LAPORAN AKHIR PRAKTIKUM

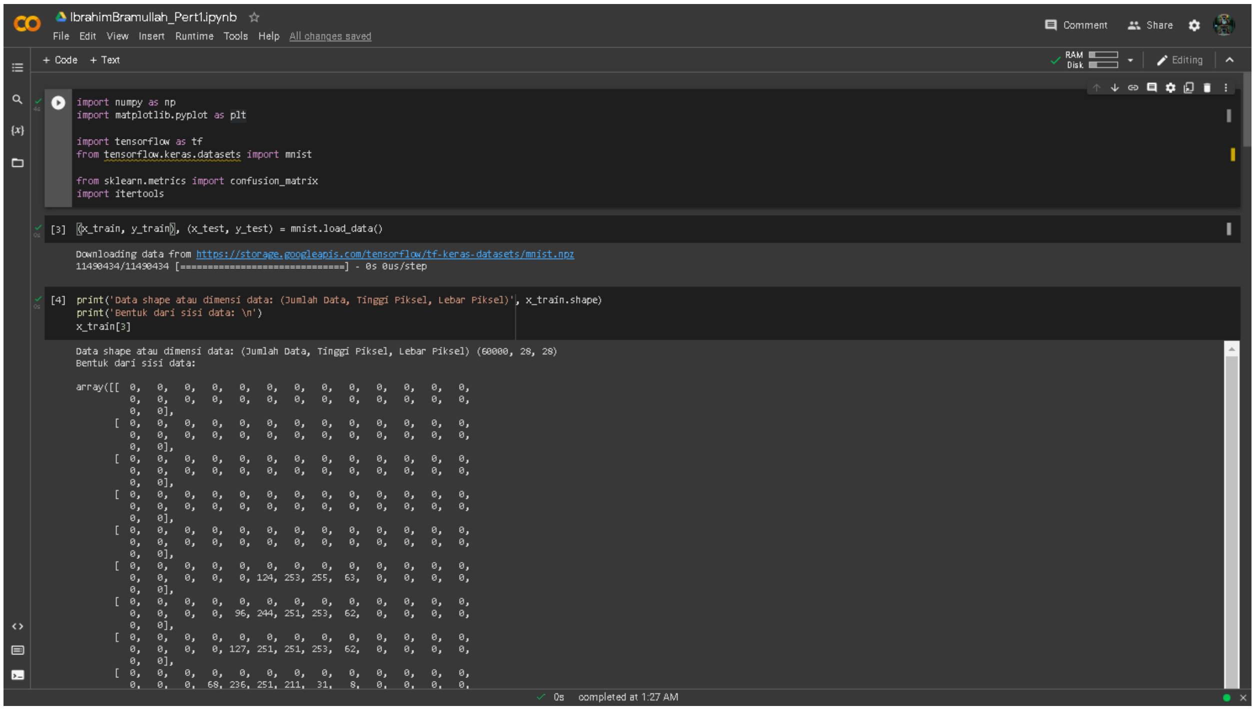
Mata Praktikum	: Kecerdasan Buatan
Kelas	: 3IA02
Praktikum ke-	: 1 (6)
Tanggal	: 23/11/22
Materi	: Google Colab
NPM	: 50420562
Nama	: Ibrahim Bramullah
Ketua Asisten	: David
Paraf Asisten	:
Nama Asisten	:
Jumlah Lembar	: 6 Lembar

