

# Lab2 Report

Shuo Niu [sniu22@wisc.edu](mailto:sniu22@wisc.edu)

2017/2/15

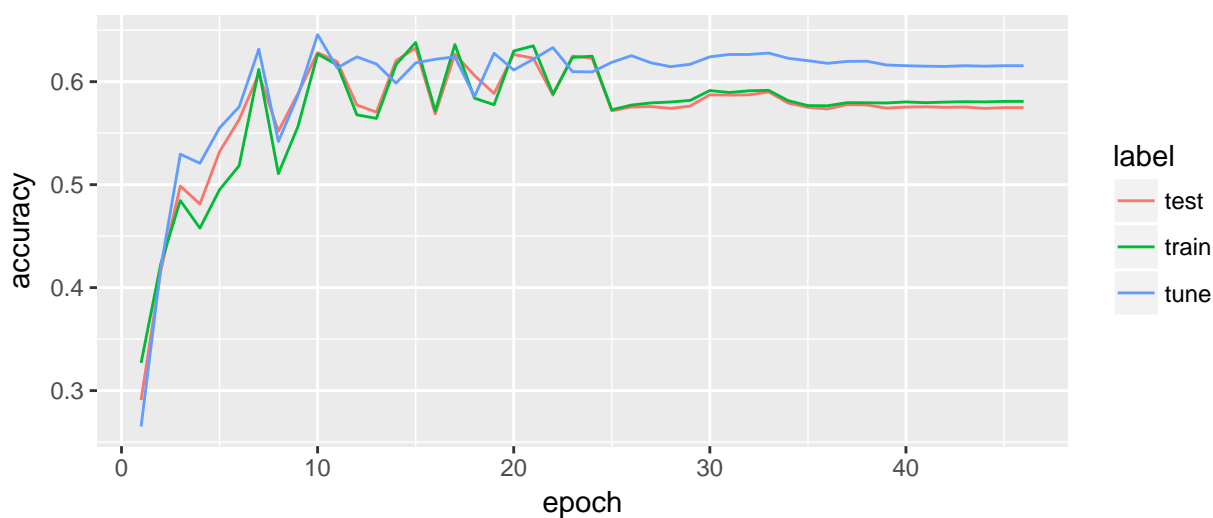
## Evaluation of the tricks in ANN

### Performance of different numbers of hidden units

With dropout at 0.5 and momentum at 0.9, I evaluated the performance of ann with 10,100,500 HUs.

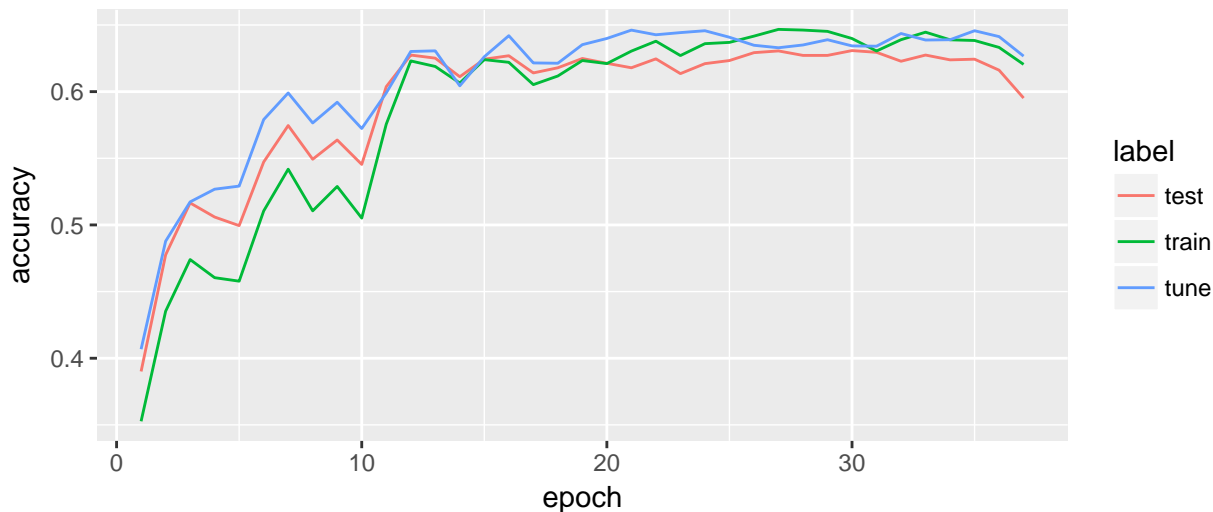
- ANN with 10 HUs:

#### ANN with 10 HUs

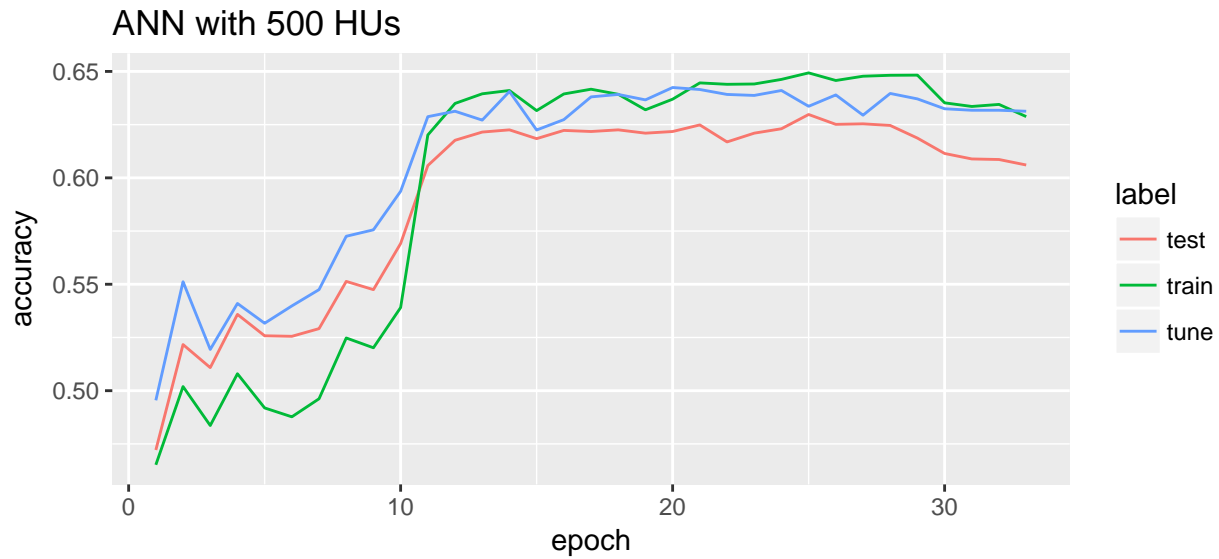


- ANN with 100 HUs:

#### ANN with 100 HUs



- ANN with 500 HUs:



- Summary: ANN with more hidden units performs better and more stable.

## Performance of momentum

With dropout at 0.5 and number of hidden units at 100, I tested the difference between with or without momentum term.

- Conclusion: momentum item help the ANN to converge faster. Under the current early stopping setting, ANN with momentum uses 19 epochs while ANN without momentum uses 40 epochs.

## Performance of dropout

With number of hidden units at 100, I tested the difference between with or without dropout strategy.

- The model with 50% dropout achieved 63.18% accuracy in 27 epochs for the testing data; The model without dropout achieved 62.22% accuracy in 14 epochs.
- Conclusion: Dropout has a positive influence on the accuracy but it may cause a slower convergence speed.

## Performance of decay

With dropout at 0.5 and momentum at 0.9, I evaluated the performance of decay item in ANN with 10 HUs. The tested decay coefficients are from  $1e-7$  to 100. The final model accuracy for each model are:

$1e-07$	$1e-06$	$1e-05$	$1e-04$	0.001	0.01	0.1	1	10	100
0.59	0.57	0.62	0.6	0.62	0.57	0.52	0.22	0.52	0.24

- Conclusion: The value of decay coefficient is extremely important. Unproper value may lead to disastrous results. Even the value is proper, there is no sign of improvement. So there is no decay term in my final model since it's risky.

## Final Model:

Choose the following parameters: `initial learning_rate = 0.05 ; dropout size = 0.5;`  
`momentum coefficient = 0.9 ; hidden units = 100`

### Model Details

- Use Relu for hidden units and Sigmoid for output units;
- Use the squared error loss function;
- Use early stopping for this network:
  - The maximum training epoch is set at 1000;
  - If the prediction accuracy of tune set continue decreasing for three epochs, continue to train the model for 50 epochs.
- Learning rate setting:
  - Every time the prediction accuracy of tune set decreases, make the learning rate shrinkage 40% :  
 $\text{new\_rate} = 0.6 * \text{old\_rate}$
  - Choose the weights with the best performance on tune set as the final weights.