Finding Minimal Neural Networks for Business Intelligence Applications

Rudy Setiono
School of Computing
National University of Singapore
www.comp.nus.edu.sg/~rudys

Outline

- Introduction
- Feed-forward neural networks
- Neural network training and pruning
- Rule extraction
- Business intelligence applications
- Conclusion
- References
- For discussion: Time-series data mining using neural network rule extraction

Introduction

- **Business Intelligence (BI) :** A set of mathematical models and analysis methodologies that exploit available data to generate information and knowledge useful for complex decision-making process.

- Mathematical models and analysis methodologies for BI include various inductive learning models for data mining such as decision trees, **artificial neural networks**, fuzzy logic, genetic algorithms, support vector machines, and intelligent agents.

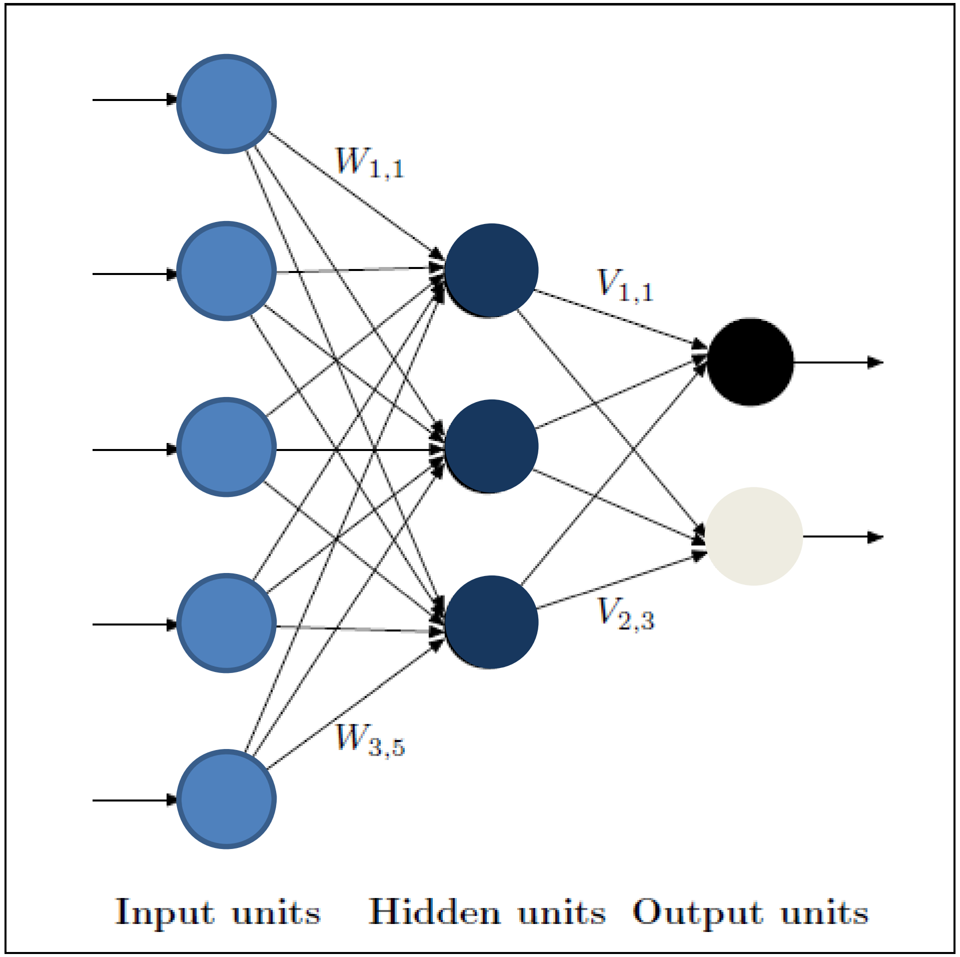
Introduction

BI Analytical Applications include:

- Customer segmentation: What market segments do my customers fall into, and what are their characteristics?
- Propensity to buy: What customers are most likely to respond to my promotion?
- Fraud detection: How can I tell which transactions are likely to be fraudulent?
- Customer attrition: Which customer is at risk of leaving?
- **Credit scoring: Which customer will successfully repay his loan, will not default on his credit card payment?**
- Time-series prediction.

Feed-forward neural networks

A **feed-forward** neural network with one hidden layer:

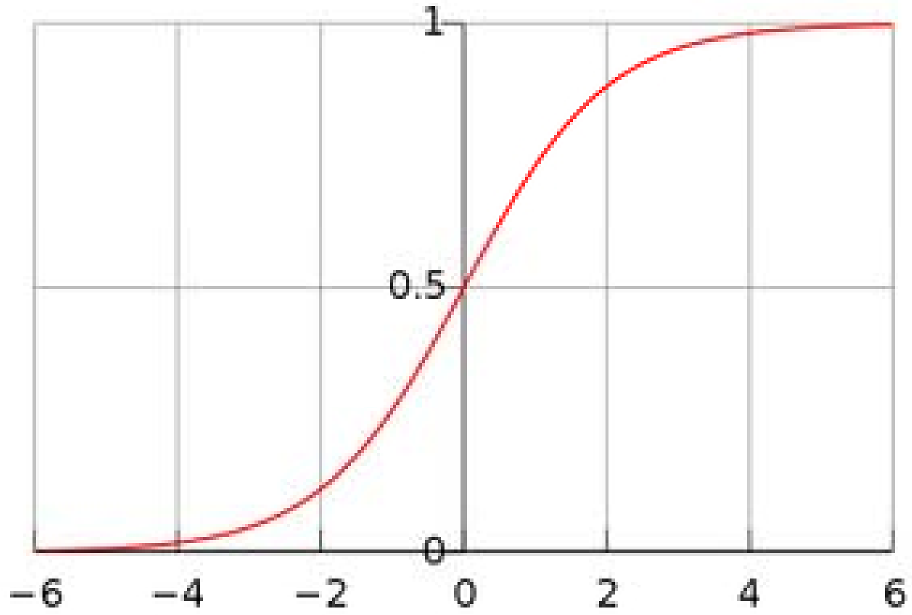


- Input variable values are given to the input units.
- The hidden units compute the activation values using input values and connection weight values W.
- The hidden unit activations are given to the output units.
- Decision is made at the output layer according to the activation values of the output units.

Feed-forward neural networks

Hidden unit activation:

- Compute the weighted input: $w_1x_1 + w_2x_2 + ... + w_nx_n$
- Apply an activation function to this weighted input, for example the logistic function $f(x) = 1/(1 + e^{-x})$:



Neural network training and pruning

Neural network training:

- Find an optimal weight (W,V).
- Minimize a function that measures how well the network predicts the desired outputs (class label)
- Error in prediction for i-th sample:

$$e_i = (\text{desired output})_i - (\text{predicted output})_i$$

- Sum of squared error function:

$$E(W,V) = \sum e_i^2$$

- Cross-entropy error function:

$$E(W,V) = - \sum d_i \log p_i + (1 - d_i) \log (1 - p_i)$$

d_i is the desired output, either 0 or 1.