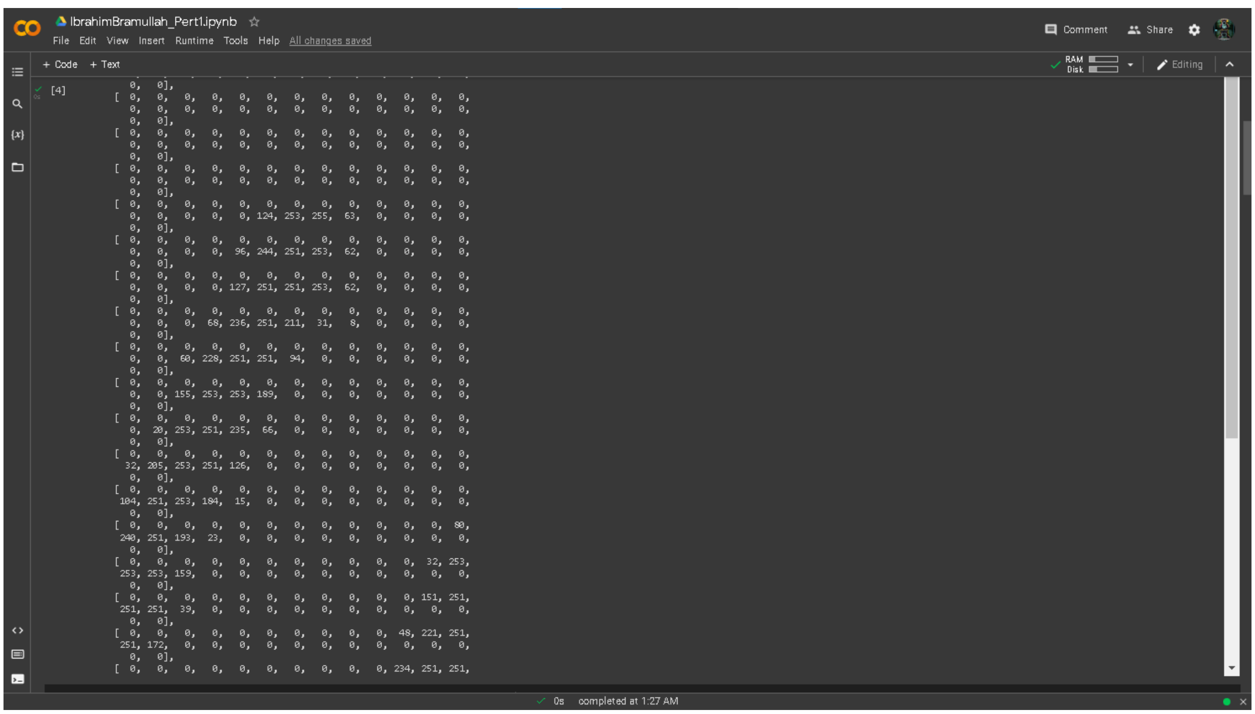
LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2022

Google Colab





IbrahimBramullah_Pert1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[5] x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1] * x_train.shape[2]))
    x_test = np.reshape(x_test, (x_test.shape[0], 784))
```

