

# Machine Learning - Deep Learning Project 2

ANTON BORRIES

2204600

anton.borries@hhu.de

JOHANNES THIEL

2164216

johannes.thiel@hhu.de

SIMON SCHWÄR

extern

January 8, 2017

## Abstract

*We train a neural Network to recognize Images from the Cifar-10 Dataset.*

*In addition to this we take a look at Recurrent Neural Networks using Long short-term memory cells and use them to predict ....*

## I. ARCHITECTURE

Our Network consists out of 5 total Layers. The first two are both Convolutional Layers and the rest are Fully Connected.

We make use of  $2 \times 2$  pooling and local response normalization between the two convolutional layers.

The whole network has about 1 Million parameters.

We found that increasing the size of the fully connected layers did not result in greater accuracy, so we choose a small size for them to speed up the training process.

We used ReLu as our Activation Function for all Layers.

**Table 1:** Convolutional Layers

Layer	1	2
Input Size	$32 \times 32$	$16 \times 16$
Filter Size	$5 \times 5$	$5 \times 5$
Output Channels	64	64
Pooling	$2 \times 2$	$2 \times 2$
Activation Function	ReLu	ReLu

**Table 2:** Fully Connected Layers

Layer	3	4	5
Input Size	$8 \times 8 \times 64$	256	128
Activation Function	ReLu	ReLu	ReLu
Dropout	No	No	No

## II. TRAINING

We used a Batch Size of 128 to train our Network.

For the loss function we decided on using the cross entropy method.

$$C = -\frac{1}{n} \sum_z [y \ln a + (1 - y) \ln(1 - a)] \quad (1)$$

This is helping us, avoiding the learning slowdown [1].

We utilized the Adam Optimizer with a learning rate of 0.001 to train the Network.

Using a higher learning rate resulted in pretty poor results of about 15% test accuracy.

## III. EVALUATION

We evaluate our network by testing against the 10000 test images in the Cifar-10 Dataset.

We found that with a low amount of optimization iterations we get varying results. For example with 500 batches of optimization we get a test accuracy between 30% up to 36%. Increasing the amount of iterations we found the result to be both better and more stable. Going up to 2500 iterations we achieve a test accuracy of 54% - 56% and with 20000 iterations we reached a test accuracy of 68.78%.

#### IV. OTHER ARCHITECTURES

We tried out another architecture. It is similar to our previous one but uses  $3 \times 3$  convolutional filters and has an extra convolutional layer. We do not apply max pooling after this layer as it didn't improve our performance.

**Table 3:** Convolutional Layers

Layer	1	2	3
Input Size	$32 \times 32$	$16 \times 16$	$8 \times 8$
Filter Size	$3 \times 3$	$3 \times 3$	$3 \times 3$
Output Channels	64	64	64
Pooling	$2 \times 2$	$2 \times 2$	None
Activation Function	ReLU	ReLU	

**Table 4:** Fully Connected Layers

Layer	4	5	6
Input Size	$8 \times 8 \times 64$	256	128
Activation Function	ReLU	ReLU	ReLU
Dropout	No	No	No

#### V. DATASET DISTORTION

We applied distortion to every batch at runtime to increase the size of our dataset. For this we changed the hue, flipped the image and changed the brightness.

We could not find any significant increase to our accuracy doing this. We think this is the case because we don't run enough training iterations to make effective use of having more data.

