Competition

Tensorboys

May 30, 2022

Pattern Recognition

Agenda

1. Multi-layer Perceptron

2. Convolutional Neural Network

Multi-layer Perceptron

MLP

- We've tested multiple architecture from $784 \times 2 \times 10$ to $784 \times 512 \times 512 \times 256 \times 128 \times 10$.
- This process was obviously automated.
- This yielded interesting results.

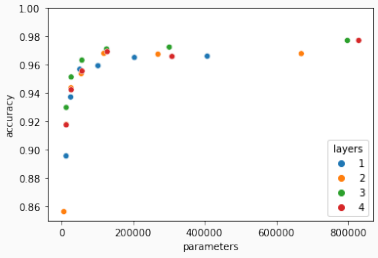

Figure 1: Relation between parameters and accuracy for different model layers. We can see a rapid performance gain in the beginning, then not much difference for many more parameters

MLP

- 20 epochs and batch size of 128.
- Some good "light" architectures with 3-4 small layers
- For the competition: $784 \times 512 \times 128 \times 64 \times 10$ trained for 100 epochs
- Precision: 97.78%

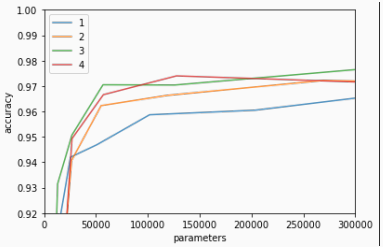

Figure 2: Zoomed out version. Useful to find the best ratio between performance and size of model

Convolutional Neural Network

CNN

- 20 epochs and batch size of 1024.
- Max Pooling after each convolution (could try without)
- one with bigger Kernel, one with a drop out layer after the flattening
- could go bigger but training time get annoying, keep things simple

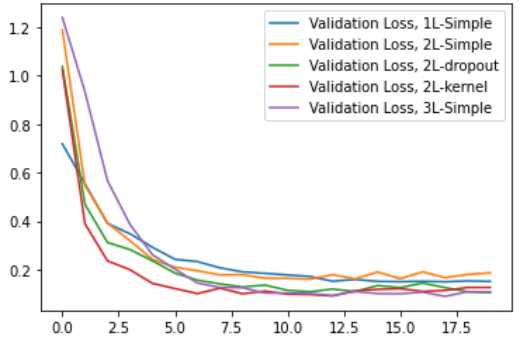

Figure 3: Performance of the different model

CNN

- 50 epochs and batch size of 512.
- Trained on the whole training/test set available from before
- Precision: 99.34%

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_10 (Conv2D)           (None, 28, 28, 32)        320
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 14, 14, 64)        18496
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 7, 7, 128)         73856
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 3, 3, 128)         0
_________________________________________________________________
flatten_5 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                11530
=================================================================
Total params: 104,202
Trainable params: 104,202
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Model used for the competition