Radial Basis Functions and Neural Networks

Desmond Blue Manthy

Overview

- Motivation
- Overview of Radial Basis Functions (RBFs) and Neural Networks (NNs)
- RBF neural network techniques
- Comparison of RBF networks to other techniques
- Conclusion and direction for further development

Abstract

- Neural networks have universal approximation capabilities. They can be used as solutions for given differential equations that define unsupervised errors
- Presents a wide survey and classification of different Multilayer Perceptron (MLP) and Radial Basis Function neural network (RBFN) techniques
- The focus is on the crux of various research articles published by numerous researchers.

[0] Manoj Kumar, Neha Yadav* [1]

Section I: Introduction

- Methods such as Runge–Kutta methods, Predictor–Corrector methods, Finite Difference methods, Finite Element methods, Splines and more require the discretization of domain via meshing
  - May be challenging in two or higher dimension problems
  - Approximate solution derivatives are discontinuous and can impact the stability of the solution
  - To obtain satisfactory solution accuracy, it may be necessary to deal with finite meshes that significantly increase the computational cost

[0] Manoj Kumar, Neha Yadav* [1]

Section I: Introduction Cont

- Approximate particular solutions can also be achieved by using multilayer perceptrons, RBFs, genetic programming, hybrid approaches based on neural networks, etc.
  - These methods involve a single independent variable regardless of the dimension of the problem
  - the solution obtained from the neural network methods are differentiable and in closed analytic form
- This article focuses on a wide survey of solution of differential equations using multilayer perceptrons and radial basis function networks

[0] Manoj Kumar, Neha Yadav* [1]

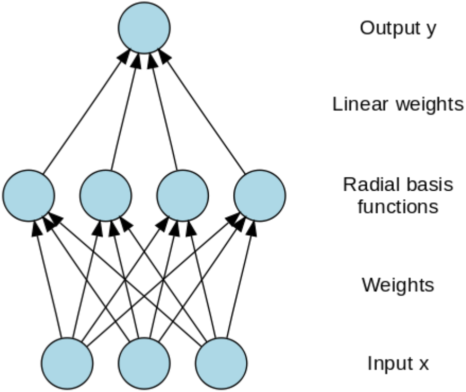
Overview of RBFs and Neural Networks

There are two distinct ways for RBFs and neural networks (NN) to be used together:

- Radial Basis Function Networks (RBFNs)
- RBF and Neural Network Hybrid Methods

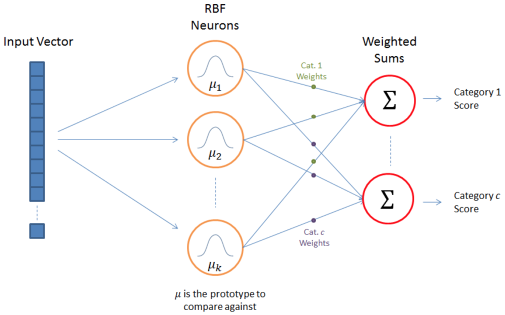
Radial Basis Function Networks (RBFNs)

RBFs are used within the neural network to provide more flexibility on the neuron level. Specifically, RBFs are used as activation functions.



RBFN as a non-linear classifier

- An RBFN performs classification by measuring the input's similarity to examples from a training set
- Each RBFN neuron stores a "prototype" which is one of the examples from the training set.
- When classifying a new input, each neuron computes the Euclidean distance between the input and its prototype



[0] Chris McCormick [2]

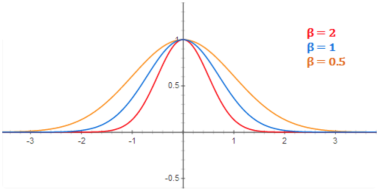
RBFN Activation Function

- There are numerous similarity functions, a common choice is based on the Gaussian.
- Equation for a Gaussian with a one-dimensional input

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- When used as an activation function remove $1/(\sigma * \sqrt{(}2 * pi)$ & set β. The height is controlled by the output node weights

$$\varphi(x) = e^{-\beta\|x-\mu\|^2}$$



[0] Chris McCormick [2]

Training RBFNs

Training consists of selecting sets of parameters

1. The prototypes
   - These can be selected using k-means clustering
2. The Beta coefficient of each neuron
   - One method for specifying the beta coefficients is to set sigma equal to the average distance between all points in the cluster

$$\sigma = \frac{1}{m}\sum_{i=1}^{m} \|x_i - \mu\|$$

   - then

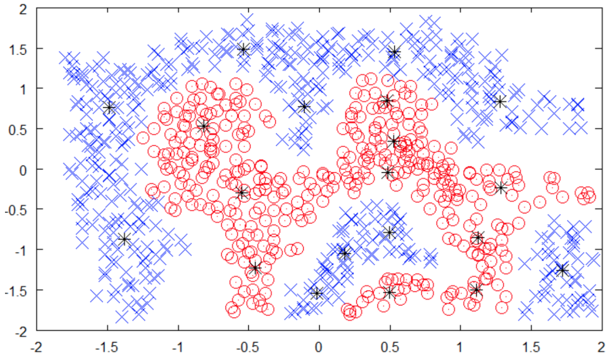
$$\beta = \frac{1}{2\sigma^2}$$

3. The output weights (between RBF nodes and outputs nodes)
   - These can be found using gradient decent

[0] Chris McCormick [2]

RBFNs in Practice

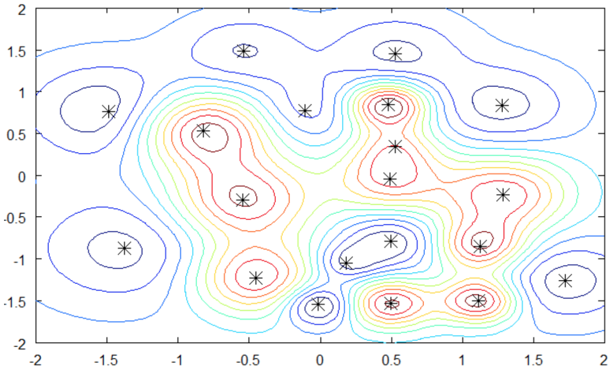
- An example of a trained RBFN with twenty neurons
- Two classes of data points
- Selected prototypes are marked by black asterisks



---
0 Chris McCormick [2]

RBFNs in Practice cont

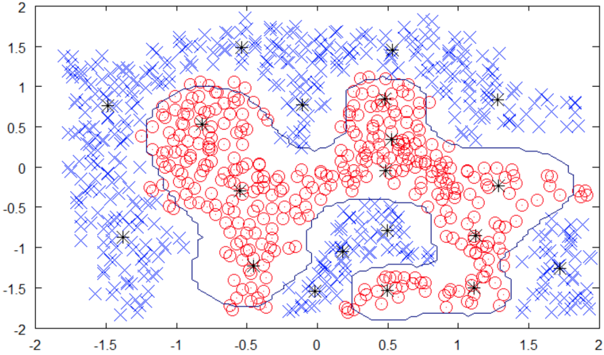
- Can visualize the red circle score over the input space with a 3D mesh
- Hills in the output values are centered around the prototypes



[0] Chris McCormick [2]

RBFNs in Practice cont[2]

- An approximation of the decision boundary



[0]Chris McCormick [2]

RBF and Neural Network Hybrid Methods

- Generally used for large applications
- RBF-FD are used for niche aspects of the problem such as solving PDEs
- NN are used on separate parts of the problem that are not well described by PDEs or other 'nice' mathematical formulas
- These NNs can still utilize RBFs internally

[0] Fornberg

RBF and NN Hybrid Methods in PRactice

- An application of RBF and NN Hybrid Methods is weather forcasting
- PDEs from fluid dynamics are great at describing the large scale motion of the atmosphere
- NNs can better handle the motion of clouds are not well described by these PDEs

Saba, Tanzila, et al. "Weather Forecasting Based on Hybrid Neural Model." Applied Water Science, vol. 7, no. 7, 2017, pp. 3869–3874., https://link.springer.com/article/10.1007/s13201-017-0538-0

[0] Fornberg

RBF Neural Network Techniques

Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey

by Manoj Kumar, Neha Yadav*

Neural networks and radial basis functions in classifying speech patterns

by Mahesan Niranjan and Frank Fallside

Classifying static speech patterns

- The paper compares the performances of three non-linear pattern classifiers in the recognition of speech patterns.
- The challenge is speech is highly non-linear and in constant motion. Thus the signal is non-linear as well as non-stationary
- 1. Create a mathematical model to capture the instantaneous shape. Under the assumption that the signal is quasi-stationary and can be modelled as the output of a linear filter
  2. Model the transitions from one phonetic segment to the other as a Markov process
  3. The third stage is pattern classification

[0] Mahesan Niranjan and Frank Fallside [3]

RBFs for Non-linear pattern classification

- Create a network using hidden nodes defining hyperellipsoids (rather than hyperplanes) to offer superior pattern classification
- Use many more hidden nodes with fewer degrees of freedom per node to improve training time (avoids pitfall of conventional back-propagation method)

[0] S Renals [4]

RBFN

- RBF hidden nodes computing their output as follows:

$$y^h_{pi} = \Phi(||x_p - c_i||)$$

- where $y^h_{p_i}$ is the output of hidden node i in response to the pth input vector x_p
- c_i is a vector of similar dimension, representing the center of a radially symmetric function Φ

[0] S Renals [4]

RBFN Cont.

