Enhancing AI Generated Text with LSTM

Amber Wells, Bryan Gonzales , Shain Dholakiya

OUR GOALS

What is an LSTM

Tech Used

Implementation of LSTM

Future

Questions

What is an LSTM

- Recurrent Neural Network (RNN) → multiple copies of the same network which each has loops within them which allows for past information to be remember
- Downfall of RNN → unable to learn long-term dependencies
- Long Short Term Memory (LSTM) → an improved RNN which can handles these long-term dependencies
- Unlike RNN LSTM has four layers (while RNN only has one layer) of neural networks

(Olah, 2015)

Tech Used

Pycharm

Amazing interface making it easy to connect your python code to TensorFlow

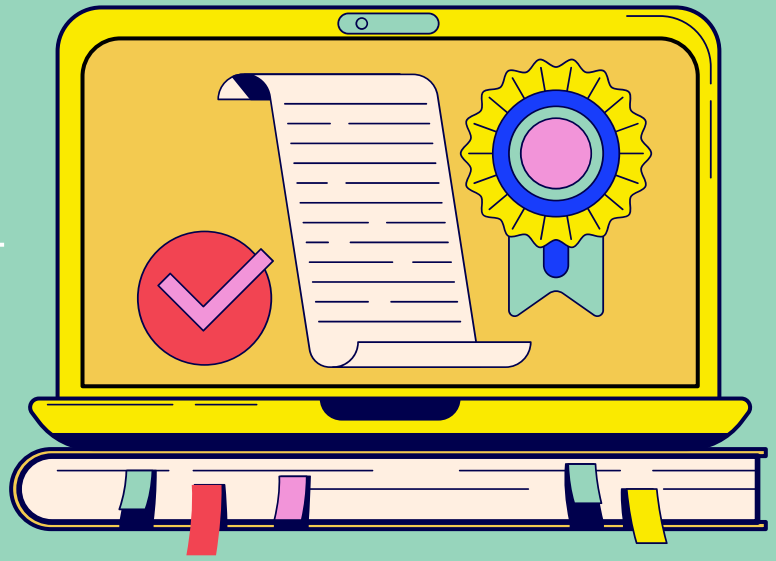
TensorFlow - Keras

A well known service that serves to help everyone get into machine learning

Implementation

Load the data & prep for training

Define the LSTM



Fit the model

Train!

Loading the Data

```
rawText = open("storyData.txt", 'r',
encoding='utf-8').read()
rawText = rawText.lower()

# sorts the characters then adds them into an
dictionary where they are numbered
sortedChars = sorted(list(set(rawText)))
charDictionary = dict((char, i) for i, char in
enumerate(sortedChars))
n_chars = len(rawText)
n_vocab = len(sortedChars)
```

- Take out anything you don't want your AI to learn → could increase learning time
- Give all the chars int values

Prepare the dataset

```
# prepare the dataset of input to output
pairs encoded as integers
seqLength = 100
dataX = []
dataY = []
for i in range(0, n_chars - seqLength, 1):
        seq_in = rawText[i:i + seqLength]
        seq_out = rawText[i + seqLength]
        dataX.append([charDictionary[char]
for char in seq_in])
        dataY.append(charDictionary[seq_out])
n_patterns = len(dataX)
```

- We split up the data here and look at the patterns that are made
- Our next step is to set it up to work with Keras but we will skip this for today

Define LSTM

```
model = Sequential()
model.add(Bidirectional(LSTM(256,
activation="relu"), input_shape=(X.shape[1],
X.shape[2])))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(256)))
model.add(Dropout(0.2))
model.add(Dense(y.shape[1],
activation='softmax'))
callbacks = [EarlyStopping(patience=2,
monitor='val)loss')]
model.compile(loss='categorical_crossentropy',
optimizer='adam',
metrics=[categorical_accuracy])
```

- Setting up a multi-layered bidirectional LSTM
- Bidirectional → trains two instead of one LSTM
- More layers → should mean that that our data should be more accurate

Fit the Model!

```
filepath = "{epoch:02d}-loss-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath,
monitor='loss', verbose=1, save_best_only=True,
mode='min')
callbacks_list = [checkpoint]
ModelCheckpoint(filepath, monitor='loss',
verbose=1, save_best_only=True, mode='min')]

model.fit(X, y, batch_size=64, shuffle=True,
epochs=20, callbacks=callbacks_list,
validation_split=0.1)
```

- This makes it so we can start to run more test!

Future Goals

More Tests!

More tests mean better data and better stories!

Set up Temperature

The temperature determines predictable the text will be

Get a good output

We still need get to where we want to be

Stronger Processing

More power = bigger tests

QUESTIONS?

RESOURCES

Brownlee, J., 202. *Text Generation With LSTM Recurrent Neural Networks In Python with Kears*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/> [Accessed 2 December 2020].

Champion, D., 2018. *Text Generation Using Bidirectional LSTM And Doc2vec Models 1/3.*. [onlint] Medium Available at: <https://medium.com/@david.campion/text-generation-using-bidirectional-lstm-and-doc2vec-models-1-3-8979eb65cb3a> [Accessed 2 December 2020].

Olah, C., 2015. *Understanding LSTM Networks*. [online] Colah's Blog. Available at: <https://colah.github.io/about.html> [Accessed 2 December 202]

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**