LOCATION PREDICTION FOR TWEETS

GROUP NAME - MAVERICK

Aim : To develop a deep learning based solution to predict geographic information for tweets.

Motivation

The current approaches bear two major limitations-

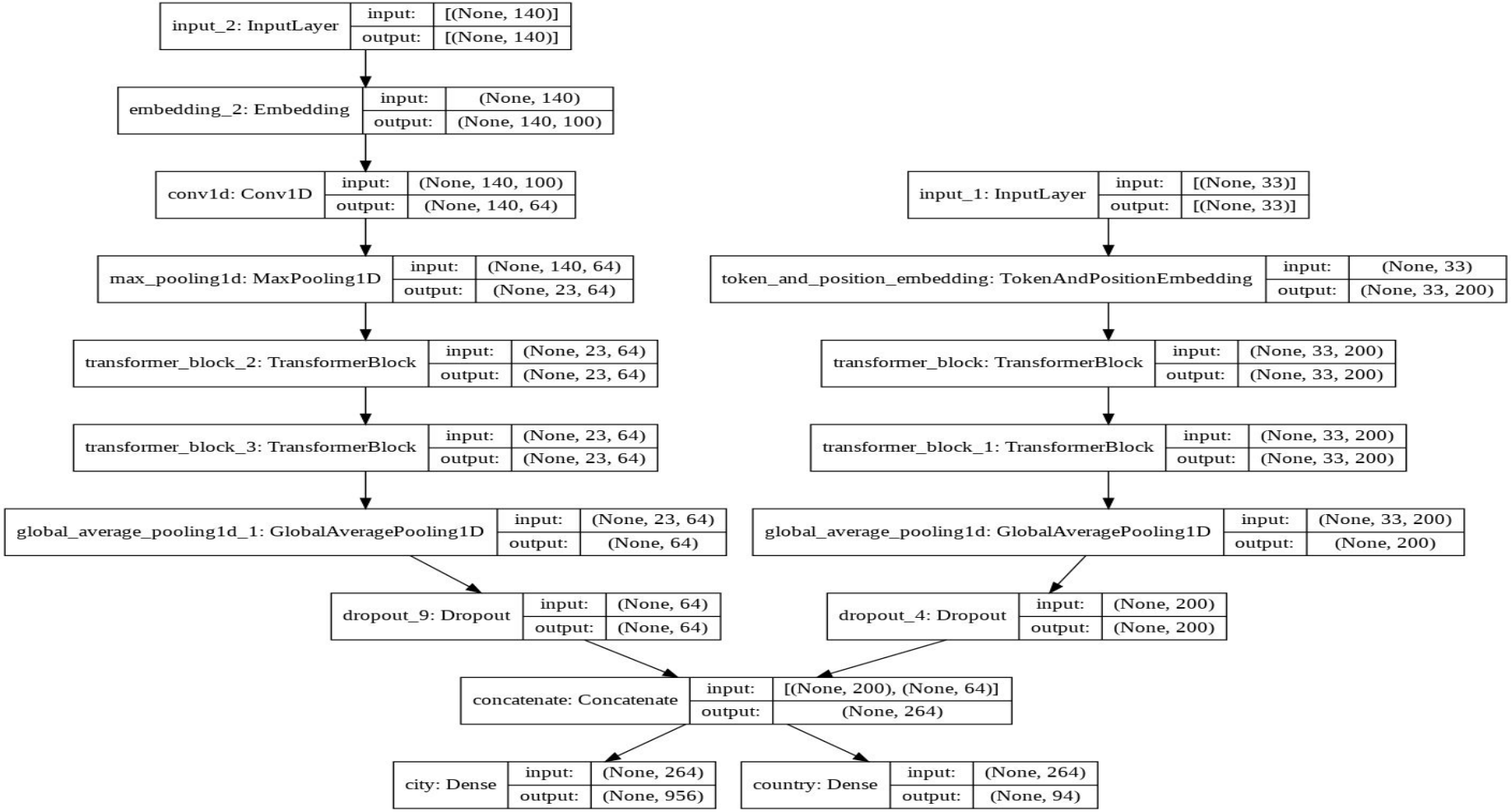
- hard to model the long term information.
- hard to explain to the end users what the model learns.

