

DeepLearning.AI Tensorflow Developer

Natural Language Processing in TensorFlow: Week 3

Sequence Models

RNN uses input from previous state in sequence model. On the other hand, LSTM preserves the context of a text sequence from earlier state. The mechanism is performed by cell state which can be bidirectional. LSTM layer takes the number of outputs as input and can be wrapped around bidirectional layer which doubles the output. LSTM layers can be stacked on top of each other with return_sequence parameter set to TRUE in first LSTM layer.

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(None,)),
    tf.keras.layers.Embedding(subword_tokenizer.vocabulary_size(), EMBEDDING_DIM),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(LSTM1_DIM, return_sequences=True)),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(LSTM2_DIM)),
    tf.keras.layers.Dense(DENSE_DIM, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Multi-layer LSTM model makes accuracy and loss curve smoother with less spikes. LSTM performs well with text classification although validation loss increases over epoch which is called overfitting. We can also use Conv1D layer followed by GlobalAveragePooling1D to perform the task with specified convolutions, filter size and activation function.

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(None,)),
    tf.keras.layers.Embedding(subword_tokenizer.vocabulary_size(), EMBEDDING_DIM),
    tf.keras.layers.Conv1D(filters=FILTERS, kernel_size=KERNEL_SIZE, activation='relu'),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Dense(DENSE_DIM, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')])
```

Even GRU layer wrapped with bidirectional layer creates overfitting. Validation set words missing in the vocabulary can create overfitting.