```
print('Data yang diubah menjadi 2 dimensi: ')
x_train[1] #hasil reshape
```

```
Data yang diubah menjadi 2 dimensi:
array([  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  51, 159, 253,
       159,  50,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  48, 238,
       252, 252, 252, 237,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  54,
       227, 253, 252, 239, 233, 252,  57,   6,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  10,
        60, 224, 252, 253, 252, 202,  84, 252, 253, 122,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0, 163, 252, 252, 252, 253, 252, 252,  96, 189, 253, 167,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,  51, 238, 253, 253, 190, 114, 253, 228,  47,  79, 255,
       168,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,  48, 238, 252, 252, 179,  12,  75, 121,  21,   0,
         0, 253, 243,  50,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,  38, 165, 253, 233, 208,  84,   0,   0,   0,
         0,   0,   0, 253, 252, 165,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   7, 178, 252, 240,  71,  19,  28,   0,
         0,   0,   0,   0,   0, 253, 252, 195,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,  57, 252, 252,  63,   0,   0,
         0,   0,   0,   0,   0,   0,   0, 253, 252, 195,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0, 198, 253, 190,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0, 255, 253, 196,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  76, 246, 252,
       112,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 253, 252,
       148,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  85,
       252, 230,  25,   0,   0,   0,   0,   0,   0,   0,   0,   7, 135,
       253, 186,  12,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,  85, 252, 223,   0,   0,   0,   0,   0,   0,   0,   0,   7,
       131, 252, 225,  71,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,  85, 252, 145,   0,   0,   0,   0,   0,   0,   0,
        48, 165, 252, 173,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,  86, 253, 225,   0,   0,   0,   0,   0,
         0, 114, 238, 253, 162,   0,   0,   0,   0,   0,   0,   0,   0,
```

✓ 0s completed at 1:27 AM

IbrahimBramullah_Pert1.ipynb

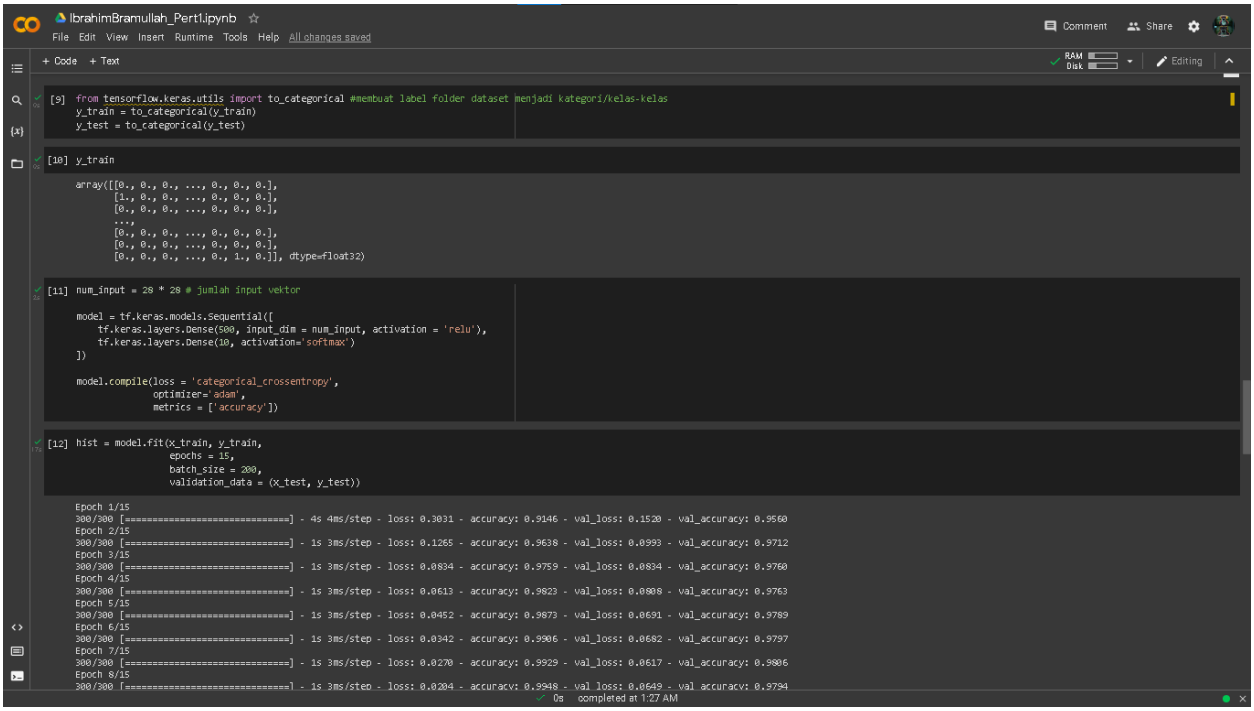
File Edit View Insert Runtime Tools Help All changes saved

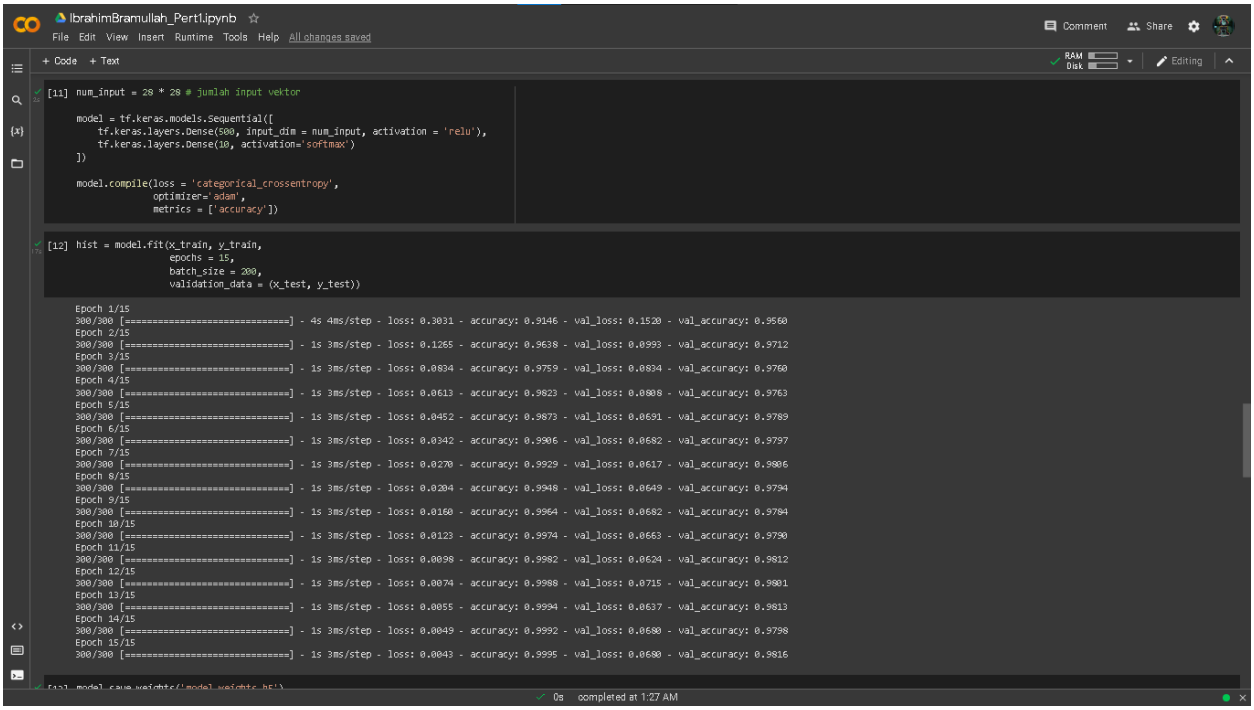
```
[7] # normalisasi tingkat kecerahan (0 - 255) menjadi (0 - 1)
    x_train = x_train/255.0
    x_test = x_test/255.0
```

