

Convolutional NN in TensorFlow

Week 1

1. to view the history of my training – create a var history and assign it to the return model.fit
2. model.layers API allows to inspect the impact of convolution
3. The validation accuracy is based on images that the model has not seen before and thus better indicator of how model will perform with new images
4. Flow_from_directory – the ability to automatically label images based on their directory name

Week 2

5. Train_datagen = ImageDataGenerator(
Rescale = 1./255, rotation_range = 40, width_shift_range=0.2, height_shift_range=0.2,
Shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest')

Week 3:

6. Transfer learning
 - a. Define an architecture. Ex.
pre_trained_model = InceptionV3(input_shape = (150, 150, 3), include_top = False, weights = None)
 - b. Load weights. Ex. pre_trained_model.load_weights(local_weights_file)
 - c. Set not to retrain your model. Ex. for layer in pre_trained_model.layers:
layer.trainable = False
 - d. Get last output last_layer = pre_trained_model.get_layer('mixed7')
last_output = last_layer.output
 - e. Then define new model. Ex.

x = layers.Flatten()(last_output)

x = layers.Dense(1024, activation='relu')(x)

x = layers.Dropout(0.2)(x)

x = layers.Dense(1, activation='sigmoid')(x)
model = Model(pre_trained_model.input, x)
model.compile(optimizer = RMSprop(lr=0.0001),
loss = 'binary_crossentropy',
metrics = ['accuracy'])
 - f. Define validation generator and training generator
 - g. Then fit your new model

7.

Week 4

8.