Report_SER

August 31, 2020

1 Introduction

Do not spend too much time trying to get very tiny metrics improvement. Once you have a model with a correct predictive power, you should better spend time explaining your data cleaning & preparation pipeline as well as explanations & visualizations of the results.

The goal is to see your fit with our company culture & engineering needs, spending 50h on an over-complicated approach will not give you bonus points compared to a simple, yet effective, to-the-point solution.

1.1 About the data

The dataset you will be working with is called Emo-DB and can be found here.

It is a database containing samples of emotional speech in German. It contains samples labeled with one of 7 different emotions: Anger, Boredom, Disgust, Fear, Happiness, Sadness and Neutral.

Please download the full database and refer to the documentation to understand how the samples are labeled (see "Additional information")

The goal of this project is to develop a model which is able to **classify samples of emotional speech**. Feel free to use any available library you would need, but beware of re-using someone else's code without mentionning it!

1.2 Deliverable

The end-goal is to deliver us a zip file containing: * This report filled with your approach, in the form of an **iPython Notebook**. * A **5-10 slides PDF file**, containing a technical presentation covering the important aspects of your work * A Dockerfile which defines a container for the project. The container should handle everything (download the data, run the code, etc…). When running the container it should expose the jupyter notebook on one port and expose a Flask API on another one. The Flask app contains two endpoints: - One for training the model - One for querying the last trained model with an audio file of our choice in the dataset * A README.md which should contain the commands to build and run the docker container, as well as how to perform the queries to the API. * Any necessary .py, .sh or other files needed to run your code.

2 Notes

Parts of the code are inspired from the following sources:

- https://github.com/marcogdepinto/emotion-classification-from-audio-files
- https://github.com/seth814/Audio-Classification
- https://github.com/jameslyons/python_speech_features
- https://github.com/ajhalthor/audio-classifier-convNet

3 Libraries Loading

[1]:
```
import librosa
from librosa import display
import os
import matplotlib.pyplot as plt
from matplotlib.pyplot import specgram
import glob
import pandas as pd
import numpy as np
from tqdm import tqdm
from scipy.io import wavfile
from python_speech_features import mfcc, logfbank
```

[2]:
```
from keras.layers import Conv2D, MaxPool2D, Flatten, LSTM
from keras.layers import Dropout, Dense, TimeDistributed
from keras.models import Sequential
from keras.utils import to_categorical
from sklearn.utils.class_weight import compute_class_weight
from keras.callbacks import ModelCheckpoint

#import tensorflow as tf
import keras
from keras.preprocessing import sequence
from keras.layers import Embedding
from keras.utils import to_categorical
from keras.layers import Input, Activation
from keras.layers import Conv1D, Conv2D, MaxPooling1D, MaxPooling2D,␣
↪AveragePooling1D
from keras.models import Model
from keras.optimizers import Adadelta
from keras import regularizers
```

```
Using TensorFlow backend.
```

[3]:
```
import pickle
import sys
```

[4]:
```python
sys.path.append('../scripts')
from cfg import Config
from pickledataset import PickleDataset
```

[5]:
```python
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
from sklearn.ensemble import BaggingClassifier
from sklearn.svm import SVC
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪plot_confusion_matrix
```

[6]:
```python
import IPython.display as ipd
```

4 Data Preparation & Cleaning

[7]:
```python
pwd
```

[7]: '/home/jupyter/tutorials/tf2_course/visium/notebooks'

Data example

Lets look at a single audio file to see how it is structured.

[8]:
```python
signal, _ = librosa.load('../data/wav/03a01Fa.wav')
sr, _ = wavfile.read('../data/wav/03a01Fa.wav')
```

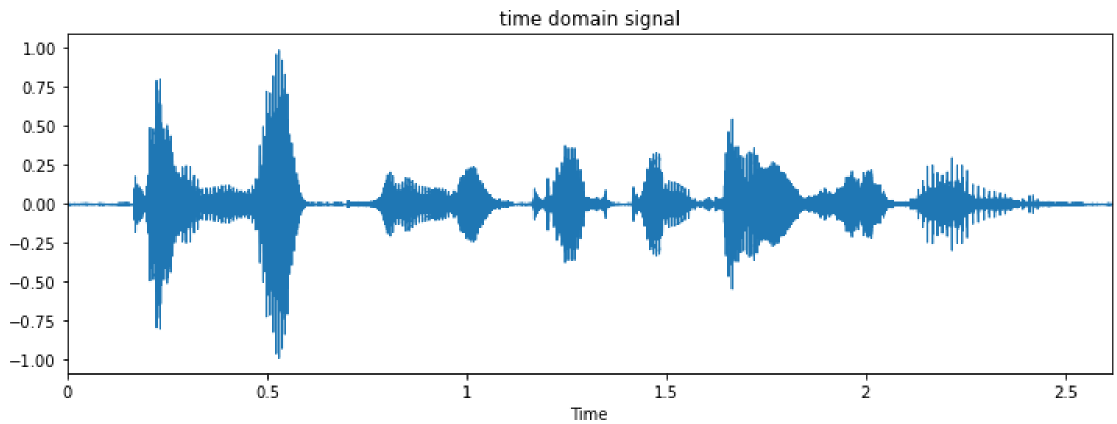
[9]:
```python
print(signal.shape)
print(sr)
```

```
(41857,)
16000
```

We see that the audio file has one channel (mono) and it has a sampling rate frequency of 16 kHz.

[10]:
```python
fig, ax = plt.subplots(figsize=(12,4))
librosa.display.waveplot(signal, sr=sr)
plt.title('time domain signal')
```

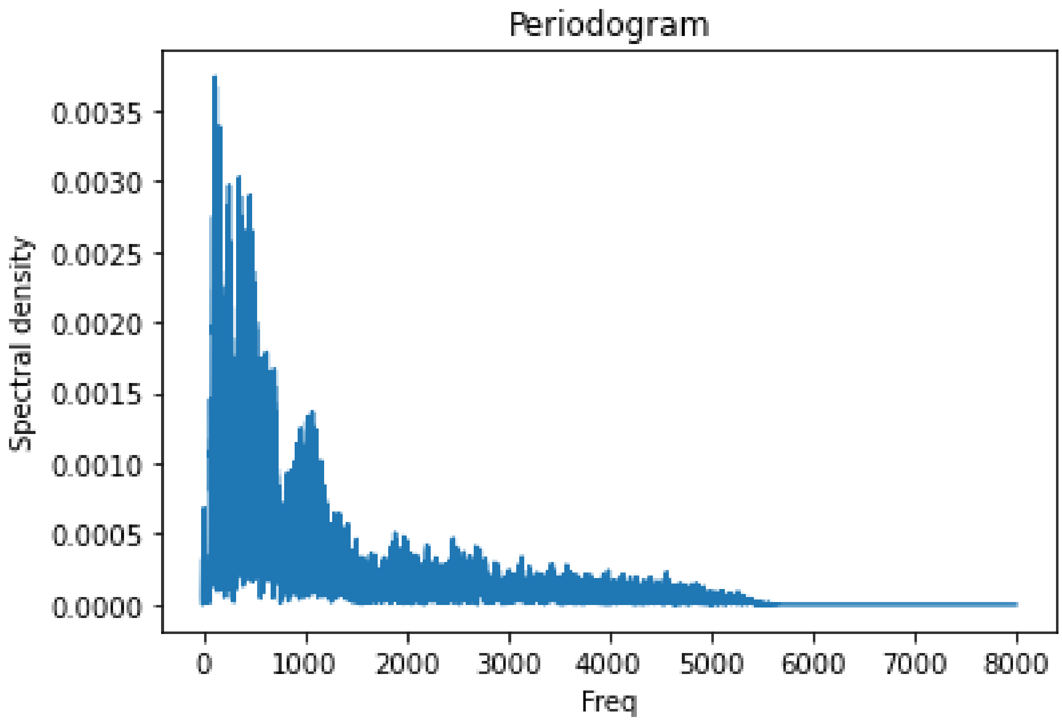
```
plt.show()
```



Usually, one analyzes time-series data in the frequency domain and not the time domain. To switch from time domain to frequency domain, we can compute the FFT of the data to obtain a periodogram (the power spectral density estimate) which will highlight the important frequencies.

[11]:
```
def calc_fft(y, rate):
    n = len(y)
    freq = np.fft.rfftfreq(n, d=1/rate)
    Y = abs(np.fft.rfft(y)/n)
    return (Y, freq)
```

[12]:
```
fft = calc_fft(signal, sr)
data = list(fft)
Y, freq = data[0], data[1]

fig, ax = plt.subplots()
plt.plot(freq, Y)
plt.title('Periodogram')
plt.ylabel('Spectral density')
plt.xlabel('Freq')
plt.show()
```

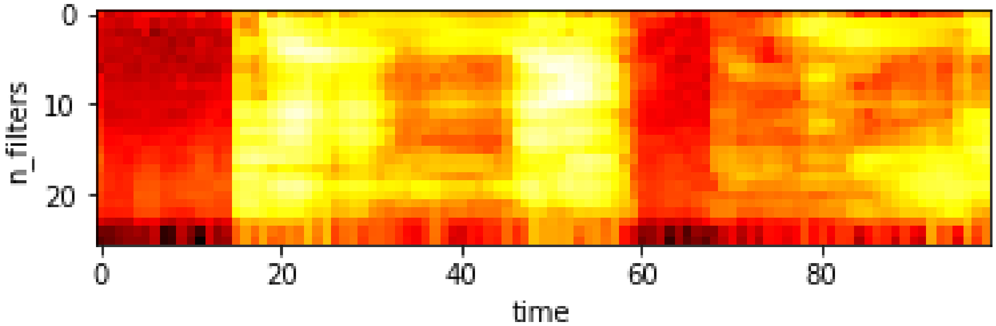


The periodogram goes to 8 kHz, but audio is typically recorded at 44.1 kHz. The Nyquist freq is 44.1 / 2 = 22.5 kHz and is the highest frequency from the environment we can represent. We usually downsample audio to 16 kHz (most information contained in the audio is below 8 kHz which would be the Nyquist freq).

We take the Short Time Fourier Transform where we use small intervals of audio (assume stationarity), take a window size of 25 ms (standard practice), slide the window forward 10 ms, use a Hanning window for the FFT computation to avoid spectral leakage. We then obtain 2d images with "signatures" for each sound.

https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

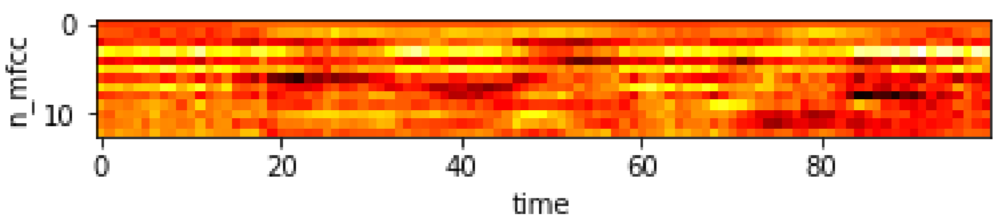
[13]:
```
bank = logfbank(signal[:sr], sr, nfilt=26, nfft=400).T

fig, ax = plt.subplots()
plt.imshow(bank,cmap='hot', interpolation='nearest')
plt.ylabel('n_filters')
plt.xlabel('time')
plt.show()
```



As humans, it's easy to tell apart low freqs (10 vs 100 Hz), but difficult for high freqs (15'000 vs 15'100 Hz). In order to differentiate higher freqs, we scale freqs with log to obtain the Mel scale. We care about differences in small freqs not differences in large freqs. To do so, we use a filter bank (26 triangular filters to bin energies) and build input features based on the power spectral density (image of size 26x100 if taking 1s of data).

http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[14]:
```
mfccs = mfcc(signal[:sr], sr, numcep=13, nfilt=26, nfft=400).T

fig, ax = plt.subplots()
plt.imshow(mfccs, cmap='hot', interpolation='nearest')
plt.ylabel('n_mfcc')
plt.xlabel('time')
plt.show()
```



By taking the STFT, we have overlap in samples which leads to correlated samples. To decorrelate the samples, we perform a DCT on the filter bank energies and get 26x100 samples again. We usually take a fraction of the mfccs (eg. the first 13, or 40 mfccs since we are only interested in the low freqs).

http://datagenetics.com/blog/november32012/index.html

Finally, we can train a model to classify audio samples based on their mfcc decomposition. One could imagine using a CNN model since the mffcs are 2-dimensional. However, as we will see, we can apply more traditional models as well.

4.0.1 EDA

[15]:
```
path = '../data/wav'
df = os.listdir(path)
print(df[:5])
```

```
['16b10Fb.wav', '16a01Tb.wav', '12b03Ta.wav', '11b09Wa.wav', '09a05Lc.wav']
```

We see that the audio files are structured as follows:

- the first two digits indicate the speaker id (eg. 16)
- the next three characters/digits indicate the spoken sentence (eg. b10)
- the next character indicates the true emotion (eg. F)
- the last character before the file extension indicates the version (eg. a)

[16]:
```
speaker = [file[:2] for file in df]
text = [file[2:5] for file in df]
emotion = [file[5] for file in df]

print(np.unique(speaker), len(np.unique(speaker)))
print(np.unique(text), len(np.unique(text)))
print(np.unique(emotion), len(np.unique(emotion)))
```

```
['03' '08' '09' '10' '11' '12' '13' '14' '15' '16'] 10
['a01' 'a02' 'a04' 'a05' 'a07' 'b01' 'b02' 'b03' 'b09' 'b10'] 10
['A' 'E' 'F' 'L' 'N' 'T' 'W'] 7
```

- We see that there are 10 different speakers.
- There are 10 different spoken sentences.
- There are 7 different emotions (classes).

[17]:
```
df = pd.DataFrame(df, columns=['filename'])
df['speaker'] = speaker
df['text'] = text
df['emotion'] = emotion
print(df)
```

```
        filename speaker text emotion
0    16b10Fb.wav      16  b10       F
1    16a01Tb.wav      16  a01       T
2    12b03Ta.wav      12  b03       T
3    11b09Wa.wav      11  b09       W
4    09a05Lc.wav      09  a05       L
..           ...     ...  ...     ...
530  15b02Nd.wav      15  b02       N
531  11b10Ld.wav      11  b10       L
532  16b03Ad.wav      16  b03       A
```

```
533  03a02Wb.wav      03  a02       W
534  14a01Na.wav      14  a01       N

[535 rows x 4 columns]
```

[18]:
```
classes = np.unique(df.emotion)
num_classes = len(classes)
print(classes)
print(num_classes)
```

```
['A' 'E' 'F' 'L' 'N' 'T' 'W']
7
```

The classes are: - A: Angst - E: Ekel - F: Freude - L: Langeweile - N: Neutral - T: Trauer - W: Wut (Ärger)

We compute the length of each file in seconds.

[19]:
```
df.set_index(df.filename, inplace=True)
for file in df.index:
    rate, signal = wavfile.read(f'{path}/{file}')
    df.at[file, 'length'] = signal.shape[0] / rate
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: WavFileWarning:
Chunk (non-data) not understood, skipping it.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[20]:
```
emotion_count_by_class = df.groupby('emotion')['filename'].count()
```

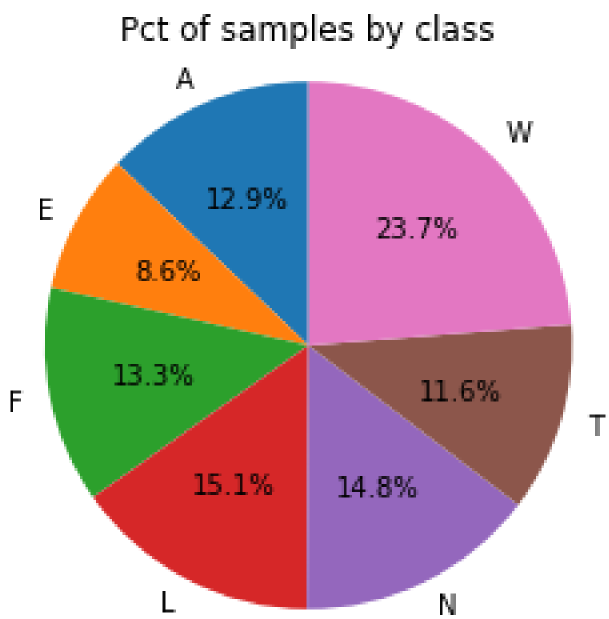
[21]:
```
emotion_count_by_class_pct = emotion_count_by_class / emotion_count_by_class.
 ↪sum()
```

[22]:
```
emotion_count_by_class_pct
```

[22]:
```
emotion
A    0.128972
E    0.085981
F    0.132710
L    0.151402
N    0.147664
T    0.115888
W    0.237383
Name: filename, dtype: float64
```

We see that the classes are not all equally balanced. There are slightly more 'W' than the rest (3x more 'W' than 'E'). Class balancing techniques may be considered in further analysis.

[23]:
```
fig, ax = plt.subplots()
ax.set_title('Pct of samples by class')
ax.pie(emotion_count_by_class_pct, labels=emotion_count_by_class_pct.index,␣
 ↪autopct='%1.1f%%',
      shadow=False, startangle=90)
ax.axis('equal')
plt.show()
```



[24]:
```
avg_sample_length_by_class = df.groupby('emotion')['length'].mean()
```

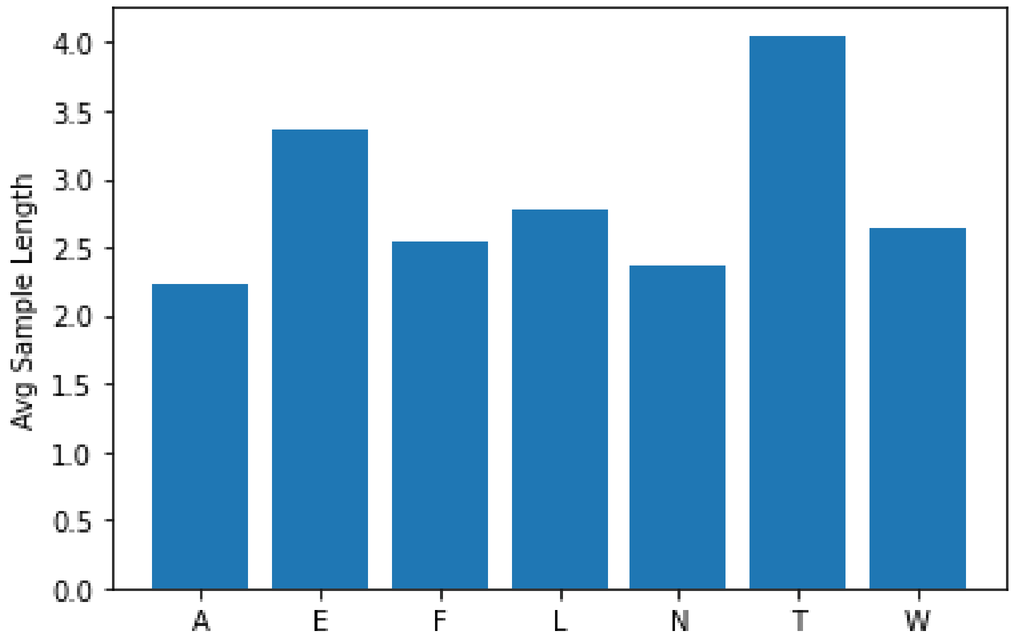
[25]:
```
avg_sample_length_by_class
```

[25]:
```
emotion
A    2.233377
E    3.352834
F    2.543967
L    2.778977
N    2.359236
T    4.052895
W    2.640795
Name: length, dtype: float64
```

We see that the average length of samples by class is approximately the same. Class 'T' has some longer samples.

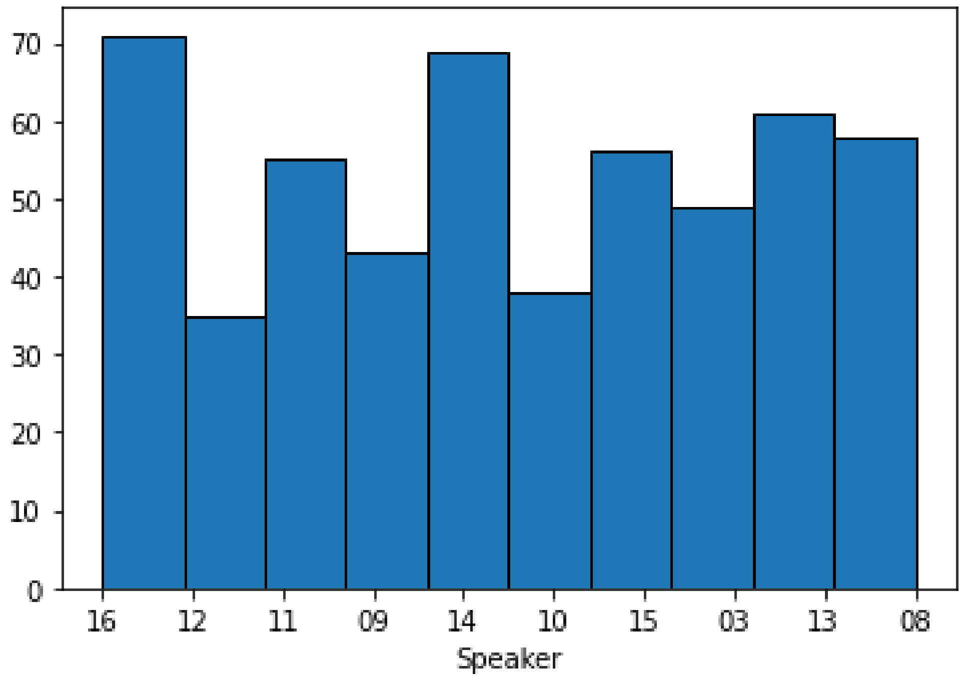
[26]:

```
fig, ax = plt.subplots()
ax.bar(np.arange(num_classes), avg_sample_length_by_class)
ax.set_ylabel('Avg Sample Length')
ax.set_xticks(np.arange(num_classes))
ax.set_xticklabels(classes)
plt.show()
```



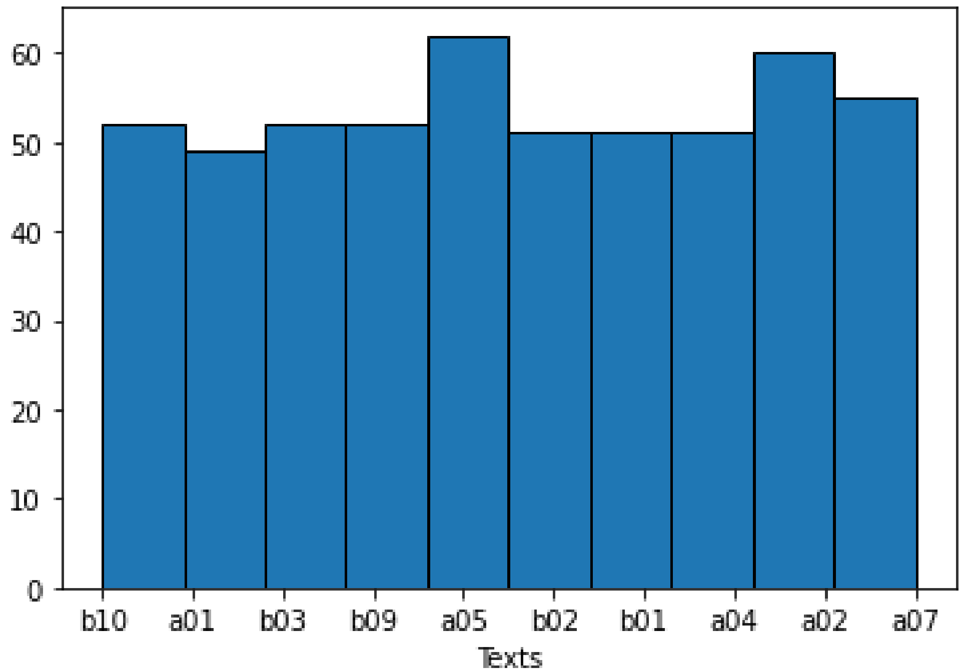
[27]:

```
fig, ax1 = plt.subplots()
df['speaker'].hist(ax=ax1, bins=10, edgecolor='black')
ax1.grid(False)
ax1.set_xlabel('Speaker')
plt.show()
```



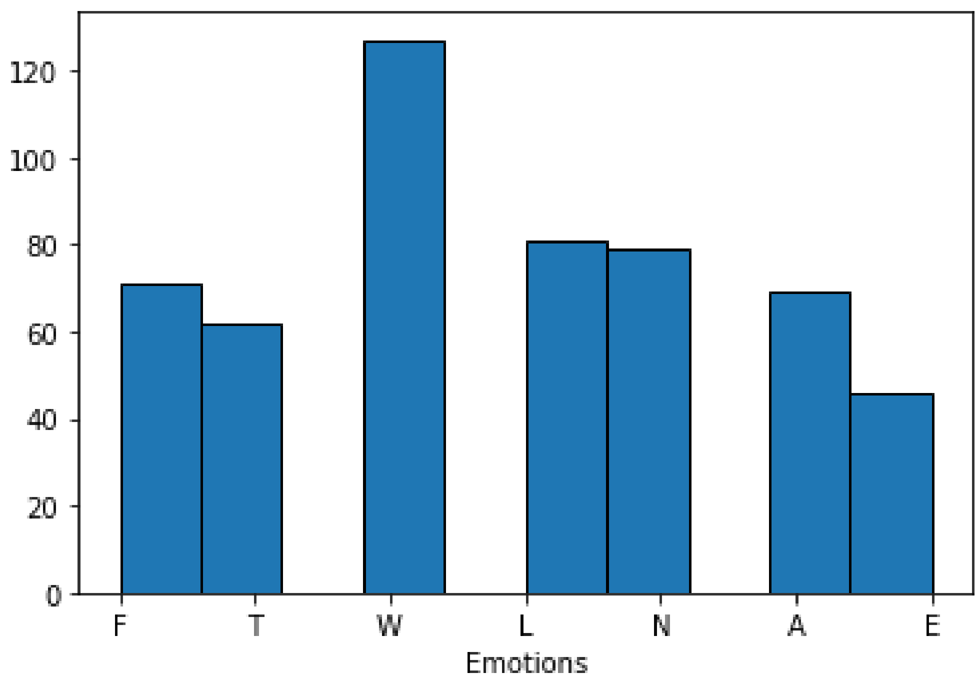
We see that all speakers roughly recorded the same amount of audio samples.

[28]:
```
fig, ax1 = plt.subplots()
df['text'].hist(ax=ax1, bins=10, edgecolor='black')
ax1.grid(False)
ax1.set_xlabel('Texts')
plt.show()
```



We see that texts are uniformly distributed.

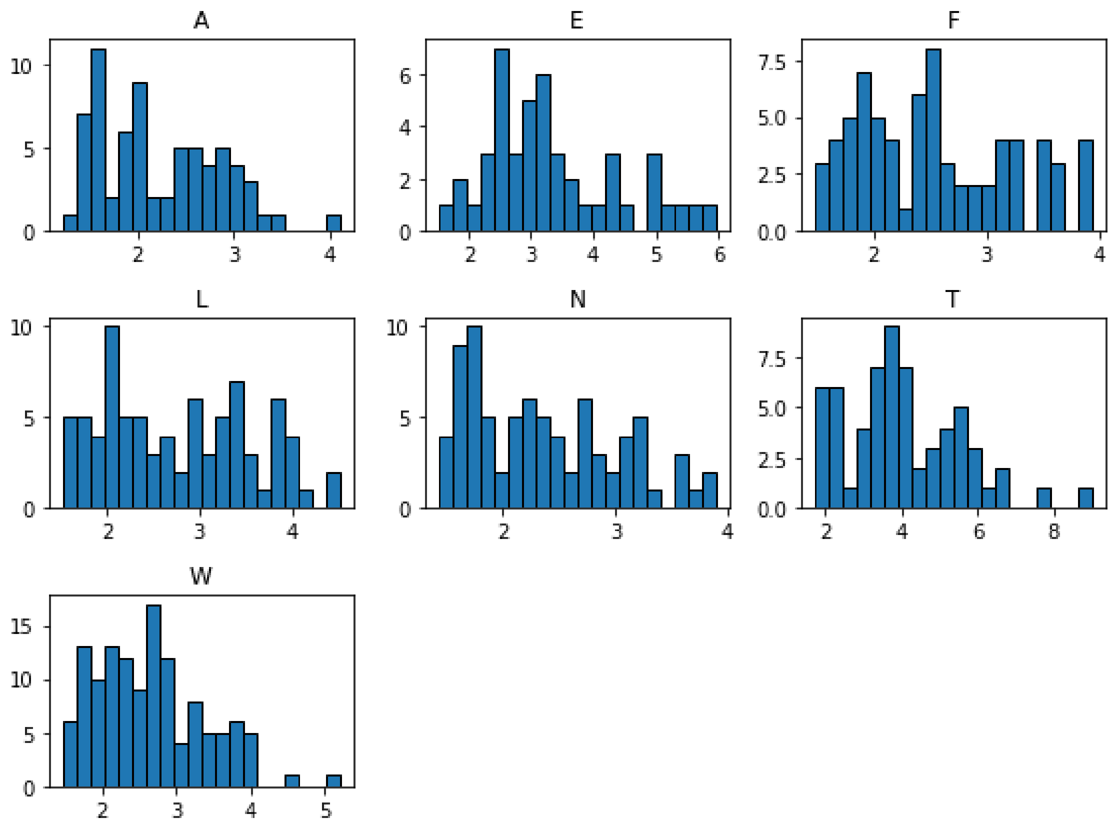
[29]:
```
fig, ax1 = plt.subplots()
df['emotion'].hist(ax=ax1, bins=10, edgecolor='black')
ax1.grid(False)
ax1.set_xlabel('Emotions')
plt.show()
```



We see that class 'W' is a lot more present than the others. This might cause issues during modelling so it is good to keep in mind. One could consider undersampling class 'W' in the training set, or oversample the other classes.

[30]:
```
fig, ax1 = plt.subplots(figsize=(8, 6))
df['length'].hist(by=df['emotion'], xrot=0, ax=ax1, bins=20, edgecolor='black')
plt.tight_layout()
plt.show()
```

```
/opt/conda/lib/python3.7/site-packages/pandas/plotting/_matplotlib/hist.py:335:
UserWarning: To output multiple subplots, the figure containing the passed axes
is being cleared
  **kwds,
```



[31]:
```
df.groupby('emotion')['length'].describe()
```

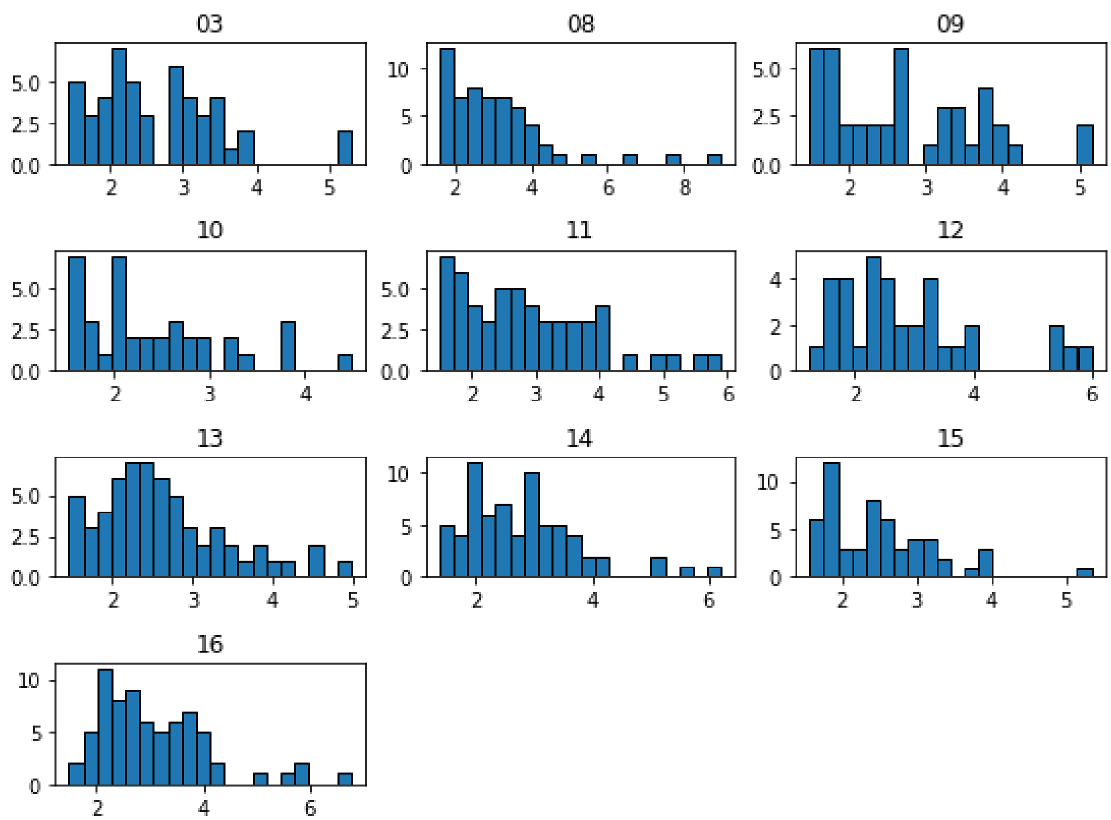
[31]:

emotion	count	mean	std	min	25%	50%	75%	max
A	69.0	2.233377	0.637358	1.225500	1.607938	2.081313	2.711750	4.101375
E	46.0	3.352834	1.073298	1.523813	2.552859	3.117188	3.943766	5.963813
F	71.0	2.543967	0.682695	1.481375	1.963625	2.463563	3.106813	3.930938
L	81.0	2.778977	0.804450	1.520063	2.074688	2.690562	3.398750	4.525812
N	79.0	2.359236	0.659165	1.430813	1.769750	2.235500	2.822000	3.899188
T	62.0	4.052895	1.532625	1.735688	3.076344	3.863375	5.102375	8.978250
W	127.0	2.640795	0.728803	1.465812	2.090094	2.609875	3.116937	5.213500

The last histogram plots and the table show that distribution of sample length grouped by emotion is roughly the same. Looking at the IQR, we see that most of the data is between 2 and 3 seconds, apart for samples with emotion ‘E’ and ‘T’.

[32]:
```
fig, ax1 = plt.subplots(figsize=(8, 6))
df['length'].hist(by=df['speaker'], xrot=0, ax=ax1, bins=20, edgecolor='black')
plt.tight_layout()
plt.show()
```

```
/opt/conda/lib/python3.7/site-packages/pandas/plotting/_matplotlib/hist.py:335:
UserWarning: To output multiple subplots, the figure containing the passed axes
is being cleared
  **kwds,
```



[33]:
```
df.groupby('speaker')['emotion'].describe()['count']
```

[33]:
```
speaker
03      49
08      58
09      43
10      38
11      55
```

```
12    35
13    61
14    69
15    56
16    71
Name: count, dtype: object
```

We see that the distribution of texts by speaker is roughly uniform. This is good since a given speaker may introduce bias in recordings if his audio samples overwhelm the others.

[34]:
```
rate = []
for file in df.filename:
    sr, _ = wavfile.read(f'{path}/{file}')
    rate.append(sr)

np.unique(rate)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: WavFileWarning:
Chunk (non-data) not understood, skipping it.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[34]:
```
array([16000])
```

We only have 16 kHz audio files.

5 Feature Engineering & Modeling

In this first modelling approach, we will extract the mfccs from the data and use these as features as mentionned in the 'Data Preparation and Cleaning' section.

We will also augment the data by varying time (lenghtening/shortening audio) and pitch. We will only augment the training and validation data to avoid information leakage into the test set.

5.0.1 vary time

[35]:
```
for file in os.listdir('../data/wav'):
    y, sr = librosa.load(f'../data/wav/{file}')
    y_changed = librosa.effects.time_stretch(y, rate=0.80)
    wavfile.write(filename=f'../data/aug/{file.split(".")[0]}_fast.wav',␣
 ↪rate=sr, data=y_changed)

for file in os.listdir('../data/wav'):
    y, sr = librosa.load(f'../data/wav/{file}')
    y_changed = librosa.effects.time_stretch(y, rate=1.20)
```

```
    wavfile.write(filename=f'../data/aug/{file.split(".")[0]}_slow.wav',␣
↪rate=sr, data=y_changed)
```

We can compare the difference between the raw sample and a lengthened file.

[36]:
```
# raw file
ipd.Audio(y, rate=sr)
```

[36]: <IPython.lib.display.Audio object>

[37]:
```
# augmented file
ipd.Audio(y_changed, rate=sr)
```

[37]: <IPython.lib.display.Audio object>

5.0.2 vary pitch

[38]:
```
steps = [-2.5, -2, -1.5, -1, -0.5, 0.5,  1, 1.5, 2, 2.5]

for n_step in steps:
    for file in os.listdir('../data/wav'):
        y, sr = librosa.load(f'../data/wav/{file}')
        y_changed = librosa.effects.pitch_shift(y, sr, n_steps=n_step)
        wavfile.write(filename=f'../data/aug/{file.split(".
↪")[0]}_pitch_{n_step}.wav', rate=sr, data=y_changed)
```

5.0.3 downsampling

Downsample to 8 kHz.

[39]:
```
for file in os.listdir('../data/wav'):
    y, sr = librosa.load(f'../data/wav/{file}', sr=8000)
    wavfile.write(filename=f'../data/aug/{file}_downsampled_8khz.wav', rate=sr,␣
↪data=y)
```

The augmented dataset consists of:

- 535 normal samples.
- 2675 samples 'positively' pitched modulated 0.5 or more semitones higher.
- 2675 samples 'negatively' pitched modulated 0.5 or more semitones lower.
- 535 samples Slowed down to 0.80.
- 535 samples speed up by 1.20.
- 535 samples downsampled to 8 kHz.

Totalling 6955 samples.

5.0.4 Remove audio "deadspace"

We compute the envelope of the signal (the estimation of signal magnitude). If the signal is below the envelope, we consider the audio has died out and remove it.

[40]:
```
def envelope(y, rate, threshold):
    mask = []
    y = pd.Series(y).apply(np.abs)
    y_mean = y.rolling(window=int(0.1*rate), min_periods=1, center=True).mean()
    for mean in y_mean:
        if mean > threshold:
            mask.append(True)
        else:
            mask.append(False)
    return mask
```

[41]:
```
for file in os.listdir('../data/aug'):
    signal, sr = librosa.load(f'../data/aug/{file}', sr=16000)
    mask = envelope(signal, sr, 0.0005)
    signal = signal[mask]
    wavfile.write(filename=f'../data/aug/{file}', rate=sr, data=signal)
```

5.0.5 mfccs

[44]:
```
lst = []
for file in glob.glob('../data/wav/*.wav'):

    data, sampling_rate = librosa.load(file, res_type='kaiser_fast')
    mfccs = np.mean(librosa.feature.mfcc(y=data, sr=sampling_rate, n_mfcc=40),␣
↪axis=1)
    file = file.split('/')[-1]
    arr = mfccs, file[5]
    lst.append(arr)
```

[45]:
```
X, y = zip(*lst)
y_test = np.asarray(y)
X_test = np.asarray(X)
print(X_test.shape, y_test.shape)
```

```
(535, 40) (535,)
```

This is the raw data. We will keep this as the test set. Below, we will construct the training/val sets with augmented data.

[46]:
```
lst = []
for file in glob.glob('../data/aug/*.wav'):
```

```
    data, sampling_rate = librosa.load(file, res_type='kaiser_fast')
    mfccs = np.mean(librosa.feature.mfcc(y=data, sr=sampling_rate, n_mfcc=40),␣
↪axis=1)
    file = file.split('/')[-1]
    arr = mfccs, file[5]
    lst.append(arr)
```

We get 40 mfccs per audio file, with each mfcc containing 112 values/features. We then take the mean value of each mfcc feature, to obtain 40 mfccs for each audio file

[47]:
```
X, y = zip(*lst)
y = np.asarray(y)
X = np.asarray(X)
```

5.0.6 Train/Test split

[48]:
```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size = 0.2,␣
↪random_state=42)
```

[49]:
```
print(X_train.shape, X_val.shape, X_test.shape)
print(y_train.shape, y_val.shape, y_test.shape)
```

```
(5564, 40) (1391, 40) (535, 40)
(5564,) (1391,) (535,)
```

Since the augmentation process is long, we save the final dataset as a pickle for quick loading in the future.

