Un modelo de predicción robusto al ruido

Edwin Salazar[1], **Marco Mora**[2], **Angel Vásquez**[3]

[1,3]Doctorado en Modelamiento Matemático,
Universidad Católica del Maule, Talca, Chile.
[2]Laboratorio de Investigaciones Tecnológicas en Reconocimiento
de Patrones (LITRP), Universidad Católica del Maule, Talca, Chile.

[1]edwin.salazar@alu.ucm.cl, [2]marcomoracofre@gmail.com, [3]jvasquez.12.c@gmail.com

1. Introducción

El error esta presente en todo sistema de medida, donde el observador, las condiciones ambientales, la precisión del instrumento, entre otras, hacen que los resultados tengan mayor o menor ruido. Un buen diseño de predicción debe tener en cuenta estos errores de modo que no conduzca a un comportamiento inestable; por tanto, no tiene sentido aspirar a encontrar un sistema que prediga exactamente los valores medidos, y se debe admitir que cometerá cierto error.

Una manera de hacer un modelo de predicción es utilizando la regresión lineal estándar, en la cual se busca ajustar una función de línea recta a través de los valores de una variable independiente, sin embargo, si se quiere una aproximación más general se puede utilizar un modelo de red neuronal artificial donde no hay ninguna restricción, la línea no tiene porqué ser recta; esta característica complica su diseño pero optimiza los resultados ya que, en base al ajuste de una función, se pueden encontrar los pesos correctos o coeficientes, lo que las hace flexibles a la hora de modelar las relaciones complejas del mundo real que son difíciles de manipular usando las técnicas clásicas.

En este sentido se busca estudiar el comportamiento frente al ruido de las redes neuronales, en particular las de avance de una sola capa oculta (ver figura 1), siendo el algoritmo Extreme Learning Machine (ELM) el más utilizado para el entrenamiento, el cual asigna aleatoriamente pesos de entrada y determina analíticamente los pesos de la salida mediante la inversa de Moore-Penrose (pseudoinversa), este algoritmo tiende a proporcionar una velocidad de aprendizaje extremadamente rápida. El algoritmo ELM estándar para m muestras distintas $(x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m$, con n neuronas ocultas como se muestra en la figura 1 puede ser expresada matemáticamente por:

$$\sum_{i=1}^{n} \beta_i g(w_i * x_j + b_i) = t_j, \quad j = 1, \ldots, m, \tag{1}$$

donde

- $g(x)$ es la función de activación, en este caso se utiliza la función sigmoide.
- w_i es el vector de peso que conecta la i-ésima neurona oculta con cada entrada.
- β_i es vector de peso que conecta la i-ésima neurona oculta con cada salida.
- b_i es el bias o sesgo de la i-ésima neurona oculta.

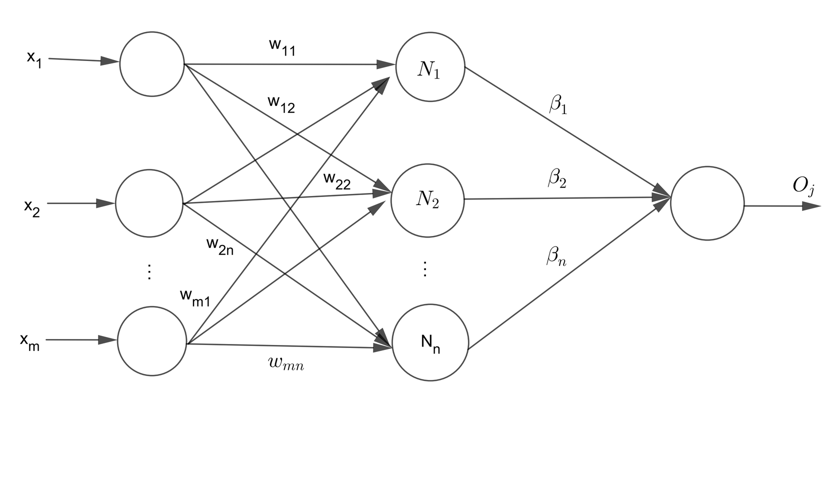

Figura 1: *Red neuronal de una sola capa*

La ecuación (1) puede escribirse en forma compacta como

$$H\beta = T,$$

luego $\beta = H^{\dagger}T$, donde $H^{\dagger}$ es la pseudoinversa. En este trabajo se muestra que la pseudoinversa pierde precisión cuando se usan datos ruidosos, es por esto que se propone un método para la robustez del algoritmo ELM al ruido. El método consiste en estudiar el comportamiento del descenso del gradiente para el cómputo de los pesos de salida de la red, ya que a diferencia de la inversa es un proceso de optimización no lineal sin restricciones dado por

$$\mathbf{a}_{(t+1)} = \mathbf{a}_{(t)} - \alpha(\nabla J)_{(t)},$$

donde a son los parámetros a estimar, $\nabla J_{(t)}$ el gradiente de la función de costo $J(t)$ y α la tasa de aprendizaje.

