SUMMARY OF COURSERA COURSE

SEQUENCE MODELS

RATINGS: 4/5

WEEK 1 – RECURRENT NEURAL NETWORKS

UNIT 1: Why sequence models

- RNNs have transformed speech recognition and NLP
- Examples of sequence data: Speech Recognition, Music Generation, Sentiment Classification, DNA sequence analysis, Machine Translation, Video Activity recognition, Name Entity Recognition they are Supervised Learning problems

UNIT 2: Notation

- Sequence models have a wide range of applications
- Name Entity Recognition is used by Search Engines
- Dictionary sizes of 30k/ 50k are not uncommon
- Use One Hot representation to represent the words in our dictionary (vocabulary)

UNIT 3: Recurrent Neural Network Model

- We cannot use standard NN because: 1) inputs and output can be different lengths in different examples, 2) doesn’t share features learned across different positions of text
- The RNN passes on its activations from the previous time step for the next time step to use
- RNN scans through the data from left to right and the parameters for each time step are shared
- A set back of this RNN is that it only uses the information that is earlier in the sequence to make prediction

UNIT 4: Backpropagation through time

- Often programming frameworks take care of the Back Propagation
- Back Prop works in the opposite direction of Forward Prop
- “BackProp through time” cools name

UNIT 5: Different types of RNNs

- We can modify the basic RNN to solve specific problem sets
- 1) “Many to Many Architecture (Same length)”: both input and output have multiple values (Named Entity Recognition)
- 2) “Many to One Architecture”: Multiple inputs but a single output (Sentiment Analysis)
- 3) “One to Many Architecture”: Single input but multiple outputs (Music Generation)

- 4) “Many to Many (different lengths) Architecture”: Machine Translation, involves an “encoding” followed by a “decoding”

UNIT 6: Language model and sequence generation

- Language modelling is one of the most basic and important tasks in natural language processing, RNNs do very well on it
- Language models tell us a probability of a sentence, it is a foundational element for speech recognition and machine translation
- Building a Language Model: 1) Training set: large corpus of English text, each step in the RNN will look at a set of preceding words, 2) Define a Cost Function

UNIT 7: Sampling Novel Sequences

- Sampling novel sequence is one of the ways we can informally get a sense of what is learned by our trained sequence model
- Whatever word is chosen in a preceding time step is used as input in the next time step
- “<EOS>”: end of sentence token, “<UNK>”: Unknown word token
- Character-level language model: for this the Vocabulary will be alphabets (a,b,c,…), with this you don’t have to worry about <UNK>, but you end up with longer sequences
- Character-level are more computationally expensive to train, used in specialized situations

UNIT 8: Vanishing gradients with RNNs

- One of the problems of a basic RNN is that it runs into vanishing gradient
- Language can have very long term dependences (a word in position 2 can affect a word in position 7)
- Vanishing gradients is one of the main problems of RNN and harder to solve, exploding gradients are less common they can be solved by “gradient clipping”

UNIT 9: Gated Recurrent Unit (GRU)

- GRU is a modification to the RNN hidden layer that makes it much better capturing long range connections and helps a lot with the vanishing gradient problems
- “Cho et al, 2014”, “Chung et al, 2014”
- GRU has a memory cell (C)
- “Gate Value”
- GRU and LSTM are the most commonly used

UNIT 10: Gated Recurrent Unit (GRU) Correction

UNIT 11: Long Short Term Memory (LSTM)

- LSTM is more powerful than the GRU
- “Hochreiter & Schmidhuber 1997”
- It uses different update and forget gate, it has 3 gates

- LSTM came much earlier than GRU, there is no general consensus on which to use
- The GRU is a much simpler model, most people will pick LSTM as a first choice

UNIT 12: Long Short Term Memory (LSTM) Correction

UNIT 13: Bidirectional RNN

- Allow us to take information from both earlier and later in the sequence
- "RNN/ GRU/ LSTM blocks"
- "Acyclic Graph"
- The Forward Prop will include both a left to right activation and a right to left activation
- A BRNN with a LSTM blocks is very commonly used in NLP
- For BRNN you need the whole sequence to make prediction (for instance in speech recognition, the people needs to finish talking before you can get an output), for real time application more complicated models are employed

