EX.NO:09
09.10.25

BUILD A RECURRENT NEURAL NETWORK

AIM

To build and train a recurrent neural network to analyze sequential data and understand temporal relationships.

objectives

1. To study the architecture of a recurrent neural network.
2. To implement a simple RNN model using Tensorflow / keras
3. To compare the RNN's performance with LSTM
4. To visualize and analyze how the model learns from sequential data.

pseudocode

Start

1. Import necessary libraries (tensorflow / keras, numpy, matplotlib)
2. Load and preprocess a sequence dataset
3. Convert input sequences to padded vectors
4. Define RNN model

Embedding layer (for text input)

SimpleRNN layer (units = 64)

Dense layer with Relu activation

Output dense layer with Sigmoid Activation.

5. Compile model
   Optimizer : Adam
   Loss : Binary crossentropy
   Metrics : Accuracy

6. Train model using training data and validate with test data

7. Evaluate model performance and visualize training curves

8. predict on new sequence input

End.

Observation

1. Training Analysis

The RNN learned to map input sequences to sentiment labels, but training took longer compared to feedforward networks.

2. performance comparison

Accuracy of RNN was moderate (~80.1%), slightly lower than LSTM.

3. Model learning

the hidden state updated dynamically with each timestep showing how RNNs handle temporal data

4. Loss and Accuracy Trends

Both training and validation loss decreased over epochs.

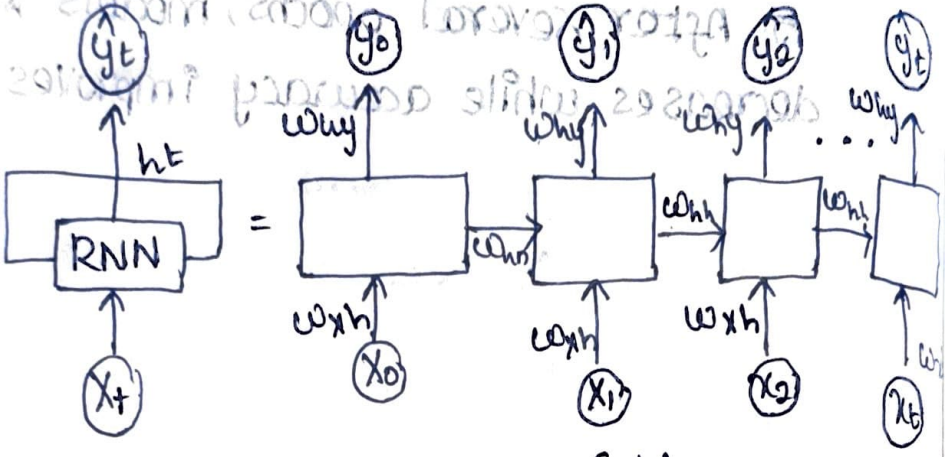
5. Limitations and Insights

Advanced variants like LSTM and GRU address these issues efficiently.

Result

implemented Build a Recurrent neural network.