Neural network training and pruning

Neural network training:

- Many optimization methods can be applied to find an optimal (W,V):
  - Gradient descent/error back propagation
  - Conjugate gradient
  - Quasi Newton method
  - Genetic algorithm
- Network is considered well trained if it can predict training data and cross-validation data with acceptable accuracy.

Neural network training and pruning

Neural network pruning: Remove irrelevant/redundant network connections

1. Initialization.

    (a) Let W be the set of network connections that are still present in the network and

    (b) let C be the set of connections that have been checked for possible removal

    (c) W corresponds to all the connections in the fully connected trained network and C is the empty set.

2. Save a copy of the weight values of all connections in the network.

3. Find w ∈ W and w Ù C such that when its weight value is set to 0, the accuracy of the network is least affected.

4. Set the weight for network connection w to 0 and retrain the network.

5. If the accuracy of the network is still satisfactory, then

    (a) Remove w, i.e. set W := W − {w}.

    (b) Reset C := ∅.

    (c) Go to Step 2.

6. Otherwise,

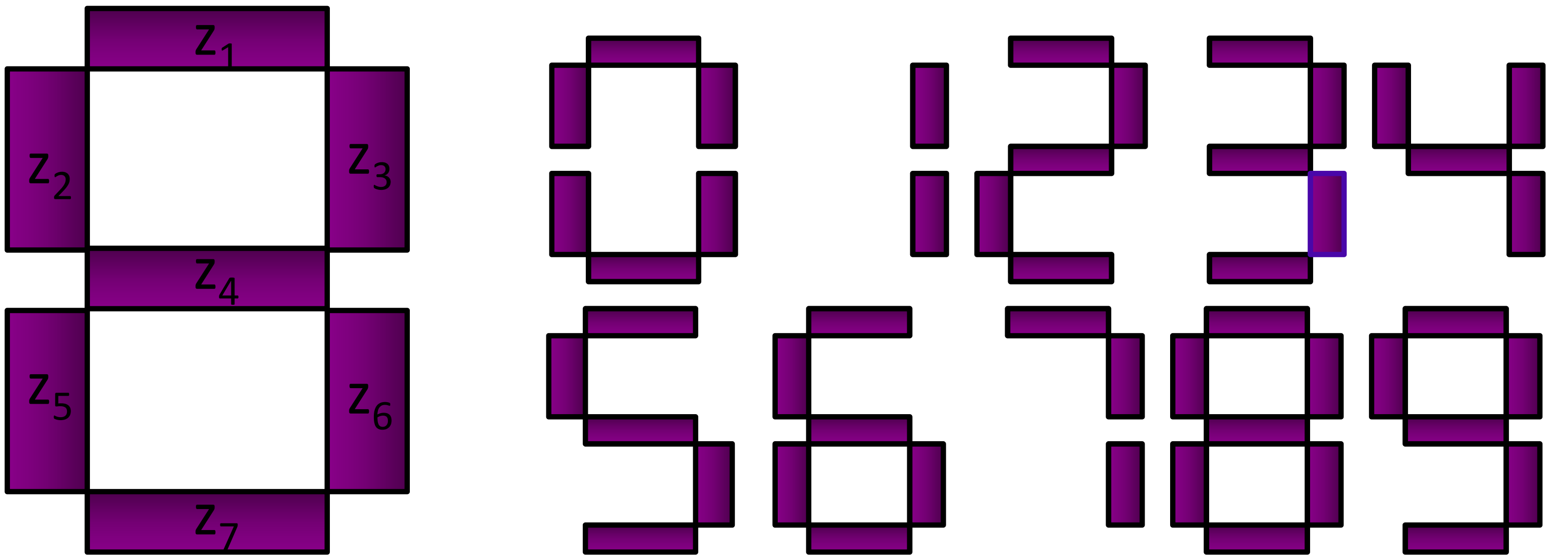
    (a) Set C := C ∪ {w}.

    (b) Restore the network weights with the values saved in Step 2 above.

    (c) If C ≠ W, go to Step 2. Otherwise, Stop.

Neural network training and pruning

Pruned neural network for LED recognition (1)

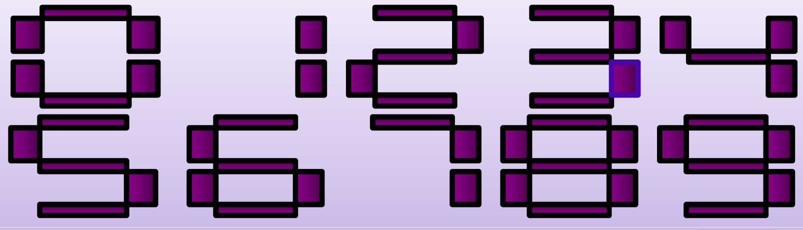


How many hidden units and network connections are needed to recognize all ten digits correctly?

Neural network training and pruning

Pruned neural network for LED recognition (2)

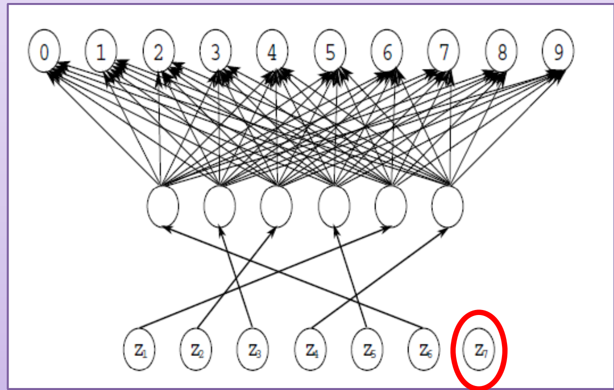
Raw data



Processed data

z_1	z_2	z_3	z_4	z_5	z_6	z_7	Digit
1	1	1	0	1	1	1	0
0	0	1	0	0	1	0	1
1	0	1	1	1	0	1	2
1	0	1	1	0	1	1	3
0	1	1	1	0	1	0	4
1	1	0	1	0	1	1	5
1	1	0	1	1	1	1	6
1	0	1	0	0	1	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9

