

Solution of 5. Assignment

Deadline: 18.2.18

Points: (maximum 50+10 Points)

Exercise 1	Total
50+10	50+10

Exercise 1: action recognition in video sequences using LSTM

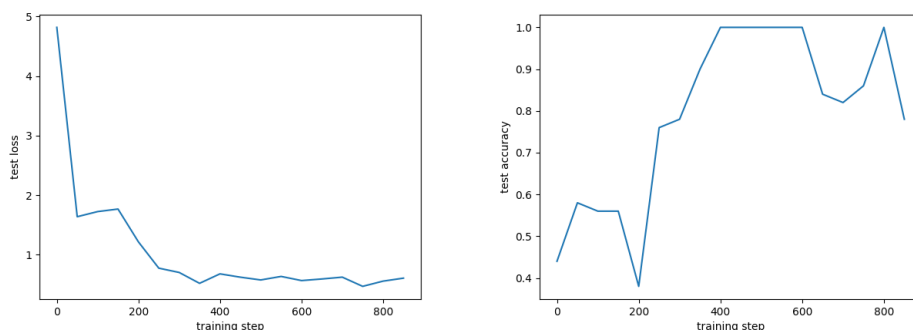
a)

The source code for our LSTM architecture can be found in **lstm.py**.

Our architecture is based on the example linked on the exercise sheet. Furthermore we added 2 convolutional layers and one fully connected layer between the input images and the lstm. Therefore the input into the lstm were the linearized feature maps of dimensions 16×7 with depth 16.

Additionally we applied dropout the output of the LSTM-net to tackle overfitting a little bit.

We trained with 70 sequences and tested with the remaining 5. Due to the small training and test sets, overfitting was quite strong and the test accuracy strongly depended on the random selection of the 5 test images. Also this might be the reason, why the test accuracy jumps wildly around.



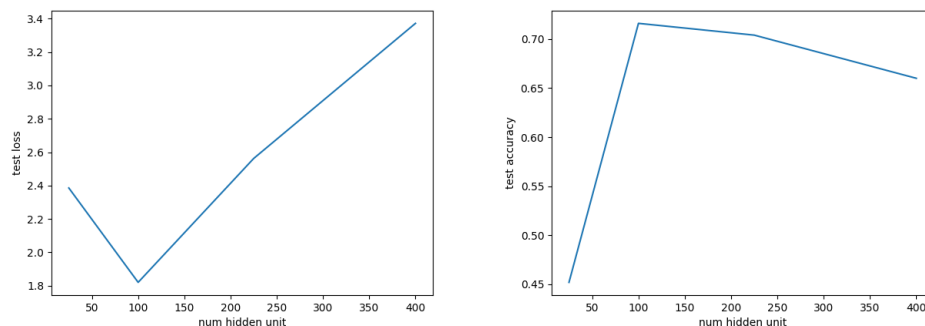
b)

In order to save computational capacities, we reduced the training steps for the 5-fold cross validation to 1500 steps.

Due to a lack of training examples we didn't have a distinct test set, but calculated the 'test' loss/accuracy by averaging over the validation sets of the 5-folds of the cross validation.

We tested the following number of hidden units to describe the lstm cell states: 25, 100, 225, 400.

The source code can be found in **cross.py**.



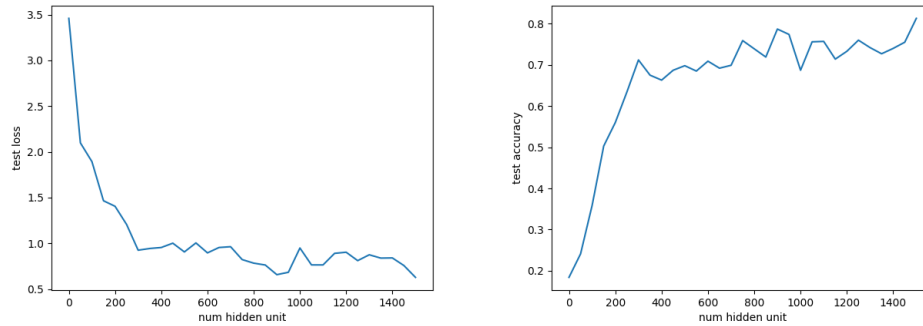
c)

We stitched together the sequences of mixed actions in the following way:

There are 5 actions with frames randomly distributed between 20 and 28 frames. Then we cut off all frames after the 100th frame.

The source code can be found in `seq_seg.py`

We achieved the following average loss and accuracy for classifying the single frames of the mixed sequences:



[] of 50+10 Points