Name: Venkata Krishnarjun Vuppala
SRN: PES2UG19CS451
Subject: Topics in Deep Learning

Semester: 6
Section: G

Assignment 1

Lab programs

1. **Implementation of XOR function using basic gates.**

PES2UG19CS451.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
1 import tensorflow.compat.v1 as tf
2 tf.disable_v2_behavior()
3
4 def xor(x,y):
5   return ((x&~y)|(~x&y))
6 input1=tf.constant([False,False,True,True])
7 input2=tf.constant([False,True,False,True])
8
9 sess = tf.compat.v1.Session()
10 print("Input is input1 = ",sess.run(input1) ," input2 = ", sess.run(input2))
11 print("output is =", sess.run(xor(input1,input2)))
```

```
Input is input1 =  [False False  True  True]  input2 =  [False  True False  True]
output is = [False  True  True False]
```

2. **Implementation of neural network using tensorflow and keras**

https://keras.io/guides/sequential_model/

PES2UG19CS451.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
import tensorflow as tf
import keras

from keras import models
from keras.layers import Dense, Dropout
from tensorflow.keras.utils import to_categorical
from keras.datasets import mnist
from keras.utils.vis_utils import model_to_dot
from IPython.display import SVG

!pip install livelossplot

import livelossplot
plot_losses = livelossplot.PlotLossesKeras()
NUM_ROWS = 28
NUM_COLS = 28
NUM_CLASSES = 10
BATCH_SIZE = 128
EPOCHS = 10



def data_summary(X_train, y_train, X_test, y_test):
    """Summarize current state of dataset"""
    print('Train images shape:', X_train.shape)
    print('Train labels shape:', y_train.shape)
    print('Test images shape:', X_test.shape)
    print('Test labels shape:', y_test.shape)
    print('Train labels:', y_train)
    print('Test labels:', y_test)
```

PES2UG19CS451.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
31
32 # Load data
33 (X_train, y_train), (X_test, y_test) = mnist.load_data()
34
35 # Check state of dataset
36 data_summary(X_train, y_train, X_test, y_test)
37
38 # Reshape data
39 X_train = X_train.reshape((X_train.shape[0], NUM_ROWS * NUM_COLS))
40 X_train = X_train.astype('float32') / 255
41 X_test = X_test.reshape((X_test.shape[0], NUM_ROWS * NUM_COLS))
42 X_test = X_test.astype('float32') / 255
43
44 # Categorically encode labels
45 y_train = to_categorical(y_train, NUM_CLASSES)
46 y_test = to_categorical(y_test, NUM_CLASSES)
47
48 # Check state of dataset
49 data_summary(X_train, y_train, X_test, y_test)
50
51 # Build neural network
52 model = models.Sequential()
53 model.add(Dense(512, activation='relu', input_shape=(NUM_ROWS * NUM_COLS,)))
54 model.add(Dropout(0.5))
55 model.add(Dense(256, activation='relu'))
56 model.add(Dropout(0.25))
57 model.add(Dense(10, activation='softmax'))
```

PES2UG19CS451.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

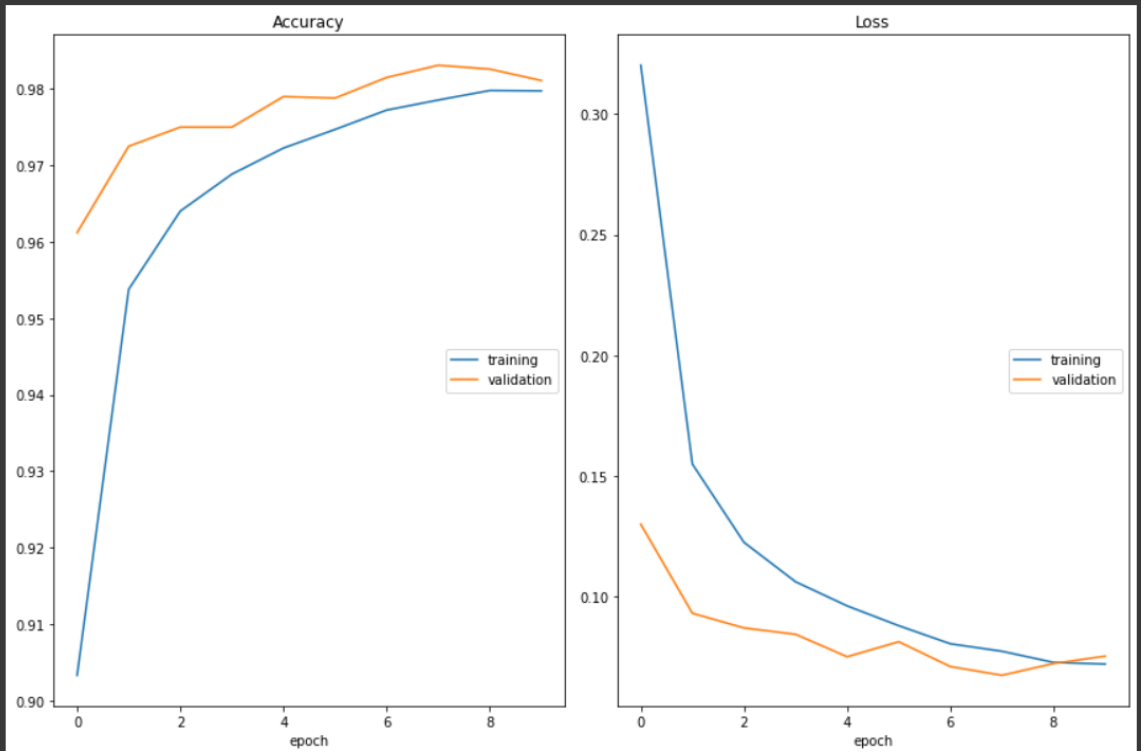
+ Code + Text