RNN architecture



$h_t = f_w(x_t, h_{t-1})$ ← function with weight

- h_t → cell state
- x_t → Inputs
- h_{t-1} → old state

Output vector

$$\hat{y}_t = W_{hy}^T h_t$$

update vector

$$h_t = \tanh(W_{hh}^T h_{t-1} + W_{xh}^T x_t)$$

Input vector

x_t

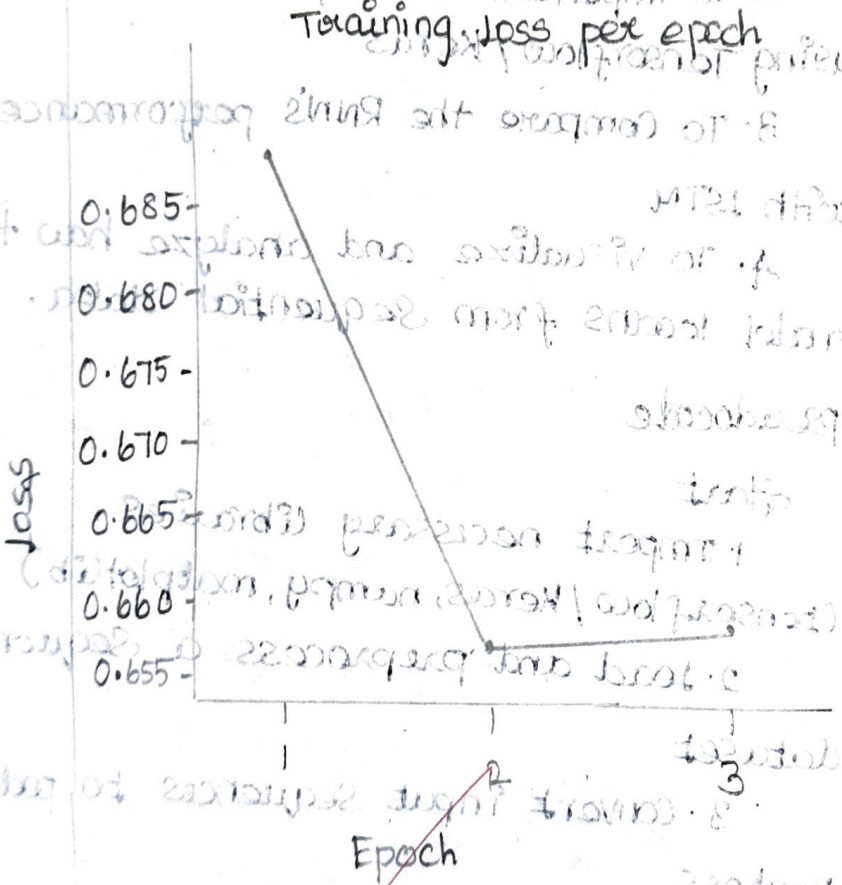
Output

Epoch 1 ; Training loss : 0.6868

Epoch 2 , Training loss : 0.6562

Epoch 3 , Training loss : 0.6567

Test Accuracy : 51.18%



New Tab | L9.ipynb (8) - Jupyter | What is CNN | fully connected layer | cnn graph - Google S | IMDB Movies Dataset

Not secure 10.1.38.19/user/ra2311047010011/lab/tree/DEEP%20LEARNING/L9.ipynb

Verify it's you

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DEEP LEARNING /

Name	Last Modified
data	last month
Breast_canc...	last month
Exp2.ipynb	last month
IMDB Datas...	18 minutes ago
imdb_top_1...	17 minutes ago
installpytor...	42 seconds ago
L_3.ipynb	last month
L1.1.ipynb	last month
L1.ipynb	last month
L2.ipynb	last month
L3.ipynb	last month
L4.2.ipynb	42 seconds ago
L4.ipynb	42 seconds ago
L5.ipynb	42 seconds ago
L6.ipynb	42 seconds ago
L9.ipynb	42 seconds ago
Lab7.ipynb	42 seconds ago
Untitled.ipy...	last month

L4.ipynb | installpytorch.ipynb | L4.2.ipynb | L5.ipynb | L6.ipynb | Lab7.ipynb | L9.ipynb

Code | Notebook | Python 3 (ipykernel)