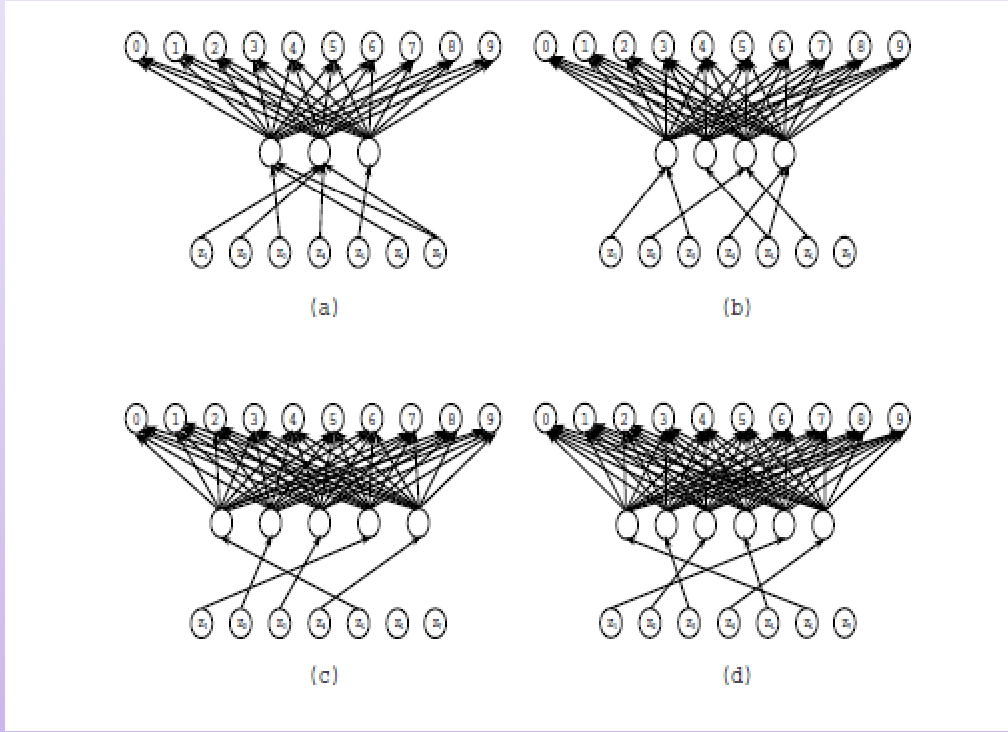
A neural network for data analysis



Neural network training and pruning

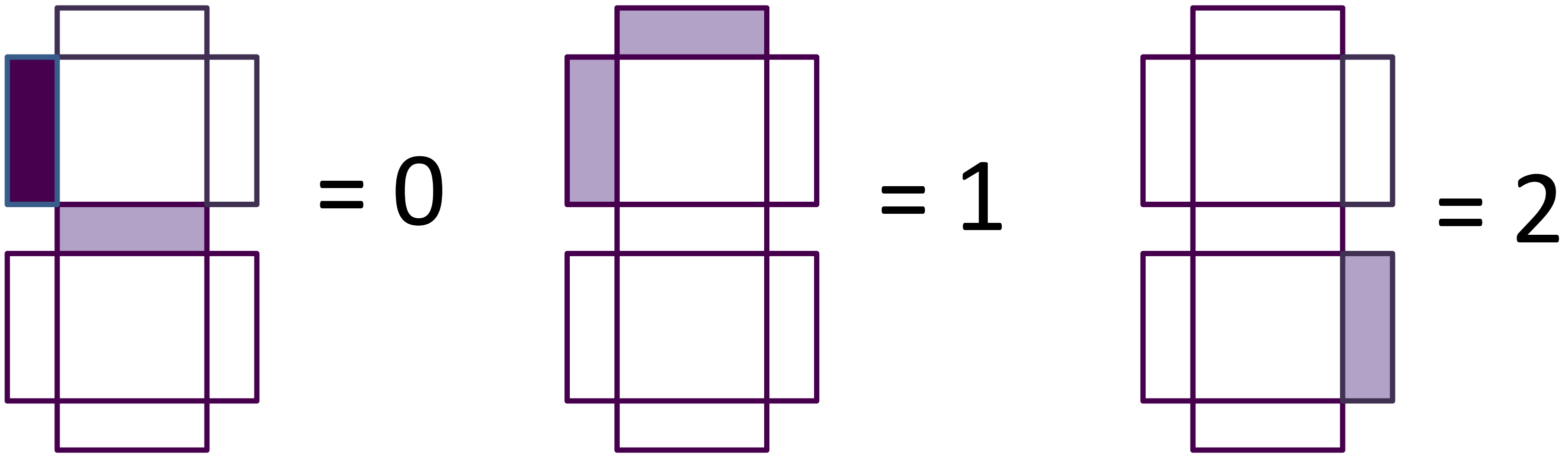
Pruned neural network for LED recognition (3)

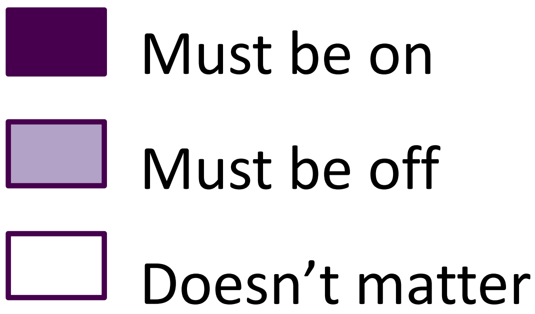
Many different pruned neural networks can recognized all 10 digits correctly.



Neural network training and pruning

Pruned neural network for LED recognition (4): What do we learn?





Classification rules can be extracted from pruned networks.

Rule extraction

Re-RX: an algorithm for rule extraction from neural networks

- New pedagocical rule extraction algorithm: Re-RX (Recursive Rule Extraction)
- Handles mix of discrete/continuous variables without need for discretization of continuous variables
  - Discrete variables: propositional rule tree structure
  - Continuous variables: hyperplane rules at leaf nodes
- Example rule:

  If Years Clients < 5 and Purpose ≠ Private Loan, then

    If Number of applicants ≥ 2 and Owns real estate = yes, then

      If Savings amount + 1.11 Income - 38249 Insurance - 0.46 Debt > -1939300, then Customer = good payer

      Else …
- Combines comprehensibility and accuracy

Rule extraction

Algorithm Re-RX(*S, D, C*):

Input: A set of samples *S* having discrete attributes *D* and continuous attributes *C*

Output: A set of classification rules

1. Train and prune a neural network using the data set *S* and all its attributes *D* and *C*.
2. Let *D'* and *C'* be the sets of discrete and continuous attributes still present in the network, respectively. Let *S'* be the set of data samples that are correctly classified by the pruned network.
3. If $D' = \emptyset$, then generate a hyperplane to split the samples in *S'* according to the values of their continuous attributes *C'* and stop. Otherwise, using only discrete attributes *D'*, generate the set of classification rules *R* for the data set *S'*.
4. For each rule R_i generated:

   If support(R_i) > δ_1 and error(R_i) > δ_2, then:

   – Let S_i be the set of data samples that satisfy the condition of rule R_i, and D_i be the set of discrete attributes that do not appear in the rule condition of R_i

   – If $D_i = \emptyset$, then generate a hyperplane to split the samples in S_i according to the values of their continuous attributes C_i and stop

   Otherwise, call Re-RX(S_i, D_i, C_i)

Business intelligence applications

- One of the key decisions financial institutions have to make

is to decide whether or not to grant credit to a customer who applies for a loan.

- The aim of **credit scoring** is to develop classification models that are able to

distinguish good from bad payers, based on the repayment behaviour of past

applicants.

- These models usually summarize all available information of an applicant in a score:
- P(applicant is good payer | age, marital status, savings amount, …).
- Application scoring: if this score is above a predetermined threshold, credit is granted;

otherwise credit is denied.