2. Materiales y Métodos

Se utiliza el software Matlab para la implementación de la red neuronal monocapa. Los datos de entrenamiento y prueba para la red son generados con el comando rand de Matlab alrededor de una función polinómica. La validación de la robustez frente al ruido del descenso del gradiente se realiza al estudiar el error en función del ruido en los datos. Además se consideran diferentes escenarios para los datos aumentando el grado del polinomio.

A continuación se muestra los pasos para la construcción de los datos ruidosos.

1. Se define la función polinómica $y = p(x)$, donde x es el vector de entrada e y es la salida esperada.
2. Haciendo `R=r*(2*rand([length(x)\ 1])-1)` en Matlab se genera un vector R de números aleatorios con distribución uniforme entre $-r$ y r del mismo tamaño del vector x ; r es la amplitud del ruido.
3. De los pasos anteriores se obtiene la salida con ruido $T = y + R$.

Es de notar que los datos generados tienen ruido aditivo que se puede variar ajustando el valor de r. Una vez se tienen los datos se hace el respectivo ajuste utilizando la pseudoinversa, comando `pinv` en Matlab, y el descenso del gradiente con función de costo el error cuadrático medio (ECM) con regularización para datos ruidosos, definida como

$$J = \beta \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2 + (1-\beta) \frac{1}{M} \sum_{j=1}^{M} a_i^2, \tag{2}$$

donde β es el término de regularización que controla el aporte del error cuadrático medio y el cuadrado de los parámetros a_i; y_i son los valores reales de la variable dependiente e $\hat{y}$ son los valores estimados en una muestra de tamaño N.

La implementación de la red se hace de acuerdo con la ecuación (1) utilizando 500 datos y de manera aleatoria se elige un 70 % para el entrenamiento y un 30 % para el test. Luego se calculan los pesos de salida de la red con la pseudoinversa y con el descenso del gradiente, para finalmente comparar los errores en el test de cada uno de los métodos.

3. Resultados

En la figura 2a se muestra el comportamiento del error de los métodos en función del ruido utilizando un polinomio de grado 2 (3 parámetros a estimar). En la figura se observa que para ruidos de baja amplitud (menores a 0.2) la pseudoinversa y el descenso del gradiente con función de costo el ECM (término de regulación igual a uno) tienen un comportamiento similar, pero a medida que el ruido aumenta la pseudoinversa pierde exactitud. Cabe resaltar que la modificación en la regularización del ECM no es siempre efectiva para cualquier tipo de datos sino que depende del ruido, por ejemplo, utilizando $\beta = 0{,}9$, los datos deberían tener aproximadamente un ruido de amplitud dos como se muestra en de la 2a.

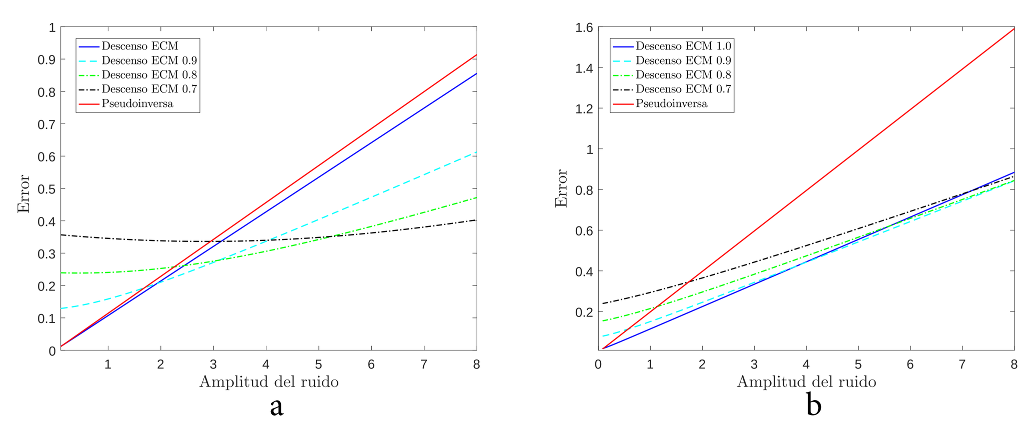

Figura 2: *Comportamiento de los métodos en función del ruido con términos de regularización $\beta = 1{,}0$, $\beta = 0{,}9$, $\beta = 0{,}8$ y $\beta = 0{,}7$: a) comportamiento en la determinación de 3 parámetros; b) comportamiento en la determinación de 15 parámetros.*

En la figura 2b se determinan 15 parámetros donde se observa que el descenso del gradiente con función de costo el ECM tiene un mejor comportamiento a los datos con ruido que la pseudoinversa.

Para el propósito de las redes neuronales como aproximador funcional se debe tener en cuenta que la predicción depende del número de neuronas, por tanto, aunque el término de regularización cumpla la función de reducir el ruido en la determinación de pocos parámetros, no hay mucha diferencia con el descenso del gradiente con función de costo ECM cuando se trata de la estimación de muchos parámetros, lo cual es importante al momento de reducir el costo computacional.

Teniendo en cuenta lo anterior se realiza la implementación de la red neuronal y se genera el modelo de predicción utilizando 500 datos que tienen un ruido $r = 1{,}5$. En 40 neuronas se obtiene el mínimo error de 2,4 en la determinación de los pesos de salida con la pseudoinversa, el ajuste de la curva se muestra en la figura 3a.

En la figura 3b se observa el ajuste de la curva cuando se determinan los pesos de salida utilizando el descenso del gradiente con función de costo el ECM, obteniendo en este caso un error de 0.32 para 40 neuronas, lo cual muestra que tiene una mejor aproximación que la pseudoinversa.

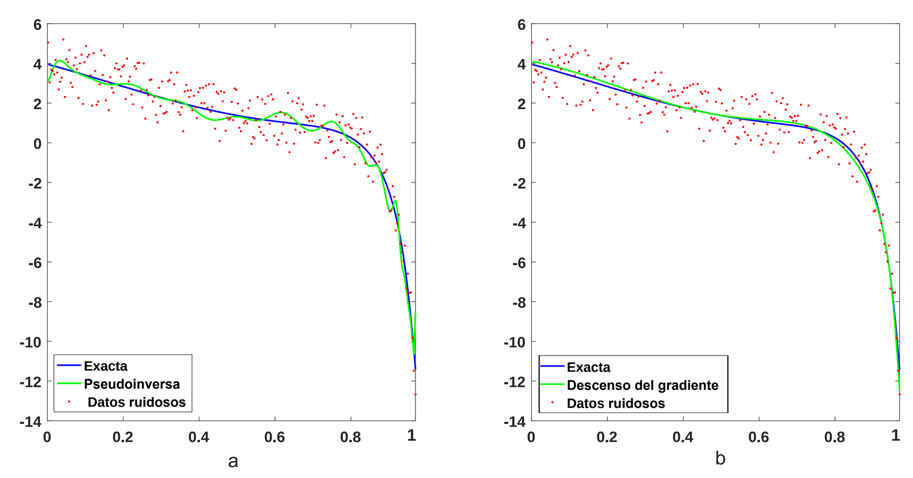

Figura 3: *Modelo generado por la red neuronal utilizando 500 datos con ruido aditivo: 350 para entrenamiento y 150 para test. a) estimación de los pesos de salida de la ELM con la pseudoinversa; b) estimación de los pesos de salida de la ELM con el descenso del gradiente con función de costo el ECM.*

4. Conclusiones

El término de regularización en la función de costo hace que el descenso del gradiente sea robusto al ruido en la determinación de pocos parámetros. Sin embargo, en el algoritmo de aprendizaje ELM no es factible puesto que generalmente se necesita determinar un número considerable de pesos de salida para generar un buen modelo. El descenso del gradiente sin término de regularización mejora la estimación de los pesos de salida cuando los datos de entrada tienen ruido aditivo.

5. Agradecimentos

Salazar, E. agradece a los compañeros de la cohorte 2019 del Doctorado en Modelamiento Matemático Aplicado.

Referencias

[1] You, Z. H., Li, S., Gao, X., Luo, X., & Ji, Z. (2014). Large-scale protein-protein interactions detection by integrating big biosensing data with computational model. BioMed research international.

[2] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3),489-501.

[3] Zhu, Q. Y., Qin, A. K., Suganthan, P. N., & Huang, G. B. (2005). Evolutionary extreme learning machine. *Pattern recognition*, 38(10), 1759-1763.

[4] Tsuruoka, Y., Tsujii, J. I., & Ananiadou, S. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1 (pp. 477-485).

[5] Friedman, J., & Popescu, B. E. (2004). Gradient directed regularization. Unpublished manuscript, *http://www-stat. stanford. edu/ jhf/ftp/pathlite. pdf.*