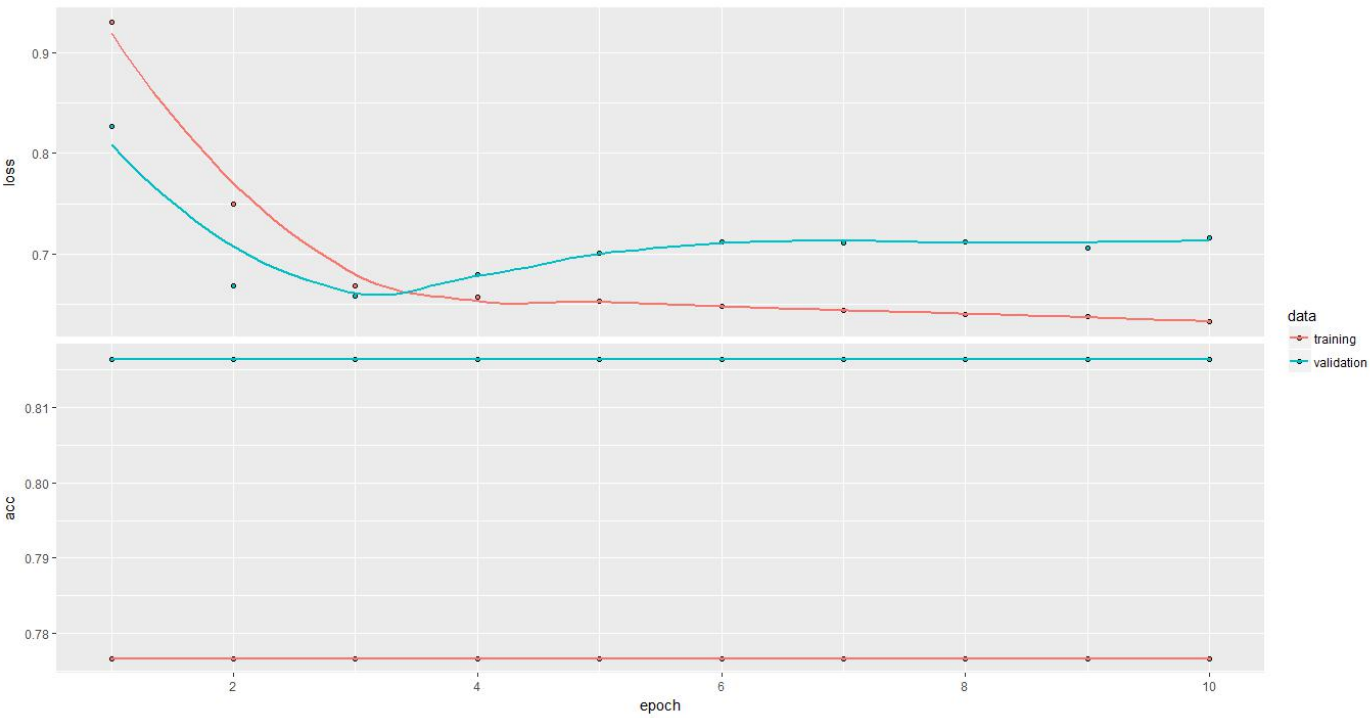
```
________________________________________________________________________________________________________
Layer (type)                                               Output Shape
          Param #
========================================================================================================
dense_1 (Dense)                                            (None, 8)
          176
________________________________________________________________________________________________________
dense_2 (Dense)                                            (None, 8)
          72
________________________________________________________________________________________________________
dense_3 (Dense)                                            (None, 8)
          72
________________________________________________________________________________________________________
dense_4 (Dense)                                            (None, 3)
          27
========================================================================================================
Total params: 347
Trainable params: 347
Non-trainable params: 0
________________________________________________________________________________________________________
```

3) Hereon we will be seeing the output of the trained network and its output on various epochs –some considerations are as follows : activation function at the hidden layer is ***ReLu*** , activation function at the output is ***Sigmoid*** , optimizer is ***ADAM*** and we have loss as ***categorical cross-entropy***

a) Number Of Epochs = 10

Training Accuracy and Loss :-





Testing Data Output :-

```
pred testingtarget // this is basically the classification of the testing data into ca
tegories namely N(normal) , S(suspect) , P(pathological)//

  [1,] 0.8192502 0.1345060 0.04624378    0             0
  [2,] 0.7961514 0.1490243 0.05482430    0             0
  [3,] 0.8167875 0.1361584 0.04705412    0             0
  [4,] 0.6575967 0.2288589 0.11354441    0             2
  [5,] 0.6551822 0.2300247 0.11479316    0             2
  [6,] 0.8356651 0.1244439 0.03989100    0             1
  [7,] 0.8278462 0.1293107 0.04284315    0             0
  [8,] 0.8165976 0.1363321 0.04707028    0             0
  [9,] 0.8061484 0.1429762 0.05087535    0             0
 [10,] 0.8477290 0.1165817 0.03568922    0             0
 [11,] 0.8510496 0.1150622 0.03388818    0             1
 [12,] 0.7630237 0.1695594 0.06741689    0             0
 [13,] 0.8174640 0.1360677 0.04646830    0             0
 [14,] 0.7927164 0.1515265 0.05575707    0             0
 [15,] 0.7612479 0.1705833 0.06816878    0             0
 [16,] 0.8096020 0.1413833 0.04901482    0             0
 [17,] 0.8146966 0.1381748 0.04712864    0             0
 [18,] 0.8211664 0.1336643 0.04516934    0             0
 [19,] 0.8258169 0.1310432 0.04313993    0             0
 [20,] 0.8400036 0.1220004 0.03799614    0             1
 [21,] 0.7174268 0.1969166 0.08565662    0             1
 [22,] 0.8145925 0.1374473 0.04796017    0             0
 [23,] 0.8209493 0.1336509 0.04539988    0             0
 [24,] 0.8096839 0.1408426 0.04947359    0             0
 [25,] 0.8253588 0.1309749 0.04366616    0             0
 [26,] 0.8406563 0.1219606 0.03738302    0             2
 [27,] 0.7724525 0.1627864 0.06476122    0             0
 [28,] 0.7875632 0.1553209 0.05711601    0             0
 [29,] 0.7589930 0.1722718 0.06873526    0             0
 [30,] 0.7586113 0.1718937 0.06949496    0             0
 [31,] 0.8241317 0.1319424 0.04392590    0             0
 [32,] 0.7690958 0.1664320 0.06447217    0             0
 [33,] 0.8093122 0.1413758 0.04931199    0             0
 [34,] 0.8090462 0.1415607 0.04939310    0             0
 [35,] 0.8090619 0.1418075 0.04913053    0             1
 [36,] 0.8517559 0.1141697 0.03407443    0             0
 [37,] 0.8045968 0.1445093 0.05089384    0             0
 [38,] 0.8049881 0.1443592 0.05065272    0             0
 [39,] 0.8077152 0.1422716 0.05001321    0             0
 [40,] 0.7579969 0.1718197 0.07018334    0             0
 [41,] 0.7472508 0.1778071 0.07494207    0             0
 [42,] 0.8108293 0.1408888 0.04828193    0             0
 [43,] 0.7328492 0.1865705 0.08058023    0             0
 [44,] 0.7856843 0.1553218 0.05899391    0             0
 [45,] 0.8123661 0.1392788 0.04835503    0             0
 [46,] 0.8614472 0.1078880 0.03066481    0             0
 [47,] 0.7921520 0.1517358 0.05611222    0             0
 [48,] 0.8101835 0.1404185 0.04939802    0             0
 [49,] 0.8005143 0.1464781 0.05300760    0             1
 [50,] 0.7997616 0.1469873 0.05325122    0             1
 [51,] 0.7778296 0.1607458 0.06142469    0             1
 [52,] 0.7796027 0.1594659 0.06093136    0             0
 [53,] 0.7808071 0.1590572 0.06013574    0             1
 [54,] 0.7934139 0.1509788 0.05560741    0             1
 [55,] 0.7958486 0.1494025 0.05474891    0             0
 [56,] 0.7429253 0.1804126 0.07666215    0             0
 [57,] 0.7300381 0.1879021 0.08205976    0             0
 [58,] 0.7487127 0.1772704 0.07401697    0             0
 [59,] 0.7717768 0.1639415 0.06428177    0             0
 [60,] 0.7678462 0.1661893 0.06596450    0             0
 [61,] 0.7730860 0.1636247 0.06328920    0             1
 [62,] 0.7963328 0.1494886 0.05417855    0             0
```

