Ex: 9

30.09.25

Build a Recurrent Neural Network (RNN)

Aim: To build a Recurrent Neural Network (RNN).

Pseudo code:

1. Initialize RNN parameters:
   - Input size
   - Hidden state size
   - Output size (no. of classes)
2. Define RNN forward pass:
   for each time step t in sequence length:
   - update hidden state using input at t & previous hidden state, output pred. based on final hidden state.
3. Define loss function (eg: cross Entropy loss)
4. Train RNN:
   for each epoch:
   for each batch:
   - forward pass: compute outputs.
   - compute loss
   - Backpropagate loss
   - update model parameters.
5. Evaluate the model on test data:
   - Calc. accuracy, recall, f1 score, etc.
6. Plot training loss & accuracy over epochs.

Justification:

RNN: chosen because it can process sequential data by maintaining hidden state.

Result:

RNN successfully built & trained on sequential data.

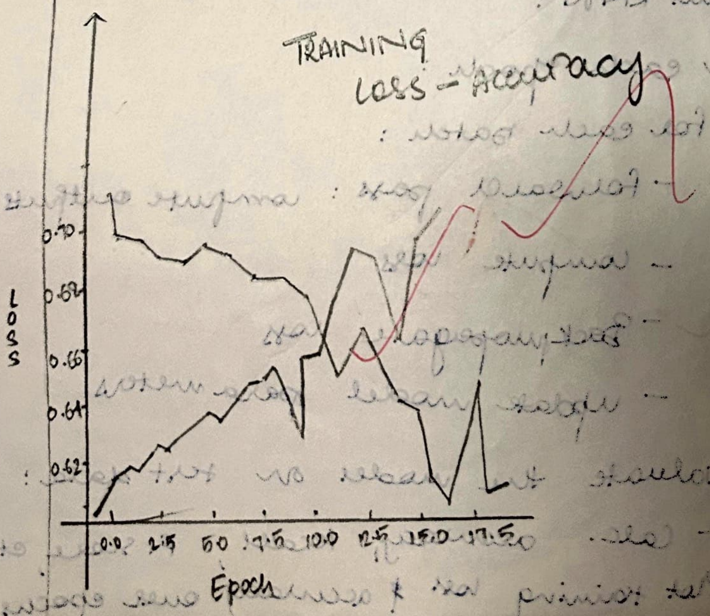
Successfully implemented.

30/9

Observ-ation:

epoch 1/20 - loss : 0.7046 - Accuracy: 0.480
epoch 2/20 - loss : 0.6950 - Accuracy: 0.535
epoch 3/20 - loss : 0.6938 - Accuracy: 0.5490
epoch 4/20 - loss : 0.6888 - Accuracy: 0.545
epoch 5/20 - loss : 0.6869 - Accuracy: 0.56
epoch 6/20 - loss : 0.6895 - Accuracy: 0.55
epoch 7/20 - loss : 0.6867 - Accuracy: 0.56
epoch 8/20 - loss : 0.6821 - Accuracy: 0.574
epoch 9/20 - loss : 0.6820 - Accuracy: 0.56
epoch 10/20 - loss : 0.6779 - Accuracy: 0.600
epoch 11/20 - loss : 0.6675 - Accuracy: 0.5990
epoch 12/20 - loss : 0.6591 - Accuracy: 0.617
epoch 13/20 - loss : 0.6710 - Accuracy: 0.588
epoch 14/20 - loss : 0.6536 - Accuracy: 0.623
epoch 15/20 - loss : 0.6512 - Accuracy: 0.622
epoch 16/20 - loss : 0.6239 - Accuracy: 0.668
epoch 17/20 - loss : 0.6163 - Accuracy: 0.663
epoch 18/20 - loss : 0.6420 - Accuracy: 0.63
epoch 19/20 - loss : 0.6131 - Accuracy: 0.667
epoch 20/20 - loss : 0.6138 - Accuracy: 0.679

Final Metrics:
Accuracy : 0.6790
Recall : 0.6738
F1 score : 0.6825



[17]

```python
%matplotlib inline
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.metrics import f1_score, accuracy_score
import matplotlib.pyplot as plt
import numpy as np

def generate_data(n_samples=1000, seq_len=10):
    X = np.random.rand(n_samples, seq_len, 1)
    y = (X.sum(axis=1) > 5).astype(int)
    return torch.tensor(X, dtype=torch.float32), torch.tensor(y, dtype=torch.long).squeeze()

X, y = generate_data()
train_size = int(0.8 * len(X))
X_train, X_val = X[:train_size], X[train_size:]
y_train, y_val = y[:train_size], y[train_size:]

class SimpleRNN(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(SimpleRNN, self).__init__()
        self.rnn = nn.RNN(input_size, hidden_size, batch_first=True)
        self.fc = nn.Linear(hidden_size, num_classes)
    def forward(self, x):
        out, _ = self.rnn(x)
        out = out[:, -1, :]
        out = self.fc(out)
        return out

model = SimpleRNN(input_size=1, hidden_size=16, num_classes=2)
```

Run all

Share

RAM
Disk

[17]
1s

```
        self.fc = nn.Linear(hidden_size, num_classes)
    def forward(self, x):
        out, _ = self.rnn(x)
        out = out[:, -1, :]
        out = self.fc(out)
        return out

model = SimpleRNN(input_size=1, hidden_size=16, num_classes=2)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.01)

epochs = 50
train_losses = []
val_losses = []

for epoch in range(epochs):
    model.train()
    optimizer.zero_grad()
    outputs = model(X_train)
    loss = criterion(outputs, y_train)
    loss.backward()
    optimizer.step()
    train_losses.append(loss.item())
    model.eval()
    with torch.no_grad():
        val_outputs = model(X_val)
        val_loss = criterion(val_outputs, y_val)
        val_losses.append(val_loss.item())
    if (epoch+1) % 10 == 0:
        print(f"Epoch [{epoch+1}/{epochs}] | Train Loss: {loss.item():.4f} | Val Loss: {val_loss.item():.4f}")

model.eval()
```

Untitled8.ipynb
Saving failed since 7:38 PM

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```python
    model.eval()
    with torch.no_grad():
        val_outputs = model(X_val)
        val_loss = criterion(val_outputs, y_val)
        val_losses.append(val_loss.item())
    if (epoch+1) % 10 == 0:
        print(f"Epoch [{epoch+1}/{epochs}] | Train Loss: {loss.item():.4f} | Val Loss: {val_loss.item():.4f}")

model.eval()
with torch.no_grad():
    y_pred = model(X_val).argmax(dim=1).numpy()
    y_true = y_val.numpy()

acc = accuracy_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)

print("\nFinal Evaluation:")
print(f"Accuracy: {acc*100:.2f}%")
print(f"F1 Score: {f1:.4f}")

plt.figure(figsize=(8,5))
plt.plot(train_losses, label='Training Loss')
plt.plot(val_losses, label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.title('Training vs Validation Loss')
plt.show()
```

```
Epoch [10/50] | Train Loss: 0.6448 | Val Loss: 0.6290
Epoch [20/50] | Train Loss: 0.4371 | Val Loss: 0.3371
```

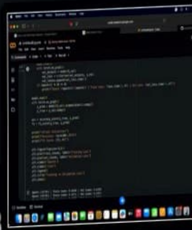
Variables Terminal

[17]
✓ 1s

```python
plt.plot(val_losses, label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.title('Training vs Validation Loss')
plt.show()
```

```
Epoch [10/50] | Train Loss: 0.6448 | Val Loss: 0.6290
Epoch [20/50] | Train Loss: 0.4371 | Val Loss: 0.3371
Epoch [30/50] | Train Loss: 0.3088 | Val Loss: 0.2713
Epoch [40/50] | Train Loss: 0.2222 | Val Loss: 0.2002
Epoch [50/50] | Train Loss: 0.1768 | Val Loss: 0.1569

