Notes from bibliography

Roman Dušek

February 1, 2021

Contents

1 Fundamental of Higher Order Neural Networks for Modeling and Simulation (Gupta and Bukovsky, n.d.)

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 algortihm 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, n.d.)

3 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

3.1 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

3.2 Gradient optimization methods

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

3.3 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

3.3.1 Weighty update stability of static and dynamic version

4 Artificial High Order NN for Economics and Bussiness (Zhang, n.d.)

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. n.d. "Nonconventional Neural Architectures and their Advantages for Technical Applications." In. Vol. 2012. Faculty of Mechanical Engineering, Czech Technical University in Prague.

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. n.d. "Fundamentals of Higher Order Neural Networks for Modeling and Simulation." In *Fundamentals of Higher Order Neural Networks for Modeling and Simulation*. Vol. 2012. IGI Global.

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

Mandic, Danilo P., and Jonathon A. Chambers. 2001. *Recurrent Neural Networks for Prediction*. Edited by Simon Haykin. 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."

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*. First. Waltham: Elsevier.

Zhang, Ming. n.d. *Artificial Higher Order Neural Networks for Economics and Business*.