- Similar scoring models are now also used to estimate the credit risk of entire loan

portfolios in the context of Basel II.

Business intelligence applications

- **Basel II capital accord:** framework regulating minimum capital requirements for banks.
- Customer data $\Rightarrow$ credit risk score $\Rightarrow$ how much capital to set aside for a portfolio of loans.
- Data collected from various operational systems in the bank, based on which scores are periodically updated.
- Banks are required to demonstrate and periodically validate their scoring models, and report to the national regulator.

Experiment 1: CARD datasets.

- The 3 CARD datasets:

Data set	Training set		Test set		Total	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
CARD1	291	227	92	80	383	307
CARD1	284	234	99	73	383	307
CARD3	290	228	93	79	383	307

- Original input: 6 continuous attributes and 9 discrete attributes
- Input after coding: C_4,C_6,C_{41},C_{44},C_{49}, and C_{51} plus binary-valued

attributes D_1,D_2,D_3,D_5,D_7, … ,D_{40},D_{42},D_{43},D_{45},D_{46},D_{47},D_{48}, and D_{50}

Experiment 1: CARD datasets.

- 30 neural networks for each of the data sets were trained
- Neural network starts has one hidden neuron.
- The number of input neurons, including one bias input was 52
- The initial weights of the networks were randomly and

uniformly generated in the interval [−1, 1]

- In addition to the accuracy rates, the Area under the Receiver

Operating Characteristic (ROC) Curve (AUC) is also computed.

Experiment 1: CARD datasets.

$$AUC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\eta(\alpha_i,\beta_j)}{mn}$$

where

$$\eta(\alpha_i,\beta_j) = \begin{cases} 1 & : \alpha_i > \beta_j \\ 0 & : \text{otherwise} \end{cases}$$

- Where α_i are the predicted outputs for Class 1 samples i = 1,2, … m and β_j are predicted output for Class 0 samples, j = 1,2, … n.
- AUC is a more appropriate performance measure than ACC when the class distribution is skewed.

Experiment 1: CARD datasets.

Data set	#connections	ACC(θ_1)	AUC$_d$(θ_1)	ACC(θ_2)	AUC$_d$(θ_2)
CARD1 (TR)	9.13 ± 0.94	88.38 ± 0.56	87.98 ± 0.32	86.80 ± 0.90	86.03 ± 1.04
CARD1(TS)		87.79 ± 0.57	87.75 ± 0.43	88.35 ± 0.56	88.16 ± 0.48
CARD2(TR)	7.17 ± 0.38	88.73 ± 0.56	88.72 ± 0.57	86.06 ± 1.77	85.15 ± 2.04
CARD2(TS)		81.76 ± 1.28	82.09 ± 0.88	85.17 ± 0.37	84.25 ± 0.55
CARD3(TR)	7.57 ± 0.63	88.02 ± 0.51	88.02 ± 0.69	86.48 ± 1.07	87.07 ± 0.60
CARD3(TS)		84.67 ± 2.45	84.28 ± 2.48	87.15 ± 0.88	87.15 ± 0.85

- θ is the cut-off point for neural network classification: if output is greater than θ, than predict Class 1, else predict Class 0.
- θ_1 and θ_2 are cut-off points selected to maximize the accuracy on the training data and the test data sets, respectively.
- AUC$_d$ = AUC for the discrete classifier = (1 – fp + tp)/2

Business intelligence applications

Experiment 1: CARD datasets.

- One pruned neural network was selected for rule extraction for each of the 3 CARD data sets:

Data set	# connections	AUC (TR)	AUC (TS)	Unpruned inputs
CARD1	8	93.13%	92.75%	D_{12}, D_{13}, D_{42}, D_{43},C_{49},C_{51}
CARD2	9	93.16%	89.36%	D_{7}, D_{8}, D_{29}, D_{42}, D_{44},C_{49},C_{51}
CARD3	7	93.20%	89.11%	D_{42}, D_{43}, D_{47},C_{49},C_{51}

- Error rate comparison versus other methods:

Methods	CARD1	CARD2	CARD3
Genetic Algorithm	12.56	17.85	14.65
NN (other)	13.95	18.02	18.02
NeuralWorks	14.07	18.37	15.13
NeuroShell	12.73	18.72	15.81
Pruned NN (θ_1)	12.21	18.24	15.33
Pruned NN (θ_2)	11.65	14.83	12.85

Experiment 1: CARD datasets.

- Neural networks with just one hidden unit and very few connections outperform more complex neural networks!
- Rule can be extracted to provide more understanding about the classification.
- Rules for CARD1 from Re-RX:

  - Rule R_1: If $D_{12} = 1$ and $D_{42} = 0$, then predict Class 0,
  - Rule R_2: else if $D_{13} = 1$ and $D_{42} = 0$, then predict Class 0,
  - Rule R_3: else if $D_{42} = 1$ and $D_{43} = 1$, then predict Class 1,
  - Rule R_4: else if $D_{12} = 1$ and $D_{42} = 1$, then
    - Rule R_{4a}: If $R_{49} - 0.503R_{51} > 0.0596$, then predict Class 0, else
    - Rule R_{4b}: predict Class 1,
  - Rule R_5: else if $D_{12} = 0$ and $D_{13} = 0$, then predict Class 1,
  - Rule R_6: else if $R_{51} = 0.496$, then predict Class 1,
  - Rule R_7: else predict Class 0.

Experiment 1: CARD datasets.

- Rules for CARD2:
  - Rule R_1: If $D_7 = 1$ and $D_{42} = 0$, then predict Class 0,
  - Rule R_2: else if $D_8 = 1$ and $D_{42} = 0$, then predict Class 0,
  - Rule R_3: else if $D_7 = 1$ and $D_{42} = 1$, then
    - Rule R_{3a}: if $I_{29} = 0$, then
      - Rule R_{3a-i}: if $C_{49} - 0.583C_{51} < 0.061$, then predict Class 1,
      - Rule R_{3a-ii}: else predict Class 0,
    - Rule R_{3b}: else
      - Rule R_{3b-i}: if $C_{49} - 0.583C_{51} < -0.274$, then predict Class 1,
      - Rule R_{3b-ii}: else predict Class 0.
  - Rule R_4: else if $D_7 = 0$ and $D_8 = 0$, then predict Class 0,
  - Rule R_5: else predict Class 0.

Experiment 1: CARD datasets.

- Rules for CARD3:
  - Rule R_1: If $D_{42} = 0$, then
    - Rule R_{1a}: if $C_{51} > 1.000$, then predict Class 1,
    - Rule R_{1b}: else predict Class 0,
  - Rule R_2: else
    - Rule R_{2a}: if $D_{43} = 0$, then
      - Rule R_{2a-i}: if $C_{49} - 0.496C_{51} < 0.0551$, then predict Class 1,
      - Rule R_{2a-ii}: else predict Class 0,
    - Rule R_{2b}: else
      - Rule R_{2b-i}: if $C_{49} - 0.496C_{51} < 2.6525$, then predict Class 1,
      - Rule R_{2b-ii}: else predict Class 0,

Experiment 2: German credit data set.

