Improving number recognition for neural networks

Author: Johan Bakker (j.m.bakker-2@student.tudelft.nl) | Supervisor: Wendelin Böhmer (j.w.bohmer@tudelft.nl)

1 Introduction

- Neural networks can recognize numbers and their performance is similar to humans
- When those numbers are put on a background the performance decreases
- Neural networks show the tendency to recognize the background, this behavior is called "spurious correlation"
- Therefore this research presents a new way to train the neural network to decrease spurious correlation

2 Research question

Does using similarity loss improve the out of distribution generalization of neural networks?

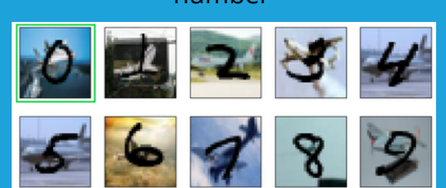
With the following subquestions:

1. *How does the addition of similarity loss influence the performance of the neural network on out of distribution data?*
2. *How effective is the similarity loss network compared to the baseline network with cross-entropy loss?*
3. *Does longer training benefit the similarity loss network?*

3 Methodology

Data generation

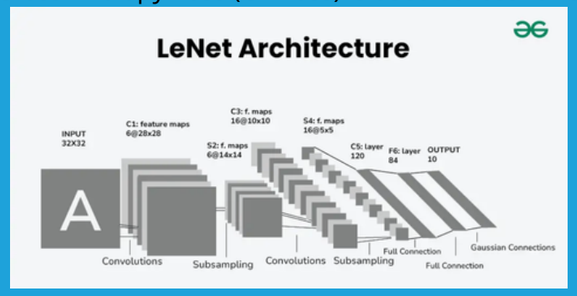
- The datasets used for this research are the MNIST dataset for the numbers and the CIFAR10 dataset for the background images
- These 2 datasets are combined in one dataset, the training part of this set has a disjunct set of CIFAR10 images for every MNIST number



3 Methodology (cont.)

Model

- As baseline model the LeNet model was used, this is a Convulution Neural Network (CNN)
- CNNs lay a filter over every pixel of a image such that for example corners or borders are recognized
- Small model (62k parameters) to prevent overfitting
- Cross-Entropy Loss (CE-Loss) is used as loss function



Similarity loss

- On top of using CE-Loss (comparing the last layers of the neural network), similarity loss is added
- Similarity loss compares second-to-last layers with each other
- The idea is that the similarity loss brings similar numbers closer together in the output of the neural network
- Similarity loss uses Mean Squared Error (MSE) as loss function, the definitions of this loss are shown below:

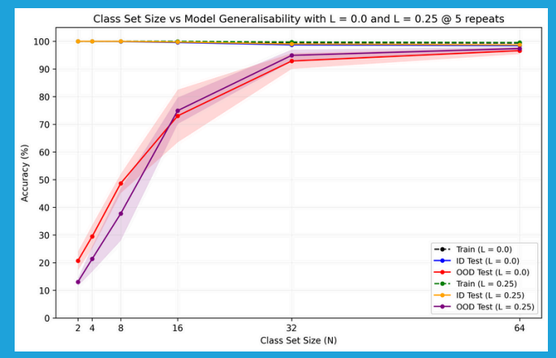
$$\mu_q = \frac{1}{|\mathcal{T}_q|} \sum_{i \in \mathcal{T}_q} \mathbf{z}_i, \quad S_i = \|\mathbf{z}_i - \mu_q\|_2^2$$

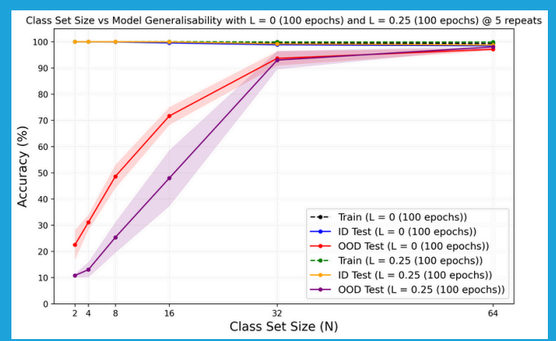
$$\mathcal{L}_{avg,\,q} = \sum_{i \in \mathcal{T}_q} S_i, \quad \mathcal{L}_{sim} = \sum_{q \in \mathcal{Q}} \mathcal{L}_{avg,\,q}$$

$$\mathcal{L}_{total} = \mathcal{L}_{crossentropy} + \lambda * \mathcal{L}_{sim}$$

4 Experimental evaluation

- Lines are means & shaded areas are 95% confidence intervals, red line is the baseline, purple line is baseline plus similarity loss with lambda 0.25
- Overlapping in the first graph indicates no significant change with the similarity loss
- Second graph shows that more epochs equals worse performance with similarity loss





5 Conclusions & Discussions

- Current implementation of the similarity loss does not yield a different outcome than the baseline
- This could be explained by the clustering of all images with the same label even when they have the same background
- A different implementation where the images with the same labels are only compared to images with different backgrounds could be a good start for further research