Applications

- Marketing Recommendation Systems
- Event Detection Systems

Methods Used

- Multi-head self Attention Mechanism
- Subword Feature
- Multitask Learning


input_2: InputLayer
input: [(None, 140)]
output: [(None, 140)]
embedding_2: Embedding
input: (None, 140)
output: (None, 140, 100)
conv1d: Conv1D
input: (None, 140, 100)
output: (None, 140, 64)
max_pooling1d: MaxPooling1D
input: (None, 140, 64)
output: (None, 23, 64)
transformer_block_2: TransformerBlock
input: (None, 23, 64)
output: (None, 23, 64)
transformer_block_3: TransformerBlock
input: (None, 23, 64)
output: (None, 23, 64)
global_average_pooling1d_1: GlobalAveragePooling1D
input: (None, 23, 64)
output: (None, 64)
dropout_9: Dropout
input: (None, 64)
output: (None, 64)
input_1: InputLayer
input: [(None, 33)]
output: [(None, 33)]
token_and_position_embedding: TokenAndPositionEmbedding
input: (None, 33)
output: (None, 33, 200)
transformer_block: TransformerBlock
input: (None, 33, 200)
output: (None, 33, 200)
transformer_block_1: TransformerBlock
input: (None, 33, 200)
output: (None, 33, 200)
global_average_pooling1d: GlobalAveragePooling1D
input: (None, 33, 200)
output: (None, 200)
dropout_4: Dropout
input: (None, 200)
output: (None, 200)
concatenate: Concatenate
input: [(None, 200), (None, 64)]
output: (None, 264)
city: Dense
input: (None, 264)
output: (None, 956)
country: Dense
input: (None, 264)
output: (None, 94)

Dataset

Training Example	5000
Testing Example	630
No of country	94
No of City	956

Testing Accuracy

Country Prediction	43.80 %
City Prediction	4.60 %

Training Accuracy

Country Prediction	58.80 %
City Prediction	7.02%

Additional Task Implemented

- Sandwich Transformer
- Early Stopping
- Adding Multihead Attention layer after Concatenation
- Adding a LSTM layer after Concatenation
- Having Multiple filters rather than single Convolution layer
- Hyperparameter Tuning

Sandwich Transformer

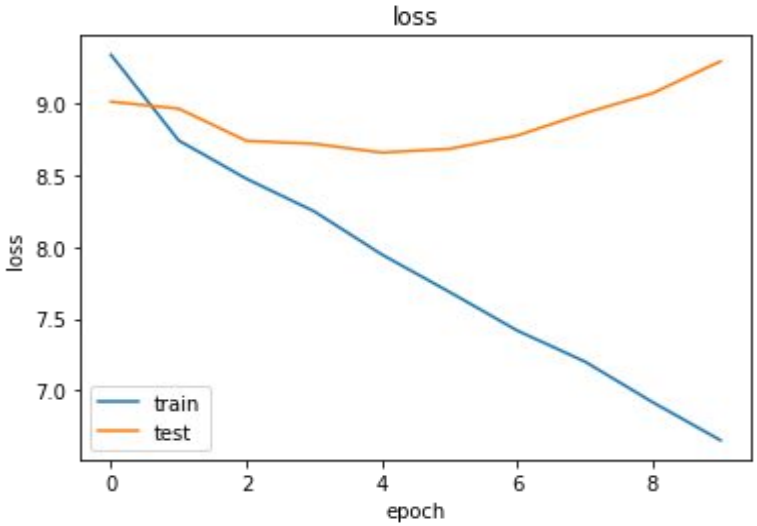
(a) Interleaved Transformer

(b) Sandwich Transformer

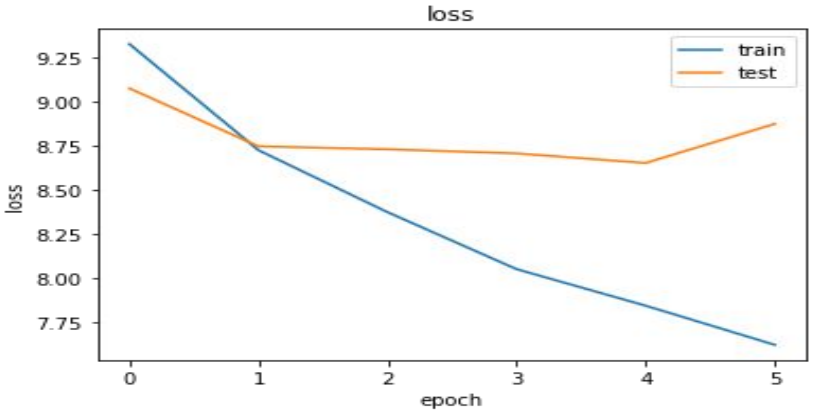
[Improving Transformer Models by Reordering their Sublayers]

