Saarthi.AI - Report **Mohsin Ali**

Problem Statement

- For a given text, need to extract all the labels such as action that needs to be taken, action to be taken on which object and location where that object is present
- A kind of Multilabel Classification problem

Approach

- Designed a Deep learning model using Fast text embeddings and a LSTM layer, this model has 3 different sizes of output, as we were solving a multilabel classification problem.

Data

- Dataset contains a train_data.csv, valid_data.csv, having columns path of audio data, transcription of the audio data, action that needs to be taken, action to be taken on which object and location where that object is present

Preprocessing

- Removing HTML tags, if any
- Decontracting
- Lowering the words
- Removing punctuations and numericals
- Removing wide spaces

Tokenization

- Created a tokenizer with a vocab size of 5000
- Consider max_len = 10, as max len of text was 10
- Using pre padding and truncating
- Also, saving the tokenizer in a json format, which will be used in testing phase(Making sure no data leakage is happening)

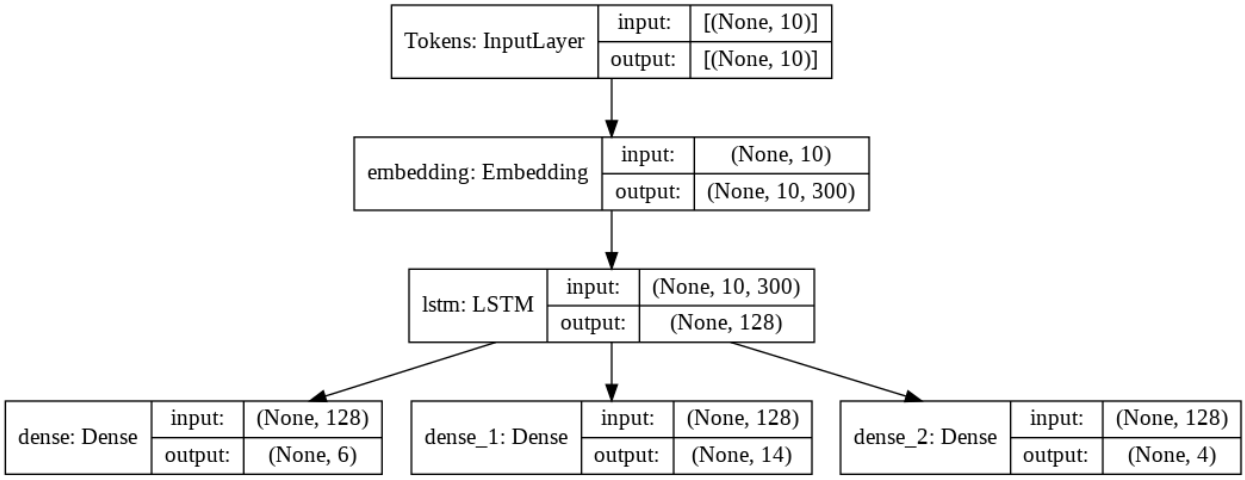
Label Encoding

- Defined a function called encoder, this function takes the labels(action, object and location) and returns encoded and binary encoded vectors respectively
- For, y_train_action_b and y_train_action, it is how it is encoded
- ```
  print(y_train_action_b[0], y_train_action[0])
  ```
  ```
  [0. 0. 1. 0. 0. 0.] 2
  ```
- Similarly, For object and location there will be the same kind of vectorization for train, test and validation

Model

- Input size for our model is (None, 30)
- The Embedding layer outputs a 300 dimension vectors for each token and these vectors are from pre-trained fasttext model, reason for using fasttext is because of out of vocabulary words(OOV), When a word is OOV, fasttext breaks it into sub-words, in the worst case it breaks into character level, where as other model like glove, google news were not
  ```

- These 300 dimension vectors are then passed to a 128 unit LSTM model, whose out is again passed to 3 different output models with activation as softmax



Training

- Multi-GPU training using MirroredStrategy, Which automatically detects number GPU's and this supports synchronous distributed training on multiple GPUs on one server. It creates one replica per GPU device. Each variable in the model is mirrored across all the replicas. These variables are kept in sync with each other by applying identical updates.
- Used callbacks like LearningRateScheduler and ReduceLROnPlateau to control the overfitting playing with learning rate
- Used TensorBoard to save logs and ModelCheckpoint to save model
- Some parameters are, Glorat normal initialization, Adam optimizer with learning rate 1e-4, loss as categorical_crossentropy, 50 epochs and batch size is 100

Testing

- After saving Tokinizer in Json format, Used it to vectorize test data
- Used save model to predict the output
- This testing file will take a test.csv file and return loss and accuracy

Results

- Model has very good results, both test and train accuracy is 100
- There was no overfitting observed