[1]:
```
import pandas as pd
```

[7]:
```
df=pd.read_csv("IMDB Dataset.csv")
```

[8]:
```
df.head(5)
```

[8]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

[9]:
```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

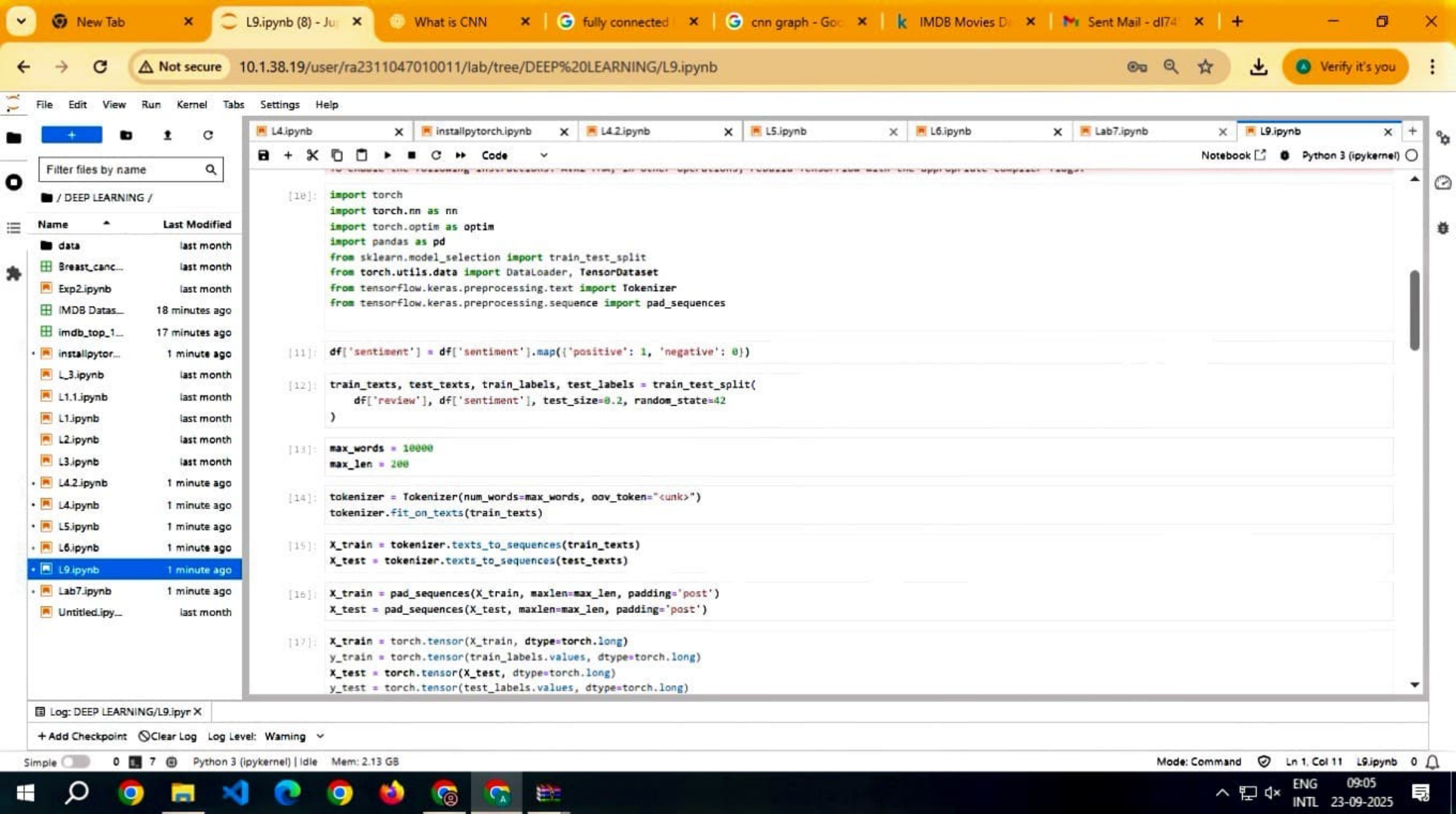
```
2025-09-23 08:48:52.757119: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1758597532.923064  381952 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
E0000 00:00:1758597532.996783  381952 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
W0000 00:00:1758597534.217134  381952 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
W0000 00:00:1758597534.217186  381952 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
W0000 00:00:1758597534.217194  381952 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
W0000 00:00:1758597534.217200  381952 computation_placer.cc:177] computation placer already registered. Please check linkage and avoid linking the same target more than once.
2025-09-23 08:48:54.309572: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
```

Log: DEEP LEARNING/L9.ipyr

+ Add Checkpoint | Clear Log | Log Level: Warning

Simple | 0 | 7 | Python 3 (ipykernel) | Idle | Mem: 2.13 GB | Mode: Command | Ln 1, Col 11 | L9.ipynb | 0

ENG INTL 09:04 23-09-2025


New Tab
L9.ipynb (8) - Ju
What is CNN
fully connected
cnn graph - Goo
IMDB Movies D
Sent Mail - dl74
Not secure
10.1.38.19/user/ra2311047010011/lab/tree/DEEP%20LEARNING/L9.ipynb
Verify it's you
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ DEEP LEARNING /
Name
Last Modified
data last month
Breast_canc... last month
Exp2.ipynb last month
IMDB Datas... 18 minutes ago
imdb_top_1... 17 minutes ago
installpytor... 1 minute ago
L_3.ipynb last month
L1.1.ipynb last month
L1.ipynb last month
L2.ipynb last month
L3.ipynb last month
L4.2.ipynb 1 minute ago
L4.ipynb 1 minute ago
L5.ipynb 1 minute ago
L6.ipynb 1 minute ago
L9.ipynb 1 minute ago
Lab7.ipynb 1 minute ago
Untitled.ipy... last month
L4.ipynb
installpytorch.ipynb
L4.2.ipynb
L5.ipynb
L6.ipynb
Lab7.ipynb
L9.ipynb
Code
Notebook
Python 3 (ipykernel)
[10]:
import torch
import torch.nn as nn
import torch.optim as optim
import pandas as pd
from sklearn.model_selection import train_test_split
from torch.utils.data import DataLoader, TensorDataset
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
[11]:
df['sentiment'] = df['sentiment'].map({'positive': 1, 'negative': 0})
[12]:
train_texts, test_texts, train_labels, test_labels = train_test_split(
df['review'], df['sentiment'], test_size=0.2, random_state=42
)
[13]:
max_words = 10000
max_len = 200
[14]:
tokenizer = Tokenizer(num_words=max_words, oov_token="<unk>")
tokenizer.fit_on_texts(train_texts)
[15]:
X_train = tokenizer.texts_to_sequences(train_texts)
X_test = tokenizer.texts_to_sequences(test_texts)
[16]:
X_train = pad_sequences(X_train, maxlen=max_len, padding='post')
X_test = pad_sequences(X_test, maxlen=max_len, padding='post')
[17]:
X_train = torch.tensor(X_train, dtype=torch.long)
y_train = torch.tensor(train_labels.values, dtype=torch.long)
X_test = torch.tensor(X_test, dtype=torch.long)
y_test = torch.tensor(test_labels.values, dtype=torch.long)
Log: DEEP LEARNING/L9.ipyr
+ Add Checkpoint
Clear Log
Log Level: Warning
Simple
Python 3 (ipykernel) | Idle
Mem: 2.13 GB
Mode: Command
Ln 1, Col 11
L9.ipynb
ENG
INTL
09:05
23-09-2025

