Machine Learning - Deep Learning Project 2

Anton Borries
2204600
anton.borries@hhu.de

Johannes Thiel
2164216
johannes.thiel@hhu.de

Simon Schwär
extern (HSD/RSH)
simon.schwaer@hs-duesseldorf.de

January 8, 2017

Abstract

We train a neural Network to recognize images from the CIFAR-10 dataset.
Additionally, we take a look at Recurrent Neural Networks using Long Short-Term Memory cells and use them to predict an arbitrary dataset and the international airline passenger volume.

I. Architecture

Our network consists of 5 total layers. The first two are both convolutional layers and the latter are fully connected.
We make use of 2 x 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 chose 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 x 32	16 x 16
Filter Size	5 x 5	5 x 5
Output Channels	64	64
Pooling	2 x 2	2 x 2
Activation Function	ReLu	ReLu

Table 2: *Fully Connected Layers*

Layer	3	4	5
Input Size	8 x 8 x 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 to use the cross entropy method.

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

This is helping us to avoid 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.
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 results to be both better and more stable. Going up to 2500 iterations we achive 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 x 3 convolutional filters and has an additional convolutional layer. We did 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 x 32	16 x 16	8 x 8
Filter Size	3 x 3	3 x 3	3 x 3
Output Channels	64	64	64
Pooling	2 x 2	2 x 2	None
Activation Function	ReLu	ReLu	

Table 4: *Fully Connected Layers*

Layer	4	5	6
Input Size	8 x 8 x 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 observe any significant increase in our accuracy by doing this. We think this is the case because we did not 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 accross the board.
This might be because of our fully connected layers being rather small and maybe not having enough parameters after dropout to learn the data.

VII. Visualization

We visualize our network using Tensorboard.

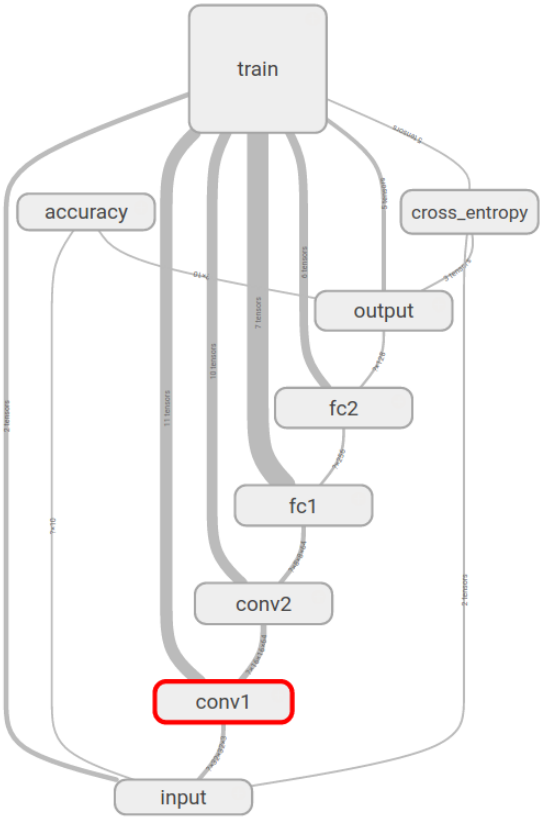

Figure 1: *Full network*

VIII. Recurrent Neural Network Architecture

We chose a very minimalist architecture for this task as it already provides good results and our tests showed that stacking more LSTM

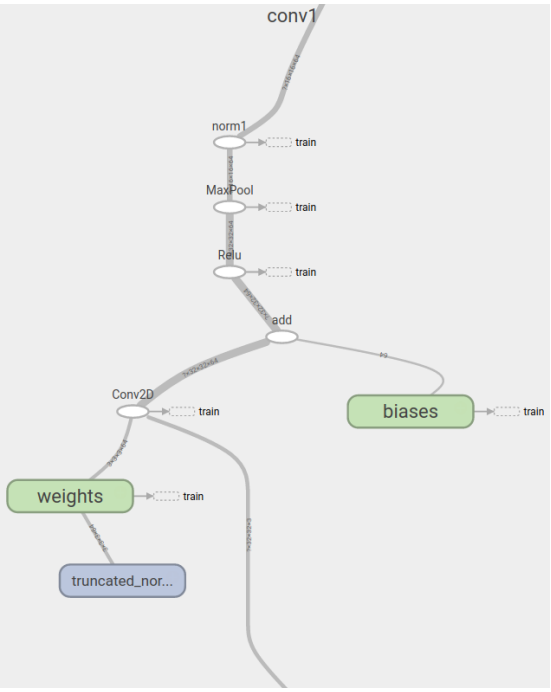

Figure 2: *The first convolutional layer*

cells did not provide better results.
We use one LSTM cell followed by a fully connected layer with one unit.

IX. LSTM Cells

The concept 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 for our training set and the last 300 datapoints for our test set.

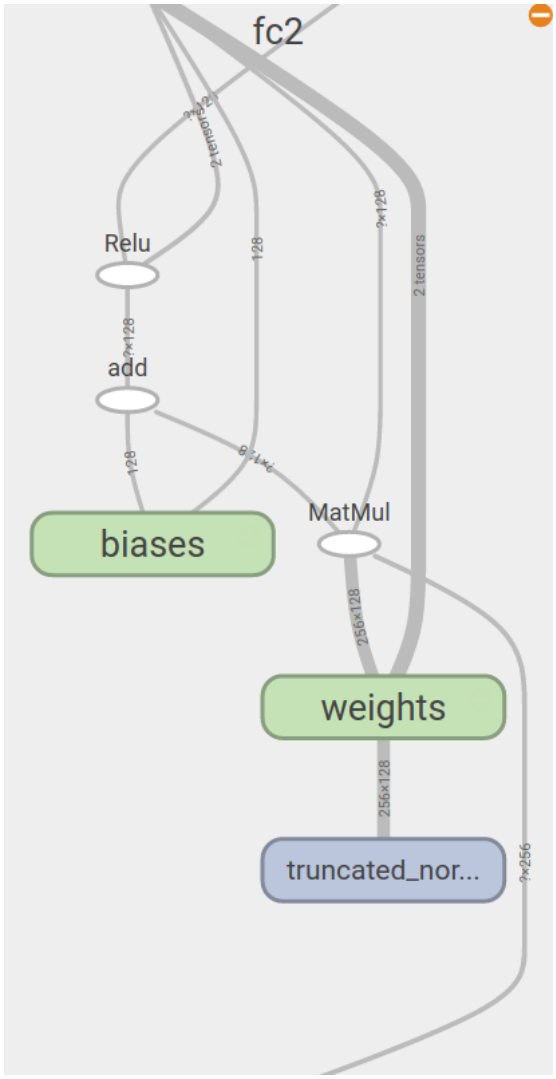

Figure 3: *The first fully connected layer*

As cost function we use the Mean Squared Error and optimize with the Adam optimizer.
After training our network on the first 700 datapoints, we evaluate it using our test set. Over the whole test set we get an average MSE of 0.000159.

XI. Prediction Fit

Plotting our prediction and the ground truth visualizes the performance of our network.
We can see that it is able to accurately predict the frequency of the curve but it seems to have

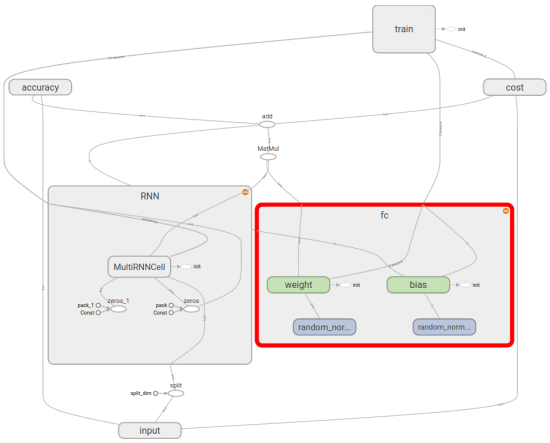

Figure 4: *Visualization of our RNN*

a little offset on the y-axis

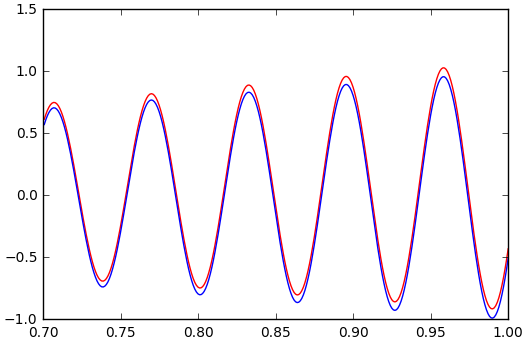

Figure 5: *The red line is our prediction, the blue one is the ground truth*

XII. Real World Example

We chose 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 from 1949 until 1960. We can observe 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 predicition fit the network is able to accuratly predict the general trend and frequency of the spikes.

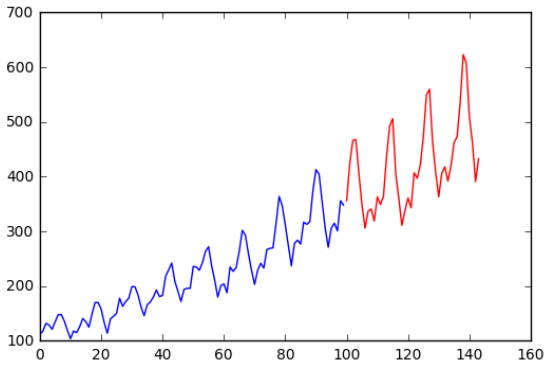

Figure 6: *The red line is our prediction, the blue one is the ground truth*

References

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