PHY 835: Exercise 5
Released March 11, 2021;Target Due Date: March 25, 2021

1. MNIST – Conv2D

The digits in the MNIST dataset are in the center. Here we will undo this operation and compare the performance of convolutional neural networks vs fully connected networks.

a) Create a new dataset of size 50x50 where you place the handwritten digit at different random positions in the dataset.
b) Now train a neural network with a single hidden dense layer (as on the original MNIST dataset in the lectures).
c) Now try to improve your performance in comparison to your previous layout by using an architecture involving convolutional layers (Conv2D).
d) (Optional) We have mentioned various options to improve the performance of networks. Check whether methods like Dropout, Batch Normalization, Pooling layers can improve your results. Try to fine-tune the performance (you may also try deeper architectures, i.e. with more hidden layers.)

2. Network analysis

The aim of this exercise is to familiarize yourself with analyzing tools for trained neural networks, i.e. opening the black box. In Keras you can access the value of the weights using **get_weights().** Here we analyze the network (CIFAR1.ipynb) which you can find on Canvas under “Files/Jupiter Notebooks/Ex 5/”, where you can also find a link to the pre-trained weights (keras_cifar10_trained_model.h5)

a) In a first step, identify examples where the network is not yet performing well, i.e. which are incorrectly classified.
b) In a second step, visualize the average activation of several hidden layers and in particular different hidden filters in the convolutional layers. Try to identify the role of some of the hidden filters. It might be useful to consider the activations for a particular class of the dataset.

3. Rotated MNIST

The aim of this exercise is to write your custom layer with weight sharing as discussed for the group convolutional example in the lecture.

a) Write a function which rotates images by a multiply of 90 degrees. Test your function on MNIST.
b) Generate a dataset of rotated MNIST digits using this function. Ensure that the classes in your dataset are balanced.
c) Generate two benchmarks with one hidden dense layer with 100 and 400 hidden units (ReLu activation).

d) Write a custom layer which implements the group convolutional example where 4 rotated filters are the output, i.e. the weights among the filters are shared appropriately. Verify that your layer is performing the matrix multiplication you intended to design.
e) Use this layer to generate a second benchmark using the same number of weights for the hidden layer as in the benchmark with 100 hidden units.
f) Compare the performance and training between the three benchmark models.