Early Stopping

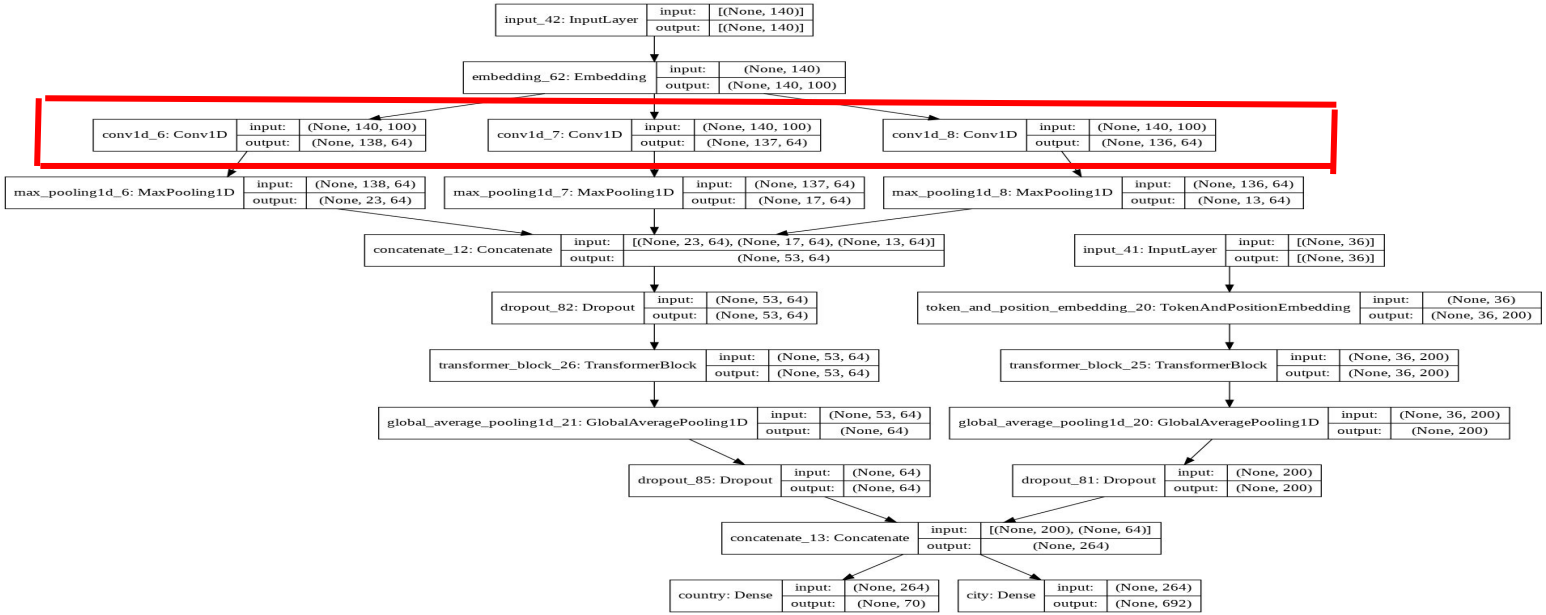
Without Early Stopping



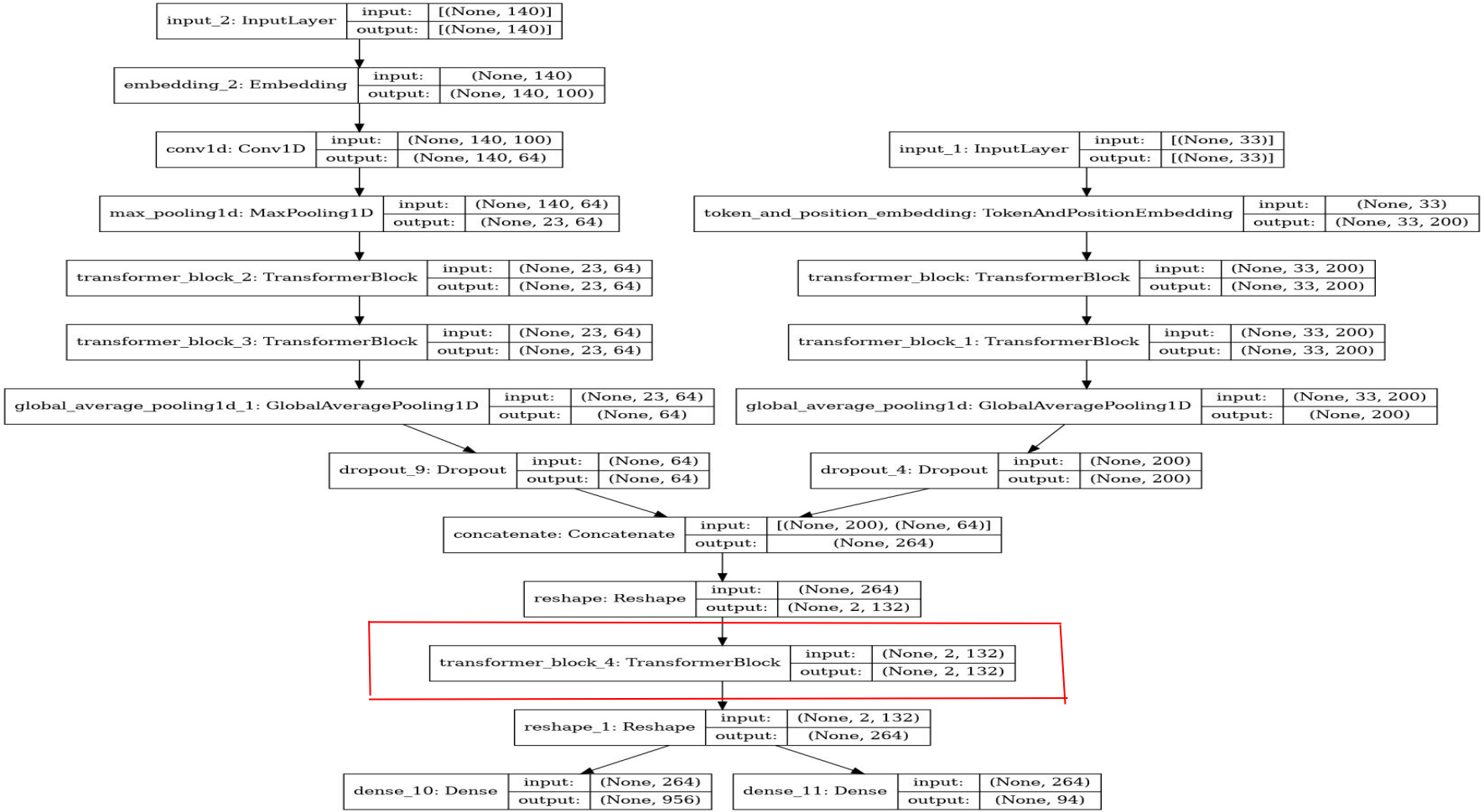
With Early Stopping



