Layered Cascade Neural Network

Tengfei Shen [u4981890]
Supervisor: Prof. Tom Gedeon
Main challenge for neural networks

Why model selection is important?

Model complexity is determined by:
- Weight number
- Magnitude
- Connection Topology

Model selection of the neural networks
- Perform a search through models
- Begin with an overly complex model, then simplify it.
- Begin with a simple model, then complicate it.
Cascade Neural Network

Start with a simple network, successively add a hidden neuron and each new hidden neuron receives inputs from all inputs and previous hidden neurons and connect to outputs.

Constructive Algorithms

- CasCor (Cascade Correlation) Algorithm - Fahlman, 1991
- CasPer Algorithm - Treadgold & Gedeon, 1997
  - A_CasPer - Treadgold & Gedeon, 1997
  - AT_CasPer - Treadgold & Gedeon, 1998
  - Layered_CasPer – Gedeon & Shen, 2011
The Layered_CasPer Algorithm

**Modification:**
Hidden neurons form as layers
No connections between neurons that are in the same layer

**Main feature:**
Less computational cost than A_CasPer
Objectives of Project:
Understand, implement and evaluate the Layered_CasPer constructive algorithm.

Contributions of Project:
- Cascade Neural Network Toolbox
- A series of experiments for evaluating the Layered_CasPer algorithm
Cascade Neural Network Toolbox

Is written in the programming language of Matlab
25 functions, over than 1800 lines
Took over 8 weeks

What the program can do:

• Implement CasPer, A_CasPer, AT_CasPer and Layered_CasPer cascade neural networks.
• Allow users design their experimental tasks.
• Display statistics for each stage of the network building process
• Display the final performance statistics and write as a csv file
• Save the matrix of final weights as a csv file
• Read a weight matrix from a csv file into current weight matrix
Layered Cascade Neural Network

Screenshots of the program

Output: Diagram of Training Errors
Layered Cascade Neural Network

Screenshots of the program

Output: Final performance statistics of the network

No. of Run: 1
Final performance statistics of the network which has the best validation result

Final Epochs: 3919
Number of installed hidden neurons: 6
Connection Crossings: 166887700
Final RMSE: 0.13613
Final Validation RMSE: 0.13666
Final Testing RMSE: 0.10898
Final Testing Correct Percentage: 0.98276
Validation of Layered Cascade Neural Network

Three Experiments:

1. Results Comparison of Classification Tasks
   - Compare to CasCor, CasPer, A_CasPer and AT_CasPer on 10 datasets (Proben1)
   - Has 6 of the best results on 10 datasets
   - Good performance on datasets which have a large number of inputs
   - High value of variance of results

2. Two Spirals Benchmark
   - Better than CasCor
   - Similar to CasPer, IDS Method, MLP with Neuro-Glial Network, Chaos Glial Network, Neuro-Fuzzy Classifier

3. Results Comparison of Regression Tasks
   - Compare to CasCor and A_CasPer on 5 datasets (Proben1)
   - Similar performance on 4 datasets, Worst performance on 1 dataset
   - Weak performance on datasets with noise
## Results of Classification Task (partial)

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Algorithm</th>
<th>AVG Best Number of hidden neuron</th>
<th>AVG Connection Crossing</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>StDv</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Median</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max</td>
</tr>
<tr>
<td>Card</td>
<td>at_casper</td>
<td>5.2</td>
<td>1.92E+08</td>
<td>12.32</td>
</tr>
<tr>
<td></td>
<td>acasper</td>
<td>5.2</td>
<td>1.92E+08</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>layere_casper1</td>
<td>7.6</td>
<td>4.84E+07</td>
<td>13.31</td>
</tr>
<tr>
<td></td>
<td>layere_casper2</td>
<td>6.3</td>
<td>1.23E+08</td>
<td><strong>12.38</strong></td>
</tr>
<tr>
<td>Gene</td>
<td>at_casper</td>
<td>5.2</td>
<td>1.92E+08</td>
<td>12.32</td>
</tr>
<tr>
<td></td>
<td>acasper</td>
<td>5.2</td>
<td>1.92E+08</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>layere_casper1</td>
<td>7.6</td>
<td>4.84E+07</td>
<td>13.31</td>
</tr>
<tr>
<td></td>
<td>layere_casper2</td>
<td>6.3</td>
<td>1.23E+08</td>
<td><strong>12.38</strong></td>
</tr>
<tr>
<td>Glass</td>
<td>at_casper</td>
<td>4.8</td>
<td>7.98E+06</td>
<td>28.42</td>
</tr>
<tr>
<td></td>
<td>acasper</td>
<td>4.8</td>
<td>7.98E+06</td>
<td>2.78</td>
</tr>
<tr>
<td></td>
<td>layere_casper1</td>
<td>4.3</td>
<td>7.95E+06</td>
<td>30.38</td>
</tr>
<tr>
<td></td>
<td>layere_casper2</td>
<td>5.7</td>
<td>1.87E+07</td>
<td><strong>27.42</strong></td>
</tr>
<tr>
<td>Heartc</td>
<td>at_casper</td>
<td>2.4</td>
<td>6.84E+06</td>
<td>18.34</td>
</tr>
<tr>
<td></td>
<td>acasper</td>
<td>2.4</td>
<td>6.84E+06</td>
<td>1.09</td>
</tr>
<tr>
<td></td>
<td>layere_casper1</td>
<td>6.1</td>
<td>4.05E+07</td>
<td>17.87</td>
</tr>
<tr>
<td></td>
<td>layere_casper2</td>
<td>6.3</td>
<td>1.39E+08</td>
<td><strong>17.67</strong></td>
</tr>
<tr>
<td>Horse</td>
<td>at_casper</td>
<td>3.2</td>
<td>2.36E+07</td>
<td>29.73</td>
</tr>
<tr>
<td></td>
<td>acasper</td>
<td>3.2</td>
<td>2.36E+07</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>layere_casper1</td>
<td>3</td>
<td>1.11E+07</td>
<td>26.26</td>
</tr>
<tr>
<td></td>
<td>layere_casper2</td>
<td>4</td>
<td>4.41E+07</td>
<td><strong>23.85</strong></td>
</tr>
</tbody>
</table>
Soybean Test Results

Layered Cascade Neural Network

![Box plot comparing test error percentages for different methods. The methods are A_CasPer, AT_CasPer, and Layered_CasPer. The box plots show the distribution of test error percentages with outliers.]
Two Spirals Benchmark

Fig. 14. The two spirals testing set

Fig. 15. Result of the Layered_CasPer algorithm

Fig. 16. Result of the CasPer algorithm (12 hidden neurons)

Fig. 17. Result of the CasCor algorithm (17 hidden neurons)
Potential of Layered Cascade Neural Network

- **Self—Evaluating Layered Cascade Neural Network**
  Let the algorithm decide whether to install the new hidden neuron in the current layer or to add it as a new layer by evaluation using candidate pool.

- **Random Limit Layered Cascade Neural Network**
  Generate a network with a random structure and reach extreme cases frequently, and then catch the best result from the extreme cases.

- **Limited Connections Layered Cascade Neural Network**
  Limit the number of input connections to further reduce the computational cost.
Conclusion

• Good performance on most datasets for classification and regression tasks.
• Less computational cost than CasPer and A_CasPer
• High value of variance of results for classification tasks
• Best on the datasets which have a large number of inputs
• Good potential

Any question?