

# AssignmentReport-Group1

March 4, 2022

## 0.0.1 Assignment 3 Report

This is an outline for your report to ease the amount of work required to create your report. Jupyter notebook supports markdown, and I recommend you to check out this [cheat sheet](#). If you are not familiar with markdown.

Before delivery, **remember to convert this file to PDF**. You can do it in two ways: 1. Print the webpage (ctrl+P or cmd+P) 2. Export with latex. This is somewhat more difficult, but you'll get somewhat of a "prettier" PDF. Go to File -> Download as -> PDF via LaTeX. You might have to install nbconvert and pandoc through conda; `conda install nbconvert pandoc`.



## 1 Task 1

### 1.1 task 1a)

	2	-1	11	-2	-13	5	Str Pad
14	10	-4	8	2	-18		
-1	-1	-4	8	2	-18		
-5	6	6	6	6	6		
-9	-9	-9	-9	-9	-9		

## 1.2 task 1b)

It's the max pooling which reduces the sensitivity to translational variations.

The convolutional layer detects some feature, resulting in a high value in a specific pixel. The max pooling downsamples the input by extracting the max value from a grid, basically saying "in that area" such a feature is detected.



1.3 task 1c)

Task 1c

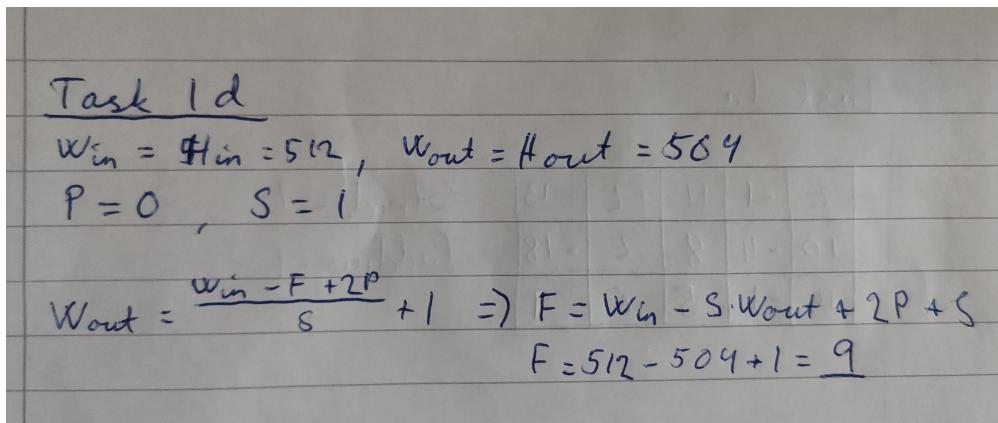
$$S=1, F=5, W_{in} = W_{out}, H_{in} = H_{out}$$

$$W = \frac{W - F + 2P}{S} + 1 \Leftrightarrow SW = W - F + 2P + S$$

$$\Rightarrow P = \frac{(S-1)W + F - S}{2} = \frac{(1-1)W + 5 - 1}{2} = \underline{\underline{2}}$$

(Some for height)

#### 1.4 task 1d)



#### 1.5 task 1e)

$$W_{out} = \frac{W_{in} - F}{S} + 1 = \frac{504 - 2}{2} + 1 = 252$$

#### 1.6 task 1f)

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1 = \frac{252 - 3 + 0}{1} + 1 = 250$$

#### 1.7 task 1g)

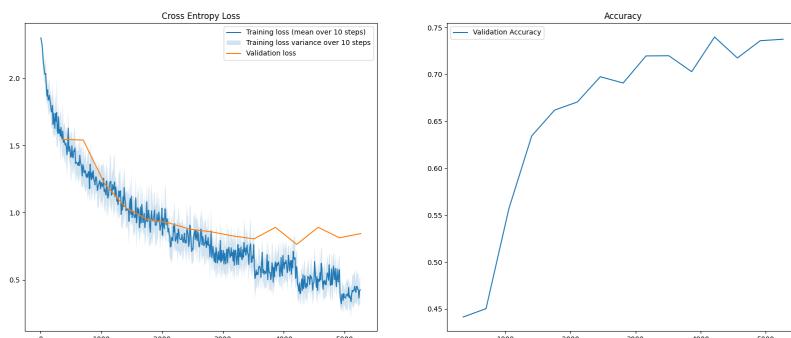
Dimensions in the convolutional layers (all filters of size 5x5): Layer | Filters | In | Out | Parameters — | — | — | — | — | 1 | 32 | 32x32x3 | 16x16x32 | 800 | 2 | 64 | 16x16x32 | 8x8x64 | 1600 | 3 | 128 | 8x8x64 | 4x4x128 | 3200

Parameters in the fully connected layers: Layer | Inputs | Neurons | Weights | Biases | Total parameters — | — | — | — | — | 4 | 2048 | 64 | 131072 | 64 | 131136 | 5 | 64 | 10 | 640 | 10 | 650

Total parameters =  $800 + 1600 + 3200 + 131136 + 650 = 137386$

## 2 Task 2

### 2.0.1 Task 2a)



## 2.0.2 Task 2b)

Final accuracy:

Train: 0.875

Validation: 0.726

Test: 0.272

## 3 Task 3

### 3.0.1 Task 3a)

Layer	Layer Type	Hidden Units / Filters	Activation Function
1	Conv2D	64	ReLU
1	BatchNorm2D	-	-
1	MaxPool2D	-	-
2	Conv2D	128	ReLU
2	BatchNorm2D	-	-
2	MaxPool2D	-	-
3	Conv2D	256	ReLU
3	BatchNorm2D	-	-
3	MaxPool2D	-	-
—	—	—	—
	Flatten	-	-
4	Fully-Connected	64	ReLU
4	BatchNorm1D	-	-
5	Fully-Connected	128	ReLU
5	BatchNorm1D	-	-
6	Fully-Connected	10	Softmax

**Model 1 Details:** - Optimizer: Stochastic Gradient Descent - Regularization: None - Learning rate: 0.03 - Batch size: 64 - Weight initialization: Default - Data augmentation: RandomHorizontalFlip and RandomRotate [-10, 10] degrees

**Accuracy:** - Train: 0.834 - Validation: 0.773 - Test: 0.790

Layer	Layer Type	Hidden Units / Filters	Activation Function
1	Conv2D	64	ReLU
1	BatchNorm2D	-	-
1	MaxPool2D	-	-
2	Conv2D	128	ReLU
2	BatchNorm2D	-	-
2	MaxPool2D	-	-
3	Conv2D	256	ReLU
3	BatchNorm2D	-	-
3	MaxPool2D	-	-
—	—	—	—

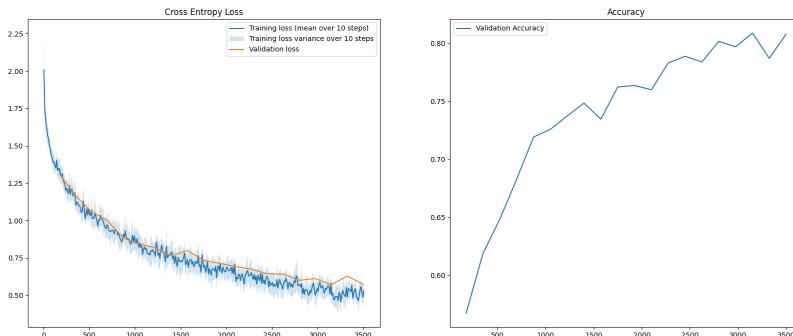
Layer	Layer Type	Hidden Units / Filters	Activation Function
	Flatten	-	-
4	Fully-Connected	64	ReLU
4	BatchNorm1D	-	-
5	Fully-Connected	128	ReLU
5	BatchNorm1D	-	-
6	Fully-Connected	10	Softmax

**Model 2 Details:** - Optimizer: Adam - Regularization: None - Learning rate: 0.001 - Batch size: 128 - Weight initialization: Default - Data augmentation: RandomHorizontalFlip and RandomRotate [-15, 15] degrees

**Accuracy:**

### 3.0.2 Task 3b)

Model	Train loss	Train accuracy	Validation accuracy	Test accuracy
1	0.52	0.826	0.774	0.767
2	0.43	0.867	0.816	0.825



**Model 1**

### 3.0.3 Task 3c)

**Data augmentation:** A little bit worse validation accuracy, but a lot worse train accuracy which means overfitting is reduced.

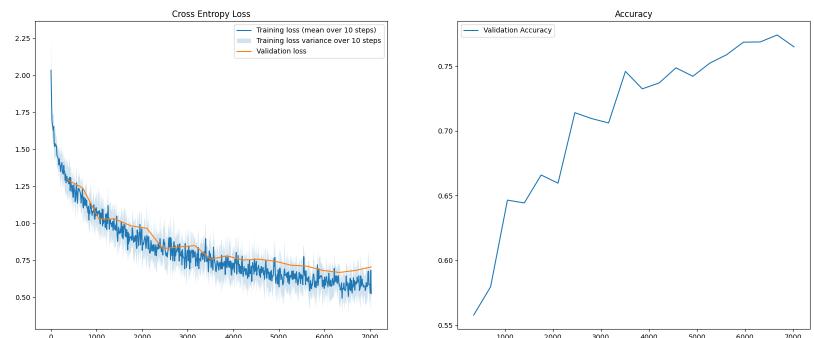
**Filter size:** Didn't make much of a difference. Smaller filter was slightly better.

**More filters:** Improved validation accuracy from 0.72 to 0.745

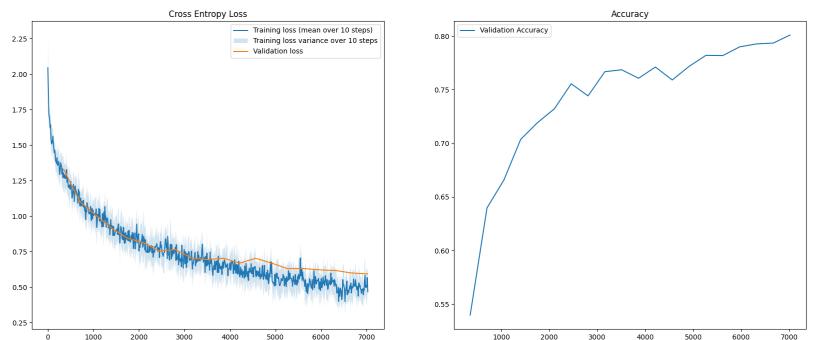
**Adam optimizer:** Much better validation accuracy (about 0.8). Showed signs of overfitting.

**Dropout:** Reduced overfitting and increased validation accuracy to 0.815

### 3.0.4 Task 3d)

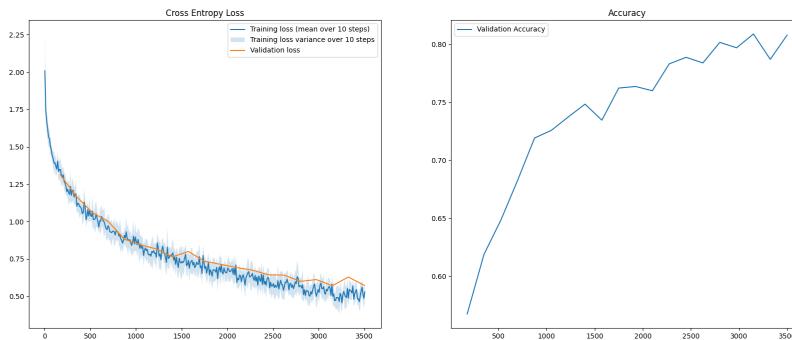


Before (SGD optimizer):



After (Adam optimizer):

### 3.0.5 Task 3e)



Final test accuracy:

0.825

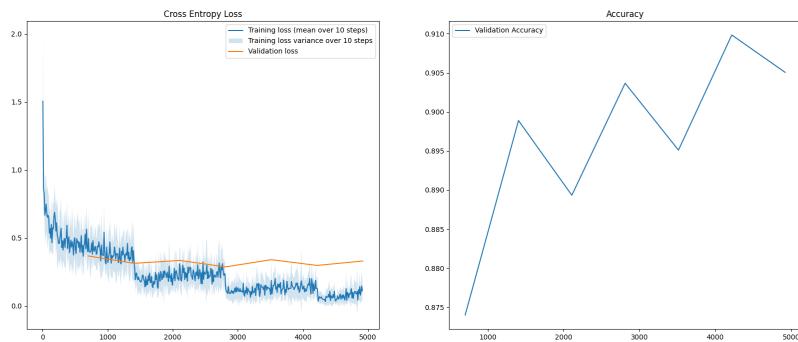
### 3.0.6 Task 3f)

The final model shows sign of overfitting. From the previous plot you can see that training loss is somewhat lower than validation loss.

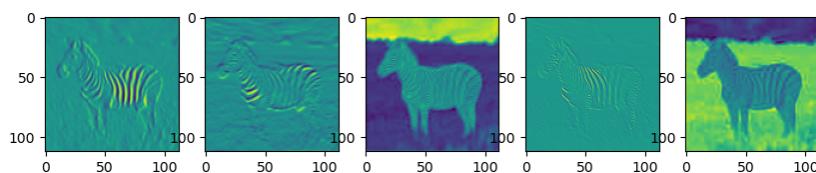
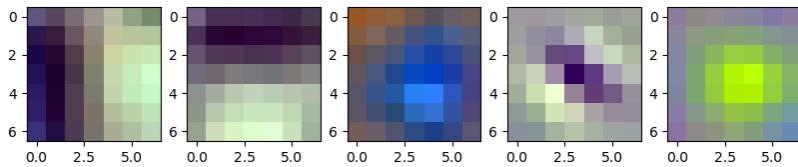
## 4 Task 4

### 4.1 Task 4a)

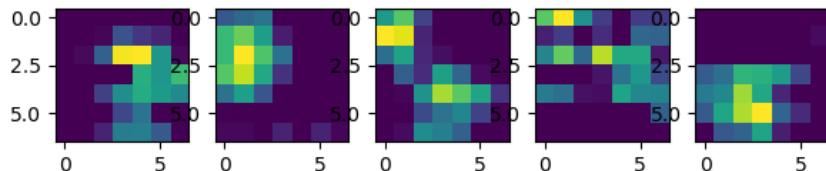
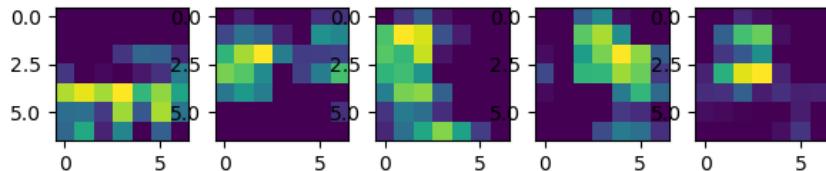
Final test accuracy: 0.893



### 4.2 Task 4b)



### 4.3 Task 4c)



[ ]: