SUMMARY OF COURSERA COURSE

NATURAL LANGUAGE PROCESSING IN TENSORFLOW

RATINGS: /5

WEEK 1 – SENTIMENT IN TEXT

UNIT 1: Introduction, a conversation with Andrew Ng

- Using TF to process text, it is messier than images
- NN deal with numbers, so how do we convert text to numbers
- Load text and preprocess it

UNIT 2: Introduction

- Build classifier based on text models

UNIT 3: Word based encodings

- Training a NN with just the character encoding will not help us understand the meaning of a word
- We can train based on word enodings, Keras and TF gives us some APIs to implement this

UNIT 4: Using APIs

- "tensorflow.keras.preprocessing.text import Tokenizer", it turns the text to vector values
- Word based encoding

UNIT 5: Check out the code

- Downloaded "Course_3_Week_1_Lesson_1.ipynb"
- Created "Course_3_Week_1_1.ipynb"

UNIT 6: Notebook for lesson 1

- A "tokenizer" is your friend when doing NLP, it makes your work easier when training the NN
- Punctuations and spaces are removed by the Tokenizer

UNIT 7: Text to sequence

- Tokens are used to replace the words
- "tokenizer.texts_to_sequence()"

UNIT 8: Looking more at the Tokenizer

- We need a lot of training data to get a broad vocabulary

- Use a token for words that are not in the vocabulary
- The input size for text being loaded into the NN must have a level uniformity of size, we use padding to achieve this

UNIT 9: Padding

- "tensorflow.keras.preprocessing.sequence import pad_sequences"
- "padding = post", to add the padding at the end

UNIT 10: Check out the code!

- Downloaded "Course_3_Week_1_Lesson_2.ipynb"

UNIT 11: Notebook for Lesson 2

- Padding helps to get the sequences to the same length

UNIT 12: Sarcasm, really?

- Sarcasm in News Dataset on Kaggle

UNIT 13: Working with the Tokenizer

UNIT 14: News headlines dataset for sarcasm detection

UNIT 15: Check out the code

- Downloaded "Course_3_Week_1_Lesson_3.ipynb"

UNIT 16: Notebook for Lesson 3

UNIT 17: Week 1 Quiz

UNIT 18: Week 1 Wrap up

- We looked into how to tokenize words, turning them into numbers and using a dictionary to look up which word goes with which number

UNIT 19: Ungraded External Tool – Explore the BBC news archive

WEEK 2 – WORD EMBEDDINGS

UNIT 1: A conversation with Andrew Ng

- Keras layer in TF to implement "word embeddings"
- It is one of the most important ideas in NLP
- You can download a pre-trained word embeddings
- Word embeddings are very important for Applied NLP

UNIT 2: Introduction

- Word Embeddings, words with similar meanings are clustered together in n-dimensional space

UNIT 3: The IMDB Dataset

- Part of the vision of TF to make ML and DL easier to learn and easier to use is the concept of having built-in datasets
- TFDS: TensorFlow Dataset System

UNIT 4: IMDB reviews dataset

UNIT 5: Looking into the details

- "tf.keras.layers.Embedding()"

UNIT 6: How can we use vectors?

- Word embeddings projecting words as vectors into n-dimensional space based on their label (meaning) and words with similar labels (vectors) tend to cluster together

UNIT 7: More into the details

UNIT 8: Check out the code

- Downloaded "Course_3_Week_2_Lesson_1.ipynb"
- Created "Course_3_Week_2_1.ipynb"

UNIT 9: Notebook for lesson 1

- The TF Embedding Projector: https://projector.tensorflow.org/

UNIT 10: Remember the sarcasm dataset

- Build a classifier for the sarcasm dataset

UNIT 11: Building a classifier for the sarcasm dataset

UNIT 12: Let's talk about the loss function

- Tweak the hyper parameters to explore for optimal choice
- Setting the hyper parameters as variables make them easier to tweak

UNIT 13: Check out the code

- Downloaded "Course_3_Week_2_Lesson_2.ipynb"

UNIT 14: Pre-tokenized datasets

- A version of the IMDB dataset that has been tokenized using sub words

UNIT 15: TensorFlow datasets

- Check out: https://www.tensorflow.org/datasets/overview

UNIT 16: Diving into the code part 1

- Using a sub-word text encoded version of the IMDB reviews dataset

UNIT 17: Sub-words text encoder

- Check out: https://www.tensorflow.org/datasets/api_docs/python/tfds/features/text/SubwordTextEncoder

UNIT 18: Diving into the code part 2

- Instead of "Flatten()" we use "GlobalAveragePooling1D()"
- Sub-words tokens are nonsensical; they only make sense when in a sequence

UNIT 19: Check out the code

- Ran on colab and downloaded "Course_3_Week_2_Lesson_3.ipynb"

UNIT 20: Notebook for lesson 3

- Sub words only make sense in a sequence, with RNNs sub words can be a very powerful tool

UNIT 21: Quiz

UNIT 22: Week 2 Wrap Up

- Not only do the meanings of the words matter, but also the sequence in which they are found is very important

UNIT 23: Exercise 2

WEEK 3 – SEQUENCE MODELS

UNIT 1: A conversation with Andrew Ng

- The sequence of words affects the meaning
- RNN, LSTM, GRU help with this

UNIT 2: Introduction

- RNNs work like Fibonacci numbers where previous output acts as input for the next output

UNIT 3: Link to Andrew's sequence modelling course

UNIT 4: LSTMs

- LSTM is an update to RNNs, it helps to keep context form earlier tokens relevant in later context

UNIT 5: More info on LSTMs

UNIT 6: Implementing LSTMs in code

- We can stack "LSTM layers"
- "return sequence = True", for the first LSTM layer if we are stacking LSTMs

UNIT 7: Check out the code

- Downloaded "Course_3_Week_3_Lesson_1a.ipynb" and "Course_3_Week_3_Lesson_1b.ipynb"
- Created "Course_3_Week_3_1.ipynb"

UNIT 8: Accuracy and loss

- The jaggyness of our accuracy and loss graph can show that out model needs to be tuned better
- Explore some other RNN types

UNIT 9: A word from Lawrence

- Some other options of RNNs, Conv1D, GRUs

UNIT 10: Looking into the code

- Replacing the "Flatten/ Global1DPooling" using RNNs like LSTM
- Be careful to adjust your training parameter when working with different NN type/architecture

UNIT 11: Using a Convolutional Network

- We can use a Convolution layers "Conv1D"

UNIT 12: Check out the code

- Ran and Downloaded "Course_3_Week_3_Lesson_1c.ipynb"

UNIT 13: Going back to the IMDB dataset

- The model takes more time when we add RNN layers to it
- "Conv1D", "LSTM", "GRU"

UNIT 14: Check out the code

- Ran and Downloaded "Course_3_Week_3_Lesson_2d.ipynb"

UNIT 15: Tips from Lawrence

- Try out the different options for the RNN architecture, you are more susceptible to overfitting with text than with images

UNIT 16: Exploring different sequence models

- Explore LSTM and Conv1D on the sarcasm news detector
- Ran and Downloaded "Course_3_Week_3_Lesson_2.ipynb"

UNIT 17: Quiz

UNIT 18: Week 3 Wrap up

- Next week we will try to generate text

UNIT 19: Exercise 3 – Exploring overfitting in NLP

- We can do transfer learning for Embedding (e.g GloVe)

WEEK 4 – SEQUENCE MODELS AND LITERATURE

UNIT 1: A conversation with Andrew Ng

- Sequence models can synthesize new text based on the training vocabulary, for example generation Shakespeare text

UNIT 2: Introduction

- Generation of new text
- Using phrases as our X and the next word as our Y

UNIT 3: Looking into the code

UNIT 4: Training the data

- Taking the last word in the sequence as the "label"

UNIT 5: More on training the data

- "tf.keras.utils.to_categorical(labels, num_classes)": used to one hot encode our labels

UNIT 6: Check out the code

- Ran and Downloaded "Course_3_Week_4_Lesson_1_Notebook.ipynb"

UNIT 7: Notebook for lesson 1

UNIT 8: Finding what the next word should be

UNIT 9: Example

- Try using “Bidirectional(LSTM())”

UNIT 10: Predicting a word

- The more words you predict the more chance you are to get gibberish
- Training with more corpus might help with that

UNIT 11: Poetry

- We are going to use a larger corpus

UNIT 12: Link to Laurence Poetry

UNIT 13: Looking into the code

- You can experiment with the dimensionality of the Embeddings, RNN units, Optimizer learning rate, different epochs

UNIT 14: Laurence the poet!

UNIT 15: Check out the code

- Ran and Downloaded “Course_3_Week_4_Lesson_2_Notebook.ipynb”

UNIT 16: Your next task

- Character based text generation

UNIT 17: Link to generating text using a character based RNN

- Checkout: https://www.tensorflow.org/tutorials/text/text_generation

UNIT 18: Quiz

UNIT 19: Exercise 4

- Check out: https://keras.io/api/layers/regularizers/

UNIT 20: Wrap Up

- We have learnt how to do NLP in TF, there is still more to it but this course covers the basics and equips us for the next level

UNIT 21: A conversation with Andrew Ng

- Sequence models are useful for time series applications like stock market, time series electric recordings of the heart
- Build on all we have learned