```
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense,
Bidirectional, Dropout
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split

# Define the dataset
data = open('/kaggle/input/poem-generation/poem.txt',
encoding="utf8").read()

# Tokenization and Padding for Poetry Generation
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
total_words = len(tokenizer.word_index) + 1

input_sequences = []
for line in data.split('\n'):
    token_list = tokenizer.texts_to_sequences([line])[0]
    for i in range(1, len(token_list)):
        n_gram_sequence = token_list[:i+1]
        input_sequences.append(n_gram_sequence)

max_sequence_len = max([len(x) for x in input_sequences])
input_sequences = np.array(pad_sequences(input_sequences,
maxlen=max_sequence_len, padding='pre'))

X, y = input_sequences[:,:-1], input_sequences[:,-1]
y = tf.keras.utils.to_categorical(y, num_classes=total_words)

# Model Architecture for Poetry Generation
poetry_model = Sequential()
poetry_model.add(Embedding(total_words, 100,
input_length=max_sequence_len-1))
poetry_model.add(Bidirectional(LSTM(150)))
poetry_model.add(Dropout(0.2))
poetry_model.add(Dense(total_words, activation='softmax'))
poetry_model.compile(loss='categorical_crossentropy',
optimizer='adam', metrics=['accuracy'])

# Training for Poetry Generation
poetry_model.fit(X, y, epochs=100, verbose=1)

# Example of Poetry Generation
def generate_poem(seed_text, next_words, max_sequence_len):
    for _ in range(next_words):
        token_list = tokenizer.texts_to_sequences([seed_text])[0]
```

```
        token_list = pad_sequences([token_list],
maxlen=max_sequence_len-1, padding='pre')
        predicted = np.argmax(poetry_model.predict(token_list), axis=-
1)
        output_word = ""
        for word, index in tokenizer.word_index.items():
            if index == predicted:
                output_word = word
                break
        seed_text += " " + output_word
    return seed_text

seed_text = "And"
generated_poem = generate_poem(seed_text, 10, max_sequence_len)
print("Generated Poem:", generated_poem)
```

```
Epoch 1/100
510/510 [==============================] - 28s 47ms/step - loss:
6.9182 - accuracy: 0.0630
Epoch 2/100
510/510 [==============================] - 24s 47ms/step - loss:
6.4735 - accuracy: 0.0740
Epoch 3/100
510/510 [==============================] - 24s 46ms/step - loss:
6.2215 - accuracy: 0.0833
Epoch 4/100
510/510 [==============================] - 24s 47ms/step - loss:
5.9607 - accuracy: 0.0961
Epoch 5/100
510/510 [==============================] - 23s 46ms/step - loss:
5.6746 - accuracy: 0.1080
Epoch 6/100
510/510 [==============================] - 23s 46ms/step - loss:
5.3604 - accuracy: 0.1215
Epoch 7/100
510/510 [==============================] - 27s 53ms/step - loss:
5.0384 - accuracy: 0.1365
Epoch 8/100
510/510 [==============================] - 26s 51ms/step - loss:
4.7001 - accuracy: 0.1567
Epoch 9/100
510/510 [==============================] - 26s 52ms/step - loss:
4.3543 - accuracy: 0.1866
Epoch 10/100
510/510 [==============================] - 25s 49ms/step - loss:
4.0165 - accuracy: 0.2227
Epoch 11/100
510/510 [==============================] - 25s 49ms/step - loss:
3.6769 - accuracy: 0.2702
Epoch 12/100
```

```
510/510 [==============================] - 25s 49ms/step - loss:
3.3558 - accuracy: 0.3225
Epoch 13/100
510/510 [==============================] - 24s 48ms/step - loss:
3.0648 - accuracy: 0.3731
Epoch 14/100
510/510 [==============================] - 24s 46ms/step - loss:
2.7846 - accuracy: 0.4191
Epoch 15/100
510/510 [==============================] - 24s 47ms/step - loss:
2.5524 - accuracy: 0.4647
Epoch 16/100
510/510 [==============================] - 24s 47ms/step - loss:
2.3408 - accuracy: 0.5009
Epoch 17/100
510/510 [==============================] - 24s 47ms/step - loss:
2.1495 - accuracy: 0.5413
Epoch 18/100
510/510 [==============================] - 23s 46ms/step - loss:
1.9862 - accuracy: 0.5770
Epoch 19/100
510/510 [==============================] - 24s 47ms/step - loss:
1.8364 - accuracy: 0.6055
Epoch 20/100
510/510 [==============================] - 24s 47ms/step - loss:
1.6946 - accuracy: 0.6325
Epoch 21/100
510/510 [==============================] - 24s 46ms/step - loss:
1.5809 - accuracy: 0.6595
Epoch 22/100
510/510 [==============================] - 23s 46ms/step - loss:
1.4811 - accuracy: 0.6749
Epoch 23/100
510/510 [==============================] - 24s 47ms/step - loss:
1.3985 - accuracy: 0.6919
Epoch 24/100
510/510 [==============================] - 23s 46ms/step - loss:
1.3146 - accuracy: 0.7114
Epoch 25/100
510/510 [==============================] - 24s 46ms/step - loss:
1.2316 - accuracy: 0.7280
Epoch 26/100
510/510 [==============================] - 23s 45ms/step - loss:
1.1711 - accuracy: 0.7391
Epoch 27/100
510/510 [==============================] - 24s 46ms/step - loss:
1.1001 - accuracy: 0.7543