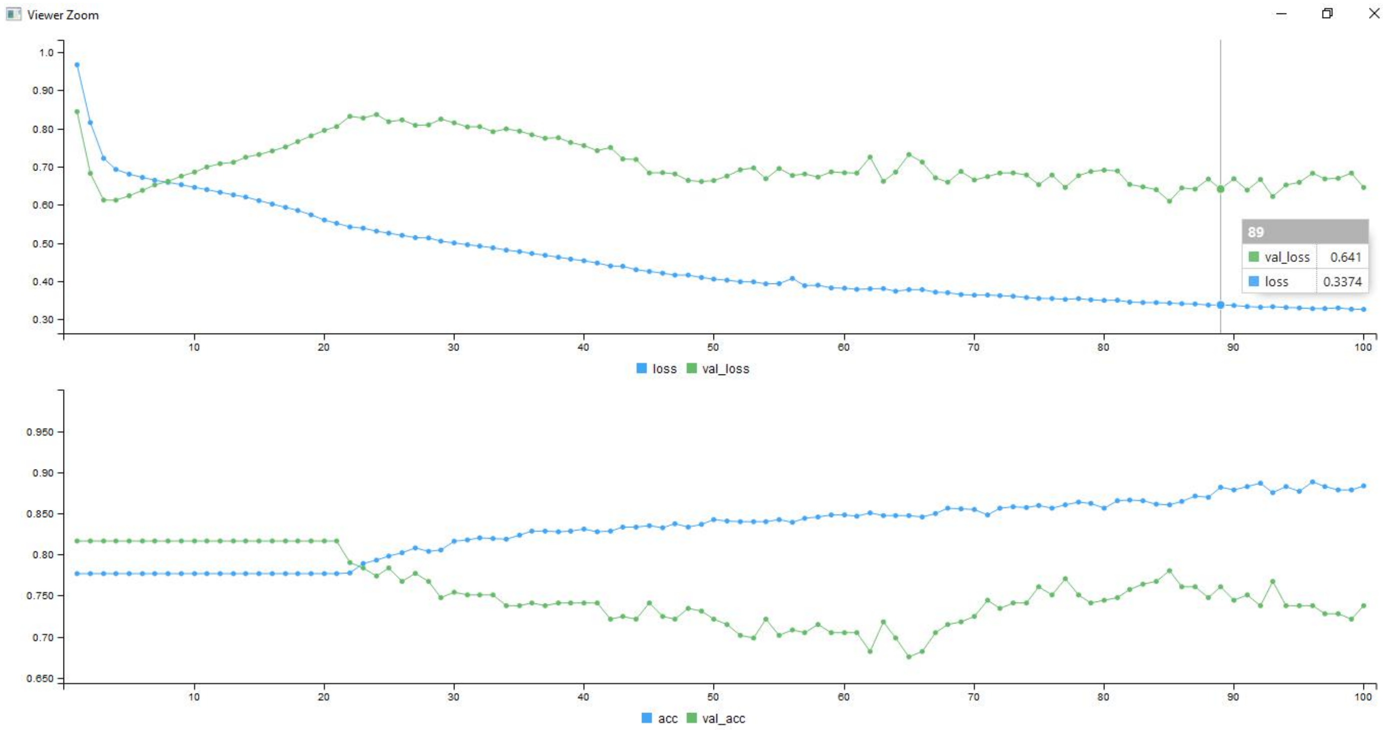
[63,]	0.7801046	0.1590965	0.06079886	0	0
[64,]	0.7721891	0.1638213	0.06398965	0	0
[65,]	0.7616735	0.1699901	0.06833642	0	0
[66,]	0.7479214	0.1775062	0.07457235	0	0
[67,]	0.6810951	0.2159857	0.10291921	0	1
[68,]	0.7394673	0.1839072	0.07662555	0	1
[69,]	0.7922646	0.1518560	0.05587953	0	0
[70,]	0.8037873	0.1446721	0.05154057	0	0
[71,]	0.7583856	0.1730025	0.06861195	0	1
[72,]	0.8147402	0.1374263	0.04783344	0	0
[73,]	0.7862329	0.1553620	0.05840516	0	0
[74,]	0.7819150	0.1580850	0.05999999	0	0
[75,]	0.7686539	0.1652088	0.06613730	0	0
[76,]	0.7773212	0.1602386	0.06244026	0	0
[77,]	0.7541006	0.1746049	0.07129454	0	0
[78,]	0.7941634	0.1504092	0.05542744	0	0
[79,]	0.7836518	0.1565428	0.05980542	0	0
[80,]	0.7892624	0.1533156	0.05742206	0	0
[81,]	0.7184860	0.1963318	0.08518227	0	1
[82,]	0.8128012	0.1392749	0.04792388	0	0
[83,]	0.7641763	0.1707963	0.06502738	0	1
[84,]	0.7076705	0.2024022	0.08992732	0	1
[85,]	0.7277403	0.1909248	0.08133484	0	1
[86,]	0.7261079	0.1909594	0.08293275	0	1
[87,]	0.6928455	0.2106666	0.09648787	0	2
[88,]	0.7510933	0.1774666	0.07144007	0	1
[89,]	0.6916916	0.2115196	0.09678880	0	2
[90,]	0.7403982	0.1836342	0.07596764	0	1
[91,]	0.7252105	0.1921609	0.08262859	0	1
[92,]	0.7115840	0.2001656	0.08825041	0	1
[93,]	0.7121238	0.1998699	0.08800627	0	1
[94,]	0.7454726	0.1801010	0.07442641	0	1
[95,]	0.7610822	0.1705994	0.06831843	0	1
[96,]	0.6588722	0.2283476	0.11278021	0	2
[97,]	0.7364257	0.1862894	0.07728484	0	1
[98,]	0.7027228	0.2051466	0.09213055	0	1
[99,]	0.6794631	0.2177722	0.10276470	0	2
[100,]	0.7501377	0.1772746	0.07258768	0	1
[101,]	0.7450772	0.1802670	0.07465584	0	1
[102,]	0.8079399	0.1428800	0.04918009	0	1
[103,]	0.7939106	0.1506609	0.05542844	0	0
[104,]	0.7820986	0.1580074	0.05989399	0	0
[105,]	0.6728698	0.2210165	0.10611378	0	2
[106,]	0.7502402	0.1770071	0.07275268	0	1
[107,]	0.7549301	0.1738038	0.07126606	0	1
[108,]	0.7544388	0.1750559	0.07050546	0	1
[109,]	0.7491606	0.1778589	0.07298037	0	1
[110,]	0.7608753	0.1714281	0.06769655	0	1
[111,]	0.7994208	0.1481813	0.05239793	0	1
[112,]	0.8004743	0.1466666	0.05285915	0	0
[113,]	0.8659840	0.1054250	0.02859097	0	0
[114,]	0.7608522	0.1711114	0.06803633	0	0
[115,]	0.7231676	0.1933017	0.08353066	0	1
[116,]	0.7538412	0.1754443	0.07071458	0	1
[117,]	0.7317120	0.1885075	0.07978050	0	1
[118,]	0.7302606	0.1893500	0.08038949	0	2
[119,]	0.7303780	0.1896290	0.07999299	0	1
[120,]	0.7752083	0.1628698	0.06192193	0	1
[121,]	0.7558675	0.1739386	0.07019388	0	1
[122,]	0.7652016	0.1681137	0.06668465	0	1
[123,]	0.8048032	0.1446275	0.05056927	0	0
[124,]	0.7625459	0.1713147	0.06613935	0	1
[125,]	0.7726092	0.1639347	0.06345610	0	1
[126,]	0.7774072	0.1609453	0.06164753	0	1
[127,]	0.7866889	0.1548240	0.05848721	0	0
[128,]	0.7757334	0.1612835	0.06298313	0	0
[129,]	0.8321033	0.1276204	0.04027634	0	1
[130,]	0.8456872	0.1185794	0.03573348	0	1

