HW2_RNN_Negative_Reviews

October 23, 2023

0.1 Homework 2: RNN

Problem 1: Text Generation (fake reviews) Text generation serves various purposes like machine translation, chatbots, virtual assistants, and AIGC. To understand the text generation process with a language model, consider it as an iterative approach. Initially, we predict the first word from the input sequence and subsequently use that predicted word in the input to generate the second word. Repeat the process until done.

If we recall, we focused on letter generation in class. Now, using the review data in HW1, let's build upon the in-class examples to create a simulated review using a GRU or LSTM model. For simplicity's sake, let's concentrate on coding the training process without delving into hyperparameter tuning or model evaluations. We'll generate both a negative review (combining 1-star, 2-star, and 3-star comments) and a positive review (utilizing 4-star and 5-star comments). Essentially, our goal is to generate tokens instead of individual letters.

1 Negative reviews next word prediction

[65]:
```
# import google drive

from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

1.1 Read the Amazon comments data

[67]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import string


import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

```python
from nltk.stem import WordNetLemmatizer

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
```

[4]:
```python
# import nltk
# nltk.download('all')
```

[68]:
```python
os.chdir('/content/drive/MyDrive/Text Analytics/HW2')
```

[69]:
```python
df = pd.read_csv('Amazon_Comments.csv',  delimiter="^", header=None,␣
 ↪names=["No", "Title", "Date", "Bool", "Review", "Rating"])

df = df.reset_index(drop=True)

df.set_index('No', inplace=True)

df.head()
```

[69]:
```
                                                Title        Date   Bool  \
No
1   These are hands down the best quality bands fo…  2016-01-16  False
2                                 High Quality Bands  2016-01-22  False
3                                         Five Stars  2015-12-27  False
4   The resistance is great. I would agree that th…  2016-01-13  False
5                               Good quality product  2016-01-20  False

                                               Review  Rating
No
1    These are hands down the best quality bands f…     5.0
2    I just got this set yesterday as well as a se…     5.0
3    My husband uses these and finds them to be go…     5.0
4    I got these for Christmas and have been using…     4.0
5    Haven\t had it long enough to use all of the …     5.0
```

[70]:
```python
# Drop Title, date, bool from df

df1 = df.copy()

df1.drop(columns=['Title', 'Date', 'Bool'], inplace=True)

df1.head()
```

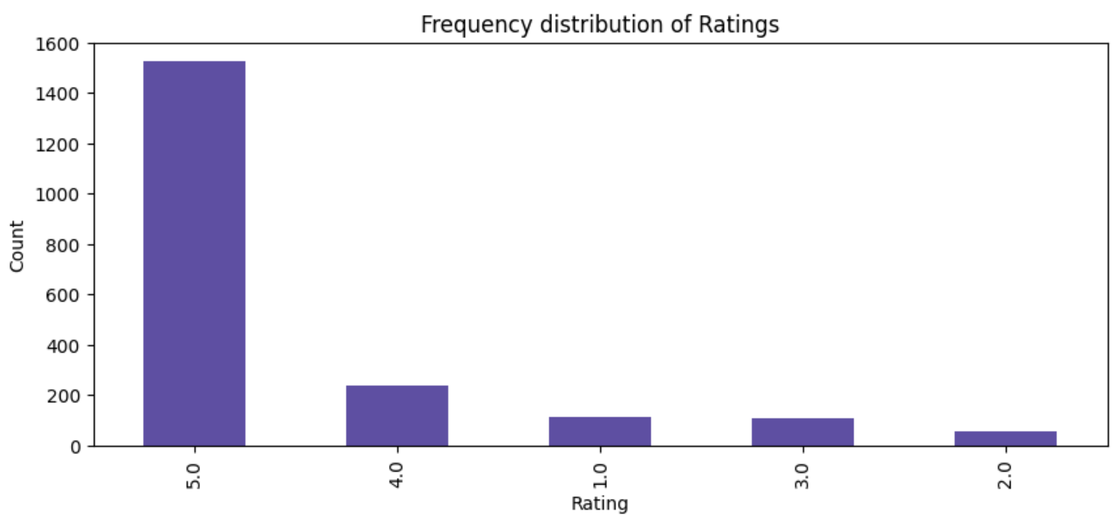
[70]:

```
                                                Review  Rating
No
1    These are hands down the best quality bands f…     5.0
2    I just got this set yesterday as well as a se…     5.0
3    My husband uses these and finds them to be go…     5.0
4    I got these for Christmas and have been using…     4.0
5    Haven\t had it long enough to use all of the …     5.0
```

1.1.1 Visualize the frequency distribution of Ratings

[78]:

```python
# Visualize the frequency distribution of Ratings

df1['Rating'].value_counts().plot(kind='bar', colormap="Spectral_r",␣
 ↪figsize=(10,4))
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Frequency distribution of Ratings')
plt.show()
```



The highest number of ratings in the Amazon review data consist of 5 star ratings followed by 4 star and 1 star ratings respectively.

1.2 Label the data as positive and negative

[71]:

```python
# Add sentiment column -- If Rating =1 or 2 or 3 then label that review as␣
 ↪negative else Flag as Positive

df1['sentiment'] = df['Rating'].map({1.0: 'Negative', 2.0: 'Negative', 3.0:␣
 ↪'Negative', 4.0: 'Positive', 5.0: 'Positive'})
```

```python
df1
```

[71]:

```
                                                 Review  Rating sentiment
No
1     These are hands down the best quality bands f…     5.0  Positive
2     I just got this set yesterday as well as a se…     5.0  Positive
3     My husband uses these and finds them to be go…     5.0  Positive
4     I got these for Christmas and have been using…     4.0  Positive
5     Haven\t had it long enough to use all of the …     5.0  Positive
…                                                   …     …       …
2034                            Just 1 like Nonna\s!     5.0  Positive
2035                                    Works great!     5.0  Positive
2036                                       very good     5.0  Positive
2037                                           great     5.0  Positive
2038  material changed and effect so easy. couple u…     1.0  Negative

[2038 rows x 3 columns]
```

[72]:

```python
# Label the positive sentiment as 1 and negative sentiment as 0

df1['Label'] = df1['sentiment'].map({'Positive': 1, 'Negative': 0})
df1
```

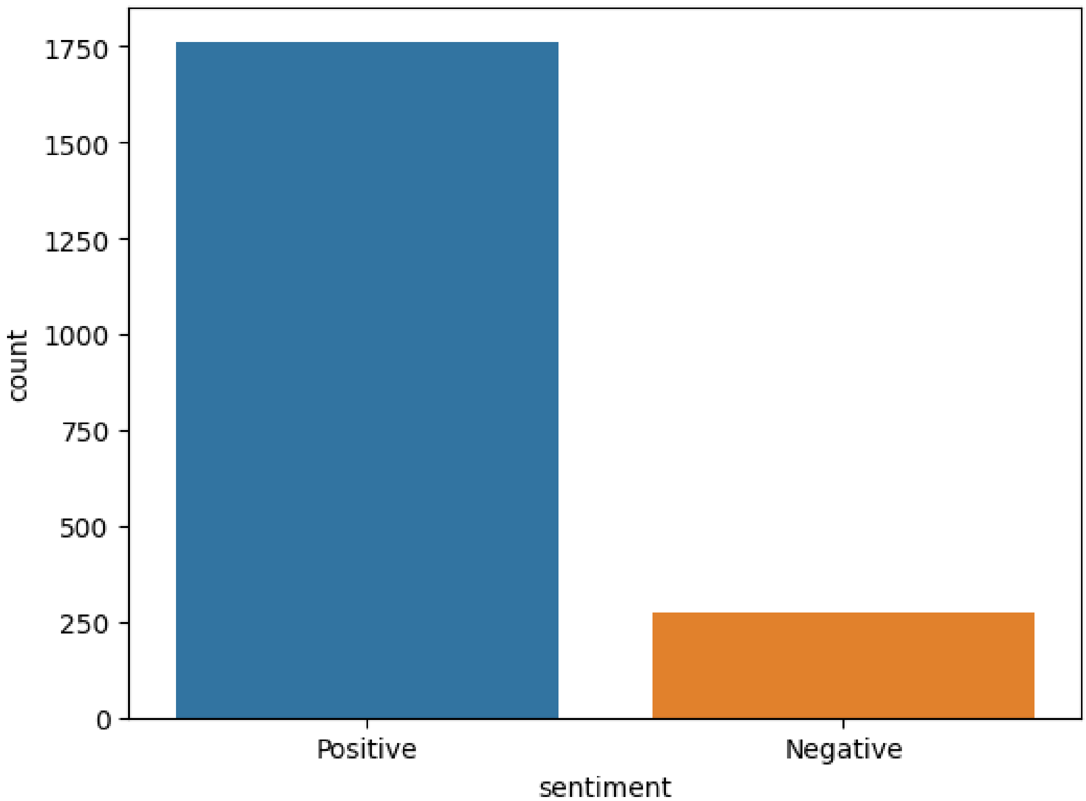
[72]:

```
                                                 Review  Rating sentiment  \
No
1     These are hands down the best quality bands f…     5.0  Positive
2     I just got this set yesterday as well as a se…     5.0  Positive
3     My husband uses these and finds them to be go…     5.0  Positive
4     I got these for Christmas and have been using…     4.0  Positive
5     Haven\t had it long enough to use all of the …     5.0  Positive
…                                                   …     …       …
2034                            Just 1 like Nonna\s!     5.0  Positive
2035                                    Works great!     5.0  Positive
2036                                       very good     5.0  Positive
2037                                           great     5.0  Positive
2038  material changed and effect so easy. couple u…     1.0  Negative

      Label
No
1         1
2         1
3         1
4         1
5         1
…     …
2034      1
2035      1
```

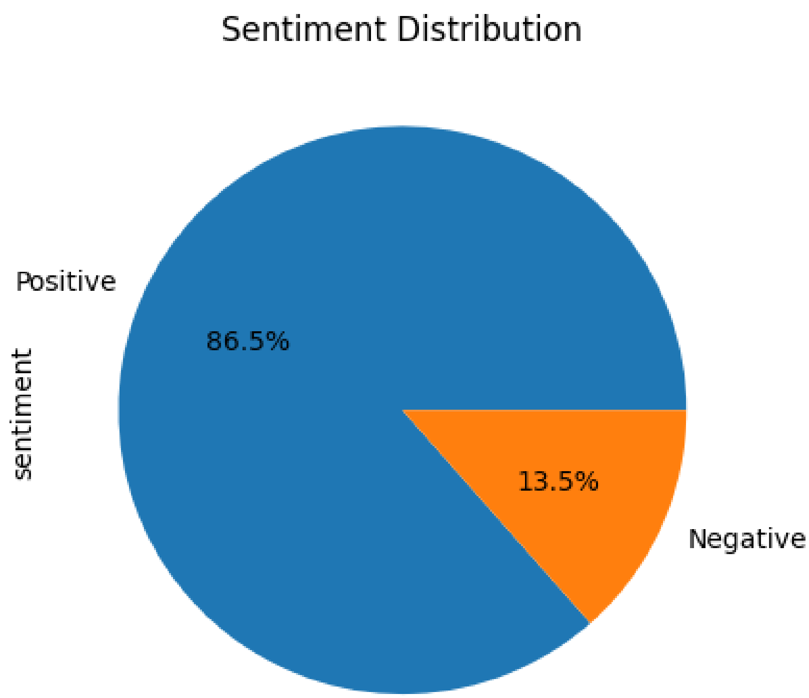
```
2036      1
2037      1
2038      0

[2038 rows x 4 columns]
```

1.3 Visualize the distribution of the Sentiment

[10]:
```python
# Visualize a count plot for the sentiment column with sns.countplot

import seaborn as sns
sns.countplot(x='sentiment', data=df1)
plt.show()
```



[11]:
```python
# Visualize a pie chart for the sentiment column

df1.sentiment.value_counts().plot.pie(autopct='%1.1f%%')
plt.title('Sentiment Distribution')
plt.show()
```



1.4 Data Cleaning

[12]:
```
# Clean the review data

# Function to clean and preprocess text
def clean_text(text):

    # Remove special characters and numbers
    text = re.sub(r'[^a-zA-Z\s]', '', text)

    # Convert text to lowercase
    text = text.lower()

    # Remove punctuation
    text = ''.join([char for char in text if char not in string.punctuation])


    # Tokenize text
    tokens = nltk.word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
```

```
    tokens = [word for word in tokens if word not in stop_words and not word.
↪isdigit()]

    #Stemming (you can replace with lemmatization if preferred)
    #stemmer = PorterStemmer()

    #tokens = [stemmer.stem(word) for word in tokens]

    # Create a lemmatizer object.
    lemmatizer = WordNetLemmatizer()

   #Lemmatization
    tokens = [lemmatizer.lemmatize(word) for word in tokens]

    # Reconstruct cleaned text
    cleaned_text = ' '.join(tokens)

    return cleaned_text
```

[13]:
```
# Apply the clean_text function to each review in the DataFrame

df2 = df1.copy()

df2['Clean_Review'] = df1['Review'].apply(clean_text)

# Print the cleaned reviews

df2.head()
```

[13]:
```
                                               Review  Rating sentiment  \
No
1    These are hands down the best quality bands f…     5.0  Positive
2    I just got this set yesterday as well as a se…     5.0  Positive
3    My husband uses these and finds them to be go…     5.0  Positive
4    I got these for Christmas and have been using…     4.0  Positive
5    Haven\t had it long enough to use all of the …     5.0  Positive

    Label                                      Clean_Review
No
1       1  hand best quality band money year old male wan…
2       1  got set yesterday well set another company cou…
3       1  husband us find good arthritis stretching exer…
4       1  got christmas using multiple day week since re…
5       1  havent long enough use component far im impres…
```

[14]:
```python
# Generate tokens for Clean_Review

df2['Clean_review_Tokens'] = df2['Clean_Review'].apply(nltk.word_tokenize)

df2
```

[14]:
```
                                                 Review  Rating sentiment  \
No
1     These are hands down the best quality bands f…     5.0  Positive
2     I just got this set yesterday as well as a se…     5.0  Positive
3     My husband uses these and finds them to be go…     5.0  Positive
4     I got these for Christmas and have been using…     4.0  Positive
5     Haven\t had it long enough to use all of the …     5.0  Positive
…                                                  …     …       …
2034                            Just 1 like Nonna\s!     5.0  Positive
2035                                    Works great!     5.0  Positive
2036                                       very good     5.0  Positive
2037                                           great     5.0  Positive
2038  material changed and effect so easy. couple u…     1.0  Negative

      Label                                       Clean_Review  \
No
1         1  hand best quality band money year old male wan…
2         1  got set yesterday well set another company cou…
3         1  husband us find good arthritis stretching exer…
4         1  got christmas using multiple day week since re…
5         1  havent long enough use component far im impres…
…       …                                                  …
2034      1                                       1 like nonnas
2035      1                                          work great
2036      1                                                good
2037      1                                               great
2038      0  material changed effect easy couple usage alum…

                                   Clean_review_Tokens
No
1     [hand, best, quality, band, money, year, old, …
2     [got, set, yesterday, well, set, another, comp…
3     [husband, us, find, good, arthritis, stretchin…
4     [got, christmas, using, multiple, day, week, s…
5     [havent, long, enough, use, component, far, im…
…                                                  …
2034                                 [1, like, nonnas]
2035                                     [work, great]
2036                                            [good]
2037                                           [great]
2038  [material, changed, effect, easy, couple, usag…
```

```
[2038 rows x 6 columns]
```

1.5 Filter the Negative Reviews Data

[20]:
```python
# Subset the negative sentiment data

negative_df = df2[df2['sentiment'] == 'Negative']
negative_df
```

[20]:
```
                                                  Review  Rating sentiment  \
No
13     They tore up after a little over a year, and …     1.0  Negative
29     The product is good, the problem is I didn\t …     2.0  Negative
40     Is a good price for the quality and presentat…     3.0  Negative
52      The quality of the rubber band is not vey good.     2.0  Negative
68     What a waste of money, first off let me state…     1.0  Negative
…                                                    …     …       …
1988   The coffee is great, but the bottom of the ha…     3.0  Negative
1991   I did not realize I was buying the 2 ounce, i…     1.0  Negative
1992   I bought it to replace my 6 cup. I didn\t rea…     3.0  Negative
1993   The coffee funnel was bent and not round at t…     3.0  Negative
2038   material changed and effect so easy. couple u…     1.0  Negative

      Label                                    Clean_Review  \
No
13        0  tore little year im strong poor quality also o…
29        0  product good problem didnt get yellow green ba…
40        0  good price quality presentation uncomfortable use
52        0                       quality rubber band vey good
68        0  waste money first let state product may work e…
…       …                                                …
1988      0  coffee great bottom handle melted electric coi…
1991      0  realize buying ounce say one cup actually cup …
1992      0  bought replace cup didnt realize cup one arriv…
1993      0  coffee funnel bent round bottom point thank he…
2038      0  material changed effect easy couple usage alum…