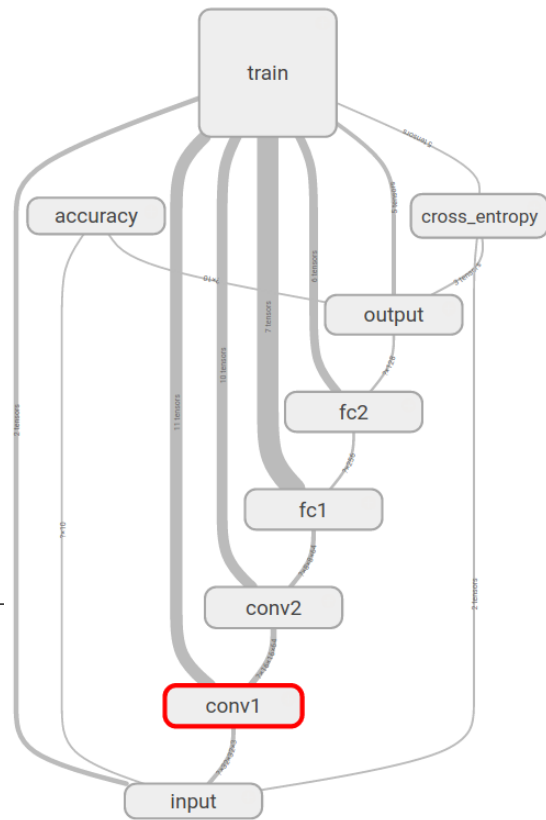
#### VI. DROPOUT

We tried applying a dropout layer for both our networks. This resulted in worse accuracy across the board.

We think this is because our fully connected layers are already pretty small and maybe don't have enough parameters after dropout to learn the data.

#### VII. VISUALIZATION

We visualize our network using TensorBoard.



**Figure 1:** Full network

#### VIII. RECURRENT NEURAL NETWORK ARCHITECTURE

We choose a very minimal architecture for this task as it already provides good results and our tests showed stacking more LSTM cells does

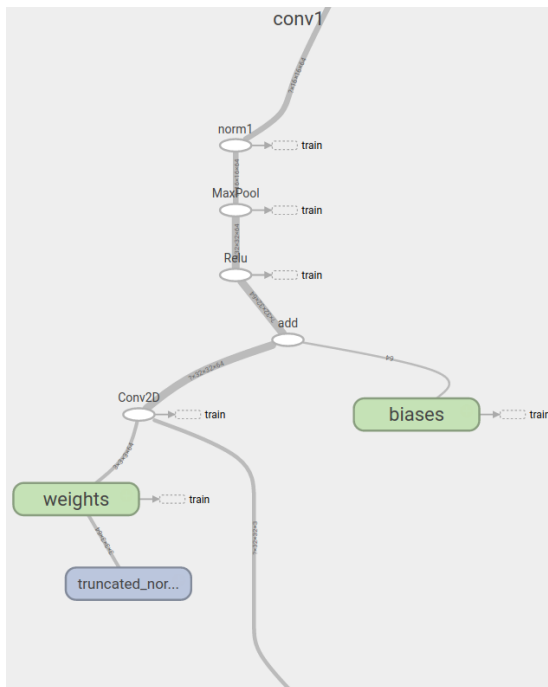


Figure 2: The first convolutional layer

not provide better results.

We use one LSTM cell followed by a fully connected layer with one unit.

## IX. LSTM CELLS

The idea of Long Short Term Memory cells is to give the network a memory of previous inputs, so they can be used to make predictions.

The LSTM cells have a hidden state that can be used to "memorize" information. During training the network learns which information to keep.

A LSTM takes three inputs:

- The normal input
- The output of the previous cell
- The hidden state of the previous cell

## X. TRAINING

We use the first 700 data entries as our training set and the last 300 datapoints as our test set. As our cost function we use the Mean Squared Error and optimize with the AdamOptimizer.