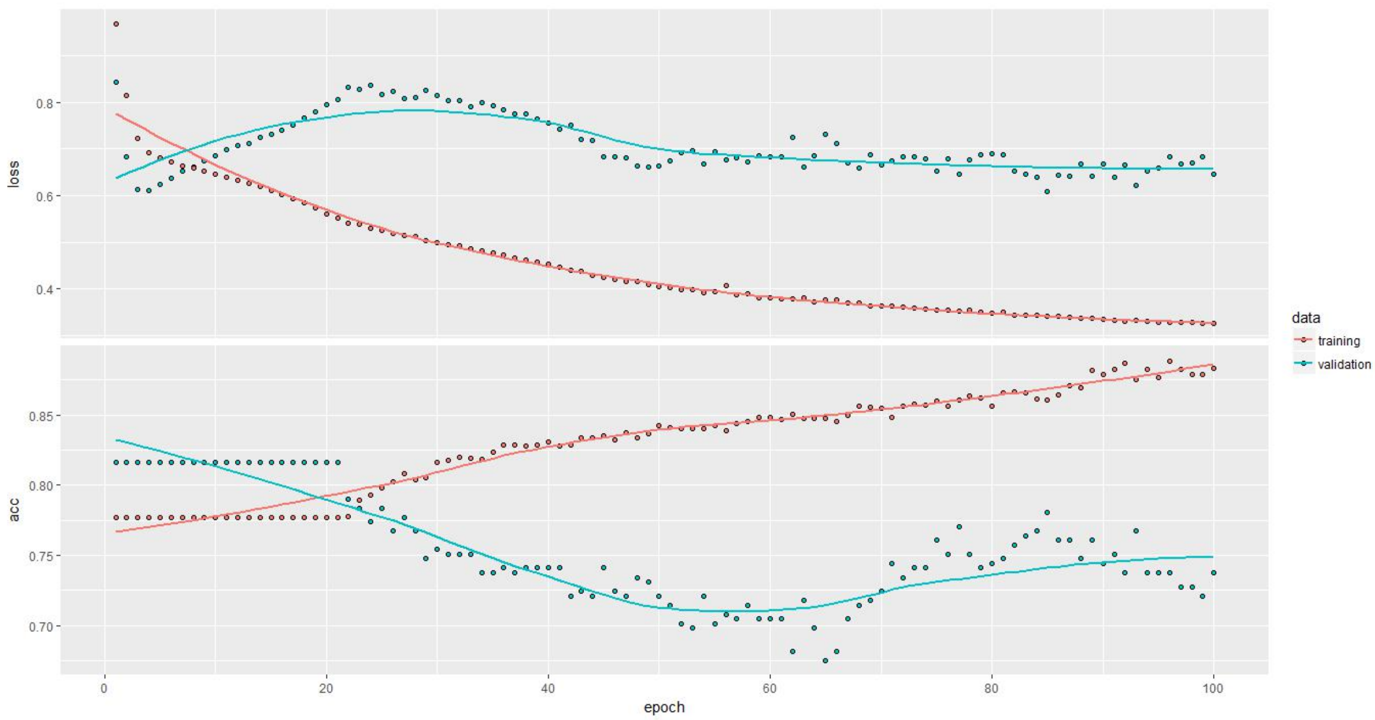
[131,]	0.8079936	0.1424796	0.04952686	0	0
[132,]	0.8077485	0.1425934	0.04965812	0	0
[133,]	0.7999476	0.1476867	0.05236570	0	1
[134,]	0.7841240	0.1567588	0.05911729	0	1
[135,]	0.7319708	0.1880627	0.07996649	0	1
[136,]	0.7904108	0.1532784	0.05631088	0	0
[137,]	0.8176651	0.1360202	0.04631469	0	0
[138,]	0.7515818	0.1759353	0.07248289	0	0
[139,]	0.7498354	0.1770093	0.07315527	0	0
[140,]	0.7155001	0.1964678	0.08803201	0	1
[141,]	0.7738408	0.1640424	0.06211675	0	2
[142,]	0.7386786	0.1847438	0.07657763	0	2
[143,]	0.7809283	0.1588606	0.06021107	0	1
[144,]	0.8132082	0.1388782	0.04791367	0	0
[145,]	0.8131596	0.1389201	0.04792029	0	0
[146,]	0.8057050	0.1433458	0.05094925	0	0
[147,]	0.8450459	0.1184961	0.03645806	0	0
[148,]	0.8380290	0.1230971	0.03887392	0	0
[149,]	0.8284242	0.1291082	0.04246755	0	0
[150,]	0.8234396	0.1322936	0.04426674	0	0
[151,]	0.8166284	0.1367116	0.04665997	0	0
[152,]	0.8156736	0.1373025	0.04702384	0	0
[153,]	0.8156542	0.1373258	0.04701989	0	0
[154,]	0.8163107	0.1369130	0.04677626	0	0
[155,]	0.8171996	0.1363106	0.04648969	0	0
[156,]	0.7996791	0.1472787	0.05304229	0	0
[157,]	0.7923948	0.1517369	0.05586819	0	0
[158,]	0.7801176	0.1589319	0.06095053	0	0
[159,]	0.8300359	0.1281536	0.04181049	0	0
[160,]	0.8262281	0.1305418	0.04323001	0	0
[161,]	0.7806703	0.1596535	0.05967627	0	1
[162,]	0.7948719	0.1506770	0.05445107	0	0
[163,]	0.7755181	0.1629379	0.06154398	0	1
[164,]	0.8069757	0.1428928	0.05013143	0	0