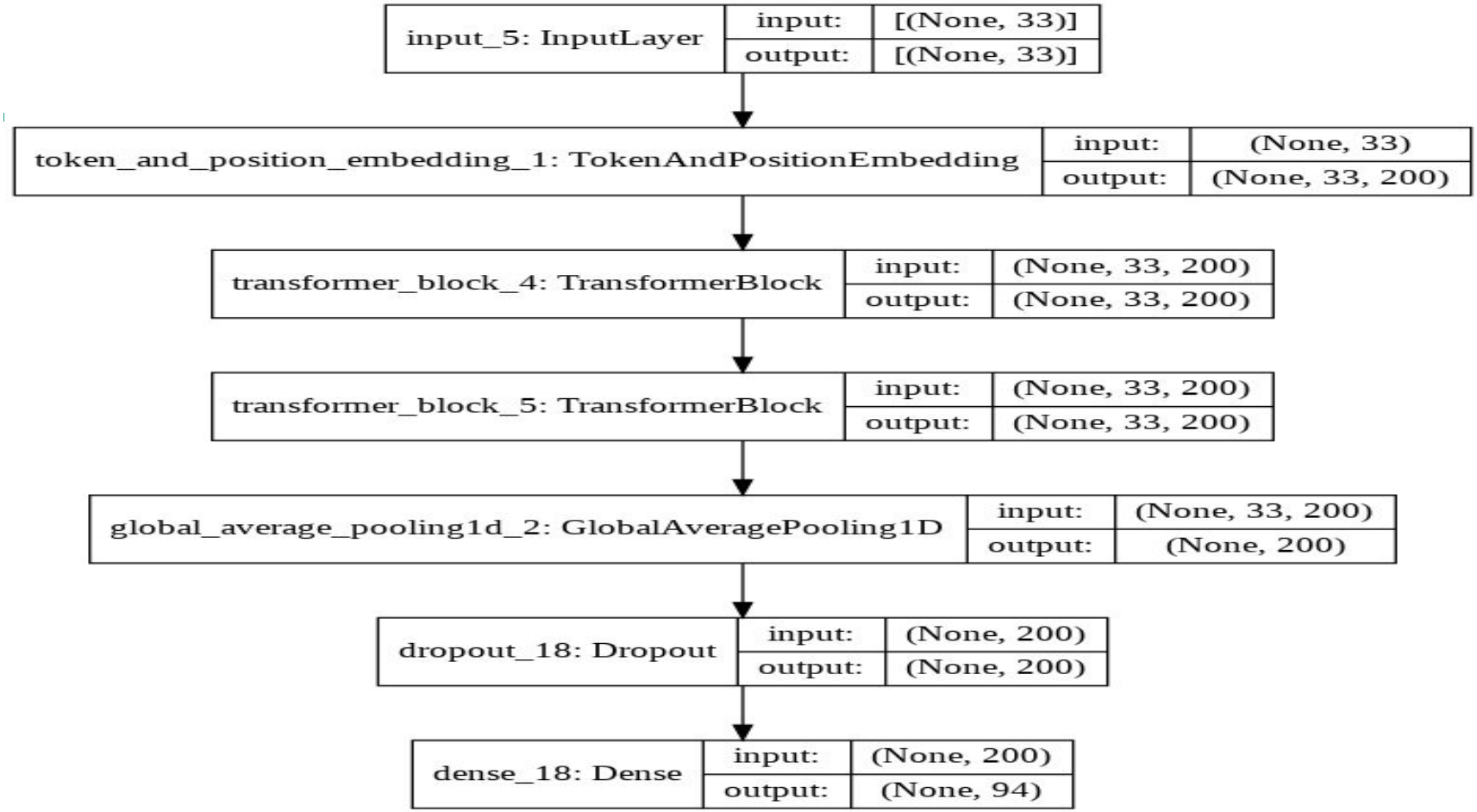
Having Multiple Parallel convolution layer with different filter sizes rather than single Convolution layer



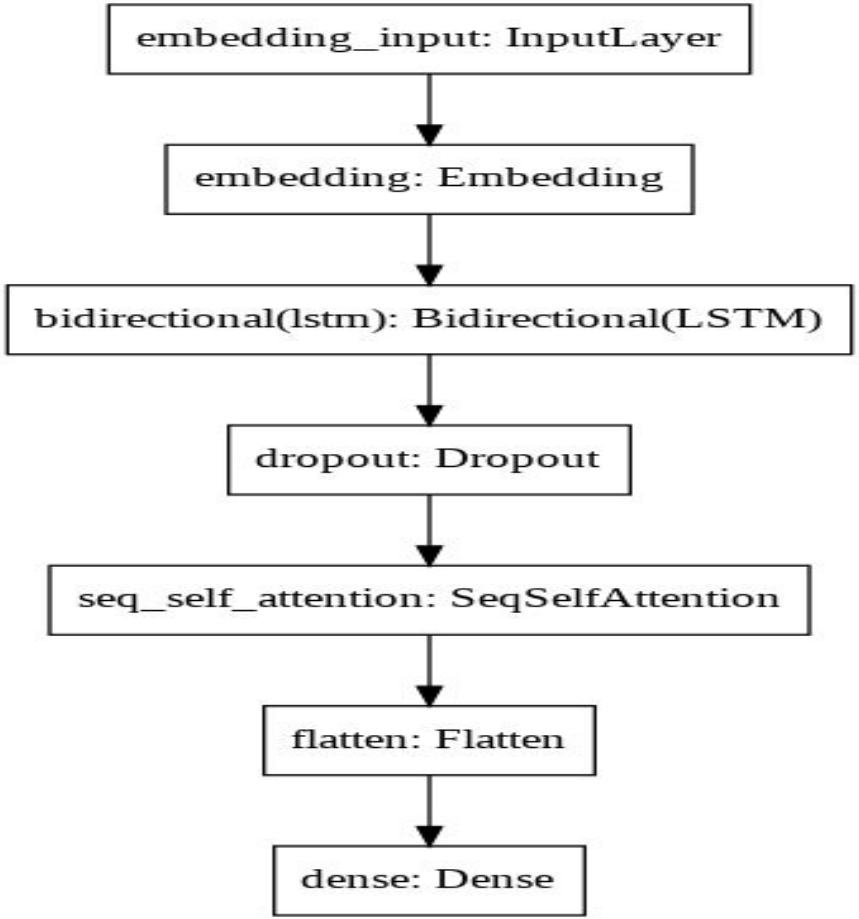
Adding LSTM Layer / Multihead Attention Layer after Concatenation


