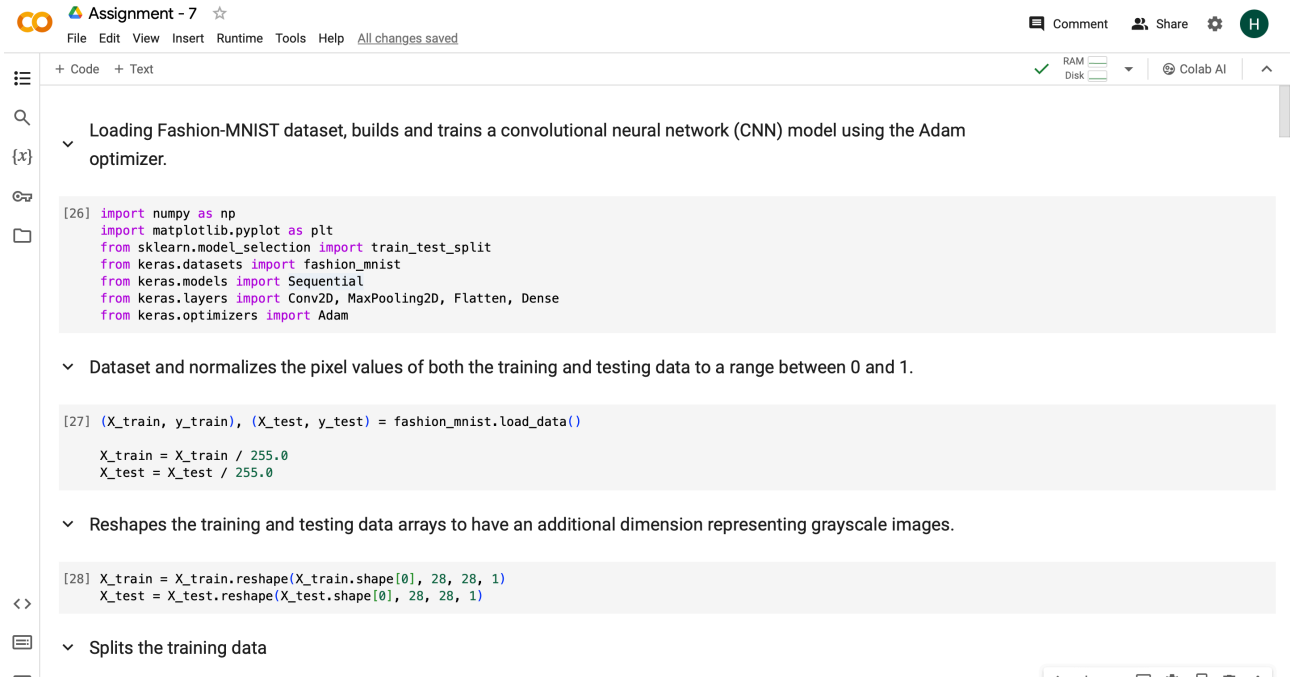
Neural Networks & Deep Learning Assignment-7

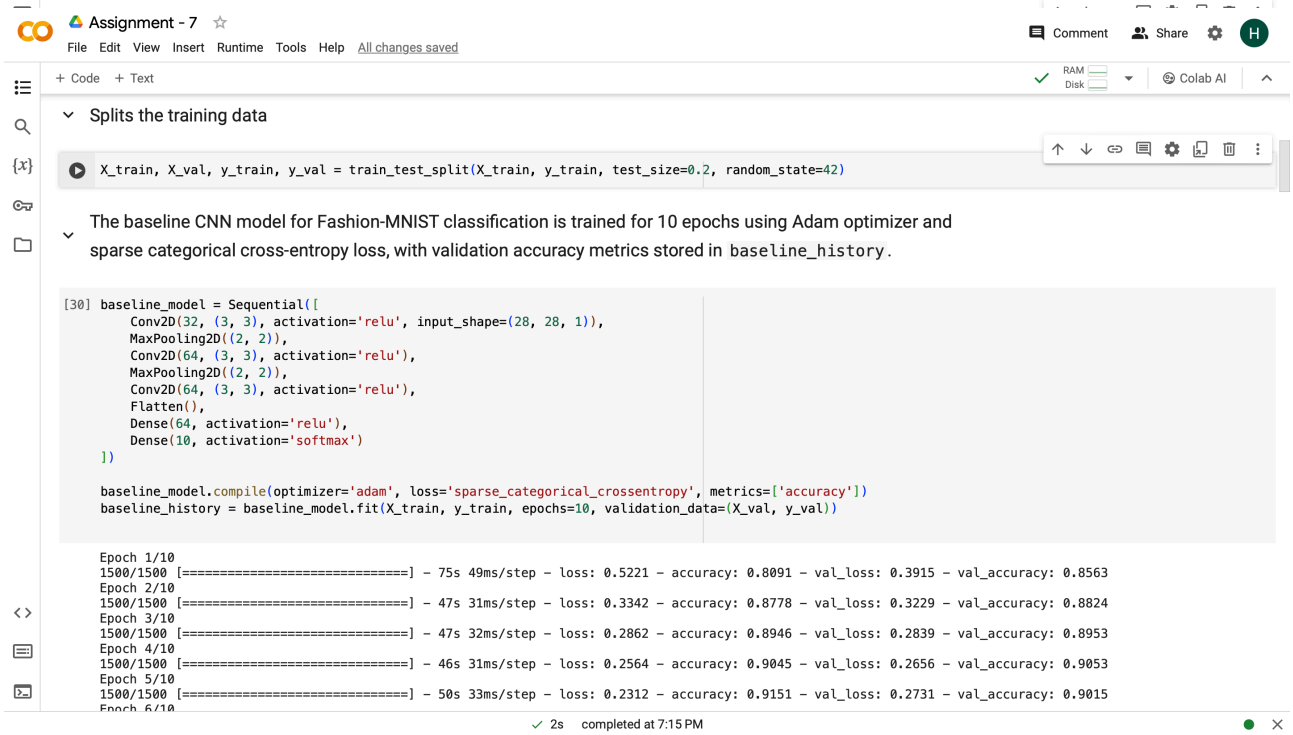
Harshavardhan Reddy Pattepur
700754833

Repository Link :
https://github.com/harshavardhanreddy27/NNDL-Assignment---7

Video Link:
https://drive.google.com/file/d/1pKvDbWiDBSMttHmu758UheQDC3QJgRpr/view?usp=share_link

Code Screenshots:





Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

The code line evaluates the baseline model's accuracy on the testing dataset (`X_test`, `y_test`) and prints the resulting accuracy score.

```
[31] baseline_loss, baseline_accuracy = baseline_model.evaluate(X_test, y_test)
     print("Baseline Model Accuracy:", baseline_accuracy)
```

```
313/313 [==============================] - 4s 12ms/step - loss: 0.2911 - accuracy: 0.9047
Baseline Model Accuracy: 0.904699981212616
```

The code saves and loads a Keras model from "improved_model.h5", then uses the loaded model to predict outcomes on the testing dataset X_test.

+ Code + Text

```
[41] from keras.models import load_model

     baseline_model.save("improved_model.h5")
     loaded_model = load_model("improved_model.h5")
     predictions = loaded_model.predict(X_test)
```

```
313/313 [==============================] - 2s 6ms/step
```

Shows training and validation plots

```
[39] plt.figure(figsize=(12, 6))

     plt.subplot(1, 2, 1)
     plt.plot(baseline_history.history['loss'], label='Training Loss')
     plt.plot(baseline_history.history['val_loss'], label='Validation Loss')
```

✓ 2s completed at 7:15 PM

Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

Shows training and validation plots

```
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(baseline_history.history['loss'], label='Training Loss')
plt.plot(baseline_history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training and Validation Loss over Epochs')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(baseline_history.history['accuracy'], label='Training Accuracy')
plt.plot(baseline_history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy over Epochs')
plt.legend()

plt.tight_layout()
plt.show()
```

✓ 2s completed at 7:15 PM

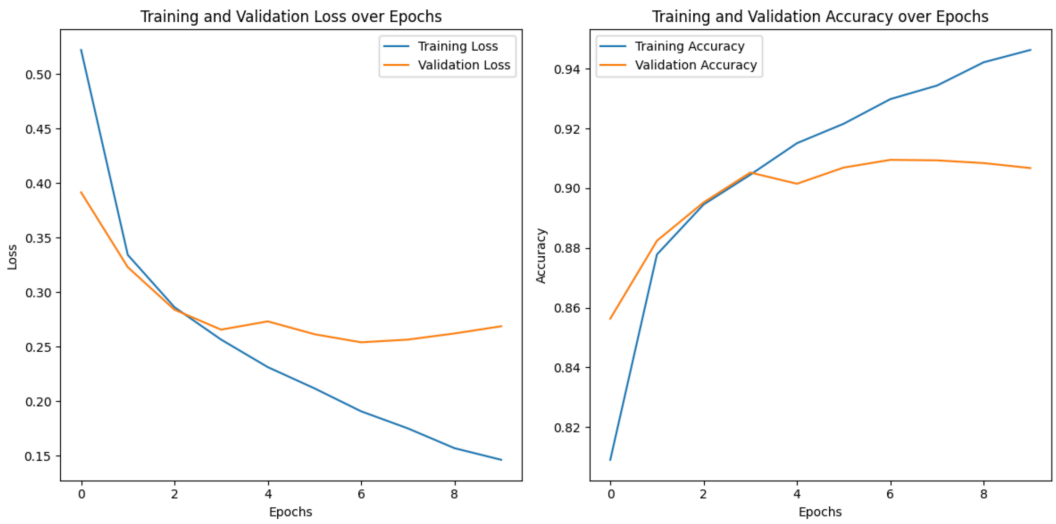
Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI



Heatmap visualization of the confusion matrix

✓ 2s completed at 7:15 PM

Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

Heatmap visualization of the confusion matrix

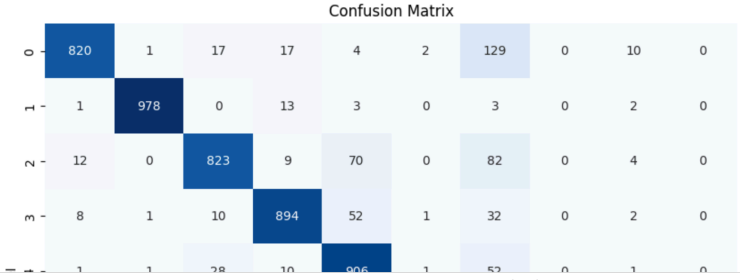
```
from sklearn.metrics import confusion_matrix
import seaborn as sns

y_pred_probs = baseline_model.predict(X_test)
y_pred = np.argmax(y_pred_probs, axis=1)

conf_matrix = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

```
313/313 [==============================] - 3s 9ms/step
```



✓ 2s completed at 7:15 PM

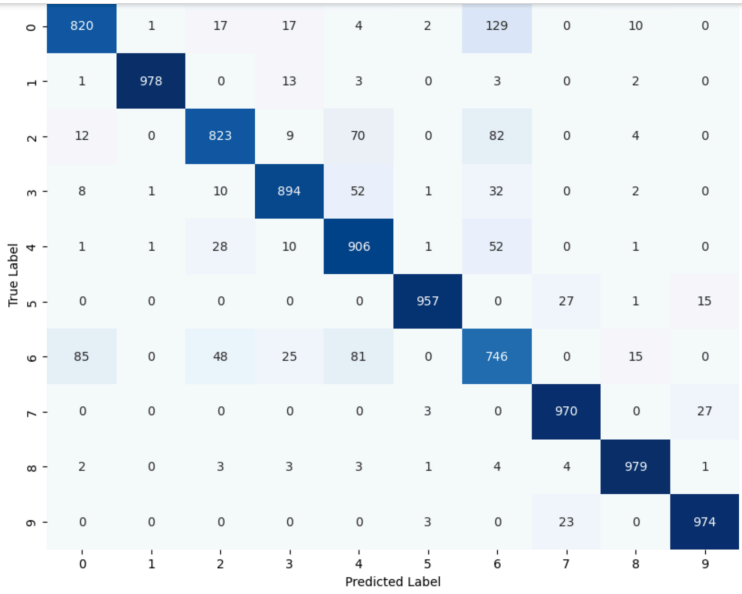
Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI



✓ 2s completed at 7:15 PM

Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

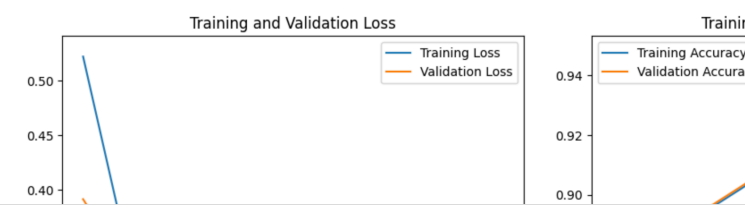
RAM Disk Colab AI

```
plt.figure(figsize=(12, 6))

# Loss subplot
plt.subplot(1, 2, 1)
plt.plot(baseline_history.history['loss'], label='Training Loss')
plt.plot(baseline_history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training and Validation Loss')
plt.legend()