[50]:
```
pickle_data = PickleDataset()
pickle_data.train = (X_train, y_train)
pickle_data.val = (X_val, y_val)
pickle_data.test = (X_test, y_test)
```

[51]:
```
with open('../pickles/data.pkl', 'wb') as handle:
        # for compatibility with python 2: protocol=2
        pickle.dump(pickle_data, handle, protocol=2)
```

load data

[52]:
```
with open('../pickles/data.pkl', 'rb') as handle:
    data = pickle.load(handle)

X_train, y_train = zip(data.train)
X_train, y_train = np.array(X_train)[0], np.array(y_train)[0]
X_val, y_val = zip(data.val)
X_val, y_val = np.array(X_val)[0], np.array(y_val)[0]
X_test, y_test = zip(data.test)
```

```python
X_test, y_test = np.array(X_test)[0], np.array(y_test)[0]
```

6 Results & Visualizations

6.0.1 Multinomial Regression

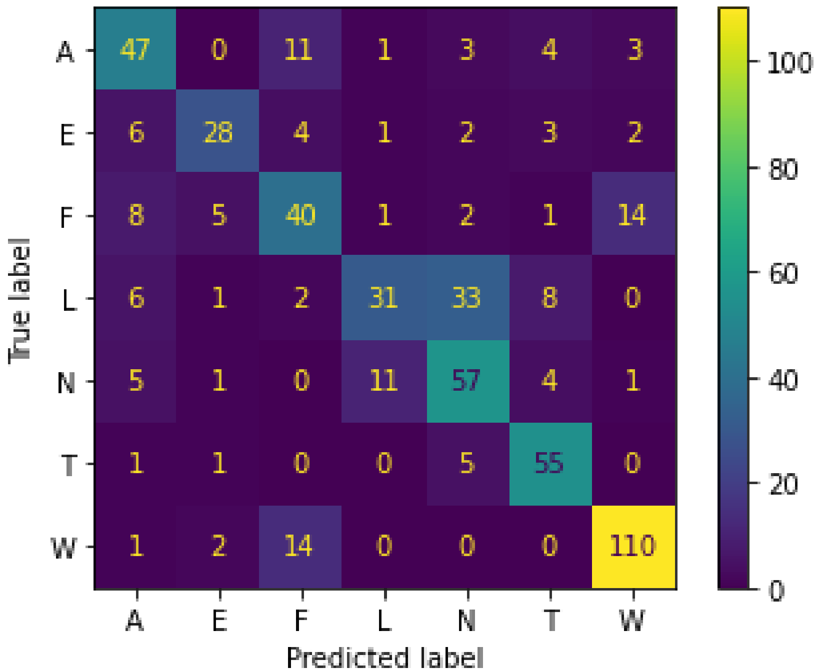
[53]:
```python
train_samples = X_train.shape[0]
mnr = LogisticRegression(multi_class='ovr',
                         solver='saga', tol=0.1).fit(X_train, y_train)

predictions = mnr.predict(X_test)
score = mnr.score(X_test, y_test)
print("Test score with: %.4f" % score)
```

```
Test score with: 0.6879
```

[54]:
```python
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       0.64      0.68      0.66        69
           E       0.74      0.61      0.67        46
           F       0.56      0.56      0.56        71
           L       0.69      0.38      0.49        81
           N       0.56      0.72      0.63        79
           T       0.73      0.89      0.80        62
           W       0.85      0.87      0.86       127

    accuracy                           0.69       535
   macro avg       0.68      0.67      0.67       535
weighted avg       0.69      0.69      0.68       535
```

[55]:
```python
plot_confusion_matrix(mnr, X_test, y_test)
plt.show()
```



6.0.2 kNN

[56]:
```
knn = KNeighborsClassifier(n_neighbors=3, weights='distance', algorithm='auto',
                           leaf_size=5, metric='euclidean').fit(X_train, y_train)
predictions = knn.predict(X_test)
score = knn.score(X_test, y_test)

print("Test score: %.4f" % score)
```

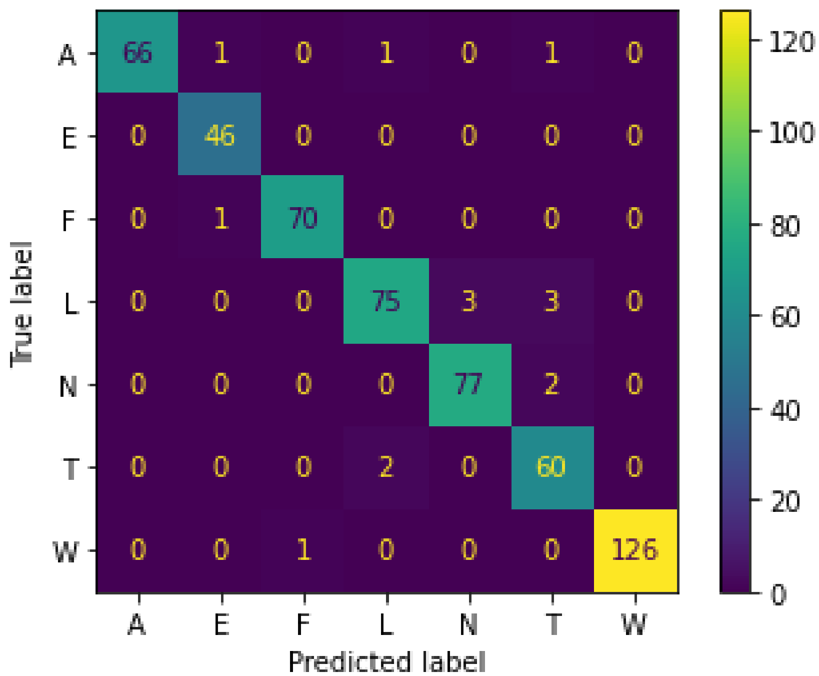
```
Test score: 0.9720
```

[57]:
```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       1.00      0.96      0.98        69
           E       0.96      1.00      0.98        46
           F       0.99      0.99      0.99        71
           L       0.96      0.93      0.94        81
           N       0.96      0.97      0.97        79
           T       0.91      0.97      0.94        62
           W       1.00      0.99      1.00       127

    accuracy                           0.97       535
```

```
   macro avg       0.97      0.97      0.97       535
weighted avg       0.97      0.97      0.97       535
```

[58]:
```
plot_confusion_matrix(knn, X_test, y_test)
plt.show()
```



save kNN model

[59]:
```
with open('../pickles/knn.pkl', 'wb') as model_pkl:
    pickle.dump(knn, model_pkl)
```

6.0.3 CV

[60]:
```
parameters = {'n_neighbors':[3, 5, 10, 15, 20, 30, 40, 50], 'weights':
 ↪('uniform', 'distance'),
               'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],␣
 ↪'leaf_size': np.arange(5, 50, 5),
              'metric': ['euclidean', 'manhattan', 'chebyshev', 'minkowski']}
```

[61]:
```
clf = GridSearchCV(knn, parameters)
clf.fit(X_train, y_train)
```

[61]:
```
GridSearchCV(estimator=KNeighborsClassifier(leaf_size=5, metric='euclidean',
                                            n_neighbors=3, weights='distance'),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'leaf_size': array([ 5, 10, 15, 20, 25, 30, 35, 40,
45]),
                         'metric': ['euclidean', 'manhattan', 'chebyshev',
                                    'minkowski'],
                         'n_neighbors': [3, 5, 10, 15, 20, 30, 40, 50],
                         'weights': ('uniform', 'distance')})
```

[62]:
```
clf.best_params_
```

[62]:
```
{'algorithm': 'auto',
 'leaf_size': 5,
 'metric': 'manhattan',
 'n_neighbors': 3,
 'weights': 'distance'}
```

[63]:
```
clf.best_score_
```

[63]:
```
0.9399703956511342
```

[64]:
```
clf.cv_results_['mean_test_score']
```

[64]:
```
array([0.92972862, 0.93817636, 0.91588988, …, 0.81470053, 0.72915075,
       0.7945707 ])
```

6.0.4 Decision Tree

[65]:
```
dtree = DecisionTreeClassifier().fit(X_train, y_train)
predictions = dtree.predict(X_test)
score = dtree.score(X_test, y_test)

print("Test score: %.4f" % score)
```

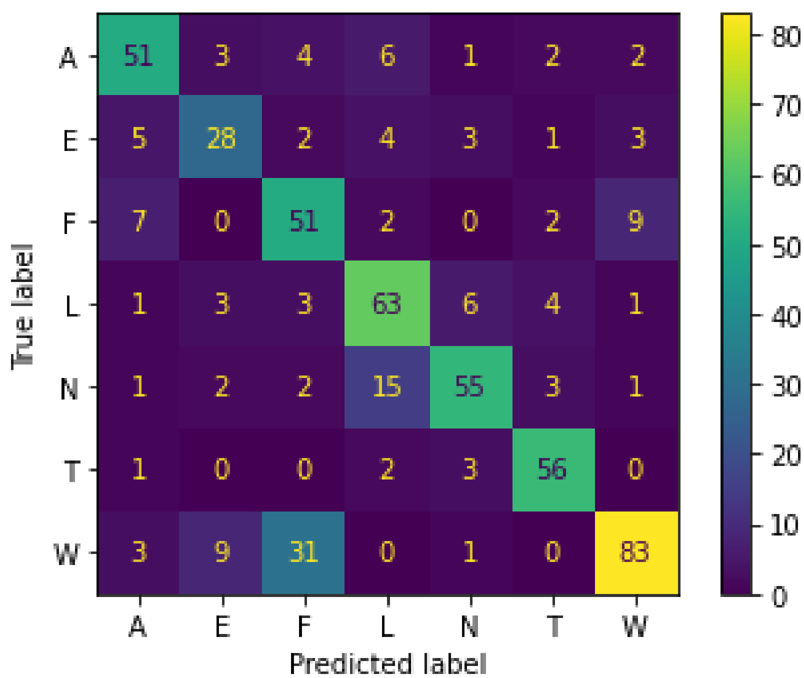
```
Test score: 0.7234
```

[66]:
```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       0.74      0.74      0.74        69
           E       0.62      0.61      0.62        46
           F       0.55      0.72      0.62        71
           L       0.68      0.78      0.73        81
           N       0.80      0.70      0.74        79
           T       0.82      0.90      0.86        62
```

```
           W       0.84      0.65      0.73       127

    accuracy                           0.72       535
   macro avg       0.72      0.73      0.72       535
weighted avg       0.74      0.72      0.73       535
```

[67]:
```
plot_confusion_matrix(dtree, X_test, y_test)
plt.show()
```



6.0.5 Random Forest

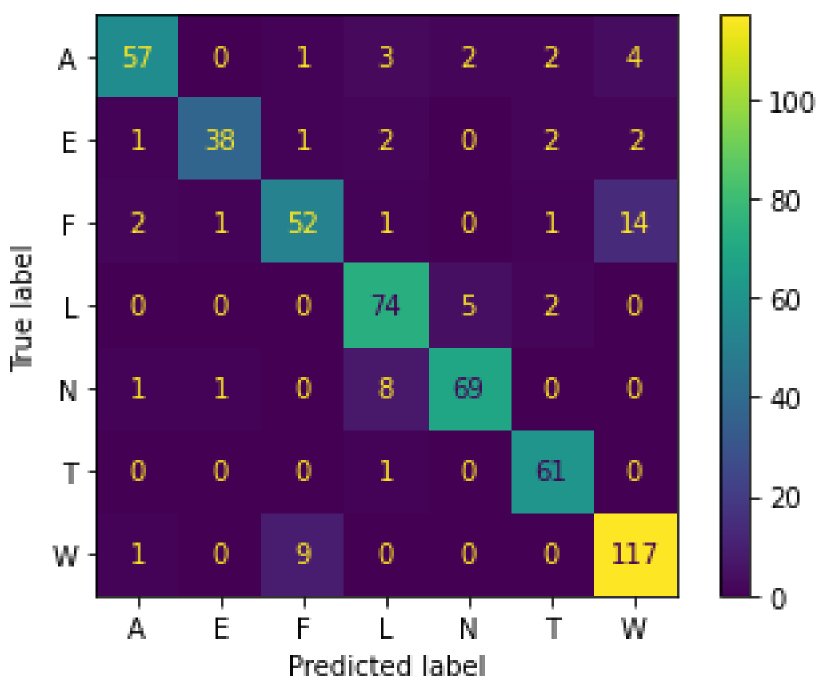
[68]:
```
rforest = RandomForestClassifier(criterion='gini', max_depth=10,␣
 ↪max_features='log2',
                                max_leaf_nodes=100, min_samples_leaf=3,␣
 ↪min_samples_split=20,
                                n_estimators=22000, random_state=42).
 ↪fit(X_train, y_train)

predictions = rforest.predict(X_test)
score = rforest.score(X_test, y_test)
print("Test score: %.4f" % score)
```

```
Test score: 0.8748
```

[69]:
```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       0.92      0.83      0.87        69
           E       0.95      0.83      0.88        46
           F       0.83      0.73      0.78        71
           L       0.83      0.91      0.87        81
           N       0.91      0.87      0.89        79
           T       0.90      0.98      0.94        62
           W       0.85      0.92      0.89       127

    accuracy                           0.87       535
   macro avg       0.88      0.87      0.87       535
weighted avg       0.88      0.87      0.87       535
```

[70]:
```
plot_confusion_matrix(rforest, X_test, y_test)
plt.show()
```



SVM

[71]:
```
svm = svm.SVC(gamma=0.001, C=100.).fit(X_train, y_train)
predictions = svm.predict(X_test)
score = svm.score(X_test, y_test)
```

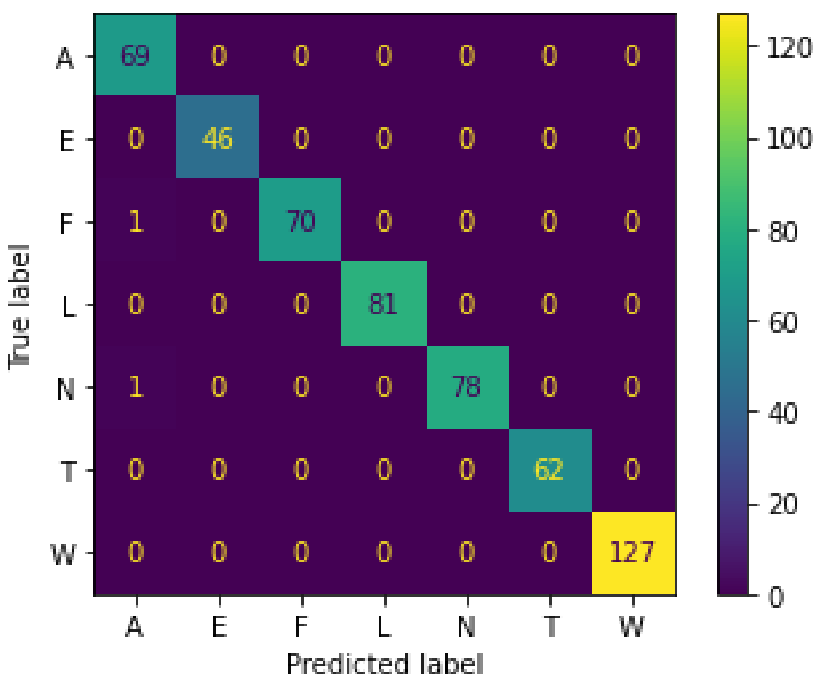
```python
print("Test score: %.4f" % score)
```

```
Test score: 0.9832
```

[72]:
```python
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       0.99      1.00      0.99        69
           E       1.00      0.98      0.99        46
           F       1.00      1.00      1.00        71
           L       0.97      0.93      0.95        81
           N       0.94      0.97      0.96        79
           T       0.98      1.00      0.99        62
           W       1.00      1.00      1.00       127

    accuracy                           0.98       535
   macro avg       0.98      0.98      0.98       535
weighted avg       0.98      0.98      0.98       535
```

[73]:
```python
plot_confusion_matrix(clf, X_test, y_test)
plt.show()
```



Bagging

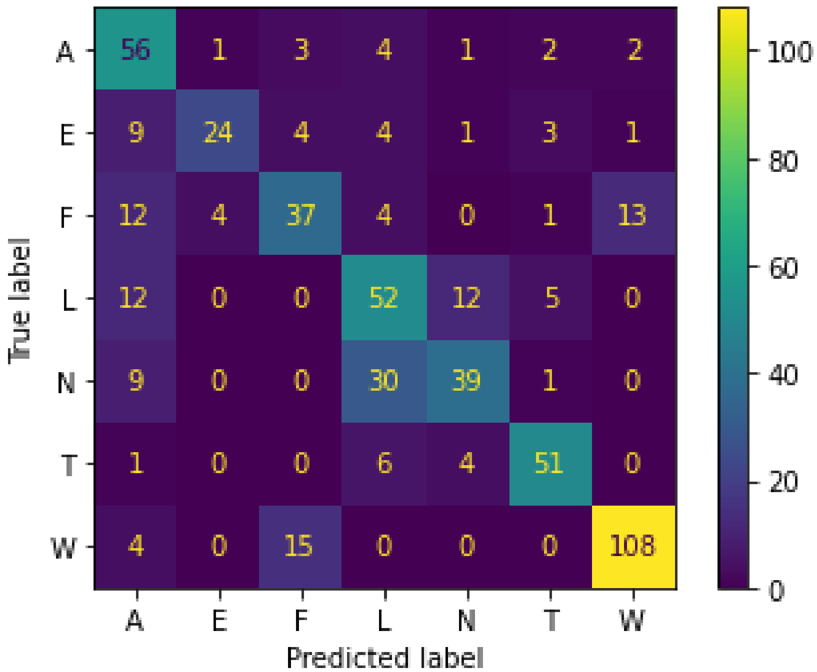
[74]:
```
svc = SVC()
svm_bag = BaggingClassifier(base_estimator=svc,
                             n_estimators=100, random_state=0).fit(X_train,␣
 ↪y_train)
predictions = svm_bag.predict(X_test)
score = svm_bag.score(X_test, y_test)

print("Test score: %.4f" % score)
```

```
Test score: 0.6860
```

[75]:
```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       0.54      0.81      0.65        69
           E       0.83      0.52      0.64        46
           F       0.63      0.52      0.57        71
           L       0.52      0.64      0.57        81
           N       0.68      0.49      0.57        79
           T       0.81      0.82      0.82        62
           W       0.87      0.85      0.86       127

    accuracy                           0.69       535
   macro avg       0.70      0.67      0.67       535
weighted avg       0.70      0.69      0.69       535
```

[76]:
```
plot_confusion_matrix(svm_bag, X_test, y_test)
plt.show()
```



6.0.6 Boosting/XGBoost

[77]:
```
gboost = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
         max_depth=1, random_state=0).fit(X_train, y_train)
predictions = gboost.predict(X_test)
score = gboost.score(X_test, y_test)

print("Test score: %.4f" % score)
```

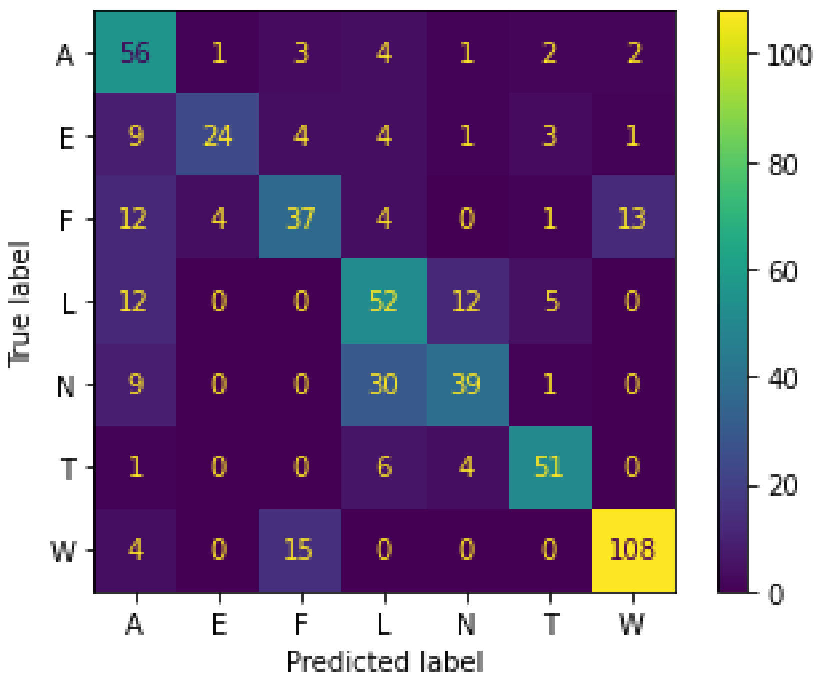
```
Test score: 0.7794
```

[78]:
```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           A       0.65      0.74      0.69        69
           E       0.82      0.78      0.80        46
           F       0.63      0.72      0.67        71
           L       0.73      0.77      0.75        81
           N       0.78      0.73      0.76        79
           T       0.92      0.92      0.92        62
           W       0.92      0.80      0.86       127

    accuracy                           0.78       535
```

```
   macro avg       0.78      0.78      0.78       535
weighted avg       0.79      0.78      0.78       535
```

[79]:
```python
plot_confusion_matrix(svm_bag, X_test, y_test)
plt.show()
```



6.0.7 Bagged kNN

[80]:
```python
bagging_knn = BaggingClassifier(KNeighborsClassifier(),
                                max_samples=0.5, max_features=0.5).
 ↪fit(X_train, y_train)

predictions = bagging_knn.predict(X_test)
score = bagging_knn.score(X_test, y_test)

print("Test score: %.4f" % score)
```

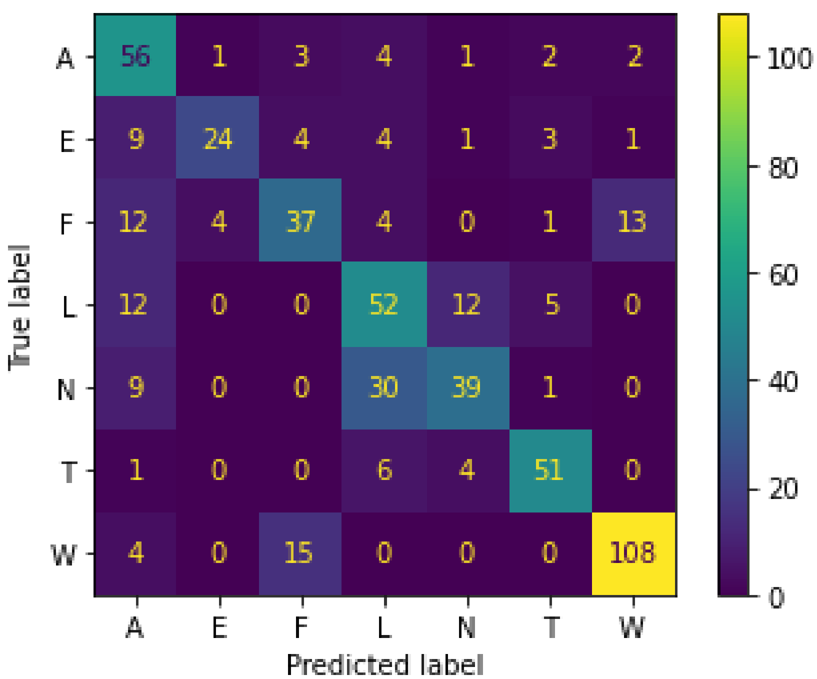
```
Test score: 0.9570
```

[81]:
```python
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support
```

```
           A       0.98      0.94      0.96        69
           E       0.94      0.98      0.96        46
           F       0.99      0.93      0.96        71
           L       0.95      0.91      0.93        81
           N       0.94      0.94      0.94        79
           T       0.93      1.00      0.96        62
           W       0.97      0.99      0.98       127

    accuracy                           0.96       535
   macro avg       0.96      0.96      0.96       535
weighted avg       0.96      0.96      0.96       535
```

[82]:
```
plot_confusion_matrix(svm_bag, X_test, y_test)
plt.show()
```



6.0.8 CNN conv1d - Keras

[83]:
```
x_traincnn = np.expand_dims(X_train, axis=2)
x_valcnn = np.expand_dims(X_val, axis=2)
x_testcnn = np.expand_dims(X_test, axis=2)
```

[84]:
```
x_traincnn.shape, x_valcnn.shape, x_testcnn.shape
```

[84]: ((5564, 40, 1), (1391, 40, 1), (535, 40, 1))

6.0.9 1. no scale:

6.0.10 model 1

[106]:
```
model = Sequential()

model.add(Conv1D(256, 5, padding='same',input_shape=(40,1)))
model.add(Activation('relu'))
model.add(Conv1D(128, 5, padding='same'))
model.add(Activation('relu'))
model.add(Dropout(0.1))
model.add(MaxPooling1D(pool_size=(8)))
model.add(Conv1D(128, 5, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(128, 5, padding='same'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(7))
model.add(Activation('softmax'))
opt = keras.optimizers.RMSprop(lr=0.00001, decay=1e-6)
model.compile(loss='categorical_crossentropy', optimizer=opt,␣
 ↪metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_4"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_13 (Conv1D)           (None, 40, 256)           1536
_________________________________________________________________
activation_16 (Activation)   (None, 40, 256)           0
_________________________________________________________________
conv1d_14 (Conv1D)           (None, 40, 128)           163968
_________________________________________________________________
activation_17 (Activation)   (None, 40, 128)           0
_________________________________________________________________
dropout_4 (Dropout)          (None, 40, 128)           0
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 5, 128)            0
_________________________________________________________________
conv1d_15 (Conv1D)           (None, 5, 128)            82048
_________________________________________________________________
activation_18 (Activation)   (None, 5, 128)            0
_________________________________________________________________
```

```
conv1d_16 (Conv1D)           (None, 5, 128)            82048
_________________________________________________________________
activation_19 (Activation)   (None, 5, 128)            0
_________________________________________________________________
flatten_4 (Flatten)          (None, 640)               0
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 4487
_________________________________________________________________
activation_20 (Activation)   (None, 7)                 0
=================================================================
Total params: 334,087
Trainable params: 334,087
Non-trainable params: 0
_________________________________________________________________
```

[86]:
```
y_train[y_train == 'W'] = 0
y_train[y_train == 'L'] = 1
y_train[y_train == 'E'] = 2
y_train[y_train == 'A'] = 3
y_train[y_train == 'F'] = 4
y_train[y_train == 'T'] = 5
y_train[y_train == 'N'] = 6

y_test[y_test == 'W'] = 0
y_test[y_test == 'L'] = 1
y_test[y_test == 'E'] = 2
y_test[y_test == 'A'] = 3
y_test[y_test == 'F'] = 4
y_test[y_test == 'T'] = 5
y_test[y_test == 'N'] = 6

y_val[y_val == 'W'] = 0
y_val[y_val == 'L'] = 1
y_val[y_val == 'E'] = 2
y_val[y_val == 'A'] = 3
y_val[y_val == 'F'] = 4
y_val[y_val == 'T'] = 5
y_val[y_val == 'N'] = 6
```

[87]:
```
y_train = to_categorical(y_train)
y_val = to_categorical(y_val)
y_test = to_categorical(y_test)
```

[ ]:
```
checkpoint = ModelCheckpoint('../models/conv1d/model.{epoch:02d}-{accuracy:.
↪4f}-{val_accuracy:.4f}-{loss:.4f}-{val_loss:.4f}.h5', monitor='val_loss',␣
↪verbose=1, mode='min', save_best_only=True,
                            save_weights_only=False, period=1)
```