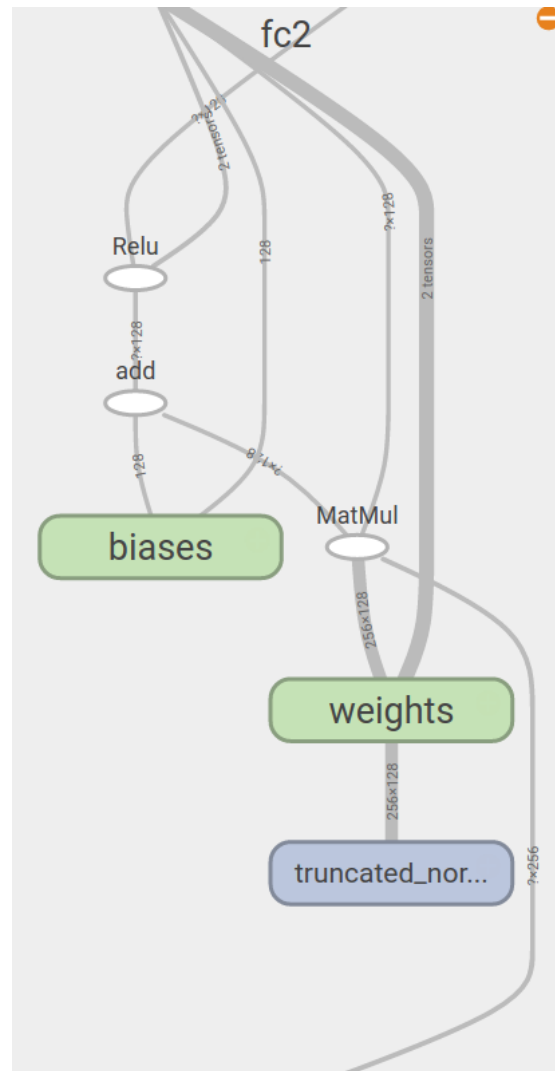


Figure 3: The first fully connected layer

After training out network on the first 700 datapoints we evaluate it using our test set. Over the whole testset we get an average MSE of 0.000159.

## XI. PREDICTION FIT

Plotting our prediction and the ground truth shows what our network learned and what it got wrong.

We can see that it is able to accurately predict the frequency of the curve but it seems to be a little bit offset on the y-axis.

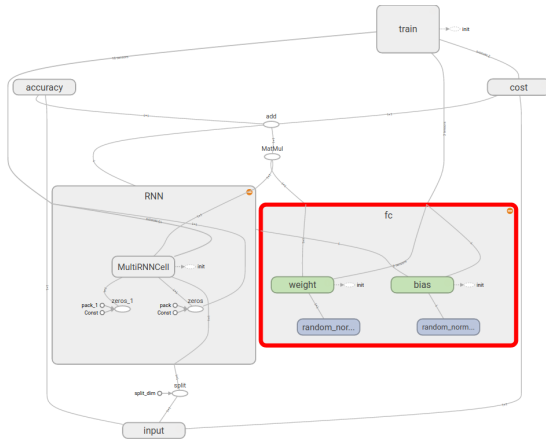


Figure 4: Vizusalization of our rnn

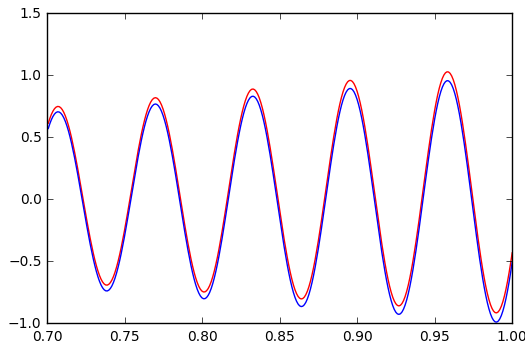


Figure 5: The red line is our prediction the blue one is the ground truth

## XII. REAL WORLD EXAMPLE

We choose a dataset of international airline passengers as a real world example for using Recurrent Neural Networks with LSTM cells. The dataset contains the amount of International airline passengers for every month between 1949 up until 1960. We can find a general upwards trend with periodic spikes.

For this task we used a RNN architecture similar to the one previously used but stacked 5 LSTM cells as this provided a good improvement over just using one.

We split up the data in a training and a test set. The training set contains the first 100 datapoints and the test set contains the other 44 datapoints.

After training we evaluate the network with

our test set and get an average MSE of 0.141. Looking at the prediction fit the network is able to accurately predict the general trend and frequency of the spikes.

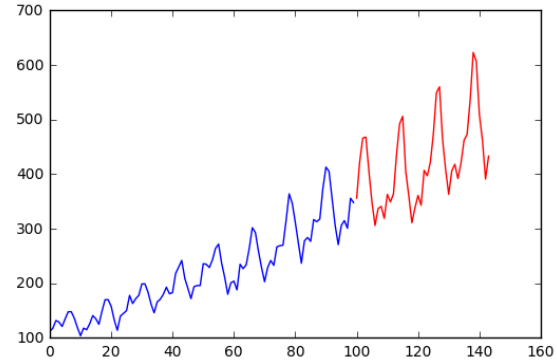


Figure 6: The red line is our prediction the blue one is the ground truth

## REFERENCES

- [1] Cross Entropy,  
<http://neuralnetworksanddeeplearning.com/chap3.html>, 06 12 2016.