Question 2 :

All input audio clips are sampled at 44·1 kHz. Duration of clips vary from less than a second to about 26 seconds. Even though number of samples for each label are the same, total duration for each label varies. Test samples could also vary in duration. For a clip of 1 second duration with a single channel, the input is a 1-D array of length 44,100 which represents the signal amplitude in time domain.

In order to process all inputs with same system, we have to either :

a) standardise input size

b) design neural arch. that can process inputs of varying size.

We choose to standardise input size.

Data - preprocessing :

Leading, trailing & intermediary silence in audio clips are trimmed. Hyper-parameter, 'time-res' refers to duration of audio clips (seconds) that will be fed to neural network. (sampled at 44·1 kHz).

This interval hopes to be shortest interval than can capture the discerning feature of any all class. The next sample is considered after a delay of 0·25 seconds. If signal is less than 0·5 seconds (time-res is 0·5 seconds), it is zero-padded cented with zero padding.

For ex, a 1 second duration clip will be partitioned into
3 samples ([0,0·5], [0·25, 0·75], [0·5, 1]).

Fast-fourier transform of 0·5 s interval clip is taken & its amplitude spectrum is considered as input to the neral n/w. The amplitude spectrum produces an array of 22050 length. (for 0·5 second). But, since signals is a reflection after halfway, array [0 : 11025] is considered as input to neural network

Initially, mel frequency cepstral coefficients (mfcc) were used as feature but here the discerning feature could be anywhere in time. Taking frequency spectrum of signal helps to remove temporal dependency. Mfcc will be useful for tasks requiring disambiguation along time axis like speech recognition. Here we are concerned with event detection.

After data-preprocessing, dataset became unbalanced. Therefore weighted random sampler was used for the dataloader.

- No external data was used for training/validation or test.

Neural Network Architecture:

Idea was to use a simple network.

Input layer : 1-D array of size 11025.

It is followed by a fully connected layer of 1000 units.

The non-linearity for activation is Relu function.

Next hidden layer is also a fully connected layer of 1000 units

Non-linearity used is Relu function

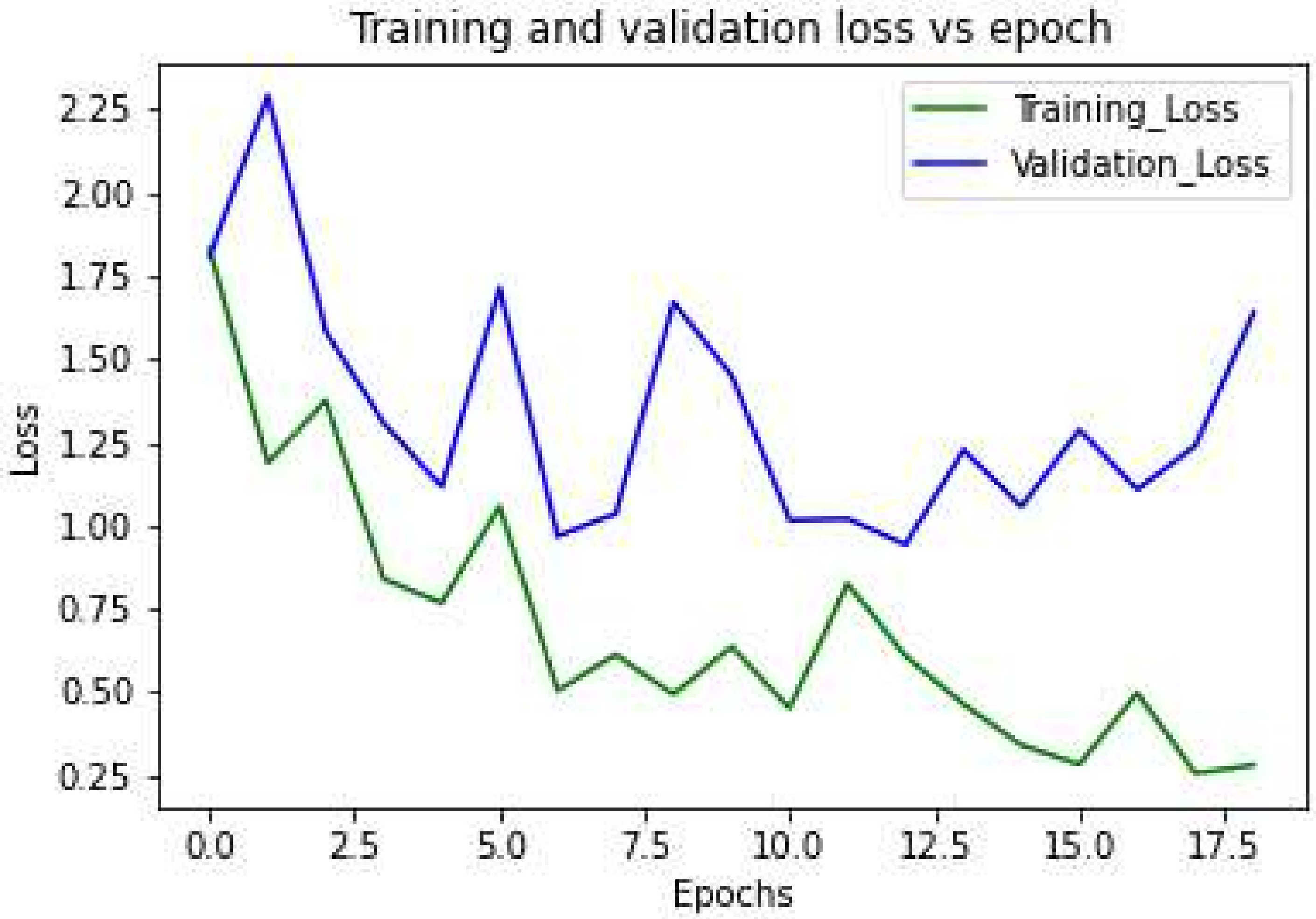
Next hidden layer has 7 neurons, fully connected to prev. layer.

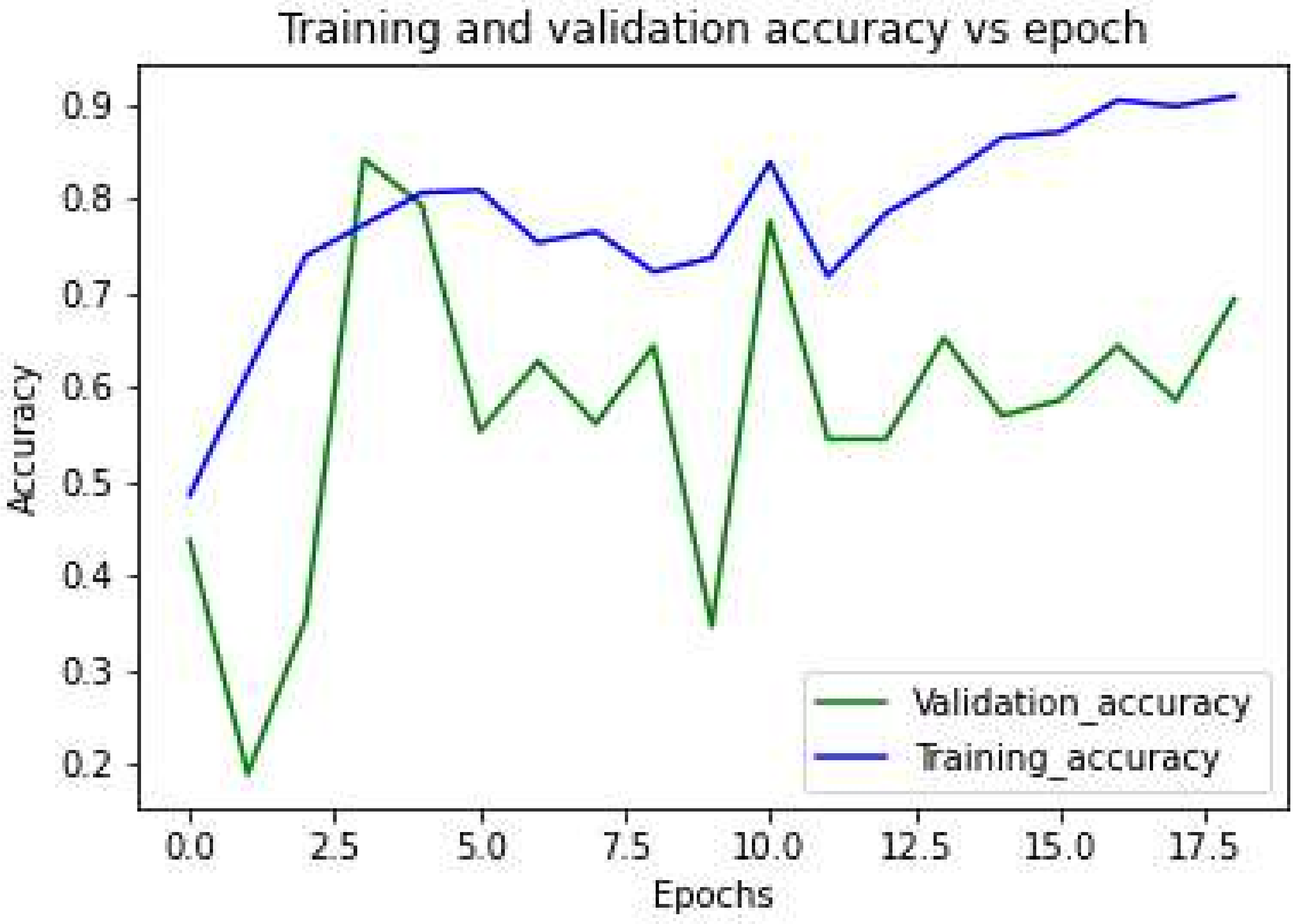
7 neurons are connected to softmax layer to output probabilities corresponding to each of 7 classes.

Loss function used in cross-entropy loss.

Fully connected layers are used to make sure all possible combinations of frequency is considered for computing useful features if necessary. Learning rate is 10^{-3}. Validation loss & training loss was compared to stop training.

Guidelines for using the code are provided as comments in code itself


Training and validation loss vs epoch
Training_Loss
Validation_Loss
Loss
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epochs


Training and validation accuracy vs epoch
Validation_accuracy
Training_accuracy
Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epochs