UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

Tarea n°2: Sistemas Adaptativos – 543820.

Jocelyn Matus Ancavil.

Profesor, Daniel Sbarbaro.

Concepción, jueves 14 de octubre 2021

Problema 1

Se desea encontrar los parámetros de un combinador lineal sin polarización tal que pueda categorizar en dos clases los vigentes vectores: Todos los vectores tienen igual probabilidad de ocurrencia.

$\{p_1 = [\ -1\ 2\ ],\ t_1 = -1\}\{p_2 = [\ 2\ -1],\ t_2 = -1\}\{p_3 = [\ 0\ -1],\ t_3 = 1\}\{p_4 = [\ -1\ 0\ ],\ t_4 = 1\}$ W = [2 0] 1.1. Dibujar el diagrama del modelo. (0.25)

El diagrama del modelo es el siguiente:

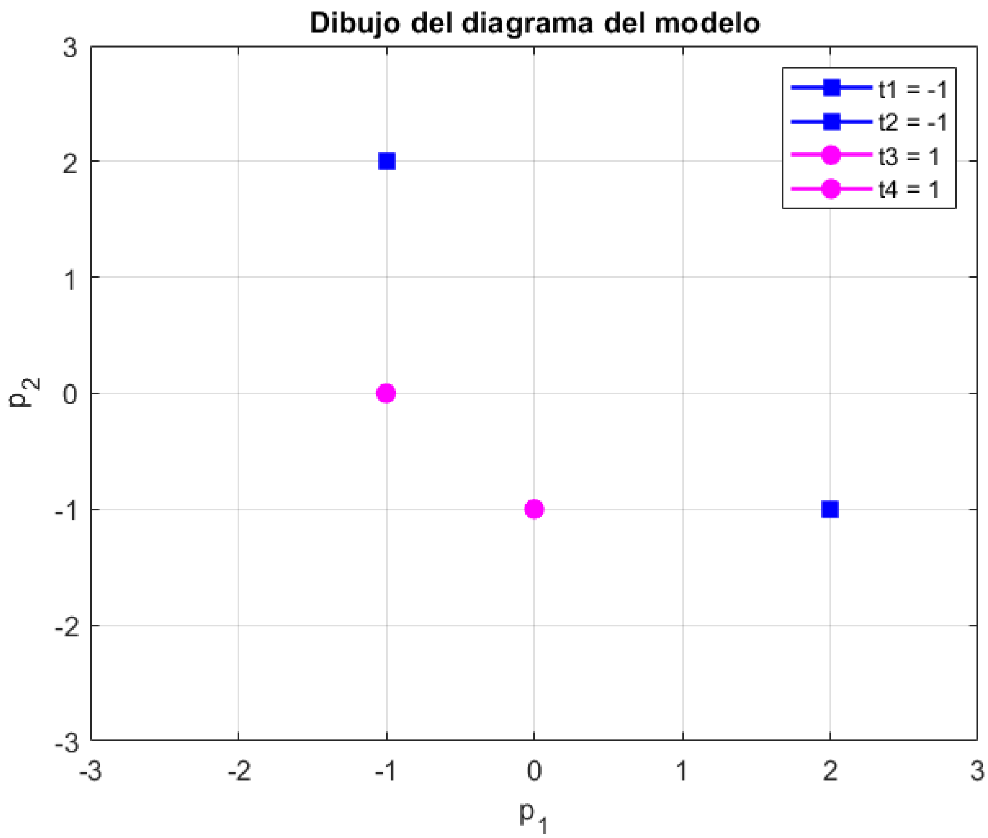


Figura N°1, Dibujo del diagrama del modelo. Obtenido mediante comando plot

Se puede ver que, a través de una recta, se puede realizar una clasificación lineal entre las dos clases t1 y t2, además es posible usar solo dos capas ya que las clases son conexos.