input_2: InputLayer
input: [(None, 140)]
output: [(None, 140)]
embedding_2: Embedding
input: (None, 140)
output: (None, 140, 100)
conv1d: Conv1D
input: (None, 140, 100)
output: (None, 140, 64)
max_pooling1d: MaxPooling1D
input: (None, 140, 64)
output: (None, 23, 64)
transformer_block_2: TransformerBlock
input: (None, 23, 64)
output: (None, 23, 64)
transformer_block_3: TransformerBlock
input: (None, 23, 64)
output: (None, 23, 64)
global_average_pooling1d_1: GlobalAveragePooling1D
input: (None, 23, 64)
output: (None, 64)
dropout_9: Dropout
input: (None, 64)
output: (None, 64)
input_1: InputLayer
input: [(None, 33)]
output: [(None, 33)]
token_and_position_embedding: TokenAndPositionEmbedding
input: (None, 33)
output: (None, 33, 200)
transformer_block: TransformerBlock
input: (None, 33, 200)
output: (None, 33, 200)
transformer_block_1: TransformerBlock
input: (None, 33, 200)
output: (None, 33, 200)
global_average_pooling1d: GlobalAveragePooling1D
input: (None, 33, 200)
output: (None, 200)
dropout_4: Dropout
input: (None, 200)
output: (None, 200)
concatenate: Concatenate
input: [(None, 200), (None, 64)]
output: (None, 264)
reshape: Reshape
input: (None, 264)
output: (None, 2, 132)
transformer_block_4: TransformerBlock
input: (None, 2, 132)
output: (None, 2, 132)
reshape_1: Reshape
input: (None, 2, 132)
output: (None, 264)
dense_10: Dense
input: (None, 264)
output: (None, 956)
dense_11: Dense
input: (None, 264)
output: (None, 94)

