Rough Set Application in Neural Network based Time Series Forecasting

Chakra Balayar
CS836 Rough Sets and Applications
Dept. of Computer Science, UofR

Outline

- Goals
- Time Series Forecasting?
- Neural Networks
- Rough Set Theory
- Some Applications
- Conclusions and Future Works
Goals

- **Forecasting** has intense social benefits
- review one of the most popular forecasting technique – **neural network**
- Search the possibility to integrate **Rough Set Theory** in neural network -> hybrid technique
- Present some application scenarios.

Time Series Forecasting

- **What is it?**
  - Extrapolating past behavior into the future.
- **Example data series**
  - Financial (e.g., stocks, rates, sales)
  - Physically observed (e.g., weather, sunspots)
  - Mathematical (e.g., Fibonacci sequence)
Time Series Forecasting - Difficulties

- Why is it difficult?
  - Insufficient data availability
  - Noise or Error prone data
  - Non-stationarity in data

Time Series Forecasting - Importance

- Why is it important?
  - Preventing undesirable events, circumstances preceding the event
  - Works as decision support system
  - Profiting from forecast (e.g., sales, stock)
Time Series Forecasting Methods

• Classical Methods
  – traditional rigid mathematical models fail to model turning points in time series.

• New generation Methodologies
  – Neural Network
  – Genetic Algorithms
  – Hybrid Knowledge based Systems

Neural Network?

- “A neural network is an interconnected assembly of simple processing units whose functionality is loosely based on the human neuron.”
- First model in 1940s which was less mathematical and lacks computing power.
Why neural network?

- founded on philosophy of mimicking knowledge acquisition and organization skills of human brain.
- offers significant support in terms of organizing, classifying, and summarizing data.
- helps to discern patterns among input data, requires few assumptions, and achieves a high degree of prediction accuracy.
- Most Importantly, **ability to approximate nonlinear function.**

Why neural network? (cont.)

- potentially promising alternative tool for recognition, classification and forecasting.
- high accuracy, adaptability, robustness, effectiveness and efficiency.
- can be used to synthesize disparate data and find hidden patterns and complex relationships between different data.
Neural Network Components

- **Input neurons (variables)**
  - Need expertise to decide the number of neurons.

- **Hidden layers and their neurons**
  - Use rule of thumb or trial and error
  - Most cases, one hidden layer is sufficient

- **Output neuron/s (variable/s)**
  - In time series, just one

- **Learning**
  - Various learning algorithms provided
Structure of A Neuron

Fig. 2. Basic Processing Unit

Learning the Network - Backpropagation

- Initialize the weights randomly with small numbers (e.g. between -0.05 and 0.05.)
- Until Convergence (low error or other stopping criteria) do
  - Present a training pattern
  - Calculate the error of the output nodes
  - Calculate the error of the hidden nodes
  - Continue propagating error back until the input layer is reached
  - Update all weights i.e. adjust the weights making the gap less.
### Example 1 – Soda Sales Forecasting

<table>
<thead>
<tr>
<th>Game</th>
<th>Tickets sold</th>
<th>Temperature</th>
<th>Home Team</th>
<th>Day of week</th>
<th>Soda Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game1</td>
<td>5</td>
<td>32</td>
<td>0</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Game2</td>
<td>2</td>
<td>45</td>
<td>1</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Game3</td>
<td>11</td>
<td>66</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Game4</td>
<td>7</td>
<td>89</td>
<td>1</td>
<td>6</td>
<td>5.6</td>
</tr>
<tr>
<td>Game5</td>
<td>2</td>
<td>45</td>
<td>1</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Game6</td>
<td>12</td>
<td>98</td>
<td>0</td>
<td>6</td>
<td>50</td>
</tr>
</tbody>
</table>

### Example 1 (cont.)

<table>
<thead>
<tr>
<th>Game</th>
<th>Input1</th>
<th>Input2</th>
<th>Input3</th>
<th>Input4</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game1</td>
<td>5</td>
<td>32</td>
<td>0</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Game2</td>
<td>2</td>
<td>45</td>
<td>1</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Game3</td>
<td>11</td>
<td>66</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Game4</td>
<td>7</td>
<td>89</td>
<td>1</td>
<td>6</td>
<td>5.6</td>
</tr>
<tr>
<td>Game5</td>
<td>5</td>
<td>45</td>
<td>1</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Game6</td>
<td>12</td>
<td>98</td>
<td>0</td>
<td>6</td>
<td>50</td>
</tr>
</tbody>
</table>
Learning the Network (cont.)