```
[8] x_train[0]
```

```
       0.97647059, 0.25098039, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.18039216, 0.50980392,
       0.71764706, 0.99215686, 0.99215686, 0.81176471, 0.00784314,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.15294118,
       0.58039216, 0.89803922, 0.99215686, 0.99215686, 0.99215686,
       0.98039216, 0.71372549, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.09411765, 0.44705882, 0.86666667, 0.99215686, 0.99215686,
       0.99215686, 0.99215686, 0.78823529, 0.30588235, 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.09019608, 0.25882353, 0.83529412, 0.99215686,
       0.99215686, 0.99215686, 0.99215686, 0.77647059, 0.31764706,
       0.00784314, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.07058824, 0.67058824, 0.85882353,
       0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.76470588,
       0.31372549, 0.03529412, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.21568627, 0.6745098 ,
       0.88627451, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
       0.95686275, 0.52156863, 0.04313725, 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.53333333, 0.99215686, 0.99215686, 0.99215686,
       0.83137255, 0.52941176, 0.51764706, 0.0627451 , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
```

✓ 0s completed at 1:27 AM


IbrahimBramullah_Pert1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[9] from tensorflow.keras.utils import to_categorical #membuat label folder dataset menjadi kategori/kelas-kelas
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
[10] y_train
array([[0., 0., 0., ..., 0., 0., 0.],
[1., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
...,
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 1., 0.]], dtype=float32)
[11] num_input = 28 * 28 # jumlah input vektor
model = tf.keras.models.Sequential([
tf.keras.layers.Dense(500, input_dim = num_input, activation = 'relu'),
tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(loss = 'categorical_crossentropy',
optimizer='adam',
metrics = ['accuracy'])
[12] hist = model.fit(x_train, y_train,
epochs = 15,
batch_size = 200,
validation_data = (x_test, y_test))
Epoch 1/15
300/300 [==============================] - 4s 4ms/step - loss: 0.3031 - accuracy: 0.9146 - val_loss: 0.1520 - val_accuracy: 0.9560
Epoch 2/15
300/300 [==============================] - 1s 3ms/step - loss: 0.1265 - accuracy: 0.9638 - val_loss: 0.0993 - val_accuracy: 0.9712
Epoch 3/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0834 - accuracy: 0.9759 - val_loss: 0.0834 - val_accuracy: 0.9760
Epoch 4/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0613 - accuracy: 0.9823 - val_loss: 0.0808 - val_accuracy: 0.9763
Epoch 5/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0452 - accuracy: 0.9873 - val_loss: 0.0691 - val_accuracy: 0.9789
Epoch 6/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0342 - accuracy: 0.9906 - val_loss: 0.0682 - val_accuracy: 0.9797
Epoch 7/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0270 - accuracy: 0.9929 - val_loss: 0.0617 - val_accuracy: 0.9806
Epoch 8/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0204 - accuracy: 0.9948 - val_loss: 0.0649 - val_accuracy: 0.9794
0s completed at 1:27 AM


IbrahimBramullah_Pert1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[11] num_input = 28 * 28 # jumlah input vektor
model = tf.keras.models.Sequential([
tf.keras.layers.Dense(500, input_dim = num_input, activation = 'relu'),
tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(loss = 'categorical_crossentropy',
optimizer='adam',
metrics = ['accuracy'])
[12] hist = model.fit(x_train, y_train,
epochs = 15,
batch_size = 200,
validation_data = (x_test, y_test))
Epoch 1/15
300/300 [==============================] - 4s 4ms/step - loss: 0.3031 - accuracy: 0.9146 - val_loss: 0.1520 - val_accuracy: 0.9560
Epoch 2/15
300/300 [==============================] - 1s 3ms/step - loss: 0.1265 - accuracy: 0.9638 - val_loss: 0.0993 - val_accuracy: 0.9712
Epoch 3/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0834 - accuracy: 0.9759 - val_loss: 0.0834 - val_accuracy: 0.9760
Epoch 4/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0613 - accuracy: 0.9823 - val_loss: 0.0808 - val_accuracy: 0.9763
Epoch 5/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0452 - accuracy: 0.9873 - val_loss: 0.0691 - val_accuracy: 0.9789
Epoch 6/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0342 - accuracy: 0.9906 - val_loss: 0.0682 - val_accuracy: 0.9797
Epoch 7/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0270 - accuracy: 0.9929 - val_loss: 0.0617 - val_accuracy: 0.9806
Epoch 8/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0204 - accuracy: 0.9948 - val_loss: 0.0649 - val_accuracy: 0.9794
Epoch 9/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0160 - accuracy: 0.9964 - val_loss: 0.0682 - val_accuracy: 0.9784
Epoch 10/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0123 - accuracy: 0.9974 - val_loss: 0.0663 - val_accuracy: 0.9790
Epoch 11/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0098 - accuracy: 0.9982 - val_loss: 0.0624 - val_accuracy: 0.9812
Epoch 12/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0074 - accuracy: 0.9988 - val_loss: 0.0715 - val_accuracy: 0.9801
Epoch 13/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0055 - accuracy: 0.9994 - val_loss: 0.0637 - val_accuracy: 0.9813
Epoch 14/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0049 - accuracy: 0.9992 - val_loss: 0.0680 - val_accuracy: 0.9798
Epoch 15/15
300/300 [==============================] - 1s 3ms/step - loss: 0.0043 - accuracy: 0.9995 - val_loss: 0.0680 - val_accuracy: 0.9816
0s completed at 1:27 AM

IbrahimBramullah_Pert1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

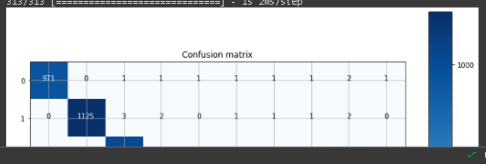
```
[13] model.save_weights('model_weights.h5')
```

