

## Report

### Weka with some examples

Using the weather set as an example and with the default values we get:

=== Summary ===

Correctly Classified Instances	10	71.4286 %
Incorrectly Classified Instances	4	28.5714 %
Kappa statistic	0.3778	
Mean absolute error	0.287	
Root mean squared error	0.5268	
Relative absolute error	60.2616 %	
Root relative squared error	106.7798 %	
Total Number of Instances	14	

Modifying the learning rate from 0.3 to 0.1:

=== Summary ===

Correctly Classified Instances	10	71.4286 %
Incorrectly Classified Instances	4	28.5714 %
Kappa statistic	0.3778	
Mean absolute error	0.2937	
Root mean squared error	0.5163	
Relative absolute error	61.6852 %	
Root relative squared error	104.648 %	
Total Number of Instances	14	

We don't see much difference here, but as the learning rate is being modified, the RMSE increases too. Here's a screenshot of the same dataset with the learning rate at .99

=== Summary ===

Correctly Classified Instances	10	71.4286 %
Incorrectly Classified Instances	4	28.5714 %
Kappa statistic	0.3778	
Mean absolute error	0.294	
Root mean squared error	0.5316	
Relative absolute error	61.7433 %	
Root relative squared error	107.7483 %	
Total Number of Instances	14	

These variations were caused by modifying one single variable, in this case the learning rate. It does seem that the optimal training value is 0.3. With less value, it's having a bit more error and so with large values. It is worth mention that the training time is difficult in ANN. With a single neuron, the training time is not that great, but with big ANN with many layers and many hidden nodes, this time will increase with every input and expected output. Getting the weights of the connections right can insert a certain overhead that might not be optimal for small solutions. This is due to the level of specifications required

for real-life problems. They need enough information to model and generalize a useful network. Also, with big ANN, time and space is an issue to always keep in mind. The need to process huge amounts of data, for example millions of databases, makes large ANN slow when the hardware capabilities are not build to support this kind of queries.