Learning with Regularization Networks Mixtures

Petra Vidnerová

Department of Theoretical Computer Science
Institute of Computer Science
Academy of Sciences of the Czech Republic

ITAT 2007

Outline

Introduction

Supervised learning and Regularization Networks

Mixtures of Regularization Networks

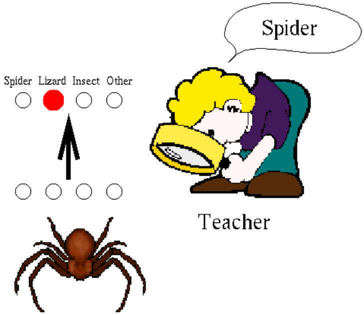
Experimental Results

Summary and Future Work

Supervised Learning

Learning

- given set of data samples
- find underlying trend, description of data



Supervised Learning

- data – input-output patterns
- create model representing IO mapping
- classification, regression, prediction, etc.

Regularization Networks

Regularization Networks

- method for supervised learning
- a family of feed-forward neural networks with one hidden layer
- derived from regularization theory
- very good theoretical background

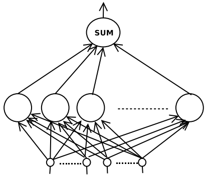
Regularization Approach

- focus on generalization ability
- optimise simultaneously data term and regularization term
  $H[f] = \sum_{i=1}^{N} (f(\vec{x}_i) - y_i)^2 + \gamma \Phi[f]$

Regularization Network

Network Architecture

$$f(x) = \sum_{i=1}^{N} w_i K(\vec{x}, \vec{x}_i)$$



- function K called basis or kernel function

Kernel function

- positive-definite function (such as Gaussian)
- choice of K represents our knowledge or assumption about the problem
- choice of K is crucial for the performance of the network

RN learning

Basic Algorithm

1. set the centers of kernel functions to the data points
2. compute the output weights by solving linear system

$$(\gamma I + K)\vec{w} = \vec{y}$$

Metaparameters

- kernel function and regularization parameter by cross-validation or based on our knowledge

Other algorithms

- generalized regularization networks
- learning based on gradient optimisation, evolution, etc.

Motivation

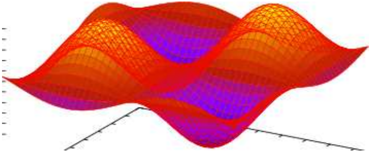
RN – optimistic point of view

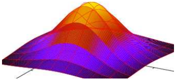
- if kernel function is chosen, tasks of learning reduces to well-posed linear system
- well studied and optimised linear optimisation algorithms available
- if optimal kernel function, very good results (approximation, generalization)

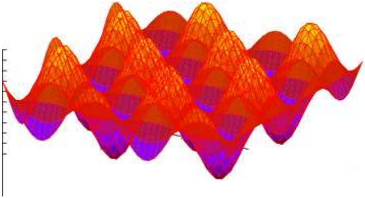
RN – pessimistic point of view

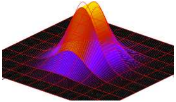
- choice of kernel function depends on particular task
- the size of linear system depends on the size of data set
- for large data sets too time consuming

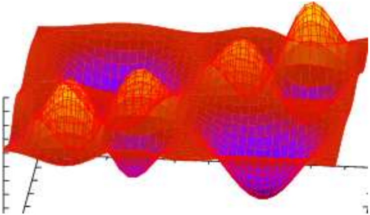
Example







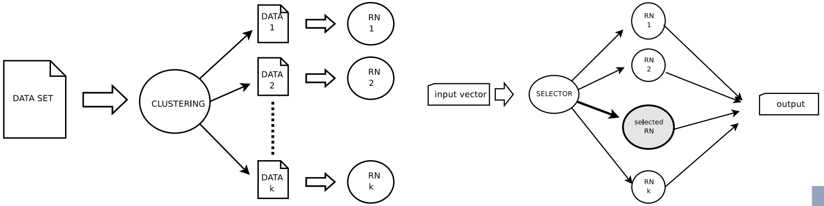




?