UNIT 14: Deep RNNs

- To learn very complex functions sometimes it is useful to stack multiple layers of RNNs together to build even deeper versions of these models
- For RNNs having 3 hidden layers is already deep enough cause of the temporal dimensions

UNIT 15: Quiz

UNIT 16: Programming Assignment – Building a RNN step by step

UNIT 17: Programming Assignment – Dinosaur Island: Character-Level Language Modelling

WEEK 2 – NATURAL LANGUAGE PROCESSING AND WORD EMBEDDINGS

UNIT 1: Word Representation

- How RNN, GRU, LSTM can be applied to NLP
- Wording Embedding
- Instead of a One Hot Representation we can use a "Featurized Representation" it helps the model to generalize better
- "t-SNE" to visualize Featured Representation (word embedding) by mapping (non linearly) it into 2D space

UNIT 2: Using Word Embeddings

- Algorithm used for Word Embedding can examine lot of text corpus and learn from it
- It allows for transfer learning
- Transfer Learning and Word Embedding: 1) learn word embedding from large text corpus (1-100B words) or download pre-trained embedding online, 2) Transfer

embedding to new task with smaller training set, 3) Optional continue to fine tune the word embedding with new data

- Word embedding is very useful for NLP task with small datasets
- The words “embedding” and “encoding” are used interchangeably

UNIT 3: Properties of word enbeddings

- They can help with “analogy reasoning” which might not be a major NLP application but gives a sense of how Word Embedding works
- “Mikolov et al, 2013”
- “Cosine Similarity” is commonly used for analogy reasoning

UNIT 4: Embedding Matrix

- Our goal to implement Embedding matrix E, and used gradient descent to learn the parameters of E
- E (all the embedding) * oj (one-hot vector of word j) = ej (embedding of word j)
- In practice we use specialized function to look up an embedding instead of doing a matrix multiplication

UNIT 5: Learning Word Embedding

- Simpler algorithms nowadays do very well compare to highly complex model that were initially used in the early days
- “Bengio et al, 2003”
- Feed the embedding as input into a NN
- Fixed History helps to specify the number of words to use as input
- Context (input) > Target word (output)
- Sometime the context has to be both from the left and the right if the target output is located in the middle
- Skip Gram Model

UNIT 6: Word2Vec

- “Mikolov et al, 2013”
- Word2Vec Skip-grams model: takes in one word as input and outputs one target word some random distance from the context
- Softmax classification in the skip-gram model has computational issues, using “hierarchical softmax” helps with this

UNIT 7: Negative Sampling

- “Mikolov et al, 2013”
- 1) Pick a context and a target word and give a label of “1”, 2) Then pair the context with “k” random words from the dictionary and give a label of “0” (negative label) to generate training set
- Use k = 5-20 for small dataset and 2-5 for large dataset

- Instead of a 10000 way softmax, we make use of 10000 binary classifications, only need to update k+1 binary classifier on each iterations
- Selecting negative examples frequency raised to the power of ¾
- You can use pre-trained Word2Vec models

UNIT 8: GloVe Word Vectors

- “Pennington et al, 2014”
- “GloVe”: global vectors for word representation
- It is difficult to humanly understand the rows of a word embedding matrix

UNIT 9: GloVe Word Vectors Correction

UNIT 10: Sentiment Classification

- Sentiment Classification is the task of looking at a piece of text and telling if someone likes of dislikes the thing you are talking about
- One of the problem is the lack of large training set, with word embedding this problem can be fixed
- It ignores word order but using RNN helps to fix this
- RNN for sentiment classification: 1) use word embedding for word j as input into a RNN unit 2) This is a many to one architecture

UNIT 11: Debiasing Word Embedding

- ML and AI algorithms are increasing trusted to help with or to make extremely important decisions
- Word embedding can reflect biases of the text used to train the model which can be negative
- “Bolukbasi et al, 2016”
- Addressing Bias in Word Embedding: 1) Identity Bias Direction 2) Neutralize: for every word that is not definitional, project to get rid of bias 3) Equalize pairs
- Most words in English are non-definitional

UNIT 12: Quiz

UNIT 13: Programming Assignment – Operations on word vectors

UNIT 14: Programming Assignment – Emojify

WEEK 3 – SEQUENCE MODELS AND ATTENTION MECHANISM

UNIT 1: Basic Models

- Last week of the specialization
- Sequence to sequence models useful for machine translation and speech recognition
- “Sutskever et al, 2014”, “Cho et at, 2014”

- Encoder > Decoder
- Machine Translation really works well nowadays
- Image Captioning: Encode Image with ConvNets > then feed it to an RNN to decode it
- “Karpathy and Fei Fei, 2015”, “Mao et al, 2014”, “Vinyals et al, 2014”

UNIT 2: Picking the most likely sentence

- There are similarities between the “sequence to sequence” models and the “language” models
- The Decoding Network part of STS models looks like the “language model”
- Machine Translation: “Conditional Language Model”
- Probability of English Sentence given an input French Sentence, find the English sentence that maximizes this Probability
- Possible combination of English words is exponentially large
- We will use a search algorithm to pick the most likely sentence

UNIT 3: Beam Search

- Beam Search algorithm: most widely used search algorithm for picking the most likely sentence
- At each step we instantiate copies of the network corresponding to the “beam width”
- If “beam width” is set to 1, it is then turned to “greedy search algorithm”

UNIT 4: Refinements to Beam Search

- “Length Normalization”: normalizing by the length of words in our translation
- The larger “beam width” the more computational expensive it gets, but give better results and vice versa
- In production “beam width” of 10 is good, but in research field we can see values over 1000
- Beam Search is not guaranteed to find exact maximum value of argmax P(y | x)

UNIT 5: Error Analysis in Beam Search

- Error analysis help us to focus our time on doing the most useful work for your project
- Beam Search is an approximate search algorithm also called “heuristic search algorithm”
- Determine if the error is mainly from the BS or from the RNN model
- Your models as two aspects: RNN (encoder and decoder) and BS
- Determine what fraction of errors are due to BS vs RNN model

UNIT 6: Bleu Score (optional)

- If there are multiple good translations how do you evaluate a machine translatin? We use “Bleu Score”
- “Bleu”: Bilingual Evaluation Understudy

- Bleu Score is an understudy used as an alternative to evaluate machine translation systems
- We use a “modified precision”: count clip (no of times in any of our references) / count (no of times in our translation)
- “Papineni et al, 2002”
- Bleu Score gives a single row number evaluation metric, it is used for Machine Translation, Image Captioning

UNIT 7: Bleu Score Correction

UNIT 8: Attention Model Intuition

- Encoder – Decoder Architecture does well for “short sentence” but the performance takes a dive at “long sentences”, “Attention Models” help with this
- “Bahdanau et al, 2014”
- “Attention Weights”: how much attention should be paid to certain parts of the input when generating certain part of the output

UNIT 9: Attention Model

- “Xu et al, 2015”
- It has quadratic time complexity (cost)
- It has been applied to image captioning
- One of the most powerful ideas in Deep Learning

UNIT 10: Corrections

UNIT 11: Speech Recognition

- Audio Clip as Input and Transcript as Output
- An audio clip plots air pressure versus time
- 3000 hours of audio input is not too bad for academia
- Use of 100,000 hours of audio input used for industry scale products
- “CTC (Connectionist temporal classification) cost for speech recognition”: Graves et al, 2006
- Basic rule for CTC is “collapse repeated characters not separated by blanks”
- You can also use Attention Models

UNIT 12: Trigger Word Detection

- Trigger Word system: Amazon Echo, Baidu DuerOS, Apple Siri (Hey Siri), Google Home
- It is still a developing field
- Input Audio Clip and Output 0/1 if trigger word is mentioned in the Audio Clip

UNIT 13: Conclusion and Thank you

- Deep Learning is a “Super Power”
- Use the ideas to solve challenges

UNIT 14: Workera’s Standardized Tests for AI Skills

- Try out Workera after Tensorflow in practice specialization

UNIT 15: Quiz

UNIT 16: Programming Assignment – Neural Machine Translation with Attention

UNIT 17: Programming Assignment – Trigger Word Detection