Introduction to Tensorlow 2.0

Week 1:

1. In keras keras.layers.Dense – define a layer of connected neurons in it
2. Model.compile(optimizer=’sgd’, loss=’mean_squared_error’) - To define optimizer and loss function
3. Optimizer – Generates a new and improved guess
4. Loss – shows how good the current guess is
5. Convergence – the process of getting very close to the answer

Week 2:

6. Computer vision – is a field of having a computer understand and label what is present in and image
7. Tf.keras.layers.Flatten(input_shape) - Flattens the input.
8. Activation – attribute of Dense layer(ex. Activation = tf.nn.relu/tf,nn.softmax)

Week 3:

9. Convolution – a technique to isolate feature from image - tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(28, 28, 1))
10. Pooling – a technique to reduce information in an image while maintaining features tf.keras.layers.MaxPooling2D(2, 2)

Week 4:

11. From tensorflow.keras.preprocessing.image import ImageDataGeneration
    1. Train_dataget = ImageDataGeneration(rescale=1./255)
    2. Train_generator = train_datagen.flow_from_directory(Train_dir, target_size=(300, 300)(autorescale), batch_size=128, class_mode=’binary’)
12. model.fit_generator(

```
      train_generator,
      steps_per_epoch=2,
      epochs=15,
      verbose=1,
      callbacks=[callbacks])
```

13. Callback

```
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
      if(logs.get('acc')>DESIRED_ACCURACY):
        print("\nReached 99.9% accuracy so cancelling training!")
        self.model.stop_training = True
  callbacks = myCallback()
```