PROJECT

NEURAL NETWORKS

Asmaa mahdi 2111900
Rimas ALshehri 2110240

Dr. Samia Snoussi

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 1:

- **What is the output of the model.summary()?**

- The `model.summary()` function provides a summary of the model architecture. It includes information such as the model's name and the total number of parameters in the model.
- The summary consists of a table-like structure that displays details for each layer in the model.
- Each layer is listed with its type, output shape, and the number of parameters it has.
- The output shape represents the dimensions of the layer's output tensor.
- The parameter count indicates the total number of learnable parameters in the layer that will be updated during training.
- The summary also shows the total number of trainable parameters in the model.
- Additionally, it indicates whether there are any non-trainable parameters in the model.
- The `model.summary()` output is valuable for understanding the model's architecture, complexity, and parameter distribution.

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 2:

- What the initial training accuracy and validation accuracy of the CNN?

```
initial_train_accuracy = H.history['accuracy'][0]
initial_val_accuracy = H.history['val_accuracy'][0]

print("Initial Training Accuracy:", initial_train_accuracy)
print("Initial Validation Accuracy:", initial_val_accuracy)
```

IInitial Training Accuracy: 0.0234375
Initial Validation Accuracy: 0.0

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 3:

- **How many convolutional layers and pooling layers does this network have?**

The network described in the code has a total of 3 convolutional layers and 3 pooling layers.

Convolutional Layers:

1. The first convolutional layer (`Conv2D`) has 8 filters and a kernel size of (5, 5)
2. The second convolutional layer (`Conv2D`) has 16 filters and a kernel size of (5, 5)
3. The third convolutional layer (`Conv2D`) has 32 filters and a kernel size of (3, 3)

Pooling Layers:

1. The first pooling layer(`MaxPooling2D`) reduces the size of the feature maps by a factor of 4.
2. The second pooling layer(`MaxPooling2D`) reduces the size of the feature maps by a factor of 4.
3. The third pooling layer(`MaxPooling2D`) reduces the size of the feature maps by a factor of 2.

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 4:

Generally, the larger the size of the image the more the information in it. The maxpooling layers after first and second Convolutional layer decrease the size of the image by 4. Check if this is causing the network to have such a poor validation accuracy? If the size of pooling layers size is changed from (4,4) to (2,2) what is the effect on accuracy of the network?

```
#Add a conv layer having 8 filters followed by a relu layer (image size 64 x 64)
model.add(Conv2D(8, (5, 5), activation='relu', input_shape=(64, 64, 1), padding='same'))
#Reduce the size of the images by 2 using maxpooling layer (image size 16 x 16)
model.add(MaxPooling2D(pool_size=(2,2)))
#Add a conv layer having 16 filters followed by a relu layer (image size 64 x 64)
model.add(Conv2D(16, (5, 5), activation='relu', padding='same'))
#Reduce the size of the images by 2 using maxpooling layer (image size 4 x 4)
model.add(MaxPooling2D(pool_size=(2,2)))
#Add a conv layer having 16 filters followed by a relu layer (image size 64 x 64)
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
#Reduce the size of the images by 2 using maxpooling layer (image size 2 x 2)
model.add(MaxPooling2D(pool_size=(2, 2)))
```

- The pool size of the max pooling layers is (4, 4), which decreases the picture size by a factor of 4 in both dimensions. This downsampling may result in the loss of fine-grained information, which may impair the network's capacity to categorize pictures effectively, particularly in the validation set.

- The downsampling value reduces as the pooling layer size(2,2) decreases, allowing the network to retain more spatial information. This may aid in the preservation of finer features and increase validation accuracy.also,Making the image more understandable to the machine and aiding the detection of key characteristics.

- When the pool size is 4, the validation accuracy is 0.00%.
- When the poll max is set at 2, the validation accuracy is 60%

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 5:

Dr. Hinton, has highlighted that aggressively using pooling layers may result in loss of important information. Is there a way that the CNN architecture starts producing better training and validation accuracy?

To enhance training and validation accuracy without heavily relying on pooling layers,we would consider the following approaches:

- Reduce the stride of convolutional layers.
- Optimal for smaller pooling sizes, such as (2,2) instead of (4,4).
- Augment the number of filters in convolutional layers to capture a greater variety of features.
- Incorporate dropout layers to mitigate overfitting.
- Explore diverse activation functions, such as ReLU or Leaky ReLU.

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 6:

Make changes to the convolutional neural network to get the best validation accuracy. You are not allowed to change the number of epochs or batch size for this task.

In the code, we made the following modifications to optimize the CNN:

- Adjust Pooling Layer Sizes: Alter the pool sizes from (4, 4) to (2, 2) to retain more image details.
- Integrate Dropout Layers: Incorporate dropout layers to mitigate overfitting, particularly following dense or convolutional layers.
- Raise Filter Count: Enhance the number of filters in the convolutional layers to capture more intricate features.
- Modify Activation Functions: Explore different activation functions, such as using ReLU for the convolutional layers.
- Change Convolutional Layer Parameters: Experiment with diverse kernel sizes and potentially modify strides.

PROJECT

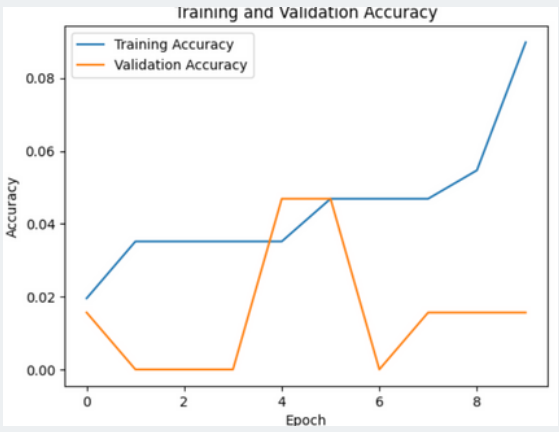
NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

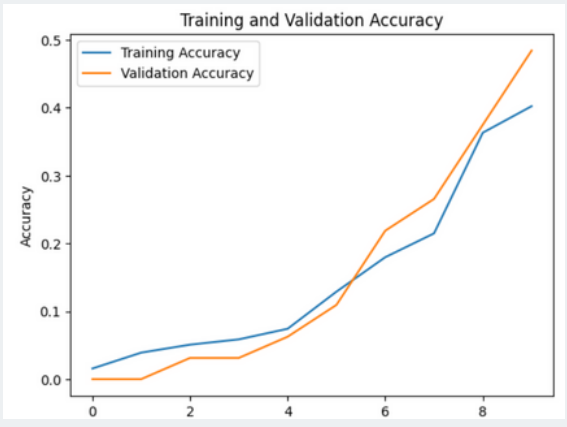
QUESTION 7:

Plot the difference between training and validation accuracy for each epoch.

```
# Plot the difference between training and validation accuracy for each epoch
plt.plot(H.history['accuracy'], label='Training Accuracy')
plt.plot(H.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```



BEFORE ENHANCMENT



AFTER ENHANCMENT

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

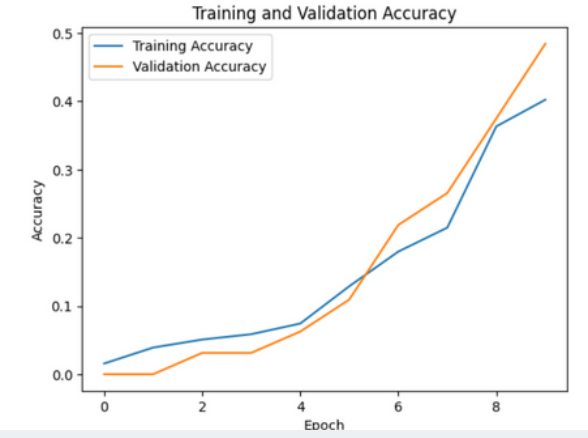
QUESTION 8:

For the best network architecture change the batch size to 16 and plot the training vs validation accuracy graph. What happened to the validation accuracy after last epoch as compared to when the batch size was 32.

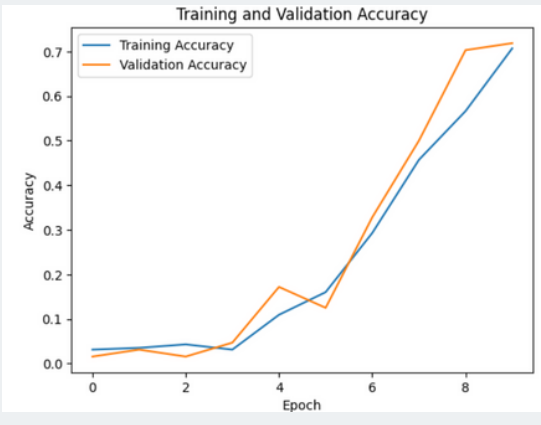
```
#Train the network using the above deinfed network architecture and give accuracy results for training and validation data
H = model.fit(X_train, Y_train, validation_data=(X_val, Y_val), batch_size=16, epochs=10, verbose=1)
```

By reducing the batch size, the model has the potential to converge more quickly and exhibit enhanced generalization capabilities. Consequently, the validation accuracy is likely to improve compared to when the batch size was set to 32. With a smaller batch size, the model can update its weights more frequently during training, increasing the likelihood of finding an improved optimal solution within the parameter space.

finally, Validation Accuracy in last epoch: 71% when batch size is 16 . However,Validation Accuracy in last epoch: 48% when batch size, is 32



BATCH SIZE 32



BATCH SIZE 16

PROJECT

NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

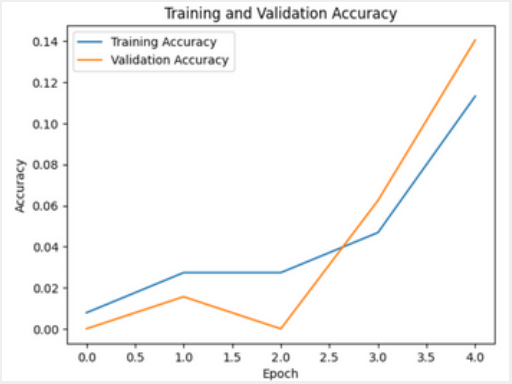
QUESTION 9:

For the best network architecture change the number of epochs to 5 and 20 and share the final validation accuracy for 5, 10 and 20 epochs. What do the results highlight?

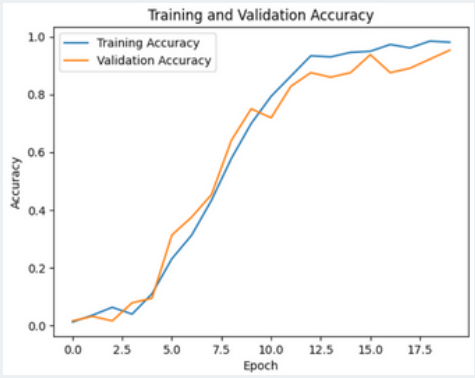
Increasing the number of epochs allows the model to go through more training iterations, which results in better performance. The model has more chances to learn from the training data and alter its weights and biases as a result. As a result, validation accuracy continues to improve.in contrast to batch size, where the smaller it is, the greater the validation accuracy.

- When the epoch is 5, the validation accuracy is 14%.
- When the epoch is 10, the validation accuracy is 75%
- When the epoch is 20, the validation accuracy is 90%

EPOCH 10



EPOCH 5



EPOCH 20

PROJECT

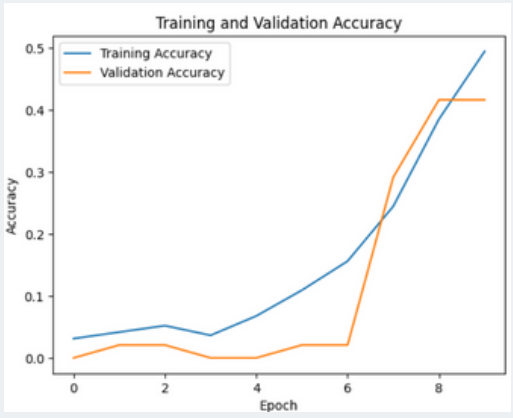
NEURAL NETWORKS

DON'T WATCH THE CLOCK; DO WHAT IT DOES. KEEP GOING

QUESTION 10:

For the best network architecture and batch size =16 and epochs =10, change the test data size to 40% and share what is the effect on validation accuracy of the algorithm?

Expanding the test dataset to 40% will shrink the training dataset, causing the model to learn from a smaller portion of the data. Consequently, this will lead to decreased accuracy in both the training and validation sets.



TEST DATA SIZE 40%