TDT4265: Computer Vision

Assignment 3

Anders Kjelsrud

February 2023

0	0	0	0	0	0	0
0	2	1	2	3	1	0
0	4	5	0	7	0	0
0	3	9	1	1	4	0
0	0	0	0	0	0	0

$$* \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Flip the kernel and do cross-correlation:

$$\text{Result} = \begin{pmatrix} -7 & 4 & -6 & 2 & 13 \\ -20 & 10 & 2 & -2 & 18 \\ -23 & 8 & 14 & -6 & 9 \end{pmatrix}.$$

1 Task 1

b) Max pooling helps reduce translational invariance because the maximum of a region is likely to be invariant to small shifts in the input. A convolutional layer could learn patterns at different spatial locations in the image, but are not necessarily invariant. Activation functions do not have any inherent translational invariance.

c) We have that the output dimensions from a conv layer are given from

$$H_2 = \frac{H_1 - F_H + 2P_H}{S_H} + 1 \tag{1}$$

$$W_2 = \frac{W_1 - F_W + 2P_W}{S_W} + 1 \tag{2}$$

Since we want the output dimension to remain unchanged, let $H_2 = H_1 = H$. Then rearranging (1) gives

$$P_H = \frac{H(S-1) + F_W - S}{2} \tag{3}$$

The same can be done for (2) since they are symmetrical. In the case where $S = 1, F = 7$, $P = \frac{7-1}{2} = 3$. We have to use a padding of 3.

d) Using (1) we find that

$$F_H = H_1 - SH_2 + 2P_H + S = 512 - 508 + 2 \cdot 0 + 1 = 5$$

Therefore the kernel size is 5×5.

e) The formula for the output dimension of a pooling layer is [(I - F) / S] + 1 x D

$$H_2 = \frac{H_1 - F}{S} + 1 \tag{4}$$

Using (4) we find that the height of the output is

$$\frac{508 - 2}{2} + 1 = 254$$

So the dimensions are 254×254.

f) Again, using (1) we find that

$$H_2 = \frac{254 - 3 + 2 \cdot 0}{1} + 1 = 252$$

Then the output dimensions are 252×252.

g) The number of parameters in a convolutional layer is equal to `(width of filter` $\cdot$ `height of filter` $\cdot$ `# of filters in previous layer + 1)` $\cdot$ `# of filters in current layer`. Using this

(a) Layer 1: $((5 \cdot 5 \cdot 3) + 1) \cdot 32 = 2432$

(b) Layer 2: $((5 \cdot 5 \cdot 32) + 1) \cdot 64 = 51264$

(c) Layer 3: $((5 \cdot 5 \cdot 64) + 1) \cdot 128 = 204928$

In order to find the number of parameters from the fully connected layer we need to know the shape of the output from the third layer. To find this we multiply the dimensions of the image from the conv. layer by the number of filters. We also have to take into account the pooling layers which will have reduced the image shape by a factor of $\frac{1}{2^3}$, i.e. from 32×32 to 4×4. So there are $4 \cdot 4 \cdot 128 = 2048$ inputs to the FC layer.

(d) Layer 4: $(2048 \cdot 64) + 1 \cdot 64 = 131136$

(e) Layer 5: $(64 \cdot 10) + 1 \cdot 10 = 650$

Thus the total number of parameters in the network is 390410.

2 Task 2

a) Early stopping kicks in after 8 epochs. Training and validation loss are shown in Figure 1.

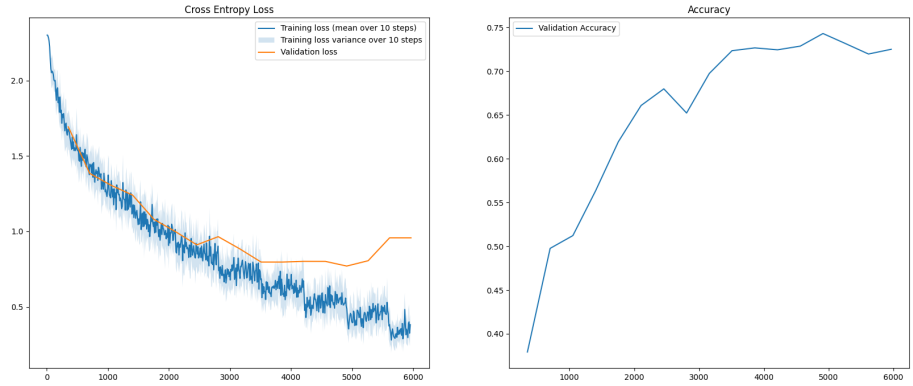


Figure 1: Loss

b) Final training, validation and test accuracy are shown in Table 1.