# Accuracy subplot
plt.subplot(1, 2, 2)
plt.plot(baseline_history.history['accuracy'], label='Training Accuracy')
plt.plot(baseline_history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()

plt.tight_layout()
plt.show()
```



✓ 2s completed at 7:15 PM

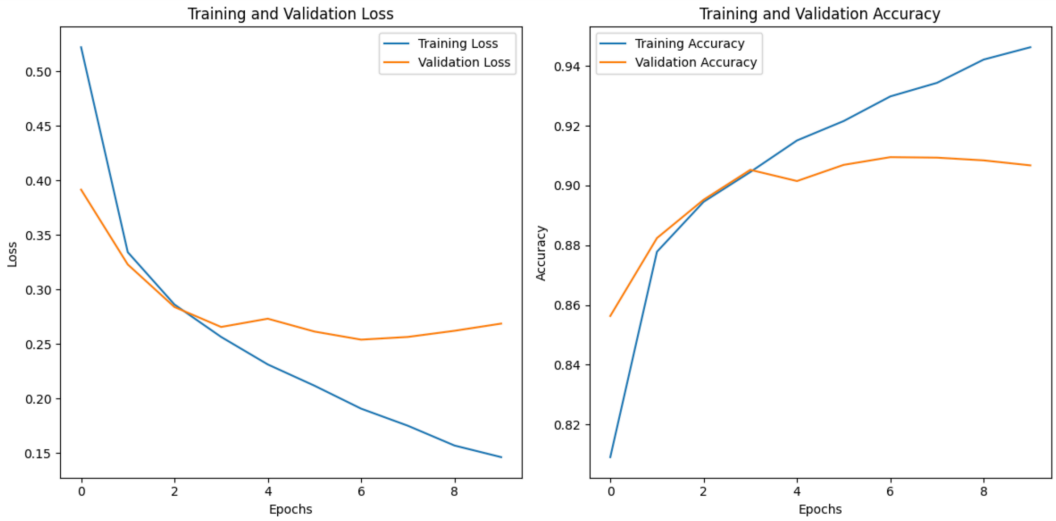
Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI




```
[36] from sklearn.metrics import roc_curve, auc
     from sklearn.preprocessing import LabelBinarizer
```

✓ 2s completed at 7:15 PM

Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

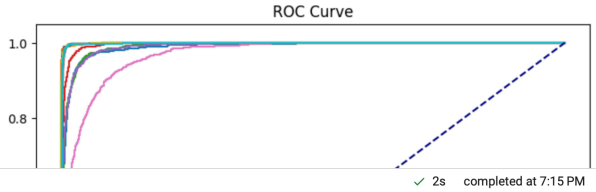
```
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import LabelBinarizer

lb = LabelBinarizer()
y_test_bin = lb.fit_transform(y_test)
y_prob = baseline_model.predict(X_test)

fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(len(lb.classes_)):
    fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_prob[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

plt.figure(figsize=(8, 6))
for i in range(len(lb.classes_)):
    plt.plot(fpr[i], tpr[i], label=f'Class {lb.classes_[i]} (AUC = {roc_auc[i]:0.2f})')
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc="lower right")
plt.show()
```

```
313/313 [==============================] - 4s 13ms/step
```



✓ 2s completed at 7:15 PM

Assignment - 7

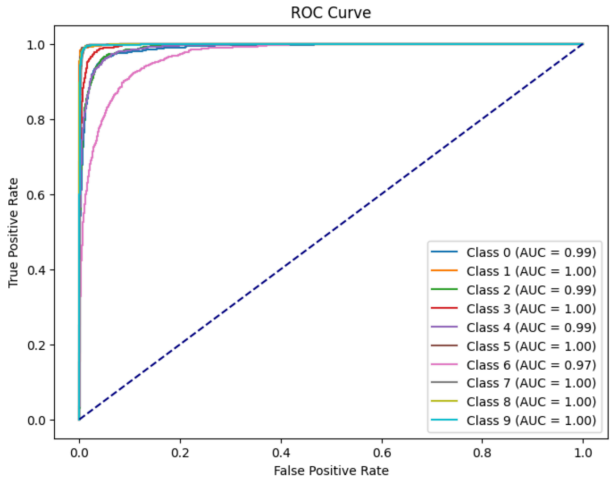
File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

```
313/313 [==============================] - 4s 13ms/step
```




```
[37] from sklearn.metrics import precision_recall_curve

     precision = dict()
     recall = dict()
     pr_auc = dict()
```

✓ 2s completed at 7:15 PM

Assignment - 7

File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

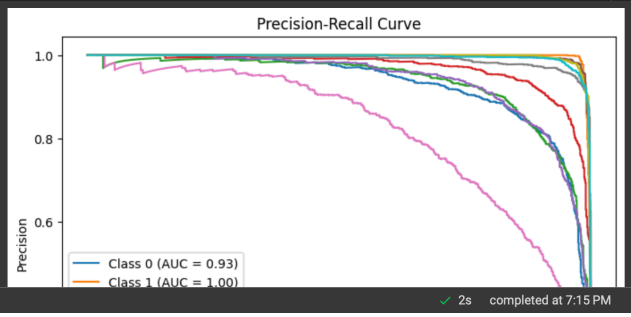
+ Code + Text

RAM Disk Colab AI

```
from sklearn.metrics import precision_recall_curve

precision = dict()
recall = dict()
pr_auc = dict()
for i in range(len(lb.classes_)):
    precision[i], recall[i], _ = precision_recall_curve(y_test_bin[:, i], y_prob[:, i])
    pr_auc[i] = auc(recall[i], precision[i])

plt.figure(figsize=(8, 6))
for i in range(len(lb.classes_)):
    plt.plot(recall[i], precision[i], label=f'Class {lb.classes_[i]} (AUC = {pr_auc[i]:0.2f})')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.legend(loc="lower left")
plt.show()
```



✓ 2s completed at 7:15 PM

Assignment - 7

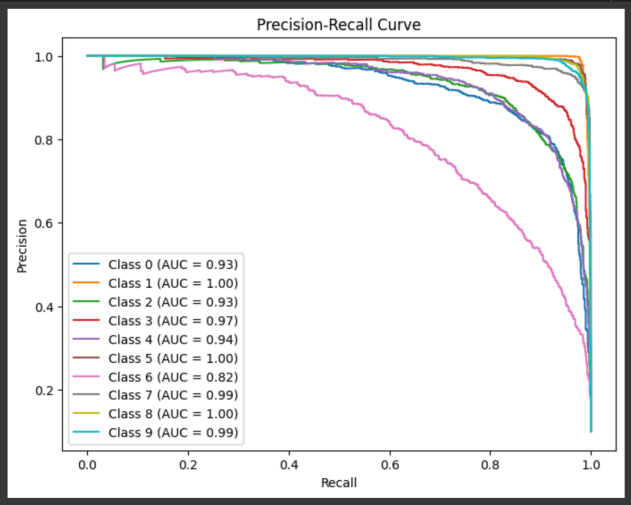
File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

```
plt.legend(loc="lower left")
plt.show()
```



The key modifications made to the baseline model for the new Fashion-MNIST dataset are as follows:

2s completed at 7:15 PM

Assignment - 7

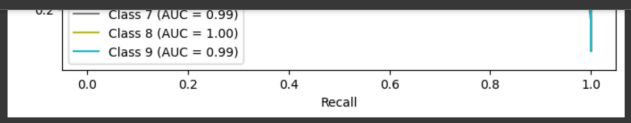
File Edit View Insert Runtime Tools Help All changes saved

Comment Share H

+ Code + Text

RAM Disk Colab AI

[37]



The key modifications made to the baseline model for the new Fashion-MNIST dataset are as follows:

1. **Dataset Selection**:Fashion-MNIST, which has grayscale images of clothing items and 10 classes. This change accounts for differences in data distribution and complexity.
2. **Model Architecture**: Retained a similar architecture but adjusted for Fashion-MNIST's input shape (28x28 grayscale images). Consisted of convolutional layers, max-pooling layers, flattening, and dense layers with ReLU activation for convolutional layers and softmax for the output layer.
3. **Normalization**: Applied pixel value normalization by dividing by 255.0, ensuring stable training by scaling input features.
4. **Training Parameters**: Trained for 10 epochs with Adam optimizer and sparse categorical cross-entropy loss. Parameters such as batch size and learning rate remained consistent.
5. **Evaluation and Analysis**: Evaluated the model's performance on the test set using accuracy and loss metrics. Visualizations like confusion matrix, loss-accuracy plots, ROC curve, and precision-recall curve provided insights into classification, convergence, and discrimination ability.
6. **Improvement Assessment**: Compared the modified model's metrics and visualizations with the baseline to assess performance. If the modified model showed higher accuracy, lower loss, improved convergence, and discrimination, it demonstrated positive response to changes.

2s completed at 7:15 PM