

Instituto Politécnico Nacional
Escuela Superior de Cómputo

Asignatura:
Neural Networks

Grupo: 3CM2

Tarea 9. Backpropagation

Alumno:
Garcia Garcia Rafael
Profesor: Moreno Armendariz Marco Antonio

El algoritmo de backpropagation es uno de los más populares dentro del aprendizaje de las redes neuronales y nos ayuda a que una red de aprendizaje supervisado pueda ir ajustando sus valores de pesos y bias tomando como factores principales la sensibilidad, de una red por medio de la derivada de sus funciones de activación.

Durante esta tarea se programó una red neuronal con la siguiente arquitectura:

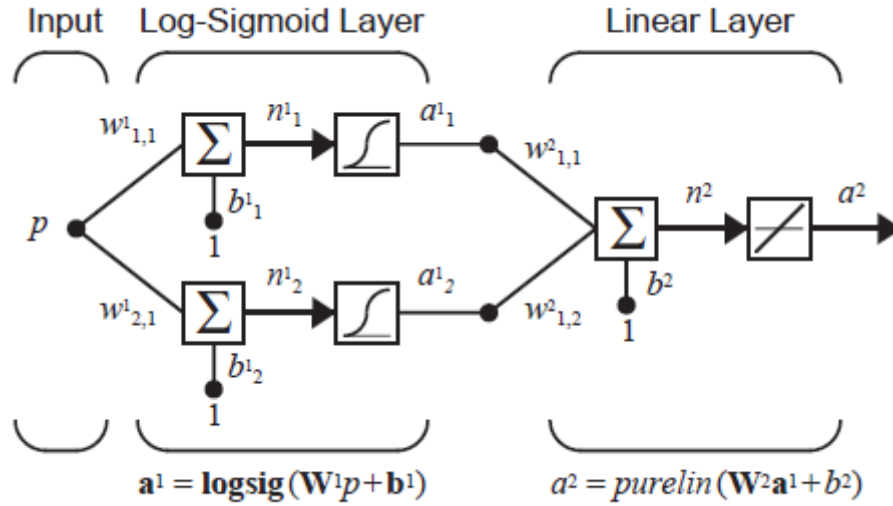


Figura 1. Arquitectura de la red neuronal (Hagan)

La arquitectura muestra una red neuronal en modo regresivo que busca aprender a representar los valores de la siguiente función:

$$g(p) = 1 + \sin\left(\frac{\pi}{4}p\right) \text{ for } -2 \leq p \leq 2.$$

Desarrollo de la práctica:

Lo primero que se realizó en la práctica fue la propagación hacia delante de las funciones de activaciones y nuestro set de entrenamiento el cuál iba de -2 a 2. Una vez hecha la propagación hacia delante de la red neuronal se evalúa el error y posteriormente se realiza la ejecución del algoritmo de backpropagation con las siguientes ecuaciones.

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}),$$

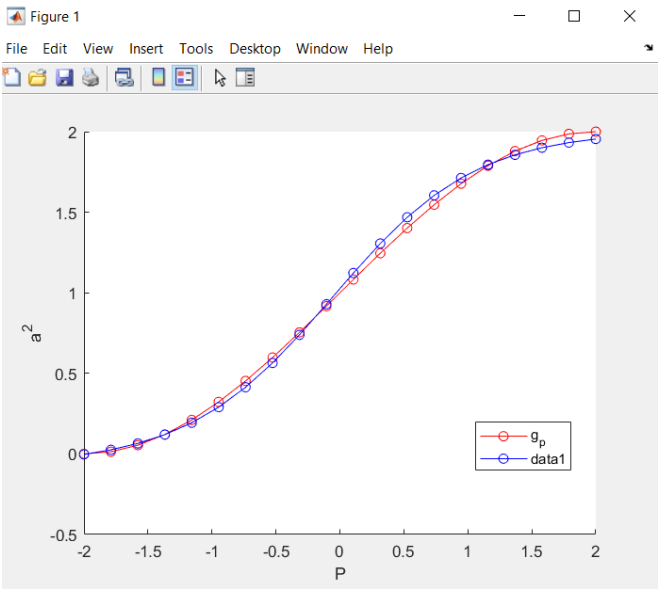
$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \text{ for } m = M-1, \dots, 2, 1,$$

where

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & \dot{f}^m(n_{s^m}^m) \end{bmatrix},$$

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}.$$

Como salida del experimento se representa con la gráfica del color azul, esto es importante porque la que está representada de color rojo es la ecuación $g(p)$, se puede concluir de esto que nuestra tarea tuvo un rotundo éxito



Gráfica 1. $G(p)$ vs Red neuronal

Por último se anexa la salida del programa en consola, esta salida muestra la evolución del error y como converge con Epoch. También se muestran los valores finales de pesos y bias junto con la salida del vector a y del error final.

(Este es el error de la red)	La red convergió
2.3131	Error final
	8.3774e-04
0.9029	Valores de pesos finales:
0.4614	W1
0.2533	-1.7049
	1.6607
0.1536	b_1
0.0926	0.6676
	0.4532
0.0542	W_2
0.0305	-0.9012 1.2026
0.0167	b_2
0.0090	0.8535
0.0048	Vector de salida a:
	0.0319
0.0025	0.0633
	0.1059
	0.1628
	0.2371

0.0014	0.3315
	0.4476
8.3774e-04	0.5847
	0.7398
	0.9071
	1.0792
	1.2477
	1.4049
	1.5450
	1.6643
	1.7618
	1.8388
	1.8979
	1.9421
	1.9745

Bibliografía

Hagan, M. T. (s.f.). Neural Network Design. En M. T. Hagan, *Neural Network Design* (pág. 1010).