```
58
59 # Compile model
60 model.compile(optimizer='rmsprop',
61               loss='categorical_crossentropy',
62               metrics=['accuracy'])
63
64 # Train model
65 model.fit(X_train, y_train,
66           batch_size=BATCH_SIZE,
67           epochs=EPOCHS,
68           callbacks=[plot_losses],
69           verbose=1,
70           validation_data=(X_test, y_test))
71
72 score = model.evaluate(X_test, y_test, verbose=0)
73 print('Test loss:', score[0])
74 print('Test accuracy:', score[1])
75
76 model.summary()
77
78
```

PES2UG19CS451.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

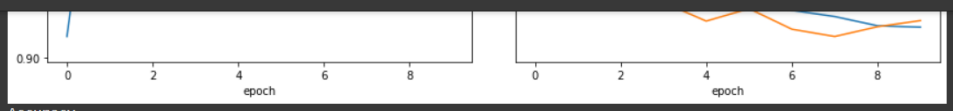
+ Code + Text



PES2UG19CS451.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text




```
Accuracy
        training                (min:    0.903, max:    0.980, cur:    0.980)
        validation              (min:    0.961, max:    0.983, cur:    0.981)
Loss
        training                (min:    0.072, max:    0.320, cur:    0.072)
        validation              (min:    0.068, max:    0.130, cur:    0.075)
60000/60000 [==============================] - 7s 117us/sample - loss: 0.0722 - acc: 0.9797 - val_loss: 0.0755 - val_acc: 0.9811
Test loss: 0.0754892836997984
Test accuracy: 0.9811
Model: "sequential"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense (Dense)                (None, 512)               401920

 dropout (Dropout)            (None, 512)               0

 dense_1 (Dense)              (None, 256)               131328

 dropout_1 (Dropout)          (None, 256)               0

 dense_2 (Dense)              (None, 10)                2570

=================================================================
Total params: 535,818
Trainable params: 535,818
Non-trainable params: 0
_________________________________________________________________
```

3. Implementation of computational graphs using tensorflow for simple expressions like e=(a+b)∗(b+1).