                                    Clean_review_Tokens
No
13     [tore, little, year, im, strong, poor, quality…
29     [product, good, problem, didnt, get, yellow, g…
40     [good, price, quality, presentation, uncomfort…
52                   [quality, rubber, band, vey, good]
68     [waste, money, first, let, state, product, may…
…                                                    …
1988   [coffee, great, bottom, handle, melted, electr…
```

```
1991   [realize, buying, ounce, say, one, cup, actual…
1992   [bought, replace, cup, didnt, realize, cup, on…
1993   [coffee, funnel, bent, round, bottom, point, t…
2038   [material, changed, effect, easy, couple, usag…

[275 rows x 6 columns]
```

1.6 Unigram Token Frequency for Negative Data

[21]:
```python
import collections
from collections import Counter
from itertools import chain

word_tokenize = nltk.word_tokenize

# Tokenize the text column

# Print the tokenized text
corpus = negative_df['Clean_review_Tokens']
corpus = corpus.tolist()
# Flatten list of lists to a single list
tokens = list(chain(*corpus))
unique_freq = collections.Counter(tokens)
# Count each unique element
unique_freq_df = pd.DataFrame.from_dict(unique_freq, orient='index').
 ↪reset_index() # Convert to dataframe
# Rename columns
unique_freq_df = unique_freq_df.rename(columns={'index': 'Token', 0: 'Count'})
# Sort by count
unique_freq_df.sort_values('Count', ascending=False, inplace=True)
unique_freq_df = unique_freq_df

unique_freq_df1 = unique_freq_df.reset_index(drop=True)
unique_freq_df2 = unique_freq_df1.set_index("Token")
print(len(unique_freq_df))
unique_freq_df2
```

```
1916
```

[21]:
```
        Count
Token
one        64
work       63
use        62
filter     55
water      54
…          …
```

```
christmas      1
gifted         1
operates       1
sloppy         1
deserve        1

[1916 rows x 1 columns]
```

[22]:

```python
# Storing the frequencies of unigrams

freq1 = unique_freq_df['Count'].reset_index(drop=True)
print(type(freq1))
freq1
```

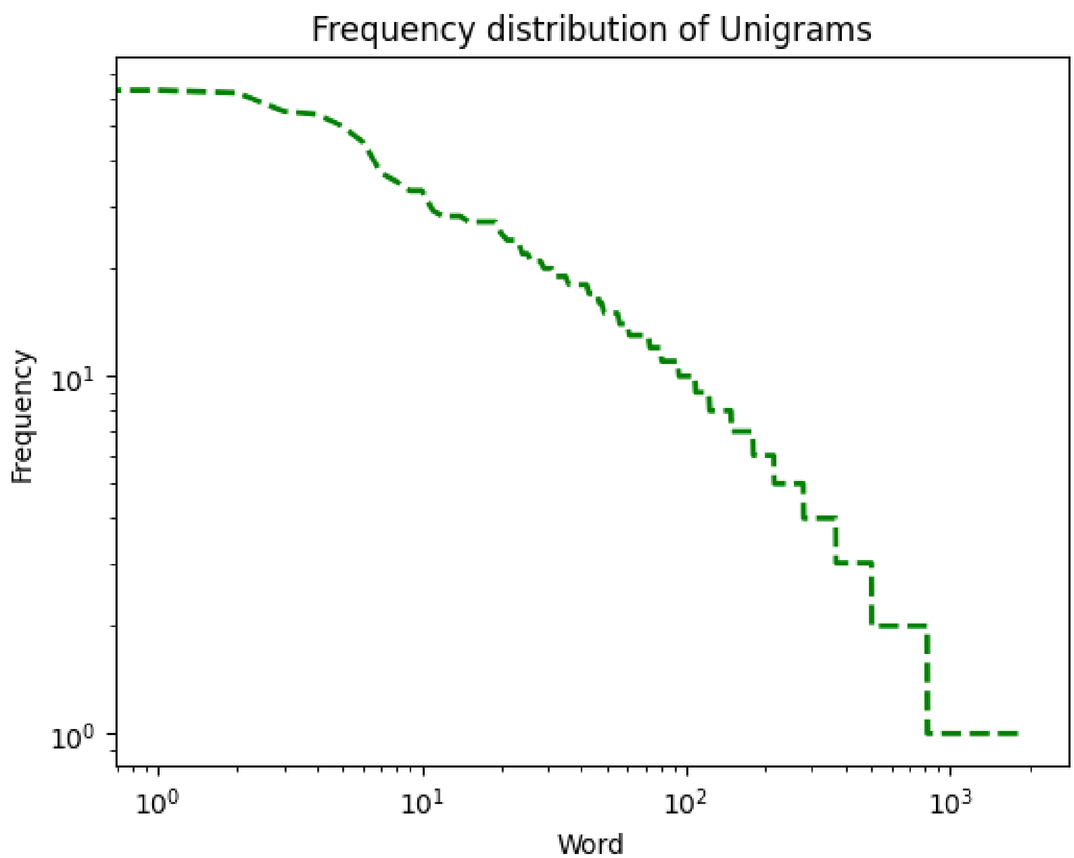
```
<class 'pandas.core.series.Series'>
```

[22]:

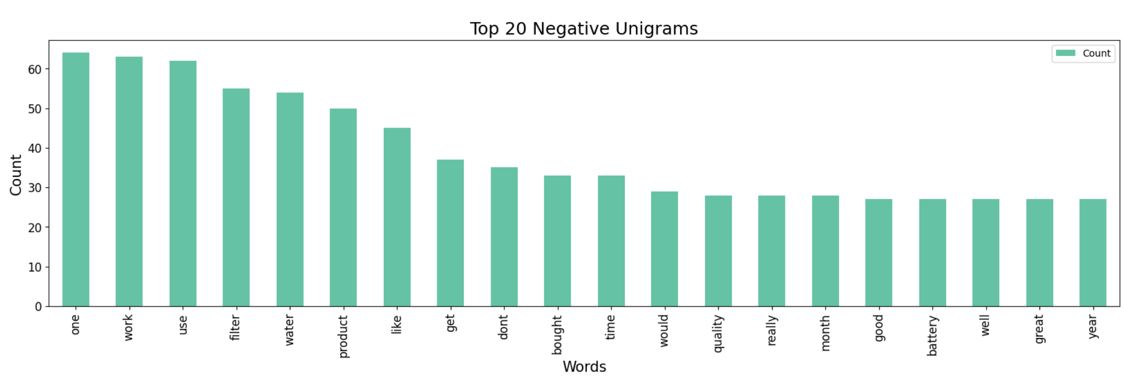
```
0       64
1       63
2       62
3       55
4       54
        ..
1911     1
1912     1
1913     1
1914     1
1915     1
Name: Count, Length: 1916, dtype: int64
```

[23]:

```python
# Plotting the log scale of frequencies of unigrams

plt.plot(freq1, color="Green", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Unigrams")
plt.show()
```



[26]:
```
# Plot the df3 Unigrams

unique_freq_df2.head(20).plot(kind='bar', figsize=(20, 5), cmap="Set2")
plt.xlabel('Words', fontsize = 15)
plt.ylabel('Count', fontsize=15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.title('Top 20 Negative Unigrams', fontsize=18)
plt.show()
```



1.7 Bigram Token Frequency for Negative Data

[27]:
```python
# Generate bigrams from df2['Clean_review_Tokens']

bigram_list = [list(nltk.bigrams(text)) for text in␣
 ↪negative_df['Clean_review_Tokens']]

# Create a Counter object to count the frequency of each bigram
bigram_count = collections.Counter(list(chain(*bigram_list)))

# Convert the Counter object to a DataFrame
bigram_df = pd.DataFrame.from_dict(bigram_count, orient='index').reset_index()

# Rename the columns
bigram_df = bigram_df.rename(columns={'index': 'Bigram', 0: 'Count'})

# Sort the DataFrame by frequency in descending order
bigram_df.sort_values('Count', ascending=False, inplace=True)

# Print the top 20 bigrams
bigram_df.head(20)

bigram_df1 = bigram_df.reset_index(drop=True)
bigram_df2 = bigram_df1.set_index("Bigram")
bigram_df2
```

[27]:
```
                           Count
Bigram
(dont, know)                   7
(didnt, work)                  7
(water, filter)                6
(waffle, iron)                 6
(water, pitcher)               6
…                            …
(le, efficiently)              1
(pop, le)                      1
(surprisingly, expensive)      1
(popper, surprisingly)         1
(deserve, star)                1

[5055 rows x 1 columns]
```

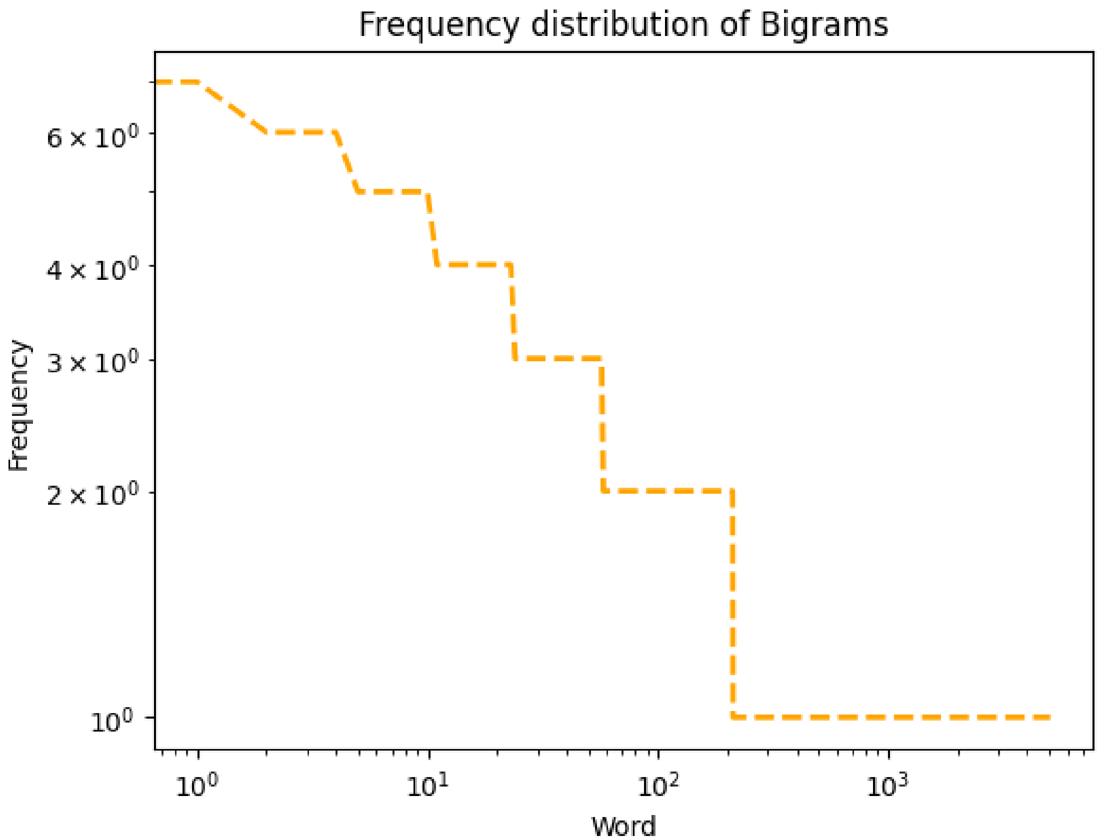
[28]:
```python
# Storing the frequencies of unigrams
```

```python
freq2 = bigram_df2['Count'].reset_index(drop=True)
print(type(freq2))
freq2
```

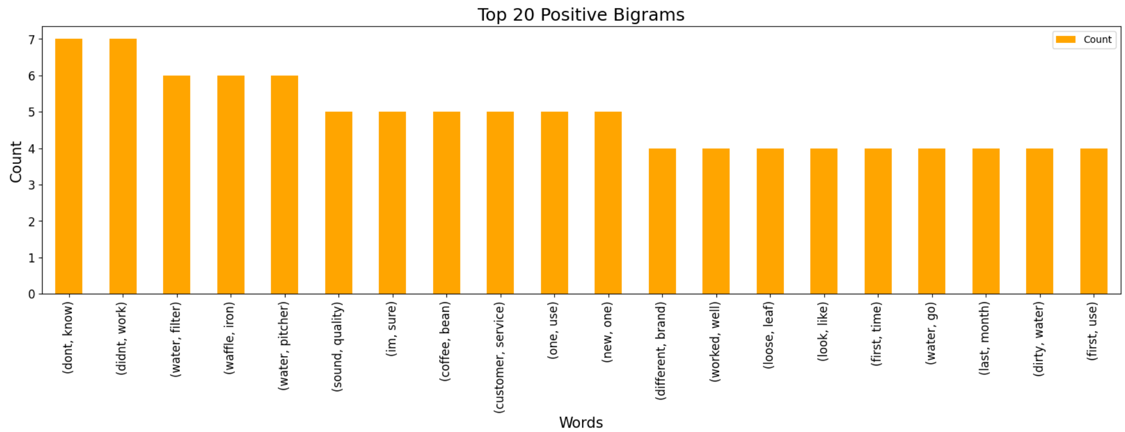
```
<class 'pandas.core.series.Series'>
```

[28]:
```
0       7
1       7
2       6
3       6
4       6
       ..
5050    1
5051    1
5052    1
5053    1
5054    1
Name: Count, Length: 5055, dtype: int64
```

[29]:
```python
# Plotting the log scale of frequencies of Bigrams

plt.plot(freq2, color="Orange", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Bigrams")
plt.show()
```



[30]:
```
# Plot the df3 Bigrams frequencies

bigram_df2.head(20).plot(kind='bar', figsize=(20, 5), color="Orange")
plt.xlabel('Words', fontsize = 15)
plt.ylabel('Count', fontsize=15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.title('Top 20 Positive Bigrams', fontsize=18)
plt.show()
```



1.8 Trigram Frequency

[35]:
```
# Generate trigrams from df2['Clean_review_Tokens']

trigram_list = [list(nltk.trigrams(text)) for text in␣
↪negative_df['Clean_review_Tokens']]

# Create a Counter object to count the frequency of each trigram
trigram_count = collections.Counter(list(chain(*trigram_list)))

# Convert the Counter object to a DataFrame
trigram_df = pd.DataFrame.from_dict(trigram_count, orient='index').reset_index()

# Rename the columns
trigram_df = trigram_df.rename(columns={'index': 'Trigram', 0: 'Count'})

# Sort the DataFrame by frequency in descending order
trigram_df.sort_values('Count', ascending=False, inplace=True)

# Print the top 20 trigrams
trigram_df.head(20)

trigram_df1 = trigram_df.reset_index(drop=True)
trigram_df2 = trigram_df1.set_index("Trigram")
trigram_df2
```

[35]:
```
                          Count
Trigram
(hot, air, popper)            3
(loose, leaf, tea)            3
(grinding, coffee, bean)      2
```

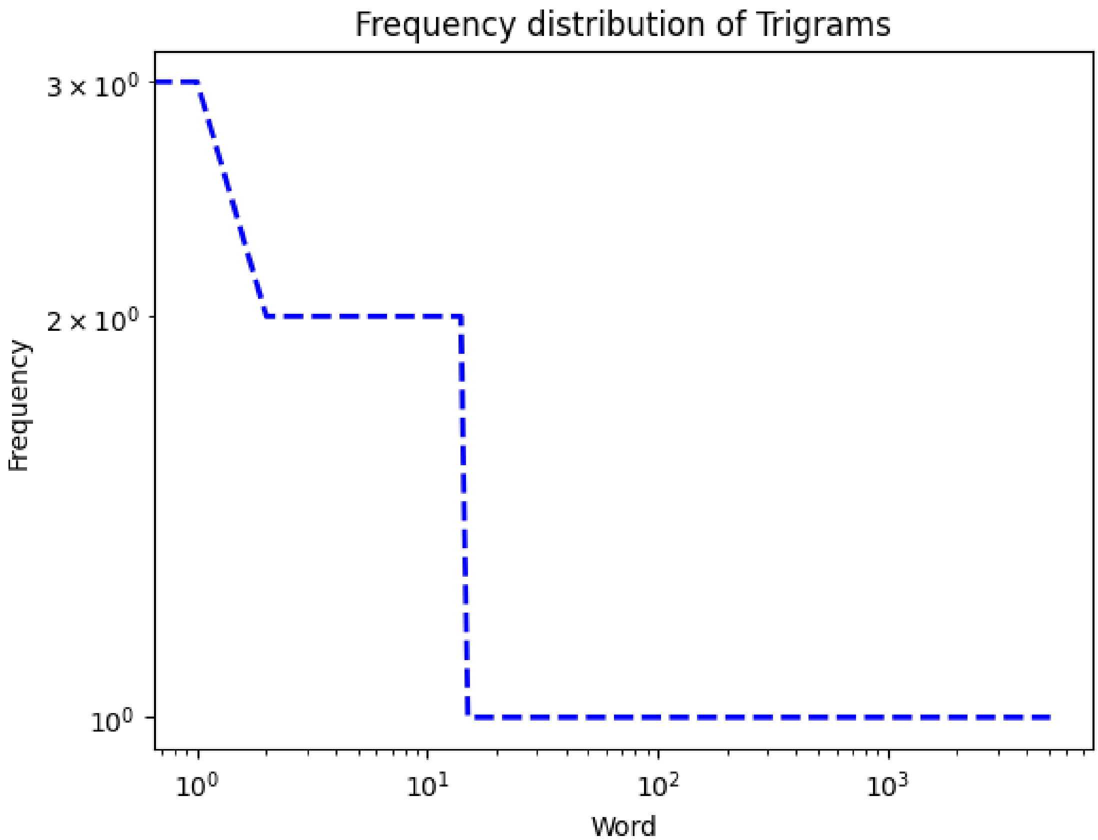
```
(water, get, filtered)         2
(magnet, strong, enough)       2
…                            …
(business, star, american)     1
(go, business, star)           1
(sale, go, business)           1
(purchase, sale, go)           1
(material, deserve, star)      1