Epoch 28/100
510/510 [==============================] - 24s 46ms/step - loss:
```

```
1.0602 - accuracy: 0.7656
Epoch 29/100
510/510 [==============================] - 24s 46ms/step - loss:
1.0109 - accuracy: 0.7730
Epoch 30/100
510/510 [==============================] - 23s 45ms/step - loss:
0.9737 - accuracy: 0.7800
Epoch 31/100
510/510 [==============================] - 24s 46ms/step - loss:
0.9372 - accuracy: 0.7868
Epoch 32/100
510/510 [==============================] - 24s 46ms/step - loss:
0.8989 - accuracy: 0.7965
Epoch 33/100
510/510 [==============================] - 23s 45ms/step - loss:
0.8805 - accuracy: 0.7975
Epoch 34/100
510/510 [==============================] - 24s 47ms/step - loss:
0.8414 - accuracy: 0.8061
Epoch 35/100
510/510 [==============================] - 24s 46ms/step - loss:
0.8264 - accuracy: 0.8104
Epoch 36/100
510/510 [==============================] - 24s 47ms/step - loss:
0.8046 - accuracy: 0.8171
Epoch 37/100
510/510 [==============================] - 24s 47ms/step - loss:
0.7918 - accuracy: 0.8154
Epoch 38/100
510/510 [==============================] - 27s 52ms/step - loss:
0.7757 - accuracy: 0.8210
Epoch 39/100
510/510 [==============================] - 27s 52ms/step - loss:
0.7636 - accuracy: 0.8210
Epoch 40/100
510/510 [==============================] - 26s 50ms/step - loss:
0.7611 - accuracy: 0.8194
Epoch 41/100
510/510 [==============================] - 26s 51ms/step - loss:
0.7388 - accuracy: 0.8250
Epoch 42/100
510/510 [==============================] - 26s 50ms/step - loss:
0.7302 - accuracy: 0.8274
Epoch 43/100
510/510 [==============================] - 25s 48ms/step - loss:
0.7151 - accuracy: 0.8310
Epoch 44/100
510/510 [==============================] - 25s 49ms/step - loss:
0.7038 - accuracy: 0.8308
```

```
Epoch 45/100
510/510 [==============================] - 24s 48ms/step - loss:
0.7005 - accuracy: 0.8311
Epoch 46/100
510/510 [==============================] - 24s 48ms/step - loss:
0.6950 - accuracy: 0.8321
Epoch 47/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6816 - accuracy: 0.8328
Epoch 48/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6732 - accuracy: 0.8353
Epoch 49/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6799 - accuracy: 0.8340
Epoch 50/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6703 - accuracy: 0.8370
Epoch 51/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6696 - accuracy: 0.8365
Epoch 52/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6622 - accuracy: 0.8356
Epoch 53/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6534 - accuracy: 0.8400
Epoch 54/100
510/510 [==============================] - 23s 45ms/step - loss:
0.6504 - accuracy: 0.8389
Epoch 55/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6465 - accuracy: 0.8382
Epoch 56/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6482 - accuracy: 0.8378
Epoch 57/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6464 - accuracy: 0.8367
Epoch 58/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6321 - accuracy: 0.8418
Epoch 59/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6337 - accuracy: 0.8421
Epoch 60/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6316 - accuracy: 0.8418
Epoch 61/100
```

```
510/510 [==============================] - 23s 46ms/step - loss:
0.6210 - accuracy: 0.8414
Epoch 62/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6209 - accuracy: 0.8432
Epoch 63/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6307 - accuracy: 0.8416
Epoch 64/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6204 - accuracy: 0.8435
Epoch 65/100
510/510 [==============================] - 23s 45ms/step - loss:
0.6237 - accuracy: 0.8417
Epoch 66/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6221 - accuracy: 0.8424
Epoch 67/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6147 - accuracy: 0.8415
Epoch 68/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6149 - accuracy: 0.8421
Epoch 69/100
510/510 [==============================] - 23s 45ms/step - loss:
0.6167 - accuracy: 0.8419
Epoch 70/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6090 - accuracy: 0.8442
Epoch 71/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5989 - accuracy: 0.8467
Epoch 72/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6034 - accuracy: 0.8445
Epoch 73/100
510/510 [==============================] - 28s 55ms/step - loss:
0.6041 - accuracy: 0.8448
Epoch 74/100
510/510 [==============================] - 26s 50ms/step - loss:
0.6066 - accuracy: 0.8432
Epoch 75/100
510/510 [==============================] - 26s 51ms/step - loss:
0.5997 - accuracy: 0.8426
Epoch 76/100
510/510 [==============================] - 26s 52ms/step - loss:
0.5939 - accuracy: 0.8440
Epoch 77/100
510/510 [==============================] - 26s 51ms/step - loss:
```

```
0.5939 - accuracy: 0.8457
Epoch 78/100
510/510 [==============================] - 25s 48ms/step - loss:
0.5913 - accuracy: 0.8455
Epoch 79/100
510/510 [==============================] - 24s 48ms/step - loss:
0.5996 - accuracy: 0.8430
Epoch 80/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5898 - accuracy: 0.8449
Epoch 81/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5932 - accuracy: 0.8450
Epoch 82/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5943 - accuracy: 0.8439
Epoch 83/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5861 - accuracy: 0.8452
Epoch 84/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5872 - accuracy: 0.8469
Epoch 85/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5836 - accuracy: 0.8440
Epoch 86/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5880 - accuracy: 0.8446
Epoch 87/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5810 - accuracy: 0.8466
Epoch 88/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5814 - accuracy: 0.8477
Epoch 89/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5829 - accuracy: 0.8448
Epoch 90/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5831 - accuracy: 0.8448
Epoch 91/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5788 - accuracy: 0.8474
Epoch 92/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5816 - accuracy: 0.8467
Epoch 93/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5807 - accuracy: 0.8456
```

```
Epoch 94/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5746 - accuracy: 0.8470
Epoch 95/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5713 - accuracy: 0.8465
Epoch 96/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5747 - accuracy: 0.8467
Epoch 97/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5778 - accuracy: 0.8447
Epoch 98/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5734 - accuracy: 0.8451
Epoch 99/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5729 - accuracy: 0.8473
Epoch 100/100
510/510 [==============================] - 23s 45ms/step - loss:
0.5767 - accuracy: 0.8459
1/1 [==============================] - 1s 809ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 25ms/step
Generated Poem: And the world it can see so now and i were

import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense,
Bidirectional, Dropout
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
import ipywidgets as widgets
from IPython.display import display

# Define the dataset
data = open('/kaggle/input/poem-generation/poem.txt',
encoding="utf8").read()

# Tokenization and Padding for Poetry Generation
```

```python
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
total_words = len(tokenizer.word_index) + 1

input_sequences = []
for line in data.split('\n'):
    token_list = tokenizer.texts_to_sequences([line])[0]
    for i in range(1, len(token_list)):
        n_gram_sequence = token_list[:i+1]
        input_sequences.append(n_gram_sequence)

max_sequence_len = max([len(x) for x in input_sequences])
input_sequences = np.array(pad_sequences(input_sequences,
maxlen=max_sequence_len, padding='pre'))

X, y = input_sequences[:,:-1], input_sequences[:,-1]
y = tf.keras.utils.to_categorical(y, num_classes=total_words)

# Model Architecture for Poetry Generation
poetry_model = Sequential()
poetry_model.add(Embedding(total_words, 100,
input_length=max_sequence_len-1))
poetry_model.add(Bidirectional(LSTM(150)))
poetry_model.add(Dropout(0.2))
poetry_model.add(Dense(total_words, activation='softmax'))
poetry_model.compile(loss='categorical_crossentropy',
optimizer='adam', metrics=['accuracy'])

# Training for Poetry Generation
poetry_model.fit(X, y, epochs=100, verbose=1)

# Function to generate poem
def generate_poem(seed_text, next_words, max_sequence_len):
    for _ in range(next_words):
        token_list = tokenizer.texts_to_sequences([seed_text])[0]
        token_list = pad_sequences([token_list],
maxlen=max_sequence_len-1, padding='pre')
        predicted = np.argmax(poetry_model.predict(token_list), axis=-
1)
        output_word = ""
        for word, index in tokenizer.word_index.items():
            if index == predicted:
                output_word = word
                break
        seed_text += " " + output_word
    return seed_text

# User input widget for poem generation
poem_input = widgets.Text(placeholder='Enter your seed text for poem
generation')
```

```
display(poem_input)

# Button to trigger poem generation
poem_button = widgets.Button(description='Generate Poem')
display(poem_button)

# Output widget for displaying generated poem
poem_output = widgets.Output()
display(poem_output)

def generate_poem_output(b):
    with poem_output:
        poem_output.clear_output()
        seed_text = poem_input.value.strip()
        if seed_text:
            generated_poem = generate_poem(seed_text, 10,
max_sequence_len)
            print("Generated Poem:")
            print(generated_poem)
        else:
            print("Please enter a seed text.")