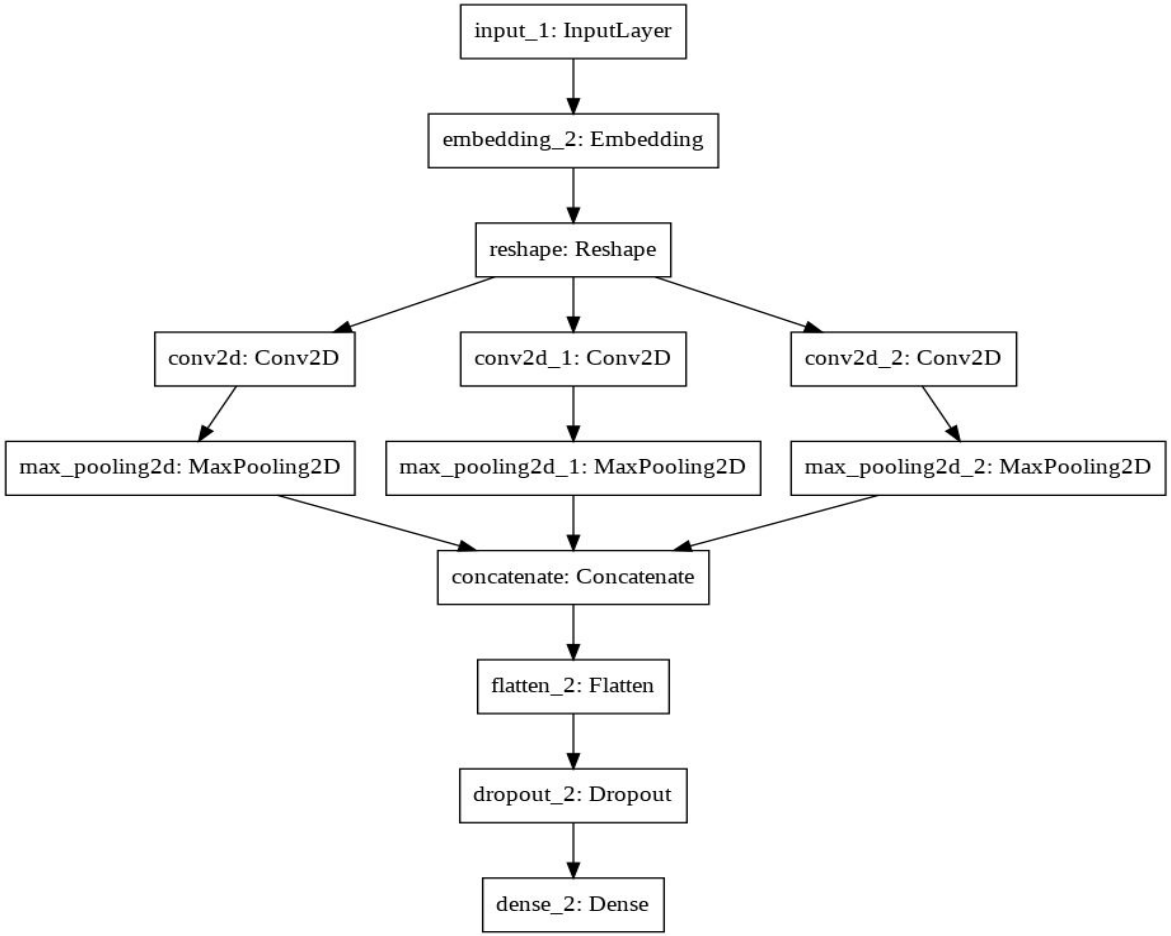
Other Types of model


input_5: InputLayer
input:
[(None, 33)]
output:
[(None, 33)]
token_and_position_embedding_1: TokenAndPositionEmbedding
input:
(None, 33)
output:
(None, 33, 200)
transformer_block_4: TransformerBlock
input:
(None, 33, 200)
output:
(None, 33, 200)
transformer_block_5: TransformerBlock
input:
(None, 33, 200)
output:
(None, 33, 200)
global_average_pooling1d_2: GlobalAveragePooling1D
input:
(None, 33, 200)
output:
(None, 200)
dropout_18: Dropout
input:
(None, 200)
output:
(None, 200)
dense_18: Dense
input:
(None, 200)
output:
(None, 94)

LSTM + Self Attention



CNN


input_1: InputLayer
embedding_2: Embedding
reshape: Reshape
conv2d: Conv2D
conv2d_1: Conv2D
conv2d_2: Conv2D
max_pooling2d: MaxPooling2D
max_pooling2d_1: MaxPooling2D
max_pooling2d_2: MaxPooling2D
concatenate: Concatenate
flatten_2: Flatten
dropout_2: Dropout
dense_2: Dense

Accuracy

	Country Prediction (trained on 5000 examples)	City Prediction (trained on 5000 examples)	Country Prediction (trained on 30,000 examples)	City Prediction (trained on 30,000 examples)
Without any modification	43.80 %	4.60%	48%	**6.7%**
Sandwich Transformer	44.44%	3.9%	48.3%	5.93%
Extra transformer block after concatenation	45.55%	4.92%	**49.03%**	5.50%
Extra transformer layer after concatenation + multiple convolution parallel layer	44.8%	5.8%	**50.22%**	**6.3%**
LSTM + self attention	40%	4.76%	46.19%	6.14%
CNN	43.17%	3.49%	**49.45%**	5.54%
Only word embedding of the given model used (right part only)	45.08%	2.7%	46.24%	5.41%

Thanks!