

Exercise Sheet 5

Submission by Monday 8th 17:00 via Moodle.

Corrections due by Thursday 11th 16:00 via Moodle.

Format for all questions: individual Jupyter notebooks .ipynb.

1 MNIST – Conv2D (4 + 2 bonus points)

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.

- (1) Create a new dataset of size 50x50 where you place the handwritten digit at different random positions in the dataset.
- (1) Now train a neural network with a single hidden dense layer (as on the original MNIST dataset in the lectures).
- (2) Now try to improve your performance in comparison to your previous layout by using an architecture involving convolutional layers (Conv2D).
- ★ We have mentioned various options to improve the performance of networks. Check whether methods like Dropout, BatchNormalization, 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 (3 points)

The aim of this exercise is to familiarise yourself with analysing 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 analyse the network we have discussed in the lectures, you can find a link to the pre-trained weights on the course website.¹

- (1) In a first step, identify examples where the network is not yet performing well, i.e. which are incorrectly classified.
- (2) In a second step, visualise 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 (7 points)

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

- Write a function which rotates images by a multiply of 90 degrees. Test your function on MNIST. (1 point)
- Generate a dataset of rotated MNIST digits using this function. Ensure that the classes in your dataset are balanced. (1 point)

¹Note that there are also pre-trained deep networks available in your Keras distribution.

- Generate two benchmarks with one hidden dense layer with 100 and 400 hidden units (ReLU activation). (1 point)
- 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. (2 point)
- 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. (1 point)
- Compare the performance and training between the three benchmark models (1 point).