Mixtures of Regularization Networks

- divide data
- clustering – k-means clustering (SOMs, etc.)
- learn one RN on each subset
- subsets may overlap
- RN parameters optimised for the subset
- linear system replaced by k smaller ones



Experiments

Data

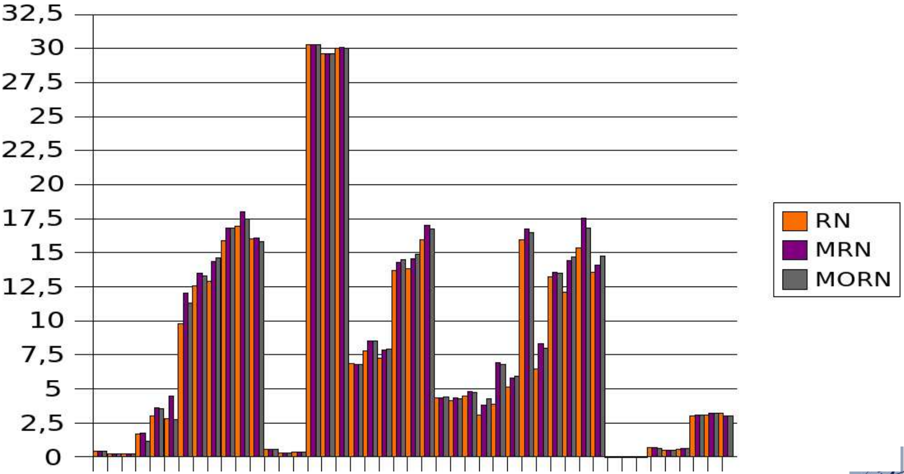
- benchmark data sets – Proben1 data repository
- separate data for training and testing
- 3 variants for each tasks

Methodology

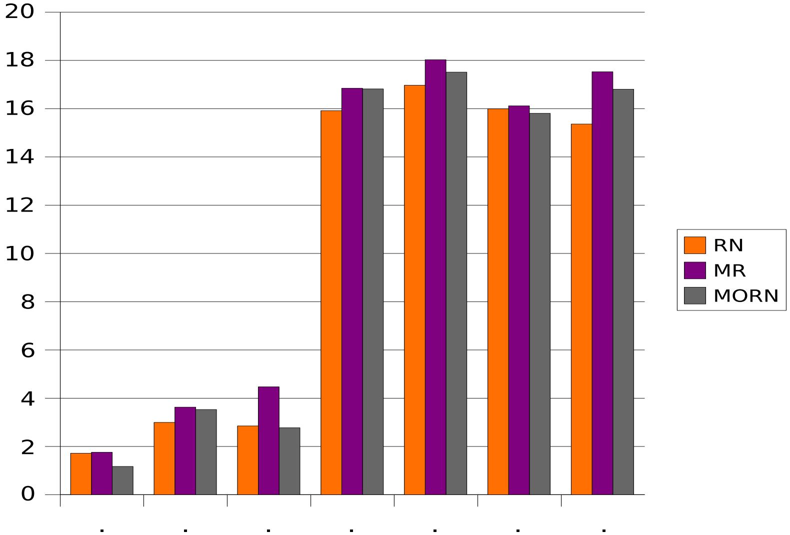
- γ and width optimised by cross-validation
- learn on training set (estimation of weights w)
- evaluate error on testing set (generalization ability)
- clustering phase – *k-means* clustering
- linear optimisation – LAPACK

Comparison of RN and mixtures of RN

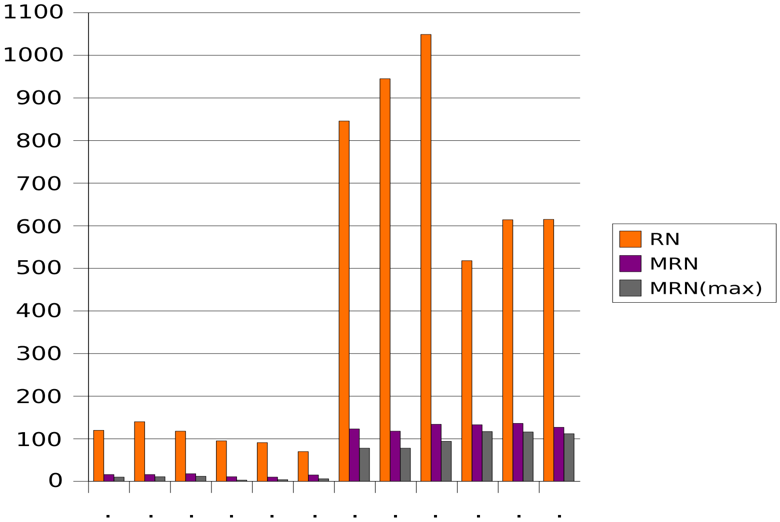
Error function



Comparison of RN and mixtures of RN



Time comparison



Conclusion

Summary

- sometimes it is useful to divide the task into several subtask
- significant decrease of time requirements
- preserving approximation and generalization ability

Future work

- how it influence the number of hidden units?
- application of RNs and other kernel based methods in mobile robots

Thank you. Questions?