[5082 rows x 1 columns]
```

[36]:
```python
# Storing the frequencies of unigrams

freq3 = trigram_df2['Count'].reset_index(drop=True)
print(type(freq3))
freq3
```

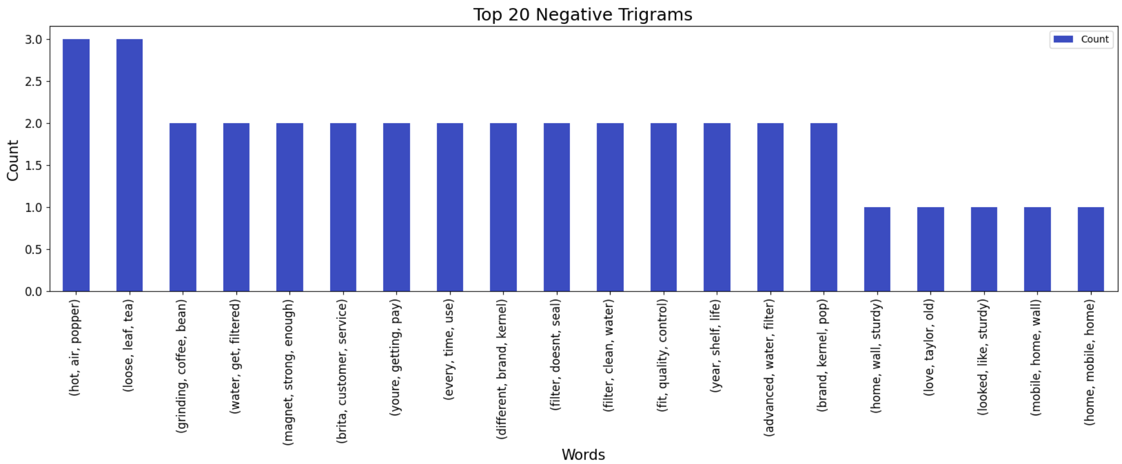
```
<class 'pandas.core.series.Series'>
```

[36]:
```
0       3
1       3
2       2
3       2
4       2
       ..
5077    1
5078    1
5079    1
5080    1
5081    1
Name: Count, Length: 5082, dtype: int64
```

[37]:
```python
# Plotting the log scale of frequencies of unigrams

plt.plot(freq3, color="Blue", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Trigrams")
plt.show()
```

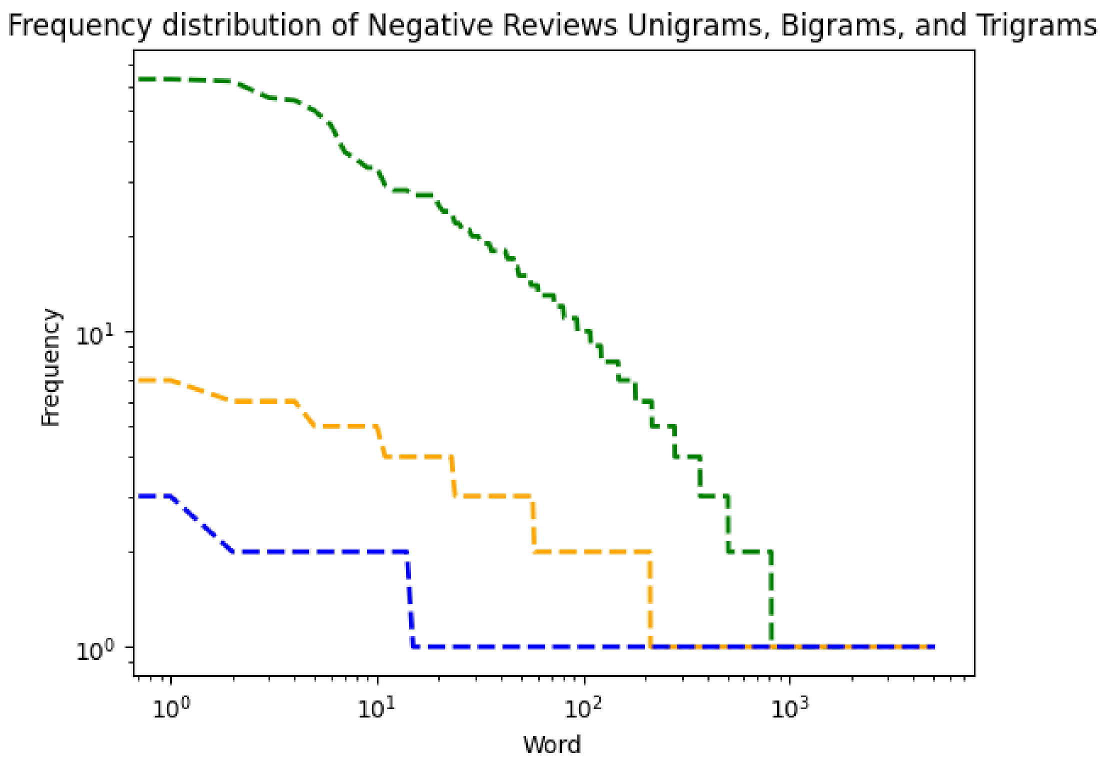


[39]:
```python
# Plot the top 20 Trigrams

trigram_df2.head(20).plot(kind='bar', figsize=(20, 5), cmap="coolwarm")
plt.xlabel('Words', fontsize = 15)
plt.ylabel('Count', fontsize=15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.title('Top 20 Negative Trigrams', fontsize=18)
plt.show()
```



1.9 Compare the Unigram, Bigram and Trigram token counts distributions

[41]:
```
# Compare the Frequency distributions of Unigrams, Bigrams, and Trigrams

plt.plot(freq1, color="Green", linewidth=2, linestyle='--')
plt.plot(freq2, color="Orange", linewidth=2, linestyle='--')
plt.plot(freq3, color="Blue", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Negative Reviews Unigrams, Bigrams, and␣
↪Trigrams")
plt.show()
```



The graph shows the frequency distribution of token counts in unigrams, bigrams, and trigrams in the positive sentiment reviews. The x-axis represents the words and the y-axis represents the frequency of occurrence. The graph shows that unigrams are the most frequent type of token, followed by bigrams and trigrams. This is because unigrams are the smallest units of meaning, and they are more likely to occur in a sentence than bigrams or trigrams. The graph also shows that there is a long tail of low-frequency tokens, which means that there are many words that only occur once or twice in the reviews. This is not surprising, as there are a large number of possible words that can be used to express a sentiment.

1.10 Next word prediction using LSTM

[42]:
```
#Import the necessary libraries

import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.layers import BatchNormalization
```

```
from keras.layers import Dropout
```

1.11 Data Pre-processing

[43]:
```
# Tokenize the text data

tokenizer = Tokenizer()
tokenizer.fit_on_texts(negative_df['Clean_review_Tokens'])
total_words = len(tokenizer.word_index) + 1
print(total_words)
```

```
1917
```

[48]:
```
# Create input sequences
input_sequences = []
for line in negative_df['Clean_review_Tokens']:
    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)
```

[ ]:
```
# Print input_sequences
# input_sequences
```

[49]:
```
# Print the max sequence length

max_sequence_len = max([len(seq) for seq in input_sequences])
print(max_sequence_len)
```

```
378
```

[50]:
```
# Pad sequences and split into predictors and label

padded_input_sequences = np.array(pad_sequences(input_sequences,␣
 ↪maxlen=max_sequence_len, padding='pre'))

print(padded_input_sequences)
```

```
[[   0    0    0 …    0  821   37]
 [   0    0    0 …  821   37   16]
 [   0    0    0 …   37   16   30]
 …
 [   0    0    0 … 1915  123  236]
 [   0    0    0 …  123  236 1916]
 [   0    0    0 …  236 1916  123]]
```

[47]:
```
# Define X and y

X = padded_input_sequences[:, :-1]
y = padded_input_sequences[:, -1]

# Print the dimensions of X and y

print("The shape of padded input sequence X is:", X.shape)
print("The shape of padded input sequence y is:", y.shape)
```

```
The shape of padded input sequence X is: (5367, 377)
The shape of padded input sequence y is: (5367,)
```

[51]:
```
# Convert target data to one-hot encoding
y = tf.keras.utils.to_categorical(y, num_classes=total_words)

# Print the new dimensions of X and y
print("The shape of X is:", X.shape)
print("The shape of y is:", y.shape)
```

```
The shape of X is: (5367, 377)
The shape of y is: (5367, 1917)
```

[55]:
```
# Define the model

model = Sequential()
model.add(Embedding(total_words, 100, input_length=max_sequence_len-1))
model.add(LSTM(128, return_sequences=True))
model.add((LSTM(128)))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=0.01),␣
↪metrics=['accuracy'])
model.summary()
```

```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate`
or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.Adam.
```

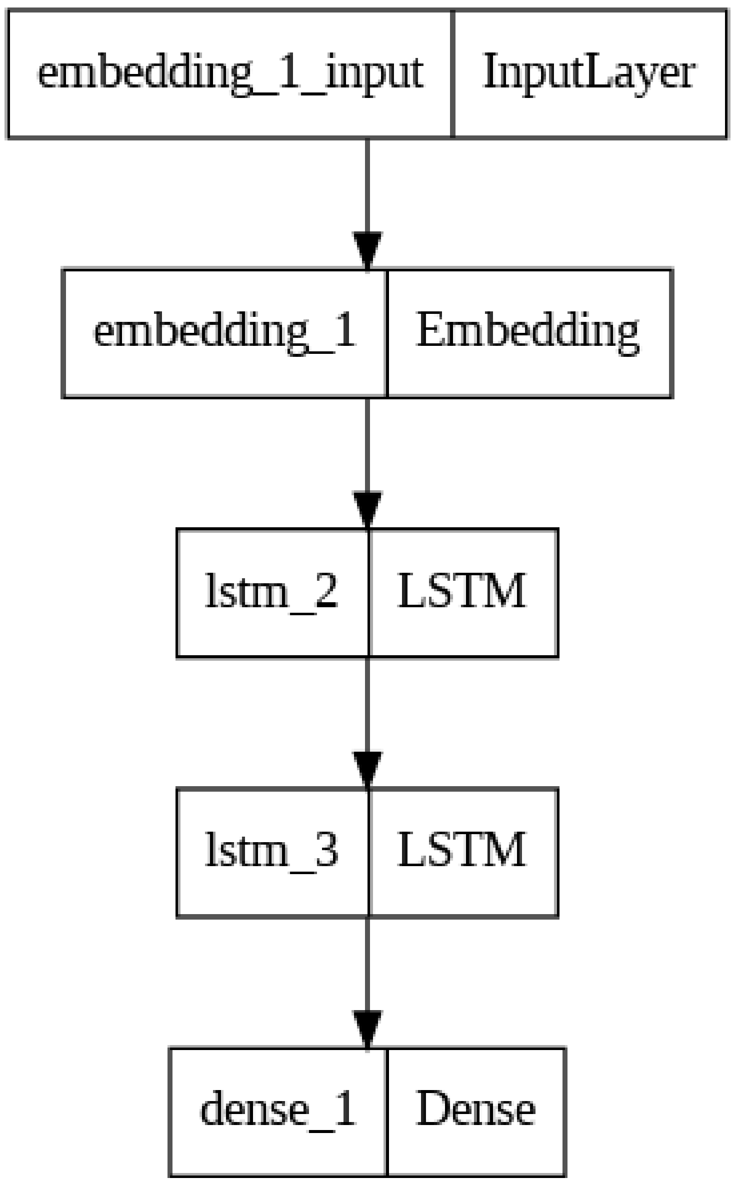
```
Model: "sequential_1"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 377, 100)          191700

 lstm_2 (LSTM)               (None, 377, 128)          117248

 lstm_3 (LSTM)               (None, 128)               131584

 dense_1 (Dense)             (None, 1917)              247293
```

```
=================================================================
Total params: 687825 (2.62 MB)
Trainable params: 687825 (2.62 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[56]:

```python
from tensorflow import keras
from tensorflow.keras.utils import plot_model

keras.utils.plot_model(model, to_file='model.png', show_layer_names=True)
```

[56]:



[57]:
```
#Fit the model

history = model.fit(X, y, validation_split=0.05, batch_size=128, epochs=500,␣
 ↪shuffle=True).history
```

```
Epoch 1/500
40/40 [==============================] - 17s 226ms/step - loss: 7.3756 -
accuracy: 0.0094 - val_loss: 7.3491 - val_accuracy: 0.0223
Epoch 2/500
40/40 [==============================] - 8s 195ms/step - loss: 7.0381 -
accuracy: 0.0116 - val_loss: 7.4586 - val_accuracy: 0.0149
Epoch 3/500
40/40 [==============================] - 5s 136ms/step - loss: 6.9788 -
accuracy: 0.0092 - val_loss: 7.5572 - val_accuracy: 0.0149
Epoch 4/500
40/40 [==============================] - 5s 138ms/step - loss: 6.9670 -
accuracy: 0.0096 - val_loss: 7.5925 - val_accuracy: 0.0037
Epoch 5/500
40/40 [==============================] - 5s 113ms/step - loss: 6.9554 -
accuracy: 0.0082 - val_loss: 7.6676 - val_accuracy: 0.0149
Epoch 6/500
40/40 [==============================] - 4s 91ms/step - loss: 6.9397 - accuracy:
0.0108 - val_loss: 7.6583 - val_accuracy: 0.0223
Epoch 7/500
40/40 [==============================] - 3s 76ms/step - loss: 6.8724 - accuracy:
0.0114 - val_loss: 7.8269 - val_accuracy: 0.0112
Epoch 8/500
40/40 [==============================] - 3s 72ms/step - loss: 6.7957 - accuracy:
0.0102 - val_loss: 7.8991 - val_accuracy: 0.0223
Epoch 9/500
40/40 [==============================] - 2s 63ms/step - loss: 6.7517 - accuracy:
0.0092 - val_loss: 7.9869 - val_accuracy: 0.0112
Epoch 10/500
40/40 [==============================] - 3s 77ms/step - loss: 6.7148 - accuracy:
0.0118 - val_loss: 8.1376 - val_accuracy: 0.0037
Epoch 11/500
40/40 [==============================] - 2s 49ms/step - loss: 6.6560 - accuracy:
0.0116 - val_loss: 8.1497 - val_accuracy: 0.0074
Epoch 12/500
40/40 [==============================] - 2s 63ms/step - loss: 6.5857 - accuracy:
0.0126 - val_loss: 8.2663 - val_accuracy: 0.0074
Epoch 13/500
40/40 [==============================] - 2s 49ms/step - loss: 6.5079 - accuracy:
0.0137 - val_loss: 8.3634 - val_accuracy: 0.0037
Epoch 14/500
40/40 [==============================] - 2s 49ms/step - loss: 6.4198 - accuracy:
0.0145 - val_loss: 8.4538 - val_accuracy: 0.0074
```

```
Epoch 15/500
40/40 [==============================] - 2s 39ms/step - loss: 6.3414 - accuracy:
0.0153 - val_loss: 8.5936 - val_accuracy: 0.0037
Epoch 16/500
40/40 [==============================] - 2s 63ms/step - loss: 6.2673 - accuracy:
0.0167 - val_loss: 8.6740 - val_accuracy: 0.0149
Epoch 17/500
40/40 [==============================] - 3s 67ms/step - loss: 6.1989 - accuracy:
0.0155 - val_loss: 8.7407 - val_accuracy: 0.0037
Epoch 18/500
40/40 [==============================] - 2s 53ms/step - loss: 6.1354 - accuracy:
0.0159 - val_loss: 8.8654 - val_accuracy: 0.0037
Epoch 19/500
40/40 [==============================] - 2s 58ms/step - loss: 6.0710 - accuracy:
0.0192 - val_loss: 8.9419 - val_accuracy: 0.0037
Epoch 20/500
40/40 [==============================] - 2s 49ms/step - loss: 6.0065 - accuracy:
0.0190 - val_loss: 9.0247 - val_accuracy: 0.0037
Epoch 21/500
40/40 [==============================] - 2s 53ms/step - loss: 5.9508 - accuracy:
0.0228 - val_loss: 9.0476 - val_accuracy: 0.0037
Epoch 22/500
40/40 [==============================] - 2s 48ms/step - loss: 5.8933 - accuracy:
0.0220 - val_loss: 9.1481 - val_accuracy: 0.0000e+00
Epoch 23/500
40/40 [==============================] - 2s 53ms/step - loss: 5.8323 - accuracy:
0.0265 - val_loss: 9.1983 - val_accuracy: 0.0037
Epoch 24/500
40/40 [==============================] - 2s 49ms/step - loss: 5.7776 - accuracy:
0.0259 - val_loss: 9.2908 - val_accuracy: 0.0037
Epoch 25/500
40/40 [==============================] - 2s 44ms/step - loss: 5.7221 - accuracy:
0.0294 - val_loss: 9.3211 - val_accuracy: 0.0000e+00
Epoch 26/500
40/40 [==============================] - 2s 58ms/step - loss: 5.6672 - accuracy:
0.0328 - val_loss: 9.3639 - val_accuracy: 0.0037
Epoch 27/500
40/40 [==============================] - 2s 44ms/step - loss: 5.6144 - accuracy:
0.0326 - val_loss: 9.4003 - val_accuracy: 0.0000e+00
Epoch 28/500
40/40 [==============================] - 2s 54ms/step - loss: 5.5578 - accuracy:
0.0363 - val_loss: 9.4922 - val_accuracy: 0.0037
Epoch 29/500
40/40 [==============================] - 2s 39ms/step - loss: 5.5019 - accuracy:
0.0371 - val_loss: 9.5262 - val_accuracy: 0.0000e+00
Epoch 30/500
40/40 [==============================] - 2s 44ms/step - loss: 5.4459 - accuracy:
0.0414 - val_loss: 9.5688 - val_accuracy: 0.0037
```

```
Epoch 31/500
40/40 [==============================] - 2s 40ms/step - loss: 5.3958 - accuracy:
0.0481 - val_loss: 9.6042 - val_accuracy: 0.0074
Epoch 32/500
40/40 [==============================] - 2s 44ms/step - loss: 5.3434 - accuracy:
0.0457 - val_loss: 9.6549 - val_accuracy: 0.0037
Epoch 33/500
40/40 [==============================] - 2s 44ms/step - loss: 5.2896 - accuracy:
0.0524 - val_loss: 9.7153 - val_accuracy: 0.0000e+00
Epoch 34/500
40/40 [==============================] - 2s 53ms/step - loss: 5.2434 - accuracy:
0.0506 - val_loss: 9.7424 - val_accuracy: 0.0037
Epoch 35/500
40/40 [==============================] - 2s 43ms/step - loss: 5.1943 - accuracy:
0.0559 - val_loss: 9.7663 - val_accuracy: 0.0000e+00
Epoch 36/500
40/40 [==============================] - 2s 39ms/step - loss: 5.1361 - accuracy:
0.0600 - val_loss: 9.8415 - val_accuracy: 0.0000e+00
Epoch 37/500
40/40 [==============================] - 2s 58ms/step - loss: 5.0814 - accuracy:
0.0612 - val_loss: 9.8581 - val_accuracy: 0.0037
Epoch 38/500
40/40 [==============================] - 2s 39ms/step - loss: 5.0325 - accuracy:
0.0641 - val_loss: 9.9125 - val_accuracy: 0.0037
Epoch 39/500
40/40 [==============================] - 2s 39ms/step - loss: 4.9774 - accuracy:
0.0710 - val_loss: 9.9714 - val_accuracy: 0.0037
Epoch 40/500
40/40 [==============================] - 2s 49ms/step - loss: 4.9310 - accuracy:
0.0775 - val_loss: 9.9860 - val_accuracy: 0.0000e+00
Epoch 41/500
40/40 [==============================] - 2s 39ms/step - loss: 4.8809 - accuracy:
0.0792 - val_loss: 10.0621 - val_accuracy: 0.0037
Epoch 42/500
40/40 [==============================] - 2s 48ms/step - loss: 4.8327 - accuracy:
0.0851 - val_loss: 10.1147 - val_accuracy: 0.0037
Epoch 43/500
40/40 [==============================] - 2s 44ms/step - loss: 4.7902 - accuracy:
0.0896 - val_loss: 10.1344 - val_accuracy: 0.0037
Epoch 44/500
40/40 [==============================] - 2s 44ms/step - loss: 4.7395 - accuracy:
0.1000 - val_loss: 10.1788 - val_accuracy: 0.0037
Epoch 45/500
40/40 [==============================] - 2s 39ms/step - loss: 4.6858 - accuracy:
0.1073 - val_loss: 10.2356 - val_accuracy: 0.0000e+00
Epoch 46/500
40/40 [==============================] - 2s 44ms/step - loss: 4.6380 - accuracy:
0.1159 - val_loss: 10.2847 - val_accuracy: 0.0037
```

```
Epoch 47/500
40/40 [==============================] - 2s 44ms/step - loss: 4.5912 - accuracy:
0.1267 - val_loss: 10.3139 - val_accuracy: 0.0000e+00
Epoch 48/500
40/40 [==============================] - 2s 53ms/step - loss: 4.5384 - accuracy:
0.1318 - val_loss: 10.3535 - val_accuracy: 0.0000e+00
Epoch 49/500
40/40 [==============================] - 2s 38ms/step - loss: 4.4926 - accuracy:
0.1426 - val_loss: 10.3946 - val_accuracy: 0.0000e+00
Epoch 50/500
40/40 [==============================] - 2s 44ms/step - loss: 4.4463 - accuracy:
0.1563 - val_loss: 10.4055 - val_accuracy: 0.0000e+00
Epoch 51/500
40/40 [==============================] - 2s 44ms/step - loss: 4.4013 - accuracy:
0.1616 - val_loss: 10.4848 - val_accuracy: 0.0000e+00
Epoch 52/500
40/40 [==============================] - 2s 39ms/step - loss: 4.3505 - accuracy:
0.1754 - val_loss: 10.5355 - val_accuracy: 0.0074
Epoch 53/500
40/40 [==============================] - 2s 44ms/step - loss: 4.3051 - accuracy:
0.1846 - val_loss: 10.5568 - val_accuracy: 0.0000e+00
Epoch 54/500
40/40 [==============================] - 2s 53ms/step - loss: 4.2550 - accuracy:
0.1954 - val_loss: 10.6036 - val_accuracy: 0.0000e+00
Epoch 55/500
40/40 [==============================] - 2s 44ms/step - loss: 4.2118 - accuracy:
0.2085 - val_loss: 10.6518 - val_accuracy: 0.0000e+00
Epoch 56/500
40/40 [==============================] - 2s 43ms/step - loss: 4.1691 - accuracy:
0.2142 - val_loss: 10.6912 - val_accuracy: 0.0000e+00
Epoch 57/500
40/40 [==============================] - 2s 48ms/step - loss: 4.1221 - accuracy:
0.2250 - val_loss: 10.7100 - val_accuracy: 0.0000e+00
Epoch 58/500
40/40 [==============================] - 2s 44ms/step - loss: 4.0761 - accuracy:
0.2364 - val_loss: 10.7751 - val_accuracy: 0.0000e+00
Epoch 59/500
40/40 [==============================] - 2s 44ms/step - loss: 4.0288 - accuracy:
0.2509 - val_loss: 10.7854 - val_accuracy: 0.0000e+00
Epoch 60/500
40/40 [==============================] - 2s 49ms/step - loss: 3.9802 - accuracy:
0.2630 - val_loss: 10.8526 - val_accuracy: 0.0000e+00
Epoch 61/500
40/40 [==============================] - 2s 39ms/step - loss: 3.9361 - accuracy:
0.2711 - val_loss: 10.8855 - val_accuracy: 0.0000e+00
Epoch 62/500
40/40 [==============================] - 2s 39ms/step - loss: 3.8902 - accuracy:
0.2848 - val_loss: 10.9414 - val_accuracy: 0.0000e+00
```

```
Epoch 63/500
40/40 [==============================] - 2s 43ms/step - loss: 3.8532 - accuracy:
0.2893 - val_loss: 10.9736 - val_accuracy: 0.0000e+00
Epoch 64/500
40/40 [==============================] - 2s 43ms/step - loss: 3.8059 - accuracy:
0.2995 - val_loss: 10.9960 - val_accuracy: 0.0000e+00
Epoch 65/500
40/40 [==============================] - 2s 44ms/step - loss: 3.7653 - accuracy:
0.3111 - val_loss: 11.0683 - val_accuracy: 0.0000e+00
Epoch 66/500
40/40 [==============================] - 2s 48ms/step - loss: 3.7208 - accuracy:
0.3199 - val_loss: 11.0621 - val_accuracy: 0.0000e+00
Epoch 67/500
40/40 [==============================] - 2s 44ms/step - loss: 3.6876 - accuracy:
0.3258 - val_loss: 11.1501 - val_accuracy: 0.0000e+00
Epoch 68/500
40/40 [==============================] - 2s 39ms/step - loss: 3.6353 - accuracy:
0.3358 - val_loss: 11.1655 - val_accuracy: 0.0000e+00
Epoch 69/500
40/40 [==============================] - 2s 39ms/step - loss: 3.5959 - accuracy:
0.3468 - val_loss: 11.1859 - val_accuracy: 0.0000e+00
Epoch 70/500
40/40 [==============================] - 2s 39ms/step - loss: 3.5506 - accuracy:
0.3531 - val_loss: 11.2510 - val_accuracy: 0.0000e+00
Epoch 71/500
40/40 [==============================] - 2s 48ms/step - loss: 3.5093 - accuracy:
0.3656 - val_loss: 11.2508 - val_accuracy: 0.0000e+00
Epoch 72/500
40/40 [==============================] - 2s 39ms/step - loss: 3.4634 - accuracy:
0.3754 - val_loss: 11.2787 - val_accuracy: 0.0000e+00
Epoch 73/500
40/40 [==============================] - 2s 40ms/step - loss: 3.4347 - accuracy:
0.3780 - val_loss: 11.3368 - val_accuracy: 0.0000e+00
Epoch 74/500
40/40 [==============================] - 2s 44ms/step - loss: 3.3905 - accuracy:
0.3845 - val_loss: 11.3728 - val_accuracy: 0.0000e+00
Epoch 75/500
40/40 [==============================] - 2s 44ms/step - loss: 3.3446 - accuracy:
0.3960 - val_loss: 11.4068 - val_accuracy: 0.0000e+00
Epoch 76/500
40/40 [==============================] - 2s 39ms/step - loss: 3.3062 - accuracy:
0.4066 - val_loss: 11.4614 - val_accuracy: 0.0037
Epoch 77/500
40/40 [==============================] - 2s 48ms/step - loss: 3.2670 - accuracy:
0.4137 - val_loss: 11.4930 - val_accuracy: 0.0000e+00
Epoch 78/500
40/40 [==============================] - 2s 38ms/step - loss: 3.2290 - accuracy:
0.4170 - val_loss: 11.4821 - val_accuracy: 0.0000e+00
```

```
Epoch 79/500
40/40 [==============================] - 2s 43ms/step - loss: 3.1951 - accuracy:
0.4243 - val_loss: 11.5472 - val_accuracy: 0.0000e+00
Epoch 80/500
40/40 [==============================] - 2s 39ms/step - loss: 3.1493 - accuracy:
0.4321 - val_loss: 11.5955 - val_accuracy: 0.0000e+00
Epoch 81/500
40/40 [==============================] - 2s 49ms/step - loss: 3.1065 - accuracy:
0.4463 - val_loss: 11.6221 - val_accuracy: 0.0000e+00
Epoch 82/500
40/40 [==============================] - 2s 44ms/step - loss: 3.0718 - accuracy:
0.4541 - val_loss: 11.6724 - val_accuracy: 0.0000e+00
Epoch 83/500
40/40 [==============================] - 2s 39ms/step - loss: 3.0372 - accuracy:
0.4586 - val_loss: 11.6772 - val_accuracy: 0.0000e+00
Epoch 84/500
40/40 [==============================] - 2s 39ms/step - loss: 3.0004 - accuracy:
0.4608 - val_loss: 11.7469 - val_accuracy: 0.0000e+00
Epoch 85/500
40/40 [==============================] - 2s 44ms/step - loss: 2.9713 - accuracy:
0.4704 - val_loss: 11.7429 - val_accuracy: 0.0000e+00
Epoch 86/500
40/40 [==============================] - 2s 38ms/step - loss: 2.9271 - accuracy:
0.4761 - val_loss: 11.7822 - val_accuracy: 0.0000e+00
Epoch 87/500
40/40 [==============================] - 2s 44ms/step - loss: 2.8846 - accuracy:
0.4845 - val_loss: 11.8252 - val_accuracy: 0.0000e+00
Epoch 88/500
40/40 [==============================] - 2s 48ms/step - loss: 2.8541 - accuracy:
0.4910 - val_loss: 11.8454 - val_accuracy: 0.0000e+00
Epoch 89/500
40/40 [==============================] - 2s 39ms/step - loss: 2.8216 - accuracy:
0.4980 - val_loss: 11.8908 - val_accuracy: 0.0000e+00
Epoch 90/500
40/40 [==============================] - 2s 39ms/step - loss: 2.7882 - accuracy:
0.5016 - val_loss: 11.9218 - val_accuracy: 0.0000e+00
Epoch 91/500
40/40 [==============================] - 2s 40ms/step - loss: 2.7526 - accuracy:
0.5104 - val_loss: 11.9603 - val_accuracy: 0.0000e+00
Epoch 92/500
40/40 [==============================] - 2s 48ms/step - loss: 2.7190 - accuracy:
0.5135 - val_loss: 11.9925 - val_accuracy: 0.0000e+00
Epoch 93/500
40/40 [==============================] - 2s 38ms/step - loss: 2.6858 - accuracy:
0.5210 - val_loss: 12.0226 - val_accuracy: 0.0000e+00
Epoch 94/500
40/40 [==============================] - 2s 39ms/step - loss: 2.6615 - accuracy:
0.5247 - val_loss: 12.0408 - val_accuracy: 0.0000e+00
```

```
Epoch 95/500
40/40 [==============================] - 2s 39ms/step - loss: 2.6198 - accuracy:
0.5390 - val_loss: 12.0994 - val_accuracy: 0.0000e+00
Epoch 96/500
40/40 [==============================] - 2s 39ms/step - loss: 2.5893 - accuracy:
0.5383 - val_loss: 12.1045 - val_accuracy: 0.0000e+00
Epoch 97/500
40/40 [==============================] - 2s 39ms/step - loss: 2.5591 - accuracy:
0.5418 - val_loss: 12.1604 - val_accuracy: 0.0000e+00
Epoch 98/500
40/40 [==============================] - 2s 39ms/step - loss: 2.5267 - accuracy:
0.5512 - val_loss: 12.1955 - val_accuracy: 0.0000e+00
Epoch 99/500
40/40 [==============================] - 2s 39ms/step - loss: 2.4955 - accuracy:
0.5573 - val_loss: 12.2243 - val_accuracy: 0.0000e+00
Epoch 100/500
40/40 [==============================] - 2s 43ms/step - loss: 2.4660 - accuracy:
0.5645 - val_loss: 12.2371 - val_accuracy: 0.0000e+00
Epoch 101/500
40/40 [==============================] - 2s 39ms/step - loss: 2.4355 - accuracy:
0.5710 - val_loss: 12.2905 - val_accuracy: 0.0000e+00
Epoch 102/500
40/40 [==============================] - 2s 44ms/step - loss: 2.4049 - accuracy:
0.5802 - val_loss: 12.3318 - val_accuracy: 0.0000e+00
Epoch 103/500
40/40 [==============================] - 2s 39ms/step - loss: 2.3771 - accuracy:
0.5851 - val_loss: 12.3483 - val_accuracy: 0.0000e+00
Epoch 104/500
40/40 [==============================] - 2s 39ms/step - loss: 2.3440 - accuracy:
0.5896 - val_loss: 12.4042 - val_accuracy: 0.0000e+00
Epoch 105/500
40/40 [==============================] - 2s 39ms/step - loss: 2.3169 - accuracy:
0.5957 - val_loss: 12.4168 - val_accuracy: 0.0000e+00
Epoch 106/500
40/40 [==============================] - 2s 39ms/step - loss: 2.2871 - accuracy:
0.5959 - val_loss: 12.4667 - val_accuracy: 0.0000e+00
Epoch 107/500
40/40 [==============================] - 2s 44ms/step - loss: 2.2583 - accuracy:
0.6075 - val_loss: 12.4985 - val_accuracy: 0.0000e+00
Epoch 108/500
40/40 [==============================] - 2s 38ms/step - loss: 2.2384 - accuracy:
0.6016 - val_loss: 12.5220 - val_accuracy: 0.0000e+00
Epoch 109/500
40/40 [==============================] - 2s 39ms/step - loss: 2.2093 - accuracy:
0.6132 - val_loss: 12.6056 - val_accuracy: 0.0000e+00
Epoch 110/500
40/40 [==============================] - 2s 39ms/step - loss: 2.1847 - accuracy:
0.6193 - val_loss: 12.6136 - val_accuracy: 0.0000e+00
```

```
Epoch 111/500
40/40 [==============================] - 2s 39ms/step - loss: 2.1528 - accuracy:
0.6238 - val_loss: 12.6179 - val_accuracy: 0.0000e+00
Epoch 112/500
40/40 [==============================] - 2s 39ms/step - loss: 2.1289 - accuracy:
0.6250 - val_loss: 12.6401 - val_accuracy: 0.0000e+00
Epoch 113/500
40/40 [==============================] - 2s 44ms/step - loss: 2.1069 - accuracy:
0.6330 - val_loss: 12.6734 - val_accuracy: 0.0000e+00
Epoch 114/500
40/40 [==============================] - 2s 39ms/step - loss: 2.0754 - accuracy:
0.6381 - val_loss: 12.7251 - val_accuracy: 0.0000e+00
Epoch 115/500
40/40 [==============================] - 2s 39ms/step - loss: 2.0491 - accuracy:
0.6448 - val_loss: 12.7567 - val_accuracy: 0.0000e+00
Epoch 116/500
40/40 [==============================] - 2s 38ms/step - loss: 2.0314 - accuracy:
0.6461 - val_loss: 12.8148 - val_accuracy: 0.0000e+00
Epoch 117/500
40/40 [==============================] - 2s 39ms/step - loss: 2.0097 - accuracy:
0.6491 - val_loss: 12.7997 - val_accuracy: 0.0000e+00
Epoch 118/500
40/40 [==============================] - 2s 48ms/step - loss: 1.9900 - accuracy:
0.6493 - val_loss: 12.8493 - val_accuracy: 0.0000e+00
Epoch 119/500
40/40 [==============================] - 2s 49ms/step - loss: 1.9598 - accuracy:
0.6579 - val_loss: 12.8774 - val_accuracy: 0.0000e+00
Epoch 120/500
40/40 [==============================] - 2s 39ms/step - loss: 1.9342 - accuracy:
0.6630 - val_loss: 12.9181 - val_accuracy: 0.0000e+00
Epoch 121/500
40/40 [==============================] - 2s 40ms/step - loss: 1.9101 - accuracy:
0.6693 - val_loss: 12.9607 - val_accuracy: 0.0000e+00
Epoch 122/500
40/40 [==============================] - 2s 44ms/step - loss: 1.8883 - accuracy:
0.6697 - val_loss: 12.9670 - val_accuracy: 0.0000e+00
Epoch 123/500
40/40 [==============================] - 2s 38ms/step - loss: 1.8679 - accuracy:
0.6732 - val_loss: 13.0338 - val_accuracy: 0.0000e+00
Epoch 124/500
40/40 [==============================] - 2s 39ms/step - loss: 1.8443 - accuracy:
0.6781 - val_loss: 13.0348 - val_accuracy: 0.0000e+00
Epoch 125/500
40/40 [==============================] - 2s 39ms/step - loss: 1.8245 - accuracy:
0.6826 - val_loss: 13.0830 - val_accuracy: 0.0000e+00
Epoch 126/500
40/40 [==============================] - 2s 39ms/step - loss: 1.8066 - accuracy:
0.6883 - val_loss: 13.0856 - val_accuracy: 0.0000e+00
```

```
Epoch 127/500
40/40 [==============================] - 2s 39ms/step - loss: 1.7853 - accuracy:
0.6848 - val_loss: 13.1281 - val_accuracy: 0.0000e+00
Epoch 128/500
40/40 [==============================] - 2s 44ms/step - loss: 1.7658 - accuracy:
0.6918 - val_loss: 13.1527 - val_accuracy: 0.0000e+00
Epoch 129/500
40/40 [==============================] - 2s 39ms/step - loss: 1.7434 - accuracy:
0.7003 - val_loss: 13.2040 - val_accuracy: 0.0000e+00
Epoch 130/500
40/40 [==============================] - 2s 39ms/step - loss: 1.7250 - accuracy:
0.6995 - val_loss: 13.2214 - val_accuracy: 0.0000e+00
Epoch 131/500
40/40 [==============================] - 2s 38ms/step - loss: 1.6999 - accuracy:
0.7040 - val_loss: 13.2661 - val_accuracy: 0.0000e+00
Epoch 132/500
40/40 [==============================] - 2s 44ms/step - loss: 1.6748 - accuracy:
0.7134 - val_loss: 13.2955 - val_accuracy: 0.0000e+00
Epoch 133/500
40/40 [==============================] - 2s 40ms/step - loss: 1.6540 - accuracy:
0.7158 - val_loss: 13.3182 - val_accuracy: 0.0000e+00
Epoch 134/500
40/40 [==============================] - 2s 39ms/step - loss: 1.6389 - accuracy:
0.7132 - val_loss: 13.3498 - val_accuracy: 0.0000e+00
Epoch 135/500
40/40 [==============================] - 2s 39ms/step - loss: 1.6186 - accuracy:
0.7228 - val_loss: 13.4094 - val_accuracy: 0.0000e+00
Epoch 136/500
40/40 [==============================] - 2s 39ms/step - loss: 1.6015 - accuracy:
0.7230 - val_loss: 13.4125 - val_accuracy: 0.0000e+00
Epoch 137/500
40/40 [==============================] - 2s 39ms/step - loss: 1.5794 - accuracy:
0.7309 - val_loss: 13.4581 - val_accuracy: 0.0000e+00
Epoch 138/500
40/40 [==============================] - 2s 39ms/step - loss: 1.5698 - accuracy:
0.7295 - val_loss: 13.4670 - val_accuracy: 0.0000e+00
Epoch 139/500
40/40 [==============================] - 2s 38ms/step - loss: 1.5532 - accuracy:
0.7285 - val_loss: 13.5377 - val_accuracy: 0.0000e+00
Epoch 140/500
40/40 [==============================] - 2s 39ms/step - loss: 1.5304 - accuracy:
0.7372 - val_loss: 13.5432 - val_accuracy: 0.0000e+00
Epoch 141/500
40/40 [==============================] - 2s 39ms/step - loss: 1.5115 - accuracy:
0.7428 - val_loss: 13.5779 - val_accuracy: 0.0000e+00
Epoch 142/500
40/40 [==============================] - 2s 39ms/step - loss: 1.4914 - accuracy:
0.7466 - val_loss: 13.6001 - val_accuracy: 0.0000e+00
```

```
Epoch 143/500
40/40 [==============================] - 2s 39ms/step - loss: 1.4790 - accuracy:
0.7470 - val_loss: 13.6584 - val_accuracy: 0.0000e+00
Epoch 144/500
40/40 [==============================] - 2s 39ms/step - loss: 1.4656 - accuracy:
0.7511 - val_loss: 13.6848 - val_accuracy: 0.0000e+00
Epoch 145/500
40/40 [==============================] - 2s 39ms/step - loss: 1.4444 - accuracy:
0.7574 - val_loss: 13.6984 - val_accuracy: 0.0000e+00
Epoch 146/500
40/40 [==============================] - 2s 39ms/step - loss: 1.4288 - accuracy:
0.7534 - val_loss: 13.7292 - val_accuracy: 0.0000e+00
Epoch 147/500
40/40 [==============================] - 2s 38ms/step - loss: 1.4120 - accuracy:
