SUMMARY OF COURSERA COURSE

INTRODUCTION TO TENSORFLOW FOR AI, ML, AND DL

RATINGS: 4/5

WEEK 1 – A NEW PROGRAMMING PARADIGM

UNIT 1: Before you begin: Tensor Flow 2.0 and this course

- The course teaches both TF 2.0 and 1.x

UNIT 2: Introduction – A conversation with Andrew Ng

- One of the best tools for DL is TF
- This course teaches us how to use TF effectively
- The Deep Learning industry is exploding
- New tools tend to open up new possibilities

UNIT 3: A primer in Machine Learning

- Coding has been the bread and butter for developers
- Inputs (Rules and Data) → Traditional Programming → Output (Answers)
- Inputs (Answers and Data) → Machine Learning → Output (Rules)
- Machine Learning help to figure out the rules, by figuring the pattern in data
- NN is a slightly more advanced implementation of Machine Learning

UNIT 4: The “Hello World” of Neural Networks

- Machine Learning is all about a computer learning the patterns that distinguish things
- Using Python, TF and Keras
- “model = keras.Sequential()” > “model.compile()” > “model.fit()” > “model.predict()”
- “Dense()”: to define a layer of connected neurons
- You define the input shape in the first layer of the NN
- Most of math and calculus of DL are abstracted away by functions in TF and Keras
- The NN starts like a guess work which gets better through back prop
- When using NN they handle predictions in probabilities

UNIT 5: From rules to data

UNIT 6: Working through “Hello World” in TF and Python

- Details behind the concept and paradigms of ML
- You can use “Google Colab” or stick to my Jupyter Lab
- The “loss” usually gets lower with subsequent “epochs”

UNIT 7: Try it for yourself

- Download the notebook or try it on Colab
- Downloaded “Course_1_Part_2_Lesson_2_Notebook.ipynb”
- Created “Course_1_Week_1_1.ipynb”

UNIT 8: Quiz

UNIT 9: Intro to Google Colab

- FAQ: https://research.google.com/colaboratory/faq.html

UNIT 10: Get started with Google Colab

- Colab is an executable document
- It is like a Jupyter Notebook stored in Google drive
- Connects notebook to cloud based runtime
- Share Colab notebooks
- Google “SEEDBANK”

UNIT 11: Programming Assignment - Exercise 1 (Housing Prices)

UNIT 12: Week 1 Resources

- TF playground: http://playground.tensorflow.org/

WEEK 2 – INTRODUCTION TO COMPUTER VISION

UNIT 1: A Conversation with Andrew Ng

- Fitting X and Y is the basis of Deep Learning
- Computers are getting good in CV

UNIT 2: An Introduction to CV

- In Machine Learning a computer infers the rules between data and label
- The smaller the image size the better, but no too small that we cannot recognize the details

UNIT 3: Exploring how to use data

- Machine Learning depends on having good data to train a system with
- Fashion-mnist dataset: https://github.com/zalandoresearch/fashion-mnist

UNIT 4: Write code to load training data

- Going deeper in TF and Keras
- Fashion Mnist has a Keras API
- “keras.dataset.fashion_mnist”, “.load_data()”

UNIT 5: The structure of Fashion MNIST data

- Using number has a label is the first step in avoiding bias

UNIT 6: Coding a Computer Vision Neural Network

- "flatten()": turns image to a flat array

UNIT 7: See how it's done

UNIT 8: Walk through a Notebook for Computer Vision

UNIT 9: Get Hands-on with Computer Vision

- "spend some quality time playing around with the notebook, experiment with the code"
- Attempt some of the exercises provided in the notebook
- Downloaded "Course_1_Part_4_Lesson_2_Notebook.ipynb"
- Created "Course_1_Week_2_1.ipynb"

UNIT 10: Using Callbacks to control training

- The training loop gives allowance for callbacks
- Create Callback class with methods > Instantiate Class > Add it the "model.fit"

UNIT 11: See how to implement Callbacks

UNIT 12: Walk through a notebook with Callbacks

- Callback seems pretty cool

UNIT 13: Quiz

UNIT 14: Programming Assignment – Exercise 2 (Handwriting Recognition)

UNIT 15: Week 2 Resources

WEEK 3 – ENHANCING VISION WITH CONVOLUTIONAL NEURAL NETWORKS

UNIT 1: A conversation with Andrew Ng

- "DNN": Dense Neural Networks
- CNNs are very good for Computer Vision task
- Convolution seems complicated but they are quite simple