- The difference between the desired output and network answer is called the **network error** and should be as low as possible.
- Once all the rows of data are passed (one epoch), the errors are used to adjust the weights.
- The objective is to make the network answer close to desired one.
Weaknesses of neural network?

- No mechanism to filter input parameters and measure their dependency
- Noise in the data
  - In Neural network, the conflicting objects (same input different output) are noise.
- More failed cases may have greater potential for **failed** output than **not failed**.

Weaknesses of Neural Network

- Learning takes longer time to converge
- Not intuitive (can’t tell why matches $x$ to $o$)
  - black box problem or semantic concerns
- No theoretically well-founded way to **assess the quality** of BP learning
- Only guarantees local minima, not global
- Generalization not guaranteed
  - over-fitting/over-training problem
What then?

- To lessen learning, we need to minimize the input parameters and data set. - Rough Set Theory can be used
- To give semantics some kind of intermediary technique which has solid mathematical foundations necessary. - Rough Set can play important role

Rough Set Theory

- Based on sound mathematical tools
- Induction of approximation concepts which can play important role Uncertainty management.
- Can have bigger role in semantic meaning of data.
Rough Sets Properties

- Knowledge as a combination of attributes i.e. Information/Decision Systems (Tables)
- Indiscernibility – classifies objects.
- Set Approximation – boundaries among objects
- Reducts and Core – how preserving the quality
- Dependency of Attributes – dependency of decision attribute to condition attribute

Applicability of Rough Set Theory

- Two fold benefits
  - RST can be used as data pre-processing element in Neural network i.e. to lessen the training time
  - It also can be used as an inherent part of the model – to make neuro-rough (rough-neuro) model => Hybridization
Application I

- Market Failure Forecasting
  - The variables set comprised of eight financial ratios such as:
    
    net income to total asset,
    net income to sales,
    owner's equity to total asset,
    total borrowings and bonds payable to total asset,
    net working capital to total asset,
    cash flow to total liability,
    inventory turnover and
    current asset to current liability

Application I (cont.)

![Diagram of Rough Neural Network]

Fig. 3. Rough Neural Network
Application I (cont.)

Fig. 4 Rough Neural Hybrid Models

Application I - Results

- RNN1 and RNN2 outperformed NN
  - The accuracy was more than 6% better
  - RNN1 and RNN2 had almost similar results.
- Hybrid 1 and hybrid 2 outperformed all of them (NN, RNN1 and RNN2)
  - The accuracy was 4% better than RNN1 and RNN2
Application II

• Rough Set Backpropagation (RSBP) for Load Forecasting
• Noise and weak interdependency of data is avoided applying rough set
  – Dependency of attributes
  – Attribute reduction

Application II (cont.)

• Results were found encouraging
  – Out of seven, BP performed better in only one case where as RSBP performed better in other cases (both maximum and minimum load forecasting)
  – Proves the importance and applicability of Rough Sets in Neural networks.
The combination of rough set and neural network has just started to work out.

There are many areas to improve and consider.

Specially, neural network couldn’t get success due to
   – Over-fitting problem
   – Over-training problem
   – Local minima trapping problem

Over-fitting
   – a pitfall of NN
   – NN focuses on fitting training data so well that performance on out of sample data will decrease.
   – Solution is weight elimination
   – Which says if various model fits the data, then choose the simplest one.

Can we use Rough set?
Is this end? (cont.)

- Over-training
  - is one of the major pitfalls that must be avoided when developing neural networks.
  - occurs when a network memorizes the subtleties specific to the training set, lessening its ability to generalize to new data.

How can we utilize Rough Sets to avoid over-training?

Is this end?

- Local minima
  - characterized by a number of unhelpful features, such as local minima (which are lower than the surrounding terrain, but above the global minimum)

How can Rough Set approach can be utilized in finding global minima?
Conclusions

- Rough Set Theory provides solid mathematical tools to bring semantic meaning in neural network
- It also reduces learning time and the network complexity which are main obstacles on neural network development
- Many applications were developed utilizing the advantageous properties of RST and NN.

Future Works

- Using Rough Set for Task Allocation for available devices which is previously implemented using BP neural network
- Using rough set as input parameter reduction tool.
# References

5. Son, N. H., Szczuka, M. S., Slezak, D. Neural Networks Design: Rough Set Approach to Continuous Data,  

Thank you very much!