0.7638 - val_loss: 13.7872 - val_accuracy: 0.0000e+00
Epoch 148/500
40/40 [==============================] - 2s 43ms/step - loss: 1.3929 - accuracy:
0.7613 - val_loss: 13.8062 - val_accuracy: 0.0000e+00
Epoch 149/500
40/40 [==============================] - 2s 39ms/step - loss: 1.3767 - accuracy:
0.7672 - val_loss: 13.8309 - val_accuracy: 0.0000e+00
Epoch 150/500
40/40 [==============================] - 2s 39ms/step - loss: 1.3673 - accuracy:
0.7689 - val_loss: 13.8590 - val_accuracy: 0.0000e+00
Epoch 151/500
40/40 [==============================] - 2s 39ms/step - loss: 1.3516 - accuracy:
0.7721 - val_loss: 13.8804 - val_accuracy: 0.0000e+00
Epoch 152/500
40/40 [==============================] - 2s 39ms/step - loss: 1.3378 - accuracy:
0.7703 - val_loss: 13.9186 - val_accuracy: 0.0000e+00
Epoch 153/500
40/40 [==============================] - 2s 39ms/step - loss: 1.3188 - accuracy:
0.7780 - val_loss: 13.9632 - val_accuracy: 0.0000e+00
Epoch 154/500
40/40 [==============================] - 2s 44ms/step - loss: 1.3038 - accuracy:
0.7825 - val_loss: 14.0186 - val_accuracy: 0.0000e+00
Epoch 155/500
40/40 [==============================] - 2s 43ms/step - loss: 1.2844 - accuracy:
0.7832 - val_loss: 14.0327 - val_accuracy: 0.0000e+00
Epoch 156/500
40/40 [==============================] - 2s 39ms/step - loss: 1.2724 - accuracy:
0.7856 - val_loss: 14.0274 - val_accuracy: 0.0000e+00
Epoch 157/500
40/40 [==============================] - 2s 44ms/step - loss: 1.2558 - accuracy:
0.7899 - val_loss: 14.0613 - val_accuracy: 0.0000e+00
Epoch 158/500
40/40 [==============================] - 2s 39ms/step - loss: 1.2430 - accuracy:
0.7923 - val_loss: 14.0990 - val_accuracy: 0.0000e+00
```

```
Epoch 159/500
40/40 [==============================] - 2s 39ms/step - loss: 1.2296 - accuracy:
0.7934 - val_loss: 14.1171 - val_accuracy: 0.0000e+00
Epoch 160/500
40/40 [==============================] - 2s 48ms/step - loss: 1.2177 - accuracy:
0.7936 - val_loss: 14.1340 - val_accuracy: 0.0000e+00
Epoch 161/500
40/40 [==============================] - 2s 39ms/step - loss: 1.2096 - accuracy:
0.7985 - val_loss: 14.1718 - val_accuracy: 0.0000e+00
Epoch 162/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1885 - accuracy:
0.7999 - val_loss: 14.2155 - val_accuracy: 0.0000e+00
Epoch 163/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1743 - accuracy:
0.8025 - val_loss: 14.2469 - val_accuracy: 0.0000e+00
Epoch 164/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1635 - accuracy:
0.8082 - val_loss: 14.2307 - val_accuracy: 0.0000e+00
Epoch 165/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1506 - accuracy:
0.8062 - val_loss: 14.2914 - val_accuracy: 0.0000e+00
Epoch 166/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1391 - accuracy:
0.8095 - val_loss: 14.3031 - val_accuracy: 0.0000e+00
Epoch 167/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1226 - accuracy:
0.8170 - val_loss: 14.3576 - val_accuracy: 0.0000e+00
Epoch 168/500
40/40 [==============================] - 2s 39ms/step - loss: 1.1063 - accuracy:
0.8186 - val_loss: 14.3970 - val_accuracy: 0.0000e+00
Epoch 169/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0936 - accuracy:
0.8207 - val_loss: 14.4243 - val_accuracy: 0.0000e+00
Epoch 170/500
40/40 [==============================] - 2s 38ms/step - loss: 1.0839 - accuracy:
0.8240 - val_loss: 14.4381 - val_accuracy: 0.0000e+00
Epoch 171/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0724 - accuracy:
0.8240 - val_loss: 14.4906 - val_accuracy: 0.0000e+00
Epoch 172/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0695 - accuracy:
0.8231 - val_loss: 14.5182 - val_accuracy: 0.0000e+00
Epoch 173/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0548 - accuracy:
0.8237 - val_loss: 14.4962 - val_accuracy: 0.0000e+00
Epoch 174/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0420 - accuracy:
0.8260 - val_loss: 14.5777 - val_accuracy: 0.0000e+00
```

```
Epoch 175/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0281 - accuracy:
0.8305 - val_loss: 14.5781 - val_accuracy: 0.0000e+00
Epoch 176/500
40/40 [==============================] - 2s 39ms/step - loss: 1.0119 - accuracy:
0.8339 - val_loss: 14.6050 - val_accuracy: 0.0000e+00
Epoch 177/500
40/40 [==============================] - 2s 39ms/step - loss: 0.9999 - accuracy:
0.8378 - val_loss: 14.6304 - val_accuracy: 0.0000e+00
Epoch 178/500
40/40 [==============================] - 2s 38ms/step - loss: 0.9932 - accuracy:
0.8354 - val_loss: 14.6742 - val_accuracy: 0.0000e+00
Epoch 179/500
40/40 [==============================] - 2s 39ms/step - loss: 0.9852 - accuracy:
0.8405 - val_loss: 14.7189 - val_accuracy: 0.0000e+00
Epoch 180/500
40/40 [==============================] - 2s 39ms/step - loss: 0.9773 - accuracy:
0.8413 - val_loss: 14.7546 - val_accuracy: 0.0000e+00
Epoch 181/500
40/40 [==============================] - 2s 39ms/step - loss: 0.9594 - accuracy:
0.8425 - val_loss: 14.7346 - val_accuracy: 0.0000e+00
Epoch 182/500
40/40 [==============================] - 2s 44ms/step - loss: 0.9490 - accuracy:
0.8466 - val_loss: 14.7789 - val_accuracy: 0.0000e+00
Epoch 183/500
40/40 [==============================] - 2s 44ms/step - loss: 0.9367 - accuracy:
0.8474 - val_loss: 14.8218 - val_accuracy: 0.0000e+00
Epoch 184/500
40/40 [==============================] - 2s 40ms/step - loss: 0.9272 - accuracy:
0.8490 - val_loss: 14.8586 - val_accuracy: 0.0000e+00
Epoch 185/500
40/40 [==============================] - 2s 39ms/step - loss: 0.9215 - accuracy:
0.8503 - val_loss: 14.8712 - val_accuracy: 0.0000e+00
Epoch 186/500
40/40 [==============================] - 2s 38ms/step - loss: 0.9099 - accuracy:
0.8541 - val_loss: 14.8898 - val_accuracy: 0.0000e+00
Epoch 187/500
40/40 [==============================] - 2s 43ms/step - loss: 0.8974 - accuracy:
0.8548 - val_loss: 14.8947 - val_accuracy: 0.0000e+00
Epoch 188/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8872 - accuracy:
0.8566 - val_loss: 14.9641 - val_accuracy: 0.0000e+00
Epoch 189/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8841 - accuracy:
0.8599 - val_loss: 14.9720 - val_accuracy: 0.0000e+00
Epoch 190/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8715 - accuracy:
0.8578 - val_loss: 15.0236 - val_accuracy: 0.0000e+00
```

```
Epoch 191/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8614 - accuracy:
0.8645 - val_loss: 15.0270 - val_accuracy: 0.0000e+00
Epoch 192/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8498 - accuracy:
0.8639 - val_loss: 15.0686 - val_accuracy: 0.0000e+00
Epoch 193/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8406 - accuracy:
0.8633 - val_loss: 15.1053 - val_accuracy: 0.0000e+00
Epoch 194/500
40/40 [==============================] - 2s 38ms/step - loss: 0.8390 - accuracy:
0.8668 - val_loss: 15.1368 - val_accuracy: 0.0000e+00
Epoch 195/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8307 - accuracy:
0.8660 - val_loss: 15.1188 - val_accuracy: 0.0000e+00
Epoch 196/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8144 - accuracy:
0.8705 - val_loss: 15.1500 - val_accuracy: 0.0000e+00
Epoch 197/500
40/40 [==============================] - 2s 39ms/step - loss: 0.8036 - accuracy:
0.8719 - val_loss: 15.2462 - val_accuracy: 0.0000e+00
Epoch 198/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7985 - accuracy:
0.8762 - val_loss: 15.2574 - val_accuracy: 0.0000e+00
Epoch 199/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7856 - accuracy:
0.8758 - val_loss: 15.2484 - val_accuracy: 0.0000e+00
Epoch 200/500
40/40 [==============================] - 2s 44ms/step - loss: 0.7762 - accuracy:
0.8766 - val_loss: 15.2608 - val_accuracy: 0.0000e+00
Epoch 201/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7771 - accuracy:
0.8778 - val_loss: 15.2636 - val_accuracy: 0.0000e+00
Epoch 202/500
40/40 [==============================] - 2s 38ms/step - loss: 0.7659 - accuracy:
0.8770 - val_loss: 15.2866 - val_accuracy: 0.0000e+00
Epoch 203/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7567 - accuracy:
0.8790 - val_loss: 15.3540 - val_accuracy: 0.0000e+00
Epoch 204/500
40/40 [==============================] - 2s 40ms/step - loss: 0.7458 - accuracy:
0.8837 - val_loss: 15.3661 - val_accuracy: 0.0000e+00
Epoch 205/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7361 - accuracy:
0.8852 - val_loss: 15.3827 - val_accuracy: 0.0000e+00
Epoch 206/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7269 - accuracy:
0.8854 - val_loss: 15.3984 - val_accuracy: 0.0000e+00
```

```
Epoch 207/500
40/40 [==============================] - 2s 44ms/step - loss: 0.7187 - accuracy:
0.8872 - val_loss: 15.4796 - val_accuracy: 0.0000e+00
Epoch 208/500
40/40 [==============================] - 2s 39ms/step - loss: 0.7085 - accuracy:
0.8888 - val_loss: 15.4730 - val_accuracy: 0.0000e+00
Epoch 209/500
40/40 [==============================] - 2s 38ms/step - loss: 0.7026 - accuracy:
0.8892 - val_loss: 15.5083 - val_accuracy: 0.0000e+00
Epoch 210/500
40/40 [==============================] - 2s 38ms/step - loss: 0.6958 - accuracy:
0.8925 - val_loss: 15.5922 - val_accuracy: 0.0000e+00
Epoch 211/500
40/40 [==============================] - 2s 44ms/step - loss: 0.6907 - accuracy:
0.8909 - val_loss: 15.5654 - val_accuracy: 0.0000e+00
Epoch 212/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6903 - accuracy:
0.8937 - val_loss: 15.5867 - val_accuracy: 0.0000e+00
Epoch 213/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6772 - accuracy:
0.8951 - val_loss: 15.6407 - val_accuracy: 0.0000e+00
Epoch 214/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6657 - accuracy:
0.8974 - val_loss: 15.6365 - val_accuracy: 0.0000e+00
Epoch 215/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6612 - accuracy:
0.8976 - val_loss: 15.6795 - val_accuracy: 0.0000e+00
Epoch 216/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6528 - accuracy:
0.8994 - val_loss: 15.7246 - val_accuracy: 0.0000e+00
Epoch 217/500
40/40 [==============================] - 2s 38ms/step - loss: 0.6438 - accuracy:
0.9021 - val_loss: 15.7267 - val_accuracy: 0.0000e+00
Epoch 218/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6358 - accuracy:
0.9021 - val_loss: 15.7195 - val_accuracy: 0.0000e+00
Epoch 219/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6312 - accuracy:
0.9051 - val_loss: 15.7456 - val_accuracy: 0.0000e+00
Epoch 220/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6245 - accuracy:
0.9025 - val_loss: 15.8342 - val_accuracy: 0.0000e+00
Epoch 221/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6161 - accuracy:
0.9053 - val_loss: 15.8498 - val_accuracy: 0.0000e+00
Epoch 222/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6094 - accuracy:
0.9039 - val_loss: 15.8374 - val_accuracy: 0.0000e+00
```

```
Epoch 223/500
40/40 [==============================] - 2s 39ms/step - loss: 0.6028 - accuracy:
0.9100 - val_loss: 15.8878 - val_accuracy: 0.0000e+00
Epoch 224/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5972 - accuracy:
0.9072 - val_loss: 15.9246 - val_accuracy: 0.0000e+00
Epoch 225/500
40/40 [==============================] - 2s 38ms/step - loss: 0.5912 - accuracy:
0.9088 - val_loss: 15.9278 - val_accuracy: 0.0000e+00
Epoch 226/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5900 - accuracy:
0.9090 - val_loss: 15.9671 - val_accuracy: 0.0000e+00
Epoch 227/500
40/40 [==============================] - 2s 44ms/step - loss: 0.5804 - accuracy:
0.9109 - val_loss: 15.9678 - val_accuracy: 0.0000e+00
Epoch 228/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5744 - accuracy:
0.9143 - val_loss: 15.9601 - val_accuracy: 0.0000e+00
Epoch 229/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5661 - accuracy:
0.9133 - val_loss: 16.0554 - val_accuracy: 0.0000e+00
Epoch 230/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5573 - accuracy:
0.9170 - val_loss: 16.0011 - val_accuracy: 0.0000e+00
Epoch 231/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5501 - accuracy:
0.9157 - val_loss: 16.0199 - val_accuracy: 0.0000e+00
Epoch 232/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5466 - accuracy:
0.9158 - val_loss: 16.1060 - val_accuracy: 0.0000e+00
Epoch 233/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5376 - accuracy:
0.9209 - val_loss: 16.1121 - val_accuracy: 0.0000e+00
Epoch 234/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5326 - accuracy:
0.9204 - val_loss: 16.1490 - val_accuracy: 0.0000e+00
Epoch 235/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5269 - accuracy:
0.9219 - val_loss: 16.1895 - val_accuracy: 0.0000e+00
Epoch 236/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5203 - accuracy:
0.9213 - val_loss: 16.2119 - val_accuracy: 0.0037
Epoch 237/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5211 - accuracy:
0.9225 - val_loss: 16.2067 - val_accuracy: 0.0000e+00
Epoch 238/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5122 - accuracy:
0.9223 - val_loss: 16.2491 - val_accuracy: 0.0000e+00
```

```
Epoch 239/500
40/40 [==============================] - 2s 39ms/step - loss: 0.5036 - accuracy:
0.9239 - val_loss: 16.2473 - val_accuracy: 0.0000e+00
Epoch 240/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4979 - accuracy:
0.9260 - val_loss: 16.2875 - val_accuracy: 0.0000e+00
Epoch 241/500
40/40 [==============================] - 2s 38ms/step - loss: 0.4933 - accuracy:
0.9247 - val_loss: 16.3249 - val_accuracy: 0.0000e+00
Epoch 242/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4902 - accuracy:
0.9284 - val_loss: 16.3240 - val_accuracy: 0.0000e+00
Epoch 243/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4822 - accuracy:
0.9274 - val_loss: 16.3701 - val_accuracy: 0.0000e+00
Epoch 244/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4816 - accuracy:
0.9278 - val_loss: 16.4017 - val_accuracy: 0.0000e+00
Epoch 245/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4715 - accuracy:
0.9304 - val_loss: 16.4400 - val_accuracy: 0.0000e+00
Epoch 246/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4657 - accuracy:
0.9323 - val_loss: 16.4685 - val_accuracy: 0.0000e+00
Epoch 247/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4604 - accuracy:
0.9308 - val_loss: 16.4955 - val_accuracy: 0.0000e+00
Epoch 248/500
40/40 [==============================] - 2s 43ms/step - loss: 0.4576 - accuracy:
0.9317 - val_loss: 16.5516 - val_accuracy: 0.0000e+00
Epoch 249/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4518 - accuracy:
0.9321 - val_loss: 16.5174 - val_accuracy: 0.0000e+00
Epoch 250/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4482 - accuracy:
0.9351 - val_loss: 16.6000 - val_accuracy: 0.0000e+00
Epoch 251/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4424 - accuracy:
0.9359 - val_loss: 16.5766 - val_accuracy: 0.0000e+00
Epoch 252/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4384 - accuracy:
0.9327 - val_loss: 16.6656 - val_accuracy: 0.0000e+00
Epoch 253/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4373 - accuracy:
0.9337 - val_loss: 16.6306 - val_accuracy: 0.0000e+00
Epoch 254/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4288 - accuracy:
0.9368 - val_loss: 16.6819 - val_accuracy: 0.0000e+00
```

```
Epoch 255/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4230 - accuracy:
0.9366 - val_loss: 16.6936 - val_accuracy: 0.0000e+00
Epoch 256/500
40/40 [==============================] - 2s 38ms/step - loss: 0.4194 - accuracy:
0.9388 - val_loss: 16.7066 - val_accuracy: 0.0000e+00
Epoch 257/500
40/40 [==============================] - 2s 38ms/step - loss: 0.4153 - accuracy:
0.9374 - val_loss: 16.7508 - val_accuracy: 0.0000e+00
Epoch 258/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4077 - accuracy:
0.9392 - val_loss: 16.7618 - val_accuracy: 0.0000e+00
Epoch 259/500
40/40 [==============================] - 2s 39ms/step - loss: 0.4019 - accuracy:
0.9402 - val_loss: 16.7606 - val_accuracy: 0.0000e+00
Epoch 260/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3978 - accuracy:
0.9402 - val_loss: 16.7634 - val_accuracy: 0.0000e+00
Epoch 261/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3961 - accuracy:
0.9421 - val_loss: 16.8397 - val_accuracy: 0.0000e+00
Epoch 262/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3889 - accuracy:
0.9443 - val_loss: 16.8788 - val_accuracy: 0.0000e+00
Epoch 263/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3871 - accuracy:
0.9421 - val_loss: 16.8780 - val_accuracy: 0.0000e+00
Epoch 264/500
40/40 [==============================] - 2s 38ms/step - loss: 0.3822 - accuracy:
0.9449 - val_loss: 16.9101 - val_accuracy: 0.0000e+00
Epoch 265/500
40/40 [==============================] - 2s 38ms/step - loss: 0.3759 - accuracy:
0.9449 - val_loss: 16.9532 - val_accuracy: 0.0000e+00
Epoch 266/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3699 - accuracy:
0.9472 - val_loss: 16.9851 - val_accuracy: 0.0000e+00
Epoch 267/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3675 - accuracy:
0.9480 - val_loss: 16.9617 - val_accuracy: 0.0000e+00
Epoch 268/500
40/40 [==============================] - 2s 44ms/step - loss: 0.3649 - accuracy:
0.9455 - val_loss: 17.0018 - val_accuracy: 0.0000e+00
Epoch 269/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3600 - accuracy:
0.9478 - val_loss: 17.0463 - val_accuracy: 0.0000e+00
Epoch 270/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3583 - accuracy:
0.9484 - val_loss: 17.0807 - val_accuracy: 0.0000e+00
```

```
Epoch 271/500
40/40 [==============================] - 2s 44ms/step - loss: 0.3534 - accuracy:
0.9478 - val_loss: 17.1067 - val_accuracy: 0.0000e+00
Epoch 272/500
40/40 [==============================] - 2s 38ms/step - loss: 0.3492 - accuracy:
0.9494 - val_loss: 17.1337 - val_accuracy: 0.0000e+00
Epoch 273/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3462 - accuracy:
0.9486 - val_loss: 17.1520 - val_accuracy: 0.0000e+00
Epoch 274/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3498 - accuracy:
0.9482 - val_loss: 17.1727 - val_accuracy: 0.0000e+00
Epoch 275/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3442 - accuracy:
0.9500 - val_loss: 17.1807 - val_accuracy: 0.0037
Epoch 276/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3408 - accuracy:
0.9508 - val_loss: 17.2756 - val_accuracy: 0.0000e+00
Epoch 277/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3451 - accuracy:
0.9496 - val_loss: 17.2575 - val_accuracy: 0.0000e+00
Epoch 278/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3366 - accuracy:
0.9494 - val_loss: 17.2594 - val_accuracy: 0.0000e+00
Epoch 279/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3261 - accuracy:
0.9527 - val_loss: 17.3220 - val_accuracy: 0.0000e+00
Epoch 280/500
40/40 [==============================] - 2s 38ms/step - loss: 0.3192 - accuracy:
0.9525 - val_loss: 17.3271 - val_accuracy: 0.0000e+00
Epoch 281/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3148 - accuracy:
0.9541 - val_loss: 17.3130 - val_accuracy: 0.0000e+00
Epoch 282/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3119 - accuracy:
0.9529 - val_loss: 17.3805 - val_accuracy: 0.0000e+00
Epoch 283/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3094 - accuracy:
0.9541 - val_loss: 17.3926 - val_accuracy: 0.0000e+00
Epoch 284/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3033 - accuracy:
0.9547 - val_loss: 17.4281 - val_accuracy: 0.0000e+00
Epoch 285/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3024 - accuracy:
0.9541 - val_loss: 17.4095 - val_accuracy: 0.0000e+00
Epoch 286/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3006 - accuracy:
0.9543 - val_loss: 17.4377 - val_accuracy: 0.0000e+00
```

```
Epoch 287/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2964 - accuracy:
0.9561 - val_loss: 17.4902 - val_accuracy: 0.0000e+00
Epoch 288/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2934 - accuracy:
0.9549 - val_loss: 17.5053 - val_accuracy: 0.0000e+00
Epoch 289/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2887 - accuracy:
0.9580 - val_loss: 17.5807 - val_accuracy: 0.0000e+00
Epoch 290/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2877 - accuracy:
0.9557 - val_loss: 17.5577 - val_accuracy: 0.0000e+00
Epoch 291/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2850 - accuracy:
0.9582 - val_loss: 17.5988 - val_accuracy: 0.0000e+00
Epoch 292/500
40/40 [==============================] - 2s 44ms/step - loss: 0.2795 - accuracy:
0.9586 - val_loss: 17.6766 - val_accuracy: 0.0000e+00
Epoch 293/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2775 - accuracy:
0.9580 - val_loss: 17.5915 - val_accuracy: 0.0000e+00
Epoch 294/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2753 - accuracy:
0.9588 - val_loss: 17.6914 - val_accuracy: 0.0000e+00
Epoch 295/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2712 - accuracy:
0.9606 - val_loss: 17.6796 - val_accuracy: 0.0000e+00
Epoch 296/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2682 - accuracy:
0.9602 - val_loss: 17.6833 - val_accuracy: 0.0000e+00
Epoch 297/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2652 - accuracy:
0.9602 - val_loss: 17.7507 - val_accuracy: 0.0000e+00
Epoch 298/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2647 - accuracy:
0.9582 - val_loss: 17.7826 - val_accuracy: 0.0000e+00
Epoch 299/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2612 - accuracy:
0.9602 - val_loss: 17.7452 - val_accuracy: 0.0000e+00
Epoch 300/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2572 - accuracy:
0.9588 - val_loss: 17.7986 - val_accuracy: 0.0000e+00
Epoch 301/500
40/40 [==============================] - 2s 44ms/step - loss: 0.2537 - accuracy:
0.9614 - val_loss: 17.8179 - val_accuracy: 0.0000e+00
Epoch 302/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2544 - accuracy:
0.9616 - val_loss: 17.8753 - val_accuracy: 0.0000e+00