[165,]	0.7517151	0.1765943	0.07169068	0	1
[166,]	0.7611075	0.1708344	0.06805811	0	1
[167,]	0.7613258	0.1708589	0.06781531	0	1
[168,]	0.7621051	0.1699419	0.06795300	0	0
[169,]	0.7526606	0.1756419	0.07169748	0	1
[170,]	0.8530895	0.1131532	0.03375737	0	0
[171,]	0.8287549	0.1288272	0.04241800	0	0
[172,]	0.8512216	0.1143761	0.03440237	0	0
[173,]	0.8331319	0.1262474	0.04062070	0	0
[174,]	0.8260993	0.1307082	0.04319243	0	0
[175,]	0.8361557	0.1243105	0.03953382	0	0
[176,]	0.8121507	0.1396071	0.04824233	0	0
[177,]	0.8231536	0.1327281	0.04411837	0	0
[178,]	0.8461105	0.1180621	0.03582739	0	0
[179,]	0.7533776	0.1740976	0.07252484	0	0
[180,]	0.7571439	0.1717742	0.07108195	0	0
[181,]	0.8088304	0.1414865	0.04968310	0	0
[182,]	0.7782796	0.1595661	0.06215421	0	0
[183,]	0.8223091	0.1331467	0.04454417	0	0
[184,]	0.8197582	0.1347532	0.04548859	0	0
[185,]	0.8127602	0.1395947	0.04764506	0	0
[186,]	0.8223429	0.1329127	0.04474430	0	0
[187,]	0.8274476	0.1298531	0.04269938	0	0
[188,]	0.7911599	0.1526397	0.05620039	0	0
[189,]	0.8259712	0.1310938	0.04293486	0	0
[190,]	0.8126943	0.1391069	0.04819880	0	0
[191,]	0.8110836	0.1402369	0.04867946	0	0
[192,]	0.8038947	0.1449060	0.05119938	0	0
[193,]	0.8141975	0.1385117	0.04729089	0	0
[194,]	0.8248362	0.1317840	0.04337973	0	0
[195,]	0.8003572	0.1472795	0.05236342	0	0
[196,]	0.8149475	0.1377762	0.04727621	0	0
[197,]	0.8291720	0.1289401	0.04188794	0	0
[198,]	0.6868677	0.2136669	0.09946547	0	1

```
$loss
[1] 0.6810004
$acc
[1] 0.7628524
```

