SUMMARY OF COURSERA COURSE

CONVOLUTIONAL NEURAL NETWORK IN TENSORFLOW

RATINGS: 4/5

WEEK 1 – EXPLORING A LARGER DATASET

UNIT 1: Introduction, A conversation with Andrew Ng

- Go much further, apply the techniques to larger dataset
- With larger dataset you are in less prone to overfitting
- “Transfer Learning”, able to build off other peoples work it shows the open source nature of the software industry

UNIT 2: Before you Begin: TensorFlow 2.0 and this course

- It teaches both TF 1.x and 2.0

UNIT 3: A conversation with Andrew Ng

- We will learn to deal with messy data

UNIT 4: The cats vs dogs dataset

- Image data can be messy, having corrupt files
- Kaggle cat vs dog competition dataset

UNIT 5: Training with the cats vs. dogs dataset

- ImageGenerator helps auto label images based on their directories
- Similar workflow used for horses vs humans

UNIT 6: Looking at the notebook

- Downloaded “Course_2_Part_2_Lesson_2_Notebook.ipynb”
- Created “Course_2_Week_1_1.ipynb”

UNIT 7: Working through the notebook

- Using a reduced dataset of just 3000 images of cat vs dog
- “training generator”, “validation generator”
- Have some fun with the notebook

UNIT 8: What you will see next

- Technique for fixing errors

UNIT 9: Fixing through cropping

- By cropping the model did better for a particular image
- But will cropping be beneficial to the whole dataset????

UNIT 10: Visualizing the effect of the convolutions

- "visualization model": helps us to see the path of an image through the image

UNIT 11: Looking at accuracy and loss

- Assigning the ".fit()" outputs to a variable in order to plot the train/validation loss and accuracy
- "history = model.fit(….)"

UNIT 12: What have we seen so far?

- Next week learn about other techniques for reducing overfitting

UNIT 13: Quiz

UNIT 14: Week 1 Wrap Up

- There is now a significance improvement in the Deep Learning Field due to better computing power and powerful frameworks

UNIT 15: Programming Assignment – Exercise 1 – Cats vs Dogs

WEEK 2 – AUGMENTATION: A TECHNIQUE TO AVOID OVERFITTING

UNIT 1: A conversation with Andrew Ng

- Image augmentation and data augmentation are one the most widely used tools in Deep Learning to increase the dataset size and make your NN perform better
- Image Generator helps to do this on the fly

UNIT 2: Image Augmentation

- It helps to avoid overfitting, the less data you have the less chance you have of getting accurate predictions

UNIT 3: Introducing Augmentation

- Augmentation make our smaller dataset more effective

UNIT 4: Start Coding

- "keras.preprocessing"

UNIT 5: Coding Augmentation with ImageDataGenerator

- ImageDataGenerator as an image augmentation capability
- “rotation_range”, “width_shift_range”, “height_shift_range”, “shear_range”, “zoom_range”, “horizontal_flip”, “fill_mode”
- Checkout keras documentation for more options

UNIT 6: Looking at the notebook

- Downloaded “Course_2_Part_4_Lesson_2_Notebook_(Cats_v_Dogs_Augmentation).ipynb”
- Created “Course_2_Week_2_1.ipynb”

UNIT 7: Demonstrating overfitting in cats vs dogs

UNIT 8: The impact of augmentation on Cats vs Dogs

- Adding augmentation to help combat overfitting

UNIT 9: Adding augmentation to cats vs. dogs

- Looking at the epoch versus accuracy graph we can decide if more epochs will give us a boost

UNIT 10: Try it yourself

- Let us go back to the horses vs human dataset from course one and see if augmentation will help

UNIT 11: Exploring augmentation with horses vs humans

- Augmentation adds randomness to the training set but if the validation set does not have the same randomness it cannot not do much
- So we need a broad dataset for both training and validation

UNIT 12: What have we seen so far

- Looked in Image Augmentation, with it you can simulate a larger dataset from a smaller one
- It helps to combat overfitting

UNIT 13: Quiz

UNIT 14: Week 2 Wrap Up

- Play around with image augmentation
- Next week we will try out transfer learning

UNIT 15: Programming Assignment – Exercise 2 – Cats vs Dogs using augmentation

WEEK 3 – TRANSFER LEARNING

UNIT 1: A Conversation with Andrew Ng

- Transfer learning one the most important techniques of deep learning, with TF it takes just a handful lines of codes
- The open source nature of AI is a world changer
- Transfer Learning is easy to implement in TF
- Transfer Learning is like "standing on the shoulders of Giants"

UNIT 2: Understanding transfer learning – the concepts

- Take an existing model and make use of the features that model has learned and train your own Dense Layer
- "Inception" NN Architecture

UNIT 3: Start Coding!

- Documentation on how to lock and freeze layers: https://www.tensorflow.org/tutorials/images/transfer_learning
- Downloaded: "Course_2_Part_6_Lesson_3_Notebook.ipynb"
- Created "Course_2_Week_3_1.ipynb"

UNIT 4: Coding Transfer Learning from the Inception Mode

- "layer.trainable = False": to freeze or lock layer from being trainable

UNIT 5: Adding your DNN

- Adding our own DNN at the bottom of the pre trained model which can be retrained to our data

UNIT 6: Coding your own model with transferred features

- You can select the "last layer" you want to use from the "pre trained layer" using the ".get_layer(layer_name)" method and take its output "last_layer.output"

UNIT 7: Using Dropouts

- Using Dropouts helps to reducing overfitting by removing a random number of neurons in our neural network

UNIT 8: Exploring Dropouts

UNIT 9: Applying Transfer Learning to Cats v Dogs

UNIT 10: Exploring Transfer Learning with Inception

- Classification of multi objects in the next week

UNIT 11: What have we seen so far

UNIT 12: Quiz

UNIT 13: Week 3 Wrap up

- Moving from binary to multiclass classification in the next week

UNIT 14: Programming Assignment – Horse vs Humans using Transfer Learning

WEEK 4 – MULTICLASS CLASSIFICATIONS

UNIT 1: A conversation with Andrew Ng

- Computer graphics to generated/synthesize datasets for CNNs
- Computer graphics datasets are becoming more widely used

UNIT 2: Moving from binary to multiclass classification

UNIT 3: Introducing the Rock-Paper-Scissors dataset

- Rock Paper Scissor was generated as an experiment in determining if a CGI-based dataset can be used for classification

UNIT 4: Explore multiclass with Rock Paper Scissor dataset

- “class_mode = categorical”
- “loss = categorical_crossentropy”

UNIT 5: Check out the code

- Downloaded “Course_2_Part_8_Lesson_2_Notebook_(RockPaperScissors).ipynb”
- Created “Course_2_Week_4_1.ipynb”

UNIT 6: Train a classifier with Rock Paper Scissors

UNIT 7: Try testing the classifier

- Downloaded a test set to test the model “rps-validation.zip”

UNIT 8: Test the Rock Paper Scissor classifier

- The label from the Image Generator are done “alphabetically”

UNIT 9: What have we seen so far

UNIT 10: Quiz

UNIT 11: Programming Assignment – Exercise 4 Multiclass Classifier

UNIT 12: Wrap Up

UNIT 13: A conversation with Andrew Ng

- NLP is really taking up