poem_button.on_click(generate_poem_output)

# Function to perform sentiment analysis
def predict_sentiment(text):
    # Perform sentiment analysis here (replace with your sentiment
analysis model)
    # For demonstration, returning a random sentiment label
    sentiments = ['positive', 'neutral', 'negative']
    return np.random.choice(sentiments)

# User input widget for sentiment analysis
sentiment_input = widgets.Text(placeholder='Enter your text for
sentiment analysis')
display(sentiment_input)

# Button to trigger sentiment analysis
sentiment_button = widgets.Button(description='Analyze Sentiment')
display(sentiment_button)

# Output widget for displaying sentiment analysis result
sentiment_output = widgets.Output()
display(sentiment_output)

def analyze_sentiment(b):
    with sentiment_output:
        sentiment_output.clear_output()
        input_text = sentiment_input.value.strip()
        if input_text:
```

```
            sentiment = predict_sentiment(input_text)
            print("Sentiment:", sentiment)
        else:
            print("Please enter some text for sentiment analysis.")

sentiment_button.on_click(analyze_sentiment)
```

```
Epoch 1/100
510/510 [==============================] - 29s 48ms/step - loss:
6.9309 - accuracy: 0.0622
Epoch 2/100
510/510 [==============================] - 25s 49ms/step - loss:
6.4867 - accuracy: 0.0735
Epoch 3/100
510/510 [==============================] - 24s 46ms/step - loss:
6.2337 - accuracy: 0.0828
Epoch 4/100
510/510 [==============================] - 24s 48ms/step - loss:
5.9488 - accuracy: 0.0982
Epoch 5/100
510/510 [==============================] - 24s 47ms/step - loss:
5.6433 - accuracy: 0.1088
Epoch 6/100
510/510 [==============================] - 24s 47ms/step - loss:
5.3115 - accuracy: 0.1245
Epoch 7/100
510/510 [==============================] - 23s 46ms/step - loss:
4.9807 - accuracy: 0.1435
Epoch 8/100
510/510 [==============================] - 24s 47ms/step - loss:
4.6397 - accuracy: 0.1634
Epoch 9/100
510/510 [==============================] - 24s 46ms/step - loss:
4.2833 - accuracy: 0.1936
Epoch 10/100
510/510 [==============================] - 23s 45ms/step - loss:
3.9416 - accuracy: 0.2340
Epoch 11/100
510/510 [==============================] - 23s 46ms/step - loss:
3.6085 - accuracy: 0.2777
Epoch 12/100
510/510 [==============================] - 23s 46ms/step - loss:
3.2869 - accuracy: 0.3300
Epoch 13/100
510/510 [==============================] - 23s 46ms/step - loss:
2.9928 - accuracy: 0.3796
Epoch 14/100
510/510 [==============================] - 23s 45ms/step - loss:
2.7270 - accuracy: 0.4275
Epoch 15/100
```

```
510/510 [==============================] - 24s 46ms/step - loss:
2.4888 - accuracy: 0.4721
Epoch 16/100
510/510 [==============================] - 24s 47ms/step - loss:
2.2804 - accuracy: 0.5138
Epoch 17/100
510/510 [==============================] - 24s 47ms/step - loss:
2.0955 - accuracy: 0.5509
Epoch 18/100
510/510 [==============================] - 23s 46ms/step - loss:
1.9346 - accuracy: 0.5805
Epoch 19/100
510/510 [==============================] - 23s 46ms/step - loss:
1.7940 - accuracy: 0.6081
Epoch 20/100
510/510 [==============================] - 24s 46ms/step - loss:
1.6685 - accuracy: 0.6326
Epoch 21/100
510/510 [==============================] - 23s 46ms/step - loss:
1.5638 - accuracy: 0.6562
Epoch 22/100
510/510 [==============================] - 24s 46ms/step - loss:
1.4505 - accuracy: 0.6793
Epoch 23/100
510/510 [==============================] - 24s 47ms/step - loss:
1.3673 - accuracy: 0.6948
Epoch 24/100
510/510 [==============================] - 25s 50ms/step - loss:
1.2864 - accuracy: 0.7133
Epoch 25/100
510/510 [==============================] - 29s 56ms/step - loss:
1.2204 - accuracy: 0.7311
Epoch 26/100
510/510 [==============================] - 25s 49ms/step - loss:
1.1560 - accuracy: 0.7431
Epoch 27/100
510/510 [==============================] - 25s 49ms/step - loss:
1.0948 - accuracy: 0.7564
Epoch 28/100
510/510 [==============================] - 24s 48ms/step - loss:
1.0531 - accuracy: 0.7643
Epoch 29/100
510/510 [==============================] - 24s 48ms/step - loss:
1.0175 - accuracy: 0.7712
Epoch 30/100
510/510 [==============================] - 24s 47ms/step - loss:
0.9684 - accuracy: 0.7781
Epoch 31/100
510/510 [==============================] - 24s 47ms/step - loss:
```

```
0.9239 - accuracy: 0.7902
Epoch 32/100
510/510 [==============================] - 24s 47ms/step - loss: 
0.9026 - accuracy: 0.7952
Epoch 33/100
510/510 [==============================] - 24s 47ms/step - loss: 
0.8750 - accuracy: 0.7975
Epoch 34/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.8565 - accuracy: 0.8036
Epoch 35/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.8331 - accuracy: 0.8084
Epoch 36/100
510/510 [==============================] - 24s 47ms/step - loss: 
0.8096 - accuracy: 0.8118
Epoch 37/100
510/510 [==============================] - 24s 47ms/step - loss: 
0.7904 - accuracy: 0.8141
Epoch 38/100
510/510 [==============================] - 23s 46ms/step - loss: 
0.7784 - accuracy: 0.8198
Epoch 39/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.7646 - accuracy: 0.8179
Epoch 40/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.7481 - accuracy: 0.8219
Epoch 41/100
510/510 [==============================] - 24s 47ms/step - loss: 
0.7455 - accuracy: 0.8227
Epoch 42/100
510/510 [==============================] - 23s 46ms/step - loss: 
0.7240 - accuracy: 0.8276
Epoch 43/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.7170 - accuracy: 0.8278
Epoch 44/100
510/510 [==============================] - 24s 47ms/step - loss: 
0.7161 - accuracy: 0.8282
Epoch 45/100
510/510 [==============================] - 23s 46ms/step - loss: 
0.7045 - accuracy: 0.8289
Epoch 46/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.6876 - accuracy: 0.8345
Epoch 47/100
510/510 [==============================] - 24s 46ms/step - loss: 
0.6946 - accuracy: 0.8313
```

```
Epoch 48/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6885 - accuracy: 0.8331
Epoch 49/100
510/510 [==============================] - 23s 45ms/step - loss:
0.6760 - accuracy: 0.8360
Epoch 50/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6714 - accuracy: 0.8383
Epoch 51/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6681 - accuracy: 0.8358
Epoch 52/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6732 - accuracy: 0.8331
Epoch 53/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6605 - accuracy: 0.8366
Epoch 54/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6499 - accuracy: 0.8392
Epoch 55/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6533 - accuracy: 0.8374
Epoch 56/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6444 - accuracy: 0.8392
Epoch 57/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6407 - accuracy: 0.8399
Epoch 58/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6402 - accuracy: 0.8399
Epoch 59/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6423 - accuracy: 0.8375
Epoch 60/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6365 - accuracy: 0.8390
Epoch 61/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6419 - accuracy: 0.8378
Epoch 62/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6287 - accuracy: 0.8410
Epoch 63/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6251 - accuracy: 0.8410
Epoch 64/100
```

```
510/510 [==============================] - 24s 46ms/step - loss:
0.6211 - accuracy: 0.8429
Epoch 65/100
510/510 [==============================] - 23s 46ms/step - loss:
0.6232 - accuracy: 0.8399
Epoch 66/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6205 - accuracy: 0.8417