```
train_data = TensorDataset(X_train, y_train)
test_data = TensorDataset(X_test, y_test)
```

```
train_loader = DataLoader(train_data, batch_size=32, shuffle=True)
test_loader = DataLoader(test_data, batch_size=32)
```

```
class RNNClassifier(nn.Module):
    def __init__(self, vocab_size, embed_dim, hidden_dim, output_dim):
        super(RNNClassifier, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embed_dim)
        self.rnn = nn.RNN(embed_dim, hidden_dim, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)

    def forward(self, x):
        embedded = self.embedding(x)
        output, hidden = self.rnn(embedded)
        return self.fc(hidden.squeeze(0))
```

```
vocab_size = max_words
embed_dim = 64
hidden_dim = 128
output_dim = 2
```

```
model = RNNClassifier(vocab_size, embed_dim, hidden_dim, output_dim)
```

```
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

```
train_losses = []

for epoch in range(3):  # train for 3 epochs
    total_loss = 0
```

```
train_losses = []

for epoch in range(3):  # train for 3 epochs
    total_loss = 0
    model.train()
    for texts, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(texts)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()

    avg_loss = total_loss / len(train_loader)
    train_losses.append(avg_loss)
    print(f"Epoch {epoch+1}, Training Loss: {avg_loss:.4f}")
```

```
Epoch 1, Training Loss: 0.6868
Epoch 2, Training Loss: 0.6562
Epoch 3, Training Loss: 0.6567
```

```
model.eval()
correct, total = 0, 0
with torch.no_grad():
    for texts, labels in test_loader:
        outputs = model(texts)
        predictions = torch.argmax(outputs, dim=1)
        correct += (predictions == labels).sum().item()
        total += labels.size(0)

print(f"Test Accuracy: {100 * correct / total:.2f}%")
```

```
Test Accuracy: 57.18%
```

```
import matplotlib.pyplot as plt
```

