Solution for Project #2
Deep Learning, HHU 2016 / 2017

Thomas Germer Michael Janschek Patrick Brzoska

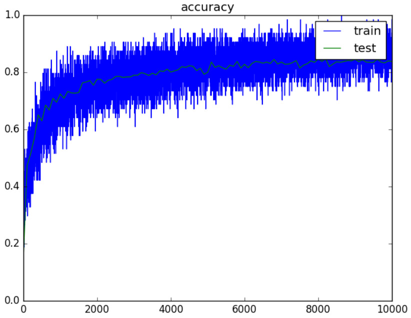
1 Code and Architecture

The code is provided as GitHub repository and is available under

`https://github.com/99991/DeepLearningProjects/tree/master/projects`

To test various neural network configurations, run

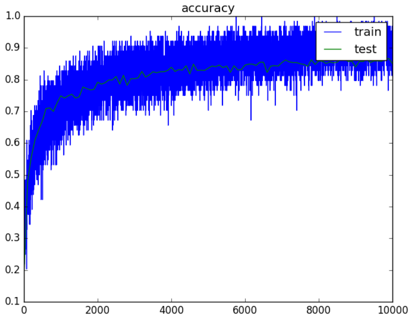
`https://github.com/99991/DeepLearningProjects/blob/master/projects/test/test_runner.py`



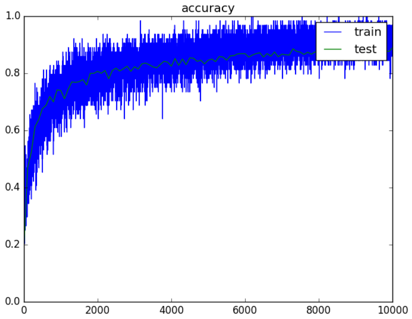
(a) accuracy

Parameter	Value
batch_size	64
num_batches	10001
kernel_size	3
num_kernels	32
num_hidden	500
depth	3
dropout_keep_probability	0.5

(b) Basic configuration



(a) accuracy



(b) accuracy

number of batches	number of kernels	Accuracy
30000	32	87%
10000	64	85.1%
10000	128	88.9%
14600	128	91%

(c) Increase of iterations and kernels

2 Tasks

1. **CNN on CIFAR-10**

   (a) To accommodate the higher complexity of the CIFAR-10 dataset we chose to train a relatively deep CNN compared to our first project. Also, with the number of batches at 10000 and each batch having 64 samples, the training time was increased.

   The leaky Rectified Linear Unit (ReLU) activation function was used to counter the "dying ReLU" problem. Instead of the function being zero when $x < 0$, it is set to a small negative value of $0.1 * x$.

   (b) Table 1b shows the configuration of our network, while figure 1a shows its achieved accuracy. With these parameters a test accuracy of 82.8% was reached, which seems to be a fairly good number compared to other implementations.

   (c) By increasing the number of batches to 30000 a test accuracy of 87% was achieved. Increasing the kernel size to 64 while retaining the batch size of 10000 an accuracy of 85.1% was reached. A kernel size of 128 resulted in an even higher test accuracy of 88.9%.

   (d) In [cite] average pooling has yielded into better result, but max pooling has worked better within our network. Also, using global average pooling as a regularizer as in [cite] has not had a positiv effect on the test accuracy.

(e) Using random cropping as a way to increase the data size has not lead to any significant improvement. Random brightness and saturation has not yielded any success, although this has helped within other implementations [cite].

(f) Adding dropout hasn't resulted in any success. With the configuration shown in 1b adding dropout resulted in a test accuracy of 74.8%.

(g) Please see figure 3.

2. **Recurrent Neural Network**

(a) Our RNN uses a hidden layer in front of the output layer

(b) A Long short-term memory (LSTM) cell utilizes a hidden state in order to calculate a prediction with each time step using information from the previous value of a given sample.

(c) c.

(d) d.

(e) e.

(f) f.

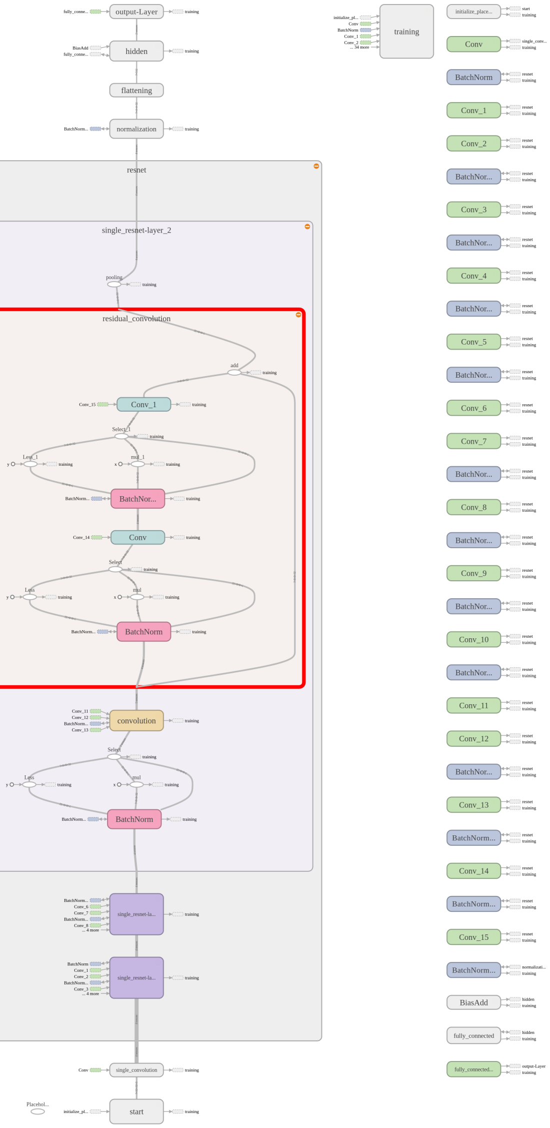


Figure 3: Tensorboard