UDACITY

PROJECT SPECIFICATION

DNN Speech Recognizer

STEP 2: Model 0: RNN

CRITERIA	MEETS SPECIFICATIONS
Trained Model 0	The submission trained the model for at least 20 epochs, and none of the loss values in `model_0.pickle` are undefined. The trained weights for the model specified in `simple_rnn_model` are stored in `model_0.h5`.

STEP 2: Model 1: RNN + TimeDistributed Dense

CRITERIA	MEETS SPECIFICATIONS
Completed `rnn_model` Module	The submission includes a `sample_models.py` file with a completed `rnn_model` module containing the correct architecture.
Trained Model 1	The submission trained the model for at least 20 epochs, and none of the loss values in `model_1.pickle` are undefined. The trained weights for the model specified in `rnn_model` are stored in `model_1.h5`.

CRITERIA	MEETS SPECIFICATIONS

STEP 2: Model 2: CNN + RNN + TimeDistributed Dense

CRITERIA	MEETS SPECIFICATIONS
Completed `cnn_rnn_model` Module	The submission includes a `sample_models.py` file with a completed `cnn_rnn_model` module containing the correct architecture.
Trained Model 2	The submission trained the model for at least 20 epochs, and none of the loss values in `model_2.pickle` are undefined. The trained weights for the model specified in `cnn_rnn_model` are stored in `model_2.h5`.

STEP 2: Model 3: Deeper RNN + TimeDistributed Dense

CRITERIA	MEETS SPECIFICATIONS
Completed `deep_rnn_model` Module	The submission includes a `sample_models.py` file with a completed `deep_rnn_model` module containing the correct architecture.
Trained Model 3	The submission trained the model for at least 20 epochs, and none of the loss values in `model_3.pickle` are undefined. The trained weights for the model specified in `deep_rnn_model` are stored in `model_3.h5`.

CRITERIA	MEETS SPECIFICATIONS

STEP 2: Model 4: Bidirectional RNN + TimeDistributed Dense

CRITERIA	MEETS SPECIFICATIONS
Completed `bidirectional_rnn_model` Module	The submission includes a `sample_models.py` file with a completed `bidirectional_rnn_model` module containing the correct architecture.
Trained Model 4	The submission trained the model for at least 20 epochs, and none of the loss values in `model_4.pickle` are undefined. The trained weights for the model specified in `bidirectional_rnn_model` are stored in `model_4.h5`.

STEP 2: Compare the Models

CRITERIA	MEETS SPECIFICATIONS
Question 1	The submission includes a detailed analysis of why different models might perform better than others.

STEP 2: Final Model

CRITERIA	MEETS SPECIFICATIONS
Trained Final Model	The submission trained the model for at least 20 epochs, and none of the loss values in `model_end.pickle` are undefined. The trained weights for the model specified in `final_model` are stored in `model_end.h5`.
Completed `final_model` Module	The submission includes a `sample_models.py` file with a completed `final_model` module containing a final architecture that is not identical to any of the previous architectures.
Question 2	The submission includes a detailed description of how the final model architecture was designed.

Suggestions to Make Your Project Stand Out!

(1) Add a Language Model to the Decoder

The performance of the decoding step can be greatly enhanced by incorporating a language model. Build your own language model from scratch, or leverage a repository or toolkit that you find online to improve your predictions.

(2) Train on Bigger Data

In the project, you used some of the smaller downloads from the LibriSpeech corpus. Try training your model on some larger datasets - instead of using `dev-clean.tar.gz`, download one of the larger training sets on the website.

(3) Try out Different Audio Features

In this project, you had the choice to use *either* spectrogram or MFCC features. Take the time to test the performance of *both* of these features. For a special challenge, train a network that uses raw audio waveforms!

Student FAQ