Department of Computer Engineering
IC@Network Lab

A Robust Online Sequential Extreme Learning Machine

Hoang Trong Minh Tuan, Hieu T. Huynh,
Nguyen H. Vo, Yonggwan Won*

International Symposium on Neural Network (ISNN 2007)

Feed-Forward Neural Networks

Chapter 1
Introduction

- Review FFNN:
  - A powerful method used in various applications (regression, classification).
  - Slow training speed (gradient-descent/iterative based approaches)
  - Overfitting
  - Local minima
  - Some parameters need to be tuned manually.
  - (In theory) single hidden layer can express arbitrary boundary

Feed-Forward Neural Networks

- In many application fields: single hidden layer neural network is enough to have a good classification boundary.
  - Given any bounded nonconstant piecewise continuous activation function g, for any target function f and any randomly generated parameter sequence $\{a_L, b_L\}$ and $\tilde{N}$ is the number of hidden nodes.

$$\lim_{\tilde{N}\to\infty} \left\| f(\mathbf{x}) - f_{\tilde{N}}(\mathbf{x}) \right\| = 0$$

- Huang et al. proposed extreme learning machine (ELM)
  - ELM is a single-hidden-layer feed-forward neural network

Extreme Learning Machine (ELM)

Chapter 1
Introduction

- ELM approach is based on finding the estimation for the linear system forming the hidden-to-output part of the system.

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \quad .$$

- ELM is a batch learning method

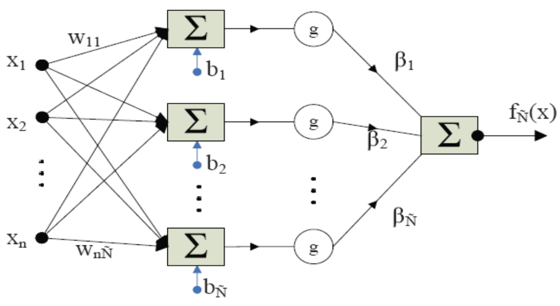


Fig. 1. A sample SLFN with n-Ñ-m network structure

Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab

Extreme Learning Machine (ELM)

Chapter 1

Introduction

- ELM algorithm:
  - Assign input weights w_{ij} and hidden nodes bias with random values.
  - Assume N > Ñ:

$$\underset{N\times\tilde{N}}{\mathbf{H}} = \begin{bmatrix} g(\mathbf{w}_1.\mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1.\mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix} .$$

$$\underset{\tilde{N}\times m}{\boldsymbol{\beta}} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}}^T \end{bmatrix} \text{ and } \underset{N\times m}{\mathbf{T}} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix} .$$

  - Find output weight:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \quad . \qquad \hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}.\mathbf{T} \quad .$$

Online Sequential Extreme Learning Machine (OS-ELM)

Chapter 2

Motivation

- ELM is a batch learning in nature.
- Obstacle in real world application:
  - When data set is large (computational cost with hidden-to-output matrix)
  - When the data is costly/hard to collect
- OS-ELM propose the solution for these difficulties
  - Learn one-by-one
  - Learn chunk-by-chunk (same or different chunk sizes)

Online Sequential Extreme Learning Machine (OS-ELM)

Chapter 2

Motivation

- OS-ELM algorithm:

Step 1 (*Boosting phase*). Given a chunk of initial training set $\aleph_0 = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N_0}$, $\mathrm{N}_0 \times \tilde{\mathrm{N}}$.

a) Assign the input weight and bias randomly within the range [-1, 1].
b) Calculate the initial hidden layer output matrix $\mathbf{H}_0$.

$$\mathbf{H}_0 = \begin{bmatrix} g(\mathbf{w}_1.\mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1.\mathbf{x}_{N_0} + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_{N_0} + b_{\tilde{N}}) \end{bmatrix}.$$

c) Estimate the initial output weight $\boldsymbol{\beta}^{(0)} = \mathbf{P}_0 \mathbf{H}_0^T \mathbf{T}_0$, where $\mathbf{P}_0 = \left(\mathbf{H}_0^T \mathbf{H}_0\right)^{-1}$ and $\mathbf{T}_0 = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_0}]^T$
d) Set the index for data chunk k to zero ($k = 0$).

Online Sequential Extreme Learning Machine (OS-ELM)

Chapter 2

Motivation

Step 2 (*Sequential learning phase*). For each further $(k+1)$-th chunk of new observations

$$\aleph_{k+1} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=(\sum_{j=1}^{k} N_j)+1}^{\sum_{j=1}^{k+1} N_j}.$$

where $\mathrm{N}_{(k+1)}$ denotes the number of samples in the $(k+1)$-th chunk.
a) Calculate the partial hidden layer output matrix $\mathbf{H}_{k+1}$ for the $(k+1)$-th chunk, as shown below

$$\underset{N_{k+1} \times \tilde{N}}{\mathbf{H}_{k+1}} = \begin{bmatrix} g(\mathbf{w}_1.\mathbf{x}_{(1)} + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_{(1)} + b_1) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1.\mathbf{x}_{(N_{k+1})} + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_{(N_{k+1})} + b_1) \end{bmatrix}.$$

b) Calculate the output weight matrix $\boldsymbol{\beta}^{(k+1)}$.

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k \mathbf{H}_{k+1}^T \left(\mathbf{I} + \mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T\right)^{-1} \mathbf{H}_{k+1} \mathbf{P}_k$$
$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \mathbf{P}_{k+1} \mathbf{H}_{k+1}^T \left(\mathbf{T}_{k+1} - \mathbf{H}_{k+1} \boldsymbol{\beta}^{(k)}\right).$$

c) Set $k = k + 1$. Go to step 2)

Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab

Online Sequential Extreme Learning Machine (OS-ELM)

Chapter 2

Motivation

- Weak-points of OS-ELM:

$$\mathbf{P}_0 = \left(\mathbf{H}_0^T \mathbf{H}_0\right)^{-1}$$

  - In real applications: $\mathbf{H}^T\mathbf{H}$ tends to be either singular or ill-conditioned matrix.
  - Adjust parameter manually:
    - Satellite image + California housing: bias in the range [0.2, 4.2]
    - Image segment: bias in the range [3, 11]
    - DNA: bias in the range [20, 60]

Robust Online Sequential Extreme Learning Machine (ROS-ELM)

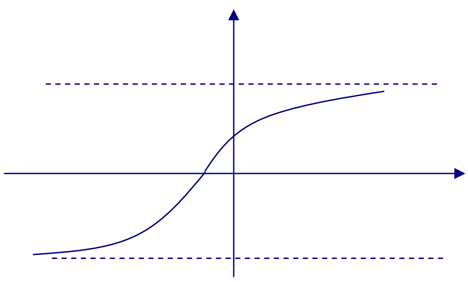
Chapter 3

ROS-ELM

- Find the condition to guarantee $\mathbf{H}^T\mathbf{H}$ is full rank
  - If $\mathbf{H}$ is full rank, then $\mathbf{H}^T\mathbf{H}$ is full rank
  - If $\mathbf{H}^T\mathbf{H}$ is full rank, then $\mathbf{H}^T\mathbf{H}$ is invertible.
- Input weight selection

$$w_{ij} = c.r_{ij} \quad .$$

where r_{ij} is a random variable of normal distribution ($\mu = 0$, $\sigma = 1$); c is a user-defined scalar that can be adjusted to obtain the input-to-hidden weights that do not saturate the sigmoid functions.



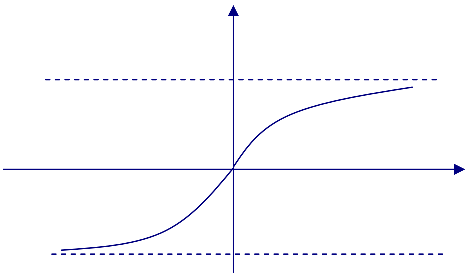
Robust Online Sequential Extreme Learning Machine (ROS-ELM)

Chapter 3

ROS-ELM

- Input bias:

$$\mathbf{b} = -diag\left(\mathbf{X}\mathbf{W}^T\right) \quad .$$



Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab

Datasets

Chapter 4

Experiments

- Dataset for benchmarks
  - Regression
  - Classification

Table 1. Specifications of benchmark data sets with number of hidden nodes for testing

Dataset	Attribute	Classes	Training data	Testing data	Nodes
Auto-MPG	7	-	320	78	25
Abalone	8	-	3,000	1,177	25
Image Segment	19	7	1,500	810	180
Satellite Image	36	6	4,435	2,000	400

Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab

Experiment process

Chapter 4
Experiments

- All features of regression application: normalized into the range [0, 1]
- Input attributes of classification applications: normalized into the range [-1, 1].
- Number of hidden nodes in each test cases: manually select the best ones
- Fifty trials for each test and we compute the average result

Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab

Regression results

Table 2. Performance comparison on Regression Applications

Chapter 4
Experiments

Datasets	Learning mode	Algorithms	RMSE	
			Training	Testing
Auto-MPG	1-by-1	OS-ELM	0.082955	0.091689
		ROS-ELM	0.082419	0.090321
	20-by-20	OS-ELM	0.083818	0.089498
		ROS-ELM	0.083442	0.083744
	20-by-30	OS-ELM	0.082900	0.090005
		ROS-ELM	0.082050	0.089840
Abalone	1-by-1	OS-ELM	0.075512	0.076820
		ROS-ELM	0.075619	0.076102
	20-by-20	OS-ELM	0.075317	0.077300
		ROS-ELM	0.075817	0.075977
	20-by-30	OS-ELM	0.075245	0.077549
		ROS-ELM	0.075547	0.076814

Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab

- Result from Liang et al.

TABLE II
COMPARISON BETWEEN OS-ELM AND OTHER SEQUENTIAL ALGORITHMS ON REGRESSION APPLICATIONS

Datasets	Algorithms	Time (seconds)	RMSE Training	RMSE Testing	# nodes
Auto-MPG	**OS-ELM (Sigmoid)**	0.0444	0.0680	0.0745	25
	OS-ELM (RBF)	0.0915	0.0696	0.0759	25
	Stochastic BP	0.0875	0.1112	0.1028	13
	GAP-RBF[18]	0.4520	0.1144	0.1404	3.12
	MRAN[18]	1.4644	0.1086	0.1376	4.46
	RANEKF[18]	1.0103	0.1088	0.1387	5.14
	RAN[18]	0.8042	0.2923	0.3080	4.44
Abalone	**OS-ELM (Sigmoid)**	0.5900	0.0754	0.0777	25
	OS-ELM (RBF)	1.2478	0.0759	0.0783	25
	Stochastic BP	0.7472	0.0996	0.0972	11
	GAP-RBF[18]	83.784	0.0963	0.0966	23.62
	MRAN[18]	1500.4	0.0836	0.0837	87.571
	RANEKF[18]	90806	0.0738	0.0794	409
	RAN[18]	105.17	0.0931	0.0978	345.58

Classification results

Table 3. Performance comparison on Classification Applications

Chapter 4
Experiments

Datasets	Learning mode	Algorithms	Accuracy Training	Accuracy Testing
Image Segment	1-by-1	OS-ELM	96.8160	94.3852
		ROS-ELM	97.2320	94.8519
	20-by-20	OS-ELM	96.8147	94.2198
		ROS-ELM	97.3067	94.9852
	20-by-30	OS-ELM	96.7440	94.2519
		ROS-ELM	96.9613	94.9111
Satellite Image	1-by-1	OS-ELM	91.9324	88.9170
		ROS-ELM	92.7806	89.8520
	20-by-20	OS-ELM	91.9436	88.9040
		ROS-ELM	92.7251	89.7690
	20-by-30	OS-ELM	91.9729	88.7860
		ROS-ELM	92.6011	89.6550

- Result from Liang et al.

TABLE III
COMPARISON BETWEEN OS-ELM AND OTHER SEQUENTIAL ALGORITHMS ON CLASSIFICATION APPLICATIONS

Datasets	Algorithms	Time (seconds)	Accuracy (%)		# nodes
			Training	Testing	
Image Segmentation	**OS-ELM (Sigmoid)**	9.9981	97.00	94.88	180
	OS-ELM (RBF)	12.197	96.65	94.53	180
	Stochastic BP	2.5776	83.71	82.55	80
	GAP-RBF	1724.3	-	89.93	44.2
	MRAN	7004.5	-	93.30	53.1
Satellite Image	**OS-ELM (Sigmoid)**	302.48	91.88	88.93	400
	OS-ELM (RBF)	319.14	93.18	89.01	400
	Stochastic BP	3.1415	85.23	83.75	25
	MRAN	2469.4	-	86.36	20.4

Conclusion

Chapter 5
Conclusion

- ROS-ELM proves it generalization and higher performance compared to OS-ELM.
- ROS-ELM is a little slower than OS-ELM due to the matrix multiplication in the initialization phase, but the overall difference is subtle.
- The combination of ELM and ROS-ELM can be an effective solution in different domains.

Thank you!

Chonnam National University - KOREA
Dept. of Computer Engineering
IC@Network Lab