1.2. ¿Cuáles son los pesos óptimos? Incluir todos los cálculos. (0.25)

El peso optimo se obtiene utilizando la expresión Hebb:

$$W = TP^{+} \qquad (1)$$

Siendo P^+ la pseudo inversa de la matriz P. Se tiene que realizar el cálculo de la pseudo inversa ya que los vectores de la matriz P no son ortogonales entre si, por lo que no es posible calcular expresiones cerrada como en el caso de Fourier y se debe recurrir a métodos de optimización.

En este caso, las matrices T y P son las siguientes:

$$P = \begin{bmatrix} -1 & 2 & 0 & -1 \\ 2 & -1 & -1 & 0 \end{bmatrix} \quad (2)$$

$$T = \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} \quad (3)$$

Realizando los cálculos, se obtiene lo siguiente:

$$W = \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} \left(\begin{bmatrix} -1 & 2 \\ 2 & -1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 2 & 0 & -1 \\ 2 & -1 & -1 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} -1 & 2 \\ 2 & -1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$\rightarrow W = \begin{bmatrix} -1 & -1 \end{bmatrix} \quad (4)$$

Entonces, los pesos óptimos son los que se muestran en la ecuación (4)

1.3. Dibuje el límite de decisión óptimo. (0.25)

El límite de decisión optimo, según los pesos que se calcularon en el ítem anterior, es el siguiente:

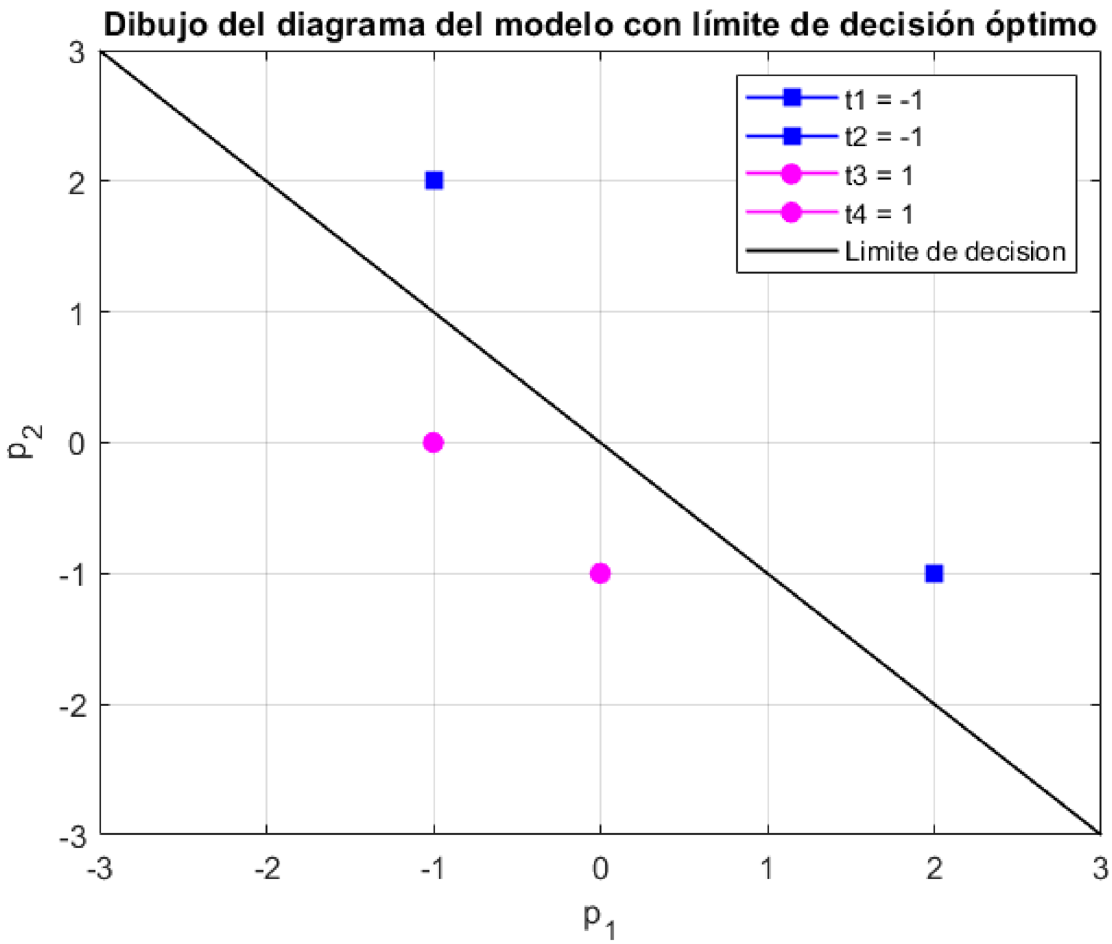


Figura N°2, Dibujo del diagrama del modelo, con límite de decisión optimo incluido. Obtenido mediante comando plot

El límite de decisión es una recta que esta equidistante a todos los puntos, separándolos en las clases correspondientes a cada uno.

1.4. ¿Cómo cree que cambiaría el límite si se permitiera que la red tuviera un sesgo? Indique la nueva posición aproximada en su boceto de la parte d). (0.25)

No habría un mayor cambio, en realidad, ninguno, ya que, sin que haya un sesgo, el límite de decisión es óptimo. Si se fuera a aplicar un sesgo, este ya no sería optimo según la clasificación que se quiere realizar.

1.5. ¿Cuál es la tasa de aprendizaje estable máxima para el algoritmo LMS?(0.25)

La tasa de aprendizaje máxima del algoritmo LMS viene de la siguiente expresión:

$$0 < \alpha < \frac{1}{\max(\lambda_i)} \quad (5)$$

Con α la tasa de aprendizaje máxima, y λ_i los valores propios de la matriz R de información. Esta matriz R proviene de la función del error cuadrático:

$$F(x) = c - 2x^T h + x^T R x \quad (6)$$

Siendo los coeficientes de esta función las siguientes matrices:

$$c = E(t^2), \qquad t = Matriz\ T \quad (7)$$

$$h = E[t \cdot z]\,, \qquad z = Matriz\ P \quad (8)$$

$$R = E[z \cdot z^T] \quad (9)$$

Entonces, teniendo eso en cuenta, se necesita, para saber la tasa de aprendizaje, los valores propios de la matriz R. Realizando el cálculo, se obtiene esto:

$$R = E\left[\begin{bmatrix} -1 & 2 & 0 & -1 \\ 2 & -1 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 & 2 \\ 2 & -1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}\right]$$

$$\rightarrow R = \begin{bmatrix} 1.5 & -1 \\ -1 & 1.5 \end{bmatrix} \quad (10)$$

Y los valores propios de esta son los siguientes:

$$\lambda_{1,2} = 0.5, \ \ 2.5 \quad (11)$$

Aplicándolo en la expresión (5), la tasa de aprendizaje máxima es:

$$\alpha = 0.4 \quad (12)$$

1.6. Dibuje la gráfica de contorno de la superficie de rendimiento del error cuadrático medio. (0.25)

La grafica de contorno es la siguiente:

Figura N°3, Contorno de la superficie de rendimiento.

La función converge a un mínimo que se encuentra en (-1, -1), con un valor de 1.

1.7. En su gráfica de contorno de la parte f) dibuje la ruta del algoritmo LMS para una tasa de aprendizaje muy pequeña (por ejemplo, 0.001) a partir de la condición inicial . Esto no requiere ningún cálculo, pero explique cómo obtuvo su respuesta. (0.25)

La trayectoria del algoritmo LMS es el siguiente:

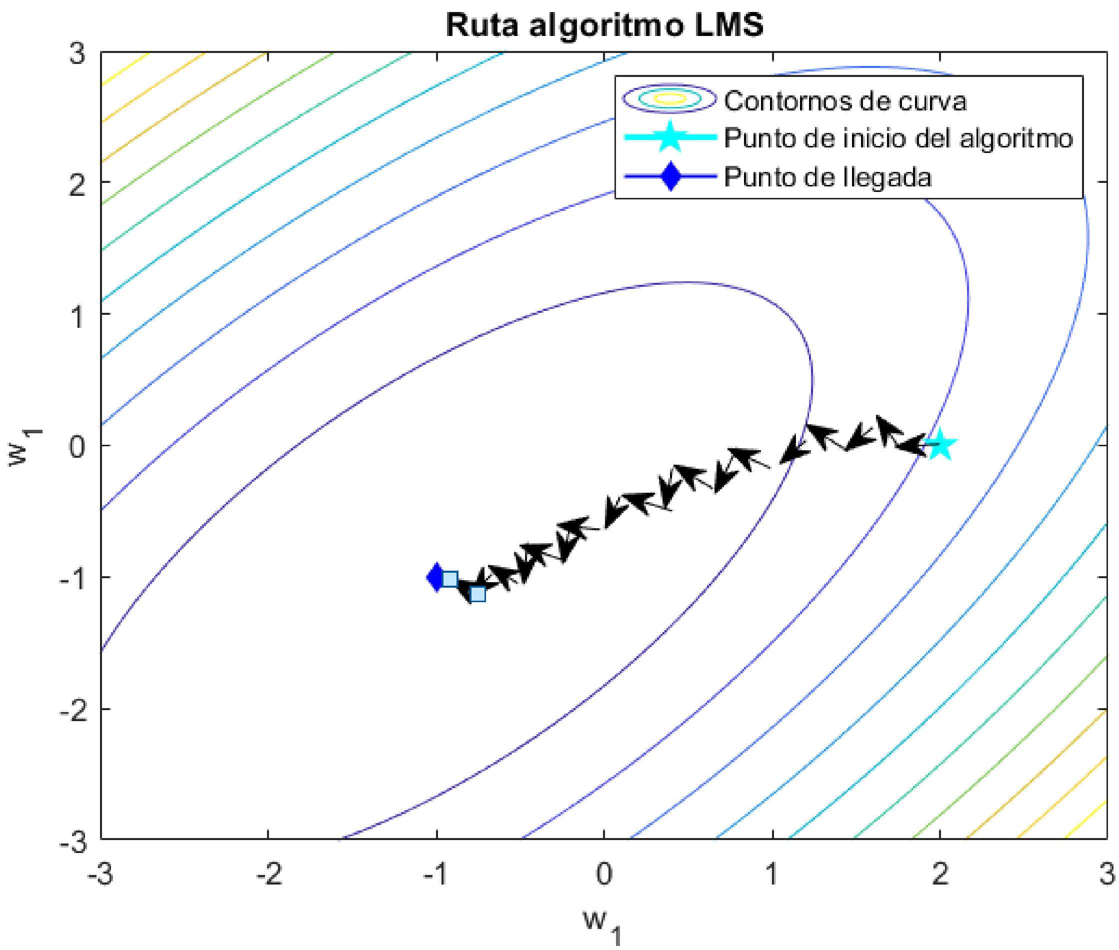


Figura N°4, Dibujo ruta algoritmo LMS. Obtenida mediante el comando contour

La trayectoria, sin realizar algún calculo, se obtuvo asi: primero, para valores muy pequeños de tasa de aprendizaje, la trayectoria se mueve de forma perpendicular a las curvas de contorno. Al algoritmo se le aplica una entrada p, se calculan los pesos y se actualizan, y después se realiza de nuevo con otra entrada, realizando el mismo proceso hasta llegar al mínimo local. Al tener distintas entradas, la dirección del algoritmo se va cambiando constantemente, por lo que no se crea una trayectoria suave hacia el mínimo local.