Final Evaluation:
Accuracy: 94.00%
F1 Score: 0.9375
```



8:23

Edit View History Bookmarks Window Help Tue Nov 4 8:24 PM

colab.research.google.com

chnology diagrams - Google Search | SRM Student Portal | Untitled8.ipynb - Colab | Asmi0612/Deeplearning_RA2311047010012: Lab note obse

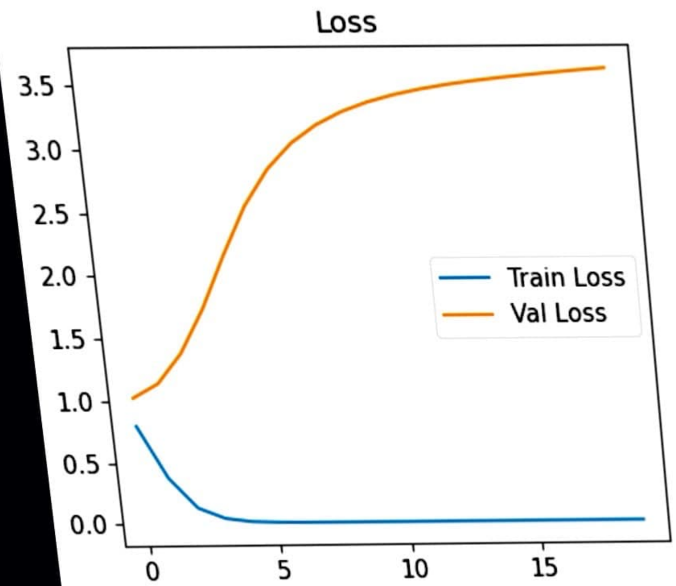
tled8.ipynb ☆ Saving failed since 7:38 PM

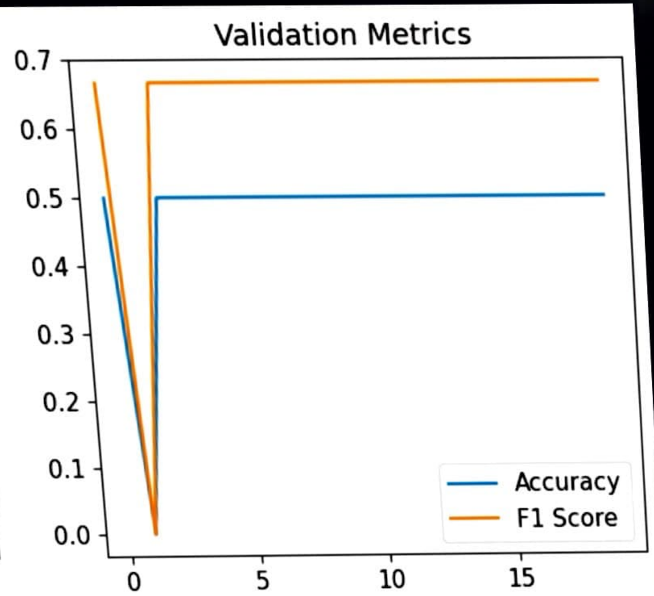
it View Insert Runtime Tools Help

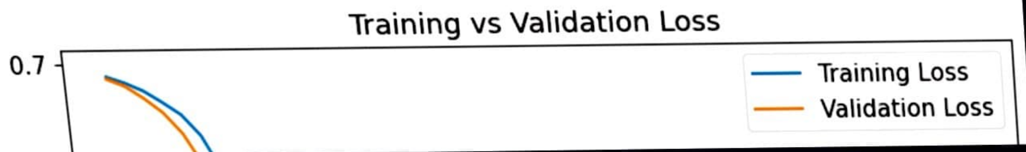
+ Code + Text ▷ Run all

Share A

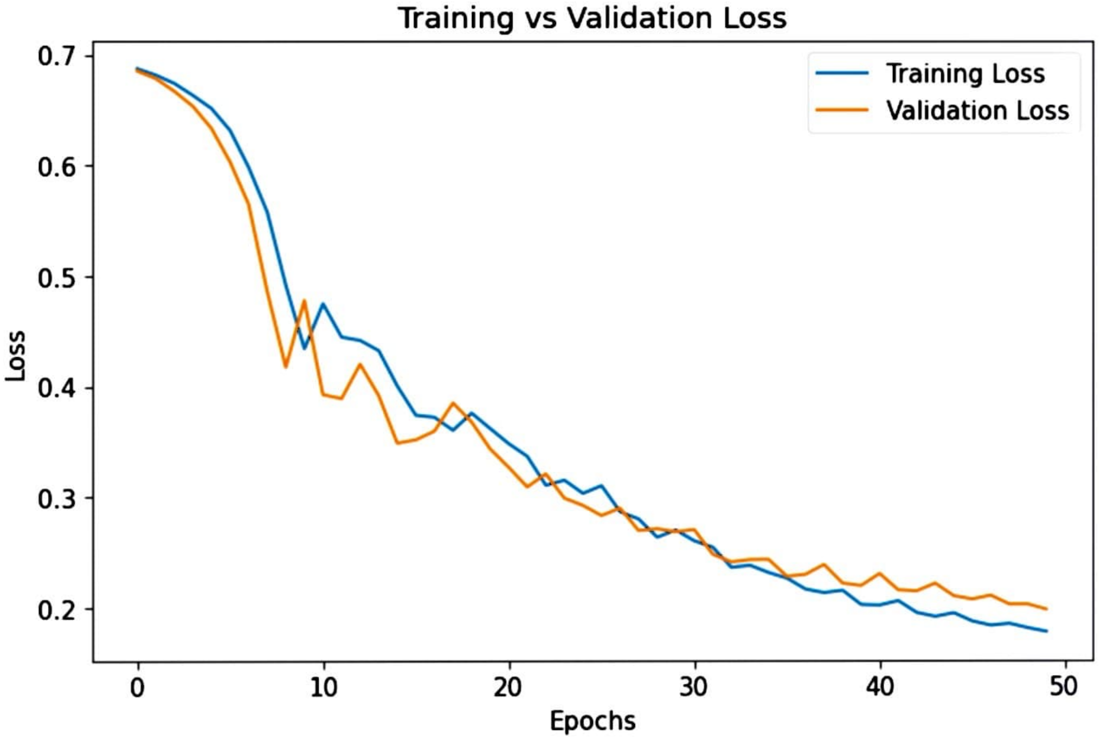
RAM
Disk

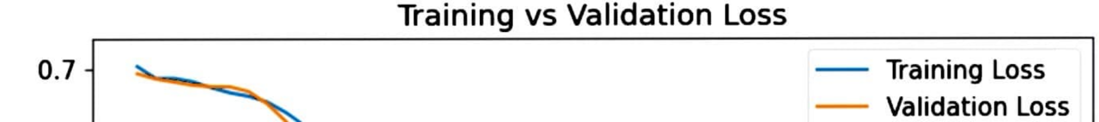







0
5
10
15
0
5
10
15


Training vs Validation Loss
Training Loss
Validation Loss