Table 1: Loss and accuracies

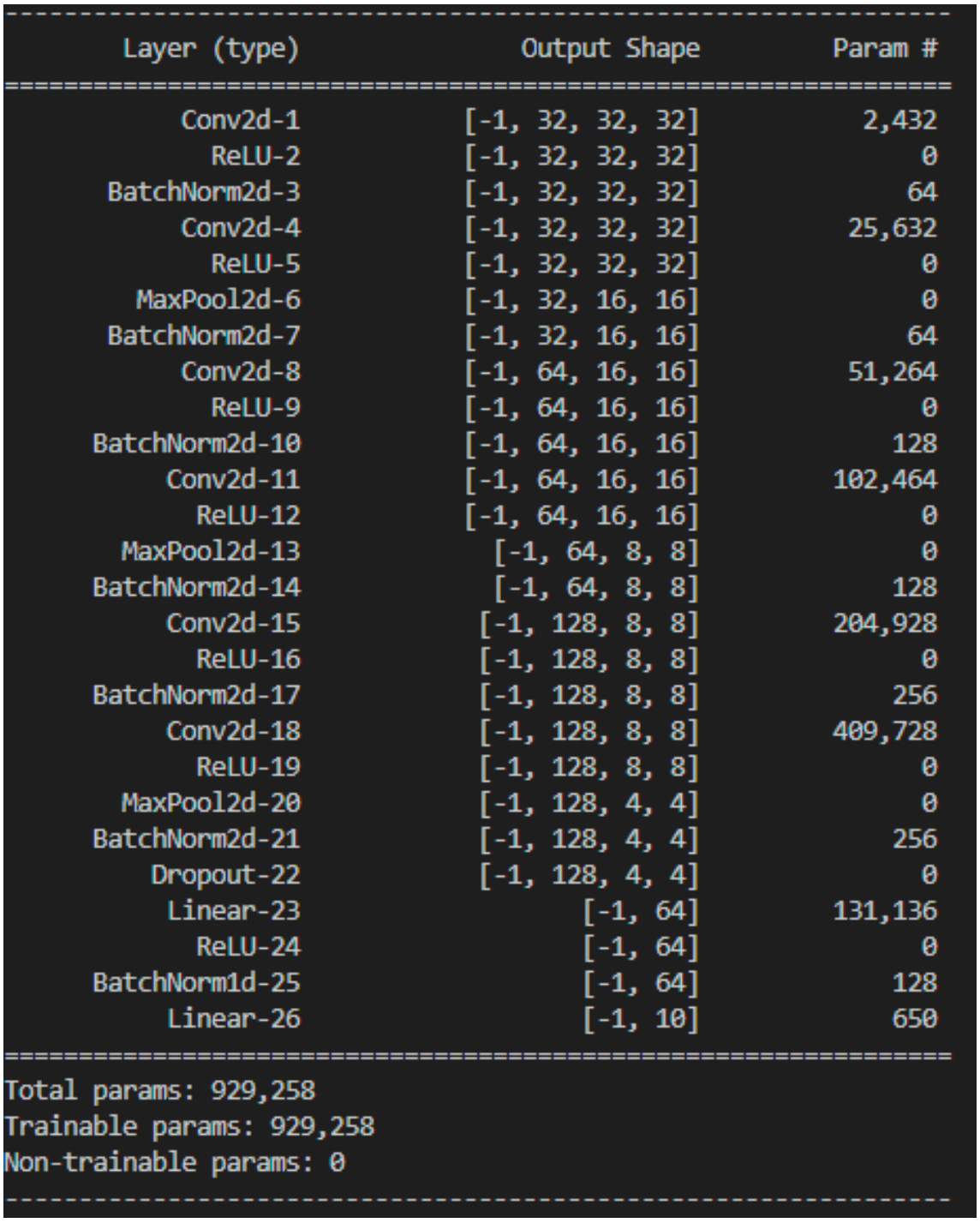
Train loss	0.37511
Train accuracy	0.87426

Validation loss	0.76999
Validation accuracy	0.743

Test loss	0.80094
Test accuracy	0.7424

3 Task 3

a) The model architecture is shown in Figure 2. Adam was used as the optimizer with `lr` = 0.001, `L2` regularization with $\lambda = 0.0001$, batch size of 64 and standard torch kaiming weight-initialization.



Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	2,432
ReLU-2	[-1, 32, 32, 32]	0
BatchNorm2d-3	[-1, 32, 32, 32]	64
Conv2d-4	[-1, 32, 32, 32]	25,632
ReLU-5	[-1, 32, 32, 32]	0
MaxPool2d-6	[-1, 32, 16, 16]	0
BatchNorm2d-7	[-1, 32, 16, 16]	64
Conv2d-8	[-1, 64, 16, 16]	51,264
ReLU-9	[-1, 64, 16, 16]	0
BatchNorm2d-10	[-1, 64, 16, 16]	128
Conv2d-11	[-1, 64, 16, 16]	102,464
ReLU-12	[-1, 64, 16, 16]	0
MaxPool2d-13	[-1, 64, 8, 8]	0
BatchNorm2d-14	[-1, 64, 8, 8]	128
Conv2d-15	[-1, 128, 8, 8]	204,928
ReLU-16	[-1, 128, 8, 8]	0
BatchNorm2d-17	[-1, 128, 8, 8]	256
Conv2d-18	[-1, 128, 8, 8]	409,728
ReLU-19	[-1, 128, 8, 8]	0
MaxPool2d-20	[-1, 128, 4, 4]	0
BatchNorm2d-21	[-1, 128, 4, 4]	256
Dropout-22	[-1, 128, 4, 4]	0
Linear-23	[-1, 64]	131,136
ReLU-24	[-1, 64]	0
BatchNorm1d-25	[-1, 64]	128
Linear-26	[-1, 10]	650

Total params: 929,258
Trainable params: 929,258
Non-trainable params: 0

Figure 2: Model architecture

b) Table 2 shows the final metrics. Figure 3 shows the validation accuracy as well as training and validation loss vs the number of training steps.

Table 2: Loss and accuracies

Train loss	0.2501
Train accuracy	0.9163

Validation loss	0.5866
Validation accuracy	0.8064

Test loss	0.6239
Test accuracy	0.8078

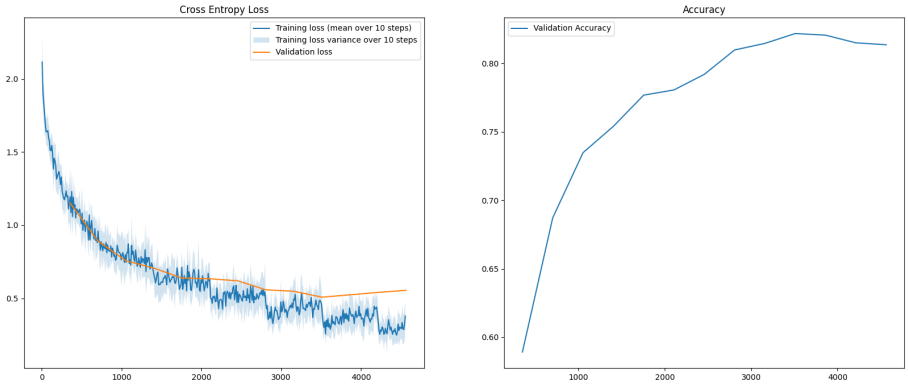


Figure 3: Loss and accuracy

c) I did the following to improve accuracy:

- Changed optimizer to Adam
- Batch normalization
- More conv layers
- Dropout on the last conv layer
- Weight decay

Most of the above mentioned methods are generally smart to try and often improve performance. The biggest improvement came from adding more layers. I think this worked well because classifying 10 classes is quite complex and therefore a deeper model was needed to learn this. Batch normalization and weight regularization helped speed up training significantly. I also tried using `Tanh` as the activation function in conjunction with xavier weight initialization, but that did not work well. Adding more dense layers did not help either.

d) The baseline model in Figure 4 has the topology given in the assignment text, but with Adam optimizer and batch normalization. The only difference is that the improved model has a few more layers, namely twice as many conv layers. This gave a huge boost in performance. See Figure 2 for the complete architecture.

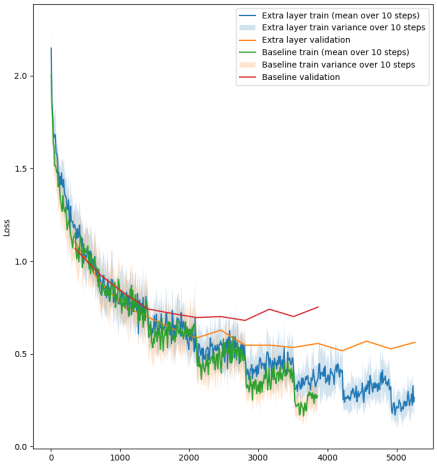


Figure 4: Before and after improvements

e) The model shown in Figure 3 gives an accuracy of $> 80\%$.

f) I see no signs of overfitting.

4 Task 4

Hyperparameters used:

- Optimizer: Adam
- Batch size: 32
- lr: $5 \cdot 10^{-4}$

Training results are shown in Figure 5. The final test accuracy is 0.885. Other metrics are shown in Table 3.

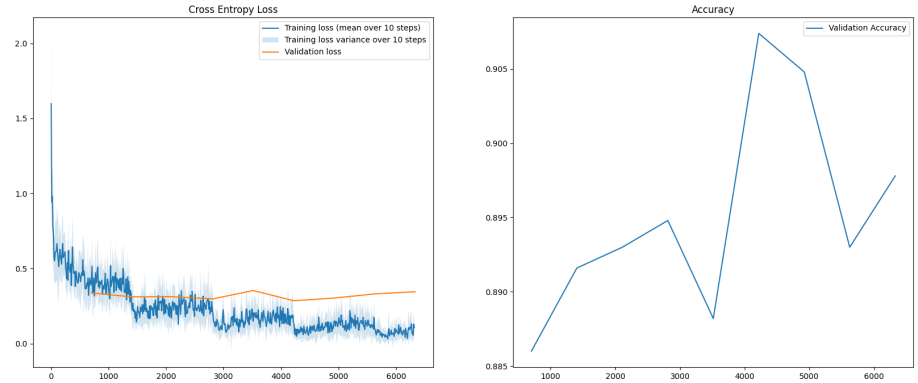


Figure 5: Resnet

Table 3: Loss and accuracies

Train loss	0.09205
Train accuracy	0.97057

Validation loss	0.33688
Validation accuracy	0.89540

Test loss	0.36257
Test accuracy	0.885