

# First COMIDENCO Meeting

Extreme Learning Machine, introducción, estado  
del arte

Héctor Estigarribia    Fredy Ramírez    Cristhian Mendieta

Facultad de Ciencias y Tecnologías  
Universidad Nacional de Caaguazú

13 de julio de 2017

# Tabla de contenido

## 1 Redes neuronales

- Idea básica
- Back-propagation

## 2 Extreme Learning Machine

- Desventajas del Back-propagation
- Teoría básica
- Casos
- Resumen
- Herramientas
- Fuentes

# Tabla de contenido

## 1 Redes neuronales

- Idea básica
- Back-propagation

## 2 Extreme Learning Machine

Idea básica

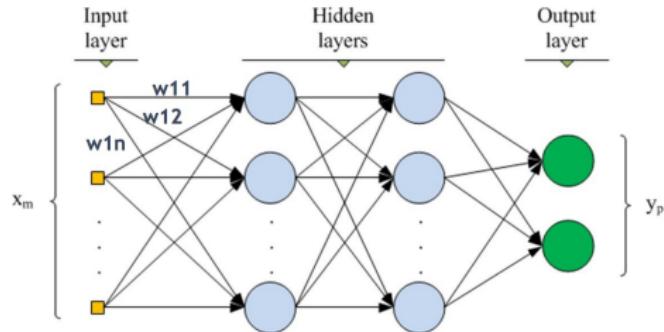
# Redes neuronales

## Idea Básica



# Perceptrón

## Perceptrón multicapa

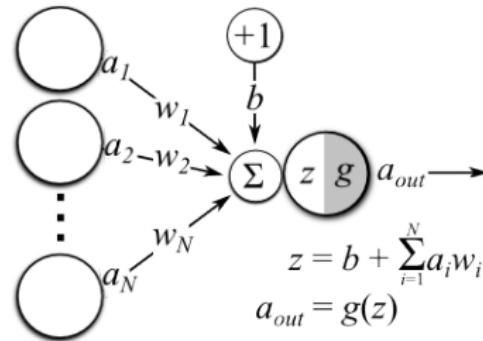


$$0 \leq x_i \leq 1 \quad \forall i \rightarrow x_i = \frac{t_i - t_{min}}{t_{max} - t_{min}} \quad t_i : \text{datos}$$

Idea básica

# Función de activación

## Función de activación



$b$ : bias (umbral)       $g(z)$ : función de activación       $0 \leq g(z) \leq 1$   
 Típicamente:  $g(z) = (1 - e^{-z})^{-1}$  (sigmoide)

# Redes neuronales

## Back-propagation

- $\mathbf{y}(\omega_i)$ : vector de salida (calculado)
- $\mathbf{s}$ : vector de salida (conocido)

Se busca minimizar el error  $e = \|\mathbf{s} - \mathbf{y}\|^2$ , resolviendo el sistema

$\frac{\partial e}{\partial \omega_i} = 0$ . Las aproximaciones sucesivas se obtienen haciendo:

$$\omega_{i+1} = \omega_i - \alpha \frac{\partial e}{\partial \omega_i} \quad (\alpha : \text{pequeño})$$

### Fases:

- **Entrenamiento**: cálculo de los pesos  $\omega_i$
- **Prueba**: Validación con otros datasets conocidos.

A partir de lo cual la red neuronal está preparada para hacer predicciones.

# Tabla de contenido

## 1 Redes neuronales

## 2 Extreme Learning Machine

- Desventajas del Back-propagation
- Teoría básica
- Casos
- Resumen
- Herramientas
- Fuentes

# Extreme Learning Machine

**Guang-Bin Huang**



Full Professor School of Electrical and Electronic Engineering Nanyang  
Technological University, Singapore ELM Web Portal:  
[www.ntu.edu.sg/home/egbhuang](http://www.ntu.edu.sg/home/egbhuang)