```

```
Epoch 303/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2534 - accuracy:
0.9629 - val_loss: 17.8626 - val_accuracy: 0.0000e+00
Epoch 304/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2504 - accuracy:
0.9621 - val_loss: 17.8878 - val_accuracy: 0.0000e+00
Epoch 305/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2462 - accuracy:
0.9635 - val_loss: 17.9030 - val_accuracy: 0.0000e+00
Epoch 306/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2688 - accuracy:
0.9600 - val_loss: 17.8909 - val_accuracy: 0.0000e+00
Epoch 307/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2790 - accuracy:
0.9574 - val_loss: 17.9924 - val_accuracy: 0.0000e+00
Epoch 308/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2802 - accuracy:
0.9574 - val_loss: 17.9667 - val_accuracy: 0.0000e+00
Epoch 309/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2811 - accuracy:
0.9566 - val_loss: 18.0660 - val_accuracy: 0.0000e+00
Epoch 310/500
40/40 [==============================] - 2s 44ms/step - loss: 0.2641 - accuracy:
0.9598 - val_loss: 18.0684 - val_accuracy: 0.0000e+00
Epoch 311/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2509 - accuracy:
0.9627 - val_loss: 18.0093 - val_accuracy: 0.0000e+00
Epoch 312/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2351 - accuracy:
0.9631 - val_loss: 18.1098 - val_accuracy: 0.0000e+00
Epoch 313/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2261 - accuracy:
0.9641 - val_loss: 18.0986 - val_accuracy: 0.0000e+00
Epoch 314/500
40/40 [==============================] - 2s 44ms/step - loss: 0.2211 - accuracy:
0.9657 - val_loss: 18.1235 - val_accuracy: 0.0000e+00
Epoch 315/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2177 - accuracy:
0.9655 - val_loss: 18.1764 - val_accuracy: 0.0000e+00
Epoch 316/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2148 - accuracy:
0.9667 - val_loss: 18.1821 - val_accuracy: 0.0000e+00
Epoch 317/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2122 - accuracy:
0.9657 - val_loss: 18.2059 - val_accuracy: 0.0000e+00
Epoch 318/500
40/40 [==============================] - 2s 44ms/step - loss: 0.2099 - accuracy:
0.9663 - val_loss: 18.2268 - val_accuracy: 0.0000e+00
```

```
Epoch 319/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2081 - accuracy:
0.9682 - val_loss: 18.2542 - val_accuracy: 0.0000e+00
Epoch 320/500
40/40 [==============================] - 2s 38ms/step - loss: 0.2060 - accuracy:
0.9653 - val_loss: 18.2325 - val_accuracy: 0.0000e+00
Epoch 321/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2041 - accuracy:
0.9665 - val_loss: 18.2474 - val_accuracy: 0.0000e+00
Epoch 322/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2022 - accuracy:
0.9670 - val_loss: 18.2895 - val_accuracy: 0.0000e+00
Epoch 323/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2002 - accuracy:
0.9686 - val_loss: 18.3243 - val_accuracy: 0.0000e+00
Epoch 324/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1979 - accuracy:
0.9680 - val_loss: 18.3500 - val_accuracy: 0.0000e+00
Epoch 325/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1973 - accuracy:
0.9667 - val_loss: 18.3299 - val_accuracy: 0.0000e+00
Epoch 326/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1947 - accuracy:
0.9684 - val_loss: 18.3713 - val_accuracy: 0.0000e+00
Epoch 327/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1926 - accuracy:
0.9698 - val_loss: 18.3839 - val_accuracy: 0.0000e+00
Epoch 328/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1906 - accuracy:
0.9692 - val_loss: 18.3874 - val_accuracy: 0.0000e+00
Epoch 329/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1887 - accuracy:
0.9696 - val_loss: 18.4241 - val_accuracy: 0.0000e+00
Epoch 330/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1870 - accuracy:
0.9692 - val_loss: 18.4565 - val_accuracy: 0.0000e+00
Epoch 331/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1848 - accuracy:
0.9680 - val_loss: 18.4701 - val_accuracy: 0.0000e+00
Epoch 332/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1830 - accuracy:
0.9692 - val_loss: 18.5065 - val_accuracy: 0.0000e+00
Epoch 333/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1823 - accuracy:
0.9708 - val_loss: 18.5077 - val_accuracy: 0.0000e+00
Epoch 334/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1811 - accuracy:
0.9704 - val_loss: 18.4988 - val_accuracy: 0.0000e+00
```

```
Epoch 335/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1796 - accuracy:
0.9696 - val_loss: 18.5437 - val_accuracy: 0.0000e+00
Epoch 336/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1765 - accuracy:
0.9692 - val_loss: 18.5763 - val_accuracy: 0.0000e+00
Epoch 337/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1743 - accuracy:
0.9710 - val_loss: 18.5736 - val_accuracy: 0.0000e+00
Epoch 338/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1777 - accuracy:
0.9706 - val_loss: 18.6233 - val_accuracy: 0.0000e+00
Epoch 339/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1754 - accuracy:
0.9710 - val_loss: 18.6232 - val_accuracy: 0.0000e+00
Epoch 340/500
40/40 [==============================] - 2s 44ms/step - loss: 0.1730 - accuracy:
0.9700 - val_loss: 18.6270 - val_accuracy: 0.0000e+00
Epoch 341/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1792 - accuracy:
0.9700 - val_loss: 18.6949 - val_accuracy: 0.0000e+00
Epoch 342/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1816 - accuracy:
0.9719 - val_loss: 18.7981 - val_accuracy: 0.0000e+00
Epoch 343/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1809 - accuracy:
0.9712 - val_loss: 18.7812 - val_accuracy: 0.0000e+00
Epoch 344/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1913 - accuracy:
0.9684 - val_loss: 18.7628 - val_accuracy: 0.0000e+00
Epoch 345/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2037 - accuracy:
0.9661 - val_loss: 18.7607 - val_accuracy: 0.0000e+00
Epoch 346/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2141 - accuracy:
0.9637 - val_loss: 18.8106 - val_accuracy: 0.0000e+00
Epoch 347/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1973 - accuracy:
0.9696 - val_loss: 18.8755 - val_accuracy: 0.0000e+00
Epoch 348/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1832 - accuracy:
0.9690 - val_loss: 18.8257 - val_accuracy: 0.0000e+00
Epoch 349/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1750 - accuracy:
0.9696 - val_loss: 18.9382 - val_accuracy: 0.0000e+00
Epoch 350/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1647 - accuracy:
0.9723 - val_loss: 18.8974 - val_accuracy: 0.0000e+00
```

```
Epoch 351/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1580 - accuracy:
0.9731 - val_loss: 18.8661 - val_accuracy: 0.0000e+00
Epoch 352/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1533 - accuracy:
0.9723 - val_loss: 18.9354 - val_accuracy: 0.0000e+00
Epoch 353/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1503 - accuracy:
0.9735 - val_loss: 18.9662 - val_accuracy: 0.0000e+00
Epoch 354/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1491 - accuracy:
0.9721 - val_loss: 18.9623 - val_accuracy: 0.0000e+00
Epoch 355/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1467 - accuracy:
0.9733 - val_loss: 19.0464 - val_accuracy: 0.0000e+00
Epoch 356/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1460 - accuracy:
0.9723 - val_loss: 19.0127 - val_accuracy: 0.0000e+00
Epoch 357/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1453 - accuracy:
0.9727 - val_loss: 19.0573 - val_accuracy: 0.0000e+00
Epoch 358/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1427 - accuracy:
0.9745 - val_loss: 19.0816 - val_accuracy: 0.0000e+00
Epoch 359/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1419 - accuracy:
0.9737 - val_loss: 19.0664 - val_accuracy: 0.0000e+00
Epoch 360/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1411 - accuracy:
0.9729 - val_loss: 19.0763 - val_accuracy: 0.0000e+00
Epoch 361/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1389 - accuracy:
0.9741 - val_loss: 19.0895 - val_accuracy: 0.0000e+00
Epoch 362/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1372 - accuracy:
0.9745 - val_loss: 19.1319 - val_accuracy: 0.0000e+00
Epoch 363/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1366 - accuracy:
0.9741 - val_loss: 19.1218 - val_accuracy: 0.0000e+00
Epoch 364/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1369 - accuracy:
0.9749 - val_loss: 19.1639 - val_accuracy: 0.0000e+00
Epoch 365/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1355 - accuracy:
0.9755 - val_loss: 19.1621 - val_accuracy: 0.0000e+00
Epoch 366/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1331 - accuracy:
0.9741 - val_loss: 19.2183 - val_accuracy: 0.0000e+00
```

```
Epoch 367/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1316 - accuracy:
0.9741 - val_loss: 19.2394 - val_accuracy: 0.0000e+00
Epoch 368/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1315 - accuracy:
0.9753 - val_loss: 19.2625 - val_accuracy: 0.0000e+00
Epoch 369/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1310 - accuracy:
0.9741 - val_loss: 19.2581 - val_accuracy: 0.0000e+00
Epoch 370/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1284 - accuracy:
0.9767 - val_loss: 19.3211 - val_accuracy: 0.0000e+00
Epoch 371/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1275 - accuracy:
0.9745 - val_loss: 19.2671 - val_accuracy: 0.0000e+00
Epoch 372/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1260 - accuracy:
0.9757 - val_loss: 19.2977 - val_accuracy: 0.0000e+00
Epoch 373/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1249 - accuracy:
0.9755 - val_loss: 19.3382 - val_accuracy: 0.0000e+00
Epoch 374/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1243 - accuracy:
0.9769 - val_loss: 19.3341 - val_accuracy: 0.0000e+00
Epoch 375/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1236 - accuracy:
0.9753 - val_loss: 19.3265 - val_accuracy: 0.0000e+00
Epoch 376/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1222 - accuracy:
0.9751 - val_loss: 19.4026 - val_accuracy: 0.0000e+00
Epoch 377/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1216 - accuracy:
0.9755 - val_loss: 19.4049 - val_accuracy: 0.0000e+00
Epoch 378/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1206 - accuracy:
0.9745 - val_loss: 19.4005 - val_accuracy: 0.0000e+00
Epoch 379/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1197 - accuracy:
0.9751 - val_loss: 19.4197 - val_accuracy: 0.0000e+00
Epoch 380/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1181 - accuracy:
0.9759 - val_loss: 19.4360 - val_accuracy: 0.0000e+00
Epoch 381/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1179 - accuracy:
0.9759 - val_loss: 19.4860 - val_accuracy: 0.0000e+00
Epoch 382/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1183 - accuracy:
0.9745 - val_loss: 19.5151 - val_accuracy: 0.0000e+00
```

```
Epoch 383/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1166 - accuracy:
0.9761 - val_loss: 19.4781 - val_accuracy: 0.0000e+00
Epoch 384/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1262 - accuracy:
0.9751 - val_loss: 19.4717 - val_accuracy: 0.0000e+00
Epoch 385/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1685 - accuracy:
0.9688 - val_loss: 19.5442 - val_accuracy: 0.0000e+00
Epoch 386/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2229 - accuracy:
0.9566 - val_loss: 19.5570 - val_accuracy: 0.0000e+00
Epoch 387/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2596 - accuracy:
0.9478 - val_loss: 19.6427 - val_accuracy: 0.0000e+00
Epoch 388/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2271 - accuracy:
0.9559 - val_loss: 19.5765 - val_accuracy: 0.0000e+00
Epoch 389/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1536 - accuracy:
0.9739 - val_loss: 19.6319 - val_accuracy: 0.0000e+00
Epoch 390/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1259 - accuracy:
0.9755 - val_loss: 19.6329 - val_accuracy: 0.0000e+00
Epoch 391/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1158 - accuracy:
0.9763 - val_loss: 19.5703 - val_accuracy: 0.0000e+00
Epoch 392/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1112 - accuracy:
0.9761 - val_loss: 19.6095 - val_accuracy: 0.0000e+00
Epoch 393/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1094 - accuracy:
0.9770 - val_loss: 19.6340 - val_accuracy: 0.0000e+00
Epoch 394/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1078 - accuracy:
0.9759 - val_loss: 19.6716 - val_accuracy: 0.0000e+00
Epoch 395/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1063 - accuracy:
0.9778 - val_loss: 19.6733 - val_accuracy: 0.0000e+00
Epoch 396/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1056 - accuracy:
0.9767 - val_loss: 19.6688 - val_accuracy: 0.0000e+00
Epoch 397/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1046 - accuracy:
0.9770 - val_loss: 19.6868 - val_accuracy: 0.0000e+00
Epoch 398/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1034 - accuracy:
0.9778 - val_loss: 19.6979 - val_accuracy: 0.0000e+00
```

```
Epoch 399/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1022 - accuracy:
0.9782 - val_loss: 19.7297 - val_accuracy: 0.0000e+00
Epoch 400/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1018 - accuracy:
0.9770 - val_loss: 19.7289 - val_accuracy: 0.0000e+00
Epoch 401/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1004 - accuracy:
0.9774 - val_loss: 19.7780 - val_accuracy: 0.0000e+00
Epoch 402/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0999 - accuracy:
0.9776 - val_loss: 19.7543 - val_accuracy: 0.0000e+00
Epoch 403/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0987 - accuracy:
0.9784 - val_loss: 19.7848 - val_accuracy: 0.0000e+00
Epoch 404/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0984 - accuracy:
0.9767 - val_loss: 19.8116 - val_accuracy: 0.0000e+00
Epoch 405/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0977 - accuracy:
0.9776 - val_loss: 19.8166 - val_accuracy: 0.0000e+00
Epoch 406/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0966 - accuracy:
0.9770 - val_loss: 19.8369 - val_accuracy: 0.0000e+00
Epoch 407/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0961 - accuracy:
0.9770 - val_loss: 19.8723 - val_accuracy: 0.0000e+00
Epoch 408/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0959 - accuracy:
0.9782 - val_loss: 19.8462 - val_accuracy: 0.0000e+00
Epoch 409/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0950 - accuracy:
0.9774 - val_loss: 19.8686 - val_accuracy: 0.0000e+00
Epoch 410/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0948 - accuracy:
0.9778 - val_loss: 19.9208 - val_accuracy: 0.0000e+00
Epoch 411/500
40/40 [==============================] - 2s 44ms/step - loss: 0.0937 - accuracy:
0.9784 - val_loss: 19.9254 - val_accuracy: 0.0000e+00
Epoch 412/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0927 - accuracy:
0.9778 - val_loss: 19.9163 - val_accuracy: 0.0000e+00
Epoch 413/500
40/40 [==============================] - 2s 44ms/step - loss: 0.0924 - accuracy:
0.9778 - val_loss: 19.9562 - val_accuracy: 0.0000e+00
Epoch 414/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0919 - accuracy:
0.9774 - val_loss: 19.9170 - val_accuracy: 0.0000e+00
```

```
Epoch 415/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0908 - accuracy:
0.9776 - val_loss: 19.9476 - val_accuracy: 0.0000e+00
Epoch 416/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0905 - accuracy:
0.9778 - val_loss: 19.9780 - val_accuracy: 0.0000e+00
Epoch 417/500
40/40 [==============================] - 2s 44ms/step - loss: 0.0903 - accuracy:
0.9770 - val_loss: 19.9920 - val_accuracy: 0.0000e+00
Epoch 418/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0897 - accuracy:
0.9770 - val_loss: 20.0048 - val_accuracy: 0.0000e+00
Epoch 419/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0894 - accuracy:
0.9782 - val_loss: 20.0306 - val_accuracy: 0.0000e+00
Epoch 420/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0886 - accuracy:
0.9772 - val_loss: 20.0601 - val_accuracy: 0.0000e+00
Epoch 421/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0875 - accuracy:
0.9774 - val_loss: 20.0230 - val_accuracy: 0.0000e+00
Epoch 422/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0873 - accuracy:
0.9765 - val_loss: 20.0413 - val_accuracy: 0.0000e+00
Epoch 423/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0873 - accuracy:
0.9784 - val_loss: 20.0548 - val_accuracy: 0.0000e+00
Epoch 424/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0859 - accuracy:
0.9774 - val_loss: 20.0828 - val_accuracy: 0.0000e+00
Epoch 425/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0857 - accuracy:
0.9774 - val_loss: 20.0756 - val_accuracy: 0.0000e+00
Epoch 426/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0857 - accuracy:
0.9776 - val_loss: 20.0959 - val_accuracy: 0.0000e+00
Epoch 427/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0846 - accuracy:
0.9778 - val_loss: 20.1200 - val_accuracy: 0.0000e+00
Epoch 428/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0841 - accuracy:
0.9780 - val_loss: 20.1490 - val_accuracy: 0.0000e+00
Epoch 429/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0837 - accuracy:
0.9778 - val_loss: 20.1450 - val_accuracy: 0.0000e+00
Epoch 430/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0831 - accuracy:
0.9774 - val_loss: 20.2168 - val_accuracy: 0.0000e+00
```

```
Epoch 431/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0825 - accuracy:
0.9782 - val_loss: 20.2174 - val_accuracy: 0.0000e+00
Epoch 432/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0822 - accuracy:
0.9774 - val_loss: 20.2142 - val_accuracy: 0.0000e+00
Epoch 433/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0824 - accuracy:
0.9778 - val_loss: 20.2306 - val_accuracy: 0.0000e+00
Epoch 434/500
40/40 [==============================] - 2s 44ms/step - loss: 0.0811 - accuracy:
0.9772 - val_loss: 20.2316 - val_accuracy: 0.0000e+00
Epoch 435/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0801 - accuracy:
0.9782 - val_loss: 20.2577 - val_accuracy: 0.0000e+00
Epoch 436/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0804 - accuracy:
0.9782 - val_loss: 20.2785 - val_accuracy: 0.0000e+00
Epoch 437/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0795 - accuracy:
0.9770 - val_loss: 20.2757 - val_accuracy: 0.0000e+00
Epoch 438/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0792 - accuracy:
0.9778 - val_loss: 20.2692 - val_accuracy: 0.0000e+00
Epoch 439/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0787 - accuracy:
0.9776 - val_loss: 20.3245 - val_accuracy: 0.0000e+00
Epoch 440/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0782 - accuracy:
0.9774 - val_loss: 20.2990 - val_accuracy: 0.0000e+00
Epoch 441/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0778 - accuracy:
0.9774 - val_loss: 20.3802 - val_accuracy: 0.0000e+00
Epoch 442/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0770 - accuracy:
0.9784 - val_loss: 20.3456 - val_accuracy: 0.0000e+00
Epoch 443/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0771 - accuracy:
0.9770 - val_loss: 20.3459 - val_accuracy: 0.0000e+00
Epoch 444/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0756 - accuracy:
0.9776 - val_loss: 20.4424 - val_accuracy: 0.0000e+00
Epoch 445/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0756 - accuracy:
0.9780 - val_loss: 20.3991 - val_accuracy: 0.0000e+00
Epoch 446/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0759 - accuracy:
0.9772 - val_loss: 20.4649 - val_accuracy: 0.0000e+00
```

```
Epoch 447/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0752 - accuracy:
0.9774 - val_loss: 20.3909 - val_accuracy: 0.0000e+00
Epoch 448/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0747 - accuracy:
0.9774 - val_loss: 20.4450 - val_accuracy: 0.0000e+00
Epoch 449/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0786 - accuracy:
0.9778 - val_loss: 20.4576 - val_accuracy: 0.0000e+00
Epoch 450/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0756 - accuracy:
0.9784 - val_loss: 20.4794 - val_accuracy: 0.0000e+00
Epoch 451/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0745 - accuracy:
0.9772 - val_loss: 20.4936 - val_accuracy: 0.0000e+00
Epoch 452/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0730 - accuracy:
0.9776 - val_loss: 20.4424 - val_accuracy: 0.0000e+00
Epoch 453/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0737 - accuracy:
0.9778 - val_loss: 20.5788 - val_accuracy: 0.0000e+00
Epoch 454/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0780 - accuracy:
0.9774 - val_loss: 20.6731 - val_accuracy: 0.0000e+00
Epoch 455/500
40/40 [==============================] - 2s 38ms/step - loss: 0.1066 - accuracy:
0.9716 - val_loss: 20.3641 - val_accuracy: 0.0000e+00
Epoch 456/500
40/40 [==============================] - 2s 39ms/step - loss: 0.1977 - accuracy:
0.9525 - val_loss: 20.6307 - val_accuracy: 0.0000e+00
Epoch 457/500
40/40 [==============================] - 2s 39ms/step - loss: 0.3136 - accuracy:
0.9249 - val_loss: 20.3025 - val_accuracy: 0.0037
Epoch 458/500
40/40 [==============================] - 2s 39ms/step - loss: 0.2175 - accuracy:
0.9504 - val_loss: 20.6819 - val_accuracy: 0.0000e+00
Epoch 459/500
40/40 [==============================] - 2s 44ms/step - loss: 0.1302 - accuracy:
0.9710 - val_loss: 20.6731 - val_accuracy: 0.0000e+00
Epoch 460/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0909 - accuracy:
0.9774 - val_loss: 20.7056 - val_accuracy: 0.0000e+00
Epoch 461/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0782 - accuracy:
0.9782 - val_loss: 20.7674 - val_accuracy: 0.0000e+00
Epoch 462/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0744 - accuracy:
0.9784 - val_loss: 20.7348 - val_accuracy: 0.0000e+00
```

