

Week 1 Quiz

What does `flow_from_directory` give you on the ImageGenerator?

The ability to easily load images for training

The ability to pick the size of training images

The ability to automatically label images based on their directory name

All of the above

If my Image is sized 150x150, and I pass a 3x3 Convolution over it, what size is the resulting image?

153x153

150x150

148x148

450x450

If my data is sized 150x150, and I use Pooling of size 2x2, what size will the resulting image be?

75x75

149x149

148x148

300x300

If I want to view the history of my training, how can I access it?

Pass the parameter 'history=true' to the model.fit

Download the model and inspect it

Create a variable 'history' and assign it to the return of model.fit or model.fit_generator

Use a model.fit_generator

What's the name of the API that allows you to inspect the impact of convolutions on the images?

The model.convolutions API

The model.pools API

The model.layers API

The model.images API

When exploring the graphs, the loss levelled out at about .75 after 2 epochs, but the accuracy climbed close to 1.0 after 15 epochs. What's the significance of this?

There was no point training after 2 epochs, as we overfit to the validation data

There was no point training after 2 epochs, as we overfit to the training data

A bigger training set would give us better validation accuracy

A bigger validation set would give us better training accuracy

Why is the validation accuracy a better indicator of model performance than training accuracy?

It isn't, they're equally valuable

There's no relationship between them

The validation accuracy is based on images that the model hasn't been trained with, and thus a better indicator of how the model will perform with new images.

The validation dataset is smaller, and thus less accurate at measuring accuracy, so its performance isn't as important

Why is overfitting more likely to occur on smaller datasets?

Because in a smaller dataset, your validation data is more likely to look like your training data

Because there isn't enough data to activate all the convolutions or neurons

Because with less data, the training will take place more quickly, and some features may be missed

Because there's less likelihood of all possible features being encountered in the training process.

Week 2 Quiz

How do you use Image Augmentation in TensorFlow

With the keras.augment API

You have to write a plugin to extend tf.layers

With the tf.augment API

Using parameters to the ImageDataGenerator

If my training data only has people facing left, but I want to classify people facing right, how would I avoid overfitting?

Use the 'horizontal_flip' parameter

Use the 'flip' parameter and set 'horizontal'

Use the 'flip' parameter

Use the 'flip_vertical' parameter around the Y axis

When training with augmentation, you noticed that the training is a little slower. Why?

Because there is more data to train on

Because the training is making more mistakes

Because the image processing takes cycles

Because the augmented data is bigger

What does the fill_mode parameter do?

There is no fill_mode parameter

It creates random noise in the image

It attempts to recreate lost information after a transformation like a shear

It masks the background of an image

When using Image Augmentation with the ImageDataGenerator, what happens to your raw image data on-disk.

It gets overwritten, so be sure to make a backup

A copy is made and the augmentation is done on the copy

Nothing, all augmentation is done in-memory

It gets deleted

How does Image Augmentation help solve overfitting?

It slows down the training process

It manipulates the training set to generate more scenarios for features in the images

It manipulates the validation set to generate more scenarios for features in the images

It automatically fits features to images by finding them through image processing techniques

When using Image Augmentation my training gets...

Slower

Faster

Stays the Same

Much Faster

Using Image Augmentation effectively simulates having a larger data set for training.

False

True

Week 3 Quiz

If I put a dropout parameter of 0.2, how many nodes will I lose?

20% of them

2% of them

20% of the untrained ones

2% of the untrained ones

Why is transfer learning useful?

Because I can use all of the data from the original training set

Because I can use all of the data from the original validation set

Because I can use the features that were learned from large datasets that I may not have access to

Because I can use the validation metadata from large datasets that I may not have access to

How did you lock or freeze a layer from retraining?

`tf.freeze(layer)`

`tf.layer.frozen = true`

`tf.layer.locked = true`

`layer.trainable = false`

How do you change the number of classes the model can classify when using transfer learning? (i.e. the original model handled 1000 classes, but yours handles just 2)

Ignore all the classes above yours (i.e. Numbers 2 onwards if I'm just classing 2)

Use all classes but set their weights to 0

When you add your DNN at the bottom of the network, you specify your output layer with the number of classes you want

Use dropouts to eliminate the unwanted classes

Can you use Image Augmentation with Transfer Learning Models?

No, because you are using pre-set features

Yes, because you are adding new layers at the bottom of the network, and you can use image augmentation when training these

Why do dropouts help avoid overfitting?

Because neighbor neurons can have similar weights, and thus can skew the final training

Having less neurons speeds up training

What would the symptom of a Dropout rate being set too high?

The network would lose specialization to the effect that it would be inefficient or ineffective at learning, driving accuracy down

Training time would increase due to the extra calculations being required for higher dropout

Which is the correct line of code for adding Dropout of 20% of neurons using TensorFlow

tf.keras.layers.Dropout(20)

tf.keras.layers.DropoutNeurons(20),

tf.keras.layers.Dropout(0.2),

tf.keras.layers.DropoutNeurons(0.2)

Week 4 Quiz

The diagram for traditional programming had Rules and Data In, but what came out?

Answers

Binary

Machine Learning

Bugs

Why does the DNN for Fashion MNIST have 10 output neurons?

To make it train 10x faster

To make it classify 10x faster

Purely Arbitrary

The dataset has 10 classes

What is a Convolution?

A technique to make images smaller

A technique to make images larger

A technique to extract features from an image

A technique to remove unwanted images

Applying Convolutions on top of a DNN will have what impact on training?

It will be slower

It will be faster

There will be no impact

It depends on many factors. It might make your training faster or slower, and a poorly designed Convolutional layer may even be less efficient than a plain DNN!

What method on an ImageGenerator is used to normalize the image?

normalize

flatten

resize()

rescale

When using Image Augmentation with the ImageDataGenerator, what happens to your raw image data on-disk.

A copy will be made, and the copies are augmented

A copy will be made, and the originals will be augmented

Nothing

The images will be edited on disk, so be sure to have a backup

Can you use Image augmentation with Transfer Learning?

No - because the layers are frozen so they can't be augmented

Yes. It's pre-trained layers that are frozen. So you can augment your images as you train the bottom layers of the DNN with them

When training for multiple classes what is the Class Mode for Image Augmentation?

class_mode='multiple'

class_mode='non_binary'

class_mode='categorical'

class_mode='all'