Speech Command Project

Supervisor : GVV Sharma

P Jayanth

MA18BTECH11004

EE5600- Speech Command Model

Aim

Speech Command Recoginition Model.

Table of Contents

Table of Contents

Generating Data and Storing Data

In this Part we generate data.I used audacity to generate many samples and preprocess the data so that I get each audio sample length to 2 seconds.

- To do so, We set Sampling Rate to 16kHz.

Generating Data and Storing Data

In this Part we generate data.I used audacity to generate many samples and preprocess the data so that I get each audio sample length to 2 seconds.

- To do so, We set Sampling Rate to 16kHz.
- Generated 80 Samples of each command and saved them in respective folder.

Generating Data and Storing Data

In this Part we generate data.I used audacity to generate many samples and preprocess the data so that I get each audio sample length to 2 seconds.

- To do so, We set Sampling Rate to 16kHz.
- Generated 80 Samples of each command and saved them in respective folder.
- I used soundfile package to read .wav files.

Generating Data and Storing Data

In this Part we generate data.I used audacity to generate many samples and preprocess the data so that I get each audio sample length to 2 seconds.

- To do so, We set Sampling Rate to 16kHz.
- Generated 80 Samples of each command and saved them in respective folder.
- I used soundfile package to read .wav files.
- Now, we Store sound-data in data_x array and corresponding command in data_y array.

Table of Contents

Splitting the data

We need to divide data into train and test samples.So that we train the model with train data and test the accuracy with test data so that we can check performance of our model.

- We divided 20% of our input data(data_x,data_y) as test data and remaining data into train data

Splitting the data

We need to divide data into train and test samples.So that we train the model with train data and test the accuracy with test data so that we can check performance of our model.

- We divided 20% of our input data(data_x,data_y) as test data and remaining data into train data
- We Split input data into train,test data such that both train data and test data have nearly equal proportion of each commands.

Splitting the data

We need to divide data into train and test samples.So that we train the model with train data and test the accuracy with test data so that we can check performance of our model.

- We divided 20% of our input data(data_x,data_y) as test data and remaining data into train data
- We Split input data into train,test data such that both train data and test data have nearly equal proportion of each commands.
- To do so,we did Stratified Sampling.

Splitting the data

We need to divide data into train and test samples.So that we train the model with train data and test the accuracy with test data so that we can check performance of our model.

- We divided 20% of our input data(data_x,data_y) as test data and remaining data into train data
- We Split input data into train,test data such that both train data and test data have nearly equal proportion of each commands.
- To do so,we did Stratified Sampling.
- Stratified Sampling is performed instead of random Sampling because Stratified Sample can provide greater precision (or less biased data that is to get equal proportions of each command) than random sample of same size.

Table of Contents

Augmenting Data

Data augmentation is the process by which we create new synthetic training samples by adding small perturbations on our initial training set.

- The objective of augmentation is to is to make our model invariant to those perturbations and enhance its ability to generalize.

Augmenting Data

Data augmentation is the process by which we create new synthetic training samples by adding small perturbations on our initial training set.

- The objective of augmentation is to is to make our model invariant to those perturbations and enhance its ability to generalize.
- In order to this to work adding the perturbations must conserve the same label as the original training sample.

Augmenting Data

Data augmentation is the process by which we create new synthetic training samples by adding small perturbations on our initial training set.

- The objective of augmentation is to is to make our model invariant to those perturbations and enhance its ability to generalize.
- In order to this to work adding the perturbations must conserve the same label as the original training sample.
- In our case we did time-shifting, I Augment each audio sample by time shifting in 50,000 length vector filled with zeros.

Augmenting Data

Data augmentation is the process by which we create new synthetic training samples by adding small perturbations on our initial training set.

- The objective of augmentation is to is to make our model invariant to those perturbations and enhance its ability to generalize.
- In order to this to work adding the perturbations must conserve the same label as the original training sample.
- In our case we did time-shifting, I Augment each audio sample by time shifting in 50,000 length vector filled with zeros.
- I took steps of 1000 to create 18 files per sample.

Table of Contents

Extracting Features

Extraction of features is a very important part in analyzing and finding relations between different things.

- Sound data can be characterized with its frequencies.

Extracting Features

Extraction of features is a very important part in analyzing and finding relations between different things.

- Sound data can be characterized with its frequencies.
- Mel-frequency cepstral coefficients are most prominent features in feature extractions.

Extracting Features

Extraction of features is a very important part in analyzing and finding relations between different things.

