Instituto Politécnico Nacional

Escuela Superior de Cómputo

Asignatura:

Neural Networks

Grupo: 3CM2

Tarea 8. Propagación hacia adelante MLP

Alumno:

Garcia Garcia Rafael

Profesor: Moreno Armendariz Marco Antonio

1. Introducción

Está tarea tenía como fin entender visualmente como influyen los valores de los pesos y bias en una red neuronal. Al programar redes Neuronales lo primero que se necesita es programar la propagación hacia delante de las diferentes capas internas de la red para posteriormente aplicar un algoritmo de aprendizaje en los valores de pesos y bias que minimice la función de costos (error).

2. Marco Teórico:

Durante la clase pudimos aprender lo que es necesario antes de plicar el algoritmo de Back propagation en un MLP por lo cuál como primera fase se necesita hacer una propagación hacia adelante en nuestra red neural que está representada de la siguiente manera:

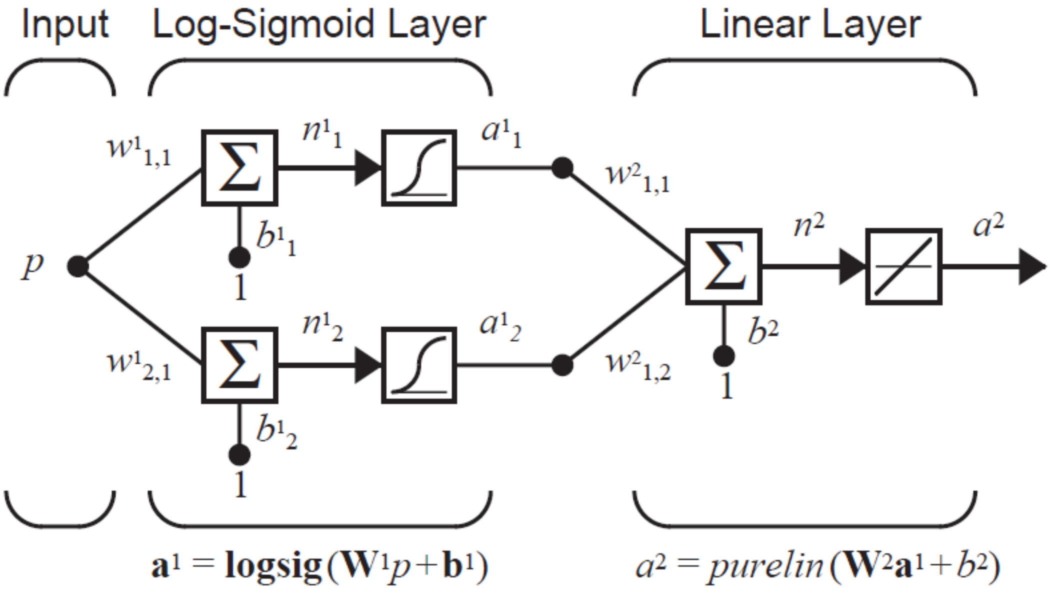


Figure 11.4 Example Function Approximation Network

Figura.1 : Red neuronal de la compuerta XOR (Hagan)

Los valores que se tomaron como base para los pesos y bias de la arquitectura fueron los siguientes:

$$w^1_{1,1} = 10,\ w^1_{2,1} = 10,\ b^1_1 = -10,\ b^1_2 = 10,$$

$$w^2_{1,1} = 1,\ w^2_{1,2} = 1,\ b^2 = 0.$$

Sobre estos valores se iteraron y para el despliegue de las primeras 4 gráficas que se muestran en la gráfica 1, se tomaron los siguientes rangos:

$$-1 \le w^2_{1,1} \le 1,\ -1 \le w^2_{1,2} \le 1,\ 0 \le b^1_2 \le 20,\ -1 \le b^2 \le 1.$$

3. Desarrollo de la tarea

Para esta pequeña introducción a el perceptrón multicapa se realizó la multiplicación de los vectores **W** y **p** más el bias, para posteriormente aplicar la función de activación en cada una de las capas y obtener un valor de salida final el cuál se verá influido de diferente manera dependiendo que valor de los pesos y bias cambie.