[88]:
```
#history = model.fit(x_traincnn, y_train, batch_size=256, epochs=250,␣
↪validation_data=(x_valcnn, y_val),
#                    callbacks=[checkpoint])

history = model.fit(x_traincnn, y_train, batch_size=256, epochs=250,␣
↪validation_data=(x_valcnn, y_val))
```

```
Train on 5564 samples, validate on 1391 samples
Epoch 1/250
5564/5564 [==============================] - 4s 741us/step - loss: 2.6676 -
accuracy: 0.1120 - val_loss: 1.8730 - val_accuracy: 0.2825
Epoch 2/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.8278 -
accuracy: 0.2759 - val_loss: 1.6807 - val_accuracy: 0.3551
Epoch 3/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.6741 -
accuracy: 0.3411 - val_loss: 1.5613 - val_accuracy: 0.4062
Epoch 4/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.5688 -
accuracy: 0.3852 - val_loss: 1.4646 - val_accuracy: 0.4630
Epoch 5/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.4752 -
accuracy: 0.4206 - val_loss: 1.3941 - val_accuracy: 0.4709
Epoch 6/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.3998 -
accuracy: 0.4527 - val_loss: 1.3318 - val_accuracy: 0.4853
Epoch 7/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.3427 -
accuracy: 0.4783 - val_loss: 1.2800 - val_accuracy: 0.5133
Epoch 8/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.2868 -
accuracy: 0.5135 - val_loss: 1.2456 - val_accuracy: 0.5255
Epoch 9/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.2532 -
accuracy: 0.5208 - val_loss: 1.2079 - val_accuracy: 0.5363
Epoch 10/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.2135 -
accuracy: 0.5284 - val_loss: 1.1762 - val_accuracy: 0.5492
Epoch 11/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.1753 -
accuracy: 0.5383 - val_loss: 1.1432 - val_accuracy: 0.5730
Epoch 12/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.1498 -
accuracy: 0.5478 - val_loss: 1.1225 - val_accuracy: 0.5600
Epoch 13/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.1225 -
accuracy: 0.5692 - val_loss: 1.0970 - val_accuracy: 0.5895
```

```
Epoch 14/250
5564/5564 [==============================] - 0s 33us/step - loss: 1.0899 -
accuracy: 0.5767 - val_loss: 1.0731 - val_accuracy: 0.5866
Epoch 15/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.0720 -
accuracy: 0.5848 - val_loss: 1.0518 - val_accuracy: 0.6053
Epoch 16/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.0457 -
accuracy: 0.6037 - val_loss: 1.0312 - val_accuracy: 0.6183
Epoch 17/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.0295 -
accuracy: 0.6055 - val_loss: 1.0116 - val_accuracy: 0.6139
Epoch 18/250
5564/5564 [==============================] - 0s 34us/step - loss: 1.0114 -
accuracy: 0.6107 - val_loss: 1.0007 - val_accuracy: 0.6362
Epoch 19/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.9913 -
accuracy: 0.6183 - val_loss: 0.9779 - val_accuracy: 0.6506
Epoch 20/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.9787 -
accuracy: 0.6271 - val_loss: 0.9637 - val_accuracy: 0.6592
Epoch 21/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.9688 -
accuracy: 0.6228 - val_loss: 0.9578 - val_accuracy: 0.6506
Epoch 22/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.9519 -
accuracy: 0.6382 - val_loss: 0.9390 - val_accuracy: 0.6535
Epoch 23/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.9354 -
accuracy: 0.6449 - val_loss: 0.9339 - val_accuracy: 0.6506
Epoch 24/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.9211 -
accuracy: 0.6542 - val_loss: 0.9178 - val_accuracy: 0.6736
Epoch 25/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.9120 -
accuracy: 0.6479 - val_loss: 0.9063 - val_accuracy: 0.6758
Epoch 26/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8995 -
accuracy: 0.6587 - val_loss: 0.8945 - val_accuracy: 0.6729
Epoch 27/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8872 -
accuracy: 0.6598 - val_loss: 0.8864 - val_accuracy: 0.6772
Epoch 28/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8807 -
accuracy: 0.6569 - val_loss: 0.8730 - val_accuracy: 0.6837
Epoch 29/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.8713 -
accuracy: 0.6673 - val_loss: 0.8646 - val_accuracy: 0.6930
```

```
Epoch 30/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.8579 -
accuracy: 0.6725 - val_loss: 0.8583 - val_accuracy: 0.6909
Epoch 31/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8475 -
accuracy: 0.6756 - val_loss: 0.8481 - val_accuracy: 0.6981
Epoch 32/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.8434 -
accuracy: 0.6720 - val_loss: 0.8394 - val_accuracy: 0.6930
Epoch 33/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8311 -
accuracy: 0.6803 - val_loss: 0.8295 - val_accuracy: 0.7038
Epoch 34/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.8207 -
accuracy: 0.6867 - val_loss: 0.8229 - val_accuracy: 0.7067
Epoch 35/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8099 -
accuracy: 0.6882 - val_loss: 0.8173 - val_accuracy: 0.7110
Epoch 36/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.8122 -
accuracy: 0.6862 - val_loss: 0.8058 - val_accuracy: 0.7081
Epoch 37/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7965 -
accuracy: 0.6950 - val_loss: 0.8089 - val_accuracy: 0.7017
Epoch 38/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7914 -
accuracy: 0.6970 - val_loss: 0.7918 - val_accuracy: 0.7139
Epoch 39/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7822 -
accuracy: 0.7081 - val_loss: 0.7864 - val_accuracy: 0.7182
Epoch 40/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7769 -
accuracy: 0.7056 - val_loss: 0.7843 - val_accuracy: 0.7088
Epoch 41/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7721 -
accuracy: 0.6999 - val_loss: 0.7710 - val_accuracy: 0.7175
Epoch 42/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7622 -
accuracy: 0.7069 - val_loss: 0.7684 - val_accuracy: 0.7124
Epoch 43/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7571 -
accuracy: 0.7074 - val_loss: 0.7629 - val_accuracy: 0.7175
Epoch 44/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7492 -
accuracy: 0.7162 - val_loss: 0.7556 - val_accuracy: 0.7203
Epoch 45/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7452 -
accuracy: 0.7078 - val_loss: 0.7480 - val_accuracy: 0.7211
```

```
Epoch 46/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7341 -
accuracy: 0.7220 - val_loss: 0.7436 - val_accuracy: 0.7362
Epoch 47/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.7347 -
accuracy: 0.7092 - val_loss: 0.7349 - val_accuracy: 0.7239
Epoch 48/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7250 -
accuracy: 0.7178 - val_loss: 0.7366 - val_accuracy: 0.7290
Epoch 49/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7184 -
accuracy: 0.7227 - val_loss: 0.7302 - val_accuracy: 0.7275
Epoch 50/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7127 -
accuracy: 0.7266 - val_loss: 0.7212 - val_accuracy: 0.7369
Epoch 51/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.7081 -
accuracy: 0.7261 - val_loss: 0.7201 - val_accuracy: 0.7268
Epoch 52/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6995 -
accuracy: 0.7324 - val_loss: 0.7114 - val_accuracy: 0.7398
Epoch 53/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6958 -
accuracy: 0.7333 - val_loss: 0.7069 - val_accuracy: 0.7383
Epoch 54/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6906 -
accuracy: 0.7374 - val_loss: 0.7078 - val_accuracy: 0.7419
Epoch 55/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6860 -
accuracy: 0.7342 - val_loss: 0.7057 - val_accuracy: 0.7469
Epoch 56/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6846 -
accuracy: 0.7347 - val_loss: 0.6968 - val_accuracy: 0.7405
Epoch 57/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6777 -
accuracy: 0.7358 - val_loss: 0.6911 - val_accuracy: 0.7383
Epoch 58/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6732 -
accuracy: 0.7407 - val_loss: 0.6954 - val_accuracy: 0.7455
Epoch 59/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6715 -
accuracy: 0.7412 - val_loss: 0.6878 - val_accuracy: 0.7462
Epoch 60/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6636 -
accuracy: 0.7453 - val_loss: 0.6827 - val_accuracy: 0.7570
Epoch 61/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6603 -
accuracy: 0.7507 - val_loss: 0.6745 - val_accuracy: 0.7556
```

```
Epoch 62/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6554 -
accuracy: 0.7473 - val_loss: 0.6691 - val_accuracy: 0.7527
Epoch 63/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6507 -
accuracy: 0.7504 - val_loss: 0.6641 - val_accuracy: 0.7491
Epoch 64/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6406 -
accuracy: 0.7541 - val_loss: 0.6612 - val_accuracy: 0.7606
Epoch 65/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6412 -
accuracy: 0.7514 - val_loss: 0.6665 - val_accuracy: 0.7534
Epoch 66/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6376 -
accuracy: 0.7563 - val_loss: 0.6565 - val_accuracy: 0.7505
Epoch 67/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6325 -
accuracy: 0.7570 - val_loss: 0.6503 - val_accuracy: 0.7584
Epoch 68/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6269 -
accuracy: 0.7593 - val_loss: 0.6483 - val_accuracy: 0.7714
Epoch 69/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6257 -
accuracy: 0.7586 - val_loss: 0.6455 - val_accuracy: 0.7642
Epoch 70/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6233 -
accuracy: 0.7658 - val_loss: 0.6447 - val_accuracy: 0.7649
Epoch 71/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6133 -
accuracy: 0.7699 - val_loss: 0.6512 - val_accuracy: 0.7606
Epoch 72/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6139 -
accuracy: 0.7572 - val_loss: 0.6454 - val_accuracy: 0.7592
Epoch 73/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6143 -
accuracy: 0.7705 - val_loss: 0.6351 - val_accuracy: 0.7628
Epoch 74/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.6084 -
accuracy: 0.7692 - val_loss: 0.6315 - val_accuracy: 0.7656
Epoch 75/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.6058 -
accuracy: 0.7694 - val_loss: 0.6310 - val_accuracy: 0.7757
Epoch 76/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5995 -
accuracy: 0.7685 - val_loss: 0.6234 - val_accuracy: 0.7620
Epoch 77/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5992 -
accuracy: 0.7716 - val_loss: 0.6157 - val_accuracy: 0.7671
```

```
Epoch 78/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5930 -
accuracy: 0.7741 - val_loss: 0.6176 - val_accuracy: 0.7699
Epoch 79/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5890 -
accuracy: 0.7773 - val_loss: 0.6210 - val_accuracy: 0.7764
Epoch 80/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.5853 -
accuracy: 0.7791 - val_loss: 0.6048 - val_accuracy: 0.7707
Epoch 81/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5855 -
accuracy: 0.7719 - val_loss: 0.6105 - val_accuracy: 0.7714
Epoch 82/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5838 -
accuracy: 0.7793 - val_loss: 0.6137 - val_accuracy: 0.7764
Epoch 83/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5802 -
accuracy: 0.7777 - val_loss: 0.6140 - val_accuracy: 0.7735
Epoch 84/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5753 -
accuracy: 0.7768 - val_loss: 0.6024 - val_accuracy: 0.7807
Epoch 85/250
5564/5564 [==============================] - 0s 35us/step - loss: 0.5723 -
accuracy: 0.7832 - val_loss: 0.5943 - val_accuracy: 0.7807
Epoch 86/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5671 -
accuracy: 0.7895 - val_loss: 0.5928 - val_accuracy: 0.7800
Epoch 87/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5665 -
accuracy: 0.7818 - val_loss: 0.5896 - val_accuracy: 0.7757
Epoch 88/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5606 -
accuracy: 0.7834 - val_loss: 0.5833 - val_accuracy: 0.7800
Epoch 89/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5594 -
accuracy: 0.7818 - val_loss: 0.5823 - val_accuracy: 0.7829
Epoch 90/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5582 -
accuracy: 0.7879 - val_loss: 0.5801 - val_accuracy: 0.7872
Epoch 91/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.5528 -
accuracy: 0.7948 - val_loss: 0.5795 - val_accuracy: 0.7865
Epoch 92/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5518 -
accuracy: 0.7885 - val_loss: 0.5841 - val_accuracy: 0.7886
Epoch 93/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5439 -
accuracy: 0.7949 - val_loss: 0.5772 - val_accuracy: 0.7901
```

```
Epoch 94/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5451 -
accuracy: 0.7940 - val_loss: 0.5700 - val_accuracy: 0.7858
Epoch 95/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.5431 -
accuracy: 0.7969 - val_loss: 0.5673 - val_accuracy: 0.7879
Epoch 96/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5384 -
accuracy: 0.7928 - val_loss: 0.5731 - val_accuracy: 0.7951
Epoch 97/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5374 -
accuracy: 0.7922 - val_loss: 0.5649 - val_accuracy: 0.7944
Epoch 98/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5333 -
accuracy: 0.7946 - val_loss: 0.5636 - val_accuracy: 0.7908
Epoch 99/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5268 -
accuracy: 0.8000 - val_loss: 0.5646 - val_accuracy: 0.7915
Epoch 100/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5283 -
accuracy: 0.8014 - val_loss: 0.5563 - val_accuracy: 0.7937
Epoch 101/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5231 -
accuracy: 0.8059 - val_loss: 0.5582 - val_accuracy: 0.7944
Epoch 102/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5210 -
accuracy: 0.8019 - val_loss: 0.5580 - val_accuracy: 0.7908
Epoch 103/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5162 -
accuracy: 0.8028 - val_loss: 0.5548 - val_accuracy: 0.8001
Epoch 104/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5186 -
accuracy: 0.8027 - val_loss: 0.5523 - val_accuracy: 0.7894
Epoch 105/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5112 -
accuracy: 0.8059 - val_loss: 0.5629 - val_accuracy: 0.7829
Epoch 106/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5166 -
accuracy: 0.8045 - val_loss: 0.5488 - val_accuracy: 0.7973
Epoch 107/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5076 -
accuracy: 0.8109 - val_loss: 0.5411 - val_accuracy: 0.8016
Epoch 108/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5114 -
accuracy: 0.8073 - val_loss: 0.5379 - val_accuracy: 0.7973
Epoch 109/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5079 -
accuracy: 0.8107 - val_loss: 0.5394 - val_accuracy: 0.8001
```

```
Epoch 110/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.5031 -
accuracy: 0.8019 - val_loss: 0.5387 - val_accuracy: 0.8009
Epoch 111/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.5028 -
accuracy: 0.8086 - val_loss: 0.5410 - val_accuracy: 0.7944
Epoch 112/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4988 -
accuracy: 0.8127 - val_loss: 0.5375 - val_accuracy: 0.8009
Epoch 113/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4971 -
accuracy: 0.8063 - val_loss: 0.5304 - val_accuracy: 0.8052
Epoch 114/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4943 -
accuracy: 0.8183 - val_loss: 0.5305 - val_accuracy: 0.7958
Epoch 115/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4912 -
accuracy: 0.8178 - val_loss: 0.5266 - val_accuracy: 0.8023
Epoch 116/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4859 -
accuracy: 0.8205 - val_loss: 0.5465 - val_accuracy: 0.7987
Epoch 117/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4886 -
accuracy: 0.8163 - val_loss: 0.5347 - val_accuracy: 0.7886
Epoch 118/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4875 -
accuracy: 0.8156 - val_loss: 0.5218 - val_accuracy: 0.8102
Epoch 119/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4881 -
accuracy: 0.8134 - val_loss: 0.5201 - val_accuracy: 0.8073
Epoch 120/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4806 -
accuracy: 0.8149 - val_loss: 0.5141 - val_accuracy: 0.8081
Epoch 121/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4812 -
accuracy: 0.8116 - val_loss: 0.5206 - val_accuracy: 0.8066
Epoch 122/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4761 -
accuracy: 0.8203 - val_loss: 0.5109 - val_accuracy: 0.8052
Epoch 123/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4730 -
accuracy: 0.8248 - val_loss: 0.5097 - val_accuracy: 0.8081
Epoch 124/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4669 -
accuracy: 0.8242 - val_loss: 0.5258 - val_accuracy: 0.7987
Epoch 125/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4756 -
accuracy: 0.8161 - val_loss: 0.5158 - val_accuracy: 0.8066
```

```
Epoch 126/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4694 -
accuracy: 0.8237 - val_loss: 0.5038 - val_accuracy: 0.8109
Epoch 127/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4665 -
accuracy: 0.8260 - val_loss: 0.5070 - val_accuracy: 0.8116
Epoch 128/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4676 -
accuracy: 0.8231 - val_loss: 0.5022 - val_accuracy: 0.8152
Epoch 129/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4631 -
accuracy: 0.8269 - val_loss: 0.5076 - val_accuracy: 0.8045
Epoch 130/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4615 -
accuracy: 0.8273 - val_loss: 0.4961 - val_accuracy: 0.8145
Epoch 131/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4622 -
accuracy: 0.8266 - val_loss: 0.4992 - val_accuracy: 0.8152
Epoch 132/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4587 -
accuracy: 0.8280 - val_loss: 0.4963 - val_accuracy: 0.8131
Epoch 133/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4557 -
accuracy: 0.8267 - val_loss: 0.4929 - val_accuracy: 0.8102
Epoch 134/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.4551 -
accuracy: 0.8296 - val_loss: 0.4907 - val_accuracy: 0.8116
Epoch 135/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4553 -
accuracy: 0.8302 - val_loss: 0.5003 - val_accuracy: 0.8088
Epoch 136/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4521 -
accuracy: 0.8298 - val_loss: 0.4894 - val_accuracy: 0.8196
Epoch 137/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4514 -
accuracy: 0.8280 - val_loss: 0.4844 - val_accuracy: 0.8217
Epoch 138/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4496 -
accuracy: 0.8359 - val_loss: 0.4847 - val_accuracy: 0.8181
Epoch 139/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4457 -
accuracy: 0.8314 - val_loss: 0.4821 - val_accuracy: 0.8160
Epoch 140/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4443 -
accuracy: 0.8330 - val_loss: 0.4870 - val_accuracy: 0.8167
Epoch 141/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4448 -
accuracy: 0.8302 - val_loss: 0.4788 - val_accuracy: 0.8181
```

```
Epoch 142/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4365 -
accuracy: 0.8330 - val_loss: 0.4796 - val_accuracy: 0.8188
Epoch 143/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4409 -
accuracy: 0.8343 - val_loss: 0.4825 - val_accuracy: 0.8181
Epoch 144/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4397 -
accuracy: 0.8375 - val_loss: 0.4789 - val_accuracy: 0.8152
Epoch 145/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4353 -
accuracy: 0.8372 - val_loss: 0.4841 - val_accuracy: 0.8174
Epoch 146/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4355 -
accuracy: 0.8363 - val_loss: 0.4775 - val_accuracy: 0.8217
Epoch 147/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4298 -
accuracy: 0.8397 - val_loss: 0.4809 - val_accuracy: 0.8210
Epoch 148/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4326 -
accuracy: 0.8382 - val_loss: 0.4720 - val_accuracy: 0.8253
Epoch 149/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4284 -
accuracy: 0.8382 - val_loss: 0.4727 - val_accuracy: 0.8231
Epoch 150/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4272 -
accuracy: 0.8397 - val_loss: 0.4682 - val_accuracy: 0.8275
Epoch 151/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4208 -
accuracy: 0.8420 - val_loss: 0.4738 - val_accuracy: 0.8246
Epoch 152/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4260 -
accuracy: 0.8397 - val_loss: 0.4657 - val_accuracy: 0.8253
Epoch 153/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4253 -
accuracy: 0.8388 - val_loss: 0.4673 - val_accuracy: 0.8275
Epoch 154/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4209 -
accuracy: 0.8445 - val_loss: 0.4644 - val_accuracy: 0.8296
Epoch 155/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4194 -
accuracy: 0.8436 - val_loss: 0.4637 - val_accuracy: 0.8210
Epoch 156/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4200 -
accuracy: 0.8400 - val_loss: 0.4634 - val_accuracy: 0.8282
Epoch 157/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4183 -
accuracy: 0.8442 - val_loss: 0.4789 - val_accuracy: 0.8160
```

```
Epoch 158/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4166 -
accuracy: 0.8442 - val_loss: 0.4574 - val_accuracy: 0.8275
Epoch 159/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4137 -
accuracy: 0.8506 - val_loss: 0.4542 - val_accuracy: 0.8296
Epoch 160/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4100 -
accuracy: 0.8490 - val_loss: 0.4600 - val_accuracy: 0.8289
Epoch 161/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4101 -
accuracy: 0.8480 - val_loss: 0.4562 - val_accuracy: 0.8303
Epoch 162/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4080 -
accuracy: 0.8456 - val_loss: 0.4534 - val_accuracy: 0.8311
Epoch 163/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4043 -
accuracy: 0.8496 - val_loss: 0.4545 - val_accuracy: 0.8296
Epoch 164/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4017 -
accuracy: 0.8577 - val_loss: 0.4580 - val_accuracy: 0.8282
Epoch 165/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4074 -
accuracy: 0.8456 - val_loss: 0.4457 - val_accuracy: 0.8318
Epoch 166/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.4056 -
accuracy: 0.8478 - val_loss: 0.4494 - val_accuracy: 0.8282
Epoch 167/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3995 -
accuracy: 0.8515 - val_loss: 0.4507 - val_accuracy: 0.8325
Epoch 168/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3965 -
accuracy: 0.8557 - val_loss: 0.4488 - val_accuracy: 0.8289
Epoch 169/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3982 -
accuracy: 0.8530 - val_loss: 0.4392 - val_accuracy: 0.8397
Epoch 170/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3900 -
accuracy: 0.8568 - val_loss: 0.4440 - val_accuracy: 0.8382
Epoch 171/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3934 -
accuracy: 0.8623 - val_loss: 0.4437 - val_accuracy: 0.8397
Epoch 172/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3928 -
accuracy: 0.8559 - val_loss: 0.4422 - val_accuracy: 0.8339
Epoch 173/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3981 -
accuracy: 0.8460 - val_loss: 0.4343 - val_accuracy: 0.8411
```

```
Epoch 174/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3910 -
accuracy: 0.8613 - val_loss: 0.4437 - val_accuracy: 0.8411
Epoch 175/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3897 -
accuracy: 0.8587 - val_loss: 0.4411 - val_accuracy: 0.8390
Epoch 176/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3851 -
accuracy: 0.8623 - val_loss: 0.4425 - val_accuracy: 0.8404
Epoch 177/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3859 -
accuracy: 0.8580 - val_loss: 0.4348 - val_accuracy: 0.8361
Epoch 178/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3865 -
accuracy: 0.8602 - val_loss: 0.4341 - val_accuracy: 0.8418
Epoch 179/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3845 -
accuracy: 0.8593 - val_loss: 0.4255 - val_accuracy: 0.8440
Epoch 180/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3836 -
accuracy: 0.8595 - val_loss: 0.4314 - val_accuracy: 0.8404
Epoch 181/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3811 -
accuracy: 0.8602 - val_loss: 0.4335 - val_accuracy: 0.8404
Epoch 182/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3811 -
accuracy: 0.8573 - val_loss: 0.4242 - val_accuracy: 0.8440
Epoch 183/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3787 -
accuracy: 0.8629 - val_loss: 0.4348 - val_accuracy: 0.8433
Epoch 184/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3731 -
accuracy: 0.8623 - val_loss: 0.4238 - val_accuracy: 0.8462
Epoch 185/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3731 -
accuracy: 0.8656 - val_loss: 0.4276 - val_accuracy: 0.8440
Epoch 186/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3746 -
accuracy: 0.8648 - val_loss: 0.4213 - val_accuracy: 0.8447
Epoch 187/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3731 -
accuracy: 0.8665 - val_loss: 0.4239 - val_accuracy: 0.8476
Epoch 188/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3708 -
accuracy: 0.8656 - val_loss: 0.4330 - val_accuracy: 0.8440
Epoch 189/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3703 -
accuracy: 0.8625 - val_loss: 0.4226 - val_accuracy: 0.8476
```

```
Epoch 190/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3719 -
accuracy: 0.8643 - val_loss: 0.4163 - val_accuracy: 0.8497
Epoch 191/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3680 -
accuracy: 0.8663 - val_loss: 0.4352 - val_accuracy: 0.8433
Epoch 192/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3670 -
accuracy: 0.8661 - val_loss: 0.4168 - val_accuracy: 0.8512
Epoch 193/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3664 -
accuracy: 0.8699 - val_loss: 0.4119 - val_accuracy: 0.8505
Epoch 194/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3608 -
accuracy: 0.8681 - val_loss: 0.4113 - val_accuracy: 0.8541
Epoch 195/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3603 -
accuracy: 0.8726 - val_loss: 0.4084 - val_accuracy: 0.8526
Epoch 196/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3624 -
accuracy: 0.8670 - val_loss: 0.4137 - val_accuracy: 0.8519
Epoch 197/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3575 -
accuracy: 0.8688 - val_loss: 0.4186 - val_accuracy: 0.8505
Epoch 198/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3583 -
accuracy: 0.8742 - val_loss: 0.4094 - val_accuracy: 0.8541
Epoch 199/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3563 -
accuracy: 0.8733 - val_loss: 0.4085 - val_accuracy: 0.8519
Epoch 200/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3517 -
accuracy: 0.8726 - val_loss: 0.4073 - val_accuracy: 0.8533
Epoch 201/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3560 -
accuracy: 0.8724 - val_loss: 0.4288 - val_accuracy: 0.8411
Epoch 202/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3544 -
accuracy: 0.8733 - val_loss: 0.4079 - val_accuracy: 0.8497
Epoch 203/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3533 -
accuracy: 0.8740 - val_loss: 0.4026 - val_accuracy: 0.8584
Epoch 204/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3471 -
accuracy: 0.8762 - val_loss: 0.4042 - val_accuracy: 0.8548
Epoch 205/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3492 -
accuracy: 0.8747 - val_loss: 0.4034 - val_accuracy: 0.8541
```

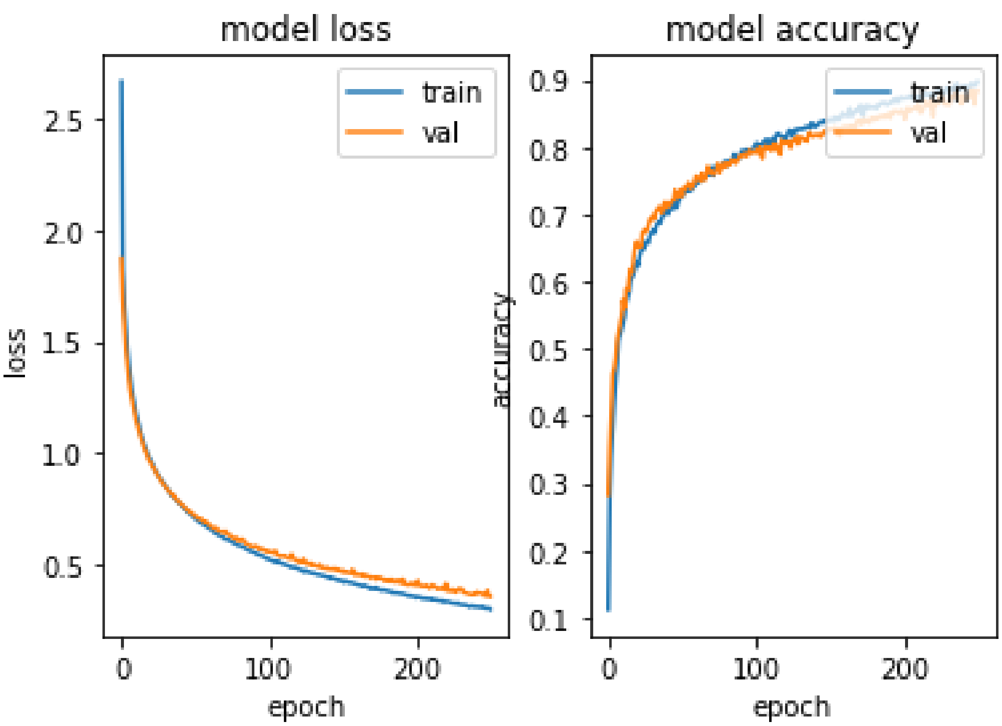
```
Epoch 206/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3512 -
accuracy: 0.8717 - val_loss: 0.4007 - val_accuracy: 0.8562
Epoch 207/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3458 -
accuracy: 0.8753 - val_loss: 0.4081 - val_accuracy: 0.8577
Epoch 208/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3477 -
accuracy: 0.8722 - val_loss: 0.4006 - val_accuracy: 0.8591
Epoch 209/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3456 -
accuracy: 0.8760 - val_loss: 0.3941 - val_accuracy: 0.8584
Epoch 210/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3488 -
accuracy: 0.8731 - val_loss: 0.3997 - val_accuracy: 0.8634
Epoch 211/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3412 -
accuracy: 0.8754 - val_loss: 0.4102 - val_accuracy: 0.8519
Epoch 212/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3424 -
accuracy: 0.8724 - val_loss: 0.4038 - val_accuracy: 0.8512
Epoch 213/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3442 -
accuracy: 0.8728 - val_loss: 0.4009 - val_accuracy: 0.8562
Epoch 214/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3393 -
accuracy: 0.8801 - val_loss: 0.3920 - val_accuracy: 0.8605
Epoch 215/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3404 -
accuracy: 0.8772 - val_loss: 0.3899 - val_accuracy: 0.8684
Epoch 216/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3375 -
accuracy: 0.8776 - val_loss: 0.3930 - val_accuracy: 0.8641
Epoch 217/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3358 -
accuracy: 0.8789 - val_loss: 0.3952 - val_accuracy: 0.8634
Epoch 218/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3394 -
accuracy: 0.8756 - val_loss: 0.3997 - val_accuracy: 0.8598
Epoch 219/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3318 -
accuracy: 0.8799 - val_loss: 0.3866 - val_accuracy: 0.8699
Epoch 220/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3327 -
accuracy: 0.8790 - val_loss: 0.4151 - val_accuracy: 0.8454
Epoch 221/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3346 -
accuracy: 0.8767 - val_loss: 0.3856 - val_accuracy: 0.8670
```

```
Epoch 222/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3330 -
accuracy: 0.8817 - val_loss: 0.3823 - val_accuracy: 0.8620
Epoch 223/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3303 -
accuracy: 0.8817 - val_loss: 0.3818 - val_accuracy: 0.8656
Epoch 224/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3275 -
accuracy: 0.8868 - val_loss: 0.3887 - val_accuracy: 0.8656
Epoch 225/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3279 -
accuracy: 0.8855 - val_loss: 0.3858 - val_accuracy: 0.8756
Epoch 226/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3271 -
accuracy: 0.8780 - val_loss: 0.3778 - val_accuracy: 0.8684
Epoch 227/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3242 -
accuracy: 0.8834 - val_loss: 0.3766 - val_accuracy: 0.8728
Epoch 228/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3281 -
accuracy: 0.8830 - val_loss: 0.3883 - val_accuracy: 0.8634
Epoch 229/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3234 -
accuracy: 0.8880 - val_loss: 0.3930 - val_accuracy: 0.8634
Epoch 230/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3210 -
accuracy: 0.8823 - val_loss: 0.3870 - val_accuracy: 0.8641
Epoch 231/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3199 -
accuracy: 0.8868 - val_loss: 0.3897 - val_accuracy: 0.8627
Epoch 232/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3205 -
accuracy: 0.8859 - val_loss: 0.3757 - val_accuracy: 0.8706
Epoch 233/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3179 -
accuracy: 0.8884 - val_loss: 0.3743 - val_accuracy: 0.8756
Epoch 234/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3185 -
accuracy: 0.8834 - val_loss: 0.3716 - val_accuracy: 0.8742
Epoch 235/250
5564/5564 [==============================] - 0s 34us/step - loss: 0.3148 -
accuracy: 0.8880 - val_loss: 0.3698 - val_accuracy: 0.8778
Epoch 236/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3104 -
accuracy: 0.8949 - val_loss: 0.3677 - val_accuracy: 0.8807
Epoch 237/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3128 -
accuracy: 0.8884 - val_loss: 0.3660 - val_accuracy: 0.8749
```

```
Epoch 238/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3160 -
accuracy: 0.8843 - val_loss: 0.3652 - val_accuracy: 0.8763
Epoch 239/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3122 -
accuracy: 0.8891 - val_loss: 0.3715 - val_accuracy: 0.8641
Epoch 240/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3099 -
accuracy: 0.8882 - val_loss: 0.3680 - val_accuracy: 0.8771
Epoch 241/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3115 -
accuracy: 0.8905 - val_loss: 0.3767 - val_accuracy: 0.8735
Epoch 242/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3126 -
accuracy: 0.8879 - val_loss: 0.3730 - val_accuracy: 0.8699
Epoch 243/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3089 -
accuracy: 0.8896 - val_loss: 0.3727 - val_accuracy: 0.8720
Epoch 244/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3081 -
accuracy: 0.8898 - val_loss: 0.3725 - val_accuracy: 0.8742
Epoch 245/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3126 -
accuracy: 0.8859 - val_loss: 0.3671 - val_accuracy: 0.8742
Epoch 246/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3069 -
accuracy: 0.8950 - val_loss: 0.3597 - val_accuracy: 0.8850
Epoch 247/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3030 -
accuracy: 0.8929 - val_loss: 0.3633 - val_accuracy: 0.8792
Epoch 248/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3041 -
accuracy: 0.8909 - val_loss: 0.3835 - val_accuracy: 0.8634
Epoch 249/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.3064 -
accuracy: 0.8925 - val_loss: 0.3565 - val_accuracy: 0.8828
Epoch 250/250
5564/5564 [==============================] - 0s 33us/step - loss: 0.2983 -
accuracy: 0.8972 - val_loss: 0.3563 - val_accuracy: 0.8835
```

[89]:
```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')
```

```python
ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```



[90]:
```python
score = model.evaluate(x=x_testcnn, y=y_test)
predictions = model.predict(x_testcnn)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
535/535 [==============================] - 0s 153us/step
Test loss: 0.3051721656990943
Test accuracy: 0.9121495485305786
```

[92]:
```python
print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support
```

```
           A       0.98      0.94      0.96       127
           E       0.97      0.72      0.82        81
           F       0.96      0.96      0.96        46
           L       0.87      0.97      0.92        69
           N       0.94      0.89      0.91        71
           T       0.91      1.00      0.95        62
           W       0.78      0.95      0.86        79

    accuracy                           0.91       535
   macro avg       0.92      0.92      0.91       535
weighted avg       0.92      0.91      0.91       535
```

6.0.11 2. standard scale:

[93]:
```python
x_traincnn.shape
```

[93]: (5564, 40, 1)

[94]:
```python
scaler = StandardScaler()
scaler.fit(x_traincnn.reshape(x_traincnn.shape[0], x_traincnn.shape[1]))

x_traincnn_scaled = scaler.transform(x_traincnn.reshape(x_traincnn.shape[0],␣
↪x_traincnn.shape[1]))
x_traincnn_scaled = np.expand_dims(x_traincnn_scaled, 2)

x_valcnn_scaled = scaler.transform(x_valcnn.reshape(x_valcnn.shape[0], x_valcnn.
↪shape[1]))
x_valcnn_scaled = np.expand_dims(x_valcnn_scaled, 2)

x_testcnn_scaled = scaler.transform(x_testcnn.reshape(x_testcnn.shape[0],␣
↪x_testcnn.shape[1]))
x_testcnn_scaled = np.expand_dims(x_testcnn_scaled, 2)
```

[101]:
```python
history = model.fit(x_traincnn_scaled, y_train, batch_size=512, epochs=350,␣
↪validation_data=(x_valcnn_scaled, y_val))
```

```
Train on 5564 samples, validate on 1391 samples
Epoch 1/350
5564/5564 [==============================] - 0s 73us/step - loss: 1.9318 -
accuracy: 0.2757 - val_loss: 1.9271 - val_accuracy: 0.3134
Epoch 2/350
5564/5564 [==============================] - 0s 26us/step - loss: 1.9199 -
accuracy: 0.3136 - val_loss: 1.9176 - val_accuracy: 0.3048
Epoch 3/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.9101 -
```

```
accuracy: 0.2948 - val_loss: 1.9089 - val_accuracy: 0.2904
Epoch 4/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.9007 -
accuracy: 0.2870 - val_loss: 1.8999 - val_accuracy: 0.2818
Epoch 5/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8909 -
accuracy: 0.2840 - val_loss: 1.8909 - val_accuracy: 0.2782
Epoch 6/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8811 -
accuracy: 0.2859 - val_loss: 1.8816 - val_accuracy: 0.2797
Epoch 7/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8710 -
accuracy: 0.2858 - val_loss: 1.8721 - val_accuracy: 0.2818
Epoch 8/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8606 -
accuracy: 0.2933 - val_loss: 1.8622 - val_accuracy: 0.2832
Epoch 9/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8499 -
accuracy: 0.2965 - val_loss: 1.8520 - val_accuracy: 0.2912
Epoch 10/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8388 -
accuracy: 0.3034 - val_loss: 1.8413 - val_accuracy: 0.2948
Epoch 11/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8271 -
accuracy: 0.3070 - val_loss: 1.8302 - val_accuracy: 0.3019
Epoch 12/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8150 -
accuracy: 0.3120 - val_loss: 1.8186 - val_accuracy: 0.3084
Epoch 13/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.8026 -
accuracy: 0.3170 - val_loss: 1.8067 - val_accuracy: 0.3106
Epoch 14/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7897 -
accuracy: 0.3237 - val_loss: 1.7943 - val_accuracy: 0.3185
Epoch 15/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7761 -
accuracy: 0.3303 - val_loss: 1.7816 - val_accuracy: 0.3300
Epoch 16/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7625 -
accuracy: 0.3370 - val_loss: 1.7685 - val_accuracy: 0.3422
Epoch 17/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7486 -
accuracy: 0.3445 - val_loss: 1.7551 - val_accuracy: 0.3480
Epoch 18/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7343 -
accuracy: 0.3503 - val_loss: 1.7413 - val_accuracy: 0.3566
Epoch 19/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7200 -
```

```
accuracy: 0.3562 - val_loss: 1.7272 - val_accuracy: 0.3616
Epoch 20/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.7050 -
accuracy: 0.3621 - val_loss: 1.7132 - val_accuracy: 0.3681
Epoch 21/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.6899 -
accuracy: 0.3684 - val_loss: 1.6989 - val_accuracy: 0.3688
Epoch 22/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.6749 -
accuracy: 0.3719 - val_loss: 1.6842 - val_accuracy: 0.3717
Epoch 23/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.6591 -
accuracy: 0.3774 - val_loss: 1.6694 - val_accuracy: 0.3789
Epoch 24/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.6440 -
accuracy: 0.3819 - val_loss: 1.6546 - val_accuracy: 0.3846
Epoch 25/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.6282 -
accuracy: 0.3862 - val_loss: 1.6394 - val_accuracy: 0.3904
Epoch 26/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.6123 -
accuracy: 0.3934 - val_loss: 1.6243 - val_accuracy: 0.3947
Epoch 27/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5964 -
accuracy: 0.4003 - val_loss: 1.6091 - val_accuracy: 0.3968
Epoch 28/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5806 -
accuracy: 0.4062 - val_loss: 1.5937 - val_accuracy: 0.4055
Epoch 29/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5646 -
accuracy: 0.4082 - val_loss: 1.5783 - val_accuracy: 0.4112
Epoch 30/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5482 -
accuracy: 0.4175 - val_loss: 1.5628 - val_accuracy: 0.4148
Epoch 31/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5325 -
accuracy: 0.4245 - val_loss: 1.5473 - val_accuracy: 0.4198
Epoch 32/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5166 -
accuracy: 0.4297 - val_loss: 1.5321 - val_accuracy: 0.4270
Epoch 33/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.5009 -
accuracy: 0.4357 - val_loss: 1.5169 - val_accuracy: 0.4364
Epoch 34/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.4852 -
accuracy: 0.4416 - val_loss: 1.5017 - val_accuracy: 0.4421
Epoch 35/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.4701 -
```

```
accuracy: 0.4488 - val_loss: 1.4869 - val_accuracy: 0.4428
Epoch 36/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.4547 -
accuracy: 0.4524 - val_loss: 1.4723 - val_accuracy: 0.4428
Epoch 37/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.4399 -
accuracy: 0.4608 - val_loss: 1.4581 - val_accuracy: 0.4479
Epoch 38/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.4253 -
accuracy: 0.4675 - val_loss: 1.4441 - val_accuracy: 0.4515
Epoch 39/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.4115 -
accuracy: 0.4705 - val_loss: 1.4304 - val_accuracy: 0.4522
Epoch 40/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3976 -
accuracy: 0.4736 - val_loss: 1.4170 - val_accuracy: 0.4522
Epoch 41/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3836 -
accuracy: 0.4783 - val_loss: 1.4040 - val_accuracy: 0.4558
Epoch 42/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3707 -
accuracy: 0.4813 - val_loss: 1.3912 - val_accuracy: 0.4615
Epoch 43/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3584 -
accuracy: 0.4833 - val_loss: 1.3792 - val_accuracy: 0.4694
Epoch 44/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3458 -
accuracy: 0.4874 - val_loss: 1.3672 - val_accuracy: 0.4709
Epoch 45/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3337 -
accuracy: 0.4889 - val_loss: 1.3551 - val_accuracy: 0.4774
Epoch 46/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3219 -
accuracy: 0.4934 - val_loss: 1.3436 - val_accuracy: 0.4752
Epoch 47/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.3108 -
accuracy: 0.4953 - val_loss: 1.3324 - val_accuracy: 0.4824
Epoch 48/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2999 -
accuracy: 0.4986 - val_loss: 1.3221 - val_accuracy: 0.4817
Epoch 49/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2900 -
accuracy: 0.4987 - val_loss: 1.3119 - val_accuracy: 0.4889
Epoch 50/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2792 -
accuracy: 0.5036 - val_loss: 1.3015 - val_accuracy: 0.4874
Epoch 51/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2697 -
```

```
accuracy: 0.5068 - val_loss: 1.2920 - val_accuracy: 0.4910
Epoch 52/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2607 -
accuracy: 0.5115 - val_loss: 1.2833 - val_accuracy: 0.4896
Epoch 53/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2516 -
accuracy: 0.5120 - val_loss: 1.2746 - val_accuracy: 0.4982
Epoch 54/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2431 -
accuracy: 0.5208 - val_loss: 1.2666 - val_accuracy: 0.4946
Epoch 55/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2350 -
accuracy: 0.5192 - val_loss: 1.2587 - val_accuracy: 0.4946
Epoch 56/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2268 -
accuracy: 0.5207 - val_loss: 1.2504 - val_accuracy: 0.5025
Epoch 57/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2195 -
accuracy: 0.5246 - val_loss: 1.2427 - val_accuracy: 0.5061
Epoch 58/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2114 -
accuracy: 0.5266 - val_loss: 1.2349 - val_accuracy: 0.5183
Epoch 59/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.2038 -
accuracy: 0.5336 - val_loss: 1.2278 - val_accuracy: 0.5212
Epoch 60/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1962 -
accuracy: 0.5347 - val_loss: 1.2207 - val_accuracy: 0.5234
Epoch 61/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1894 -
accuracy: 0.5394 - val_loss: 1.2140 - val_accuracy: 0.5248
Epoch 62/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1821 -
accuracy: 0.5397 - val_loss: 1.2071 - val_accuracy: 0.5313
Epoch 63/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1755 -
accuracy: 0.5428 - val_loss: 1.2005 - val_accuracy: 0.5341
Epoch 64/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1697 -
accuracy: 0.5483 - val_loss: 1.1949 - val_accuracy: 0.5327
Epoch 65/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1639 -
accuracy: 0.5446 - val_loss: 1.1884 - val_accuracy: 0.5392
Epoch 66/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1578 -
accuracy: 0.5491 - val_loss: 1.1824 - val_accuracy: 0.5435
Epoch 67/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1513 -
```

```
accuracy: 0.5521 - val_loss: 1.1772 - val_accuracy: 0.5421
Epoch 68/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1457 -
accuracy: 0.5550 - val_loss: 1.1717 - val_accuracy: 0.5370
Epoch 69/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1401 -
accuracy: 0.5604 - val_loss: 1.1666 - val_accuracy: 0.5356
Epoch 70/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1344 -
accuracy: 0.5588 - val_loss: 1.1610 - val_accuracy: 0.5413
Epoch 71/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1288 -
accuracy: 0.5595 - val_loss: 1.1560 - val_accuracy: 0.5492
Epoch 72/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1243 -
accuracy: 0.5609 - val_loss: 1.1512 - val_accuracy: 0.5478
Epoch 73/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1185 -
accuracy: 0.5643 - val_loss: 1.1455 - val_accuracy: 0.5586
Epoch 74/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1138 -
accuracy: 0.5667 - val_loss: 1.1414 - val_accuracy: 0.5507
Epoch 75/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1085 -
accuracy: 0.5694 - val_loss: 1.1363 - val_accuracy: 0.5543
Epoch 76/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.1035 -
accuracy: 0.5674 - val_loss: 1.1319 - val_accuracy: 0.5579
Epoch 77/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0989 -
accuracy: 0.5733 - val_loss: 1.1289 - val_accuracy: 0.5507
Epoch 78/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0942 -
accuracy: 0.5739 - val_loss: 1.1232 - val_accuracy: 0.5658
Epoch 79/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0901 -
accuracy: 0.5769 - val_loss: 1.1195 - val_accuracy: 0.5679
Epoch 80/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0855 -
accuracy: 0.5812 - val_loss: 1.1163 - val_accuracy: 0.5564
Epoch 81/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0811 -
accuracy: 0.5816 - val_loss: 1.1129 - val_accuracy: 0.5579
Epoch 82/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0773 -
accuracy: 0.5823 - val_loss: 1.1081 - val_accuracy: 0.5672
Epoch 83/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0725 -
```

```
accuracy: 0.5821 - val_loss: 1.1032 - val_accuracy: 0.5787
Epoch 84/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0686 -
accuracy: 0.5888 - val_loss: 1.0996 - val_accuracy: 0.5773
Epoch 85/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0647 -
accuracy: 0.5893 - val_loss: 1.0964 - val_accuracy: 0.5780
Epoch 86/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0603 -
accuracy: 0.5918 - val_loss: 1.0934 - val_accuracy: 0.5780
Epoch 87/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0560 -
accuracy: 0.5929 - val_loss: 1.0906 - val_accuracy: 0.5802
Epoch 88/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0524 -
accuracy: 0.5956 - val_loss: 1.0869 - val_accuracy: 0.5780
Epoch 89/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0494 -
accuracy: 0.5956 - val_loss: 1.0836 - val_accuracy: 0.5859
Epoch 90/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0453 -
accuracy: 0.5971 - val_loss: 1.0799 - val_accuracy: 0.5859
Epoch 91/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0413 -
accuracy: 0.6006 - val_loss: 1.0768 - val_accuracy: 0.5859
Epoch 92/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0385 -
accuracy: 0.6030 - val_loss: 1.0741 - val_accuracy: 0.5873
Epoch 93/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0344 -
accuracy: 0.6060 - val_loss: 1.0699 - val_accuracy: 0.5931
Epoch 94/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0310 -
accuracy: 0.6062 - val_loss: 1.0668 - val_accuracy: 0.5881
Epoch 95/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0276 -
accuracy: 0.6086 - val_loss: 1.0633 - val_accuracy: 0.5945
Epoch 96/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0239 -
accuracy: 0.6105 - val_loss: 1.0607 - val_accuracy: 0.5945
Epoch 97/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0212 -
accuracy: 0.6130 - val_loss: 1.0568 - val_accuracy: 0.5974
Epoch 98/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0166 -
accuracy: 0.6134 - val_loss: 1.0545 - val_accuracy: 0.5953
Epoch 99/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0137 -
```

```
accuracy: 0.6136 - val_loss: 1.0519 - val_accuracy: 0.5974
Epoch 100/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0101 -
accuracy: 0.6177 - val_loss: 1.0478 - val_accuracy: 0.6003
Epoch 101/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0064 -
accuracy: 0.6172 - val_loss: 1.0447 - val_accuracy: 0.6039
Epoch 102/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0040 -
accuracy: 0.6186 - val_loss: 1.0414 - val_accuracy: 0.6082
Epoch 103/350
5564/5564 [==============================] - 0s 25us/step - loss: 1.0007 -
accuracy: 0.6199 - val_loss: 1.0403 - val_accuracy: 0.6024
Epoch 104/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9980 -
accuracy: 0.6233 - val_loss: 1.0368 - val_accuracy: 0.6017
Epoch 105/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9944 -
accuracy: 0.6249 - val_loss: 1.0339 - val_accuracy: 0.6068
Epoch 106/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9917 -
accuracy: 0.6258 - val_loss: 1.0309 - val_accuracy: 0.6075
Epoch 107/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9882 -
accuracy: 0.6251 - val_loss: 1.0281 - val_accuracy: 0.6068
Epoch 108/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9853 -
accuracy: 0.6265 - val_loss: 1.0280 - val_accuracy: 0.6039
Epoch 109/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9825 -
accuracy: 0.6280 - val_loss: 1.0230 - val_accuracy: 0.6161
Epoch 110/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9791 -
accuracy: 0.6296 - val_loss: 1.0192 - val_accuracy: 0.6175
Epoch 111/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9759 -
accuracy: 0.6321 - val_loss: 1.0184 - val_accuracy: 0.6082
Epoch 112/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9733 -
accuracy: 0.6314 - val_loss: 1.0132 - val_accuracy: 0.6175
Epoch 113/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9704 -
accuracy: 0.6341 - val_loss: 1.0126 - val_accuracy: 0.6132
Epoch 114/350
5564/5564 [==============================] - 0s 26us/step - loss: 0.9674 -
accuracy: 0.6314 - val_loss: 1.0086 - val_accuracy: 0.6240
Epoch 115/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9645 -
```

```
accuracy: 0.6364 - val_loss: 1.0063 - val_accuracy: 0.6204
Epoch 116/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9621 -
accuracy: 0.6330 - val_loss: 1.0036 - val_accuracy: 0.6226
Epoch 117/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9587 -
accuracy: 0.6398 - val_loss: 1.0012 - val_accuracy: 0.6254
Epoch 118/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9558 -
accuracy: 0.6395 - val_loss: 0.9998 - val_accuracy: 0.6168
Epoch 119/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9533 -
accuracy: 0.6391 - val_loss: 0.9964 - val_accuracy: 0.6219
Epoch 120/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9504 -
accuracy: 0.6396 - val_loss: 0.9934 - val_accuracy: 0.6276
Epoch 121/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9483 -
accuracy: 0.6411 - val_loss: 0.9924 - val_accuracy: 0.6233
Epoch 122/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9455 -
accuracy: 0.6413 - val_loss: 0.9895 - val_accuracy: 0.6269
Epoch 123/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9422 -
accuracy: 0.6432 - val_loss: 0.9864 - val_accuracy: 0.6334
Epoch 124/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9396 -
accuracy: 0.6463 - val_loss: 0.9835 - val_accuracy: 0.6326
Epoch 125/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9376 -
accuracy: 0.6468 - val_loss: 0.9823 - val_accuracy: 0.6326
Epoch 126/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9347 -
accuracy: 0.6470 - val_loss: 0.9791 - val_accuracy: 0.6362
Epoch 127/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9311 -
accuracy: 0.6486 - val_loss: 0.9765 - val_accuracy: 0.6312
Epoch 128/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9302 -
accuracy: 0.6506 - val_loss: 0.9738 - val_accuracy: 0.6398
Epoch 129/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9265 -
accuracy: 0.6540 - val_loss: 0.9741 - val_accuracy: 0.6219
Epoch 130/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9241 -
accuracy: 0.6490 - val_loss: 0.9717 - val_accuracy: 0.6262
Epoch 131/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9222 -
```

```
accuracy: 0.6526 - val_loss: 0.9674 - val_accuracy: 0.6377
Epoch 132/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9189 -
accuracy: 0.6528 - val_loss: 0.9669 - val_accuracy: 0.6290
Epoch 133/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9159 -
accuracy: 0.6529 - val_loss: 0.9638 - val_accuracy: 0.6405
Epoch 134/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9140 -
accuracy: 0.6520 - val_loss: 0.9608 - val_accuracy: 0.6405
Epoch 135/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9118 -
accuracy: 0.6567 - val_loss: 0.9590 - val_accuracy: 0.6384
Epoch 136/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9094 -
accuracy: 0.6551 - val_loss: 0.9557 - val_accuracy: 0.6398
Epoch 137/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9066 -
accuracy: 0.6580 - val_loss: 0.9544 - val_accuracy: 0.6377
Epoch 138/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9035 -
accuracy: 0.6578 - val_loss: 0.9498 - val_accuracy: 0.6463
Epoch 139/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9010 -
accuracy: 0.6596 - val_loss: 0.9500 - val_accuracy: 0.6326
Epoch 140/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.9003 -
accuracy: 0.6587 - val_loss: 0.9469 - val_accuracy: 0.6485
Epoch 141/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8973 -
accuracy: 0.6589 - val_loss: 0.9444 - val_accuracy: 0.6391
Epoch 142/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8938 -
accuracy: 0.6643 - val_loss: 0.9425 - val_accuracy: 0.6477
Epoch 143/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8918 -
accuracy: 0.6600 - val_loss: 0.9415 - val_accuracy: 0.6405
Epoch 144/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8891 -
accuracy: 0.6574 - val_loss: 0.9390 - val_accuracy: 0.6499
Epoch 145/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8873 -
accuracy: 0.6614 - val_loss: 0.9364 - val_accuracy: 0.6413
Epoch 146/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8843 -
accuracy: 0.6650 - val_loss: 0.9326 - val_accuracy: 0.6542
Epoch 147/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8824 -
```

```
accuracy: 0.6637 - val_loss: 0.9305 - val_accuracy: 0.6528
Epoch 148/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8802 -
accuracy: 0.6682 - val_loss: 0.9291 - val_accuracy: 0.6556
Epoch 149/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8776 -
accuracy: 0.6659 - val_loss: 0.9258 - val_accuracy: 0.6549
Epoch 150/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8760 -
accuracy: 0.6653 - val_loss: 0.9243 - val_accuracy: 0.6564
Epoch 151/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8725 -
accuracy: 0.6706 - val_loss: 0.9227 - val_accuracy: 0.6499
Epoch 152/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8709 -
accuracy: 0.6688 - val_loss: 0.9223 - val_accuracy: 0.6463
Epoch 153/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8673 -
accuracy: 0.6693 - val_loss: 0.9161 - val_accuracy: 0.6621
Epoch 154/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8662 -
accuracy: 0.6704 - val_loss: 0.9143 - val_accuracy: 0.6578
Epoch 155/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8627 -
accuracy: 0.6715 - val_loss: 0.9157 - val_accuracy: 0.6492
Epoch 156/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8610 -
accuracy: 0.6718 - val_loss: 0.9115 - val_accuracy: 0.6564
Epoch 157/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8592 -
accuracy: 0.6718 - val_loss: 0.9093 - val_accuracy: 0.6549
Epoch 158/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8577 -
accuracy: 0.6729 - val_loss: 0.9088 - val_accuracy: 0.6578
Epoch 159/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8556 -
accuracy: 0.6760 - val_loss: 0.9064 - val_accuracy: 0.6592
Epoch 160/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8531 -
accuracy: 0.6743 - val_loss: 0.9044 - val_accuracy: 0.6556
Epoch 161/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8504 -
accuracy: 0.6763 - val_loss: 0.9016 - val_accuracy: 0.6592
Epoch 162/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8488 -
accuracy: 0.6799 - val_loss: 0.9009 - val_accuracy: 0.6564
Epoch 163/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8467 -
```

```
accuracy: 0.6790 - val_loss: 0.8989 - val_accuracy: 0.6528
Epoch 164/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8446 -
accuracy: 0.6749 - val_loss: 0.8970 - val_accuracy: 0.6556
Epoch 165/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8432 -
accuracy: 0.6788 - val_loss: 0.8947 - val_accuracy: 0.6592
Epoch 166/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8410 -
accuracy: 0.6826 - val_loss: 0.8922 - val_accuracy: 0.6621
Epoch 167/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8383 -
accuracy: 0.6828 - val_loss: 0.8899 - val_accuracy: 0.6614
Epoch 168/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8365 -
accuracy: 0.6830 - val_loss: 0.8900 - val_accuracy: 0.6607
Epoch 169/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8338 -
accuracy: 0.6842 - val_loss: 0.8865 - val_accuracy: 0.6686
Epoch 170/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8330 -
accuracy: 0.6846 - val_loss: 0.8841 - val_accuracy: 0.6686
Epoch 171/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8307 -
accuracy: 0.6862 - val_loss: 0.8839 - val_accuracy: 0.6614
Epoch 172/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8290 -
accuracy: 0.6849 - val_loss: 0.8819 - val_accuracy: 0.6664
Epoch 173/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8278 -
accuracy: 0.6851 - val_loss: 0.8804 - val_accuracy: 0.6657
Epoch 174/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8254 -
accuracy: 0.6893 - val_loss: 0.8784 - val_accuracy: 0.6621
Epoch 175/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8238 -
accuracy: 0.6878 - val_loss: 0.8757 - val_accuracy: 0.6693
Epoch 176/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8200 -
accuracy: 0.6887 - val_loss: 0.8736 - val_accuracy: 0.6700
Epoch 177/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8186 -
accuracy: 0.6875 - val_loss: 0.8710 - val_accuracy: 0.6736
Epoch 178/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8169 -
accuracy: 0.6885 - val_loss: 0.8703 - val_accuracy: 0.6657
Epoch 179/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8145 -
```

```
accuracy: 0.6916 - val_loss: 0.8695 - val_accuracy: 0.6607
Epoch 180/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8130 -
accuracy: 0.6925 - val_loss: 0.8679 - val_accuracy: 0.6722
Epoch 181/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8122 -
accuracy: 0.6896 - val_loss: 0.8665 - val_accuracy: 0.6722
Epoch 182/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8098 -
accuracy: 0.6910 - val_loss: 0.8624 - val_accuracy: 0.6700
Epoch 183/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8081 -
accuracy: 0.6937 - val_loss: 0.8604 - val_accuracy: 0.6722
Epoch 184/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8061 -
accuracy: 0.6948 - val_loss: 0.8590 - val_accuracy: 0.6679
Epoch 185/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8038 -
accuracy: 0.6941 - val_loss: 0.8575 - val_accuracy: 0.6693
Epoch 186/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8014 -
accuracy: 0.6977 - val_loss: 0.8565 - val_accuracy: 0.6686
Epoch 187/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.8001 -
accuracy: 0.6941 - val_loss: 0.8553 - val_accuracy: 0.6743
Epoch 188/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7993 -
accuracy: 0.6943 - val_loss: 0.8520 - val_accuracy: 0.6772
Epoch 189/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7971 -
accuracy: 0.6954 - val_loss: 0.8506 - val_accuracy: 0.6700
Epoch 190/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7944 -
accuracy: 0.6966 - val_loss: 0.8494 - val_accuracy: 0.6786
Epoch 191/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7929 -
accuracy: 0.6986 - val_loss: 0.8483 - val_accuracy: 0.6779
Epoch 192/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7906 -
accuracy: 0.6970 - val_loss: 0.8448 - val_accuracy: 0.6844
Epoch 193/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7902 -
accuracy: 0.6990 - val_loss: 0.8431 - val_accuracy: 0.6779
Epoch 194/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7884 -
accuracy: 0.6990 - val_loss: 0.8421 - val_accuracy: 0.6772
Epoch 195/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7865 -
```

```
accuracy: 0.7009 - val_loss: 0.8398 - val_accuracy: 0.6830
Epoch 196/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7837 -
accuracy: 0.7004 - val_loss: 0.8382 - val_accuracy: 0.6844
Epoch 197/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7829 -
accuracy: 0.7000 - val_loss: 0.8377 - val_accuracy: 0.6794
Epoch 198/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7818 -
accuracy: 0.7043 - val_loss: 0.8331 - val_accuracy: 0.6909
Epoch 199/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7797 -
accuracy: 0.7024 - val_loss: 0.8333 - val_accuracy: 0.6837
Epoch 200/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7774 -
accuracy: 0.7051 - val_loss: 0.8299 - val_accuracy: 0.6873
Epoch 201/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7765 -
accuracy: 0.7026 - val_loss: 0.8286 - val_accuracy: 0.6866
Epoch 202/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7746 -
accuracy: 0.7031 - val_loss: 0.8278 - val_accuracy: 0.6894
Epoch 203/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7721 -
accuracy: 0.7027 - val_loss: 0.8287 - val_accuracy: 0.6822
Epoch 204/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7706 -
accuracy: 0.7054 - val_loss: 0.8237 - val_accuracy: 0.6887
Epoch 205/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7696 -
accuracy: 0.7038 - val_loss: 0.8245 - val_accuracy: 0.6866
Epoch 206/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7669 -
accuracy: 0.7061 - val_loss: 0.8231 - val_accuracy: 0.6830
Epoch 207/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7643 -
accuracy: 0.7058 - val_loss: 0.8211 - val_accuracy: 0.6894
Epoch 208/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7650 -
accuracy: 0.7094 - val_loss: 0.8215 - val_accuracy: 0.6822
Epoch 209/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7625 -
accuracy: 0.7069 - val_loss: 0.8176 - val_accuracy: 0.6902
Epoch 210/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7605 -
accuracy: 0.7085 - val_loss: 0.8142 - val_accuracy: 0.6959
Epoch 211/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7586 -
```

```
accuracy: 0.7106 - val_loss: 0.8149 - val_accuracy: 0.6873
Epoch 212/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7582 -
accuracy: 0.7105 - val_loss: 0.8116 - val_accuracy: 0.6952
Epoch 213/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7553 -
accuracy: 0.7103 - val_loss: 0.8143 - val_accuracy: 0.6837
Epoch 214/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7548 -
accuracy: 0.7083 - val_loss: 0.8094 - val_accuracy: 0.6909
Epoch 215/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7536 -
accuracy: 0.7119 - val_loss: 0.8079 - val_accuracy: 0.6959
Epoch 216/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7516 -
accuracy: 0.7142 - val_loss: 0.8044 - val_accuracy: 0.6988
Epoch 217/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7505 -
accuracy: 0.7117 - val_loss: 0.8051 - val_accuracy: 0.6923
Epoch 218/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7490 -
accuracy: 0.7137 - val_loss: 0.8030 - val_accuracy: 0.6966
Epoch 219/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7471 -
accuracy: 0.7130 - val_loss: 0.8018 - val_accuracy: 0.6952
Epoch 220/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7454 -
accuracy: 0.7159 - val_loss: 0.8019 - val_accuracy: 0.6945
Epoch 221/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7435 -
accuracy: 0.7159 - val_loss: 0.7996 - val_accuracy: 0.6923
Epoch 222/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7429 -
accuracy: 0.7157 - val_loss: 0.7972 - val_accuracy: 0.6988
Epoch 223/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7398 -
accuracy: 0.7157 - val_loss: 0.7969 - val_accuracy: 0.6923
Epoch 224/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7387 -
accuracy: 0.7182 - val_loss: 0.7961 - val_accuracy: 0.6923
Epoch 225/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7377 -
accuracy: 0.7162 - val_loss: 0.7959 - val_accuracy: 0.6945
Epoch 226/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7358 -
accuracy: 0.7173 - val_loss: 0.7914 - val_accuracy: 0.7024
Epoch 227/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7346 -
```

```
accuracy: 0.7157 - val_loss: 0.7907 - val_accuracy: 0.6959
Epoch 228/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7333 -
accuracy: 0.7212 - val_loss: 0.7889 - val_accuracy: 0.7024
Epoch 229/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7322 -
accuracy: 0.7175 - val_loss: 0.7887 - val_accuracy: 0.6966
Epoch 230/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7301 -
accuracy: 0.7193 - val_loss: 0.7876 - val_accuracy: 0.7031
Epoch 231/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7283 -
accuracy: 0.7220 - val_loss: 0.7832 - val_accuracy: 0.7060
Epoch 232/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7266 -
accuracy: 0.7198 - val_loss: 0.7828 - val_accuracy: 0.7060
Epoch 233/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7264 -
accuracy: 0.7211 - val_loss: 0.7817 - val_accuracy: 0.7009
Epoch 234/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7237 -
accuracy: 0.7220 - val_loss: 0.7807 - val_accuracy: 0.7009
Epoch 235/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7226 -
accuracy: 0.7220 - val_loss: 0.7799 - val_accuracy: 0.6995
Epoch 236/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7210 -
accuracy: 0.7223 - val_loss: 0.7787 - val_accuracy: 0.7031
Epoch 237/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7192 -
accuracy: 0.7250 - val_loss: 0.7750 - val_accuracy: 0.7045
Epoch 238/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7185 -
accuracy: 0.7232 - val_loss: 0.7735 - val_accuracy: 0.7067
Epoch 239/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7153 -
accuracy: 0.7232 - val_loss: 0.7747 - val_accuracy: 0.7088
Epoch 240/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7149 -
accuracy: 0.7257 - val_loss: 0.7755 - val_accuracy: 0.7124
Epoch 241/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7116 -
accuracy: 0.7270 - val_loss: 0.7693 - val_accuracy: 0.7168
Epoch 242/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7117 -
accuracy: 0.7284 - val_loss: 0.7686 - val_accuracy: 0.7096
Epoch 243/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7108 -
```

```
accuracy: 0.7281 - val_loss: 0.7687 - val_accuracy: 0.7074
Epoch 244/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7106 -
accuracy: 0.7286 - val_loss: 0.7655 - val_accuracy: 0.7110
Epoch 245/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7076 -
accuracy: 0.7309 - val_loss: 0.7634 - val_accuracy: 0.7146
Epoch 246/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7065 -
accuracy: 0.7299 - val_loss: 0.7640 - val_accuracy: 0.7124
Epoch 247/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7060 -
accuracy: 0.7292 - val_loss: 0.7609 - val_accuracy: 0.7160
Epoch 248/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7032 -
accuracy: 0.7301 - val_loss: 0.7621 - val_accuracy: 0.7103
Epoch 249/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7020 -
accuracy: 0.7309 - val_loss: 0.7579 - val_accuracy: 0.7168
Epoch 250/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7003 -
accuracy: 0.7342 - val_loss: 0.7575 - val_accuracy: 0.7096
Epoch 251/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.7004 -
accuracy: 0.7331 - val_loss: 0.7556 - val_accuracy: 0.7124
Epoch 252/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6968 -
accuracy: 0.7342 - val_loss: 0.7556 - val_accuracy: 0.7124
Epoch 253/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6956 -
accuracy: 0.7331 - val_loss: 0.7553 - val_accuracy: 0.7117
Epoch 254/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6946 -
accuracy: 0.7360 - val_loss: 0.7536 - val_accuracy: 0.7103
Epoch 255/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6946 -
accuracy: 0.7358 - val_loss: 0.7536 - val_accuracy: 0.7139
Epoch 256/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6897 -
accuracy: 0.7365 - val_loss: 0.7500 - val_accuracy: 0.7103
Epoch 257/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6893 -
accuracy: 0.7344 - val_loss: 0.7492 - val_accuracy: 0.7160
Epoch 258/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6892 -
accuracy: 0.7374 - val_loss: 0.7466 - val_accuracy: 0.7232
Epoch 259/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6870 -
```

```
accuracy: 0.7403 - val_loss: 0.7468 - val_accuracy: 0.7160
Epoch 260/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6860 -
accuracy: 0.7403 - val_loss: 0.7449 - val_accuracy: 0.7196
Epoch 261/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6845 -
accuracy: 0.7387 - val_loss: 0.7427 - val_accuracy: 0.7218
Epoch 262/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6834 -
accuracy: 0.7412 - val_loss: 0.7426 - val_accuracy: 0.7218
Epoch 263/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6814 -
accuracy: 0.7378 - val_loss: 0.7416 - val_accuracy: 0.7232
Epoch 264/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6806 -
accuracy: 0.7439 - val_loss: 0.7399 - val_accuracy: 0.7218
Epoch 265/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6810 -
accuracy: 0.7378 - val_loss: 0.7387 - val_accuracy: 0.7239
Epoch 266/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6787 -
accuracy: 0.7403 - val_loss: 0.7356 - val_accuracy: 0.7225
Epoch 267/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6772 -
accuracy: 0.7407 - val_loss: 0.7384 - val_accuracy: 0.7218
Epoch 268/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6759 -
accuracy: 0.7439 - val_loss: 0.7367 - val_accuracy: 0.7247
Epoch 269/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6724 -
accuracy: 0.7455 - val_loss: 0.7343 - val_accuracy: 0.7247
Epoch 270/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6705 -
accuracy: 0.7442 - val_loss: 0.7336 - val_accuracy: 0.7268
Epoch 271/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6725 -
accuracy: 0.7451 - val_loss: 0.7325 - val_accuracy: 0.7247
Epoch 272/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6704 -
accuracy: 0.7478 - val_loss: 0.7287 - val_accuracy: 0.7261
Epoch 273/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6677 -
accuracy: 0.7473 - val_loss: 0.7285 - val_accuracy: 0.7311
Epoch 274/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6663 -
accuracy: 0.7495 - val_loss: 0.7263 - val_accuracy: 0.7268
Epoch 275/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6669 -
```

```
accuracy: 0.7468 - val_loss: 0.7255 - val_accuracy: 0.7318
Epoch 276/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6659 -
accuracy: 0.7460 - val_loss: 0.7246 - val_accuracy: 0.7297
Epoch 277/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6647 -
accuracy: 0.7495 - val_loss: 0.7245 - val_accuracy: 0.7290
Epoch 278/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6623 -
accuracy: 0.7487 - val_loss: 0.7223 - val_accuracy: 0.7340
Epoch 279/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6611 -
accuracy: 0.7495 - val_loss: 0.7216 - val_accuracy: 0.7290
Epoch 280/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6598 -
accuracy: 0.7507 - val_loss: 0.7238 - val_accuracy: 0.7182
Epoch 281/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6584 -
accuracy: 0.7466 - val_loss: 0.7202 - val_accuracy: 0.7290
Epoch 282/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6582 -
accuracy: 0.7507 - val_loss: 0.7171 - val_accuracy: 0.7362
Epoch 283/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6544 -
accuracy: 0.7522 - val_loss: 0.7148 - val_accuracy: 0.7304
Epoch 284/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6545 -
accuracy: 0.7531 - val_loss: 0.7141 - val_accuracy: 0.7311
Epoch 285/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6528 -
accuracy: 0.7531 - val_loss: 0.7128 - val_accuracy: 0.7333
Epoch 286/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6526 -
accuracy: 0.7527 - val_loss: 0.7112 - val_accuracy: 0.7333
Epoch 287/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6510 -
accuracy: 0.7547 - val_loss: 0.7115 - val_accuracy: 0.7376
Epoch 288/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6486 -
accuracy: 0.7527 - val_loss: 0.7085 - val_accuracy: 0.7362
Epoch 289/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6485 -
accuracy: 0.7575 - val_loss: 0.7090 - val_accuracy: 0.7318
Epoch 290/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6477 -
accuracy: 0.7563 - val_loss: 0.7084 - val_accuracy: 0.7369
Epoch 291/350
5564/5564 [==============================] - 0s 26us/step - loss: 0.6448 -
```

```
accuracy: 0.7608 - val_loss: 0.7084 - val_accuracy: 0.7326
Epoch 292/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6435 -
accuracy: 0.7584 - val_loss: 0.7056 - val_accuracy: 0.7383
Epoch 293/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6424 -
accuracy: 0.7570 - val_loss: 0.7075 - val_accuracy: 0.7326
Epoch 294/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6414 -
accuracy: 0.7570 - val_loss: 0.7018 - val_accuracy: 0.7405
Epoch 295/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6391 -
accuracy: 0.7617 - val_loss: 0.7010 - val_accuracy: 0.7390
Epoch 296/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6375 -
accuracy: 0.7593 - val_loss: 0.7032 - val_accuracy: 0.7333
Epoch 297/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6391 -
accuracy: 0.7615 - val_loss: 0.7024 - val_accuracy: 0.7369
Epoch 298/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6353 -
accuracy: 0.7610 - val_loss: 0.6975 - val_accuracy: 0.7398
Epoch 299/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6354 -
accuracy: 0.7611 - val_loss: 0.7001 - val_accuracy: 0.7369
Epoch 300/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6321 -
accuracy: 0.7613 - val_loss: 0.6969 - val_accuracy: 0.7398
Epoch 301/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6323 -
accuracy: 0.7620 - val_loss: 0.6934 - val_accuracy: 0.7390
Epoch 302/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6305 -
accuracy: 0.7640 - val_loss: 0.6946 - val_accuracy: 0.7369
Epoch 303/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6288 -
accuracy: 0.7665 - val_loss: 0.6931 - val_accuracy: 0.7426
Epoch 304/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6288 -
accuracy: 0.7631 - val_loss: 0.6925 - val_accuracy: 0.7383
Epoch 305/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6286 -
accuracy: 0.7638 - val_loss: 0.6917 - val_accuracy: 0.7362
Epoch 306/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6250 -
accuracy: 0.7662 - val_loss: 0.6883 - val_accuracy: 0.7434
Epoch 307/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6239 -
```

```
accuracy: 0.7689 - val_loss: 0.6898 - val_accuracy: 0.7398
Epoch 308/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6229 -
accuracy: 0.7656 - val_loss: 0.6875 - val_accuracy: 0.7398
Epoch 309/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6233 -
accuracy: 0.7637 - val_loss: 0.6844 - val_accuracy: 0.7448
Epoch 310/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6205 -
accuracy: 0.7682 - val_loss: 0.6876 - val_accuracy: 0.7434
Epoch 311/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6209 -
accuracy: 0.7699 - val_loss: 0.6826 - val_accuracy: 0.7448
Epoch 312/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6166 -
accuracy: 0.7730 - val_loss: 0.6822 - val_accuracy: 0.7462
Epoch 313/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6173 -
accuracy: 0.7698 - val_loss: 0.6811 - val_accuracy: 0.7448
Epoch 314/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6160 -
accuracy: 0.7721 - val_loss: 0.6803 - val_accuracy: 0.7434
Epoch 315/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6152 -
accuracy: 0.7714 - val_loss: 0.6787 - val_accuracy: 0.7469
Epoch 316/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6127 -
accuracy: 0.7708 - val_loss: 0.6783 - val_accuracy: 0.7469
Epoch 317/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6117 -
accuracy: 0.7705 - val_loss: 0.6749 - val_accuracy: 0.7498
Epoch 318/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6119 -
accuracy: 0.7757 - val_loss: 0.6744 - val_accuracy: 0.7469
Epoch 319/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6103 -
accuracy: 0.7721 - val_loss: 0.6739 - val_accuracy: 0.7448
Epoch 320/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6076 -
accuracy: 0.7759 - val_loss: 0.6730 - val_accuracy: 0.7462
Epoch 321/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6071 -
accuracy: 0.7753 - val_loss: 0.6736 - val_accuracy: 0.7448
Epoch 322/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6066 -
accuracy: 0.7755 - val_loss: 0.6692 - val_accuracy: 0.7477
Epoch 323/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6041 -
```

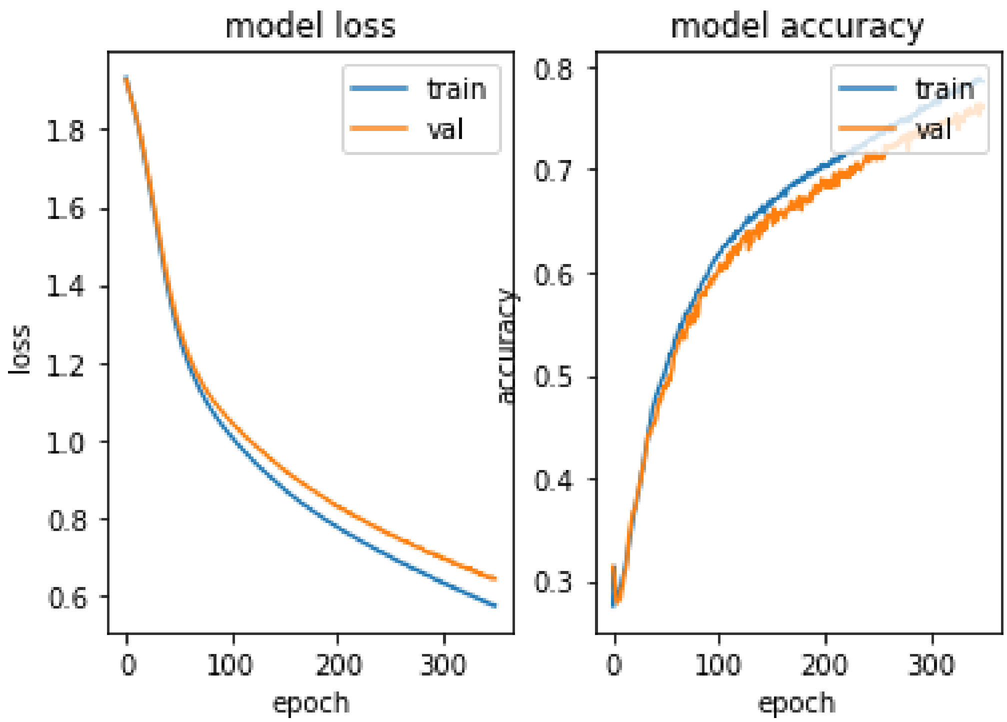
```
accuracy: 0.7762 - val_loss: 0.6687 - val_accuracy: 0.7491
Epoch 324/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6044 -
accuracy: 0.7770 - val_loss: 0.6672 - val_accuracy: 0.7527
Epoch 325/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6020 -
accuracy: 0.7773 - val_loss: 0.6701 - val_accuracy: 0.7448
Epoch 326/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6016 -
accuracy: 0.7753 - val_loss: 0.6655 - val_accuracy: 0.7534
Epoch 327/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.6002 -
accuracy: 0.7789 - val_loss: 0.6708 - val_accuracy: 0.7390
Epoch 328/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5992 -
accuracy: 0.7762 - val_loss: 0.6639 - val_accuracy: 0.7505
Epoch 329/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5989 -
accuracy: 0.7777 - val_loss: 0.6632 - val_accuracy: 0.7520
Epoch 330/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5964 -
accuracy: 0.7807 - val_loss: 0.6631 - val_accuracy: 0.7513
Epoch 331/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5959 -
accuracy: 0.7822 - val_loss: 0.6608 - val_accuracy: 0.7505
Epoch 332/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5938 -
accuracy: 0.7816 - val_loss: 0.6627 - val_accuracy: 0.7462
Epoch 333/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5924 -
accuracy: 0.7836 - val_loss: 0.6595 - val_accuracy: 0.7520
Epoch 334/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5909 -
accuracy: 0.7804 - val_loss: 0.6568 - val_accuracy: 0.7570
Epoch 335/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5915 -
accuracy: 0.7806 - val_loss: 0.6575 - val_accuracy: 0.7556
Epoch 336/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5896 -
accuracy: 0.7822 - val_loss: 0.6533 - val_accuracy: 0.7527
Epoch 337/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5875 -
accuracy: 0.7798 - val_loss: 0.6555 - val_accuracy: 0.7534
Epoch 338/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5885 -
accuracy: 0.7818 - val_loss: 0.6523 - val_accuracy: 0.7599
Epoch 339/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5854 -
```

```
accuracy: 0.7829 - val_loss: 0.6534 - val_accuracy: 0.7549
Epoch 340/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5848 -
accuracy: 0.7827 - val_loss: 0.6514 - val_accuracy: 0.7541
Epoch 341/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5821 -
accuracy: 0.7836 - val_loss: 0.6508 - val_accuracy: 0.7556
Epoch 342/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5835 -
accuracy: 0.7832 - val_loss: 0.6476 - val_accuracy: 0.7570
Epoch 343/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5807 -
accuracy: 0.7870 - val_loss: 0.6474 - val_accuracy: 0.7584
Epoch 344/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5799 -
accuracy: 0.7852 - val_loss: 0.6482 - val_accuracy: 0.7556
Epoch 345/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5794 -
accuracy: 0.7859 - val_loss: 0.6444 - val_accuracy: 0.7635
Epoch 346/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5781 -
accuracy: 0.7854 - val_loss: 0.6438 - val_accuracy: 0.7577
Epoch 347/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5747 -
accuracy: 0.7888 - val_loss: 0.6481 - val_accuracy: 0.7527
Epoch 348/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5747 -
accuracy: 0.7865 - val_loss: 0.6448 - val_accuracy: 0.7570
Epoch 349/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5760 -
accuracy: 0.7863 - val_loss: 0.6422 - val_accuracy: 0.7635
Epoch 350/350
5564/5564 [==============================] - 0s 25us/step - loss: 0.5735 -
accuracy: 0.7858 - val_loss: 0.6431 - val_accuracy: 0.7599
```

[102]:

```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
```

```
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```



[103]:
```
score = model.evaluate(x=x_testcnn_scaled, y=y_test)
predictions = model.predict(x_testcnn_scaled)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
535/535 [==============================] - 0s 66us/step
Test loss: 0.6338018636837184
Test accuracy: 0.7738317847251892
```

[104]:
```
print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
 ↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support

           A       0.94      0.81      0.87       127
           E       0.70      0.84      0.76        81
           F       0.77      0.65      0.71        46
           L       0.78      0.81      0.79        69
```

```
           N       0.64      0.63      0.64        71
           T       0.70      0.98      0.82        62
           W       0.84      0.65      0.73        79

    accuracy                           0.77       535
   macro avg       0.77      0.77      0.76       535
weighted avg       0.79      0.77      0.77       535
```

6.0.12 3. minmax scale:

[105]:
```python
scaler = MinMaxScaler()
scaler.fit(x_traincnn[:,:,0])
x_traincnn_scaled = scaler.transform(x_traincnn[:,:,0])
x_traincnn_scaled = x_traincnn_scaled.reshape(x_traincnn_scaled.shape[0],␣
 ↪x_traincnn_scaled.shape[1], 1)

x_valcnn_scaled = scaler.transform(x_valcnn[:,:,0])
x_valcnn_scaled = x_valcnn_scaled.reshape(x_valcnn_scaled.shape[0],␣
 ↪x_valcnn_scaled.shape[1], 1)

x_testcnn_scaled = scaler.transform(x_testcnn[:,:,0])
x_testcnn_scaled = x_testcnn_scaled.reshape(x_testcnn_scaled.shape[0],␣
 ↪x_testcnn_scaled.shape[1], 1)
```

[107]:
```python
history = model.fit(x_traincnn_scaled, y_train, batch_size=256, epochs=350,␣
 ↪validation_data=(x_valcnn_scaled, y_val))
```

```
Train on 5564 samples, validate on 1391 samples
Epoch 1/350
5564/5564 [==============================] - 0s 81us/step - loss: 1.9388 -
accuracy: 0.2471 - val_loss: 1.9368 - val_accuracy: 0.2358
Epoch 2/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.9322 -
accuracy: 0.2389 - val_loss: 1.9310 - val_accuracy: 0.2344
Epoch 3/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.9256 -
accuracy: 0.2381 - val_loss: 1.9250 - val_accuracy: 0.2344
Epoch 4/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.9189 -
accuracy: 0.2381 - val_loss: 1.9185 - val_accuracy: 0.2344
Epoch 5/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.9120 -
accuracy: 0.2381 - val_loss: 1.9123 - val_accuracy: 0.2344
Epoch 6/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.9052 -
accuracy: 0.2381 - val_loss: 1.9058 - val_accuracy: 0.2344
```

```
Epoch 7/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8982 -
accuracy: 0.2381 - val_loss: 1.8992 - val_accuracy: 0.2344
Epoch 8/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8915 -
accuracy: 0.2381 - val_loss: 1.8925 - val_accuracy: 0.2344
Epoch 9/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8848 -
accuracy: 0.2381 - val_loss: 1.8859 - val_accuracy: 0.2344
Epoch 10/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8784 -
accuracy: 0.2381 - val_loss: 1.8797 - val_accuracy: 0.2344
Epoch 11/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8723 -
accuracy: 0.2381 - val_loss: 1.8735 - val_accuracy: 0.2344
Epoch 12/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8660 -
accuracy: 0.2381 - val_loss: 1.8671 - val_accuracy: 0.2344
Epoch 13/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8598 -
accuracy: 0.2381 - val_loss: 1.8606 - val_accuracy: 0.2344
Epoch 14/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8534 -
accuracy: 0.2381 - val_loss: 1.8542 - val_accuracy: 0.2344
Epoch 15/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8467 -
accuracy: 0.2385 - val_loss: 1.8472 - val_accuracy: 0.2344
Epoch 16/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8396 -
accuracy: 0.2408 - val_loss: 1.8400 - val_accuracy: 0.2372
Epoch 17/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8323 -
accuracy: 0.2482 - val_loss: 1.8328 - val_accuracy: 0.2574
Epoch 18/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8246 -
accuracy: 0.2717 - val_loss: 1.8250 - val_accuracy: 0.2660
Epoch 19/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8165 -
accuracy: 0.2800 - val_loss: 1.8170 - val_accuracy: 0.2782
Epoch 20/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.8079 -
accuracy: 0.2935 - val_loss: 1.8084 - val_accuracy: 0.2832
Epoch 21/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7988 -
accuracy: 0.2976 - val_loss: 1.7994 - val_accuracy: 0.2926
Epoch 22/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7895 -
accuracy: 0.3036 - val_loss: 1.7903 - val_accuracy: 0.2969
```

```
Epoch 23/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7797 -
accuracy: 0.3068 - val_loss: 1.7806 - val_accuracy: 0.3027
Epoch 24/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7696 -
accuracy: 0.3100 - val_loss: 1.7708 - val_accuracy: 0.3077
Epoch 25/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7597 -
accuracy: 0.3129 - val_loss: 1.7611 - val_accuracy: 0.3113
Epoch 26/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7493 -
accuracy: 0.3179 - val_loss: 1.7512 - val_accuracy: 0.3142
Epoch 27/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7388 -
accuracy: 0.3215 - val_loss: 1.7407 - val_accuracy: 0.3192
Epoch 28/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7280 -
accuracy: 0.3242 - val_loss: 1.7300 - val_accuracy: 0.3228
Epoch 29/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7172 -
accuracy: 0.3278 - val_loss: 1.7195 - val_accuracy: 0.3242
Epoch 30/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.7064 -
accuracy: 0.3267 - val_loss: 1.7090 - val_accuracy: 0.3293
Epoch 31/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6955 -
accuracy: 0.3318 - val_loss: 1.6981 - val_accuracy: 0.3307
Epoch 32/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6850 -
accuracy: 0.3345 - val_loss: 1.6879 - val_accuracy: 0.3314
Epoch 33/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6743 -
accuracy: 0.3327 - val_loss: 1.6773 - val_accuracy: 0.3357
Epoch 34/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6637 -
accuracy: 0.3352 - val_loss: 1.6666 - val_accuracy: 0.3386
Epoch 35/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6536 -
accuracy: 0.3395 - val_loss: 1.6573 - val_accuracy: 0.3364
Epoch 36/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6438 -
accuracy: 0.3424 - val_loss: 1.6478 - val_accuracy: 0.3422
Epoch 37/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6343 -
accuracy: 0.3514 - val_loss: 1.6381 - val_accuracy: 0.3609
Epoch 38/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6250 -
accuracy: 0.3512 - val_loss: 1.6287 - val_accuracy: 0.3645
```

```
Epoch 39/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6159 -
accuracy: 0.3589 - val_loss: 1.6200 - val_accuracy: 0.3652
Epoch 40/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.6072 -
accuracy: 0.3657 - val_loss: 1.6118 - val_accuracy: 0.3645
Epoch 41/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5992 -
accuracy: 0.3647 - val_loss: 1.6036 - val_accuracy: 0.3717
Epoch 42/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5914 -
accuracy: 0.3663 - val_loss: 1.5953 - val_accuracy: 0.3789
Epoch 43/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5844 -
accuracy: 0.3650 - val_loss: 1.5887 - val_accuracy: 0.3789
Epoch 44/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5771 -
accuracy: 0.3704 - val_loss: 1.5818 - val_accuracy: 0.3817
Epoch 45/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5707 -
accuracy: 0.3724 - val_loss: 1.5751 - val_accuracy: 0.3746
Epoch 46/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5644 -
accuracy: 0.3769 - val_loss: 1.5699 - val_accuracy: 0.3810
Epoch 47/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5588 -
accuracy: 0.3751 - val_loss: 1.5629 - val_accuracy: 0.3781
Epoch 48/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5532 -
accuracy: 0.3738 - val_loss: 1.5568 - val_accuracy: 0.3882
Epoch 49/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5479 -
accuracy: 0.3801 - val_loss: 1.5518 - val_accuracy: 0.3861
Epoch 50/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5426 -
accuracy: 0.3814 - val_loss: 1.5463 - val_accuracy: 0.3846
Epoch 51/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5377 -
accuracy: 0.3801 - val_loss: 1.5406 - val_accuracy: 0.3817
Epoch 52/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5326 -
accuracy: 0.3826 - val_loss: 1.5353 - val_accuracy: 0.3896
Epoch 53/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5281 -
accuracy: 0.3819 - val_loss: 1.5315 - val_accuracy: 0.3918
Epoch 54/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5238 -
accuracy: 0.3826 - val_loss: 1.5255 - val_accuracy: 0.3868
```

```
Epoch 55/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5188 -
accuracy: 0.3870 - val_loss: 1.5226 - val_accuracy: 0.3875
Epoch 56/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5149 -
accuracy: 0.3877 - val_loss: 1.5167 - val_accuracy: 0.3904
Epoch 57/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5105 -
accuracy: 0.3904 - val_loss: 1.5136 - val_accuracy: 0.3925
Epoch 58/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5063 -
accuracy: 0.3902 - val_loss: 1.5076 - val_accuracy: 0.3997
Epoch 59/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.5016 -
accuracy: 0.3947 - val_loss: 1.5024 - val_accuracy: 0.3968
Epoch 60/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4980 -
accuracy: 0.3963 - val_loss: 1.4994 - val_accuracy: 0.3983
Epoch 61/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4939 -
accuracy: 0.3963 - val_loss: 1.4949 - val_accuracy: 0.4004
Epoch 62/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4899 -
accuracy: 0.3974 - val_loss: 1.4902 - val_accuracy: 0.4004
Epoch 63/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4855 -
accuracy: 0.4029 - val_loss: 1.4863 - val_accuracy: 0.4026
Epoch 64/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4816 -
accuracy: 0.4022 - val_loss: 1.4815 - val_accuracy: 0.4047
Epoch 65/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4778 -
accuracy: 0.4038 - val_loss: 1.4780 - val_accuracy: 0.4040
Epoch 66/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4736 -
accuracy: 0.4109 - val_loss: 1.4740 - val_accuracy: 0.4019
Epoch 67/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4698 -
accuracy: 0.4137 - val_loss: 1.4696 - val_accuracy: 0.4076
Epoch 68/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4658 -
accuracy: 0.4139 - val_loss: 1.4657 - val_accuracy: 0.4069
Epoch 69/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4618 -
accuracy: 0.4139 - val_loss: 1.4616 - val_accuracy: 0.4083
Epoch 70/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4577 -
accuracy: 0.4168 - val_loss: 1.4572 - val_accuracy: 0.4069
```

```
Epoch 71/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4540 -
accuracy: 0.4211 - val_loss: 1.4538 - val_accuracy: 0.4112
Epoch 72/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4500 -
accuracy: 0.4229 - val_loss: 1.4479 - val_accuracy: 0.4249
Epoch 73/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4461 -
accuracy: 0.4265 - val_loss: 1.4452 - val_accuracy: 0.4184
Epoch 74/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4415 -
accuracy: 0.4274 - val_loss: 1.4401 - val_accuracy: 0.4162
Epoch 75/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4372 -
accuracy: 0.4297 - val_loss: 1.4357 - val_accuracy: 0.4256
Epoch 76/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4329 -
accuracy: 0.4340 - val_loss: 1.4327 - val_accuracy: 0.4184
Epoch 77/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4286 -
accuracy: 0.4335 - val_loss: 1.4277 - val_accuracy: 0.4206
Epoch 78/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4245 -
accuracy: 0.4349 - val_loss: 1.4230 - val_accuracy: 0.4270
Epoch 79/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4201 -
accuracy: 0.4391 - val_loss: 1.4195 - val_accuracy: 0.4321
Epoch 80/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4164 -
accuracy: 0.4425 - val_loss: 1.4158 - val_accuracy: 0.4292
Epoch 81/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4120 -
accuracy: 0.4459 - val_loss: 1.4105 - val_accuracy: 0.4321
Epoch 82/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4071 -
accuracy: 0.4495 - val_loss: 1.4079 - val_accuracy: 0.4321
Epoch 83/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.4034 -
accuracy: 0.4466 - val_loss: 1.4013 - val_accuracy: 0.4400
Epoch 84/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3990 -
accuracy: 0.4513 - val_loss: 1.4003 - val_accuracy: 0.4364
Epoch 85/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3946 -
accuracy: 0.4524 - val_loss: 1.3930 - val_accuracy: 0.4472
Epoch 86/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3898 -
accuracy: 0.4603 - val_loss: 1.3910 - val_accuracy: 0.4385
```

```
Epoch 87/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3857 -
accuracy: 0.4594 - val_loss: 1.3843 - val_accuracy: 0.4421
Epoch 88/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3813 -
accuracy: 0.4599 - val_loss: 1.3793 - val_accuracy: 0.4479
Epoch 89/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3769 -
accuracy: 0.4628 - val_loss: 1.3756 - val_accuracy: 0.4536
Epoch 90/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3728 -
accuracy: 0.4637 - val_loss: 1.3713 - val_accuracy: 0.4558
Epoch 91/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3680 -
accuracy: 0.4664 - val_loss: 1.3678 - val_accuracy: 0.4515
Epoch 92/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3639 -
accuracy: 0.4662 - val_loss: 1.3636 - val_accuracy: 0.4623
Epoch 93/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3596 -
accuracy: 0.4693 - val_loss: 1.3596 - val_accuracy: 0.4594
Epoch 94/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3552 -
accuracy: 0.4734 - val_loss: 1.3538 - val_accuracy: 0.4680
Epoch 95/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3515 -
accuracy: 0.4720 - val_loss: 1.3494 - val_accuracy: 0.4644
Epoch 96/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3467 -
accuracy: 0.4729 - val_loss: 1.3442 - val_accuracy: 0.4709
Epoch 97/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3425 -
accuracy: 0.4801 - val_loss: 1.3422 - val_accuracy: 0.4687
Epoch 98/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3386 -
accuracy: 0.4815 - val_loss: 1.3363 - val_accuracy: 0.4781
Epoch 99/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3347 -
accuracy: 0.4802 - val_loss: 1.3316 - val_accuracy: 0.4838
Epoch 100/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3292 -
accuracy: 0.4851 - val_loss: 1.3297 - val_accuracy: 0.4781
Epoch 101/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3258 -
accuracy: 0.4872 - val_loss: 1.3229 - val_accuracy: 0.4867
Epoch 102/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3205 -
accuracy: 0.4863 - val_loss: 1.3183 - val_accuracy: 0.4917
```

```
Epoch 103/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3155 -
accuracy: 0.4867 - val_loss: 1.3135 - val_accuracy: 0.4889
Epoch 104/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3114 -
accuracy: 0.4887 - val_loss: 1.3103 - val_accuracy: 0.4932
Epoch 105/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3065 -
accuracy: 0.4903 - val_loss: 1.3045 - val_accuracy: 0.4925
Epoch 106/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.3027 -
accuracy: 0.4944 - val_loss: 1.2998 - val_accuracy: 0.4960
Epoch 107/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2981 -
accuracy: 0.5004 - val_loss: 1.2987 - val_accuracy: 0.4953
Epoch 108/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.2943 -
accuracy: 0.4975 - val_loss: 1.2948 - val_accuracy: 0.4939
Epoch 109/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.2906 -
accuracy: 0.5009 - val_loss: 1.2878 - val_accuracy: 0.4996
Epoch 110/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2862 -
accuracy: 0.5025 - val_loss: 1.2840 - val_accuracy: 0.5025
Epoch 111/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2825 -
accuracy: 0.5068 - val_loss: 1.2819 - val_accuracy: 0.5068
Epoch 112/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2787 -
accuracy: 0.5068 - val_loss: 1.2762 - val_accuracy: 0.5032
Epoch 113/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2739 -
accuracy: 0.5093 - val_loss: 1.2718 - val_accuracy: 0.5047
Epoch 114/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.2691 -
accuracy: 0.5131 - val_loss: 1.2662 - val_accuracy: 0.5090
Epoch 115/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2661 -
accuracy: 0.5110 - val_loss: 1.2643 - val_accuracy: 0.5040
Epoch 116/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2626 -
accuracy: 0.5146 - val_loss: 1.2599 - val_accuracy: 0.5140
Epoch 117/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2579 -
accuracy: 0.5115 - val_loss: 1.2557 - val_accuracy: 0.5133
Epoch 118/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.2536 -
accuracy: 0.5217 - val_loss: 1.2505 - val_accuracy: 0.5155
```

```
Epoch 119/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2500 -
accuracy: 0.5194 - val_loss: 1.2491 - val_accuracy: 0.5104
Epoch 120/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2454 -
accuracy: 0.5205 - val_loss: 1.2460 - val_accuracy: 0.5032
Epoch 121/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2419 -
accuracy: 0.5239 - val_loss: 1.2407 - val_accuracy: 0.5119
Epoch 122/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2376 -
accuracy: 0.5250 - val_loss: 1.2348 - val_accuracy: 0.5241
Epoch 123/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2340 -
accuracy: 0.5246 - val_loss: 1.2307 - val_accuracy: 0.5226
Epoch 124/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2297 -
accuracy: 0.5286 - val_loss: 1.2273 - val_accuracy: 0.5198
Epoch 125/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2262 -
accuracy: 0.5271 - val_loss: 1.2229 - val_accuracy: 0.5241
Epoch 126/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2226 -
accuracy: 0.5297 - val_loss: 1.2194 - val_accuracy: 0.5226
Epoch 127/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2181 -
accuracy: 0.5325 - val_loss: 1.2178 - val_accuracy: 0.5176
Epoch 128/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2157 -
accuracy: 0.5340 - val_loss: 1.2136 - val_accuracy: 0.5176
Epoch 129/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2112 -
accuracy: 0.5374 - val_loss: 1.2095 - val_accuracy: 0.5241
Epoch 130/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2081 -
accuracy: 0.5370 - val_loss: 1.2086 - val_accuracy: 0.5298
Epoch 131/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2043 -
accuracy: 0.5421 - val_loss: 1.2012 - val_accuracy: 0.5320
Epoch 132/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.2005 -
accuracy: 0.5397 - val_loss: 1.2006 - val_accuracy: 0.5219
Epoch 133/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1966 -
accuracy: 0.5426 - val_loss: 1.1979 - val_accuracy: 0.5262
Epoch 134/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1930 -
accuracy: 0.5408 - val_loss: 1.1909 - val_accuracy: 0.5385
```

```
Epoch 135/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1902 -
accuracy: 0.5433 - val_loss: 1.1899 - val_accuracy: 0.5349
Epoch 136/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1876 -
accuracy: 0.5431 - val_loss: 1.1863 - val_accuracy: 0.5399
Epoch 137/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1838 -
accuracy: 0.5473 - val_loss: 1.1843 - val_accuracy: 0.5313
Epoch 138/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1812 -
accuracy: 0.5440 - val_loss: 1.1800 - val_accuracy: 0.5413
Epoch 139/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1777 -
accuracy: 0.5474 - val_loss: 1.1750 - val_accuracy: 0.5478
Epoch 140/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.1750 -
accuracy: 0.5507 - val_loss: 1.1724 - val_accuracy: 0.5421
Epoch 141/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1722 -
accuracy: 0.5474 - val_loss: 1.1695 - val_accuracy: 0.5464
Epoch 142/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1687 -
accuracy: 0.5494 - val_loss: 1.1663 - val_accuracy: 0.5492
Epoch 143/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.1663 -
accuracy: 0.5534 - val_loss: 1.1624 - val_accuracy: 0.5457
Epoch 144/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1633 -
accuracy: 0.5528 - val_loss: 1.1599 - val_accuracy: 0.5485
Epoch 145/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1603 -
accuracy: 0.5527 - val_loss: 1.1571 - val_accuracy: 0.5557
Epoch 146/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1565 -
accuracy: 0.5595 - val_loss: 1.1545 - val_accuracy: 0.5564
Epoch 147/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1537 -
accuracy: 0.5564 - val_loss: 1.1509 - val_accuracy: 0.5572
Epoch 148/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1502 -
accuracy: 0.5573 - val_loss: 1.1512 - val_accuracy: 0.5457
Epoch 149/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1483 -
accuracy: 0.5613 - val_loss: 1.1455 - val_accuracy: 0.5557
Epoch 150/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1446 -
accuracy: 0.5591 - val_loss: 1.1464 - val_accuracy: 0.5557
```

```
Epoch 151/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1414 -
accuracy: 0.5615 - val_loss: 1.1395 - val_accuracy: 0.5629
Epoch 152/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1398 -
accuracy: 0.5607 - val_loss: 1.1375 - val_accuracy: 0.5586
Epoch 153/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1352 -
accuracy: 0.5645 - val_loss: 1.1350 - val_accuracy: 0.5651
Epoch 154/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1330 -
accuracy: 0.5620 - val_loss: 1.1315 - val_accuracy: 0.5672
Epoch 155/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1305 -
accuracy: 0.5679 - val_loss: 1.1282 - val_accuracy: 0.5643
Epoch 156/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1285 -
accuracy: 0.5625 - val_loss: 1.1277 - val_accuracy: 0.5607
Epoch 157/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1251 -
accuracy: 0.5647 - val_loss: 1.1242 - val_accuracy: 0.5586
Epoch 158/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1215 -
accuracy: 0.5699 - val_loss: 1.1256 - val_accuracy: 0.5586
Epoch 159/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1207 -
accuracy: 0.5661 - val_loss: 1.1188 - val_accuracy: 0.5658
Epoch 160/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1173 -
accuracy: 0.5714 - val_loss: 1.1169 - val_accuracy: 0.5658
Epoch 161/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1147 -
accuracy: 0.5699 - val_loss: 1.1127 - val_accuracy: 0.5737
Epoch 162/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1115 -
accuracy: 0.5730 - val_loss: 1.1155 - val_accuracy: 0.5651
Epoch 163/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1107 -
accuracy: 0.5735 - val_loss: 1.1094 - val_accuracy: 0.5679
Epoch 164/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1068 -
accuracy: 0.5699 - val_loss: 1.1083 - val_accuracy: 0.5787
Epoch 165/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1039 -
accuracy: 0.5764 - val_loss: 1.1090 - val_accuracy: 0.5658
Epoch 166/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.1030 -
accuracy: 0.5767 - val_loss: 1.1011 - val_accuracy: 0.5780
```

```
Epoch 167/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0998 -
accuracy: 0.5737 - val_loss: 1.1029 - val_accuracy: 0.5715
Epoch 168/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0974 -
accuracy: 0.5787 - val_loss: 1.0962 - val_accuracy: 0.5780
Epoch 169/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0944 -
accuracy: 0.5775 - val_loss: 1.0972 - val_accuracy: 0.5737
Epoch 170/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0931 -
accuracy: 0.5717 - val_loss: 1.0917 - val_accuracy: 0.5780
Epoch 171/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0919 -
accuracy: 0.5769 - val_loss: 1.0911 - val_accuracy: 0.5758
Epoch 172/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0885 -
accuracy: 0.5791 - val_loss: 1.0889 - val_accuracy: 0.5787
Epoch 173/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0858 -
accuracy: 0.5821 - val_loss: 1.0925 - val_accuracy: 0.5751
Epoch 174/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0844 -
accuracy: 0.5807 - val_loss: 1.0899 - val_accuracy: 0.5780
Epoch 175/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0813 -
accuracy: 0.5818 - val_loss: 1.0847 - val_accuracy: 0.5809
Epoch 176/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0785 -
accuracy: 0.5857 - val_loss: 1.0793 - val_accuracy: 0.5823
Epoch 177/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0764 -
accuracy: 0.5829 - val_loss: 1.0789 - val_accuracy: 0.5845
Epoch 178/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0762 -
accuracy: 0.5838 - val_loss: 1.0745 - val_accuracy: 0.5873
Epoch 179/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0721 -
accuracy: 0.5827 - val_loss: 1.0731 - val_accuracy: 0.5809
Epoch 180/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0715 -
accuracy: 0.5847 - val_loss: 1.0739 - val_accuracy: 0.5830
Epoch 181/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0700 -
accuracy: 0.5884 - val_loss: 1.0705 - val_accuracy: 0.5845
Epoch 182/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0681 -
accuracy: 0.5848 - val_loss: 1.0681 - val_accuracy: 0.5802
```

```
Epoch 183/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0650 -
accuracy: 0.5890 - val_loss: 1.0642 - val_accuracy: 0.5816
Epoch 184/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0640 -
accuracy: 0.5895 - val_loss: 1.0675 - val_accuracy: 0.5852
Epoch 185/350
5564/5564 [==============================] - 0s 34us/step - loss: 1.0631 -
accuracy: 0.5924 - val_loss: 1.0650 - val_accuracy: 0.5852
Epoch 186/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0604 -
accuracy: 0.5881 - val_loss: 1.0607 - val_accuracy: 0.5830
Epoch 187/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0582 -
accuracy: 0.5906 - val_loss: 1.0669 - val_accuracy: 0.5830
Epoch 188/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0562 -
accuracy: 0.5902 - val_loss: 1.0596 - val_accuracy: 0.5888
Epoch 189/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0552 -
accuracy: 0.5911 - val_loss: 1.0594 - val_accuracy: 0.5881
Epoch 190/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0525 -
accuracy: 0.5947 - val_loss: 1.0600 - val_accuracy: 0.5845
Epoch 191/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0522 -
accuracy: 0.5951 - val_loss: 1.0536 - val_accuracy: 0.5852
Epoch 192/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0495 -
accuracy: 0.5942 - val_loss: 1.0527 - val_accuracy: 0.5873
Epoch 193/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0468 -
accuracy: 0.5911 - val_loss: 1.0499 - val_accuracy: 0.5909
Epoch 194/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0446 -
accuracy: 0.5969 - val_loss: 1.0471 - val_accuracy: 0.5960
Epoch 195/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0439 -
accuracy: 0.5978 - val_loss: 1.0446 - val_accuracy: 0.5945
Epoch 196/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0413 -
accuracy: 0.5938 - val_loss: 1.0453 - val_accuracy: 0.5909
Epoch 197/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0386 -
accuracy: 0.5981 - val_loss: 1.0458 - val_accuracy: 0.5895
Epoch 198/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0375 -
accuracy: 0.5974 - val_loss: 1.0396 - val_accuracy: 0.6039
```

```
Epoch 199/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0368 -
accuracy: 0.6001 - val_loss: 1.0379 - val_accuracy: 0.6003
Epoch 200/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0344 -
accuracy: 0.6032 - val_loss: 1.0379 - val_accuracy: 0.5866
Epoch 201/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0329 -
accuracy: 0.6030 - val_loss: 1.0350 - val_accuracy: 0.6039
Epoch 202/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0321 -
accuracy: 0.6003 - val_loss: 1.0332 - val_accuracy: 0.5981
Epoch 203/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0308 -
accuracy: 0.6010 - val_loss: 1.0347 - val_accuracy: 0.5967
Epoch 204/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0290 -
accuracy: 0.6021 - val_loss: 1.0314 - val_accuracy: 0.5988
Epoch 205/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0282 -
accuracy: 0.5990 - val_loss: 1.0302 - val_accuracy: 0.5924
Epoch 206/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0265 -
accuracy: 0.6039 - val_loss: 1.0282 - val_accuracy: 0.5931
Epoch 207/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0232 -
accuracy: 0.6042 - val_loss: 1.0254 - val_accuracy: 0.6068
Epoch 208/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0217 -
accuracy: 0.6051 - val_loss: 1.0253 - val_accuracy: 0.6118
Epoch 209/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0212 -
accuracy: 0.6028 - val_loss: 1.0271 - val_accuracy: 0.5938
Epoch 210/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0186 -
accuracy: 0.6037 - val_loss: 1.0228 - val_accuracy: 0.6003
Epoch 211/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0192 -
accuracy: 0.6064 - val_loss: 1.0234 - val_accuracy: 0.6032
Epoch 212/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0171 -
accuracy: 0.6055 - val_loss: 1.0187 - val_accuracy: 0.6060
Epoch 213/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0145 -
accuracy: 0.6068 - val_loss: 1.0156 - val_accuracy: 0.6132
Epoch 214/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0136 -
accuracy: 0.6062 - val_loss: 1.0176 - val_accuracy: 0.6154
```

```
Epoch 215/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0127 -
accuracy: 0.6057 - val_loss: 1.0140 - val_accuracy: 0.6132
Epoch 216/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0100 -
accuracy: 0.6084 - val_loss: 1.0159 - val_accuracy: 0.5974
Epoch 217/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0097 -
accuracy: 0.6120 - val_loss: 1.0128 - val_accuracy: 0.6068
Epoch 218/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0060 -
accuracy: 0.6129 - val_loss: 1.0098 - val_accuracy: 0.6104
Epoch 219/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0062 -
accuracy: 0.6082 - val_loss: 1.0106 - val_accuracy: 0.6082
Epoch 220/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0061 -
accuracy: 0.6087 - val_loss: 1.0081 - val_accuracy: 0.6075
Epoch 221/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0046 -
accuracy: 0.6089 - val_loss: 1.0069 - val_accuracy: 0.6183
Epoch 222/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0016 -
accuracy: 0.6159 - val_loss: 1.0102 - val_accuracy: 0.6089
Epoch 223/350
5564/5564 [==============================] - 0s 33us/step - loss: 1.0015 -
accuracy: 0.6118 - val_loss: 1.0054 - val_accuracy: 0.6075
Epoch 224/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9994 -
accuracy: 0.6141 - val_loss: 1.0035 - val_accuracy: 0.6161
Epoch 225/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9988 -
accuracy: 0.6143 - val_loss: 1.0034 - val_accuracy: 0.6111
Epoch 226/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9948 -
accuracy: 0.6114 - val_loss: 1.0004 - val_accuracy: 0.6139
Epoch 227/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9962 -
accuracy: 0.6150 - val_loss: 1.0010 - val_accuracy: 0.6168
Epoch 228/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9955 -
accuracy: 0.6156 - val_loss: 0.9988 - val_accuracy: 0.6104
Epoch 229/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9909 -
accuracy: 0.6202 - val_loss: 0.9980 - val_accuracy: 0.6247
Epoch 230/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9914 -
accuracy: 0.6118 - val_loss: 0.9957 - val_accuracy: 0.6204
```

```
Epoch 231/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9908 -
accuracy: 0.6181 - val_loss: 0.9937 - val_accuracy: 0.6211
Epoch 232/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9883 -
accuracy: 0.6166 - val_loss: 0.9964 - val_accuracy: 0.6139
Epoch 233/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9871 -
accuracy: 0.6148 - val_loss: 0.9926 - val_accuracy: 0.6161
Epoch 234/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9865 -
accuracy: 0.6166 - val_loss: 0.9942 - val_accuracy: 0.6168
Epoch 235/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9861 -
accuracy: 0.6177 - val_loss: 0.9929 - val_accuracy: 0.6175
Epoch 236/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9841 -
accuracy: 0.6166 - val_loss: 0.9879 - val_accuracy: 0.6269
Epoch 237/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9842 -
accuracy: 0.6184 - val_loss: 0.9863 - val_accuracy: 0.6254
Epoch 238/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9816 -
accuracy: 0.6208 - val_loss: 0.9859 - val_accuracy: 0.6240
Epoch 239/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9810 -
accuracy: 0.6206 - val_loss: 0.9848 - val_accuracy: 0.6219
Epoch 240/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9799 -
accuracy: 0.6224 - val_loss: 0.9839 - val_accuracy: 0.6219
Epoch 241/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9786 -
accuracy: 0.6215 - val_loss: 0.9910 - val_accuracy: 0.6053
Epoch 242/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9782 -
accuracy: 0.6208 - val_loss: 0.9821 - val_accuracy: 0.6254
Epoch 243/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9770 -
accuracy: 0.6199 - val_loss: 0.9812 - val_accuracy: 0.6298
Epoch 244/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9751 -
accuracy: 0.6224 - val_loss: 0.9817 - val_accuracy: 0.6233
Epoch 245/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9742 -
accuracy: 0.6229 - val_loss: 0.9793 - val_accuracy: 0.6240
Epoch 246/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9735 -
accuracy: 0.6219 - val_loss: 0.9774 - val_accuracy: 0.6312
```

```
Epoch 247/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9718 -
accuracy: 0.6285 - val_loss: 0.9773 - val_accuracy: 0.6226
Epoch 248/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9698 -
accuracy: 0.6260 - val_loss: 0.9761 - val_accuracy: 0.6197
Epoch 249/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9699 -
accuracy: 0.6301 - val_loss: 0.9790 - val_accuracy: 0.6190
Epoch 250/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9678 -
accuracy: 0.6253 - val_loss: 0.9778 - val_accuracy: 0.6154
Epoch 251/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9656 -
accuracy: 0.6211 - val_loss: 0.9728 - val_accuracy: 0.6290
Epoch 252/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9676 -
accuracy: 0.6254 - val_loss: 0.9716 - val_accuracy: 0.6254
Epoch 253/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9666 -
accuracy: 0.6276 - val_loss: 0.9698 - val_accuracy: 0.6334
Epoch 254/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9641 -
accuracy: 0.6281 - val_loss: 0.9690 - val_accuracy: 0.6370
Epoch 255/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9637 -
accuracy: 0.6281 - val_loss: 0.9696 - val_accuracy: 0.6362
Epoch 256/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9628 -
accuracy: 0.6281 - val_loss: 0.9686 - val_accuracy: 0.6262
Epoch 257/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9624 -
accuracy: 0.6262 - val_loss: 0.9671 - val_accuracy: 0.6312
Epoch 258/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9600 -
accuracy: 0.6285 - val_loss: 0.9648 - val_accuracy: 0.6334
Epoch 259/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9585 -
accuracy: 0.6265 - val_loss: 0.9655 - val_accuracy: 0.6334
Epoch 260/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9578 -
accuracy: 0.6317 - val_loss: 0.9636 - val_accuracy: 0.6355
Epoch 261/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9582 -
accuracy: 0.6281 - val_loss: 0.9626 - val_accuracy: 0.6377
Epoch 262/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9560 -
accuracy: 0.6281 - val_loss: 0.9609 - val_accuracy: 0.6326
```

```
Epoch 263/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9544 -
accuracy: 0.6323 - val_loss: 0.9620 - val_accuracy: 0.6283
Epoch 264/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9545 -
accuracy: 0.6310 - val_loss: 0.9624 - val_accuracy: 0.6219
Epoch 265/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9539 -
accuracy: 0.6308 - val_loss: 0.9608 - val_accuracy: 0.6319
Epoch 266/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9525 -
accuracy: 0.6307 - val_loss: 0.9613 - val_accuracy: 0.6262
Epoch 267/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9509 -
accuracy: 0.6317 - val_loss: 0.9566 - val_accuracy: 0.6370
Epoch 268/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9509 -
accuracy: 0.6339 - val_loss: 0.9575 - val_accuracy: 0.6334
Epoch 269/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9475 -
accuracy: 0.6384 - val_loss: 0.9602 - val_accuracy: 0.6298
Epoch 270/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9487 -
accuracy: 0.6353 - val_loss: 0.9559 - val_accuracy: 0.6290
Epoch 271/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9469 -
accuracy: 0.6346 - val_loss: 0.9530 - val_accuracy: 0.6413
Epoch 272/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9468 -
accuracy: 0.6400 - val_loss: 0.9529 - val_accuracy: 0.6348
Epoch 273/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9441 -
accuracy: 0.6355 - val_loss: 0.9522 - val_accuracy: 0.6334
Epoch 274/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9446 -
accuracy: 0.6393 - val_loss: 0.9537 - val_accuracy: 0.6362
Epoch 275/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9444 -
accuracy: 0.6353 - val_loss: 0.9501 - val_accuracy: 0.6348
Epoch 276/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9422 -
accuracy: 0.6380 - val_loss: 0.9499 - val_accuracy: 0.6341
Epoch 277/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9419 -
accuracy: 0.6370 - val_loss: 0.9483 - val_accuracy: 0.6362
Epoch 278/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9373 -
accuracy: 0.6402 - val_loss: 0.9463 - val_accuracy: 0.6391
```

```
Epoch 279/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9399 -
accuracy: 0.6391 - val_loss: 0.9485 - val_accuracy: 0.6326
Epoch 280/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9378 -
accuracy: 0.6386 - val_loss: 0.9500 - val_accuracy: 0.6341
Epoch 281/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9376 -
accuracy: 0.6380 - val_loss: 0.9472 - val_accuracy: 0.6391
Epoch 282/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9352 -
accuracy: 0.6404 - val_loss: 0.9428 - val_accuracy: 0.6413
Epoch 283/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9368 -
accuracy: 0.6373 - val_loss: 0.9441 - val_accuracy: 0.6362
Epoch 284/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9353 -
accuracy: 0.6425 - val_loss: 0.9415 - val_accuracy: 0.6456
Epoch 285/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9346 -
accuracy: 0.6398 - val_loss: 0.9467 - val_accuracy: 0.6355
Epoch 286/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9342 -
accuracy: 0.6377 - val_loss: 0.9387 - val_accuracy: 0.6506
Epoch 287/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9316 -
accuracy: 0.6400 - val_loss: 0.9377 - val_accuracy: 0.6405
Epoch 288/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9294 -
accuracy: 0.6384 - val_loss: 0.9376 - val_accuracy: 0.6470
Epoch 289/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9304 -
accuracy: 0.6404 - val_loss: 0.9372 - val_accuracy: 0.6413
Epoch 290/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9295 -
accuracy: 0.6456 - val_loss: 0.9426 - val_accuracy: 0.6384
Epoch 291/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9281 -
accuracy: 0.6429 - val_loss: 0.9379 - val_accuracy: 0.6492
Epoch 292/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9258 -
accuracy: 0.6458 - val_loss: 0.9360 - val_accuracy: 0.6477
Epoch 293/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9272 -
accuracy: 0.6402 - val_loss: 0.9322 - val_accuracy: 0.6449
Epoch 294/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9272 -
accuracy: 0.6445 - val_loss: 0.9339 - val_accuracy: 0.6420
```

```
Epoch 295/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9239 -
accuracy: 0.6468 - val_loss: 0.9346 - val_accuracy: 0.6420
Epoch 296/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9246 -
accuracy: 0.6438 - val_loss: 0.9375 - val_accuracy: 0.6377
Epoch 297/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9216 -
accuracy: 0.6470 - val_loss: 0.9414 - val_accuracy: 0.6449
Epoch 298/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9230 -
accuracy: 0.6463 - val_loss: 0.9305 - val_accuracy: 0.6520
Epoch 299/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9198 -
accuracy: 0.6422 - val_loss: 0.9282 - val_accuracy: 0.6463
Epoch 300/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9216 -
accuracy: 0.6445 - val_loss: 0.9290 - val_accuracy: 0.6427
Epoch 301/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9205 -
accuracy: 0.6452 - val_loss: 0.9292 - val_accuracy: 0.6420
Epoch 302/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9172 -
accuracy: 0.6447 - val_loss: 0.9284 - val_accuracy: 0.6499
Epoch 303/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9183 -
accuracy: 0.6474 - val_loss: 0.9292 - val_accuracy: 0.6449
Epoch 304/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9152 -
accuracy: 0.6506 - val_loss: 0.9286 - val_accuracy: 0.6420
Epoch 305/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9152 -
accuracy: 0.6479 - val_loss: 0.9267 - val_accuracy: 0.6477
Epoch 306/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9151 -
accuracy: 0.6499 - val_loss: 0.9248 - val_accuracy: 0.6470
Epoch 307/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9140 -
accuracy: 0.6495 - val_loss: 0.9229 - val_accuracy: 0.6564
Epoch 308/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9138 -
accuracy: 0.6490 - val_loss: 0.9274 - val_accuracy: 0.6449
Epoch 309/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9132 -
accuracy: 0.6470 - val_loss: 0.9224 - val_accuracy: 0.6499
Epoch 310/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9100 -
accuracy: 0.6504 - val_loss: 0.9204 - val_accuracy: 0.6542
```

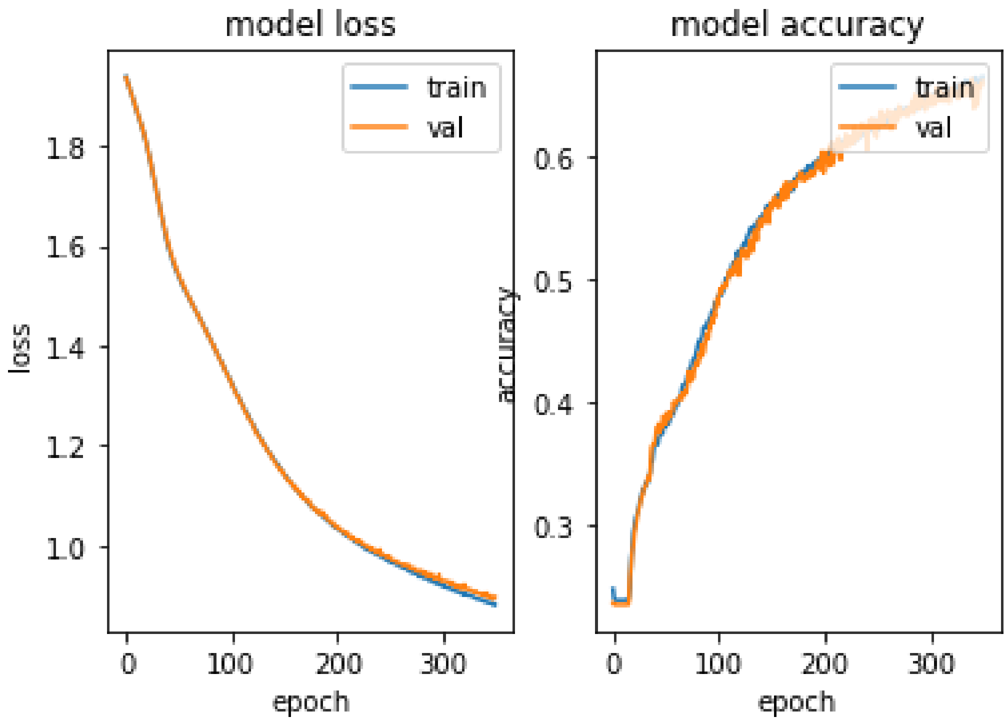
```
Epoch 311/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9102 -
accuracy: 0.6456 - val_loss: 0.9199 - val_accuracy: 0.6542
Epoch 312/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9104 -
accuracy: 0.6533 - val_loss: 0.9223 - val_accuracy: 0.6449
Epoch 313/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9079 -
accuracy: 0.6481 - val_loss: 0.9165 - val_accuracy: 0.6506
Epoch 314/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9083 -
accuracy: 0.6501 - val_loss: 0.9176 - val_accuracy: 0.6513
Epoch 315/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9073 -
accuracy: 0.6520 - val_loss: 0.9203 - val_accuracy: 0.6449
Epoch 316/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9074 -
accuracy: 0.6510 - val_loss: 0.9149 - val_accuracy: 0.6535
Epoch 317/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9053 -
accuracy: 0.6520 - val_loss: 0.9138 - val_accuracy: 0.6499
Epoch 318/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9056 -
accuracy: 0.6524 - val_loss: 0.9185 - val_accuracy: 0.6449
Epoch 319/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9026 -
accuracy: 0.6535 - val_loss: 0.9147 - val_accuracy: 0.6441
Epoch 320/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9050 -
accuracy: 0.6547 - val_loss: 0.9136 - val_accuracy: 0.6528
Epoch 321/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9007 -
accuracy: 0.6542 - val_loss: 0.9218 - val_accuracy: 0.6499
Epoch 322/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9027 -
accuracy: 0.6494 - val_loss: 0.9134 - val_accuracy: 0.6564
Epoch 323/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9020 -
accuracy: 0.6533 - val_loss: 0.9137 - val_accuracy: 0.6506
Epoch 324/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9001 -
accuracy: 0.6537 - val_loss: 0.9123 - val_accuracy: 0.6520
Epoch 325/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9002 -
accuracy: 0.6567 - val_loss: 0.9121 - val_accuracy: 0.6528
Epoch 326/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.9012 -
accuracy: 0.6526 - val_loss: 0.9108 - val_accuracy: 0.6513
```

```
Epoch 327/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8986 -
accuracy: 0.6565 - val_loss: 0.9105 - val_accuracy: 0.6470
Epoch 328/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8989 -
accuracy: 0.6553 - val_loss: 0.9106 - val_accuracy: 0.6499
Epoch 329/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8974 -
accuracy: 0.6569 - val_loss: 0.9072 - val_accuracy: 0.6578
Epoch 330/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8962 -
accuracy: 0.6560 - val_loss: 0.9065 - val_accuracy: 0.6556
Epoch 331/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8955 -
accuracy: 0.6574 - val_loss: 0.9070 - val_accuracy: 0.6499
Epoch 332/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8933 -
accuracy: 0.6596 - val_loss: 0.9084 - val_accuracy: 0.6571
Epoch 333/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8933 -
accuracy: 0.6583 - val_loss: 0.9055 - val_accuracy: 0.6564
Epoch 334/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8925 -
accuracy: 0.6625 - val_loss: 0.9027 - val_accuracy: 0.6492
Epoch 335/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8924 -
accuracy: 0.6578 - val_loss: 0.9033 - val_accuracy: 0.6571
Epoch 336/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8914 -
accuracy: 0.6607 - val_loss: 0.9026 - val_accuracy: 0.6528
Epoch 337/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8899 -
accuracy: 0.6585 - val_loss: 0.9019 - val_accuracy: 0.6564
Epoch 338/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8893 -
accuracy: 0.6587 - val_loss: 0.9017 - val_accuracy: 0.6556
Epoch 339/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8907 -
accuracy: 0.6598 - val_loss: 0.9041 - val_accuracy: 0.6485
Epoch 340/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8878 -
accuracy: 0.6587 - val_loss: 0.9002 - val_accuracy: 0.6636
Epoch 341/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8882 -
accuracy: 0.6630 - val_loss: 0.8981 - val_accuracy: 0.6556
Epoch 342/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8866 -
accuracy: 0.6594 - val_loss: 0.9045 - val_accuracy: 0.6391
```

```
Epoch 343/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8857 -
accuracy: 0.6619 - val_loss: 0.8969 - val_accuracy: 0.6578
Epoch 344/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8846 -
accuracy: 0.6645 - val_loss: 0.8987 - val_accuracy: 0.6592
Epoch 345/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8844 -
accuracy: 0.6589 - val_loss: 0.8988 - val_accuracy: 0.6420
Epoch 346/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8858 -
accuracy: 0.6578 - val_loss: 0.8998 - val_accuracy: 0.6470
Epoch 347/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8842 -
accuracy: 0.6592 - val_loss: 0.8932 - val_accuracy: 0.6571
Epoch 348/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8817 -
accuracy: 0.6609 - val_loss: 0.8931 - val_accuracy: 0.6585
Epoch 349/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8836 -
accuracy: 0.6614 - val_loss: 0.8938 - val_accuracy: 0.6571
Epoch 350/350
5564/5564 [==============================] - 0s 33us/step - loss: 0.8813 -
accuracy: 0.6652 - val_loss: 0.8966 - val_accuracy: 0.6614
```

[108]:

```python
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```



[109]:
```
score = model.evaluate(x=x_testcnn_scaled, y=y_test)
predictions = model.predict(x_testcnn_scaled)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
535/535 [==============================] - 0s 65us/step
Test loss: 0.9200951375693919
Test accuracy: 0.6560747623443604
```

[110]:
```
print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
 ↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support

           A       0.91      0.68      0.78       127
           E       0.59      0.58      0.58        81
           F       0.60      0.46      0.52        46
           L       0.67      0.64      0.65        69
           N       0.48      0.63      0.55        71
           T       0.69      0.95      0.80        62
           W       0.60      0.62      0.61        79

    accuracy                           0.66       535
```

```
   macro avg       0.65      0.65      0.64       535
weighted avg       0.68      0.66      0.66       535
```

Model is better with no scaling! Models with scaling don't converge even after >350 epochs.

6.0.13 model 2

[111]:
```
model = Sequential()

model.add(Conv1D(256, 5, padding='same',input_shape=(40,1)))
model.add(Activation('relu'))
model.add(Conv1D(128, 5, padding='same'))
model.add(Activation('relu'))
model.add(Dropout(0.1))
model.add(MaxPooling1D(pool_size=(8)))
model.add(Conv1D(128, 5, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(64, 5, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(64, 5, padding='same'))
model.add(Activation('relu'))
model.add(Conv1D(32, 5, padding='same'))
model.add(Activation('relu'))
model.add(Flatten())
model.add(Dense(7))
model.add(Activation('softmax'))
opt = keras.optimizers.RMSprop(lr=0.00001, decay=1e-6)
model.compile(loss='categorical_crossentropy', optimizer=opt,␣
 ↪metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_17 (Conv1D)           (None, 40, 256)           1536
_________________________________________________________________
activation_21 (Activation)   (None, 40, 256)           0
_________________________________________________________________
conv1d_18 (Conv1D)           (None, 40, 128)           163968
_________________________________________________________________
activation_22 (Activation)   (None, 40, 128)           0
_________________________________________________________________
dropout_5 (Dropout)          (None, 40, 128)           0
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 5, 128)            0
```

```
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 5, 128)            82048
_________________________________________________________________
activation_23 (Activation)   (None, 5, 128)            0
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 5, 64)             41024
_________________________________________________________________
activation_24 (Activation)   (None, 5, 64)             0
_________________________________________________________________
conv1d_21 (Conv1D)           (None, 5, 64)             20544
_________________________________________________________________
activation_25 (Activation)   (None, 5, 64)             0
_________________________________________________________________
conv1d_22 (Conv1D)           (None, 5, 32)             10272
_________________________________________________________________
activation_26 (Activation)   (None, 5, 32)             0
_________________________________________________________________
flatten_5 (Flatten)          (None, 160)               0
_________________________________________________________________
dense_5 (Dense)              (None, 7)                 1127
_________________________________________________________________
activation_27 (Activation)   (None, 7)                 0
=================================================================
Total params: 320,519
Trainable params: 320,519
Non-trainable params: 0
_________________________________________________________________
```

[112]:

```
#history = model.fit(x_traincnn, y_train, batch_size=16, epochs=250,␣
 ↪validation_data=(x_valcnn, y_val),
#                    callbacks=[checkpoint])
history = model.fit(x_traincnn, y_train, batch_size=512, epochs=200,␣
 ↪validation_data=(x_valcnn, y_val))
```

```
Train on 5564 samples, validate on 1391 samples
Epoch 1/200
5564/5564 [==============================] - 1s 101us/step - loss: 2.4905 -
accuracy: 0.1154 - val_loss: 2.1297 - val_accuracy: 0.1172
Epoch 2/200
5564/5564 [==============================] - 0s 27us/step - loss: 2.0450 -
accuracy: 0.1763 - val_loss: 1.8948 - val_accuracy: 0.3436
Epoch 3/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.8656 -
accuracy: 0.3269 - val_loss: 1.7790 - val_accuracy: 0.3465
Epoch 4/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.7727 -
accuracy: 0.3460 - val_loss: 1.7127 - val_accuracy: 0.3523
Epoch 5/200
```

```
5564/5564 [==============================] - 0s 28us/step - loss: 1.7098 -
accuracy: 0.3623 - val_loss: 1.6571 - val_accuracy: 0.3825
Epoch 6/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.6598 -
accuracy: 0.3817 - val_loss: 1.6066 - val_accuracy: 0.3968
Epoch 7/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.6077 -
accuracy: 0.4033 - val_loss: 1.5613 - val_accuracy: 0.4069
Epoch 8/200
5564/5564 [==============================] - 0s 29us/step - loss: 1.5681 -
accuracy: 0.4119 - val_loss: 1.5251 - val_accuracy: 0.4385
Epoch 9/200
5564/5564 [==============================] - 0s 29us/step - loss: 1.5259 -
accuracy: 0.4281 - val_loss: 1.4893 - val_accuracy: 0.4234
Epoch 10/200
5564/5564 [==============================] - 0s 29us/step - loss: 1.4917 -
accuracy: 0.4387 - val_loss: 1.4552 - val_accuracy: 0.4493
Epoch 11/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.4578 -
accuracy: 0.4524 - val_loss: 1.4248 - val_accuracy: 0.4457
Epoch 12/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.4289 -
accuracy: 0.4648 - val_loss: 1.3977 - val_accuracy: 0.4781
Epoch 13/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.3997 -
accuracy: 0.4777 - val_loss: 1.3706 - val_accuracy: 0.5068
Epoch 14/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.3717 -
accuracy: 0.4937 - val_loss: 1.3487 - val_accuracy: 0.4996
Epoch 15/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.3455 -
accuracy: 0.4942 - val_loss: 1.3237 - val_accuracy: 0.5054
Epoch 16/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.3233 -
accuracy: 0.5081 - val_loss: 1.3019 - val_accuracy: 0.5119
Epoch 17/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.2988 -
accuracy: 0.5237 - val_loss: 1.2857 - val_accuracy: 0.5090
Epoch 18/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.2774 -
accuracy: 0.5212 - val_loss: 1.2620 - val_accuracy: 0.5248
Epoch 19/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.2591 -
accuracy: 0.5325 - val_loss: 1.2426 - val_accuracy: 0.5399
Epoch 20/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.2396 -
accuracy: 0.5350 - val_loss: 1.2239 - val_accuracy: 0.5370
Epoch 21/200
```

```
5564/5564 [==============================] - 0s 28us/step - loss: 1.2199 -
accuracy: 0.5465 - val_loss: 1.2080 - val_accuracy: 0.5514
Epoch 22/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.2018 -
accuracy: 0.5444 - val_loss: 1.1931 - val_accuracy: 0.5564
Epoch 23/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1848 -
accuracy: 0.5582 - val_loss: 1.1758 - val_accuracy: 0.5665
Epoch 24/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1726 -
accuracy: 0.5604 - val_loss: 1.1605 - val_accuracy: 0.5679
Epoch 25/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1549 -
accuracy: 0.5667 - val_loss: 1.1504 - val_accuracy: 0.5636
Epoch 26/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1393 -
accuracy: 0.5746 - val_loss: 1.1411 - val_accuracy: 0.5586
Epoch 27/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1285 -
accuracy: 0.5735 - val_loss: 1.1220 - val_accuracy: 0.5845
Epoch 28/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1149 -
accuracy: 0.5789 - val_loss: 1.1134 - val_accuracy: 0.5830
Epoch 29/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.1042 -
accuracy: 0.5825 - val_loss: 1.1004 - val_accuracy: 0.5924
Epoch 30/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0879 -
accuracy: 0.5870 - val_loss: 1.0890 - val_accuracy: 0.5953
Epoch 31/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0783 -
accuracy: 0.5927 - val_loss: 1.0827 - val_accuracy: 0.6003
Epoch 32/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0684 -
accuracy: 0.5906 - val_loss: 1.0691 - val_accuracy: 0.6060
Epoch 33/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0611 -
accuracy: 0.5931 - val_loss: 1.0604 - val_accuracy: 0.6089
Epoch 34/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0497 -
accuracy: 0.6064 - val_loss: 1.0551 - val_accuracy: 0.6068
Epoch 35/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0418 -
accuracy: 0.6003 - val_loss: 1.0422 - val_accuracy: 0.6096
Epoch 36/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0301 -
accuracy: 0.6120 - val_loss: 1.0392 - val_accuracy: 0.6075
Epoch 37/200
```

```
5564/5564 [==============================] - 0s 28us/step - loss: 1.0187 -
accuracy: 0.6161 - val_loss: 1.0287 - val_accuracy: 0.6154
Epoch 38/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0089 -
accuracy: 0.6100 - val_loss: 1.0174 - val_accuracy: 0.6154
Epoch 39/200
5564/5564 [==============================] - 0s 28us/step - loss: 1.0028 -
accuracy: 0.6244 - val_loss: 1.0108 - val_accuracy: 0.6226
Epoch 40/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9923 -
accuracy: 0.6226 - val_loss: 0.9999 - val_accuracy: 0.6298
Epoch 41/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9893 -
accuracy: 0.6206 - val_loss: 0.9940 - val_accuracy: 0.6290
Epoch 42/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9791 -
accuracy: 0.6269 - val_loss: 0.9915 - val_accuracy: 0.6226
Epoch 43/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9734 -
accuracy: 0.6281 - val_loss: 0.9857 - val_accuracy: 0.6204
Epoch 44/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9625 -
accuracy: 0.6312 - val_loss: 0.9715 - val_accuracy: 0.6326
Epoch 45/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9532 -
accuracy: 0.6334 - val_loss: 0.9713 - val_accuracy: 0.6341
Epoch 46/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9537 -
accuracy: 0.6343 - val_loss: 0.9574 - val_accuracy: 0.6405
Epoch 47/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9468 -
accuracy: 0.6280 - val_loss: 0.9542 - val_accuracy: 0.6420
Epoch 48/200
5564/5564 [==============================] - 0s 29us/step - loss: 0.9359 -
accuracy: 0.6382 - val_loss: 0.9500 - val_accuracy: 0.6384
Epoch 49/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9333 -
accuracy: 0.6346 - val_loss: 0.9439 - val_accuracy: 0.6441
Epoch 50/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9270 -
accuracy: 0.6420 - val_loss: 0.9378 - val_accuracy: 0.6398
Epoch 51/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9205 -
accuracy: 0.6452 - val_loss: 0.9339 - val_accuracy: 0.6449
Epoch 52/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.9133 -
accuracy: 0.6495 - val_loss: 0.9276 - val_accuracy: 0.6528
Epoch 53/200
```

```
5564/5564 [==============================] - 0s 28us/step - loss: 0.9055 -
accuracy: 0.6459 - val_loss: 0.9235 - val_accuracy: 0.6485
Epoch 54/200
5564/5564 [==============================] - 0s 29us/step - loss: 0.9007 -
accuracy: 0.6512 - val_loss: 0.9194 - val_accuracy: 0.6470
Epoch 55/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8971 -
accuracy: 0.6549 - val_loss: 0.9100 - val_accuracy: 0.6571
Epoch 56/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8911 -
accuracy: 0.6571 - val_loss: 0.9058 - val_accuracy: 0.6542
Epoch 57/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8880 -
accuracy: 0.6506 - val_loss: 0.9010 - val_accuracy: 0.6549
Epoch 58/200
5564/5564 [==============================] - 0s 29us/step - loss: 0.8771 -
accuracy: 0.6627 - val_loss: 0.8986 - val_accuracy: 0.6535
Epoch 59/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8740 -
accuracy: 0.6556 - val_loss: 0.8952 - val_accuracy: 0.6585
Epoch 60/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8725 -
accuracy: 0.6650 - val_loss: 0.8922 - val_accuracy: 0.6585
Epoch 61/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8664 -
accuracy: 0.6623 - val_loss: 0.8781 - val_accuracy: 0.6643
Epoch 62/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8578 -
accuracy: 0.6639 - val_loss: 0.8772 - val_accuracy: 0.6671
Epoch 63/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8566 -
accuracy: 0.6643 - val_loss: 0.8723 - val_accuracy: 0.6664
Epoch 64/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8499 -
accuracy: 0.6684 - val_loss: 0.8683 - val_accuracy: 0.6715
Epoch 65/200
5564/5564 [==============================] - 0s 29us/step - loss: 0.8460 -
accuracy: 0.6697 - val_loss: 0.8665 - val_accuracy: 0.6779
Epoch 66/200
5564/5564 [==============================] - 0s 28us/step - loss: 0.8435 -
accuracy: 0.6704 - val_loss: 0.8593 - val_accuracy: 0.6751
Epoch 67/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8402 -
accuracy: 0.6720 - val_loss: 0.8605 - val_accuracy: 0.6693
Epoch 68/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8329 -
accuracy: 0.6752 - val_loss: 0.8536 - val_accuracy: 0.6758
Epoch 69/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.8295 -
accuracy: 0.6756 - val_loss: 0.8530 - val_accuracy: 0.6686
Epoch 70/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8260 -
accuracy: 0.6792 - val_loss: 0.8515 - val_accuracy: 0.6736
Epoch 71/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8192 -
accuracy: 0.6839 - val_loss: 0.8406 - val_accuracy: 0.6808
Epoch 72/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8182 -
accuracy: 0.6795 - val_loss: 0.8393 - val_accuracy: 0.6794
Epoch 73/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8142 -
accuracy: 0.6794 - val_loss: 0.8394 - val_accuracy: 0.6779
Epoch 74/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8067 -
accuracy: 0.6857 - val_loss: 0.8305 - val_accuracy: 0.6909
Epoch 75/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8074 -
accuracy: 0.6819 - val_loss: 0.8279 - val_accuracy: 0.6837
Epoch 76/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7995 -
accuracy: 0.6884 - val_loss: 0.8281 - val_accuracy: 0.6794
Epoch 77/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.8014 -
accuracy: 0.6900 - val_loss: 0.8220 - val_accuracy: 0.6844
Epoch 78/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7938 -
accuracy: 0.6903 - val_loss: 0.8163 - val_accuracy: 0.6887
Epoch 79/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7906 -
accuracy: 0.6914 - val_loss: 0.8141 - val_accuracy: 0.6830
Epoch 80/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7829 -
accuracy: 0.6957 - val_loss: 0.8100 - val_accuracy: 0.6909
Epoch 81/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7835 -
accuracy: 0.6963 - val_loss: 0.8039 - val_accuracy: 0.6945
Epoch 82/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7759 -
accuracy: 0.7031 - val_loss: 0.8026 - val_accuracy: 0.6866
Epoch 83/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7727 -
accuracy: 0.7036 - val_loss: 0.8001 - val_accuracy: 0.6887
Epoch 84/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7700 -
accuracy: 0.6993 - val_loss: 0.7930 - val_accuracy: 0.7009
Epoch 85/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.7675 -
accuracy: 0.6997 - val_loss: 0.7952 - val_accuracy: 0.6902
Epoch 86/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7613 -
accuracy: 0.7031 - val_loss: 0.7878 - val_accuracy: 0.7038
Epoch 87/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7601 -
accuracy: 0.7074 - val_loss: 0.7794 - val_accuracy: 0.7052
Epoch 88/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7530 -
accuracy: 0.7036 - val_loss: 0.7929 - val_accuracy: 0.6930
Epoch 89/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7508 -
accuracy: 0.7092 - val_loss: 0.7757 - val_accuracy: 0.7031
Epoch 90/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7494 -
accuracy: 0.7078 - val_loss: 0.7714 - val_accuracy: 0.7117
Epoch 91/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7446 -
accuracy: 0.7078 - val_loss: 0.7711 - val_accuracy: 0.7052
Epoch 92/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7404 -
accuracy: 0.7157 - val_loss: 0.7705 - val_accuracy: 0.7081
Epoch 93/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7352 -
accuracy: 0.7200 - val_loss: 0.7659 - val_accuracy: 0.7024
Epoch 94/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7333 -
accuracy: 0.7142 - val_loss: 0.7630 - val_accuracy: 0.7060
Epoch 95/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7320 -
accuracy: 0.7178 - val_loss: 0.7563 - val_accuracy: 0.7117
Epoch 96/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7254 -
accuracy: 0.7193 - val_loss: 0.7631 - val_accuracy: 0.7024
Epoch 97/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7245 -
accuracy: 0.7229 - val_loss: 0.7505 - val_accuracy: 0.7160
Epoch 98/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7221 -
accuracy: 0.7218 - val_loss: 0.7559 - val_accuracy: 0.7081
Epoch 99/200
5564/5564 [==============================] - 0s 26us/step - loss: 0.7193 -
accuracy: 0.7247 - val_loss: 0.7519 - val_accuracy: 0.7096
Epoch 100/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7139 -
accuracy: 0.7187 - val_loss: 0.7418 - val_accuracy: 0.7153
Epoch 101/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.7111 -
accuracy: 0.7270 - val_loss: 0.7446 - val_accuracy: 0.7124
Epoch 102/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7110 -
accuracy: 0.7247 - val_loss: 0.7381 - val_accuracy: 0.7254
Epoch 103/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7025 -
accuracy: 0.7268 - val_loss: 0.7310 - val_accuracy: 0.7196
Epoch 104/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7021 -
accuracy: 0.7292 - val_loss: 0.7329 - val_accuracy: 0.7218
Epoch 105/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.7007 -
accuracy: 0.7272 - val_loss: 0.7278 - val_accuracy: 0.7211
Epoch 106/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6947 -
accuracy: 0.7306 - val_loss: 0.7326 - val_accuracy: 0.7211
Epoch 107/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6928 -
accuracy: 0.7326 - val_loss: 0.7306 - val_accuracy: 0.7211
Epoch 108/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6876 -
accuracy: 0.7322 - val_loss: 0.7182 - val_accuracy: 0.7203
Epoch 109/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6863 -
accuracy: 0.7390 - val_loss: 0.7223 - val_accuracy: 0.7160
Epoch 110/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6840 -
accuracy: 0.7353 - val_loss: 0.7098 - val_accuracy: 0.7340
Epoch 111/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6814 -
accuracy: 0.7367 - val_loss: 0.7099 - val_accuracy: 0.7275
Epoch 112/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6773 -
accuracy: 0.7394 - val_loss: 0.7069 - val_accuracy: 0.7254
Epoch 113/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6728 -
accuracy: 0.7423 - val_loss: 0.7036 - val_accuracy: 0.7340
Epoch 114/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6712 -
accuracy: 0.7423 - val_loss: 0.7002 - val_accuracy: 0.7326
Epoch 115/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6714 -
accuracy: 0.7419 - val_loss: 0.6964 - val_accuracy: 0.7311
Epoch 116/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6683 -
accuracy: 0.7410 - val_loss: 0.6989 - val_accuracy: 0.7283
Epoch 117/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.6621 -
accuracy: 0.7453 - val_loss: 0.6980 - val_accuracy: 0.7304
Epoch 118/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6646 -
accuracy: 0.7394 - val_loss: 0.6883 - val_accuracy: 0.7398
Epoch 119/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6575 -
accuracy: 0.7455 - val_loss: 0.6861 - val_accuracy: 0.7354
Epoch 120/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6583 -
accuracy: 0.7460 - val_loss: 0.6906 - val_accuracy: 0.7347
Epoch 121/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6540 -
accuracy: 0.7486 - val_loss: 0.6833 - val_accuracy: 0.7383
Epoch 122/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6502 -
accuracy: 0.7480 - val_loss: 0.6796 - val_accuracy: 0.7383
Epoch 123/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6463 -
accuracy: 0.7513 - val_loss: 0.6785 - val_accuracy: 0.7318
Epoch 124/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6443 -
accuracy: 0.7527 - val_loss: 0.6748 - val_accuracy: 0.7412
Epoch 125/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6392 -
accuracy: 0.7550 - val_loss: 0.6736 - val_accuracy: 0.7390
Epoch 126/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6364 -
accuracy: 0.7574 - val_loss: 0.6742 - val_accuracy: 0.7347
Epoch 127/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6370 -
accuracy: 0.7507 - val_loss: 0.6730 - val_accuracy: 0.7390
Epoch 128/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6308 -
accuracy: 0.7593 - val_loss: 0.6694 - val_accuracy: 0.7426
Epoch 129/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6306 -
accuracy: 0.7601 - val_loss: 0.6665 - val_accuracy: 0.7484
Epoch 130/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6308 -
accuracy: 0.7529 - val_loss: 0.6618 - val_accuracy: 0.7369
Epoch 131/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6239 -
accuracy: 0.7620 - val_loss: 0.6563 - val_accuracy: 0.7469
Epoch 132/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6243 -
accuracy: 0.7563 - val_loss: 0.6543 - val_accuracy: 0.7477
Epoch 133/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.6244 -
accuracy: 0.7601 - val_loss: 0.6530 - val_accuracy: 0.7520
Epoch 134/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6152 -
accuracy: 0.7644 - val_loss: 0.6516 - val_accuracy: 0.7434
Epoch 135/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6140 -
accuracy: 0.7633 - val_loss: 0.6483 - val_accuracy: 0.7520
Epoch 136/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6147 -
accuracy: 0.7613 - val_loss: 0.6516 - val_accuracy: 0.7477
Epoch 137/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6104 -
accuracy: 0.7606 - val_loss: 0.6473 - val_accuracy: 0.7484
Epoch 138/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6112 -
accuracy: 0.7640 - val_loss: 0.6499 - val_accuracy: 0.7376
Epoch 139/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6049 -
accuracy: 0.7685 - val_loss: 0.6422 - val_accuracy: 0.7541
Epoch 140/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6037 -
accuracy: 0.7689 - val_loss: 0.6365 - val_accuracy: 0.7491
Epoch 141/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6029 -
accuracy: 0.7671 - val_loss: 0.6406 - val_accuracy: 0.7491
Epoch 142/200
5564/5564 [==============================] - 0s 26us/step - loss: 0.6024 -
accuracy: 0.7646 - val_loss: 0.6396 - val_accuracy: 0.7505
Epoch 143/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.6000 -
accuracy: 0.7680 - val_loss: 0.6353 - val_accuracy: 0.7613
Epoch 144/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5933 -
accuracy: 0.7728 - val_loss: 0.6395 - val_accuracy: 0.7563
Epoch 145/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5932 -
accuracy: 0.7732 - val_loss: 0.6290 - val_accuracy: 0.7649
Epoch 146/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5928 -
accuracy: 0.7707 - val_loss: 0.6246 - val_accuracy: 0.7642
Epoch 147/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5892 -
accuracy: 0.7705 - val_loss: 0.6240 - val_accuracy: 0.7656
Epoch 148/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5865 -
accuracy: 0.7818 - val_loss: 0.6223 - val_accuracy: 0.7513
Epoch 149/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.5875 -
accuracy: 0.7764 - val_loss: 0.6178 - val_accuracy: 0.7606
Epoch 150/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5824 -
accuracy: 0.7691 - val_loss: 0.6200 - val_accuracy: 0.7685
Epoch 151/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5777 -
accuracy: 0.7815 - val_loss: 0.6152 - val_accuracy: 0.7649
Epoch 152/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5812 -
accuracy: 0.7779 - val_loss: 0.6173 - val_accuracy: 0.7721
Epoch 153/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5771 -
accuracy: 0.7811 - val_loss: 0.6118 - val_accuracy: 0.7635
Epoch 154/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5747 -
accuracy: 0.7782 - val_loss: 0.6130 - val_accuracy: 0.7599
Epoch 155/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5705 -
accuracy: 0.7820 - val_loss: 0.6145 - val_accuracy: 0.7771
Epoch 156/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5728 -
accuracy: 0.7771 - val_loss: 0.6047 - val_accuracy: 0.7613
Epoch 157/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5676 -
accuracy: 0.7852 - val_loss: 0.6039 - val_accuracy: 0.7635
Epoch 158/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5646 -
accuracy: 0.7832 - val_loss: 0.5989 - val_accuracy: 0.7764
Epoch 159/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5637 -
accuracy: 0.7850 - val_loss: 0.6003 - val_accuracy: 0.7779
Epoch 160/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5607 -
accuracy: 0.7903 - val_loss: 0.5958 - val_accuracy: 0.7649
Epoch 161/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5600 -
accuracy: 0.7841 - val_loss: 0.5961 - val_accuracy: 0.7822
Epoch 162/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5603 -
accuracy: 0.7861 - val_loss: 0.5959 - val_accuracy: 0.7649
Epoch 163/200
5564/5564 [==============================] - 0s 26us/step - loss: 0.5562 -
accuracy: 0.7856 - val_loss: 0.6083 - val_accuracy: 0.7735
Epoch 164/200
5564/5564 [==============================] - 0s 26us/step - loss: 0.5570 -
accuracy: 0.7858 - val_loss: 0.5912 - val_accuracy: 0.7822
Epoch 165/200
```

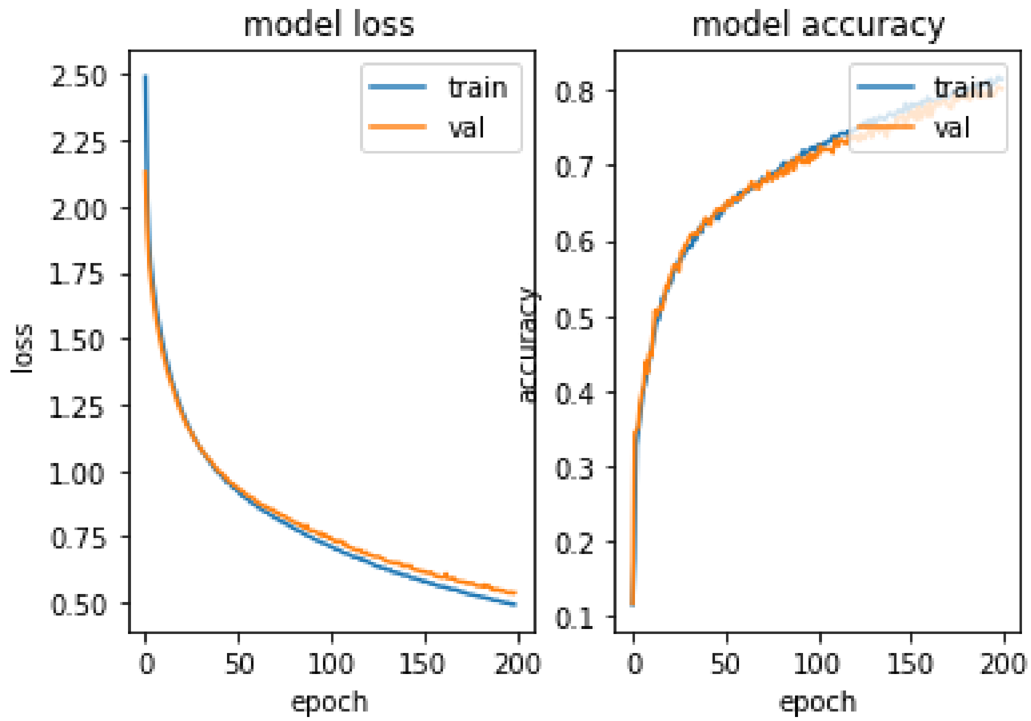
```
5564/5564 [==============================] - 0s 27us/step - loss: 0.5509 -
accuracy: 0.7861 - val_loss: 0.5909 - val_accuracy: 0.7836
Epoch 166/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5513 -
accuracy: 0.7885 - val_loss: 0.5916 - val_accuracy: 0.7793
Epoch 167/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5490 -
accuracy: 0.7944 - val_loss: 0.5943 - val_accuracy: 0.7815
Epoch 168/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5497 -
accuracy: 0.7865 - val_loss: 0.5852 - val_accuracy: 0.7843
Epoch 169/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5476 -
accuracy: 0.7858 - val_loss: 0.5813 - val_accuracy: 0.7865
Epoch 170/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5429 -
accuracy: 0.7931 - val_loss: 0.5801 - val_accuracy: 0.7843
Epoch 171/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5426 -
accuracy: 0.7901 - val_loss: 0.5791 - val_accuracy: 0.7786
Epoch 172/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5377 -
accuracy: 0.7922 - val_loss: 0.5781 - val_accuracy: 0.7764
Epoch 173/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5394 -
accuracy: 0.7948 - val_loss: 0.5785 - val_accuracy: 0.7757
Epoch 174/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5347 -
accuracy: 0.7989 - val_loss: 0.5727 - val_accuracy: 0.7915
Epoch 175/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5327 -
accuracy: 0.7973 - val_loss: 0.5754 - val_accuracy: 0.7937
Epoch 176/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5311 -
accuracy: 0.7994 - val_loss: 0.5716 - val_accuracy: 0.7850
Epoch 177/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5300 -
accuracy: 0.8005 - val_loss: 0.5740 - val_accuracy: 0.7915
Epoch 178/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5285 -
accuracy: 0.7971 - val_loss: 0.5711 - val_accuracy: 0.7908
Epoch 179/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5266 -
accuracy: 0.7983 - val_loss: 0.5657 - val_accuracy: 0.7894
Epoch 180/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5247 -
accuracy: 0.7998 - val_loss: 0.5632 - val_accuracy: 0.7930
Epoch 181/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.5245 -
accuracy: 0.7960 - val_loss: 0.5627 - val_accuracy: 0.7879
Epoch 182/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5203 -
accuracy: 0.8027 - val_loss: 0.5628 - val_accuracy: 0.7951
Epoch 183/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5207 -
accuracy: 0.7994 - val_loss: 0.5593 - val_accuracy: 0.7865
Epoch 184/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5195 -
accuracy: 0.8054 - val_loss: 0.5673 - val_accuracy: 0.8009
Epoch 185/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5167 -
accuracy: 0.8054 - val_loss: 0.5621 - val_accuracy: 0.8009
Epoch 186/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5157 -
accuracy: 0.8027 - val_loss: 0.5641 - val_accuracy: 0.7980
Epoch 187/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5142 -
accuracy: 0.8081 - val_loss: 0.5531 - val_accuracy: 0.7908
Epoch 188/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5114 -
accuracy: 0.8075 - val_loss: 0.5560 - val_accuracy: 0.7922
Epoch 189/200
5564/5564 [==============================] - 0s 26us/step - loss: 0.5078 -
accuracy: 0.8072 - val_loss: 0.5473 - val_accuracy: 0.7894
Epoch 190/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5090 -
accuracy: 0.8088 - val_loss: 0.5484 - val_accuracy: 0.7973
Epoch 191/200
5564/5564 [==============================] - 0s 26us/step - loss: 0.5080 -
accuracy: 0.8052 - val_loss: 0.5504 - val_accuracy: 0.7958
Epoch 192/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5046 -
accuracy: 0.8120 - val_loss: 0.5448 - val_accuracy: 0.7908
Epoch 193/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5025 -
accuracy: 0.8109 - val_loss: 0.5464 - val_accuracy: 0.7987
Epoch 194/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.5025 -
accuracy: 0.8145 - val_loss: 0.5451 - val_accuracy: 0.7915
Epoch 195/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.4986 -
accuracy: 0.8075 - val_loss: 0.5406 - val_accuracy: 0.7922
Epoch 196/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.4994 -
accuracy: 0.8093 - val_loss: 0.5393 - val_accuracy: 0.7930
Epoch 197/200
```

```
5564/5564 [==============================] - 0s 27us/step - loss: 0.4994 -
accuracy: 0.8122 - val_loss: 0.5371 - val_accuracy: 0.8001
Epoch 198/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.4934 -
accuracy: 0.8178 - val_loss: 0.5402 - val_accuracy: 0.8045
Epoch 199/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.4950 -
accuracy: 0.8152 - val_loss: 0.5349 - val_accuracy: 0.8030
Epoch 200/200
5564/5564 [==============================] - 0s 27us/step - loss: 0.4924 -
accuracy: 0.8145 - val_loss: 0.5376 - val_accuracy: 0.8016
```

[113]:

```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```



[114]:
```
score = model.evaluate(x=x_testcnn, y=y_test)
predictions = model.predict(x_testcnn)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
535/535 [==============================] - 0s 99us/step
Test loss: 0.5081647488558404
Test accuracy: 0.8355140089988708
```

[115]:
```
print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
 ↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support

           A       0.96      0.84      0.90       127
           E       0.90      0.65      0.76        81
           F       0.78      0.93      0.85        46
           L       0.83      0.87      0.85        69
           N       0.77      0.75      0.76        71
           T       0.82      1.00      0.90        62
           W       0.75      0.87      0.81        79

    accuracy                           0.84       535
```

```
   macro avg       0.83      0.85      0.83       535
weighted avg       0.84      0.84      0.83       535
```

6.0.14 save model

[ ]:
```
model_name = 'cnn_model_2.h5'
save_dir = os.path.join(os.getcwd(), 'models')
if not os.path.isdir(save_dir):
    os.makedirs(save_dir)

model_path = os.path.join(save_dir, model_name)
model.save(model_path)
print(f'Saved model at {model_path}')
```

simple one-layer NN

[116]:
```
model = Sequential()
model.add(Flatten(input_shape=(40,1)))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_6 (Flatten)          (None, 40)                0
_________________________________________________________________
dense_6 (Dense)              (None, 7)                 287
=================================================================
Total params: 287
Trainable params: 287
Non-trainable params: 0
_________________________________________________________________
```

[117]:
```
history = model.fit(x_traincnn, y_train, batch_size=512, epochs=200,␣
 ↪validation_data=(x_valcnn, y_val))
```

```
Train on 5564 samples, validate on 1391 samples
Epoch 1/200
5564/5564 [==============================] - 0s 21us/step - loss: 41.4142 -
accuracy: 0.0945 - val_loss: 39.4339 - val_accuracy: 0.0971
Epoch 2/200
5564/5564 [==============================] - 0s 5us/step - loss: 36.5222 -
accuracy: 0.1005 - val_loss: 34.9871 - val_accuracy: 0.1057
```

```
Epoch 3/200
5564/5564 [==============================] - 0s 5us/step - loss: 32.4792 -
accuracy: 0.1075 - val_loss: 31.3576 - val_accuracy: 0.1272
Epoch 4/200
5564/5564 [==============================] - 0s 5us/step - loss: 28.8408 -
accuracy: 0.1170 - val_loss: 27.5182 - val_accuracy: 0.1438
Epoch 5/200
5564/5564 [==============================] - 0s 5us/step - loss: 25.1255 -
accuracy: 0.1307 - val_loss: 23.6613 - val_accuracy: 0.1538
Epoch 6/200
5564/5564 [==============================] - 0s 5us/step - loss: 21.4846 -
accuracy: 0.1384 - val_loss: 19.8258 - val_accuracy: 0.1632
Epoch 7/200
5564/5564 [==============================] - 0s 5us/step - loss: 17.9175 -
accuracy: 0.1459 - val_loss: 16.2314 - val_accuracy: 0.1725
Epoch 8/200
5564/5564 [==============================] - 0s 5us/step - loss: 14.9040 -
accuracy: 0.1578 - val_loss: 13.7090 - val_accuracy: 0.1747
Epoch 9/200
5564/5564 [==============================] - 0s 5us/step - loss: 12.8272 -
accuracy: 0.1722 - val_loss: 11.8411 - val_accuracy: 0.2070
Epoch 10/200
5564/5564 [==============================] - 0s 5us/step - loss: 11.1607 -
accuracy: 0.1914 - val_loss: 10.5782 - val_accuracy: 0.2099
Epoch 11/200
5564/5564 [==============================] - 0s 5us/step - loss: 10.1014 -
accuracy: 0.2115 - val_loss: 9.7123 - val_accuracy: 0.2164
Epoch 12/200
5564/5564 [==============================] - 0s 5us/step - loss: 9.2580 -
accuracy: 0.2245 - val_loss: 8.8888 - val_accuracy: 0.2344
Epoch 13/200
5564/5564 [==============================] - 0s 5us/step - loss: 8.5164 -
accuracy: 0.2516 - val_loss: 8.1804 - val_accuracy: 0.2566
Epoch 14/200
5564/5564 [==============================] - 0s 5us/step - loss: 7.8643 -
accuracy: 0.2671 - val_loss: 7.5839 - val_accuracy: 0.2725
Epoch 15/200
5564/5564 [==============================] - 0s 5us/step - loss: 7.3001 -
accuracy: 0.2788 - val_loss: 7.0676 - val_accuracy: 0.2926
Epoch 16/200
5564/5564 [==============================] - 0s 5us/step - loss: 6.8002 -
accuracy: 0.2962 - val_loss: 6.5989 - val_accuracy: 0.2940
Epoch 17/200
5564/5564 [==============================] - 0s 5us/step - loss: 6.3676 -
accuracy: 0.3068 - val_loss: 6.1770 - val_accuracy: 0.3098
Epoch 18/200
5564/5564 [==============================] - 0s 5us/step - loss: 5.9611 -
accuracy: 0.3205 - val_loss: 5.8101 - val_accuracy: 0.3120
```

```
Epoch 19/200
5564/5564 [==============================] - 0s 5us/step - loss: 5.6053 -
accuracy: 0.3318 - val_loss: 5.4639 - val_accuracy: 0.3300
Epoch 20/200
5564/5564 [==============================] - 0s 5us/step - loss: 5.2667 -
accuracy: 0.3465 - val_loss: 5.1667 - val_accuracy: 0.3429
Epoch 21/200
5564/5564 [==============================] - 0s 5us/step - loss: 4.9692 -
accuracy: 0.3577 - val_loss: 4.8896 - val_accuracy: 0.3602
Epoch 22/200
5564/5564 [==============================] - 0s 5us/step - loss: 4.6959 -
accuracy: 0.3706 - val_loss: 4.6288 - val_accuracy: 0.3645
Epoch 23/200
5564/5564 [==============================] - 0s 5us/step - loss: 4.4450 -
accuracy: 0.3873 - val_loss: 4.4000 - val_accuracy: 0.3753
Epoch 24/200
5564/5564 [==============================] - 0s 5us/step - loss: 4.2093 -
accuracy: 0.3905 - val_loss: 4.1817 - val_accuracy: 0.3875
Epoch 25/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.9927 -
accuracy: 0.4037 - val_loss: 3.9759 - val_accuracy: 0.3932
Epoch 26/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.7986 -
accuracy: 0.4148 - val_loss: 3.8014 - val_accuracy: 0.4019
Epoch 27/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.6122 -
accuracy: 0.4240 - val_loss: 3.6185 - val_accuracy: 0.4062
Epoch 28/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.4444 -
accuracy: 0.4308 - val_loss: 3.4694 - val_accuracy: 0.4098
Epoch 29/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.2883 -
accuracy: 0.4382 - val_loss: 3.3241 - val_accuracy: 0.4177
Epoch 30/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.1440 -
accuracy: 0.4502 - val_loss: 3.1773 - val_accuracy: 0.4198
Epoch 31/200
5564/5564 [==============================] - 0s 5us/step - loss: 3.0122 -
accuracy: 0.4552 - val_loss: 3.0693 - val_accuracy: 0.4220
Epoch 32/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.8897 -
accuracy: 0.4648 - val_loss: 2.9470 - val_accuracy: 0.4306
Epoch 33/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.7768 -
accuracy: 0.4698 - val_loss: 2.8486 - val_accuracy: 0.4450
Epoch 34/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.6728 -
accuracy: 0.4829 - val_loss: 2.7496 - val_accuracy: 0.4551
```

```
Epoch 35/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.5774 -
accuracy: 0.4869 - val_loss: 2.6561 - val_accuracy: 0.4615
Epoch 36/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.4858 -
accuracy: 0.4942 - val_loss: 2.5775 - val_accuracy: 0.4666
Epoch 37/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.4047 -
accuracy: 0.5031 - val_loss: 2.5013 - val_accuracy: 0.4702
Epoch 38/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.3289 -
accuracy: 0.5058 - val_loss: 2.4202 - val_accuracy: 0.4802
Epoch 39/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.2568 -
accuracy: 0.5176 - val_loss: 2.3551 - val_accuracy: 0.4845
Epoch 40/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.1891 -
accuracy: 0.5187 - val_loss: 2.2823 - val_accuracy: 0.4874
Epoch 41/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.1203 -
accuracy: 0.5318 - val_loss: 2.2352 - val_accuracy: 0.4925
Epoch 42/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.0619 -
accuracy: 0.5338 - val_loss: 2.1578 - val_accuracy: 0.4946
Epoch 43/200
5564/5564 [==============================] - 0s 5us/step - loss: 2.0045 -
accuracy: 0.5372 - val_loss: 2.1074 - val_accuracy: 0.5047
Epoch 44/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.9504 -
accuracy: 0.5446 - val_loss: 2.0459 - val_accuracy: 0.5140
Epoch 45/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.8958 -
accuracy: 0.5485 - val_loss: 2.0019 - val_accuracy: 0.5234
Epoch 46/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.8474 -
accuracy: 0.5554 - val_loss: 1.9484 - val_accuracy: 0.5298
Epoch 47/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.8024 -
accuracy: 0.5532 - val_loss: 1.9031 - val_accuracy: 0.5306
Epoch 48/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.7575 -
accuracy: 0.5652 - val_loss: 1.8599 - val_accuracy: 0.5377
Epoch 49/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.7145 -
accuracy: 0.5629 - val_loss: 1.8179 - val_accuracy: 0.5349
Epoch 50/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.6755 -
accuracy: 0.5767 - val_loss: 1.7728 - val_accuracy: 0.5442
```

```
Epoch 51/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.6370 -
accuracy: 0.5784 - val_loss: 1.7414 - val_accuracy: 0.5485
Epoch 52/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.6036 -
accuracy: 0.5827 - val_loss: 1.6973 - val_accuracy: 0.5514
Epoch 53/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.5682 -
accuracy: 0.5864 - val_loss: 1.6693 - val_accuracy: 0.5507
Epoch 54/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.5337 -
accuracy: 0.5877 - val_loss: 1.6284 - val_accuracy: 0.5593
Epoch 55/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.5023 -
accuracy: 0.5945 - val_loss: 1.6035 - val_accuracy: 0.5694
Epoch 56/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.4807 -
accuracy: 0.5929 - val_loss: 1.5725 - val_accuracy: 0.5615
Epoch 57/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.4478 -
accuracy: 0.5953 - val_loss: 1.5438 - val_accuracy: 0.5687
Epoch 58/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.4232 -
accuracy: 0.6017 - val_loss: 1.5117 - val_accuracy: 0.5730
Epoch 59/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.3968 -
accuracy: 0.6044 - val_loss: 1.4829 - val_accuracy: 0.5809
Epoch 60/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.3707 -
accuracy: 0.6066 - val_loss: 1.4602 - val_accuracy: 0.5845
Epoch 61/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.3512 -
accuracy: 0.6086 - val_loss: 1.4350 - val_accuracy: 0.5830
Epoch 62/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.3264 -
accuracy: 0.6130 - val_loss: 1.4098 - val_accuracy: 0.5881
Epoch 63/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.3065 -
accuracy: 0.6188 - val_loss: 1.3934 - val_accuracy: 0.5881
Epoch 64/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.2842 -
accuracy: 0.6190 - val_loss: 1.3662 - val_accuracy: 0.5931
Epoch 65/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.2661 -
accuracy: 0.6201 - val_loss: 1.3487 - val_accuracy: 0.6010
Epoch 66/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.2467 -
accuracy: 0.6231 - val_loss: 1.3253 - val_accuracy: 0.6017
```

```
Epoch 67/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.2281 -
accuracy: 0.6289 - val_loss: 1.3150 - val_accuracy: 0.6082
Epoch 68/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.2115 -
accuracy: 0.6316 - val_loss: 1.2924 - val_accuracy: 0.6053
Epoch 69/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1969 -
accuracy: 0.6317 - val_loss: 1.2738 - val_accuracy: 0.6096
Epoch 70/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1849 -
accuracy: 0.6350 - val_loss: 1.2691 - val_accuracy: 0.6111
Epoch 71/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1686 -
accuracy: 0.6355 - val_loss: 1.2407 - val_accuracy: 0.6233
Epoch 72/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1510 -
accuracy: 0.6400 - val_loss: 1.2315 - val_accuracy: 0.6175
Epoch 73/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1350 -
accuracy: 0.6391 - val_loss: 1.2080 - val_accuracy: 0.6254
Epoch 74/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1232 -
accuracy: 0.6450 - val_loss: 1.1976 - val_accuracy: 0.6298
Epoch 75/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.1083 -
accuracy: 0.6398 - val_loss: 1.1873 - val_accuracy: 0.6219
Epoch 76/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0948 -
accuracy: 0.6488 - val_loss: 1.1674 - val_accuracy: 0.6370
Epoch 77/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0835 -
accuracy: 0.6512 - val_loss: 1.1629 - val_accuracy: 0.6312
Epoch 78/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0753 -
accuracy: 0.6476 - val_loss: 1.1477 - val_accuracy: 0.6290
Epoch 79/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0692 -
accuracy: 0.6546 - val_loss: 1.1366 - val_accuracy: 0.6254
Epoch 80/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0502 -
accuracy: 0.6515 - val_loss: 1.1308 - val_accuracy: 0.6362
Epoch 81/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0372 -
accuracy: 0.6547 - val_loss: 1.1095 - val_accuracy: 0.6485
Epoch 82/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0256 -
accuracy: 0.6573 - val_loss: 1.1118 - val_accuracy: 0.6413
```

```
Epoch 83/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0183 -
accuracy: 0.6573 - val_loss: 1.0912 - val_accuracy: 0.6492
Epoch 84/200
5564/5564 [==============================] - 0s 5us/step - loss: 1.0066 -
accuracy: 0.6601 - val_loss: 1.0821 - val_accuracy: 0.6477
Epoch 85/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9984 -
accuracy: 0.6578 - val_loss: 1.0798 - val_accuracy: 0.6578
Epoch 86/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9890 -
accuracy: 0.6639 - val_loss: 1.0622 - val_accuracy: 0.6520
Epoch 87/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9778 -
accuracy: 0.6650 - val_loss: 1.0587 - val_accuracy: 0.6441
Epoch 88/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9702 -
accuracy: 0.6655 - val_loss: 1.0467 - val_accuracy: 0.6592
Epoch 89/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9616 -
accuracy: 0.6677 - val_loss: 1.0390 - val_accuracy: 0.6571
Epoch 90/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9535 -
accuracy: 0.6691 - val_loss: 1.0288 - val_accuracy: 0.6506
Epoch 91/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9454 -
accuracy: 0.6713 - val_loss: 1.0232 - val_accuracy: 0.6542
Epoch 92/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9385 -
accuracy: 0.6689 - val_loss: 1.0163 - val_accuracy: 0.6686
Epoch 93/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9338 -
accuracy: 0.6693 - val_loss: 1.0141 - val_accuracy: 0.6650
Epoch 94/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9288 -
accuracy: 0.6725 - val_loss: 1.0093 - val_accuracy: 0.6535
Epoch 95/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9175 -
accuracy: 0.6706 - val_loss: 0.9923 - val_accuracy: 0.6700
Epoch 96/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9093 -
accuracy: 0.6767 - val_loss: 0.9895 - val_accuracy: 0.6628
Epoch 97/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.9019 -
accuracy: 0.6774 - val_loss: 0.9815 - val_accuracy: 0.6628
Epoch 98/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8955 -
accuracy: 0.6763 - val_loss: 0.9720 - val_accuracy: 0.6643
```

```
Epoch 99/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8912 -
accuracy: 0.6783 - val_loss: 0.9697 - val_accuracy: 0.6693
Epoch 100/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8836 -
accuracy: 0.6808 - val_loss: 0.9615 - val_accuracy: 0.6679
Epoch 101/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8783 -
accuracy: 0.6806 - val_loss: 0.9574 - val_accuracy: 0.6686
Epoch 102/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8723 -
accuracy: 0.6833 - val_loss: 0.9553 - val_accuracy: 0.6542
Epoch 103/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8654 -
accuracy: 0.6837 - val_loss: 0.9433 - val_accuracy: 0.6722
Epoch 104/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8609 -
accuracy: 0.6833 - val_loss: 0.9436 - val_accuracy: 0.6585
Epoch 105/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8560 -
accuracy: 0.6857 - val_loss: 0.9364 - val_accuracy: 0.6700
Epoch 106/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8514 -
accuracy: 0.6851 - val_loss: 0.9297 - val_accuracy: 0.6549
Epoch 107/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8485 -
accuracy: 0.6855 - val_loss: 0.9238 - val_accuracy: 0.6643
Epoch 108/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8415 -
accuracy: 0.6893 - val_loss: 0.9252 - val_accuracy: 0.6542
Epoch 109/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8368 -
accuracy: 0.6873 - val_loss: 0.9162 - val_accuracy: 0.6556
Epoch 110/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8315 -
accuracy: 0.6891 - val_loss: 0.9133 - val_accuracy: 0.6679
Epoch 111/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8276 -
accuracy: 0.6875 - val_loss: 0.9158 - val_accuracy: 0.6585
Epoch 112/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8297 -
accuracy: 0.6885 - val_loss: 0.9195 - val_accuracy: 0.6492
Epoch 113/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8226 -
accuracy: 0.6914 - val_loss: 0.9002 - val_accuracy: 0.6513
Epoch 114/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8158 -
accuracy: 0.6891 - val_loss: 0.8937 - val_accuracy: 0.6700
```

```
Epoch 115/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8117 -
accuracy: 0.6921 - val_loss: 0.8930 - val_accuracy: 0.6592
Epoch 116/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8108 -
accuracy: 0.6939 - val_loss: 0.8989 - val_accuracy: 0.6520
Epoch 117/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8055 -
accuracy: 0.6939 - val_loss: 0.8802 - val_accuracy: 0.6707
Epoch 118/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.8014 -
accuracy: 0.6914 - val_loss: 0.8803 - val_accuracy: 0.6607
Epoch 119/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7966 -
accuracy: 0.6936 - val_loss: 0.8771 - val_accuracy: 0.6636
Epoch 120/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7927 -
accuracy: 0.6954 - val_loss: 0.8723 - val_accuracy: 0.6722
Epoch 121/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7890 -
accuracy: 0.6964 - val_loss: 0.8697 - val_accuracy: 0.6657
Epoch 122/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7891 -
accuracy: 0.6979 - val_loss: 0.8786 - val_accuracy: 0.6592
Epoch 123/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7877 -
accuracy: 0.6950 - val_loss: 0.8697 - val_accuracy: 0.6671
Epoch 124/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7843 -
accuracy: 0.6955 - val_loss: 0.8656 - val_accuracy: 0.6650
Epoch 125/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7782 -
accuracy: 0.6973 - val_loss: 0.8588 - val_accuracy: 0.6693
Epoch 126/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7802 -
accuracy: 0.6964 - val_loss: 0.8528 - val_accuracy: 0.6786
Epoch 127/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7735 -
accuracy: 0.6999 - val_loss: 0.8513 - val_accuracy: 0.6650
Epoch 128/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7706 -
accuracy: 0.6970 - val_loss: 0.8493 - val_accuracy: 0.6679
Epoch 129/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7684 -
accuracy: 0.7017 - val_loss: 0.8445 - val_accuracy: 0.6664
Epoch 130/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7649 -
accuracy: 0.6995 - val_loss: 0.8466 - val_accuracy: 0.6628
```

```
Epoch 131/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7621 -
accuracy: 0.6999 - val_loss: 0.8470 - val_accuracy: 0.6664
Epoch 132/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7611 -
accuracy: 0.7013 - val_loss: 0.8395 - val_accuracy: 0.6722
Epoch 133/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7596 -
accuracy: 0.7031 - val_loss: 0.8471 - val_accuracy: 0.6650
Epoch 134/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7565 -
accuracy: 0.7035 - val_loss: 0.8332 - val_accuracy: 0.6765
Epoch 135/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7526 -
accuracy: 0.7000 - val_loss: 0.8355 - val_accuracy: 0.6657
Epoch 136/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7530 -
accuracy: 0.7035 - val_loss: 0.8392 - val_accuracy: 0.6679
Epoch 137/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7523 -
accuracy: 0.7020 - val_loss: 0.8317 - val_accuracy: 0.6729
Epoch 138/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7467 -
accuracy: 0.7043 - val_loss: 0.8327 - val_accuracy: 0.6643
Epoch 139/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7462 -
accuracy: 0.7027 - val_loss: 0.8410 - val_accuracy: 0.6535
Epoch 140/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7467 -
accuracy: 0.7004 - val_loss: 0.8267 - val_accuracy: 0.6643
Epoch 141/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7457 -
accuracy: 0.7029 - val_loss: 0.8265 - val_accuracy: 0.6700
Epoch 142/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7388 -
accuracy: 0.7065 - val_loss: 0.8216 - val_accuracy: 0.6628
Epoch 143/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7384 -
accuracy: 0.7051 - val_loss: 0.8124 - val_accuracy: 0.6743
Epoch 144/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7398 -
accuracy: 0.7008 - val_loss: 0.8154 - val_accuracy: 0.6758
Epoch 145/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7364 -
accuracy: 0.7011 - val_loss: 0.8146 - val_accuracy: 0.6715
Epoch 146/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7346 -
accuracy: 0.7035 - val_loss: 0.8119 - val_accuracy: 0.6707
```

```
Epoch 147/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7303 -
accuracy: 0.7058 - val_loss: 0.8168 - val_accuracy: 0.6643
Epoch 148/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7311 -
accuracy: 0.7052 - val_loss: 0.8092 - val_accuracy: 0.6700
Epoch 149/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7298 -
accuracy: 0.7054 - val_loss: 0.8063 - val_accuracy: 0.6729
Epoch 150/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7286 -
accuracy: 0.7090 - val_loss: 0.8085 - val_accuracy: 0.6636
Epoch 151/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7277 -
accuracy: 0.7035 - val_loss: 0.8080 - val_accuracy: 0.6715
Epoch 152/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7253 -
accuracy: 0.7090 - val_loss: 0.8062 - val_accuracy: 0.6643
Epoch 153/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7228 -
accuracy: 0.7054 - val_loss: 0.8010 - val_accuracy: 0.6786
Epoch 154/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7239 -
accuracy: 0.7056 - val_loss: 0.7995 - val_accuracy: 0.6751
Epoch 155/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7195 -
accuracy: 0.7096 - val_loss: 0.7936 - val_accuracy: 0.6808
Epoch 156/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7202 -
accuracy: 0.7083 - val_loss: 0.7930 - val_accuracy: 0.6808
Epoch 157/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7178 -
accuracy: 0.7076 - val_loss: 0.7999 - val_accuracy: 0.6794
Epoch 158/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7200 -
accuracy: 0.7076 - val_loss: 0.7916 - val_accuracy: 0.6858
Epoch 159/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7224 -
accuracy: 0.7110 - val_loss: 0.7879 - val_accuracy: 0.6765
Epoch 160/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7232 -
accuracy: 0.7063 - val_loss: 0.7966 - val_accuracy: 0.6822
Epoch 161/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7157 -
accuracy: 0.7114 - val_loss: 0.7942 - val_accuracy: 0.6671
Epoch 162/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7161 -
accuracy: 0.7081 - val_loss: 0.7908 - val_accuracy: 0.6751
```

