

LSTM

```
11 model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['acc','f1_m',precision_m, recall_m])
12
13 model.summary()
```

Model: "model"

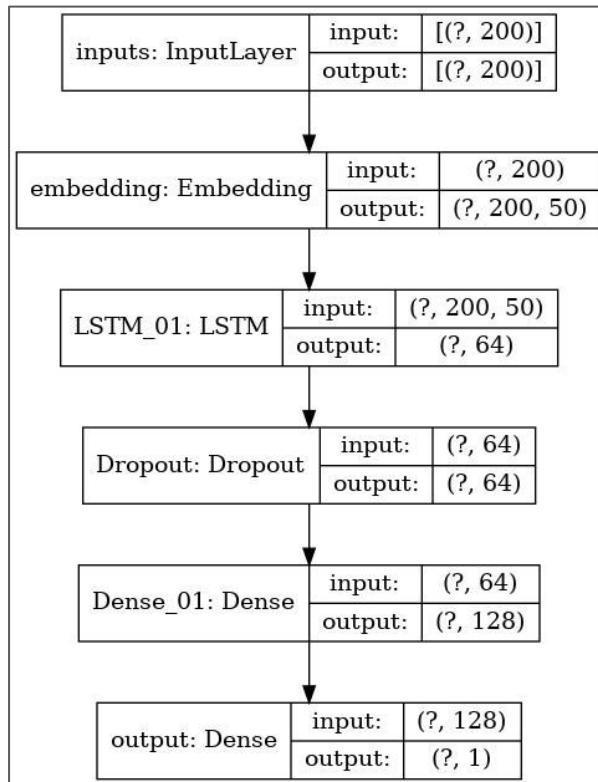
| Layer (type) | Output Shape | Param # |
|-----------------------|-----------------|---------|
| ===== | | |
| inputs (InputLayer) | [(None, 200)] | 0 |
| embedding (Embedding) | (None, 200, 50) | 100000 |
| LSTM_01 (LSTM) | (None, 64) | 29440 |
| Dropout (Dropout) | (None, 64) | 0 |
| Dense_01 (Dense) | (None, 128) | 8320 |
| output (Dense) | (None, 1) | 129 |

=====

Total params: 137,889

Trainable params: 137,889

Non-trainable params: 0



CNN

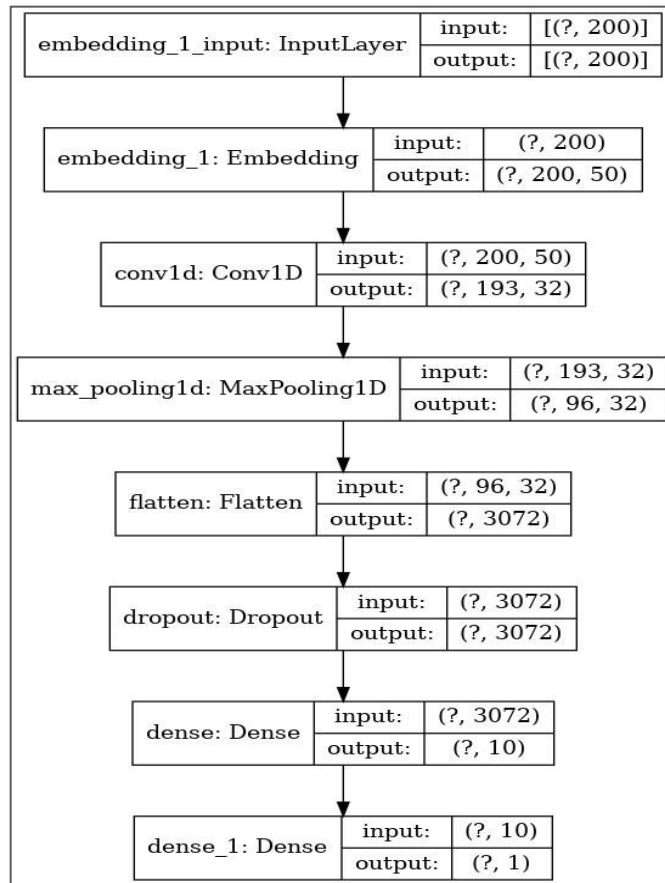
```
10 model_CNN.compile(loss='binary_crossentropy',
11                   optimizer='adam',
12                   metrics=['acc','f1_m',precision_m, recall_m])
13
14 model_CNN.summary()
```

Model: "CNN_with_embeddings"

| Layer (type) | Output Shape | Param # |
|------------------------------|-----------------|---------|
| ===== | | |
| embedding_1 (Embedding) | (None, 200, 50) | 100000 |
| ----- | | |
| conv1d (Conv1D) | (None, 193, 32) | 12832 |
| ----- | | |
| max_pooling1d (MaxPooling1D) | (None, 96, 32) | 0 |
| ----- | | |
| flatten (Flatten) | (None, 3072) | 0 |
| ----- | | |
| dropout (Dropout) | (None, 3072) | 0 |
| ----- | | |
| dense (Dense) | (None, 10) | 30730 |
| ----- | | |
| dense_1 (Dense) | (None, 1) | 11 |
| ===== | | |

Total params: 143,573

Trainable params: 143,573



BiLSTM

```
9
10 bi_model.compile(loss='binary_crossentropy',
11                  optimizer='adam',
12                  metrics=['acc',f1_m,precision_m, recall_m])
13
14 bi_model.summary()
```

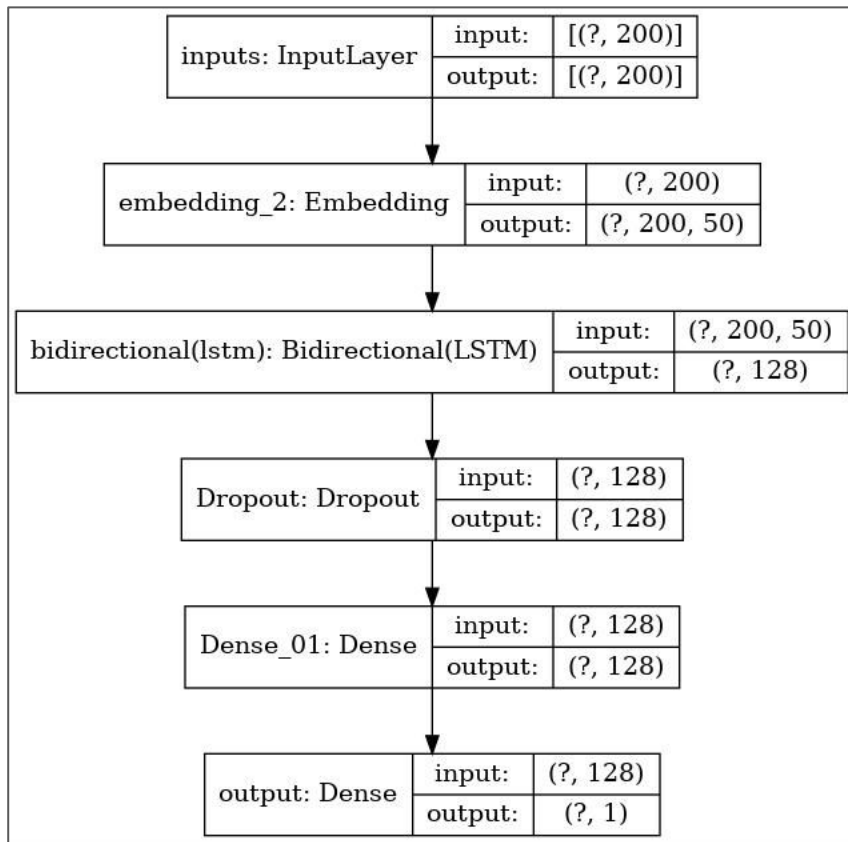
Model: "model_1"

| Layer (type) | Output Shape | Param # |
|--------------------------------------|-----------------|---------|
| ===== | | |
| inputs (InputLayer) | [(None, 200)] | 0 |
| ----- | | |
| embedding_2 (Embedding) | (None, 200, 50) | 100000 |
| ----- | | |
| bidirectional (Bidirectional (LSTM)) | (None, 128) | 58880 |
| ----- | | |
| Dropout (Dropout) | (None, 128) | 0 |
| ----- | | |
| Dense_01 (Dense) | (None, 128) | 16512 |
| ----- | | |
| output (Dense) | (None, 1) | 129 |
| ===== | | |

Total params: 175,521

Trainable params: 175,521

Non-trainable params: 0



LSTM + Attention

```

4 m.summary()
5 m.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc', f1_m, precision_m, recall_m])

```

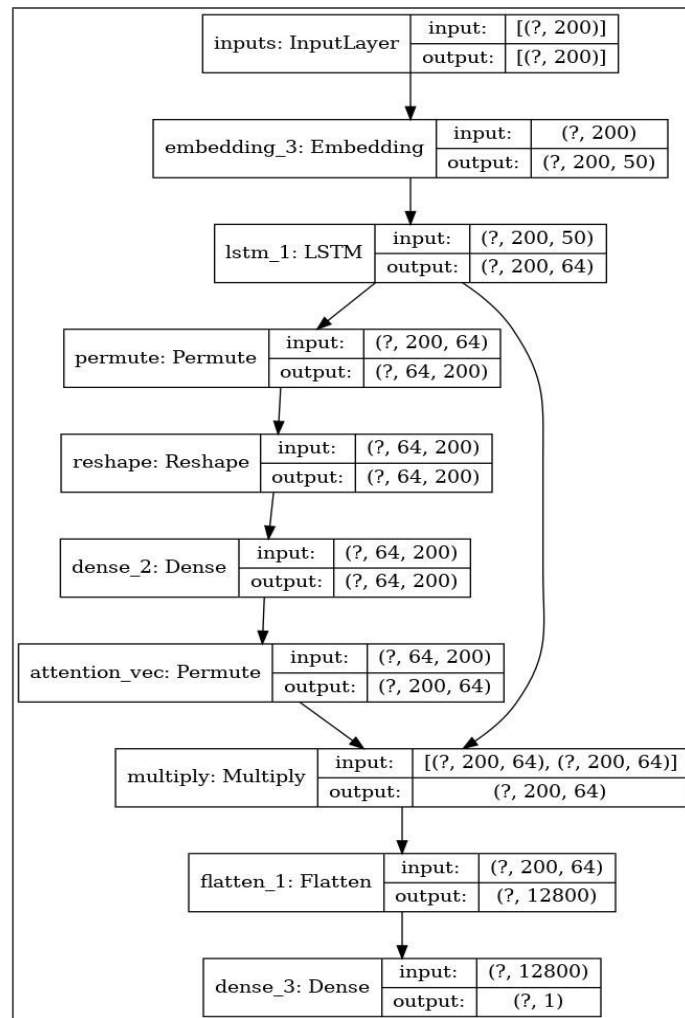
Model: "model_2"

| Layer (type) | Output Shape | Param # | Connected to |
|-------------------------|-----------------|---------|-------------------------------------|
| ===== | | | |
| inputs (InputLayer) | [(None, 200)] | 0 | |
| embedding_3 (Embedding) | (None, 200, 50) | 100000 | inputs[0][0] |
| lstm_1 (LSTM) | (None, 200, 64) | 29440 | embedding_3[0][0] |
| permute (Permute) | (None, 64, 200) | 0 | lstm_1[0][0] |
| reshape (Reshape) | (None, 64, 200) | 0 | permute[0][0] |
| dense_2 (Dense) | (None, 64, 200) | 40200 | reshape[0][0] |
| attention_vec (Permute) | (None, 200, 64) | 0 | dense_2[0][0] |
| multiply (Multiply) | (None, 200, 64) | 0 | lstm_1[0][0] attention_vec[0][0] |
| flatten_1 (Flatten) | (None, 12800) | 0 | multiply[0][0] |
| dense_3 (Dense) | (None, 1) | 12801 | flatten_1[0][0] |
| ===== | | | |

Total params: 182,441

Trainable params: 182,441

Non trainable params: 0



Baseline Transformer Models

BERT model

```
model_args = ClassificationArgs()
model_args.num_train_epochs = 10
model_args.train_batch_size = 160
model_args.eval_batch_size = 160
model_args.dataloader_num_workers = 4
model_args.overwrite_output_dir = True
model_args.max_seq_length = 200
model_args.output_dir = 'baseline_bert_model/'
model_args.learning_rate = 5e-05
model_args.manual_seed = 42
model_args.warmup_ratio = 0.07
model_args.weight_decay = 0.05

model = ClassificationModel("bert", "bert-base-cased", num_labels=2, args=model_args, cuda_device=12)

# Train the model
model.train_model(train_df, acc = sklearn.metrics.accuracy_score)
```

BERT - Code Snippet

Roberta model

```
model_args = ClassificationArgs()
model_args.num_train_epochs = 10
model_args.train_batch_size = 150
model_args.eval_batch_size = 150
model_args.dataloader_num_workers = 4
model_args.overwrite_output_dir = True
model_args.max_seq_length = 200
model_args.output_dir = 'baseline_roberta_model/'
model_args.learning_rate = 5e-05
model_args.manual_seed = 42
model_args.warmup_ratio = 0.07
model_args.weight_decay = 0.05

model = ClassificationModel("roberta", "roberta-base", num_labels=2, args=model_args, cuda_device=13)

# Train the model
model.train_model(train_df, acc = sklearn.metrics.accuracy_score)
```

RoBERTa - Code Snippet