- The data set contains 1000 samples,
- 7 continuous attributes and 13 discrete attributes.
- The aim of the classification is to distinguish between good and bad credit risks.
- Prior to training the neural network, the continuous attributes were normalized [0, 1],
- The discrete attributes were recoded as binary attributes.
- There were a total of 63 inputs.
- The binary inputs are denoted as D1,D2, . . .D56, and the normalized continuous attributes C57,C58, . . .C63.
- 666 randomly selected samples for training and the remaining 334 samples for testing.

Experiment 2: German credit data set.

- A pruned network with one hidden unit and 10 input units was found to have satisfactory accuracy.
- The relevant inputs are:

Input	Original attributes
$D_1 = 1$	Iff Status of checking account less than 0 DM
$D_2 = 1$	iff Status of checking account between 0 DM and 200 DM
$D_9 = 1$	Credit history: critical account/other credits existing (not at this bank)
$D_{21} = 1$	iff Saving accounts/bonds: less than 100 DM
$D_{22} = 1$	iff Saving accounts/bonds: between 100 DM and 500 DM
$D_{33} = 1$	iff Personal status and sex: male and single
$D_{36} = 1$	iff Other debtors/guarantors: none
$D_{38} = 1$	iff Other debtors/guarantors: guarantor
C_{57}	Duration in months
C_{59}	Installment rate in percentage of disposable income

Business intelligence applications

Experiment 2: (Partial) Rules for German credit data set.

- Rule R_1: if $D_1 = 1$ and $D_9 = 0$ and $D_{21} = 1$ and $D_{38} = 0$, then

  Class 0

- Rule R_2: else if $D_1 = 1$ and $D_9 = 0$ and $D_{22} = 1$ and $D_{33} = 0$, then predict Class 0,
- Rule R_3: else if $D_1 = 0$ and $D_2 = 0$ and $D_9 = 0$ and $D_{33} = 0$ and $D_{36} = 0$, then predict Class 0,
- Rule R_4: else if $D_2 = 1$ and $D_9 = 0$ and $D_{21} = 1$ and $D_{33} = 0$ and $D_{38} = 0$, then

  Class 0

- ……………………………………
- Rule R_9: else predict Class 1.

Experiment 2: German credit data set.

- Accuracy comparison of rules from decision tree method C4.5 and other neural network rule extraction algorithms:

Methods	Accuracy (Training set)	Accuracy (Test set)
C4.5	80.63%	71.56%
C4.5 rules	81.38%	74.25%
Neurorule	75.83%	77.84%
Trepan	75.37%	73.95%
Nefclass	73.57%	73.65%
Re-RX	**77.93%**	**78.74%**

Experiment 3: Bene1 and Bene2 credit scoring data sets.

- The Bene1 and Bene2 data sets were obtained from major financial institutions in Benelux countries.
- They contain application characteristics of customers who applied for credit.
- A bad customer is dened as someone who has been in payment arrears for more than 90 days at some point in the observed loan history.
- Statistics:

Data set	Attributes (original)	Attribute (encoded)	# training samples	# test samples	Good/Bads (%)
Bene 1	18 continuous 9 discrete	18 continuous 39 binary	2082	1041	66.7/33.3
Bene 2	18 continuous 9 discrete	18 continuous 58 binary	4793	2397	70/30

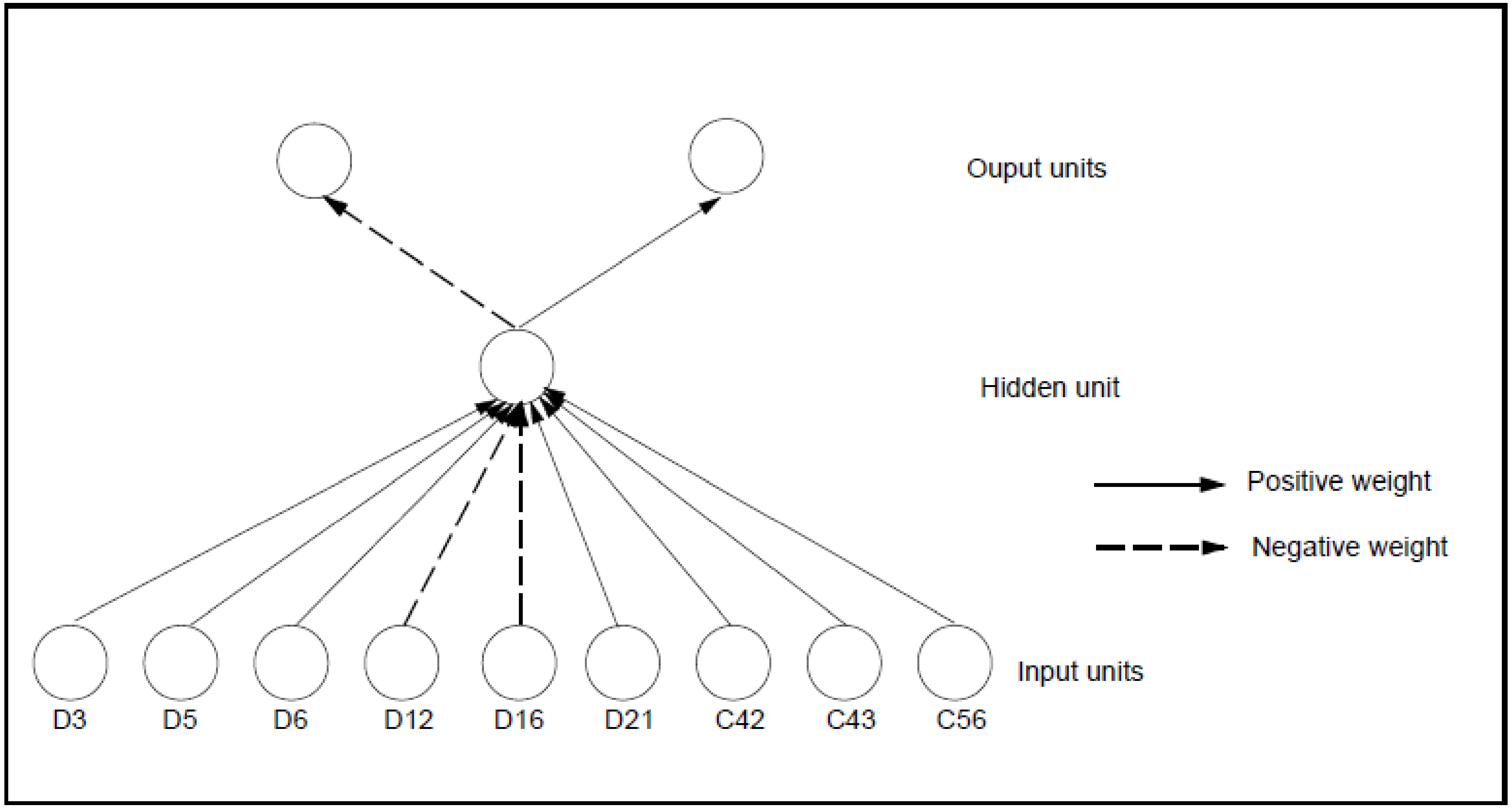
Business intelligence applications

Experiment 3: The original attributes of Bene1 credit scoring data set.

