SUMMARY OF KAGGLE COURSE

DEEP LEARNING

RATINGS: 4/5

LESSON 1 – INTRO TO DL FOR COMPUTER VISION

UNIT 1: Intro

- We will understand convolutions
- Convolutions are the basic building blocks for deep learning models in computer vision

UNIT 2: Lesson

- CV is the building block for self-driving cars
- Tensorflow is the most tool for Deep Learning
- Keras is a popular API for specifying Deep Learning
- We will start by running model first then coming back to theoretical aspect
- Tensor: something like a matrix
- Convolutions are also called “filters”, the numerical values of the convolution determine the pattern it recognizes

UNIT 3: Your Turn

- Downloaded Notebook

LESSON 2 – BUILDING MODELS FROM CONVOLUTIONS

UNIT 1: Intro

- How convolutions are combined to enable super human achievements in computer vision

UNIT 2: Lesson

- Apply filter across the image and map it to a tensor
- We can have many filter to capture different patterns on the image
- Each time we apply a new layer of convolution we discover more interesting patterns
- Object Detection
- Data used in the Image Net competition are available on Kaggle
- You can start with using pre trained models before you can train models from scratch by yourself
- Transfer learning is mind blowing cool, it starts with pre trained models

LESSON 3 – TENSORFLOW PROGRAMMING

UNIT 1: Intro

- Able to write Tensorflow and Keras code to use one of the best models in Computer Vision

UNIT 2: Lesson

- Preprocessing of image to forms that we can run into our models
- The more you work with the code the more familiar it will become

UNIT 3: Your Turn

- Downloaded Notebook

LESSON 4 – TRANSFER LEARNING

UNIT 1: Intro

- Able to use transfer learning to build highly accurate CV models for custom purposes even when you have relatively little data

UNIT 2: Lesson

- Transfer learning gives good result with little data, most CV problems have same low level pattern recognitions
- Add a new layer containing our use cases, using our data to train the last layer
- Last layer is usually a “Dense Layer”
- From Keras we import “Sequential” and “Dense”
- Data Augmentation help to improve our NN when working with small or medium size data

UNIT 3: Your Turn

- Snapshot and Downloaded Notebook

LESSON 5 – DATA AUGMENTATION

UNIT 1: Intro

- Able to use data augmentation. This trick makes it seem like you have far more data that you actually have, resulting in even better model

UNIT 2: Lesson

- “horizontal_flip = True”, it is not good for all uses cases
- Validation score are more reliable with larger validation sets

UNIT 3: Your Turn

- Downloaded Notebook

LESSON 6 – A DEEPER UNDERSTANDING OF DEEP LEARNING

UNIT 1: Intro

- Understand how stochastic gradient descent and back-propagation are used to set the weights in a deep learning model. These topics are complex, but many experts view them as the most import ideas in deep learning

UNIT 2: Lesson

- When working with tabular data it is common to have many “Dense Layers” between the input and the output
- Layers between the input and output layers are called “Hidden Layers”
- Good weights are the key to good predictions
- We use Loss Function where low values are better
- We will use Gradient Descent to select weights that minimize our loss function
- Batch size is the number of rows used for each step
- One time through the whole training set is called an “EPOCH”
- Backpropagation is the way we use to find out how to change the weights, it involves the use of chain rule

LESSON 7 – DEEP LEARNING FROM SCRATCH

UNIT 1: Intro

- Pre-trained models are not the ideal solution for many use cases. In this lesson we will learn how to build totally new models

UNIT 2: Lesson

- Transfer Learning usually works when our use cases has similar visual patterns used in our pre-trained model
- To be versatile in Deep Learning we need to be able to build from scratch
- For the first layer we have specify the “input_shape”
- Improving Model Capacity by adding more hidden layers can cause the model to perform better or cause it to over fit

UNIT 3: Your Turn

- Downloaded Notebook

LESSON 8 – DROPOUT AND STRIDES FOR LARGER MODELS

UNIT 1: Info

- Stride lengths to make model faster and reduce memory consumption
- Dropout to combat overfitting
- Both of these techniques are especially useful in large models

UNIT 2: Lesson

- Drop out the most popular technique to combat overfitting
- “strides = **”
- Strides are cleaner than other alternatives like “Max Pooling”
- “model.add(Dropout(**))”
- Before Dropout we combat overfitting by reducing model capacity

UNIT 3: Your Turn

- Downloaded Notebook