**1.**

Question 1

What does flow\_from\_directory give you on the ImageDataGenerator?

1 point

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

**2.**

Question 2

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

1 point

150x150

148x148

153x153

450x450

**3.**

Question 3

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

1 point

300x300

148x148

75x75

149x149

**4.**

Question 4

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

1 point

Create a variable ‘history’ and assign it to the return of model.fit or model.fit\_generator

Pass the parameter ‘history=true’ to the model.fit

Use a model.fit\_generator

Download the model and inspect it

**5.**

Question 5

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

1 point

The model.images API

The model.layers API

The model.convolutions API

The model.pools API

**6.**

Question 6

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?

1 point

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

**7.**

Question 7

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

1 point

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

**8.**

Question 8

Why is overfitting more likely to occur on smaller datasets?

1 point

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.

**1.**

Question 1

**How do you use Image Augmentation in TensorFLow**

1 point

**With the tf.augment API**

**Using parameters to the ImageDataGenerator**

**With the keras.augment API**

**You have to write a plugin to extend tf.layers**

**2.**

Question 2

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

1 point

**Use the ‘flip\_vertical’ parameter around the Y axis**

**Use the ‘flip’ parameter**

**Use the ‘flip’ parameter and set ‘horizontal’**

**Use the ‘horizontal\_flip’ parameter**

**3.**

Question 3

**After adding data augmentation and using the same batch size and steps per epoch, you noticed that each training epoch became a little slower than when you trained without it. Why?**

1 point

**Because there is more data to train on**

**Because the training is making more mistakes**

**Because the image preprocessing takes cycles**

**Because the augmented data is bigger**

**4.**

Question 4

**What does the fill\_mode parameter do?**

1 point

**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**

**5.**

Question 5

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

1 point

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

**6.**

Question 6

How does Image Augmentation help solve overfitting?

1 point

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

**7.**

Question 7

When using Image Augmentation my training gets...

1 point

Slower

Faster

Stays the Same

Much Faster

**8.**

Question 8

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

1 point

False

True

**1.**

Question 1

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

1 point

20% of them

2% of them

20% of the untrained ones

2% of the untrained ones

**2.**

Question 2

Why is transfer learning useful?

1 point

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

**3.**

Question 3

How did you lock or freeze a layer from retraining?

1 point

tf.freeze(layer)

tf.layer.frozen = true

tf.layer.locked = true

layer.trainable = false

**4.**

Question 4

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)

1 point

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

**5.**

Question 5

Can you use Image Augmentation with Transfer Learning Models?

1 point

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

**6.**

Question 6

Why do dropouts help avoid overfitting?

1 point

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

Having less neurons speeds up training

**7.**

Question 7

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

1 point

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

**8.**

Question 8

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

1 point

tf.keras.layers.Dropout(20)

tf.keras.layers.DropoutNeurons(20),

tf.keras.layers.Dropout(0.2),

tf.keras.layers.DropoutNeurons(0.2),

**.**

Question 1

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

1 point

Answers

Binary

Machine Learning

Bugs

**2.**

Question 2

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

1 point

To make it train 10x faster

To make it classify 10x faster

Purely Arbitrary

The dataset has 10 classes

**3.**

Question 3

What is a Convolution?

1 point

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

**4.**

Question 4

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

1 point

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!

**5.**

Question 5

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

1 point

normalize

flatten

rezize()

rescale

**6.**

Question 6

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

1 point

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

**7.**

Question 7

Can you use Image augmentation with Transfer Learning?

1 point

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

**8.**

Question 8

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

1 point

class\_mode='multiple'

class\_mode='non\_binary'

class\_mode='categorical'

class\_mode='all'