No	Attribute	Type	No	Attribute	Type
1	**Identification Number**	Continuous	2	**Amount of loan**	Continuous
3	**Amount of purchase invoice**	Continuous	4	**Percentage of financial burden**	Continuous
5	**Term**	Continuous	6	**Personal loan**	Nominal
7	**Purpose**	Nominal	8	**Private or Professional loan**	Nominal
9	**Monthly payment**	Continuous	10	**Saving account**	Continuous
11	**Other loan expenses**	Continuous	12	**Income**	Continuous
13	**Profession**	Nominal	14	**Number of years employed**	Continuous
15	**Number of years in Belgium**	Continuous	16	**Age**	Continuous
17	**Applicant type**	Nominal	18	**Nationality**	Nominal
19	**Marital status**	Nominal	20	**No. of years since last house move**	Continuous
21	**Code of regular saver**	Nominal	22	**Property**	Nominal
23	**Existing credit information**	Nominal	24	**No. of years as client**	Continuous
25	**No. of years since last loan**	Continuous	26	**No. of checking accounts**	Continuous
27	**No. of term accounts**	Continuous	28	**No. of mortgages**	Continuous
29	**No. of dependents**	Continuous	30	**Pawn**	Nominal
31	**Economical sector**	Nominal	32	**Employment status**	Nominal
33	**Title/salutation**	Nominal			

Experiment 3: Bene1 and Bene2 credit scoring data sets.

- A pruned neural network for Bene1:



Business intelligence applications

Experiment 3: Bene1 and Bene2 credit scoring data sets.

- The extracted rules for Bene1 (partial):

❑ Rule R: If Purpose = cash provisioning and Marital status = not married and Applicant type = no, then

- ❖ Rule R_1: If Owns real estate = yes, then
  - ✓ Rule R_{1a}: If term of loan < 27 months, then customer = good payer.
  - ✓ Rule R_{1b}: Else customer = defaulter.
- ❖ Rule R_2: Else customer = defaulter.

Business intelligence applications

Experiment 3: Bene1 and Bene2 credit scoring data sets.

- Accuracy comparison:

Data set	Methods	Accuracy (training data)	Accuracy (test data)	Complexity
Bene 1	C5.0 tree	78.91 %	71.06 %	35 leaves
	C5.0 rules	78.43 %	71.37 %	15 propositional rules
	NeuroLinear	77.43 %	72.72 %	3 oblique rules
	NeuroRule	73.05 %	71.85 %	6propositional rules
	Re-RX	**75.07 %**	**73.10 %**	**39 propositional rules**
Bene 2	C5.0 tree	81.80 %	71.63 %	162 leaves
	C5.0 rules	78.70 %	73.43 %	48 propositional rules
	NeuroLinear	76.05 %	73.51 %	2 oblique rules
	NeuroRule	74.27 %	74.13 %	7 propositional rules
	Re-RX	**75.65 %**	**75.26 %**	**67 propositional rules**

Experiment 4: Understanding consumer heterogeneity.

- Question: What are the factors that influence Taiwanese consumers' eating-out practices?
- The data set for this study was collected through a survey of 800 Taiwanese consumers.
- Demographic information such as gender, age and income were recorded. In addition, information about their psychological traits and eating-out considerations that might influence the frequency of eating-out were obtained.
- The training data set consists of 534 randomly selected samples (66.67%), and the test data set consists of the remaining 266 samples (33.33%).
- The samples were labeled as class 1 if the respondents' eating-out frequency is less than 25 per month on average, and as class 2 otherwise.

Experiment 4: Understanding consumer heterogeneity.

- 25 inputs with continuous values:

No	Input attribute	No	Input attribute
1	Indulgent	2	Family oriented
3	Adventurous	4	Focused on career
5	Knowledgeable about diet	6	Insensitive to price
7	Introverted	8	Inclined toward sales promotion
9	Stable life style	10	Preference for Asian meals
11	Meal importance/quality	12	Contented
13	Non assertive	14	Unsociable
15	Food indulgence	16	Not on diet
17	Specific product item	18	Tasty food
19	Hygiene	20	Service
21	Promotions	22	Pricing
23	Convenient location	24	Atmosphere
25	Image		

Personality and lifestyle

Eating-out considerations

Experiment 4: Understanding consumer heterogeneity.

Examples of questionnaires:

- Input 10. Preference for Asian meals
  - ✓ If I have a choice, I prefer eating at home.
  - ✓ I prefer Chinese cooking.
  - ✓ I must have rice everyday.
- Input 11. Meal importance/quality
  - ✓ I think dinner is the most important meal of the day.
  - ✓ I believe a brand or a product used by many people is an indication of its high quality.
  - ✓ Western breakfast is more nutritious than Chinese breakfast.
- Input 12. Contented
  - ✓ Overall, I am satisfied with my earthly possessions.
  - ✓ I am not demanding when it comes to food and drinks.
  - ✓ I usually do not mind the small details and fine dining etiquette.

Likert scale input

Factor analysis conducted to obtain the actual inputs for neural network

Experiment 4: Understanding consumer heterogeneity.

- 7 discrete inputs (demographics):

No	Input attribute	Possible values
26	Frequency of internet use	1, 2, 3, 4
27	Marital status	1,2
28	Education	1, 2, 3, 4, 5
29	Working status	1, 2, 3, 4, 5, 6, 7, 8
30	Personal monthly income	1, 2, 3, 4, 5
31	Household monthly income	1, 2, 3, 4, 5
32	Gender	1, 2
33	Age	1, 2, 3, 4, 5, 6

- Binary encoding:

Age		D_1	D_2	D_3	D_4	D_5	D_6
1	≤ 20	0	0	0	0	0	1
2	(20,30]	0	0	0	0	1	1
3	(30,40]	0	0	0	1	1	1
4	(40,50]	0	0	1	1	1	1
5	(50,60]	0	1	1	1	1	1
6	> 60	1	1	1	1	1	1

Experiment 4: Understanding consumer heterogeneity.

- The average accuracy rates and the number of connections of two sets of 30 pruned neural networks.

	One hidden unit	Two hidden units
Ave. training set accuracy	80.62 ± 0.34	80.67 ± 0.50
Ave. test set accuracy	73.60 ± 1.90	74.06 ± 1.72
Ave. # connections	12.47 ± 3.97	14.23 ± 4.49

- One of the pruned networks is selected for rule extraction.

Experiment 4: Understanding consumer heterogeneity.

- Rule involving only the discrete attributes:
  - Rule R1: If $D_{26} = 1$ and $D_{48} = 0$, then predict Class 1.
  - Rule R2: If $D_{28} = 0$, then predict Class 1.
  - Rule R3: If $D_{26} = 0$ and $D_{28} = 1$, then predict Class 2.
  - Rule R4: If $D_{28} = 1$ and $D_{48} = 1$, then predict Class 2.
  - Rule R5: Default rule, predict Class 2.
- Relevant inputs:

Input	Original attributes
$D_{26} = 0$	iff frequency of internet use = 1, 2, 3
$D_{27} = 0$	Iff frequency of internet use = 1, 2
$D_{28} = 0$	iff frequency of internet use = 1
$D_{48} = 0$	iff personal monthly income = 1

Experiment 4: Understanding consumer heterogeneity.

- Complete rule set:
  - ❑ Rule R_1: If D_{26} = 1 and D_{48} = 0, then
    - ❖ Let Sum = $C_7 + 1.28\ C_{13} - 2.03\ C_{23}$.
    - ❖ Rule R_{1a}: If Sum ≥ -6.46, then predict Class 1,
    - ❖ Rule R_{1b}: Else predict Class 2.
  - ❑ Rule R_2: If D_{28} = 0, then
    - ❖ Let Sum = $C_7 + 1.53\ C_{13} - 1.16\ C_{18}$.
    - ❖ Rule R_{2a}: If Sum ≥ -5.41, then predict Class 1,
    - ❖ Rule R_{2b}: Else predict Class 2.
  - ❑ Rule R_3: ……………………..
  - ❑ Rule R_4: If D_{28} = 1 and D_{48} = 1, then
    - ❖ Let Sum = $C_7 + 1.68\ C_{13} - 1.10\ C_{18} - 1.95\ C_{23}$.
    - ❖ Rule R_{4a}: If Sum ≥ - 9.86, then predict Class 1,
    - ❖ Rule R_{4b}: Else predict Class 2.
  - ❑ Rule R_5: Default rule, predict Class 2.

Segment 1:
- use internet most frequently but have the lowest income category
- important continuous inputs:
  - o C_7: introverted
  - o C_{13}: non assertive
  - o C_{23}: location

Experiment 4: Understanding consumer heterogeneity.

- Accuracy comparison:

Methods	Accuracy rates			
	Training set		Test set	
	Class 1	Class 1	Class 1	Class 2
Re-RX	55.20	83.37	55.56	80.79
C4.5	71.20	98.53	33.33	84.24
C4.5 rules	59.20	81.66	49.20	73.40
CART	56.00	94.13	22.22	87.68
Logistic reg	40.00	94.40	22.22	89.66

Conclusion

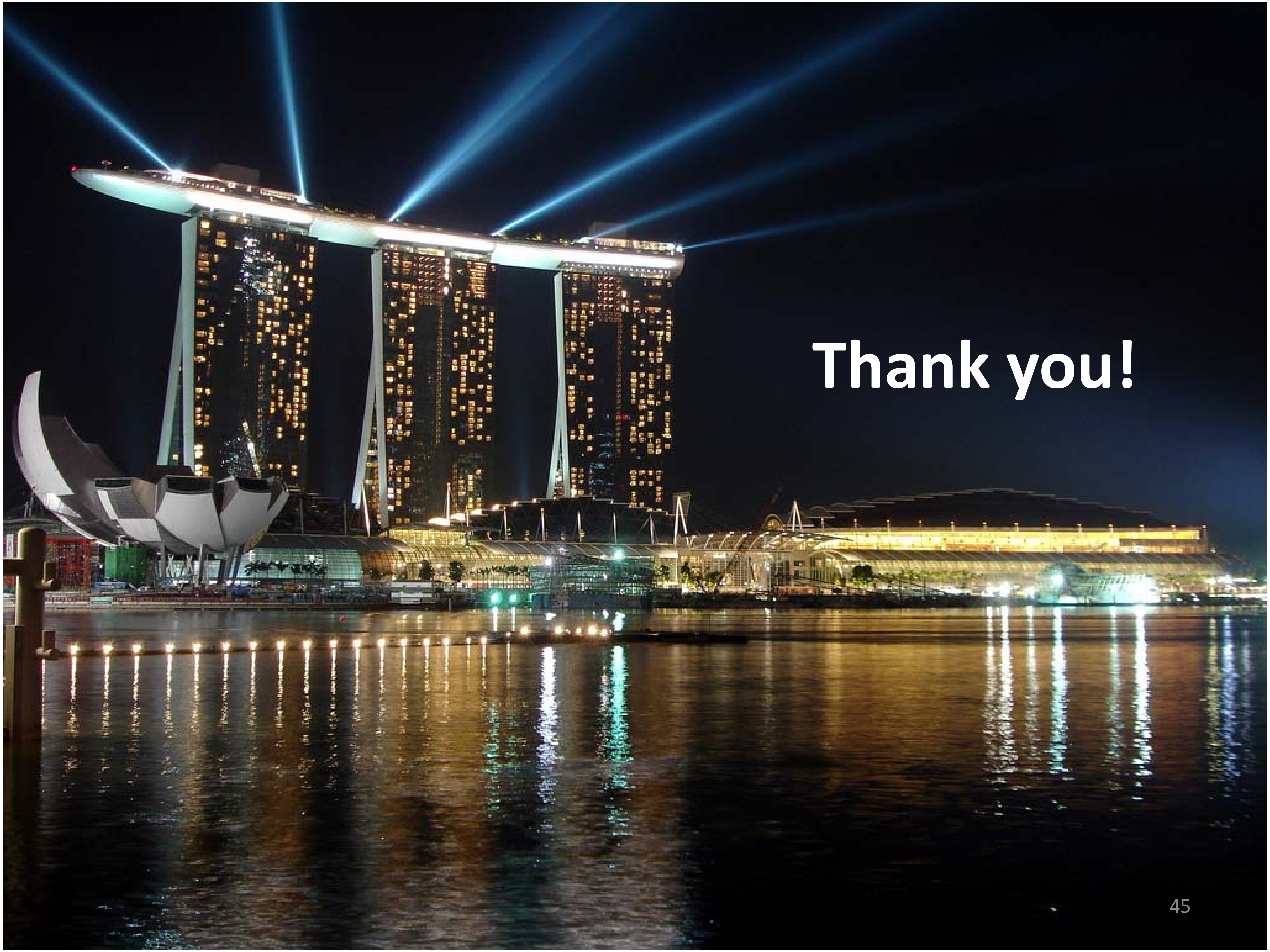
- For business intelligence applications, neural networks with as few as one hidden unit can provide good predictive accuracy.
- Pruning allows the us to extract classification rules from the networks.
- In credit scoring, two important requirements for any models are performance and interpretability.
  - **Performance**: neural networks and the rules extracted from them perform better than other methods such as decision trees and logistic regression.
  - **Interpretability**: financial regulators and law enforcement bodies require risk management models of a financial institution to be validated.