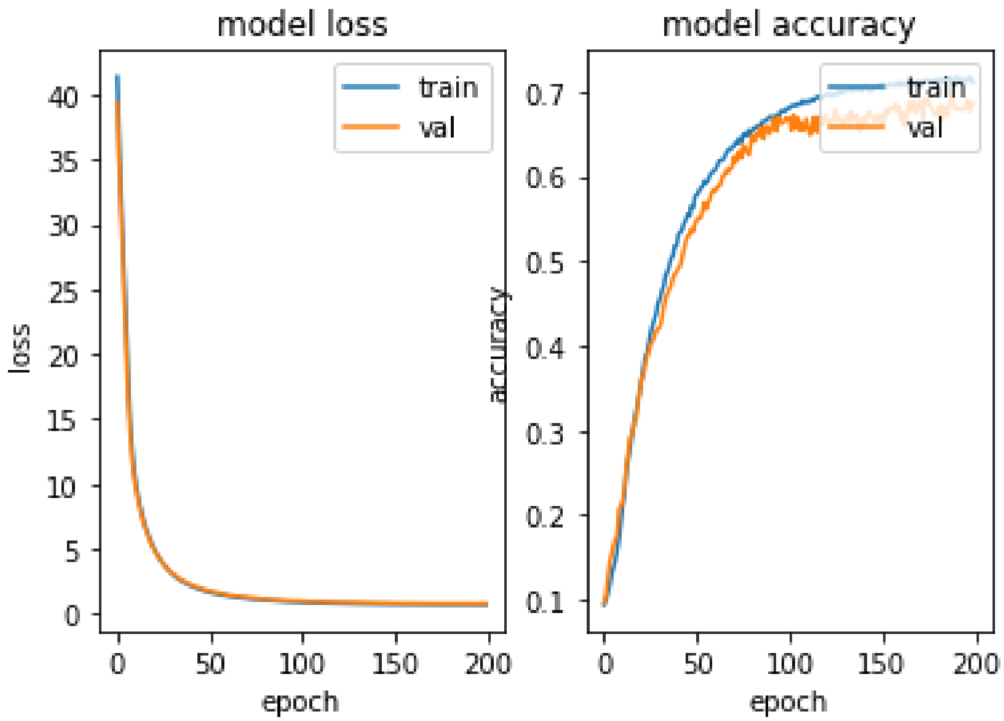
```
Epoch 163/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7148 -
accuracy: 0.7081 - val_loss: 0.8019 - val_accuracy: 0.6592
Epoch 164/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7150 -
accuracy: 0.7096 - val_loss: 0.7962 - val_accuracy: 0.6657
Epoch 165/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7106 -
accuracy: 0.7130 - val_loss: 0.7893 - val_accuracy: 0.6873
Epoch 166/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7130 -
accuracy: 0.7097 - val_loss: 0.7898 - val_accuracy: 0.6772
Epoch 167/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7140 -
accuracy: 0.7070 - val_loss: 0.7836 - val_accuracy: 0.6887
Epoch 168/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7087 -
accuracy: 0.7081 - val_loss: 0.7836 - val_accuracy: 0.6887
Epoch 169/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7088 -
accuracy: 0.7090 - val_loss: 0.7879 - val_accuracy: 0.6743
Epoch 170/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7067 -
accuracy: 0.7132 - val_loss: 0.7772 - val_accuracy: 0.6794
Epoch 171/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7068 -
accuracy: 0.7097 - val_loss: 0.7802 - val_accuracy: 0.6894
Epoch 172/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7049 -
accuracy: 0.7060 - val_loss: 0.7831 - val_accuracy: 0.6830
Epoch 173/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7062 -
accuracy: 0.7135 - val_loss: 0.7874 - val_accuracy: 0.6671
Epoch 174/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7043 -
accuracy: 0.7114 - val_loss: 0.7741 - val_accuracy: 0.6930
Epoch 175/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7031 -
accuracy: 0.7139 - val_loss: 0.7810 - val_accuracy: 0.6873
Epoch 176/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7020 -
accuracy: 0.7108 - val_loss: 0.7708 - val_accuracy: 0.6887
Epoch 177/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7003 -
accuracy: 0.7130 - val_loss: 0.7736 - val_accuracy: 0.6822
Epoch 178/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7004 -
accuracy: 0.7128 - val_loss: 0.7776 - val_accuracy: 0.6794
```

```
Epoch 179/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7011 -
accuracy: 0.7132 - val_loss: 0.7792 - val_accuracy: 0.6736
Epoch 180/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7004 -
accuracy: 0.7148 - val_loss: 0.7837 - val_accuracy: 0.6729
Epoch 181/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7045 -
accuracy: 0.7099 - val_loss: 0.7771 - val_accuracy: 0.6722
Epoch 182/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7009 -
accuracy: 0.7106 - val_loss: 0.7720 - val_accuracy: 0.6837
Epoch 183/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6986 -
accuracy: 0.7159 - val_loss: 0.7741 - val_accuracy: 0.6664
Epoch 184/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6968 -
accuracy: 0.7115 - val_loss: 0.7727 - val_accuracy: 0.6837
Epoch 185/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6957 -
accuracy: 0.7121 - val_loss: 0.7736 - val_accuracy: 0.6729
Epoch 186/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6957 -
accuracy: 0.7164 - val_loss: 0.7729 - val_accuracy: 0.6715
Epoch 187/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6967 -
accuracy: 0.7155 - val_loss: 0.7686 - val_accuracy: 0.6822
Epoch 188/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6958 -
accuracy: 0.7169 - val_loss: 0.7726 - val_accuracy: 0.6916
Epoch 189/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7005 -
accuracy: 0.7092 - val_loss: 0.7776 - val_accuracy: 0.6930
Epoch 190/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7000 -
accuracy: 0.7085 - val_loss: 0.7711 - val_accuracy: 0.6837
Epoch 191/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6965 -
accuracy: 0.7178 - val_loss: 0.7736 - val_accuracy: 0.6794
Epoch 192/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6950 -
accuracy: 0.7133 - val_loss: 0.7650 - val_accuracy: 0.6801
Epoch 193/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6977 -
accuracy: 0.7133 - val_loss: 0.7649 - val_accuracy: 0.6808
Epoch 194/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6979 -
accuracy: 0.7153 - val_loss: 0.7739 - val_accuracy: 0.6779
```

```
Epoch 195/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6967 -
accuracy: 0.7137 - val_loss: 0.7627 - val_accuracy: 0.6786
Epoch 196/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6923 -
accuracy: 0.7126 - val_loss: 0.7714 - val_accuracy: 0.6808
Epoch 197/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6960 -
accuracy: 0.7130 - val_loss: 0.7640 - val_accuracy: 0.6902
Epoch 198/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6966 -
accuracy: 0.7178 - val_loss: 0.7889 - val_accuracy: 0.6772
Epoch 199/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.7000 -
accuracy: 0.7157 - val_loss: 0.7633 - val_accuracy: 0.6808
Epoch 200/200
5564/5564 [==============================] - 0s 5us/step - loss: 0.6932 -
accuracy: 0.7108 - val_loss: 0.7681 - val_accuracy: 0.6866
```

[118]:

```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```



[119]:
```
score = model.evaluate(x=x_testcnn, y=y_test)
predictions = model.predict(x_testcnn)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
535/535 [==============================] - 0s 42us/step
Test loss: 0.6090983534527716
Test accuracy: 0.745794415473938
```

[120]:
```
print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
 ↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support

           A       0.89      0.88      0.89       127
           E       0.72      0.48      0.58        81
           F       0.77      0.74      0.76        46
           L       0.64      0.75      0.69        69
           N       0.74      0.61      0.67        71
           T       0.73      0.98      0.84        62
           W       0.65      0.73      0.69        79

    accuracy                           0.75       535
```

```
   macro avg       0.74      0.74      0.73       535
weighted avg       0.75      0.75      0.74       535
```

6.0.15 CNN -Conv2D

Here we compute the mfccs and retain the 2D format (instead of taking the mean of each mffc feature).

[121]:
```
def plot_signals(signals):
    fig, axes = plt.subplots(nrows=2, ncols=4, sharex=False,
                             sharey=True, figsize=(20,5))
    axes = axes.ravel()
    fig.suptitle('Time Series', size=16)

    for i in range(len(signals)):
        axes[i].set_title(list(signals.keys())[i])
        axes[i].plot(list(signals.values())[i])
        axes[i].get_xaxis().set_visible(False)
        axes[i].get_yaxis().set_visible(False)
    fig.delaxes(axes[-1])

def plot_fft(fft):
    fig, axes = plt.subplots(nrows=2, ncols=4, sharex=False,
                             sharey=True, figsize=(20,5))
    fig.suptitle('Fourier Transforms', size=16)
    axes = axes.ravel()

    for i in range(len(fft)):
        data = list(fft.values())[i]
        Y, freq = data[0], data[1]
        axes[i].set_title(list(fft.keys())[i])
        axes[i].plot(freq, Y)
        axes[i].get_xaxis().set_visible(False)
        axes[i].get_yaxis().set_visible(False)
    fig.delaxes(axes[-1])

def plot_fbank(fbank):
    fig, axes = plt.subplots(nrows=2, ncols=4, sharex=False,
                             sharey=True, figsize=(20,5))
    fig.suptitle('Filter Bank Coefficients', size=16)
    axes = axes.ravel()

    for i in range(len(fbank)):
        axes[i].set_title(list(fbank.keys())[i])
        axes[i].imshow(list(fbank.values())[i],
                       cmap='hot', interpolation='nearest')
```

```
        axes[i].get_xaxis().set_visible(False)
        axes[i].get_yaxis().set_visible(False)
    fig.delaxes(axes[-1])

def plot_mfccs(mfccs):
    fig, axes = plt.subplots(nrows=2, ncols=4, sharex=False,
                             sharey=True, figsize=(20,5))
    fig.suptitle('Mel Frequency Cepstrum Coefficients', size=16)
    axes = axes.ravel()

    for i in range(len(mfccs)):
        axes[i].set_title(list(mfccs.keys())[i])
        axes[i].imshow(list(mfccs.values())[i],
                cmap='hot', interpolation='nearest')
        axes[i].get_xaxis().set_visible(False)
        axes[i].get_yaxis().set_visible(False)
    fig.delaxes(axes[-1])
```

[122]:
```
def calc_fft(y, rate):
    n = len(y)
    freq = np.fft.rfftfreq(n, d=1/rate)
    Y = abs(np.fft.rfft(y)/n)
    return (Y, freq)
```

[123]:
```
df = df.drop('filename', axis=1)
df.reset_index(inplace=True)
```

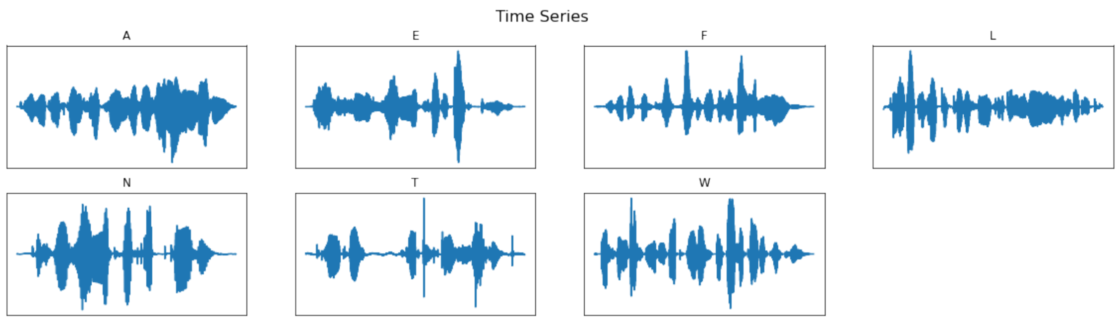
[124]:
```
df.set_index(df.filename, inplace=True)
for file in df.index:
    rate, signal = wavfile.read(f'{path}/{file}')
    df.at[file, 'length'] = signal.shape[0] / rate
```

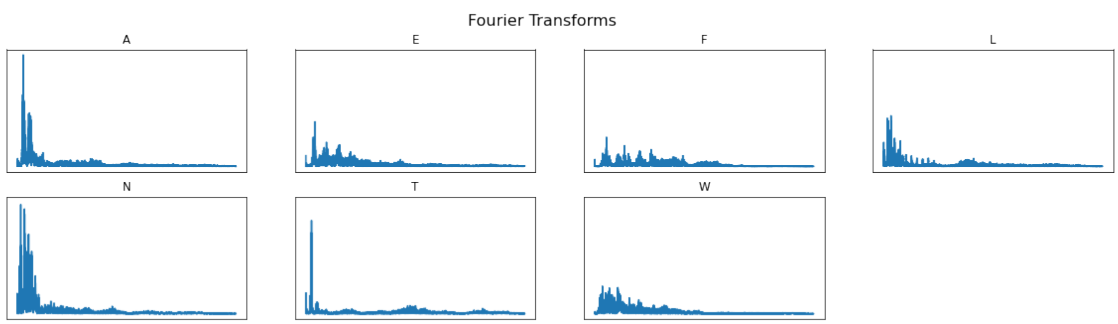
```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: WavFileWarning:
Chunk (non-data) not understood, skipping it.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

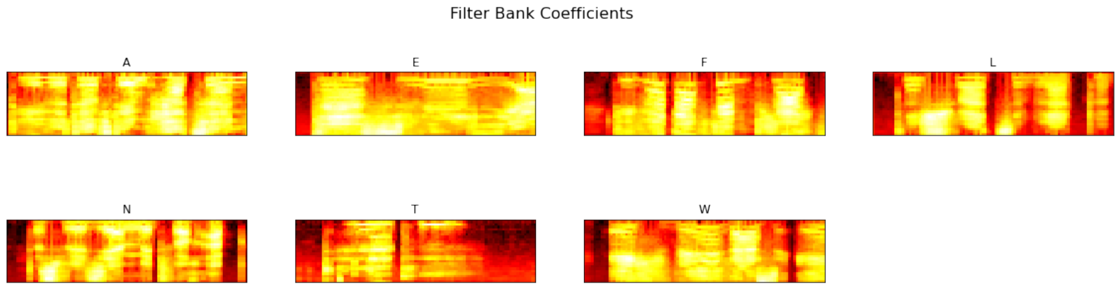
[125]:
```
signals = {}
fft = {}
fbank = {}
mfccs = {}

for c in classes:
    wav_file = df[df.emotion == c]['filename'].iloc[0]
    signal, rate = librosa.load(f'{path}/{wav_file}', sr=16000)
    signals[c] = signal
    fft[c] = calc_fft(signal, rate)
```

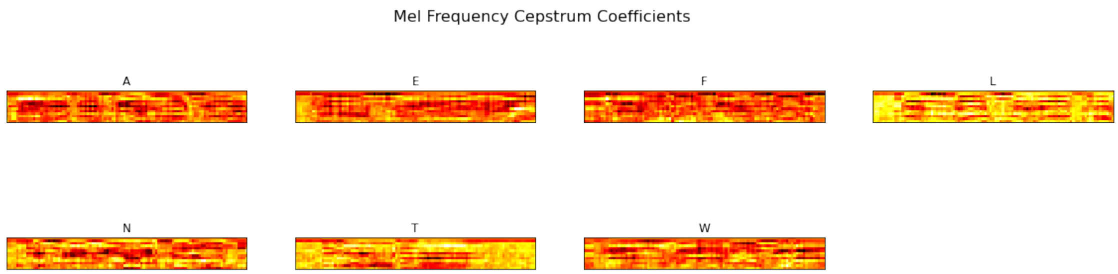
```
    # only take one second of signal
    # 16000/40=400 : since we take 25ms windows, 1s/40=0.025s
    bank = logfbank(signal[:rate], rate, nfilt=26, nfft=400).T
    fbank[c] = bank

    # numcep: nb of cepstrals we keep after DCT => throw away 50%
    mel = mfcc(signal[:rate], rate, numcep=13, nfilt=26, nfft=400).T
    mfccs[c] = mel
```

[126]:
```
plot_signals(signals)
plt.show()

plot_fft(fft)
plt.show()

plot_fbank(fbank)
plt.show()

plot_mfccs(mfccs)
plt.show()
```









With this approach, we do the same as before (compute the mfccs) but we plot the FFTs, Filterbank energies and DCTs.

- Can do noise threshold detection (Filtering): remove parts of audio with no sound => to keep only characteristic signal of each class
- FFT: can distinguish differences but not optimal
- Filterbank energies: patterns emerge (temporal relationships)
- DCT: plots are a lot more unique

6.0.16 Modelling

Randomly sample 1s chunks out of audio => don't take entire signal sequence

[127]:
```
n_samples = 2 * int(df['length'].sum() / 0.1)
n_samples
```

[127]: 29740

[128]:
```
df.set_index('filename', inplace=True)
```

[152]:
```
def build_rand_feat():
    tmp = check_data()
    if tmp:
        return tmp.data[0], tmp.data[1]

```

```
    X = []
    y = []
    _min, _max = float('inf'), -float('inf')

    for _ in tqdm(range(n_samples)):
        rand_class = np.random.choice(emotion_count_by_class_pct.index,␣
↪p=emotion_count_by_class_pct)
        file = np.random.choice(df[df.emotion == rand_class].index)
        rate, wav = wavfile.read(f'../data/wav/{file}')
        label = df.at[file, 'emotion']
        rand_index = np.random.randint(0, wav.shape[0]-config.step)
        sample = wav[rand_index:rand_index+config.step]
        X_sample = mfcc(sample, rate, numcep=config.nfeat, nfilt=config.nfilt,␣
↪nfft=config.nfft)
        _min = min(np.amin(X_sample), _min)
        _max = max(np.amax(X_sample), _max)
        X.append(X_sample)
        y.append((np.where(classes == label))[0][0])

    config.min = _min
    config.max = _max

    X, y = np.array(X), np.array(y)
    X = (X - _min) / (_max - _min)

    if config.mode == 'conv':
        X = X.reshape(X.shape[0], X.shape[1], X.shape[2], 1)
    elif config.mode == 'time':
        X = X.reshape(X.shape[0], X.shape[1], X.shape[2])

    y = to_categorical(y, num_classes=7)
    config.data = (X, y)

    with open(config.p_path, 'wb') as handle:
        # for compatibility with python 2: protocol=2
        pickle.dump(config, handle, protocol=2)

    return X, y
```

[147]:
```
def check_data():
    if os.path.isfile(config.p_path):
        print(f'Loading existing data for {config.mode} model')
        with open(config.p_path, 'rb') as handle:
            tmp = pickle.load(handle)
            return tmp
    else:
        return None
```

[148]:
```
# only pool down once because X is not high dimensional
# should try with batch norm
def get_conv_model():
    model = Sequential()
    model.add(Conv2D(16, (3, 3), activation='relu', strides=(1, 1),␣
 ↪padding='same', input_shape=input_shape))
    model.add(Conv2D(32, (3, 3), activation='relu', strides=(1, 1),␣
 ↪padding='same'))
    model.add(Conv2D(64, (3, 3), activation='relu', strides=(1, 1),␣
 ↪padding='same'))
    model.add(Conv2D(128, (3, 3), activation='relu', strides=(1, 1),␣
 ↪padding='same'))
    model.add(MaxPool2D((2,2)))
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))

    model.summary()
    model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])

    return model
```

[149]:
```
def get_recurrent_model():
    # shape of input X for RNN: (n, time, feat)
    model = Sequential()
    model.add(LSTM(128, return_sequences=True, input_shape=input_shape))
    model.add(LSTM(128, return_sequences=True))
    model.add(Dropout(0.5))
    model.add(TimeDistributed(Dense(64, activation='relu')))
    model.add(TimeDistributed(Dense(32, activation='relu')))
    model.add(TimeDistributed(Dense(16, activation='relu')))
    model.add(TimeDistributed(Dense(8, activation='relu')))
    model.add(Flatten())
    model.add(Dense(num_classes, activation='softmax'))

    model.summary()
    model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])

    return model
```

[182]:
```
config = Config(mode='conv')
```

[191]:
```
np.random.seed(42)

if config.mode == 'conv':
    X, y = build_rand_feat()
    y_flat = np.argmax(y, axis=1)
    input_shape = (X.shape[1], X.shape[2], 1)
    model = get_conv_model()
elif config.mode == 'lstm':
    X, y = build_rand_feat()
    y_flat = np.argmax(y, axis=1)
    input_shape = (X.shape[1], X.shape[2])
    model = get_recurrent_model()
```

```
Loading existing data for lstm model
Model: "sequential_17"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_5 (LSTM)                (None, 9, 128)            72704
_________________________________________________________________
lstm_6 (LSTM)                (None, 9, 128)            131584
_________________________________________________________________
dropout_16 (Dropout)         (None, 9, 128)            0
_________________________________________________________________
time_distributed_9 (TimeDist (None, 9, 64)             8256
_________________________________________________________________
time_distributed_10 (TimeDis (None, 9, 32)             2080
_________________________________________________________________
time_distributed_11 (TimeDis (None, 9, 16)             528
_________________________________________________________________
time_distributed_12 (TimeDis (None, 9, 8)              136
_________________________________________________________________
flatten_17 (Flatten)         (None, 72)                0
_________________________________________________________________
dense_45 (Dense)             (None, 7)                 511
=================================================================
Total params: 215,799
Trainable params: 215,799
Non-trainable params: 0
_________________________________________________________________
```

train/test/val

This approach will not include data augmentation due to lack of time for implementing it. Instead, 1s samples are sampled at random through all the raw data to build the sets.

[192]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1,␣
↪random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size =␣
↪0.2, random_state=43)
```

[193]:
```
print(X_train.shape, X_test.shape, X_val.shape)
print(y_train.shape, y_test.shape, y_val.shape)
```

```
(42825, 9, 13) (5948, 9, 13) (10707, 9, 13)
(42825, 7) (5948, 7) (10707, 7)
```

6.0.17 Create class weights

Weight matrix update will put more emphasis on gradient steps for under-represented classes

[156]:
```
class_weight = compute_class_weight('balanced', np.unique(y_flat), y_flat)
class_weight
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/utils/validation.py:71:
FutureWarning: Pass classes=[0 1 2 3 4 5 6], y=[3 1 3 … 0 0 2] as keyword
args. From version 0.25 passing these as positional arguments will result in an
error
  FutureWarning)
```

[156]:
```
array([1.0791393 , 1.65636313, 1.0916165 , 0.93953371, 0.97310385,
       1.22296241, 0.60832924])
```

6.0.18 Train

[ ]:
```
checkpoint = ModelCheckpoint('../models/run1/model.{epoch:02d}-{accuracy:.
↪4f}-{val_accuracy:.4f}-{loss:.4f}-{val_loss:.4f}.h5', monitor='val_loss',␣
↪verbose=1, mode='min', save_best_only=True,
                             save_weights_only=False, period=1)
```

[186]:
```
#history = model.fit(X_train, y_train, epochs=250, batch_size=512,␣
↪shuffle=True, validation_data=(X_val, y_val),
#          callbacks=[checkpoint], class_weight=class_weight)

history = model.fit(X_train, y_train, epochs=60, batch_size=512, shuffle=True,␣
↪validation_data=(X_val, y_val),
                    class_weight=class_weight)
```

```
Train on 21412 samples, validate on 5354 samples
Epoch 1/60
21412/21412 [==============================] - 1s 44us/step - loss: 1.8760 -
accuracy: 0.2408 - val_loss: 1.8047 - val_accuracy: 0.2682
```

```
Epoch 2/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.7063 -
accuracy: 0.3237 - val_loss: 1.6560 - val_accuracy: 0.3496
Epoch 3/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.6068 -
accuracy: 0.3721 - val_loss: 1.5780 - val_accuracy: 0.3904
Epoch 4/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.5636 -
accuracy: 0.3874 - val_loss: 1.5764 - val_accuracy: 0.3928
Epoch 5/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.5203 -
accuracy: 0.4031 - val_loss: 1.5096 - val_accuracy: 0.4227
Epoch 6/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.4955 -
accuracy: 0.4172 - val_loss: 1.4701 - val_accuracy: 0.4296
Epoch 7/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.4654 -
accuracy: 0.4250 - val_loss: 1.4741 - val_accuracy: 0.4193
Epoch 8/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.4353 -
accuracy: 0.4404 - val_loss: 1.4456 - val_accuracy: 0.4365
Epoch 9/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.4131 -
accuracy: 0.4516 - val_loss: 1.4000 - val_accuracy: 0.4597
Epoch 10/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.3883 -
accuracy: 0.4576 - val_loss: 1.3811 - val_accuracy: 0.4638
Epoch 11/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.3567 -
accuracy: 0.4723 - val_loss: 1.3670 - val_accuracy: 0.4684
Epoch 12/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.3292 -
accuracy: 0.4808 - val_loss: 1.3372 - val_accuracy: 0.4746
Epoch 13/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.3109 -
accuracy: 0.4925 - val_loss: 1.3114 - val_accuracy: 0.4903
Epoch 14/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.2970 -
accuracy: 0.4978 - val_loss: 1.3210 - val_accuracy: 0.4880
Epoch 15/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.2691 -
accuracy: 0.5092 - val_loss: 1.2896 - val_accuracy: 0.4942
Epoch 16/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.2460 -
accuracy: 0.5192 - val_loss: 1.2654 - val_accuracy: 0.5121
Epoch 17/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.2248 -
accuracy: 0.5295 - val_loss: 1.2974 - val_accuracy: 0.4929
```

```
Epoch 18/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.1947 -
accuracy: 0.5395 - val_loss: 1.2806 - val_accuracy: 0.4985
Epoch 19/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.1848 -
accuracy: 0.5457 - val_loss: 1.2641 - val_accuracy: 0.5189
Epoch 20/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.1612 -
accuracy: 0.5573 - val_loss: 1.2336 - val_accuracy: 0.5260
Epoch 21/60
21412/21412 [==============================] - 1s 25us/step - loss: 1.1579 -
accuracy: 0.5580 - val_loss: 1.2589 - val_accuracy: 0.5170
Epoch 22/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.1166 -
accuracy: 0.5724 - val_loss: 1.2016 - val_accuracy: 0.5422
Epoch 23/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.0962 -
accuracy: 0.5848 - val_loss: 1.1769 - val_accuracy: 0.5489
Epoch 24/60
21412/21412 [==============================] - 1s 24us/step - loss: 1.0647 -
accuracy: 0.5958 - val_loss: 1.1968 - val_accuracy: 0.5463
Epoch 25/60
21412/21412 [==============================] - 1s 25us/step - loss: 1.0526 -
accuracy: 0.6009 - val_loss: 1.1549 - val_accuracy: 0.5581
Epoch 26/60
21412/21412 [==============================] - 1s 25us/step - loss: 1.0381 -
accuracy: 0.6054 - val_loss: 1.2334 - val_accuracy: 0.5359
Epoch 27/60
21412/21412 [==============================] - 1s 25us/step - loss: 1.0071 -
accuracy: 0.6212 - val_loss: 1.1445 - val_accuracy: 0.5661
Epoch 28/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.9927 -
accuracy: 0.6253 - val_loss: 1.1662 - val_accuracy: 0.5592
Epoch 29/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.9814 -
accuracy: 0.6304 - val_loss: 1.1416 - val_accuracy: 0.5693
Epoch 30/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.9535 -
accuracy: 0.6368 - val_loss: 1.1420 - val_accuracy: 0.5732
Epoch 31/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.9393 -
accuracy: 0.6460 - val_loss: 1.1173 - val_accuracy: 0.5742
Epoch 32/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.9126 -
accuracy: 0.6553 - val_loss: 1.1118 - val_accuracy: 0.5893
Epoch 33/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.9049 -
accuracy: 0.6607 - val_loss: 1.1487 - val_accuracy: 0.5678
```

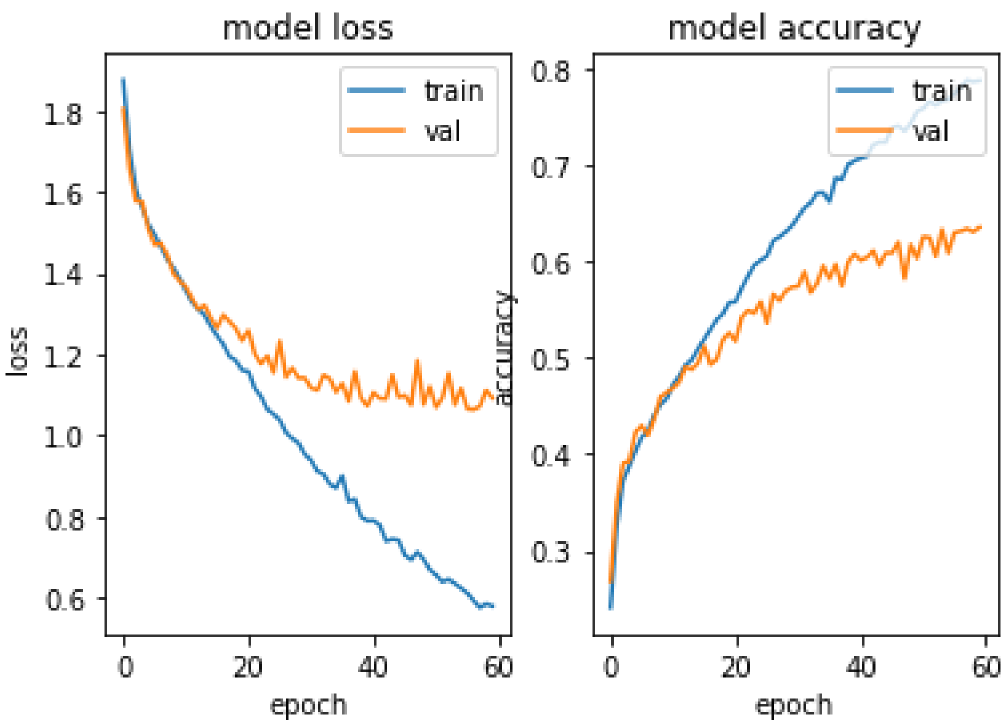
```
Epoch 34/60
21412/21412 [==============================] - 1s 25us/step - loss: 0.8806 -
accuracy: 0.6706 - val_loss: 1.1403 - val_accuracy: 0.5770
Epoch 35/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.8711 -
accuracy: 0.6713 - val_loss: 1.1069 - val_accuracy: 0.5953
Epoch 36/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.9009 -
accuracy: 0.6617 - val_loss: 1.1281 - val_accuracy: 0.5818
Epoch 37/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.8381 -
accuracy: 0.6865 - val_loss: 1.0852 - val_accuracy: 0.5968
Epoch 38/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.8435 -
accuracy: 0.6850 - val_loss: 1.1580 - val_accuracy: 0.5747
Epoch 39/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7999 -
accuracy: 0.7004 - val_loss: 1.0923 - val_accuracy: 0.5992
Epoch 40/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7899 -
accuracy: 0.7042 - val_loss: 1.0726 - val_accuracy: 0.6078
Epoch 41/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7907 -
accuracy: 0.7070 - val_loss: 1.1058 - val_accuracy: 0.6009
Epoch 42/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7794 -
accuracy: 0.7092 - val_loss: 1.0917 - val_accuracy: 0.6040
Epoch 43/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7400 -
accuracy: 0.7209 - val_loss: 1.0902 - val_accuracy: 0.6108
Epoch 44/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7452 -
accuracy: 0.7237 - val_loss: 1.1497 - val_accuracy: 0.5956
Epoch 45/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7421 -
accuracy: 0.7235 - val_loss: 1.0946 - val_accuracy: 0.6085
Epoch 46/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7052 -
accuracy: 0.7385 - val_loss: 1.0990 - val_accuracy: 0.6085
Epoch 47/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6941 -
accuracy: 0.7405 - val_loss: 1.0752 - val_accuracy: 0.6195
Epoch 48/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.7131 -
accuracy: 0.7351 - val_loss: 1.1848 - val_accuracy: 0.5820
Epoch 49/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6949 -
accuracy: 0.7431 - val_loss: 1.0765 - val_accuracy: 0.6173
```

```
Epoch 50/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6694 -
accuracy: 0.7554 - val_loss: 1.1203 - val_accuracy: 0.6029
Epoch 51/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6564 -
accuracy: 0.7585 - val_loss: 1.0699 - val_accuracy: 0.6253
Epoch 52/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6397 -
accuracy: 0.7656 - val_loss: 1.0927 - val_accuracy: 0.6246
Epoch 53/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6461 -
accuracy: 0.7618 - val_loss: 1.1537 - val_accuracy: 0.6050
Epoch 54/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6352 -
accuracy: 0.7662 - val_loss: 1.0757 - val_accuracy: 0.6332
Epoch 55/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6238 -
accuracy: 0.7678 - val_loss: 1.1162 - val_accuracy: 0.6085
Epoch 56/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.6116 -
accuracy: 0.7741 - val_loss: 1.0660 - val_accuracy: 0.6298
Epoch 57/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.5933 -
accuracy: 0.7803 - val_loss: 1.0641 - val_accuracy: 0.6311
Epoch 58/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.5760 -
accuracy: 0.7879 - val_loss: 1.0719 - val_accuracy: 0.6337
Epoch 59/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.5855 -
accuracy: 0.7860 - val_loss: 1.1115 - val_accuracy: 0.6304
Epoch 60/60
21412/21412 [==============================] - 1s 24us/step - loss: 0.5801 -
accuracy: 0.7874 - val_loss: 1.0928 - val_accuracy: 0.6352
```

[187]:

```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epoch')
```

```
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```



[188]:
```
score = model.evaluate(x=X_test, y=y_test)
predictions = model.predict(X_test)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
2974/2974 [==============================] - 0s 68us/step
Test loss: 1.0448104100653848
Test accuracy: 0.6573638319969177
```

[189]:
```
print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
 ↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support

           A       0.60      0.56      0.58       389
           E       0.66      0.58      0.61       273
           F       0.59      0.53      0.56       378
           L       0.58      0.63      0.61       440
           N       0.61      0.66      0.63       410
```

```
           T       0.79      0.68      0.73       373
           W       0.73      0.81      0.77       711

    accuracy                           0.66      2974
   macro avg       0.65      0.64      0.64      2974
weighted avg       0.66      0.66      0.66      2974
```

6.0.19 lstm

[190]:
```python
config = Config(mode='lstm')
```

[194]:
```python
history = model.fit(X_train, y_train, epochs=100, batch_size=512,␣
 ↪validation_data=(X_val, y_val), class_weight=class_weight)
```

```
Train on 42825 samples, validate on 10707 samples
Epoch 1/100
42825/42825 [==============================] - 3s 80us/step - loss: 1.8873 -
accuracy: 0.2358 - val_loss: 1.8328 - val_accuracy: 0.2650
Epoch 2/100
42825/42825 [==============================] - 3s 61us/step - loss: 1.7721 -
accuracy: 0.2878 - val_loss: 1.7261 - val_accuracy: 0.3179
Epoch 3/100
42825/42825 [==============================] - 3s 64us/step - loss: 1.6884 -
accuracy: 0.3316 - val_loss: 1.6572 - val_accuracy: 0.3395
Epoch 4/100
42825/42825 [==============================] - 3s 64us/step - loss: 1.6409 -
accuracy: 0.3493 - val_loss: 1.6207 - val_accuracy: 0.3555
Epoch 5/100
42825/42825 [==============================] - 3s 65us/step - loss: 1.6095 -
accuracy: 0.3598 - val_loss: 1.5845 - val_accuracy: 0.3723
Epoch 6/100
42825/42825 [==============================] - 3s 61us/step - loss: 1.5970 -
accuracy: 0.3667 - val_loss: 1.6339 - val_accuracy: 0.3484
Epoch 7/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.5602 -
accuracy: 0.3809 - val_loss: 1.5349 - val_accuracy: 0.3864
Epoch 8/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.5406 -
accuracy: 0.3866 - val_loss: 1.5102 - val_accuracy: 0.3952
Epoch 9/100
42825/42825 [==============================] - 3s 58us/step - loss: 1.5160 -
accuracy: 0.3949 - val_loss: 1.5016 - val_accuracy: 0.3979
Epoch 10/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.4928 -
accuracy: 0.4053 - val_loss: 1.5391 - val_accuracy: 0.3868
Epoch 11/100
```

```
42825/42825 [==============================] - 2s 57us/step - loss: 1.4707 -
accuracy: 0.4130 - val_loss: 1.5046 - val_accuracy: 0.3945
Epoch 12/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.4612 -
accuracy: 0.4199 - val_loss: 1.5130 - val_accuracy: 0.3982
Epoch 13/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.4542 -
accuracy: 0.4252 - val_loss: 1.4087 - val_accuracy: 0.4415
Epoch 14/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.4246 -
accuracy: 0.4335 - val_loss: 1.4634 - val_accuracy: 0.4155
Epoch 15/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.4134 -
accuracy: 0.4380 - val_loss: 1.5688 - val_accuracy: 0.3821
Epoch 16/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.4061 -
accuracy: 0.4418 - val_loss: 1.3730 - val_accuracy: 0.4540
Epoch 17/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.4143 -
accuracy: 0.4399 - val_loss: 1.3862 - val_accuracy: 0.4479
Epoch 18/100
42825/42825 [==============================] - 3s 58us/step - loss: 1.3729 -
accuracy: 0.4543 - val_loss: 1.3521 - val_accuracy: 0.4567
Epoch 19/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.3508 -
accuracy: 0.4637 - val_loss: 1.3310 - val_accuracy: 0.4686
Epoch 20/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.3479 -
accuracy: 0.4647 - val_loss: 1.4267 - val_accuracy: 0.4342
Epoch 21/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.3365 -
accuracy: 0.4685 - val_loss: 1.3051 - val_accuracy: 0.4728
Epoch 22/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.3429 -
accuracy: 0.4653 - val_loss: 1.3072 - val_accuracy: 0.4681
Epoch 23/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.3087 -
accuracy: 0.4769 - val_loss: 1.2998 - val_accuracy: 0.4807
Epoch 24/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.3100 -
accuracy: 0.4810 - val_loss: 1.3210 - val_accuracy: 0.4697
Epoch 25/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.3106 -
accuracy: 0.4769 - val_loss: 1.3481 - val_accuracy: 0.4511
Epoch 26/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.2899 -
accuracy: 0.4874 - val_loss: 1.2639 - val_accuracy: 0.4982
Epoch 27/100
```

```
42825/42825 [==============================] - 2s 58us/step - loss: 1.2776 -
accuracy: 0.4920 - val_loss: 1.2882 - val_accuracy: 0.4906
Epoch 28/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.2645 -
accuracy: 0.4958 - val_loss: 1.2690 - val_accuracy: 0.4929
Epoch 29/100
42825/42825 [==============================] - 3s 60us/step - loss: 1.2627 -
accuracy: 0.4970 - val_loss: 1.2395 - val_accuracy: 0.5019
Epoch 30/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.2494 -
accuracy: 0.5004 - val_loss: 1.2658 - val_accuracy: 0.4907
Epoch 31/100
42825/42825 [==============================] - 3s 58us/step - loss: 1.2564 -
accuracy: 0.5006 - val_loss: 1.2494 - val_accuracy: 0.4931
Epoch 32/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.2555 -
accuracy: 0.5012 - val_loss: 1.3381 - val_accuracy: 0.4648
Epoch 33/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.2201 -
accuracy: 0.5140 - val_loss: 1.2570 - val_accuracy: 0.4935
Epoch 34/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.2170 -
accuracy: 0.5139 - val_loss: 1.2091 - val_accuracy: 0.5163
Epoch 35/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.2064 -
accuracy: 0.5200 - val_loss: 1.2307 - val_accuracy: 0.5124
Epoch 36/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.2122 -
accuracy: 0.5197 - val_loss: 1.2441 - val_accuracy: 0.5012
Epoch 37/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.2003 -
accuracy: 0.5230 - val_loss: 1.2010 - val_accuracy: 0.5227
Epoch 38/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1760 -
accuracy: 0.5315 - val_loss: 1.1822 - val_accuracy: 0.5273
Epoch 39/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1986 -
accuracy: 0.5239 - val_loss: 1.2374 - val_accuracy: 0.5086
Epoch 40/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1676 -
accuracy: 0.5352 - val_loss: 1.1640 - val_accuracy: 0.5335
Epoch 41/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1698 -
accuracy: 0.5358 - val_loss: 1.2304 - val_accuracy: 0.5071
Epoch 42/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1531 -
accuracy: 0.5405 - val_loss: 1.2279 - val_accuracy: 0.5154
Epoch 43/100
```

```
42825/42825 [==============================] - 2s 58us/step - loss: 1.1529 -
accuracy: 0.5418 - val_loss: 1.1446 - val_accuracy: 0.5419
Epoch 44/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1377 -
accuracy: 0.5472 - val_loss: 1.1478 - val_accuracy: 0.5424
Epoch 45/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1310 -
accuracy: 0.5501 - val_loss: 1.1521 - val_accuracy: 0.5366
Epoch 46/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1333 -
accuracy: 0.5510 - val_loss: 1.1317 - val_accuracy: 0.5435
Epoch 47/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.1248 -
accuracy: 0.5541 - val_loss: 1.1522 - val_accuracy: 0.5443
Epoch 48/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1231 -
accuracy: 0.5543 - val_loss: 1.1337 - val_accuracy: 0.5476
Epoch 49/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.1244 -
accuracy: 0.5540 - val_loss: 1.1853 - val_accuracy: 0.5249
Epoch 50/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1089 -
accuracy: 0.5612 - val_loss: 1.1423 - val_accuracy: 0.5468
Epoch 51/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.1039 -
accuracy: 0.5646 - val_loss: 1.1031 - val_accuracy: 0.5588
Epoch 52/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0854 -
accuracy: 0.5717 - val_loss: 1.1215 - val_accuracy: 0.5555
Epoch 53/100
42825/42825 [==============================] - 3s 60us/step - loss: 1.0909 -
accuracy: 0.5704 - val_loss: 1.1853 - val_accuracy: 0.5346
Epoch 54/100
42825/42825 [==============================] - 3s 58us/step - loss: 1.0709 -
accuracy: 0.5762 - val_loss: 1.1043 - val_accuracy: 0.5586
Epoch 55/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0611 -
accuracy: 0.5825 - val_loss: 1.1048 - val_accuracy: 0.5654
Epoch 56/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0487 -
accuracy: 0.5870 - val_loss: 1.1028 - val_accuracy: 0.5636
Epoch 57/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.0750 -
accuracy: 0.5740 - val_loss: 1.1333 - val_accuracy: 0.5538
Epoch 58/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0560 -
accuracy: 0.5834 - val_loss: 1.0818 - val_accuracy: 0.5758
Epoch 59/100
```

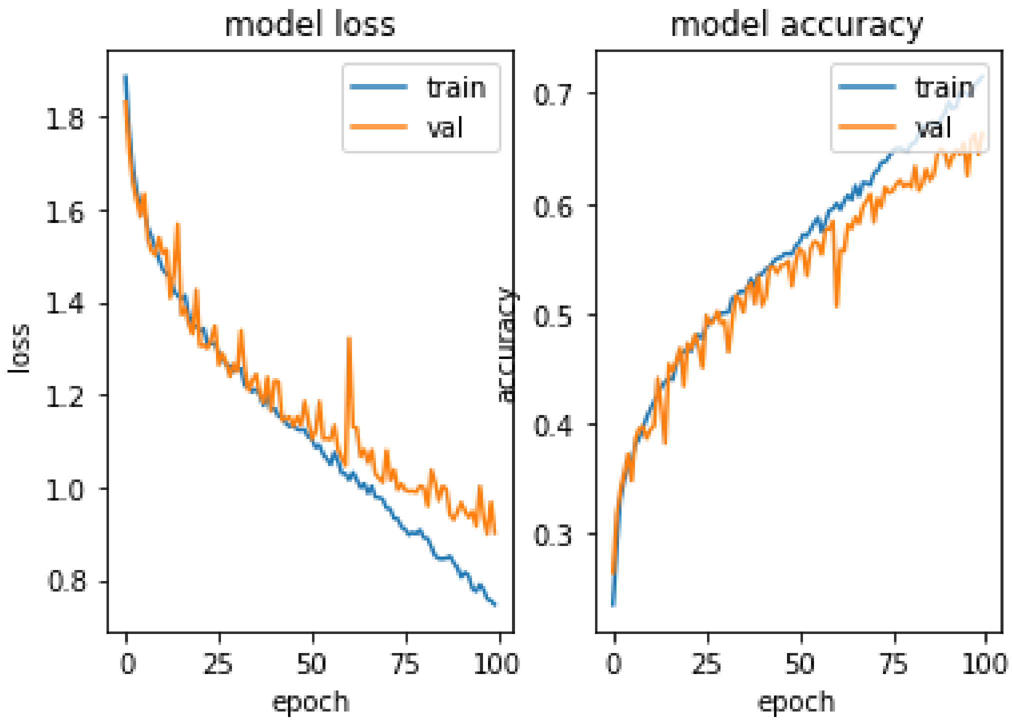
```
42825/42825 [==============================] - 2s 58us/step - loss: 1.0299 -
accuracy: 0.5936 - val_loss: 1.0614 - val_accuracy: 0.5769
Epoch 60/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0293 -
accuracy: 0.5951 - val_loss: 1.0468 - val_accuracy: 0.5837
Epoch 61/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0159 -
accuracy: 0.5998 - val_loss: 1.3217 - val_accuracy: 0.5067
Epoch 62/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0319 -
accuracy: 0.5943 - val_loss: 1.1304 - val_accuracy: 0.5578
Epoch 63/100
42825/42825 [==============================] - 3s 59us/step - loss: 1.0166 -
accuracy: 0.6018 - val_loss: 1.1276 - val_accuracy: 0.5565
Epoch 64/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9992 -
accuracy: 0.6074 - val_loss: 1.0659 - val_accuracy: 0.5809
Epoch 65/100
42825/42825 [==============================] - 2s 58us/step - loss: 1.0073 -
accuracy: 0.6034 - val_loss: 1.0805 - val_accuracy: 0.5772
Epoch 66/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9869 -
accuracy: 0.6162 - val_loss: 1.0523 - val_accuracy: 0.5877
Epoch 67/100
42825/42825 [==============================] - 2s 57us/step - loss: 1.0018 -
accuracy: 0.6067 - val_loss: 1.0798 - val_accuracy: 0.5820
Epoch 68/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.9784 -
accuracy: 0.6189 - val_loss: 1.0283 - val_accuracy: 0.5968
Epoch 69/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9777 -
accuracy: 0.6184 - val_loss: 1.0198 - val_accuracy: 0.6022
Epoch 70/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.9736 -
accuracy: 0.6169 - val_loss: 1.0102 - val_accuracy: 0.6082
Epoch 71/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9550 -
accuracy: 0.6272 - val_loss: 1.0791 - val_accuracy: 0.5829
Epoch 72/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9525 -
accuracy: 0.6298 - val_loss: 1.0153 - val_accuracy: 0.6044
Epoch 73/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9332 -
accuracy: 0.6374 - val_loss: 1.0374 - val_accuracy: 0.5967
Epoch 74/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.9313 -
accuracy: 0.6377 - val_loss: 0.9944 - val_accuracy: 0.6144
Epoch 75/100
```

```
42825/42825 [==============================] - 2s 58us/step - loss: 0.9144 -
accuracy: 0.6424 - val_loss: 1.0072 - val_accuracy: 0.6097
Epoch 76/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9086 -
accuracy: 0.6471 - val_loss: 0.9936 - val_accuracy: 0.6114
Epoch 77/100
42825/42825 [==============================] - 3s 59us/step - loss: 0.8971 -
accuracy: 0.6501 - val_loss: 0.9909 - val_accuracy: 0.6176
Epoch 78/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.9020 -
accuracy: 0.6500 - val_loss: 0.9915 - val_accuracy: 0.6208
Epoch 79/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8992 -
accuracy: 0.6484 - val_loss: 0.9890 - val_accuracy: 0.6148
Epoch 80/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.9073 -
accuracy: 0.6467 - val_loss: 1.0022 - val_accuracy: 0.6181
Epoch 81/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8905 -
accuracy: 0.6539 - val_loss: 0.9993 - val_accuracy: 0.6146
Epoch 82/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8887 -
accuracy: 0.6549 - val_loss: 0.9589 - val_accuracy: 0.6335
Epoch 83/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.8705 -
accuracy: 0.6620 - val_loss: 1.0371 - val_accuracy: 0.6120
Epoch 84/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.8513 -
accuracy: 0.6710 - val_loss: 1.0085 - val_accuracy: 0.6174
Epoch 85/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.8457 -
accuracy: 0.6697 - val_loss: 0.9712 - val_accuracy: 0.6310
Epoch 86/100
42825/42825 [==============================] - 3s 60us/step - loss: 0.8448 -
accuracy: 0.6725 - val_loss: 1.0017 - val_accuracy: 0.6209
Epoch 87/100
42825/42825 [==============================] - 3s 58us/step - loss: 0.8464 -
accuracy: 0.6738 - val_loss: 0.9962 - val_accuracy: 0.6262
Epoch 88/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8500 -
accuracy: 0.6700 - val_loss: 0.9408 - val_accuracy: 0.6443
Epoch 89/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8363 -
accuracy: 0.6763 - val_loss: 0.9296 - val_accuracy: 0.6495
Epoch 90/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8253 -
accuracy: 0.6821 - val_loss: 0.9462 - val_accuracy: 0.6459
Epoch 91/100
```

```
42825/42825 [==============================] - 2s 58us/step - loss: 0.8052 -
accuracy: 0.6920 - val_loss: 0.9665 - val_accuracy: 0.6330
Epoch 92/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.8151 -
accuracy: 0.6858 - val_loss: 0.9488 - val_accuracy: 0.6425
Epoch 93/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.8081 -
accuracy: 0.6875 - val_loss: 0.9347 - val_accuracy: 0.6488
Epoch 94/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.7817 -
accuracy: 0.7000 - val_loss: 0.9475 - val_accuracy: 0.6445
Epoch 95/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.7745 -
accuracy: 0.7018 - val_loss: 0.9160 - val_accuracy: 0.6546
Epoch 96/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.7897 -
accuracy: 0.6980 - val_loss: 1.0017 - val_accuracy: 0.6253
Epoch 97/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.7782 -
accuracy: 0.6999 - val_loss: 0.9332 - val_accuracy: 0.6579
Epoch 98/100
42825/42825 [==============================] - 2s 57us/step - loss: 0.7588 -
accuracy: 0.7071 - val_loss: 0.8986 - val_accuracy: 0.6627
Epoch 99/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.7544 -
accuracy: 0.7109 - val_loss: 0.9675 - val_accuracy: 0.6444
Epoch 100/100
42825/42825 [==============================] - 2s 58us/step - loss: 0.7461 -
accuracy: 0.7150 - val_loss: 0.9011 - val_accuracy: 0.6636
```

[195]:

```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

ax2.plot(history.history['accuracy'])
ax2.plot(history.history['val_accuracy'])
ax2.set_title('model accuracy')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper right')

plt.show()
```




```
[196]: score = model.evaluate(x=X_test, y=y_test)
predictions = model.predict(X_test)

print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
5948/5948 [==============================] - 1s 133us/step
Test loss: 0.8837402156718882
Test accuracy: 0.6753530502319336
```

```
[197]: print(classification_report(classes[np.argmax(y_test, axis=1)], classes[np.
 ↪argmax(predictions, axis=1)]))
```

```
              precision    recall  f1-score   support

           A       0.60      0.72      0.66       762
           E       0.69      0.55      0.61       516
           F       0.69      0.58      0.63       837
           L       0.63      0.50      0.56       910
           N       0.58      0.70      0.63       887
           T       0.72      0.69      0.70       659
           W       0.78      0.85      0.82      1377

    accuracy                           0.68      5948
```

```
   macro avg       0.67      0.66      0.66      5948
weighted avg       0.68      0.68      0.67      5948
```

Need a lot more epochs for training to converge.

Autoencoder

I tried to encode the mfccs to a lower dimensionality with the idea of training the models with the encoded representation, but this approach was not very successful. Indeed, the mfccs are already pretty low dimensional so it may be a waste of time. The end goal was to have a VAE to generate new samples to enrich the training data, but I did not have the time to implement this. Below you can find different autoencoder implementations.

[198]:
```
# this is the size of our encoded representations
encoding_dim = 60  # 32 floats -> compression of factor 24.5, assuming the␣
↪input is 784 floats

# this is our input placeholder
input_data = Input(shape=(117,))
# "encoded" is the encoded representation of the input
encoded = Dense(encoding_dim, activation='relu')(input_data)
# "decoded" is the lossy reconstruction of the input
decoded = Dense(117, activation='sigmoid')(encoded)

# this model maps an input to its reconstruction
autoencoder = Model(input_data, decoded)
opt = Adadelta(learning_rate=0.1)
autoencoder.compile(optimizer=opt, loss='binary_crossentropy')
autoencoder.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 117)               0
_________________________________________________________________
dense_46 (Dense)             (None, 60)                7080
_________________________________________________________________
dense_47 (Dense)             (None, 117)               7137
=================================================================
Total params: 14,217
Trainable params: 14,217
Non-trainable params: 0
_________________________________________________________________
```

[199]:
```
# this model maps an input to its encoded representation
encoder = Model(input_data, encoded)
```

```
[200]: # create a placeholder for an encoded (32-dimensional) input
encoded_input = Input(shape=(encoding_dim,))
# retrieve the last layer of the autoencoder model
decoder_layer = autoencoder.layers[-1]
# create the decoder model
decoder = Model(encoded_input, decoder_layer(encoded_input))
```

```
[201]: history = autoencoder.fit(X_train.reshape(X_train.shape[0], -1), X_train.
 ↪reshape(X_train.shape[0], -1),
                epochs=100,
                batch_size=512,
                validation_data=(X_val.reshape(X_val.shape[0], -1), X_val.
 ↪reshape(X_val.shape[0], -1)))
```

```
Train on 42825 samples, validate on 10707 samples
Epoch 1/100
42825/42825 [==============================] - 0s 9us/step - loss: 0.6983 -
val_loss: 0.6954
Epoch 2/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6944 -
val_loss: 0.6937
Epoch 3/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6934 -
val_loss: 0.6931
Epoch 4/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6930 -
val_loss: 0.6928
Epoch 5/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6927 -
val_loss: 0.6926
Epoch 6/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6926 -
val_loss: 0.6925
Epoch 7/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6925 -
val_loss: 0.6924
Epoch 8/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6924 -
val_loss: 0.6923
Epoch 9/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6923 -
val_loss: 0.6922
Epoch 10/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6922 -
val_loss: 0.6922
Epoch 11/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6921 -
```

```
val_loss: 0.6921
Epoch 12/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6921 -
val_loss: 0.6920
Epoch 13/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6920 -
val_loss: 0.6920
Epoch 14/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6919 -
val_loss: 0.6919
Epoch 15/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6919 -
val_loss: 0.6918
Epoch 16/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6918 -
val_loss: 0.6918
Epoch 17/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6918 -
val_loss: 0.6917
Epoch 18/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6917 -
val_loss: 0.6917
Epoch 19/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6917 -
val_loss: 0.6916
Epoch 20/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6916 -
val_loss: 0.6916
Epoch 21/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6916 -
val_loss: 0.6915
Epoch 22/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6915 -
val_loss: 0.6915
Epoch 23/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6915 -
val_loss: 0.6914
Epoch 24/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6914 -
val_loss: 0.6914
Epoch 25/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6914 -
val_loss: 0.6914
Epoch 26/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6914 -
val_loss: 0.6913
Epoch 27/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6913 -
```

```
val_loss: 0.6913
Epoch 28/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6913 -
val_loss: 0.6913
Epoch 29/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6913 -
val_loss: 0.6912
Epoch 30/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6912 -
val_loss: 0.6912
Epoch 31/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6912 -
val_loss: 0.6912
Epoch 32/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6912 -
val_loss: 0.6911
Epoch 33/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 34/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 35/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 36/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6910
Epoch 37/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6910 -
val_loss: 0.6910
Epoch 38/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6910 -
val_loss: 0.6910
Epoch 39/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6910 -
val_loss: 0.6910
Epoch 40/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6910 -
val_loss: 0.6909
Epoch 41/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6909 -
val_loss: 0.6909
Epoch 42/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6909 -
val_loss: 0.6909
Epoch 43/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6909 -
```

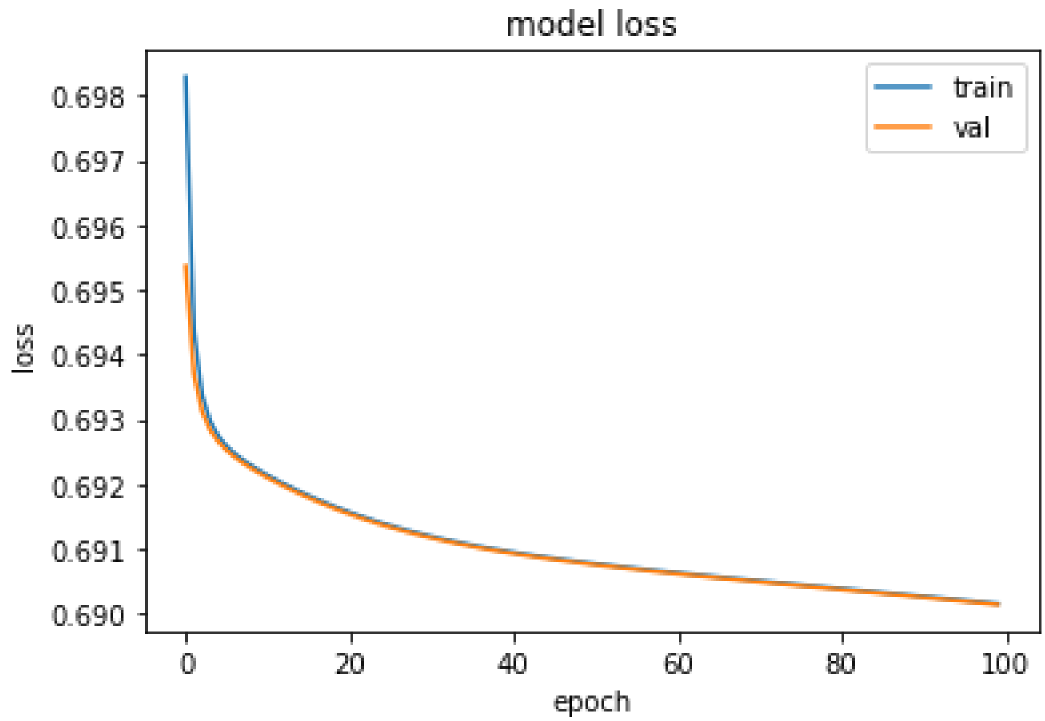
```
val_loss: 0.6909
Epoch 44/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6909 -
val_loss: 0.6909
Epoch 45/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6909 -
val_loss: 0.6908
Epoch 46/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6908
Epoch 47/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6908
Epoch 48/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6908
Epoch 49/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6908
Epoch 50/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6908
Epoch 51/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6908
Epoch 52/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6908 -
val_loss: 0.6907
Epoch 53/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
val_loss: 0.6907
Epoch 54/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
val_loss: 0.6907
Epoch 55/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
val_loss: 0.6907
Epoch 56/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
val_loss: 0.6907
Epoch 57/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
val_loss: 0.6907
Epoch 58/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
val_loss: 0.6907
Epoch 59/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6907 -
```

```
val_loss: 0.6906
Epoch 60/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6906 -
val_loss: 0.6906
Epoch 61/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6906 -
val_loss: 0.6906
Epoch 62/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6906 -
val_loss: 0.6906
Epoch 63/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6906 -
val_loss: 0.6906
Epoch 64/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6906 -
val_loss: 0.6906
Epoch 65/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6906 -
val_loss: 0.6906
Epoch 66/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6906 -
val_loss: 0.6905
Epoch 67/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6906 -
val_loss: 0.6905
Epoch 68/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 69/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 70/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 71/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 72/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 73/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 74/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
val_loss: 0.6905
Epoch 75/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6905 -
```

```
val_loss: 0.6904
Epoch 76/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 77/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 78/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 79/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 80/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 81/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 82/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6904
Epoch 83/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6904 -
val_loss: 0.6903
Epoch 84/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 85/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 86/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 87/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 88/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 89/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 90/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6903 -
val_loss: 0.6903
Epoch 91/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6903 -
```

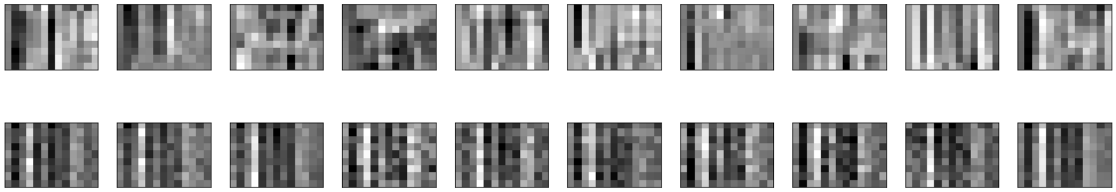
```
val_loss: 0.6902
Epoch 92/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6903 -
val_loss: 0.6902
Epoch 93/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6902 -
val_loss: 0.6902
Epoch 94/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6902 -
val_loss: 0.6902
Epoch 95/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6902 -
val_loss: 0.6902
Epoch 96/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6902 -
val_loss: 0.6902
Epoch 97/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6902 -
val_loss: 0.6902
Epoch 98/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6902 -
val_loss: 0.6902
Epoch 99/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6902 -
val_loss: 0.6901
Epoch 100/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6902 -
val_loss: 0.6901
```

[202]:
```
fig, ax1 = plt.subplots(1)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

plt.show()
```



[203]:
```
encoded_data = encoder.predict(X_test.reshape(X_test.shape[0], -1))
decoded_data = decoder.predict(encoded_data)
```

[204]:
```
n = 10  # how many mfccs we will display
plt.figure(figsize=(20, 4))
for i in range(n):
    # display original
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(X_test[i].reshape(9, 13))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # display reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(decoded_data[i].reshape(9, 13))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



sparse autoencoder

[205]:
```
# this is the size of our encoded representations
encoding_dim = 60  # 32 floats -> compression of factor 24.5, assuming the␣
 ↪input is 784 floats

# this is our input placeholder
input_data = Input(shape=(117,))
# "encoded" is the encoded representation of the input
encoded = Dense(encoding_dim, activation='relu',
                activity_regularizer=regularizers.l1(10e-5))(input_data)
# "decoded" is the lossy reconstruction of the input
decoded = Dense(117, activation='sigmoid')(encoded)

# this model maps an input to its reconstruction
sparse_autoencoder = Model(input_data, decoded)

sparse_autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')
sparse_autoencoder.summary()
```

```
Model: "model_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 117)               0
_________________________________________________________________
dense_48 (Dense)             (None, 60)                7080
_________________________________________________________________
dense_49 (Dense)             (None, 117)               7137
=================================================================
Total params: 14,217
Trainable params: 14,217
Non-trainable params: 0
_________________________________________________________________
```

[206]:
```
# this model maps an input to its encoded representation
encoder = Model(input_data, encoded)
```

[207]:
```
# create a placeholder for an encoded (32-dimensional) input
encoded_input = Input(shape=(encoding_dim,))
# retrieve the last layer of the autoencoder model
decoder_layer = autoencoder.layers[-1]
# create the decoder model
decoder = Model(encoded_input, decoder_layer(encoded_input))
```

[208]:
```
history = sparse_autoencoder.fit(X_train.reshape(X_train.shape[0], -1), X_train.
↪reshape(X_train.shape[0], -1),
                epochs=100,
                batch_size=512,
                validation_data=(X_val.reshape(X_val.shape[0], -1), X_val.
↪reshape(X_val.shape[0], -1)))
```

```
Train on 42825 samples, validate on 10707 samples
Epoch 1/100
42825/42825 [==============================] - 0s 8us/step - loss: 0.7289 -
val_loss: 0.6927
Epoch 2/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6924 -
val_loss: 0.6922
Epoch 3/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6920 -
val_loss: 0.6918
Epoch 4/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6917 -
val_loss: 0.6916
Epoch 5/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6915 -
val_loss: 0.6915
Epoch 6/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6914 -
val_loss: 0.6914
Epoch 7/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6913 -
val_loss: 0.6913
Epoch 8/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6913 -
val_loss: 0.6912
Epoch 9/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6912 -
val_loss: 0.6912
Epoch 10/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6912 -
val_loss: 0.6912
Epoch 11/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6912 -
```

```
val_loss: 0.6912
Epoch 12/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6912 -
val_loss: 0.6911
Epoch 13/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 14/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 15/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 16/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 17/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 18/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 19/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 20/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 21/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 22/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 23/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 24/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 25/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 26/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 27/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
```

```
val_loss: 0.6911
Epoch 28/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 29/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 30/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 31/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 32/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 33/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 34/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 35/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 36/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 37/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 38/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 39/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 40/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 41/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 42/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 43/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
```

```
val_loss: 0.6911
Epoch 44/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 45/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 46/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 47/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 48/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 49/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 50/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 51/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 52/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 53/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 54/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 55/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 56/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 57/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 58/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 59/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
```

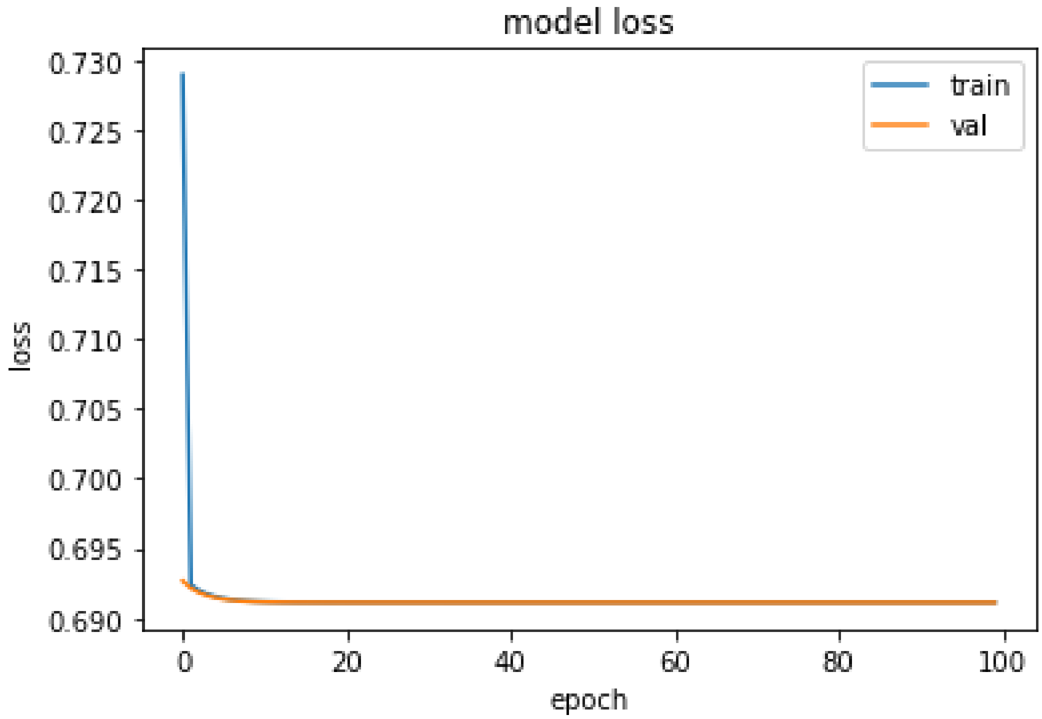
```
val_loss: 0.6911
Epoch 60/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 61/100
42825/42825 [==============================] - 0s 5us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 62/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 63/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 64/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 65/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 66/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 67/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 68/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 69/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 70/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 71/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 72/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 73/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 74/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 75/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
```

```
val_loss: 0.6911
Epoch 76/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 77/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 78/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 79/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 80/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 81/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 82/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 83/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 84/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 85/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 86/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 87/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 88/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 89/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 90/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 91/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
```

```
val_loss: 0.6911
Epoch 92/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 93/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 94/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 95/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 96/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 97/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 98/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 99/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
Epoch 100/100
42825/42825 [==============================] - 0s 6us/step - loss: 0.6911 -
val_loss: 0.6911
```

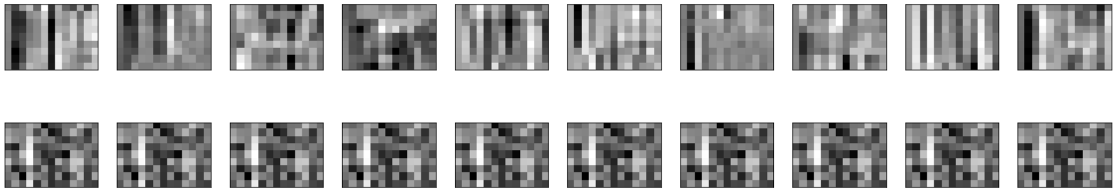
[209]:

```
fig, ax1 = plt.subplots(1)
ax1.plot(history.history['loss'])
ax1.plot(history.history['val_loss'])
ax1.set_title('model loss')
ax1.set_ylabel('loss')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper right')

plt.show()
```



[210]:
```
encoded_data = encoder.predict(X_test.reshape(X_test.shape[0], -1))
decoded_data = decoder.predict(encoded_data)
```

[211]:
```
n = 10  # how many mfccs we will display
plt.figure(figsize=(20, 4))
for i in range(n):
    # display original
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(X_test[i].reshape(9, 13))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # display reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(decoded_data[i].reshape(9, 13))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



7 Summary of Results

In summary, the best model was surprisingly the kNN (even without any class balancing). The CNNs performed worse and at a greater computational cost (even with class balancing), so it was decided to focus on the simple kNN instead.

Since the kNN model is fast to fit, I performed a grid search CV using a grid of possible hyperparameters. The best hyperparameters are determined by the CV test score (mean test accuracy across the folds).

The kNN achieves 96.9% CV test accuracy with:

- n_neighbors: 3
- weights: 'distance'
- algorithm: 'auto'
- leaf_size: '5'
- metric: 'euclidean'

and is implemented in the Flask app. Note that the current test CV displayed in the notebook is slightly lower, as the original run was not done with a fixed random seed. You can also note that the SVM model has slightly higher test accuracy during this final notebook run. This wasn't the case previously (kNN outperformed SVM) so the SVM wasn't considered to be the best model in the Flask implementation.