b) Number Of Epochs = 100

Training Accuracy and Loss





Testing Data Output :-

$loss

```
[1] 0.3940792
```

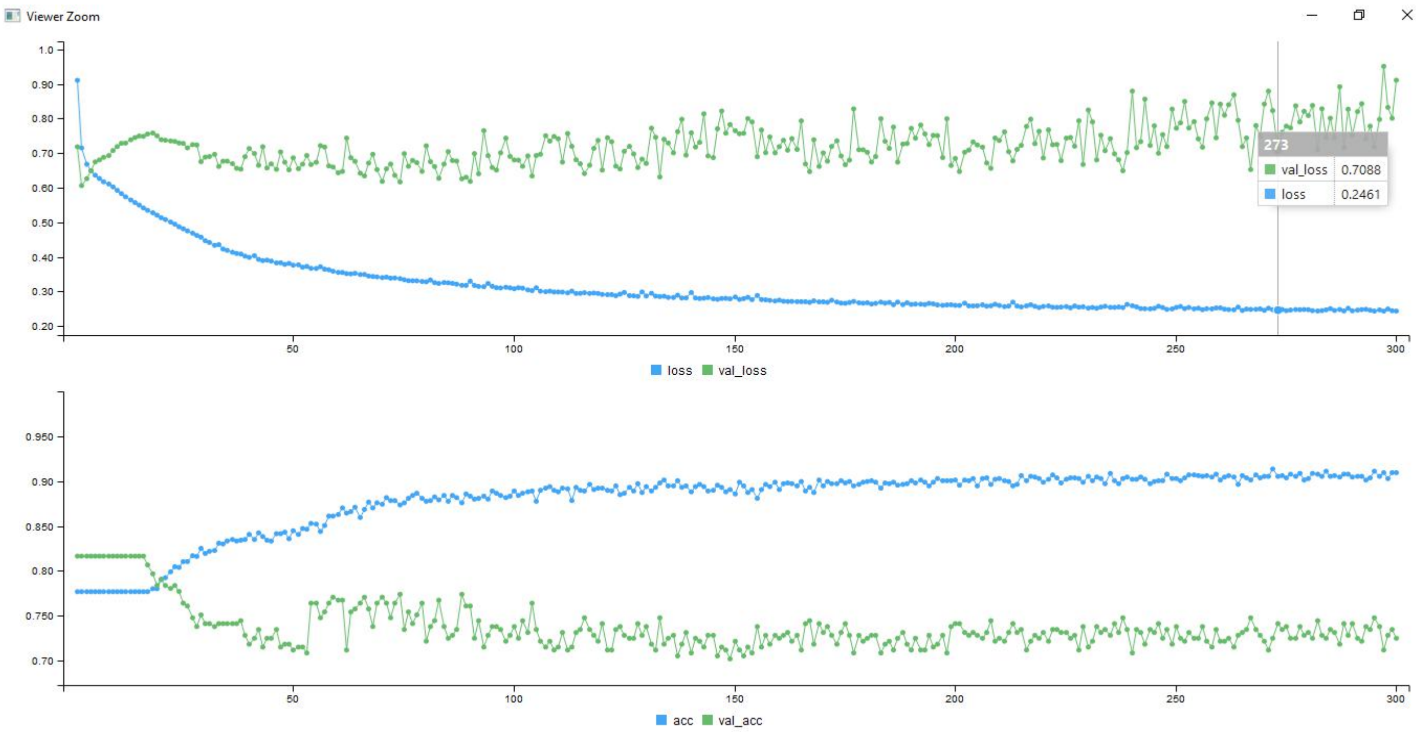
$acc

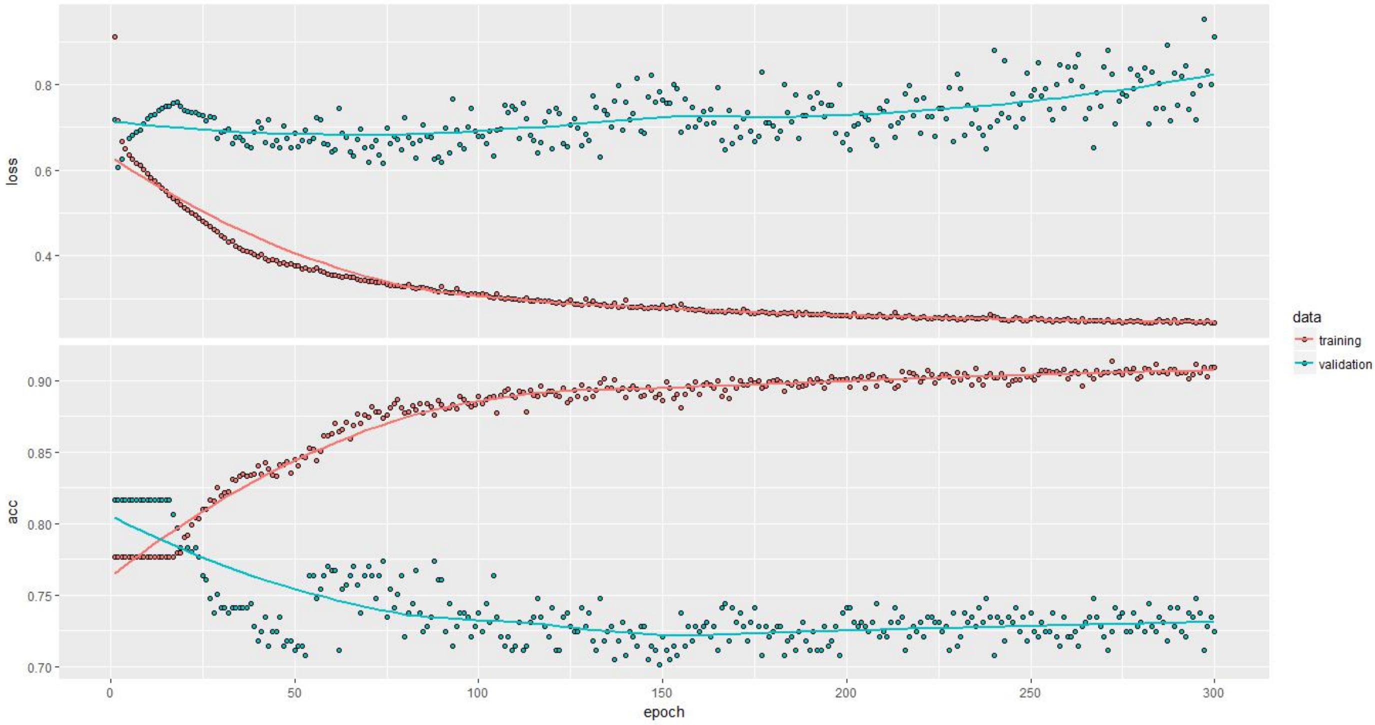
```
[1] 0.8557214
```

```
confusion_matrix

         Actual
Predicted   0   1   2
        0 428  34  14
        1  27  60   7
        2   5   0  28
```

c) Number Of Epochs = 300

Training Accuracy and Loss



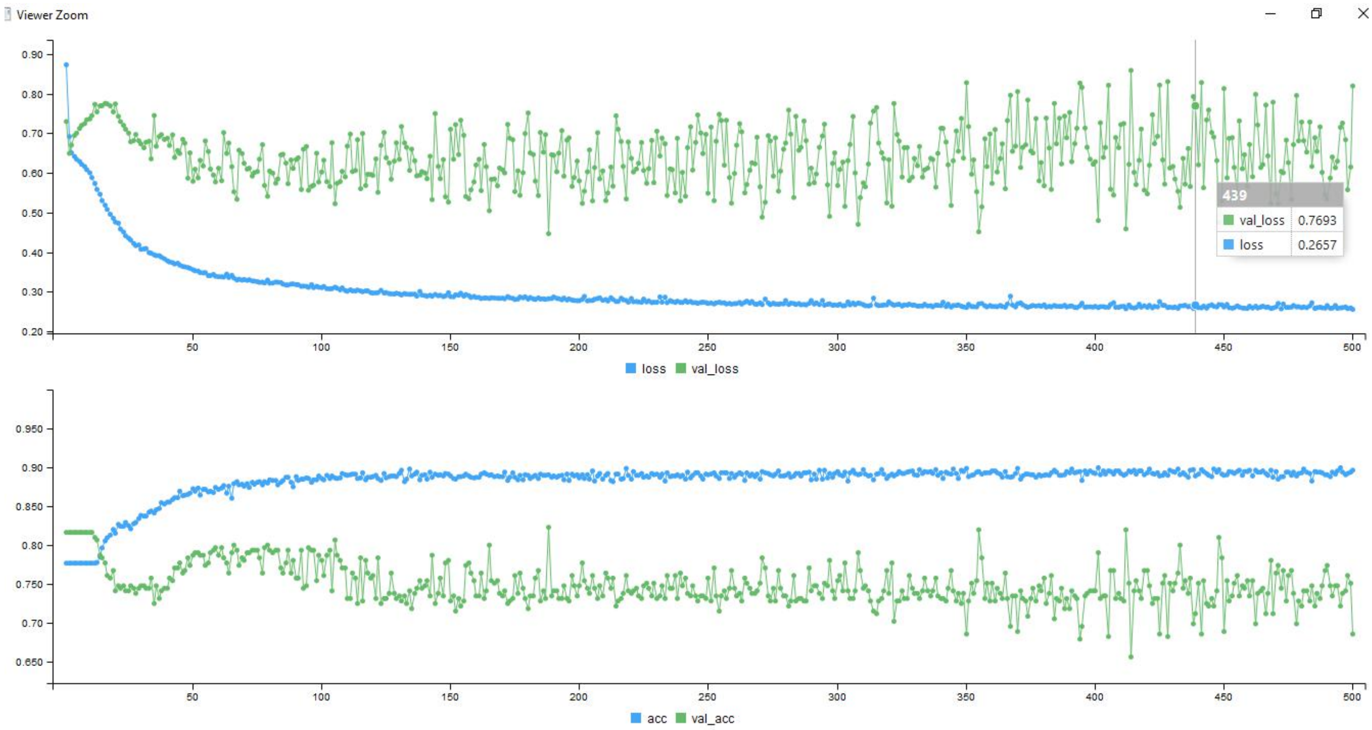


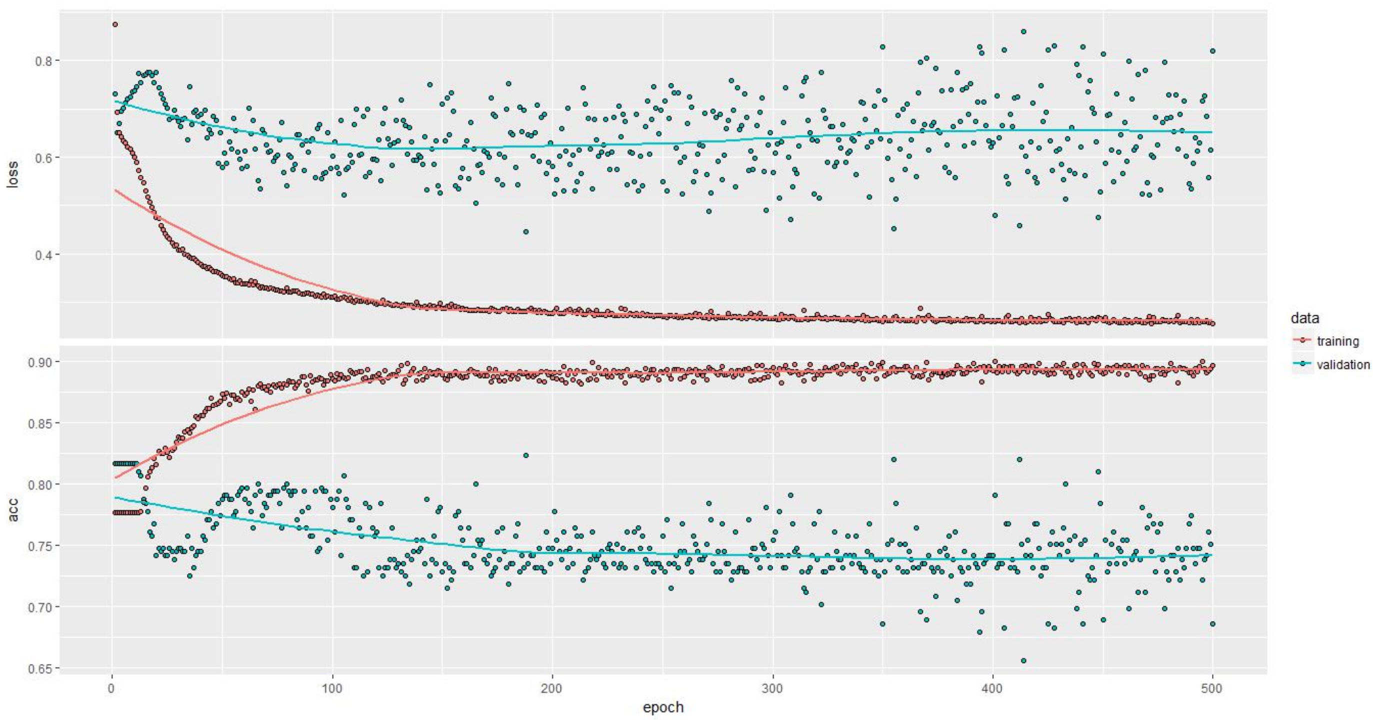
Testing Data Output :-

```
$loss
[1] 0.3794846

$acc
[1] 0.8756219

confusion_matrix
         Actual
Predicted   0   1   2
        0 420  17   5
        1  34  67   3
        2   6  10  41
```

d) Number Of Epochs = 500

Training Accuracy and Loss





Testing Data Output :-

```
$loss
[1] 0.3788026

$acc
[1] 0.8656716

confusion_matrix
         Actual
Predicted   0   1   2
        0 417  17   6
        1  35  64   2
        2   8  13  41
```

CONCLUSION :

The above observation gives us an insight into a very important feature that the accuracy and loss improves with the increment of epochs because it trains well with increase in the numbe r of iteration.

On reaching 500 epochs the accuracy and loss has somewhat become constant and thus it is u nable to achieve further improvement.