UNIT 2: What are convolutions and pooling

- A way to condense the image to the important features is where "convolution" comes in
- Convolutions change images to make certain features of the image come out
- Pooling is a way of compressing an Image

UNIT 3: Coding Convolutions and Pooling Layers

- “Conv2D” and “MaxPooling2D” layers in TF

UNIT 4: Implementing Convolutional Layers

- We define Convolution and Pooling layers in TF to do the job for us
- The input Conv2D layer has to specify the input shape of the image

UNIT 5: Learn More about Convolutions

- Check out CNN course in Deep Learning Specialization

UNIT 6: Implementing Pooling Layers

- MaxPooling2D(), it is going to take the maximum layer
- “model.summary()”: is a useful method, it shows the journey of the data (shape) through the layers
- The output shapes are determined by the parameter values of the 2D Layers

UNIT 7: Getting Hands-On, your first ConvNet

- CNNs make the network train against the results of the convolutions instead of the raw pixels

UNIT 8: Improving the Fashion Classifier with Convolutions

- It takes longer to train CNNs
- Loop through the activation to see the journey of the image through the convolution, play with the values

UNIT 9: Try it yourself

- Play with it for about 1 hour, attempt the exercise below the notebook
- Downloaded “Course_1_Part_6_Lesson_2_Notebook.ipynb”
- Created “Course_1_Week_3_1.ipynb”
- Reshape the data to a 4D ist to allow the Convolution process it using “data.reshape(10000, 28, 28, 1)”
- For number of Convolutions it is good to start with something in the order of 32

UNIT 10: Walking through convolutions

- Convolution are simply filters
- Pooling help to reduce the dimensions but retains the image

UNIT 11: Experiment with filters and pools

- https://lodev.org/cgtutor/filtering.html
- Downloaded “Course 1 - Part 6 - Lesson 3 - Notebook.ipynb”

UNIT 12: Quiz

UNIT 13: Programming Assignment – Exercise 3 (Improve MNIST with Convolutions)

UNIT 14: Week 3 Resources

WEEK 4 – USING REAL-WORLD IMAGES

UNIT 1: A Conversation with Andrew Ng

- With real world images we have to go deeper cause they are not usually grey scale but colored, they have larger number of pixels and the objects in the image are not always centered
- CNN used in Self Driving Car

UNIT 2: Explore an impactful, real-world solution

- "Disease detection with the Cassava Plant", deployed a detection algorithm on a phone, detects in a matter of seconds

UNIT 3: Understanding ImageGenerator

- What happens when we use images with different sizes, aspect ratio
- You might have to split the data yourself in some situations
- Image > Training , Validation > Horses (training, validation), Humans (training, validation)
- "from tensorflow.keras.preprocessing.image import ImageDataGenerator"
- Always point the image generator at the directory containing the sub directory (not the subdirectory)
- ".flow_from_directory(…)"
- For training NN the images have to be the same size

UNIT 4: Designing the neural network

UNIT 5: Defining a ConvNet to use complex images

- "sigmoid" is very good for binary classification

UNIT 6: Train the ConvNet with ImageGenerator

- ".fit_generator( )"

UNIT 7: Training the ConvNet with fit_generator

- Loss = "binary_crossentropy"
- Use ".fit_generator" when we are using an ImageGenerator
- "steps_per_epoch" = total images / batch size

UNIT 8: Exploring the solution

UNIT 9: Walking through developing a ConvNet

UNIT 10: Training the neural network

- “RMSProp”

UNIT 11: Walking through training the ConvNet with fit_generator

- Try to train on a “GPU”
- “PIXABAY”

UNIT 12: Experiment with the horse or human classifier

- Downloaded “Course_1_Part_8_Lesson_2_Notebook.ipynb”
- Created “Course_1_Week_4_1.ipynb”

UNIT 13: Adding automatic validation to test accuracy

- Build in validation into the training loop
- Download the validation
- Create a validation generator
- There are many white horses in the training images

UNIT 14: Get hands-on and use validation

UNIT 15: Exploring the impact of compressing images

- Let us try to reduce the image size to “150 x 150”
- It is important to measure your model against a large validation set, inspect where it got it wrong, tells us where to focus on to improve our model

UNIT 16: Quiz

UNIT 17: Programming Assignment – Exercise 4

UNIT 18: Week 4 Resources

UNIT 19: Wrap up

- We will learn how to go in depth with real world images, learn many of the techniques used in challenges such as those run on Kaggle

UNIT 20: A conversation with Andrew

- With frameworks you can write complicated codes with few lines of code