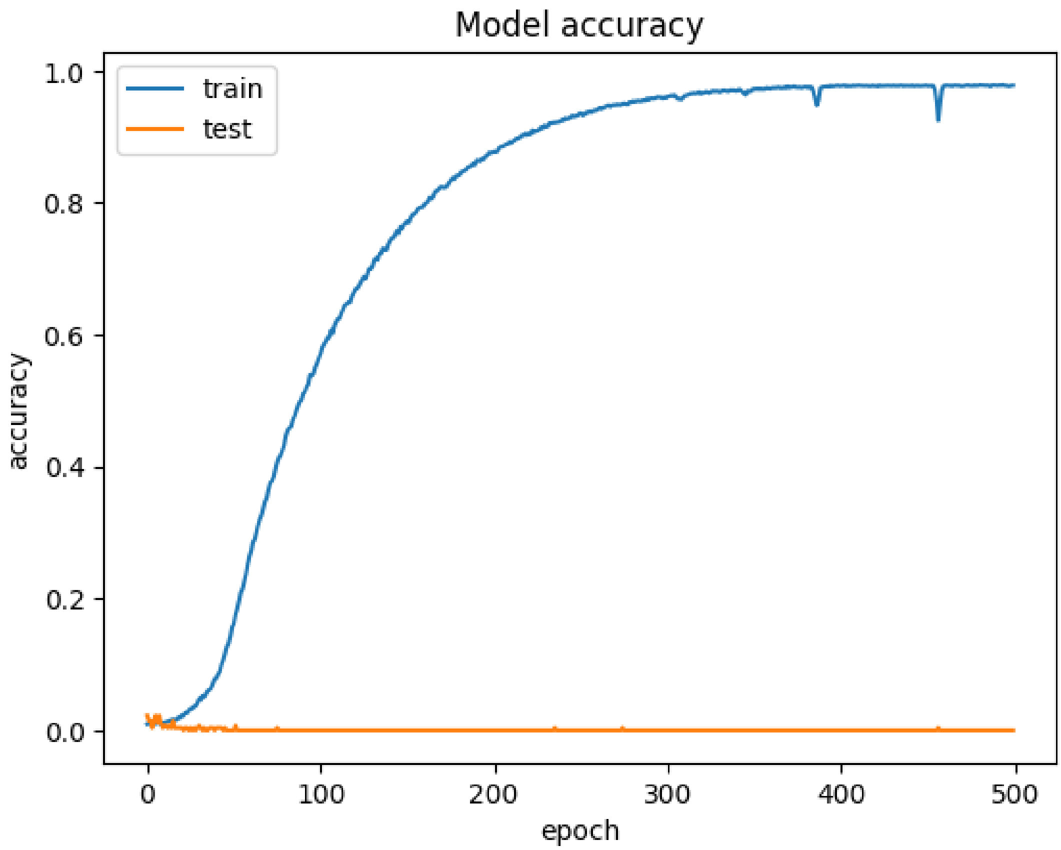
```
Epoch 463/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0731 - accuracy:
0.9778 - val_loss: 20.7559 - val_accuracy: 0.0000e+00
Epoch 464/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0720 - accuracy:
0.9780 - val_loss: 20.7399 - val_accuracy: 0.0000e+00
Epoch 465/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0709 - accuracy:
0.9769 - val_loss: 20.7727 - val_accuracy: 0.0000e+00
Epoch 466/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0703 - accuracy:
0.9782 - val_loss: 20.7388 - val_accuracy: 0.0000e+00
Epoch 467/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0699 - accuracy:
0.9784 - val_loss: 20.7602 - val_accuracy: 0.0000e+00
Epoch 468/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0691 - accuracy:
0.9778 - val_loss: 20.7698 - val_accuracy: 0.0000e+00
Epoch 469/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0689 - accuracy:
0.9782 - val_loss: 20.7641 - val_accuracy: 0.0000e+00
Epoch 470/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0686 - accuracy:
0.9769 - val_loss: 20.7688 - val_accuracy: 0.0000e+00
Epoch 471/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0679 - accuracy:
0.9770 - val_loss: 20.7865 - val_accuracy: 0.0000e+00
Epoch 472/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0673 - accuracy:
0.9774 - val_loss: 20.7910 - val_accuracy: 0.0000e+00
Epoch 473/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0673 - accuracy:
0.9784 - val_loss: 20.8062 - val_accuracy: 0.0000e+00
Epoch 474/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0669 - accuracy:
0.9788 - val_loss: 20.7994 - val_accuracy: 0.0000e+00
Epoch 475/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0663 - accuracy:
0.9776 - val_loss: 20.8144 - val_accuracy: 0.0000e+00
Epoch 476/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0660 - accuracy:
0.9778 - val_loss: 20.8243 - val_accuracy: 0.0000e+00
Epoch 477/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0660 - accuracy:
0.9774 - val_loss: 20.8535 - val_accuracy: 0.0000e+00
Epoch 478/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0656 - accuracy:
0.9788 - val_loss: 20.8468 - val_accuracy: 0.0000e+00
```

```
Epoch 479/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0654 - accuracy:
0.9776 - val_loss: 20.8420 - val_accuracy: 0.0000e+00
Epoch 480/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0649 - accuracy:
0.9780 - val_loss: 20.8584 - val_accuracy: 0.0000e+00
Epoch 481/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0646 - accuracy:
0.9792 - val_loss: 20.8626 - val_accuracy: 0.0000e+00
Epoch 482/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0644 - accuracy:
0.9784 - val_loss: 20.8878 - val_accuracy: 0.0000e+00
Epoch 483/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0650 - accuracy:
0.9786 - val_loss: 20.8856 - val_accuracy: 0.0000e+00
Epoch 484/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0640 - accuracy:
0.9778 - val_loss: 20.8829 - val_accuracy: 0.0000e+00
Epoch 485/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0630 - accuracy:
0.9786 - val_loss: 20.9343 - val_accuracy: 0.0000e+00
Epoch 486/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0631 - accuracy:
0.9778 - val_loss: 20.8990 - val_accuracy: 0.0000e+00
Epoch 487/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0628 - accuracy:
0.9765 - val_loss: 20.8875 - val_accuracy: 0.0000e+00
Epoch 488/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0626 - accuracy:
0.9786 - val_loss: 20.9222 - val_accuracy: 0.0000e+00
Epoch 489/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0625 - accuracy:
0.9788 - val_loss: 20.9636 - val_accuracy: 0.0000e+00
Epoch 490/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0626 - accuracy:
0.9782 - val_loss: 20.9271 - val_accuracy: 0.0000e+00
Epoch 491/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0620 - accuracy:
0.9778 - val_loss: 20.9318 - val_accuracy: 0.0000e+00
Epoch 492/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0619 - accuracy:
0.9788 - val_loss: 20.9644 - val_accuracy: 0.0000e+00
Epoch 493/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0614 - accuracy:
0.9780 - val_loss: 20.9391 - val_accuracy: 0.0000e+00
Epoch 494/500
40/40 [==============================] - 2s 38ms/step - loss: 0.0612 - accuracy:
0.9794 - val_loss: 20.9442 - val_accuracy: 0.0000e+00
```

```
Epoch 495/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0612 - accuracy:
0.9778 - val_loss: 20.9776 - val_accuracy: 0.0000e+00
Epoch 496/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0607 - accuracy:
0.9782 - val_loss: 20.9586 - val_accuracy: 0.0000e+00
Epoch 497/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0610 - accuracy:
0.9770 - val_loss: 20.9766 - val_accuracy: 0.0000e+00
Epoch 498/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0610 - accuracy:
0.9769 - val_loss: 21.0300 - val_accuracy: 0.0000e+00
Epoch 499/500
40/40 [==============================] - 2s 40ms/step - loss: 0.0602 - accuracy:
0.9780 - val_loss: 21.0065 - val_accuracy: 0.0000e+00
Epoch 500/500
40/40 [==============================] - 2s 39ms/step - loss: 0.0602 - accuracy:
0.9782 - val_loss: 21.0143 - val_accuracy: 0.0000e+00
```