Loss
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
50
Epochs


Training vs Validation Loss
Training Loss
Validation Loss
0.7

Safari File Edit View History Bookmarks Window Help Tue Nov 4 8:24 PM

colab.research.google.com

clean coal technology diagrams - Google Search | SRM Student Portal | Untitled8.ipynb - Colab | Asmi0612/Deeplearning_RA2311047010012: Lab note obser...

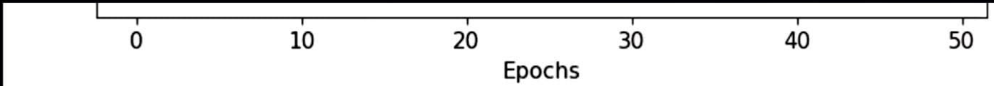
Untitled8.ipynb Saving failed since 7:38 PM

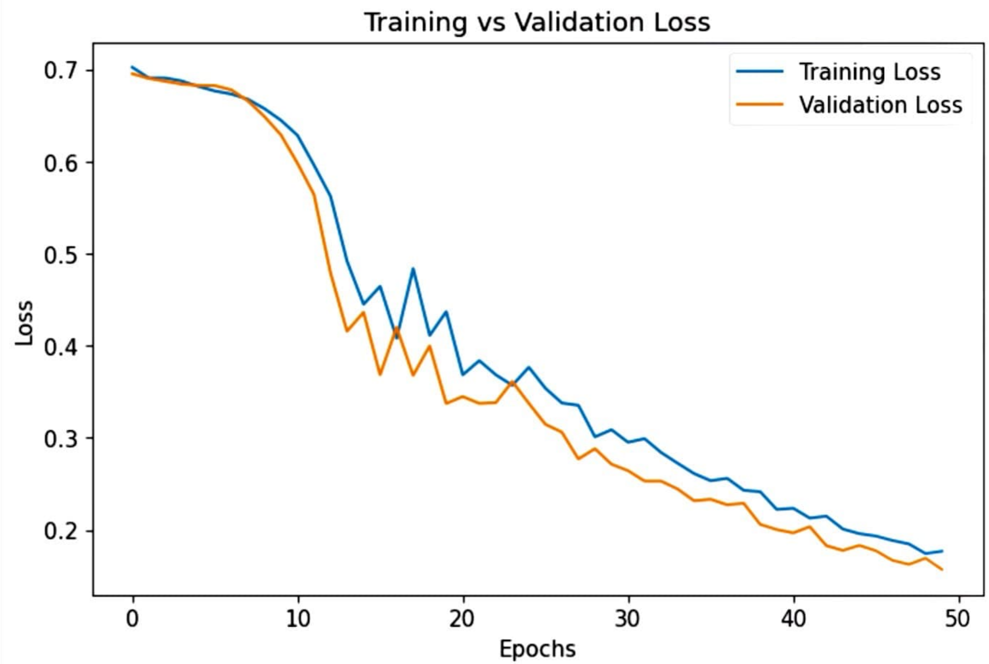
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

RAM Disk

Share





[11] 0s

```
image_files = ['/path/to/your/image1.jpg', '/other/path/image2.jpg']
```

Variables Terminal 8:23