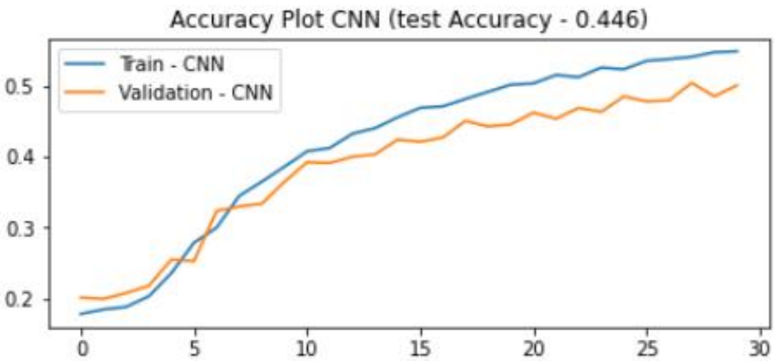
GMDL, HW 5

Liran Nochumsohn - 204693410

Problem 1:

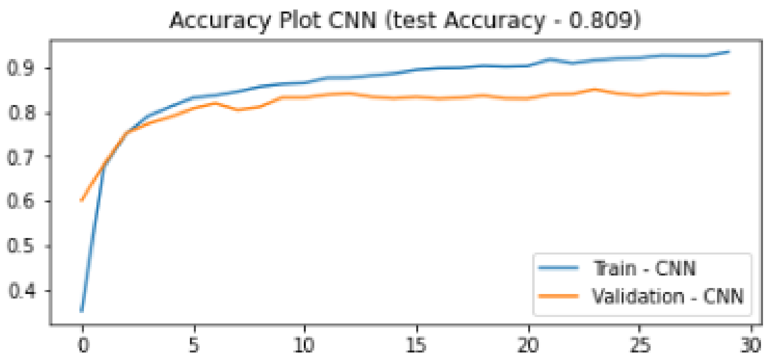
It is seen that the CNN architecture performs better on the validation set (and test set). Especially those with a kernel size greater than 1. Relative to the MLP, the mentioned CNN's exhibit less of a gap between the accuracy lines cancelling out an overfit, whereas the MLP seems to have more over fitting as well as a general lower performance on the validation set (and test set). The gap between the validation accuracy and the train accuracy for all MLP architectures are approximately 0.2 as opposed to less than 0.1 in most CNN architectures, thus confirming the overfit in MLP.

We are witnessing also an underfit with the CNN architecture with a kernel size of 1. Although the accuracy validation and train are very close to each other, both lack sufficient accuracy compared to the other architectures. Approximately 0.5 compared to 0.7-0.9 with the other architectures. As seen below (from the ipynb file):



Computer Exercise 5:

The best performing model- CNN with kernel size 10:



Problem 2:

It is **not** guaranteed to get the best performance on the test set given the best performing model on the validation set, since there might be a set of Hyper – parameters that will perform relatively better on the test set than on the validation set. This may be caused by the data examples in the test set, perhaps the data in the test set resembles better the data in the train set as opposed to the validation. We must also take into consideration that in many cases, such as ours, the validation set is smaller than the test set, so it may represent more “difficult” examples.

The basic assumption we make is that the model with the given hyper-params will perform better on the test set. That is because the validation set is has not been “seen” by the model in the training process, thus making it a measure of overfitting and worthwhileness. To avoid overfitting the model the validation set can give us an indication when to stop training a model to avoid it overfitting the trainset. The loss of the validation-loss/accuracy can give us also additional assurance alongside the train- loss/accuracy that the model will perform better on the test making it worthwhile.

Problem 3:

The chosen Transfer Learning method is fine tuning. Because although the chosen Resnet is trained on ImageNet with a rich dataset of different classes, it is not guaranteed to perform well on images with numbers. It is important to note that most of the trained classes of ImageNet come in form of living and non-living objects, and numbers of objects may be marginalized. To this end I decided to run extra training on the model to fine tune all weights with the given task of classifying numbers.

Problem 4:

The new network (Resnet – 18 with FT) performs better than the other networks with similar hyper parameters with test accuracy of 0.869. This could be explained by the magnitude of the resnet-18 architecture compared to the CNN and MLP in the previous sections. Resnet – 18 consists of 17 convolution layers and an additional average pooling layer and larger architectures have the capacity to extract more information from the data. It can also be seen with larger Resnet models (34, 50, 164 etc.) achieve better accuracy on datasets such as ImageNet. The magnitude of Resnet-18 compared to the other architectures brings advantage.

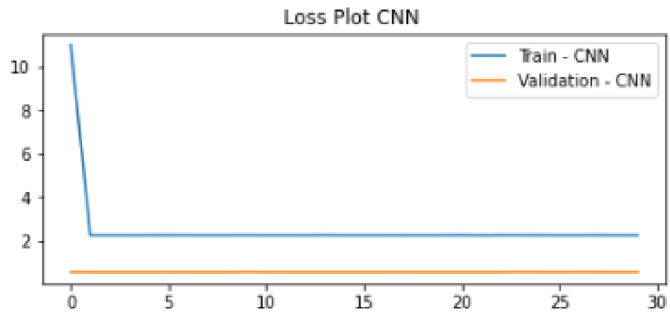
An additional factor to attribute to the model’s success is the fact the model is pretrained on ImageNet, a dataset with a large variety of classes. Given that, finetuning the weights of the model can get us a good prediction for other problems such as ours.

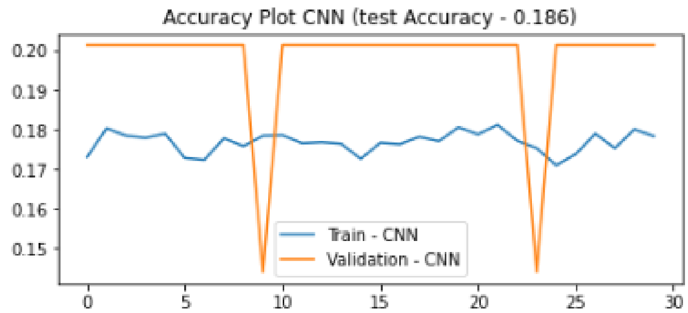
Problem 5:

We tested all architectures with learning rates 0.1 and 0.0001 and the findings are as follows:

Learning rate 0.1 – both the loss and accuracy curves in all models show underfitting. There is no proper improvement in the learning process within the duration of 30 epochs. The reason is that the optimization algorithm is continuously failing to reach lower lows since it "jumps" to far off the descent, landing in an undesirable location with respect to the lows.

Example plot from the ipynd file:





Learning rate 0.0001 – in all models with such learning rate there is no overfit excluding Resnet-18 FT. We can see that the loss plots of all models show that in fact the validation line is below the train loss line. That is because the model is still in the process of learning and has not yet reached its potential. The reason we are witnessing this phenomenon is due to the fact the trainset of a given epoch is evaluated before the backward pass, whereas the validation set is evaluated after the model has undergone the backward pass to update all the weights. Such phenomenon will slowly decay as the model learns. An example below from the ipynb file: