Notes from bibliography

Roman Dušek

February 2, 2021

Contents

1 Fundamental of Higher Order Neural Networks for Modeling and Simulation (Gupta and Bukovsky 2012)

1.1 Introduction

Biological neuron
1. Synaptic operation - strength (weight) is represented by previous knowledge.

2. Somatic operation

- aggregation (summing), thresholding, nonlinear activation and dynamic processing
- output after certain threshold

if neuron was only linear the complex cognition would disappear

First neuron modeled (1943)

$$u = \sum_{i=1}^{n} w_i x_i$$

1.1.1 Higher Order Terms of Neural Inputs

year 1986, 1987, 1991, 1992, 1993

$$u = \sum_{j=i}^{n} \sum_{i=1}^{n} w_{ij} x_i x_j$$

1.1.2 Activation functions

1.1.2.1 Sigmoid

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

1.2 SONU/QNU

- parameter reduction using upper triangular matrix of weights

$$u = \mathbf{x}_a^T \mathbf{W}_a \mathbf{x}_a = \sum_{j=i}^{n} \sum_{i=1}^{n} w_{ij} x_i x_j$$

$$y = \phi(u)$$

if a weight is high it shows correlation between components of input patterns

Learning algorithm for second order neural units

The purpose of the neural units is to minimize the error E by adapting the weight

$$E(k) = \frac{1}{2} e(k)^2 \quad ; \quad e(k) = y(k) - y_d(k)$$

$$\mathbf{W}_a(k+1) = \mathbf{W}_a(k) + \Delta \mathbf{W}_a(k)$$

$$\Delta \mathbf{W}_a(k) = -\eta \frac{\delta E(k)}{\delta \mathbf{W}_a(k)}$$

where η is learning coefficient chain rule ...

using chain rule we get changes in the weight matrix as

$$\Delta \mathbf{W}_a(k) = -\eta e(k)\phi'(u(k))\mathbf{x}_a(k)\mathbf{x}_a^T(k)$$

Table with mathemathical structure and learning rule

SONU	Math. Struct	Learning rule
Static	$y_n = \mathbf{x_a^T W x}$	Levenberg_marquard (L-M)
		Gradient descent
Dynamic	$y_n(k+n_s) = \mathbf{x_a^T W x}$	Recurrent Gradient Descent
		Backpropagation throughtime

1.2.1 Performance Assesment of SONU

1.2.1.1 XOR problem

- XOR 6params vs 9 of 3 linear units

1.3 Time Series Prediction

1.4 High order neural network units

- HONU is just a basic building block

1.4.1 Example of Cubic neural network with two inputs

1.5 Modified PNN

1.5.1 Sigma-Pi NN

1.5.2 Ridge PNN

1.6 Conclusion

- this neural network first aggregates inputs and then multiplicate

2 Nonconventional Neural Architectures and their Advantages for Technical Applications (Bukovský 2012)

2.1 Introduction

- first mathematical model of neuron 1943
- principals for modeling of dynamic systems
  - customable non-linearity
  - order of dynamics of state space representation of a neuron
  - adaptable time delays

2.1.1 HONU, HONN

- PNN - polynomial neural networks
- LNU, QNU, CNU
- linear optimization, avoidance of local minima

bio-inspired neuron, *perceptron*, *recurrent* (dynamic, hopfield)

static vs dynamic
continuous vs discrete implementation of static/dynamic HONN

2.1.2 Gradient optimization methods

- back propagation
- gradient descent rule
- Levenberg-Marquardt algorithm

2.2 RHONN

- table page 14

RTRL – real time recurrent learning

- dynamic version of gradient descent

BPTT – back propagation throught time

- batch training technique can be implemented as combination of RTRL and L-M algorithm => RHONU

2.3 Weight update stability of static and dynamic version

3 Artificial High Order NN for Economics and Bussiness (Zhang 2008)

3.1 Chapter 1

- use case of HONN
- model 1, 1b, 0
  - model 1 is containing one hidden layer with linear units
  - model 1b is containing two

Polynomial HONN uses poly-func as activation function

Neural Adaptive HONN uses adaptive functions as neuron

3.1.1 Learning algorithm of HONN

3.1.2 PHONN

3.1.3 Trigonometric HONN

- uses trigonometric functions as activation functions

3.1.4 Ultra high frequency cosine and sine HONN

3.1.5 SINC and Sine Polynomial HONN

3.2 Chapter 3

- HONN first introduced in (Giles, Maxwell 1987)
- hyperbolic tangent function

$$S(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

3.3 Chapter 5

- info about "High order Flexibel Neural Tree"

Chapter 6 - most basic motivation of stock forecasting is financial gain - motivation behind recurrent is that patterns may repeat in time

3.3.1 Background HONN

3.3.2 HONN structure

3.4 Chapter 7

Problems of ANNs - long convergence time - can be stuck in local minima - unable to handle high-frequency, non-linear and discontinuous data - black box

HONN can be considered as "open box"

3.5 Chapter 8

3.5.1 Introduction

- NN are data-driven we dont need prior assumptions
- NN can generalise
- they are universal approximators

Well know problems with NNs - different results when tested on same datasets - size sensitive, they suffer of over fitting

Offline network - goal of minimizing error over whole dataset
Online network - aim is to adapt to local properties of the observed signal. They create detailed mapping of the underlying structure within the data

3.5.2 Overview of NN

- using nonlinear transfer function they can carry out non-linear mappings
- history of milestone

3.5.2.1 Neuron structure

Activation Functions - threshold, linear, logistic sigmoid function

3.5.3 Network structures

Feed-forward -single layer (perceptron, ADALINE), multi layer **RNN** - using feedback loop

```
**Fully recurrent**
- all connections are trainable\

**Partial recurrent**
- only feed-forward units are trainable, feedback utilize by *context unit*
```

- feedback results in nonlinear behavior, that provides networks with capabilities to storage information.

3.5.3.1 Lerning RNN

Backpropagation through time - main idea is to unfold the RNN into an equivalent feed-forward network **Real Time recurrent learning** - each synaptic weight is updated for each representation of training set → no need to allocate memory proportional to the number of sequences

3.5.4 HONN

(Giles & Maxwell, 1987) - functional link network (Pao, 1989)

Popular multi layer HONN

Sigma-pi NN (Rumelhar, Hinto & William, 1986) - summing of inputs and product units (order is determined by number of inputs)

Pi-Sigma NN (Shin & Ghosh, 1992) - summing of inputs and one product unit (fewer number of weights)

Ridge polynomial neural network (Shin & Ghost, 1991) - using increasing number of pi-sigma units

3.5.4.1 High order interactions in Biological Networks

3.5.5 Pipelined RNNs

(Haykin & Li, 1995) - engineering principle - "divide and conquer" - consist of two subsections - linear and nonlinear

3.5.5.1 RTRL for PRNN

3.5.6 Second order PRNN

3.6 Chapter 9

3.6.1 Second order Single layer RNN

3.7 Chapter 10

- The lowest error of multilayer network occurs for one trained with Levenberg-Marquardt
- Multilayer networks have higher error

4 Adaptive control with RHONN (Rovithakis and Christodoulou 2000)

4.1 Introduction

- for training model uses current and previous inputs, as well as the previous outputs
- in discrete outputs we need to discretized the model
- model is trained to identify the inverse dynamics of the plant instead of the forward dynamics
- by connecting the past neural output as input we make dynamic network that is highly nonlinear

Problem with dynamic neural networks that are based on static multilayer networks is that criterial functions possesses many local minima.

4.1.1 Book goals

Chapter 2 introduces RHONN
Chapter 3 online identification

4.2 Identification using RHONN

4.2.1 Model description

$$\dot{x}_i = -a_i x_i + b_i \left[\sum_{k=1}^{L} w_{ik} z_k \right]$$

4.2.2 Learning algorithms

5 Discrete-time HONN (Sanchez, Alanís, and Loukianov 2008)

Bibliography

Bianchi, Filippo Maria, Robert Jenssen, Michael C. Kampffmeyer, Enrico Maiorino, and Antonello Rizzi. 2017. *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*. 1st ed. 2017. SpringerBriefs in Computer Science. Cham: Springer International Publishing : Imprint: Springer. https://doi.org/10.1007/978-3-319-70338-1.

Bukovský, Ivo. 2012. "Nonconventional Neural Architectures and their Advantages for Technical Applications." Habilitation thesis, Faculty of Mechanical Engineering, Czech Technical University in Prague. http://users.fs.cvut.cz/ivo.bukovsky/publications/Teze_IB_86_bw_1200dpi.pdf.

Desker, L. R., and L. C. Jain. n.d. *Recurrent Neural Netowrks Design and Applications*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. n.d. *Deep Learning*.

Gupta, M. Madan, and Ivo Bukovsky. 2012. "Fundamentals of Higher Order Neural Networks for Modeling and Simulation." In *Fundamentals of Higher Order Neural Networks for Modeling and Simulation*. IGI Global.

Kosmatopoulos, E. B., M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou. 1995. "High-Order Neural Network Structures for Identification of Dynamical Systems." *IEEE Transactions on Neural Networks* 6 (2): 422–31. https://doi.org/10.1109/72.363477.

LazyProgrammer. n.d. *Recurrent Neural Networks in Python*.

Mandic, Danilo P., and Jonathon A. Chambers. 2001. *Recurrent Neural Networks for Prediction*. Wiley Series in Adaptive and Learning Systems for Signal Processing, Communications, and Control. Chichester, UK: John Wiley & Sons, Ltd. https://doi.org/10.1002/047084535X.

Mandic, Danilo, and Jonathon A. Chambers. n.d. "Recurrent Neural Networks for Prediction Learning Algorithms, Architectures, and Stability by Danilo Mandic, Jonathon Chambers."

Olah, Christopher. 2015. "Understanding LSTM Networks." August 27, 2015. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Rios, Jorge D., Alma Y. Alanis, Nancy Arana-Daniel, Carlos Lopez-Franco, and Edgar N. Sanchez. 2019. *Neural Networks Modeling and Control: Applications for Unknown Nonlinear Delayed Systems in Discrete Time*. 1st ed. Waltham: Elsevier.

Rovithakis, George A., and Manolis A. Christodoulou. 2000. *Adaptive Control with Recurrent High-Order Neural Networks*. Advances in Industrial Control. London: Springer London. https://doi.org/10.1007/978-1-4471-0785-9.

Sanchez, Edgar N., Alma Y. Alanís, and Alexander G. Loukianov. 2008. *Discrete-Time High Order Neural Control: Trained with Kaiman Filtering*. Vol. 112. Studies in Computational Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-78289-6.

Shin, Y., and J. Ghosh. 1991. "The Pi-Sigma Network: An Efficient Higher-Order Neural Network for Pattern Classification and Function Approximation." In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, i:13–18 vol.1. https://doi.org/10.1109/IJCNN.1991.155142.

Taylor, J. G., and S. Coombes. 1993. "Learning Higher Order Correlations." *Neural Networks* 6 (3): 423–27. https://doi.org/10.1016/0893-6080(93)90009-L.

Zhang, Ming. 2008. *Artificial Higher Order Neural Networks for Economics and Business*. IGI Global.