References

- **R. Setiono, B. Baesens and C. Mues.** Rule Extraction from minimal neural networks for credit card screening, forthcoming, *International Journal of Neural Systems*.
- **Y. Hayashi, M-H. Hsieh and R. Setiono.** Understanding consumer heterogeneity: A business intelligence application of neural networks, *Knowledge Based Systems,* Vol. 23, No. 8, pages 856-863, 2010.
- **R. Setiono, B. Baesens and C. Mues.** A note on knowledge discovery using neural networks and its application to credit screening, *European Journal of Operational Research,* Vol. 192, No. 1, pages 326-332, 2009.
- **R. Setiono, B. Baesens and C. Mues.** Recursive neural network rule extraction for data with mixed attributes, *IEEE Transactions on Neural Networks,* Vol. 19, No. 2, pages 299-307, 2008.

Collaborators:

- **B. Baesens**, Department of Applied Economic Sciences, Catholic University - Leuven, Belgium.
- **Y. Hayashi**, Department of Computer Science, Meiji University, Japan.
- **M-H. Hsieh**, Department of International Business, National Taiwan University, ROC.
- **C. Mues**, School of Management, Southampton University, United Kingdom.


Thank you!

Time-series Data Mining using NN-RE

Time-series prediction (Case 1):

- prediction of the next value (or future values) in the series:

$$y_{t+1} = f(y_t, y_{t-1}, y_{t-2}, \ldots\ldots y_{t-n}) \text{ or}$$

$$y_{t+1} = f(y_t, y_{t-1}, y_{t-2}, \ldots\ldots y_{t-n}, \mathbf{x})$$

where

y_t is the value of the time-series at time t

x is a set of other input variables, e.g. economic indicator

Time-series prediction (Case 2):

- prediction of direction of the time series, i.e. if the next value in the series will be higher or lower than the current value:

$$y_{t+1} = f(y_t, y_{t-1}, y_{t-2}, \ldots.. y_{t-n})$$

if ($y_{t+1} > y_t$) then Class = 1

else Class = 0

- This is a binary classification problem

- While NN can be used for regression or classification, it is easier to extract the rules from classification neural networks.

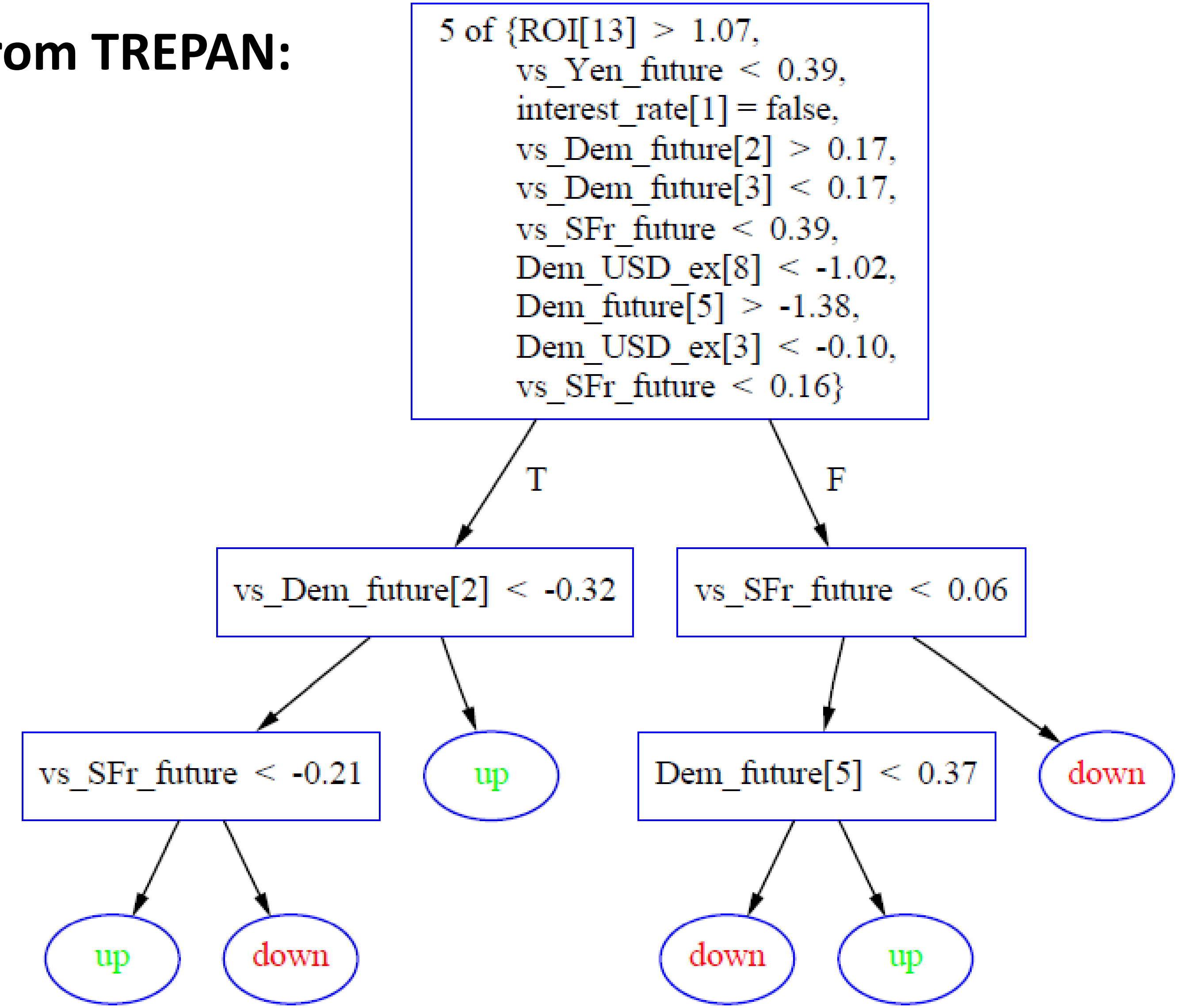
Time-series Data Mining using NN-RE

Example.

- Prediction of US Dollar versus Deutsche Mark by Craven and Shavlik (International Journal of Neural Systems, Vol. 8, No. 4, August 1997, 373-384.
- Number of input attributes: 69.
- 12 inputs represent information from the time-series, e.g. relative strength index, skewness, point and figure chart indicators.
- 57 inputs represent fundamental information beyond the series, e.g. indicators dependent on exchange rates between different countries, interest rates, stock indices, currency futures, etc.
- The data consist of daily exchange rates from January 15, 1985 to January 27, 1994.
    - last 216 days data used as test samples
    - 1607 training samples and 535 validation samples (every fourth day)

Time-series Data Mining using NN-RE

Rules from TREPAN:



Time-series Data Mining using NN-RE

Accuracy

Method	Accuracy (%)
Naïve rule	52.8
C4.5	52.8
C4.5 (selected)	54.6
ID2-of-3+	59.3
ID2-of-3+ (selected)	57.4
TREPAN	60.6
Trained NN	61.6

Tree complexity

Method	# Internal nodes	# feature references
C4.5	103	103
C4.5 (selected)	53	53
ID2-of-3+	78	303
ID2-of-3+ (selected)	103	358
TREPAN	5	14