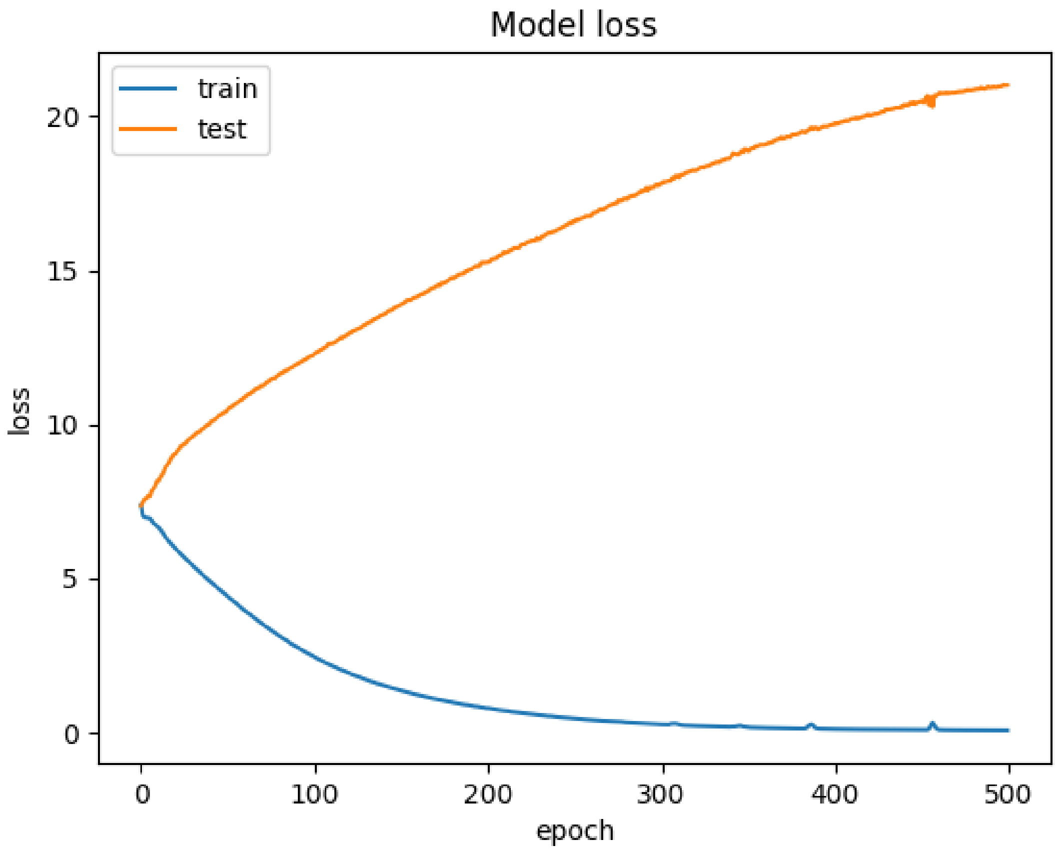
1.12 Model Evaluation

[59]:
```
plt.plot(history['accuracy'])
plt.plot(history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



The model accuracy graph sugg that the training accuracy increases from 0 to 97% but the validation accuracy remains stagnant at around 0% over 500 epochs.

[60]:
```
plt.plot(history['loss'])
plt.plot(history['val_loss'])
plt.title('Model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



The model loss graph suggests that the model training loss steadily decreases over time from around 7 to 0 while the validation loss steadily increases over the number of epochs from 7 to 20.

1.13 Model Prediction

[62]:
```
import time

# text = "husband us find good arthritis stretching"

text ="tore little year im strong poor quality also"

for i in range(10):

  # tokenize
  token_text = tokenizer.texts_to_sequences([text])[0]

  # padding
  padded_token_text = pad_sequences([token_text], maxlen=377, padding='pre')
```

```python
# predict
pos = np.argmax(model.predict(padded_token_text))

for word,index in tokenizer.word_index.items():
  if index == pos:
    text = text + " " + word
    print(text)
    time.sleep(2)
```

```
1/1 [==============================] - 1s 611ms/step
tore little year im strong poor quality also one
1/1 [==============================] - 0s 25ms/step
tore little year im strong poor quality also one handle
1/1 [==============================] - 0s 25ms/step
tore little year im strong poor quality also one handle got
1/1 [==============================] - 0s 27ms/step
tore little year im strong poor quality also one handle got bent
1/1 [==============================] - 0s 25ms/step
tore little year im strong poor quality also one handle got bent still
1/1 [==============================] - 0s 24ms/step
tore little year im strong poor quality also one handle got bent still work
1/1 [==============================] - 0s 28ms/step
tore little year im strong poor quality also one handle got bent still work tear
1/1 [==============================] - 0s 25ms/step
tore little year im strong poor quality also one handle got bent still work tear
painful
1/1 [==============================] - 0s 27ms/step
tore little year im strong poor quality also one handle got bent still work tear
painful suggest
1/1 [==============================] - 0s 26ms/step
tore little year im strong poor quality also one handle got bent still work tear
painful suggest must
```

1.14 Conclusion:

After reviewing the model performance evaluation results we can conclude that while the model training accuracy increases with the number of epochs, the validation accuracy remains stagnant and decreases with time.

Also, while the model training loss decreases with the number of epochs, the validation loss increases consistently increases with time which shows that the model is overfitting and is not able to generalize well on unseen data.

1.15 Convert the file into pdf and html format

[64]:
```
%%shell
jupyter nbconvert --to html ///content/HW2_RNN_Negative_Reviews.ipynb
```

```
[NbConvertApp] Converting notebook ///content/HW2_RNN_Negative_Reviews.ipynb to
html
[NbConvertApp] Writing 1339577 bytes to /content/HW2_RNN_Negative_Reviews.html
```

[64]: