

# Project Documentation for Mathematics of Neural Networks

Johann Kipping

January 19, 2022

## Abstract

The aim of this project is to test the learned things in the first half of the lecture series on Neural Networks on the example of the Fashion-MNIST Dataset

Things that could be done on Fashion-MNIST

- Hyperparameter search
- Activation functions
- Initialization
- Optimizers
- Regularization (Overfitting?) + Generality
- FFT/im2col/Winograd
- Attribution
- Feature Visualization
- Adversarial Attacks
- Going deep?

## 0 Getting TF to run with a GPU with CUDA CC of 3.0

As this is a documentation of work it will be mentioned that a significant amount of time needed for the setup of computations was put into getting tensorflow to compute on the available GPU (Geforce 660 Ti). The major obstacle was the fact that this particular GPU has a *CUDA compute capability* of 3.0. The least tensorflow supports as of now is 3.5. This lead to the necessity of **downgrading to tensorflow version 1.12**. This might impede the ability for coverage of advanced topics.

## 1 Idea collection and planning

This chapter aims to collect ideas to most of the topics that were discussed in the lecture and continues to formulate a plan on how to tackle them in this project work.

Loss functions used is categorical cross entropy. Why not Mean squared or something else

### Hyperparameters:

Idea: Try several choices of hyperparameters to get a feel for the effects. Do this in a grid-like fashion.

## Activation functions / network structure

Idea: Test small network structures (fast trainable) for effect of structural changes and different choices of activation functions using knowledge from the lecture and Exercise 4 from sheet 6 as starting point for network evaluation.

## Initialization

Idea: Verify the validity of the three approaches discussed in the lecture. (Zero/Random/Known data)

## Optimizers

Idea: Use Adam for all computations to reduce complexity of project as it incorporates other approaches.

## Regularization (Overfitting?) + Generality

Idea: Test for overfitting and generalization with own pictures (for fun). Maybe try regularization if poor performance is observed.

## FFT/im2col/Winograd

Idea: Let tensorflow decide on the usage of different approaches as insight was achieved in the exercises and modifications are not cheap.

## Attribution/Feature Visualization

Idea: Visualize a net (with more layers) feature maps by using Saliency maps and grad-Cam. Try out guided backpropagation if possible! Relu has to be used! Gradient ascent (google feature visualization)!

Neuron/Channel/Layer(DeepDream)

## Adversarial Attack

Idea: Use an adversarial attack to trick a net.

## 2 Getting to know Neural Networks

This section will describe the first set of investigative experiments. Mainly regarding stuff of the first couple of lectures its main aim is to allow to get an understanding and feeling for the behavior of Neural Networks.

### 2.1 Starting out

As a starting point the network structure from Exercise 4 of sheet 6 was chosen to allow further investigation. As described in section 1 Adam was be used as the optimizer throughout this section in order to reduce complexity.

## 2.2 Initialization experiments

The methods for initializations are to be tested in this section:

- zero
- known data
- random

As mentioned before the net structure used for this will be structure from Exercise 4 of sheet 6.

The random initialization will be employed before the known data approach as the results from the random initialization will be used as the known data.

### 2.2.1 Zero initialization

As expected little to no improvement can be seen from training.

### 2.2.2 Random initialization

The baseline performance seen in the exercise. Uniform he initializer was used.

### 2.2.3 Known data initialization

This method allow the last two dense layers for classification to be further trained. This improved the networks performance for the training data, not for the test data

The resulting argument to be made is that from now

on random initialization will be used due to lacking of known data. Zero initialization is not to be used.

## 2.3 Hyperparameters for baseline network

The hyperparameters for the fixed network structure of the baseline network with Adam as a fixed optimizer are batch size, epochs, learning rate, activation functions and initializers. The latter two have been or will be examined at another point in this documentation. In the following part a small grid-like search is employed.

	32	64	128		32	64	128
0.1	x	x	x	0.1	x	x	x
0.01	x	x	x	0.01	x	x	x
0.001	x	x	x	0.001	x	x	x

Epochs: 5

Epochs: 5

## 2.4 Network structure

## 3 Regularization and Attribution

Check for generality and attribute in the same step by using an image from the test set and an image that was taken by me

Using gradient tape -> saliency map  
Global average pooling -> class activation map  
Grad-CAM! am ehesten