- Sound data can be characterized with its frequencies.
- Mel-frequency cepstral coefficients are most prominent features in feature extractions.
- We will divide our data into segments of 1024 length and then we perform various operations and we end up with 39 mel-coefficients.

Extracting Features

Extraction of features is a very important part in analyzing and finding relations between different things.

- Sound data can be characterized with its frequencies.
- Mel-frequency cepstral coefficients are most prominent features in feature extractions.
- We will divide our data into segments of 1024 length and then we perform various operations and we end up with 39 mel-coefficients.
- So our data is ready for modelling!

Table of Contents

Model

.

- First we used Convolutional layers.

Model

.

- First we used Convolutional layers.
- After each CNN we added batch Normalization layers as Batch Normalization normalizes layer inputs on a per-feature basis and we know that neural networks train fast if the distribution of the input data remains similar over time.

Model

.

- First we used Convolutional layers.
- After each CNN we added batch Normalization layers as Batch Normalization normalizes layer inputs on a per-feature basis and we know that neural networks train fast if the distribution of the input data remains similar over time.
- Using Bi-directional LSTM's is optimal as we have complete data.

Model

.

- First we used Convolutional layers.
- After each CNN we added batch Normalization layers as Batch Normalization normalizes layer inputs on a per-feature basis and we know that neural networks train fast if the distribution of the input data remains similar over time.
- Using Bi-directional LSTM's is optimal as we have complete data.
- Final Output of LSTM is used to calculate importance of units of LSTM's using a Fully connected layer.

Model

.

- First we used Convolutional layers.
- After each CNN we added batch Normalization layers as Batch Normalization normalizes layer inputs on a per-feature basis and we know that neural networks train fast if the distribution of the input data remains similar over time.
- Using Bi-directional LSTM's is optimal as we have complete data.
- Final Output of LSTM is used to calculate importance of units of LSTM's using a Fully connected layer.
- Then we got Attention score at each step by doing dot-product between unit-importance and output sequences from LSTM further by doing dot-product with LSTM output sequences we get attention vectors.

Model

.

- First we used Convolutional layers.
- After each CNN we added batch Normalization layers as Batch Normalization normalizes layer inputs on a per-feature basis and we know that neural networks train fast if the distribution of the input data remains similar over time.
- Using Bi-directional LSTM's is optimal as we have complete data.
- Final Output of LSTM is used to calculate importance of units of LSTM's using a Fully connected layer.
- Then we got Attention score at each step by doing dot-product between unit-importance and output sequences from LSTM further by doing dot-product with LSTM output sequences we get attention vectors.
- Then applied fully connected layer with Softmax Activation, (Since we need to classify our data into 5 classes(multinomial)) Since we need to find the command.

Table of Contents

Training

- We fit our data to the model we prepared before.

Training

- We fit our data to the model we prepared before.
- We used Back Propagation algorithm in training our model.

Training

- We fit our data to the model we prepared before.
- We used Back Propagation algorithm in training our model.
- The Back Propagation Algorithm updates the present layers coefficients depending on errors in next layer .

Training

- We fit our data to the model we prepared before.
- We used Back Propagation algorithm in training our model.
- The Back Propagation Algorithm updates the present layers coefficients depending on errors in next layer .
- Here we used Adam Optimizer to update the coefficients.

Training

- We fit our data to the model we prepared before.
- We used Back Propagation algorithm in training our model.
- The Back Propagation Algorithm updates the present layers coefficients depending on errors in next layer .
- Here we used Adam Optimizer to update the coefficients.
- Adam Optimizer is combination of RMSprop and Stochastic gradient descent with momentum.It Uses both advantages of both methods.

Training

- We fit our data to the model we prepared before.
- We used Back Propagation algorithm in training our model.
- The Back Propagation Algorithm updates the present layers coefficients depending on errors in next layer .
- Here we used Adam Optimizer to update the coefficients.
- Adam Optimizer is combination of RMSprop and Stochastic gradient descent with momentum.It Uses both advantages of both methods.
- At the end of Training we will end up with Optimal Coefficients/solution.

Table of Contents

Testing

Inorder to know the performance of our model, we Need to test our model.

- we need to prepare test data similar to train data.(else we can keep a pipeline to do this task easily)

Testing

Inorder to know the performance of our model, we Need to test our model.

- we need to prepare test data similar to train data.(else we can keep a pipeline to do this task easily)
- Here we sent test data as validation data in model fitting.we can see model's performance after each epoch.

Testing

Inorder to know the performance of our model, we Need to test our model.

- we need to prepare test data similar to train data.(else we can keep a pipeline to do this task easily)
- Here we sent test data as validation data in model fitting.we can see model's performance after each epoch.
- we can observe that it is decreasing.

Table of Contents

Check attention

Here we build a sub-model from a trained model but we add Attention Soft-max layer as additional output layer.

- Now we pass our test data to our new model to the predict method.

Check attention

Here we build a sub-model from a trained model but we add Attention Soft-max layer as additional output layer.

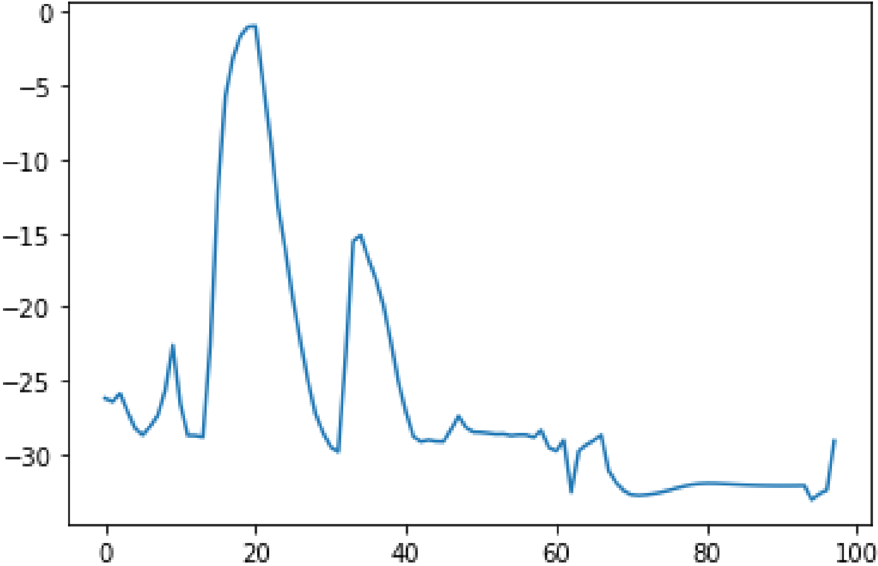
- Now we pass our test data to our new model to the predict method.
- Then we plotted log of Attention Scores and corresponding input vector before taking MFCC on other axes.

Check attention

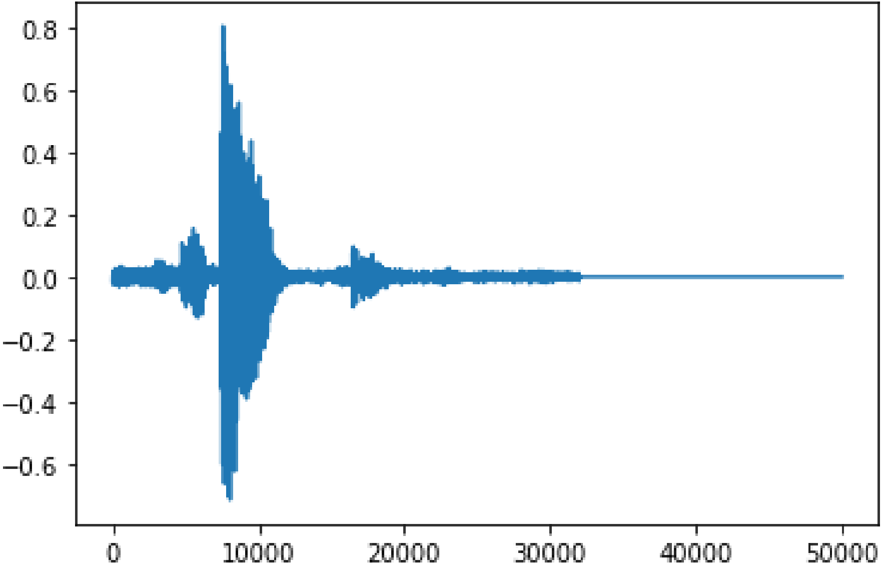
Here we build a sub-model from a trained model but we add Attention Soft-max layer as additional output layer.

- Now we pass our test data to our new model to the predict method.
- Then we plotted log of Attention Scores and corresponding input vector before taking MFCC on other axes.
- We can see that attention are high at high informative parts.

Check attention



Check attention



END

The – END