Görkem YILMAZ

21601927

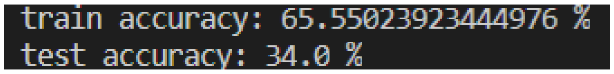
Introduction to Computer Vision

Homework 2

Report

Part 1)

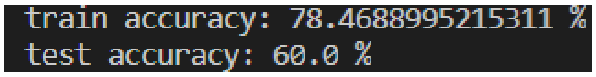
When I implemented the pseudo-algorithms and train the neuron for 50 epochs and with learning rate as 0.0001, I got the following result.



```
train accuracy: 65.55023923444976 %
test accuracy: 34.0 %
```

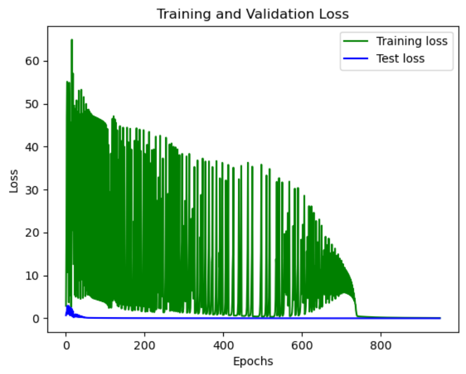
Bigger learning rates and bigger epochs are more beneficial in the matter of accuracy.

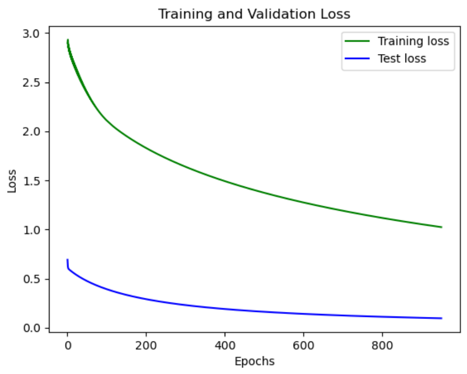
When the learning rate is 0.001 and epoch number is still 50, I get the best result.

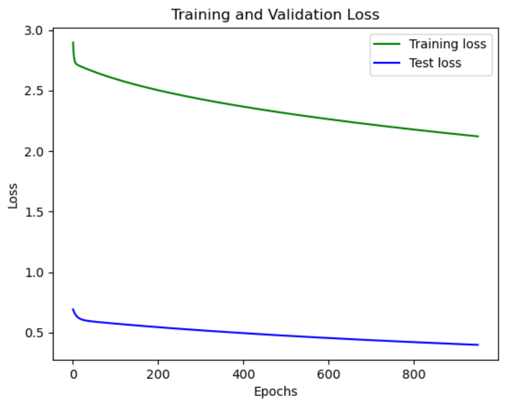


```
train accuracy: 78.4688995215311 %
test accuracy: 60.0 %
```

Part 2)







In first plot, learning rate = 0.1 and epochs = 950. In second plot learning rate = 0.001 and epochs = 950 and in the last plot, learning rate = 0.0001 and epochs = 950. As it can be seen, second plot gives the best result.

I only trained one weight and I got the best accuracies with epoch number between 800 and 1200 and also with learning rate 0.001. I decided to use these parameters by trying different parameter and comparing the results. Also, I initialized my weights as zeros.

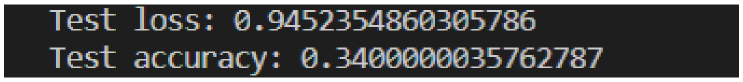
Part 3)

For the first task, I converted data to NumPy arrays and changed the input image dimensions as (64,64).

Also, since the channel does not come first in matrix, I removed the first if statement.

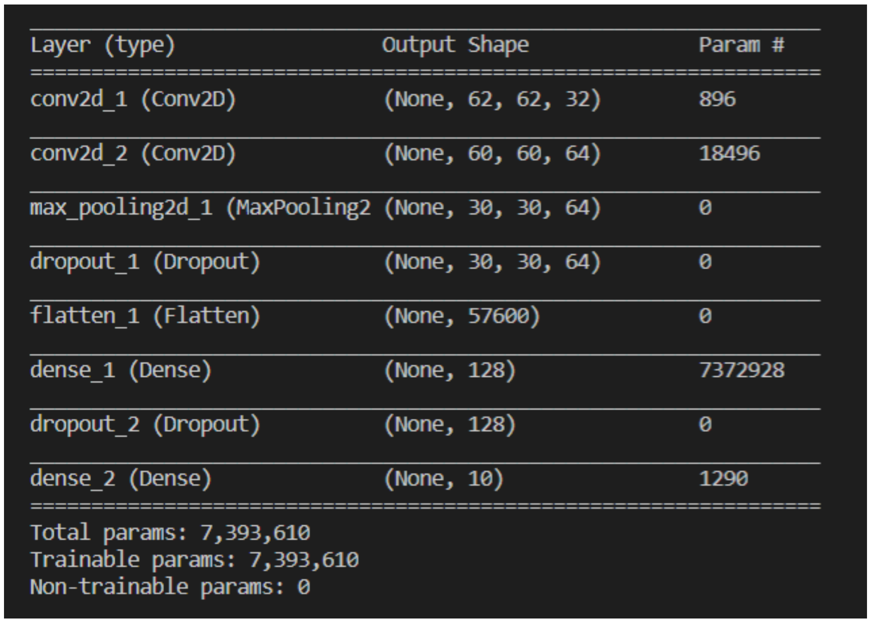
When I run the code after those changes, for 12 epochs I got the following result.

0.3400000035762787



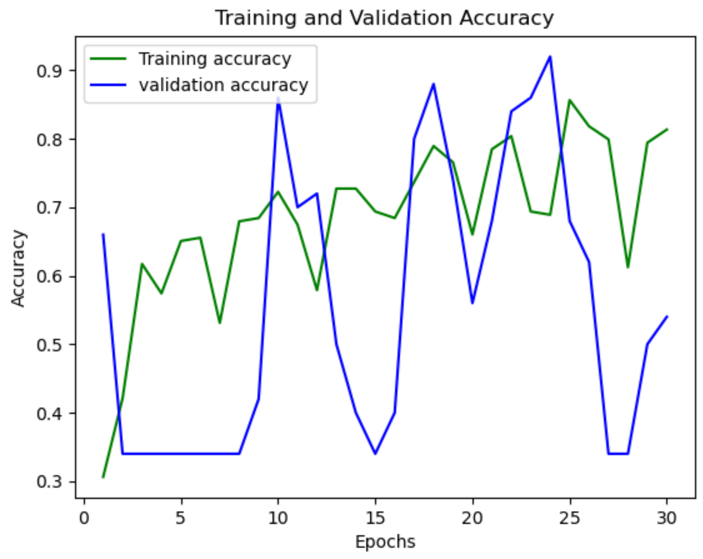
```
Test loss: 0.9452354860305786
Test accuracy: 0.3400000035762787
```

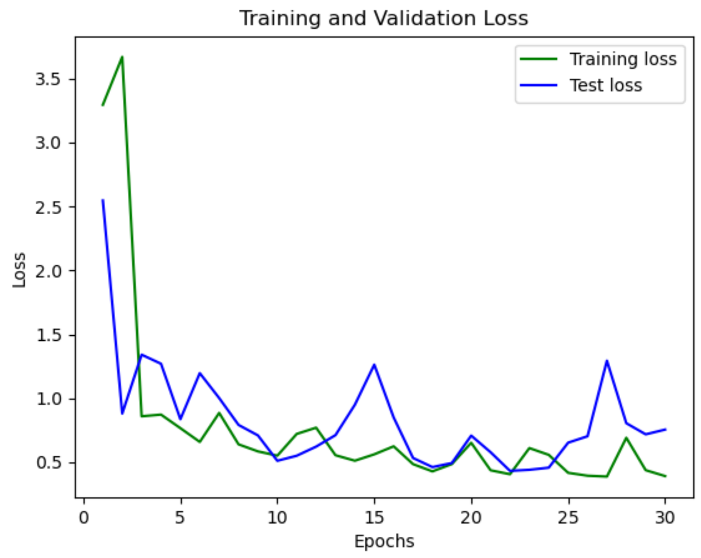
For the second task, network model for each layer is as follows:



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 60, 60, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 30, 30, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 30, 30, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 57600)             0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               7372928
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 7,393,610
Trainable params: 7,393,610
Non-trainable params: 0
```

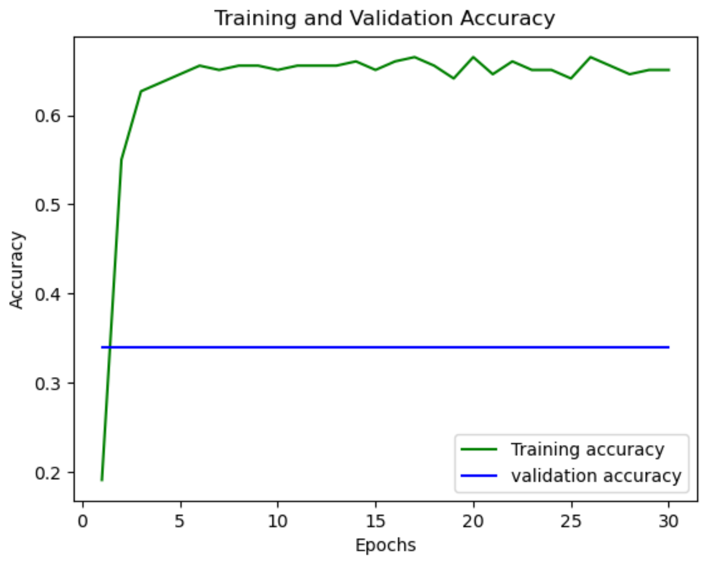
For the next task, I changed the epoch value to 30 and got the following plots.

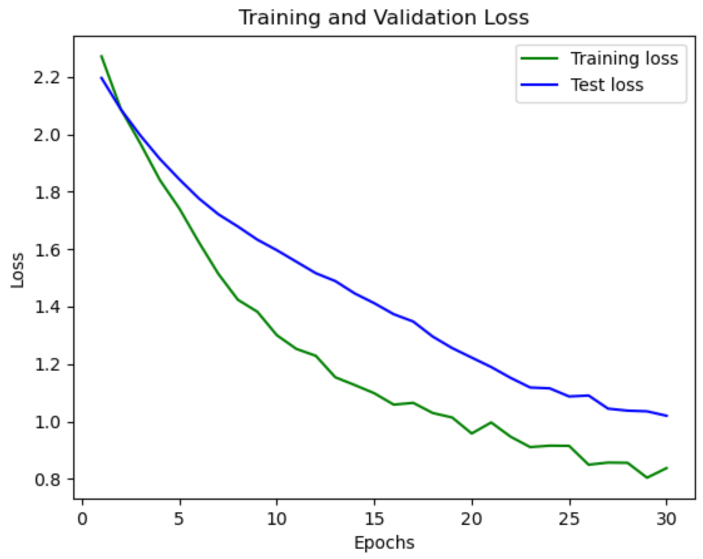




As it can be observed, it is expected to get a higher accuracy and lower loss while the number of epochs are increasing.

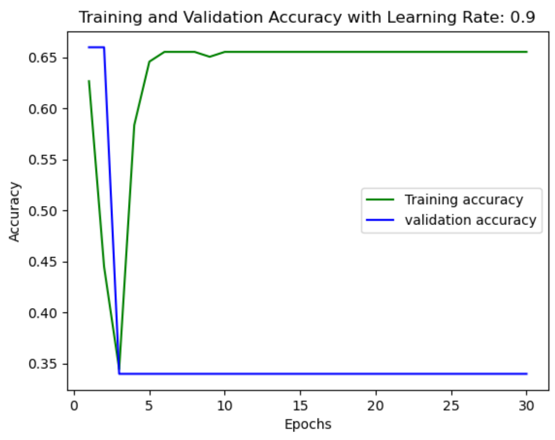
For the following task, I used the SGD cod snippet and changed the optimizer from AdaDelta to SGD optimization algorithm. I obtained the following plots after 30 epochs.

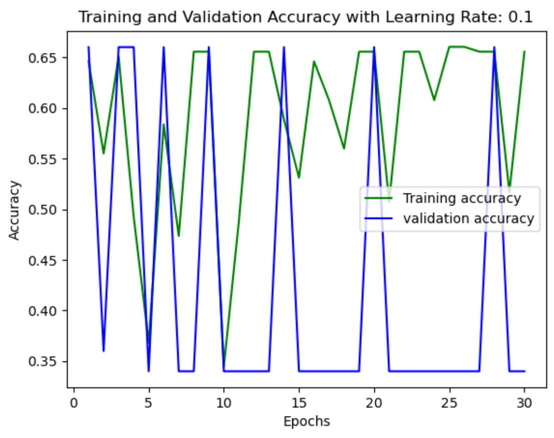


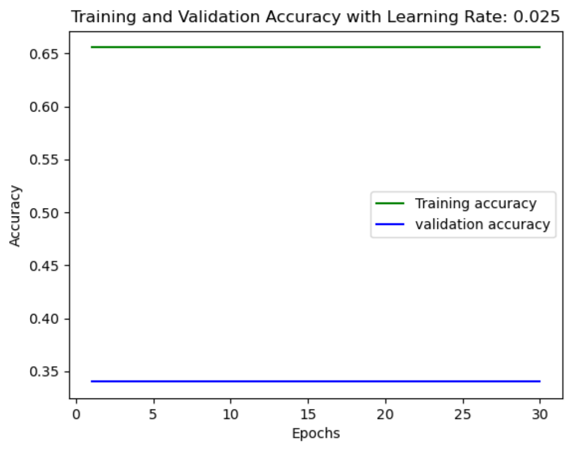


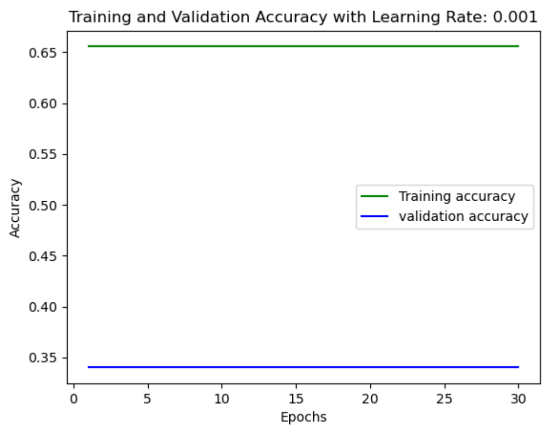
It can be seen that SGD algorithm creates more stabil loss and accuracy results. There are almost no waving throughout the epochs.

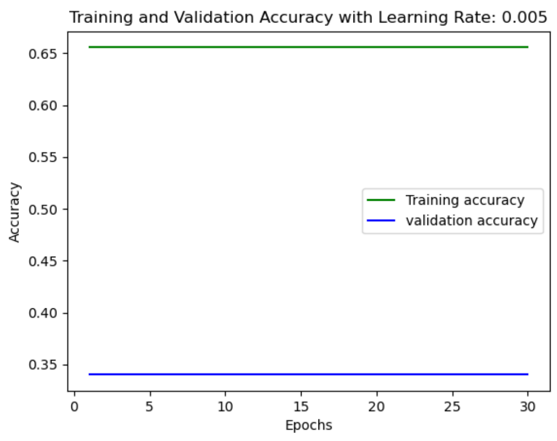
When I run the code with 5 different learning rates, I got the following results.











Higher learning rates tend to give more accuracy because they update the weight and bias more compared to lower learning rates. But I got the best result with 0.025 learning rate, not 0.1.

As for the last part, AdaDelta gives more accuracy in my case compared to SGD while the number of iterations get higher. Using a very small learning rates were not very successful because it did not have much effect on the accuracy. Using a higher learning rate for SGD gave better results. I have noticed that if learning rate is high, number of epochs can be lower. But if the learning rate is low, we need more epochs to traing the neuron better.