- Φ is tipically chosen to be a Gaussian

$$\Phi(||x_p - c_i||) = exp\{-\sum_j [\frac{(x_j - c_{ij})^2}{2\sigma_{ij}}]\}$$

- Where σ_{ij}^2 is the ijth element of a covariance matrix

[0] S Renals [4]

RBFN Cont.[2]

- "It is hypothesised that expansion of input vectors into a hidden unit space of relatively high dimension (i.e. by defining many radial basis functions) will result in a greater likelihood of the classification problem becoming linearly separable"
- In this case the output of the network may be defined as follows

$$y_{pi}^{T} = \omega_{i0} + \sum_{j} \omega_{ij} \Phi(||x_p - c_j||)$$

- y_{pi}^{T} is the output of the ith target node in response to the pth pattern
- ω_{ij} is the weight from the jth radial basis function to the ith target node
- ω_{i0} is the threshold of the ith target node
- Specifies that the outputs computed by the target units are linear in the weights

[0] S Renals [4]

RBFN Training

- The RBFN is trained by minimizing the least squares error

$$E = 0.5 \sum_p \sum_i (Y_{pi}^T - y_{pi}^T)^2$$

- Y_{pi}^T is the desired output of target node i in response to the pth input vector

---

[0] S Renals [4]

RDFN Training Cont.

- The hand-segmented data consisted of approximately 750 vowel tokens (of 20 classes) extracted from the 98 sentences
- The neural network model consisted of three layers: 72 inputs, 20 targets (representing the 20 vowel classes), and a variable number of radial basis functions
- The centers of the radial basis functions were chosen by randomly choosing input vectors, in accordance with the sample distribution
- Each network configuration was trained and tested 12 times and the mean recognition score (%) and standard deviations of training and testing classification scores were computed over the 12 experiments

---

[0] S Renals [4]

Training Results

Table 1 CLASSIFICATION SCORES FOR THREE CLASSIFIERS TRAINED ON 758 VOWEL TOKENS AND TESTED ON 759 (UNSEEN) VOWEL TOKENS

		Training set		Test set	
Classifier	*N*(*hid*)	Mean	SD	Mean	SD
		%	%	%	%
RBF	64	74·0	0·94	65·3	0·86
RBF	100	78·5	1·12	68·4	1·31
RBF	144	84·0	0·84	70·4	1·44
RBF	170	85·8	0·48	71·2	0·60
RBF	196	87·1	1·83	71·5	2·13
RBF	256	90·6	1·39	73·3	1·53
BP	36	98·0	0·30	71·6	0·44
HMM	—	92·1	—	69·4	—

RBF indicates radial basis functions neural network model, BP a back-propagation neural network model and HMM a vector-quantised hidden Markov model. Statistics from RBF classifier were computed from 12 sets of randomly chosen centres, those from BP classifier from 5 random initial states. *N*(*hid*) refers to number of hidden units or radial basis functions in neural network model

- Radial Basis Function Neural Network, back-propagation neural network model, and a vector-quantised hidden Markov model.

[0] S Renals [4]

Results

- Comparable recognition accuracy
- The network with 144 radial basis functions could be trained in less than 4 minutes
- The hidden Markov model took approximately 3 hours to train
- Training the back-propagation neural network model took approximately 3 hours (on a pipelined array processor, which gives a 5-10 times speed increase compared with the serial hardware used for the other methods)
- On similar hardware, training a 36-node back-propagation network is over two orders of magnitude slower than training a radial basis functions network with 144 RBFs

[0] S Renals [4]

Reference

- [1] Kumar, Manoj, and Neha Yadav*. "Multilayer Perceptrons and Radial Basis Function Neural Network Methods for the Solution of Differential Equations: A Survey." Computers and Mathematics with Applications - Journal - ELSEVIER, no. 62, 2011, pp. 3796–3811. https://core.ac.uk/download/pdf/82320727.pdf
- [2] McCormick, Chris. "Radial Basis Function Network (RBFN)." Radial Basis Function Network (RBFN) · Chris McCormick, 15 Aug. 2013, mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/.
- [3] Niranjan, Mahesan and Fallside, Frank. "Neural networks and radial basis functions in classifying static speech patterns" Computer Speech & Language-Journal-ELSEVIER, Vol. 4, no. 3, 1990, pp. 275-289. www.sciencedirect.com/science/article/pii/088523089090009U
- [4] Renals, S. RADIAL BASIS FUNCTION NETWORK FOR SPEECH PAITERN CLASSIFICATION. Department of Physics and center for Speech Technology Research University of Edinburgh 80 South Bridge, Edinburgh EH1 LHN, United Kingdom, 12 Dec. 1988, era.ed.ac.uk/bitstream/handle/1842/1205/Renals%20Elect.Lett..pdf