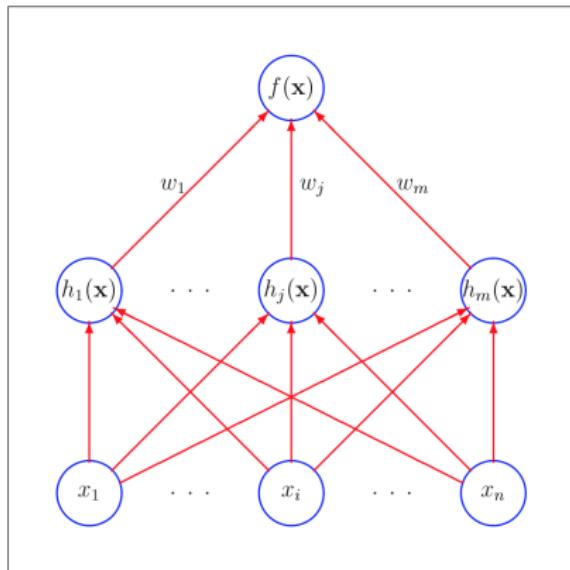
## Desventajas del Back-propagation

# Desventajas

## Desventajas del Back-propagation

- El algoritmo puede converger muy letamente o incluso diverger
- Puede parar en un mínimo local distante del mínimo global
- La red pueden ser 'super entrenada' de manera que la generalización quede perjudicada
- En muchas aplicaciones puede consumir demasiado tiempo de entrenamiento

# Características



- Una sola capa oculta (single-hidden-layer feedforward networks SLFN)
- Huang demostró que un ELM puede ser un aproximador universal si los pesos de entrada y los bias son escogidos aleatoriamente de acuerdo a cualquier distribución de probabilidad continua.
- Los pesos de la capa de salida son determinados analíticamente (no hay ciclos iterativos para ajustes de parámetros)

# Teorema

**Teorema** (capacidad de aproximación universal): Dada cualquier función no-constante seccionalmente continua como función de activación, si ajustando los parámetros de las neuronas de la capa oculta podría hacer que la red SLFN se aproxime a cualquier función continua  $f(\mathbf{x})$ , entonces, la secuencia  $\{h_i(\mathbf{x})\}_{i=1}^L$  puede ser generada de acuerdo a cualquier distribución continua de probabilidad, y  $\lim_{L \rightarrow \infty} \left\| \sum_{i=1}^L \beta_i h_i(\mathbf{x}) - f(\mathbf{x}) \right\| = 0$  con probabilidad igual a uno, con la apropiada elección de los pesos de salida  $\beta$ .

# Teoría Básica

- Las funciones de activación  $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_L(\mathbf{x})]$  tienen parámetros aleatorios, donde  $h_i(\mathbf{x}) = G_i(\boldsymbol{\omega}_i, b_i, \mathbf{x})$ ,  $\boldsymbol{\omega}_i$ : pesos;  $b_i$ : bias (umbrales),  $i = 1, \dots, L$  (número de neuronas en la capa oculta).
- De esta forma, la función de salida  $o_j$  puede expresarse como:  

$$\sum_{i=1}^n \beta_i G_i(\boldsymbol{\omega}_i, b_i, \mathbf{x}) = o_j$$
- Para la fase de entrenamiento  $\sum_{i=1}^n \beta_i G_i(\boldsymbol{\omega}_i, b_i, \mathbf{x}) = \mathbf{t}_j$ , o, en forma matricial  $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ , de donde:  $\boldsymbol{\beta} = \mathbf{H}^+ \mathbf{T}$   
 $(\mathbf{H}^+)$ : inversa generalizada de Moore-Penrose)

# Algoritmo

"La Matemática Simple es Suficiente" ELM es un algoritmo simple de sintonización de tres pasos:

- Paso 1: Generar de forma aleatoria los parámetros de los nodos ocultos
- Paso 2: Calcular la matriz de salida de la capa oculta
- Paso 3: Calcular los pesos de la capa de salida

Casos

$$\frac{\sin x}{x}$$

## Aproximación de la función $\frac{\sin x}{x}$

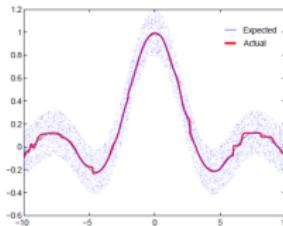
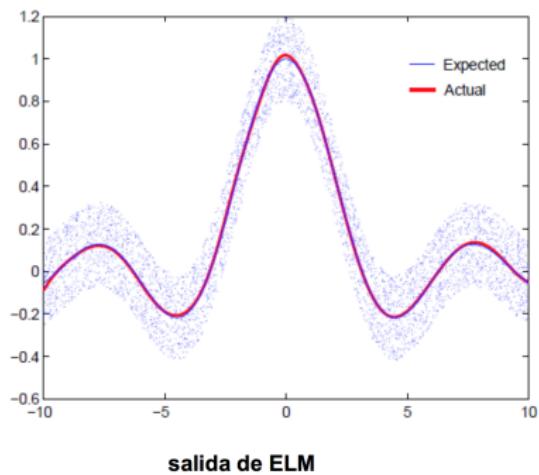
Comparación de rendimiento para la función de aprendizaje:  $\frac{\sin x}{x}$  (5000 datos de entrenamiento ruidosos y 5000 datos de prueba sin ruido). OBS:  
 SVR = Support Vector Regression - BP: Backpropagation

Algorithms	Training Time (seconds)	Training		Testing		# SVs/ nodes
		RMS	Dev	RMS	Dev	
ELM	0.125	0.1148	0.0037	0.0097	0.0028	20
BP	21.26	0.1196	0.0042	0.0159	0.0041	20
SVR	1273.4	0.1149	0.0007	0.0130	0.0012	2499.9

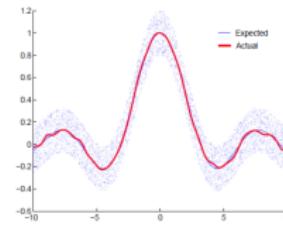
Casos

$$\frac{\sin x}{x}$$

## Aproximación de la función $\frac{\sin x}{x}$



BP



SVM

# Casos de éxito

## Utilidad exitosa en:

- Predicción de la tasa de crecimiento de la población
- Predicción de volatilidad de precios en acciones
- Predicción temprana de heladas meteorológicas (Bolivia)
- Segmentación y etiquetado de 3D
- Reconocimiento de imágenes
- Muchos otros...

## Resumen

## Resumen

- ELM necesita mucho menos tiempo de entrenamiento en comparación con BP y SVM / SVR.
- La precisión de predicción de ELM suele ser ligeramente mejor que BP y cerca de SVM / SVR en muchas aplicaciones.
- En comparación con BP y SVR, ELM se puede implementar fácilmente ya que no hay ningún parámetro a sintonizar excepto un parámetro insensible  $L$ .
- Debe señalarse que muchas funciones de activación no lineales se pueden utilizar en ELM.
- ELM necesita más nodos ocultos que BP pero mucho menos nodos que SVM / SVR, lo que implica que ELM y BP tienen mucho menor tiempo de respuesta a datos desconocidos que SVM / SVR.

## Herramientas

# Herramientas disponibles para ELM

- Python Scikit-Learn <http://scikit-learn.org/stable/>
- ELM en Python <https://github.com/dclambert/Python-ELM>
- TENSORFLOW <https://www.tensorflow.org/>
- librería de Machine learning: MLLib, JAVA, Weka, Massive Online Analysis(MOA) (<http://java.ml.sourceforge.net/>,  
<http://spark.apache.org/>, <https://rapidminer.com/>,  
<http://samoa.incubator.apache.org/>, <http://mahout.apache.org/>)

## Fuentes

## Fuentes

- HUANG, G. B. (2006, November). Introduction to Extreme Learning Machines. In Workshop on Machine Learning for Biomedical Informatics, Nov 2006.
- Milacic, L., Jovic, S., Vujovic, T., Miljkovic, J. (2017). Application of artificial neural network with extreme learning machine for economic growth estimation. *Physica A: Statistical Mechanics and its Applications*, 465, 285-288.
- Caner Ulku - Oguz Onuk (2015, june). Volatility Prediction by Using ELM and GARCH Model. Bogazici University – Istanbul.
- Xie, Z., Xu, K., Liu, L., Xiong, Y. (2014, August). 3d shape segmentation and labeling via extreme learning machine. In *Computer graphics forum* (Vol. 33, No. 5, pp. 85-95).
- Riabani Mercado, F. (2015, september). predicción temprana de heladas meteorológicas usando Extreme Learning Machine, Universidad Católica Boliviana San Pablo Cochabamba.
- Huang, G. B. (2015). What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation*, 7(3), 263-278.

# Muchas gracias