```
[14] from matplotlib.gridspec import tight_layout
     def plot_confusion_matrix(cm, classes, normalize = False,
                               title = 'Confusion matrix',
                               cmap = plt.cm.Blues):
       plt.figure(figsize = (10, 10))
       plt.imshow(cm, interpolation = 'nearest', cmap = cmap)
       plt.title(title)
       plt.colorbar()
       tick_marks = np.arange(len(classes))
       plt.xticks(tick_marks, classes, rotation = 90)
       plt.yticks(tick_marks, classes)
       if normalize :
         cm = cm.astype('float')/cm.sum(axis=1)[:, np.newaxis]
       thresh = cm.max() / 2

       for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
         plt.text(j,i, cm[i, j],
                  horizontalalignment='center',
                  color='white' if cm[i, j] > thresh else 'black')
       plt.grid('off')
       plt.tight_layout()
       plt.ylabel('True Label')
       plt.xlabel('Predicted Label')
```

```
[15] list_classes = [0,1,2,3,4,5,6,7,8,9]
     Y_pred = model.predict(X_test)
     Y_pred_classes = np.argmax(Y_pred, axis=1)
     Y_true = np.argmax(y_test, axis=1)
     confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)
     plot_confusion_matrix(confusion_mtx, classes = list_classes)
```