Epoch 67/100
510/510 [==============================] - 25s 48ms/step - loss:
0.6169 - accuracy: 0.8419
Epoch 68/100
510/510 [==============================] - 27s 52ms/step - loss:
0.6188 - accuracy: 0.8395
Epoch 69/100
510/510 [==============================] - 25s 49ms/step - loss:
0.6097 - accuracy: 0.8429
Epoch 70/100
510/510 [==============================] - 25s 49ms/step - loss:
0.6098 - accuracy: 0.8432
Epoch 71/100
510/510 [==============================] - 25s 49ms/step - loss:
0.6084 - accuracy: 0.8437
Epoch 72/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6063 - accuracy: 0.8440
Epoch 73/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6039 - accuracy: 0.8428
Epoch 74/100
510/510 [==============================] - 24s 48ms/step - loss:
0.5929 - accuracy: 0.8454
Epoch 75/100
510/510 [==============================] - 24s 47ms/step - loss:
0.6037 - accuracy: 0.8424
Epoch 76/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5997 - accuracy: 0.8438
Epoch 77/100
510/510 [==============================] - 24s 46ms/step - loss:
0.6018 - accuracy: 0.8431
Epoch 78/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5963 - accuracy: 0.8440
Epoch 79/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5928 - accuracy: 0.8473
Epoch 80/100
510/510 [==============================] - 24s 48ms/step - loss:
```

```
0.5919 - accuracy: 0.8465
Epoch 81/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5913 - accuracy: 0.8448
Epoch 82/100
510/510 [==============================] - 24s 48ms/step - loss:
0.5881 - accuracy: 0.8464
Epoch 83/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5858 - accuracy: 0.8457
Epoch 84/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5878 - accuracy: 0.8456
Epoch 85/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5892 - accuracy: 0.8456
Epoch 86/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5854 - accuracy: 0.8472
Epoch 87/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5897 - accuracy: 0.8458
Epoch 88/100
510/510 [==============================] - 23s 46ms/step - loss:
0.5896 - accuracy: 0.8443
Epoch 89/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5822 - accuracy: 0.8467
Epoch 90/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5804 - accuracy: 0.8469
Epoch 91/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5784 - accuracy: 0.8458
Epoch 92/100
510/510 [==============================] - 24s 46ms/step - loss:
0.5787 - accuracy: 0.8472
Epoch 93/100
510/510 [==============================] - 24s 47ms/step - loss:
0.5743 - accuracy: 0.8464
Epoch 94/100
510/510 [==============================] - 24s 48ms/step - loss:
0.5757 - accuracy: 0.8464
Epoch 95/100
510/510 [==============================] - 25s 49ms/step - loss:
0.5789 - accuracy: 0.8462
Epoch 96/100
510/510 [==============================] - 25s 49ms/step - loss:
0.5767 - accuracy: 0.8466
```

```
Epoch 97/100
510/510 [==============================] - 26s 51ms/step - loss: 0.5734 - accuracy: 0.8476
Epoch 98/100
510/510 [==============================] - 25s 49ms/step - loss: 0.5770 - accuracy: 0.8450
Epoch 99/100
510/510 [==============================] - 25s 49ms/step - loss: 0.5704 - accuracy: 0.8477
Epoch 100/100
510/510 [==============================] - 24s 47ms/step - loss: 0.5698 - accuracy: 0.8470
```


{"model_id":"b3cb494a1d414cafb0c7e61546322dde","version_major":2,"version_minor":0}

{"model_id":"333784d6d2b3492aacb63eef00a324bf","version_major":2,"version_minor":0}

{"model_id":"50bc58bcb1024d5aa0bfe2d486aece68","version_major":2,"version_minor":0}

{"model_id":"b3e904e8de0d4a86aa45fc3c6c4fc1b2","version_major":2,"version_minor":0}

{"model_id":"654b3dcbe377403397119bc8b485a73c","version_major":2,"version_minor":0}

{"model_id":"5d47d6ad4dbe454999a49f07521b71de","version_major":2,"version_minor":0}