New Tab | L9.ipynb (8) - Ju | What is CNN | fully connected | cnn graph - Goo | IMDB Movies D | Sent Mail - dl74

Not secure 10.1.38.19/user/ra2311047010011/lab/tree/DEEP%20LEARNING/L9.ipynb

Verify it's you

File Edit View Run Kernel Tabs Settings Help

Filter files by name

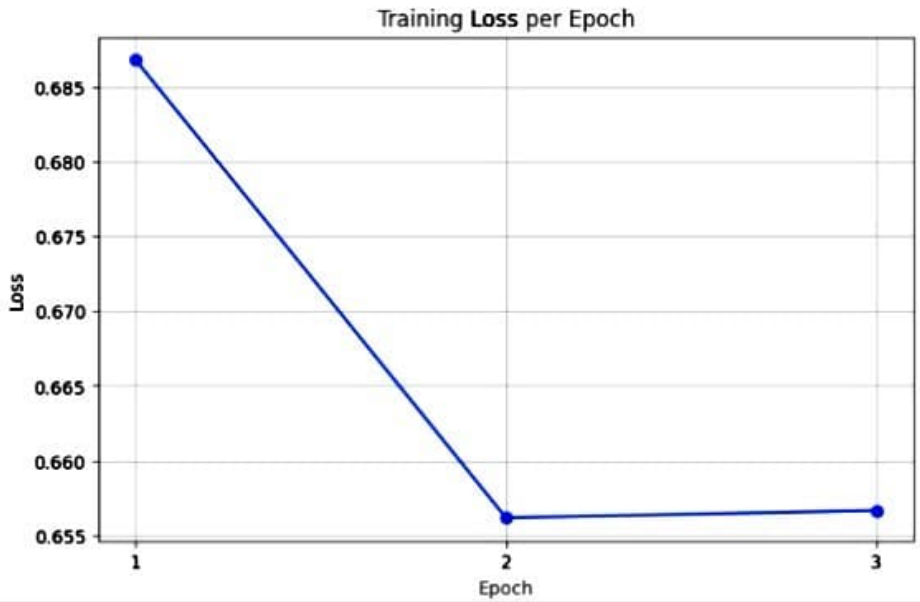
/ DEEP LEARNING /

Name	Last Modified
data	last month
Breast_canc...	last month
Exp2.ipynb	last month
IMDB Datas...	21 minutes ago
imdb_top_1...	20 minutes ago
installpytor...	3 minutes ago
L_3.ipynb	last month
L1.1.ipynb	last month
L1.ipynb	last month
L2.ipynb	last month
L3.ipynb	last month
L4.2.ipynb	3 minutes ago
L4.ipynb	3 minutes ago
L5.ipynb	3 minutes ago
L6.ipynb	3 minutes ago
L9.ipynb	3 minutes ago
Lab7.ipynb	3 minutes ago
Untitled.ipy...	last month

L4.ipynb | installpytorch.ipynb | L4.2.ipynb | L5.ipynb | L6.ipynb | Lab7.ipynb | L9.ipynb

Code | Notebook | Python 3 (ipykernel)

```
[27]: plt.figure(figsize=(8,5))
plt.plot(range(1, len(train_losses)+1), train_losses, marker='o', color='blue')
plt.title("Training Loss per Epoch")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.xticks(range(1, len(train_losses)+1))
plt.grid(True)
plt.show()
```



Log: DEEP LEARNING/L9.ipyr

+ Add Checkpoint | Clear Log | Log Level: Warning

Simple | 0 | 7 | Python 3 (ipykernel) | Idle | Mem: 2.13 GB | Mode: Command | Ln 1, Col 11 | L9.ipynb | 0

ENG INTL 09:07 23-09-2025