```
313/313 [==============================] - 1s 2ms/step
```



0s completed at 1:27 AM

IbrahimBramullah_Pert1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

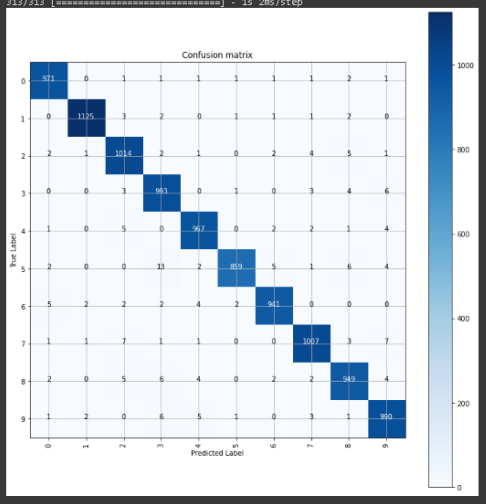
Comment Share

+ Code + Text

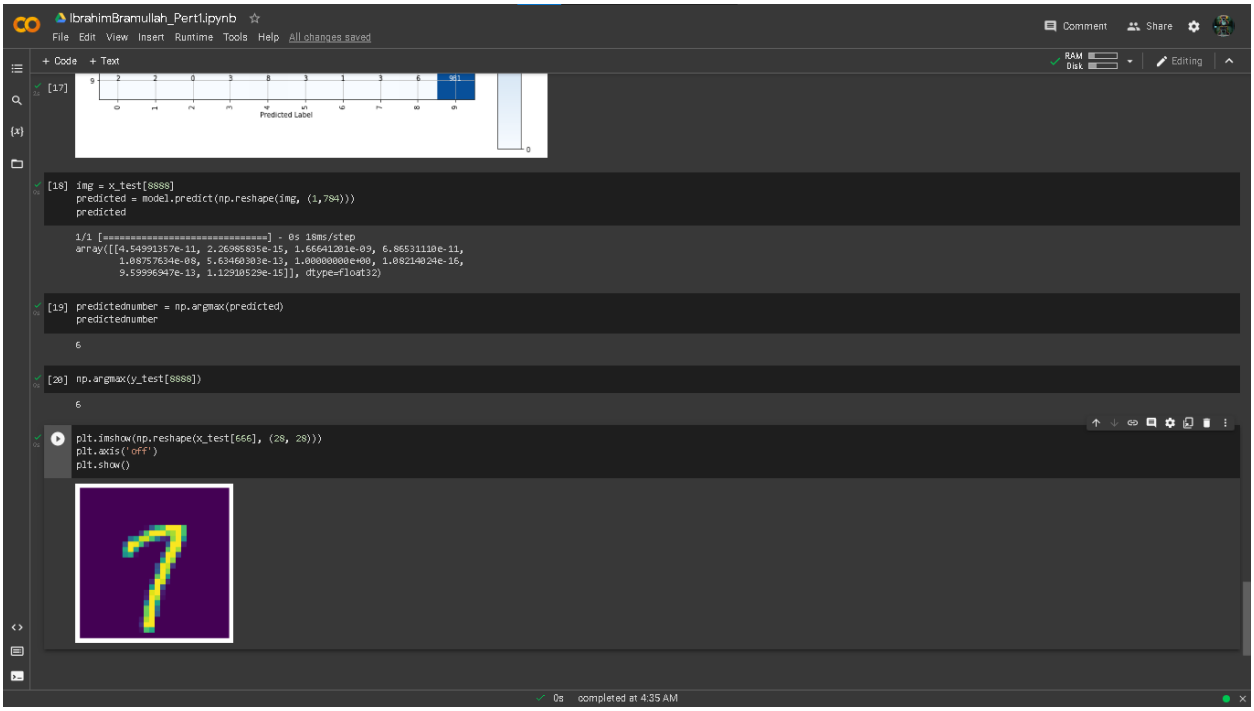
RAM Disk Editing

[15] list_classes = [0,1,2,3,4,5,6,7,8,9]
Y_pred = model.predict(X_test)
Y_pred_classes = np.argmax(Y_pred, axis=1)
Y_true = np.argmax(y_test, axis=1)
confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)
plot_confusion_matrix(confusion_mtx, classes = list_classes)

313/313 [==============================] - 1s 2ms/step



0s completed at 1:27 AM


IbrahimBramullah_Pert1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[17]
Predicted Label
[18] img = x_test[8888]
predicted = model.predict(np.reshape(img, (1,784)))
predicted
1/1 [==============================] - 0s 18ms/step
array([[4.54991357e-11, 2.26985835e-15, 1.66641201e-09, 6.86531110e-11,
1.08757634e-08, 5.63468303e-13, 1.00000000e+00, 1.08214024e-16,
9.59996947e-13, 1.12910529e-15]], dtype=float32)
[19] predictednumber = np.argmax(predicted)
predictednumber
6
[20] np.argmax(y_test[8888])
6
plt.imshow(np.reshape(X_test[666], (28, 28)))
plt.axis('off')
plt.show()
0s completed at 4:35 AM