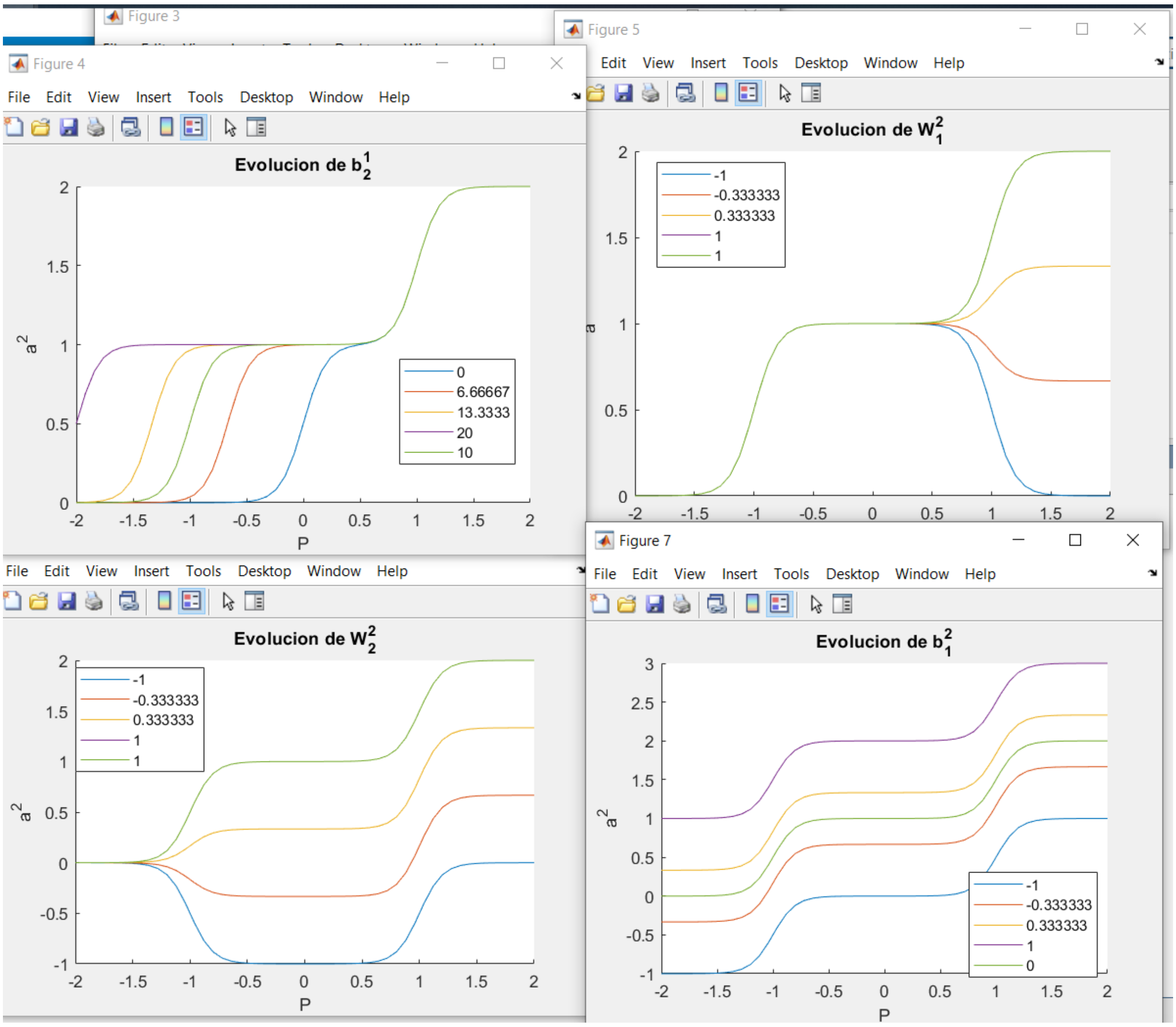
Las ecuaciones implementadas para la propagación hacia delante del MLP se sacaron del libro del curso (Hagan) y son las siguientes:

$$\mathbf{a}^0 = \mathbf{p},$$

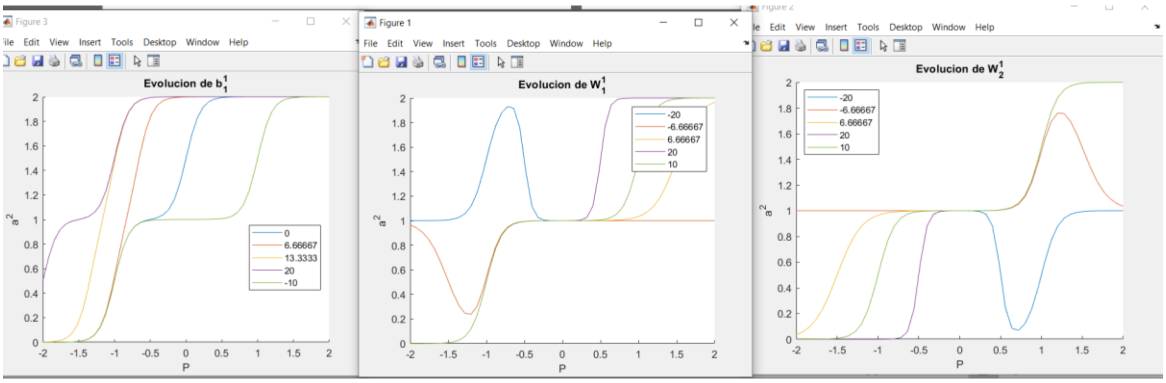
$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \text{ for } m = 0, 1, \ldots, M-1,$$

$$\mathbf{a} = \mathbf{a}^M.$$

Para lograr lo mencionado se tuvo que gráficar cada uno de los pesos y bias que existen en nuestro MLP. En las primeras 4 imágenes de la **Grafica 1** se ve que nuestro código generó las mismas salidas vistas en el libro (Hagan).



Grafica 1. 4 valores de pesos y bias propuestos en la tarea



Grafica 2. 3 imágenes faltantes del experimento.

En la **Grafica 2** se observa la salida de las otras 3 imágenes que no se mostraron en el libro.

La salida en terminal es la siguiente

```
>> tarea_8
Rango a modificar de W_1(1)
Ejemplo: [-20,20]

[-20,20]
Rango a modificar de W_1(2)
Ejemplo: [-20,20]

[-20,20]
Rango a modificar de bias_1(1)
Ejemplo: [0,20]

[0,20]
Rango a modificar de bias_1(2)
Ejemplo: [0,20]

[0,20]
Rango a modificar de W_2(1)
Ejemplo: [-1,1]

[-1,1]
Rango a modificar de W_2(2)
Ejemplo: [-1,1]

[-1,1]
Rango a modificar de bias_2(1)
Ejemplo: [-1,1]

[-1,1]
```

4. Conclusión

El experimento fue todo un éxito debido a que pudimos replicar las figuras que se mostraron en el libro, por lo cuál también se observó con éxito la influencia de pesos y bias en nuestro MLP.

Bibliografía

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