**EMRAH SARIBOZ**

**HW-8 Report**

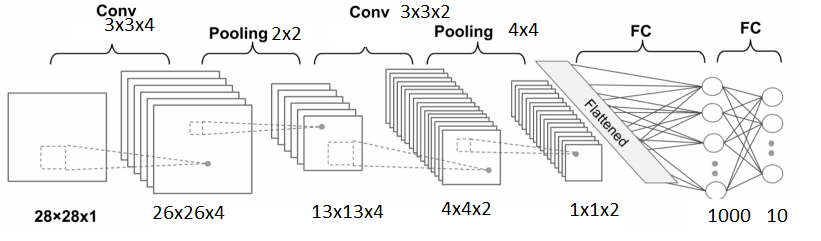
**Dataset Usage and problem definition**

* The goal of the homework is to apply CNN to recognize the hand-written dataset.
* In this homework, I used google COLAB.
* I downloaded the dataset from the link she posted and extracted it using python.

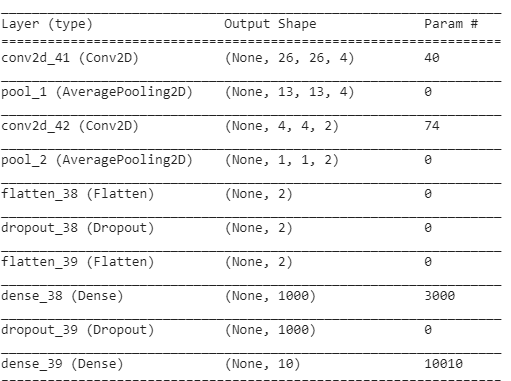
**Question:**

Show the shape of the output tensor of each layer in the CNN architecture that you design to answer Q1.

**Answer**:



Here is a model summary:



**Experiments**

By running CNN model with given credentials, I got 0.45 maximum accuracy in 100 epochs. This situation gave me a hint to change output channel size for the Conv2D processes. Here are the results from that:

1. Just by changing the output size of first and second conv2D from 4 and 3 to 64 and 32, I was able to get 0.97 accuracy in 100 epochs. In the figure below, you can see the last 12 epochs.
   * Total time: 99 seconds.

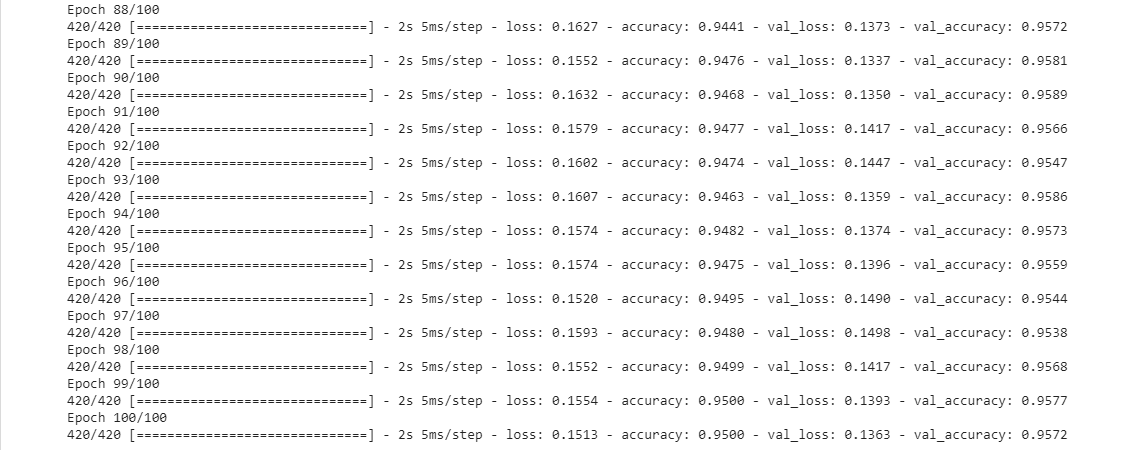
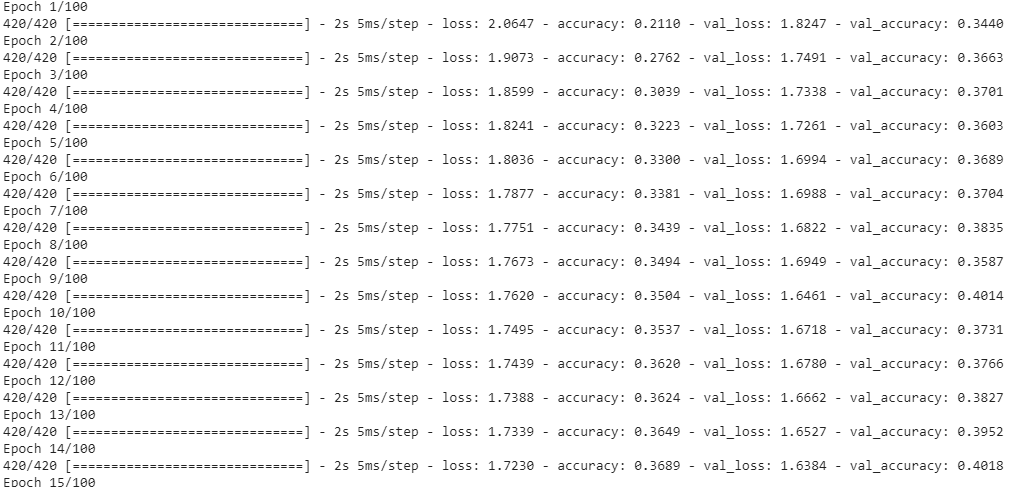
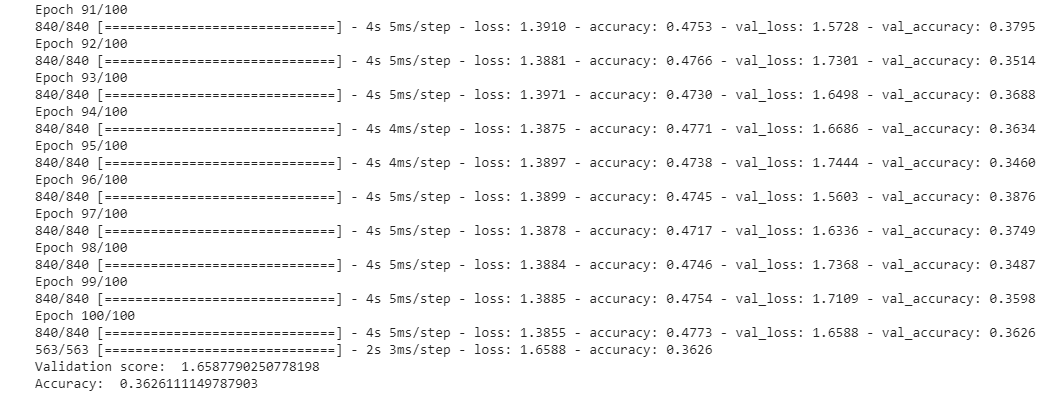


Figure 1: CNN epochs.

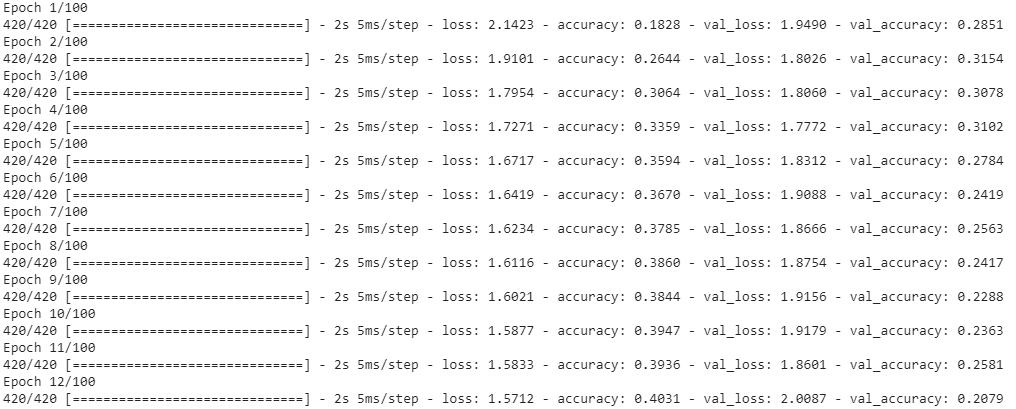
1. Instead of increasing the output channel, I increased the kernel size from 3x3 to 4x4 for both con2D operation. However, I got low accuracy. It did not make any changes.
   * Total time: 90 seconds.



1. For this experiment, I changed the batch size from 100 to 50 to see the performance. This only reduced the run time of the program. Maximum accuracy I received was 0.40.
   * Total time: 85 seconds.



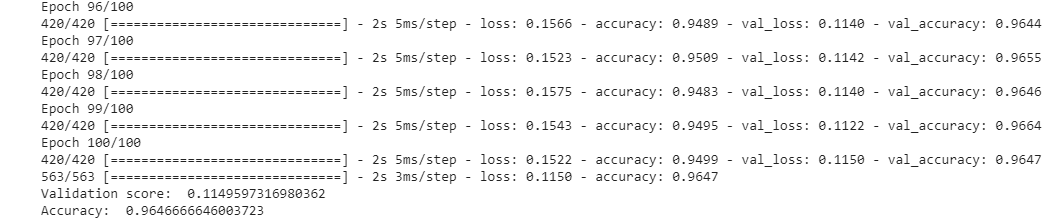
1. In third experiment, I changed the pooling function from max to average. As before, my accuracy did not go beyond 0.40 with the output sizes 4 and 2.
   * Total time: 102 seconds.



4 ) Model Evaluation

* To make sure the models generalizes well on unseen data rather than memorizing it, I used keras.model\_evaluate() function to see my loss and accuracy on testing dataset.
* This function returns two values: loss score and accuracy.

Loss-score is a difference between the predicted value and true value. The lower loss-score, the better it generalizes. For this experiment, I changed my output size from 4,2 to 64 and 32, respectively. With the 100 epochs, I got the following values.



* Here, as it can be seen from the figure above, the validation score is 0.11 and accuracy is 0.96.
* Getting this accuracy gives a hint the non-existence of the overfitting issue.