1.Question 1

Gray-scale images consist of 3 channels while colored images consist of 1 channel?

**1 / 1 point**



True



**False**

**Correct**

Great job! Gray-scaled images consists of 1 channel while colored images consists of 3 channels indicating Red, Green and Blue

2.Question 2

**Which of the following indicates that the machine learning model is significantly underfitting?**

**1 / 1 point**



When training accuracy is 90% and validation accuracy is 85%



**When training accuracy is 70% and validation accuracy is 55%**



When training accuracy is 98% and validation accuracy is 80%



When training accuracy is 85% and validation accuracy is 75%

**Correct**

Correct! Underfitting occurs when the model is too simple that it cannot reflect the complexity of the training dataset. Whenever both the training and validation accuracy are low, it usually mean that the model is underfitting.

3.Question 3

**While training your CNN model, you find that training accuracy is 96% and the validation accuracy is 80%. What could be the reason for this?**

**1 / 1 point**



Model generalized well because of underfitting



Model generalized well because of overfitting



Model generalized poorly because of underfitting



**Model generalized poorly because of overfitting**

**Correct**

Great job! Since the training accuracy is very high and the validation score is low, it means that the model has overfitted the training data and it did not generalize well.

4.Question 4

How many images will this code print out?



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

W\_grid = 10

L\_grid = 5

fig, axes = plt.subplots(L\_grid, W\_grid, figsize = (5,5))

axes = axes.ravel()

n\_training = len(X\_train) # get the length of the training dataset

# Select a random number from 0 to n\_training

for i in np.arange(0, L\_grid \* W\_grid): # create evenly spaces variables

# Select a random number

index = np.random.randint(0, n\_training)

# read and display an image with the selected index

axes[i].imshow( X\_train[index])

axes[i].set\_title(y\_train[index], fontsize = 15)

axes[i].axis('off')

**0 / 1 point**



25



50



15



10

**Incorrect**

Incorrect, please review "*Task #3: image visualization*"

5.Question 5

Which does this code do?



1

2

3

4

5

from sklearn.utils import shuffle

X\_train, y\_train = shuffle(X\_train, y\_train)

X\_train = np.sum(X\_train/3, axis = 3, keepdims = True)

X\_train = (X\_train - 128)/128

**1 / 1 point**



Data shuffling, normalization, scaling



Data shuffling, color-scale conversion, normalization



**Data shuffling, gray-scale conversion, normalization**

**Correct**

Great job! the code does the following steps in order: (1) Data shuffling, (2) gray-scale conversion, (3) normalization

6.Question 6

**Artificial Neural Networks (ANNs) are information processing models that work by trying to mimic human biological neurons. ANNs can be modeled as follows "output = Ax + b". Which of the following variables are adjustable parameters? (where A is the input, x is the weight and b is the bias)**

**1 / 1 point**



x and output



A and b



output and A



output and b



**x and b**

**Correct**

Great Job! Weight and bias are the parameters that can be optimized while training and they play a major role in the performance of the model.

7.Question 7

**You are working on a computer vision application and you need that application to detect faces. You have been tasked to use CNN for the model. Why are CNN preferred for task related to images?**

**1 / 1 point**



CNNs are able to update their weights much better compared to other networks.



CNNs are able to remember the relationship between various input images



**CNNs are able to extract high level features**



CNNs has a built-in generalization capability

**Correct**

Excellent job! the main advantage of CNNs is that they are able to automatically extract high level features from images.

8.Question 8

**You trained an artificial neural network to perform multi-class classification. After model training and validation, you find that your model is overfitting the training data. What changes can you make to the model architecture to avoid overfitting?**

**0 / 1 point**



**Use early stopping**



Use more dataset



Add an additional dense layer



Add dropout

**Incorrect**

Incorrect. Please review "*Task 8: Assess Trained Model Performance*" for more information.

9.Question 9

For binary classification problems, we use the following command to compile the model:



1

2

CNN.compile(optimizer = 'Adam', loss = 'sparse\_categortical\_crossentropy',

  metrics = ['accuracy'])

**1 / 1 point**



True



**False**

**Correct**

Great job! We use loss = binary\_categorical\_crossentropy for binary classification problems.

10.Question 10

**You are training a CNN image classifier and you have initially observed around 40th epoch that the model's validation score and training accuracy were both 94%. As the training continued, at 90th epoch, the training accuracy went up to 98% and validation score went down to 87%. What is the problem here and how can it be rectified effectively?**

**1 / 1 point**



CNN is underfiiting and we should use early stopping



CNN is underfitting and we should shuffle the data



**CNN is overfitting and we should use early stopping**



CNN is overfitting and we should shuffle the data