**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:

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

Week 3:

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

Week 4:

1. 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’)

1. model.fit\_generator(

train\_generator,

steps\_per\_epoch=2,

epochs=15,

verbose=1,

callbacks=[callbacks])

1. Callback **class** **myCallback**(tf.keras.callbacks.Callback):

**def** on\_epoch\_end(self, epoch, logs={}):

**if**(logs.get('acc')>DESIRED\_ACCURACY):

print("**\n**Reached 99.9**% a**ccuracy so cancelling training!")

self.model.stop\_training = **True**

callbacks = myCallback()