1.8 Dibuje la trayectoria del algoritmo del gradiente (Batch) y comparé con la obtenida con LMS, Comente. (0.25)

La trayectoria del gradiente (Batch) se muestra en el siguiente grafico:

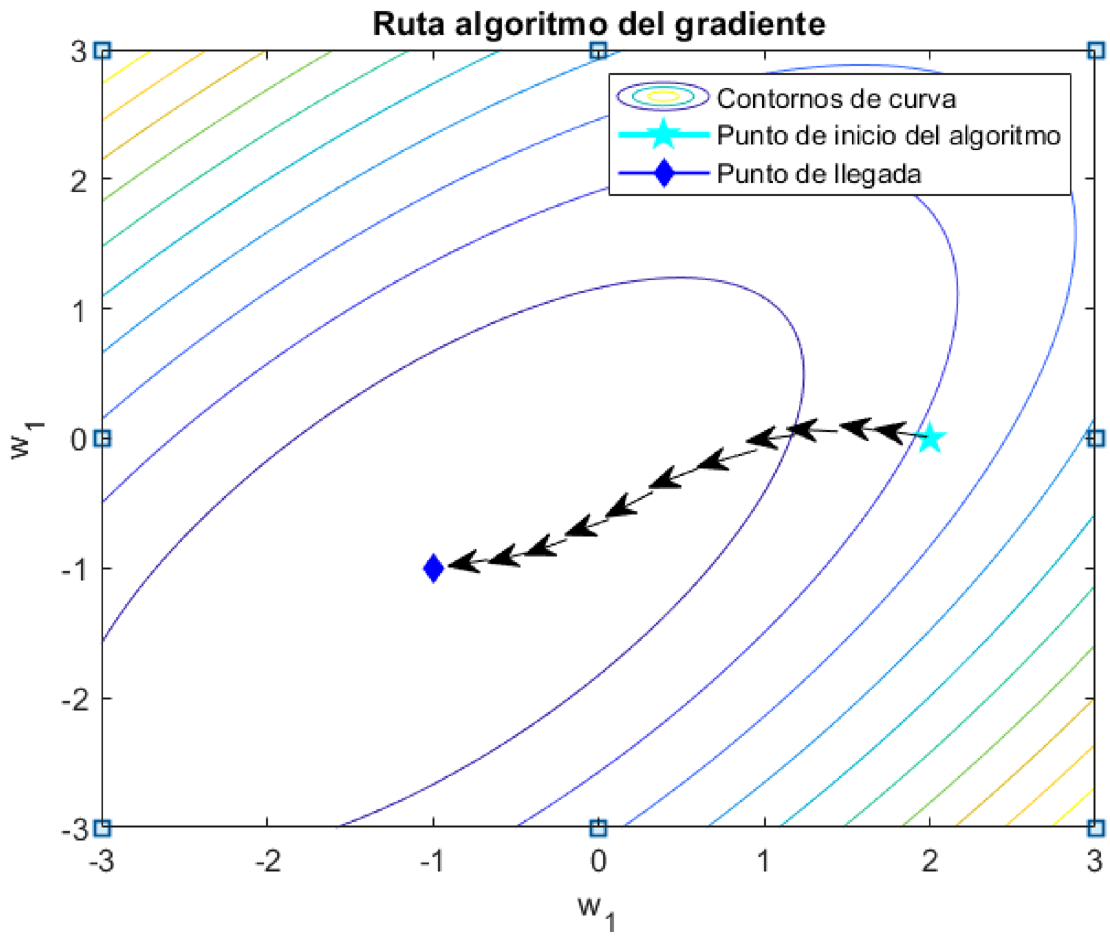


Figura N°1, Dibujo ruta algoritmo del gradiente. Obtenida mediante el comando contour

A diferencia del caso anterior, con el algoritmo del gradiente, se aplica todas las entradas p al algoritmo, se calculan los pesos correspondientes, y antes de actualizar los pesos, se promedian todos los pesos, y el peso promediado es el que se actualiza, y se repite hasta llegar al mínimo. Es por esto que, a diferencia del algoritmo anterior, la curva es suave.

Problema 2

Se desea encontrar los parámetros del modelo definido por la Figura 2.1

tal que el modelo reproduzca los siguientes valores:

$\{p_1 = [-2], t_1 = 0.8\}$ $\{p_2 = [2], t_2 = 1\}$

2.1. Graficar las curvas de nivel para el índice de rendimiento de error cuadrático medio. (0.25)

Las curvas de nivel para el índice de rendimiento de error cuadrático medio es la gráfica siguiente:

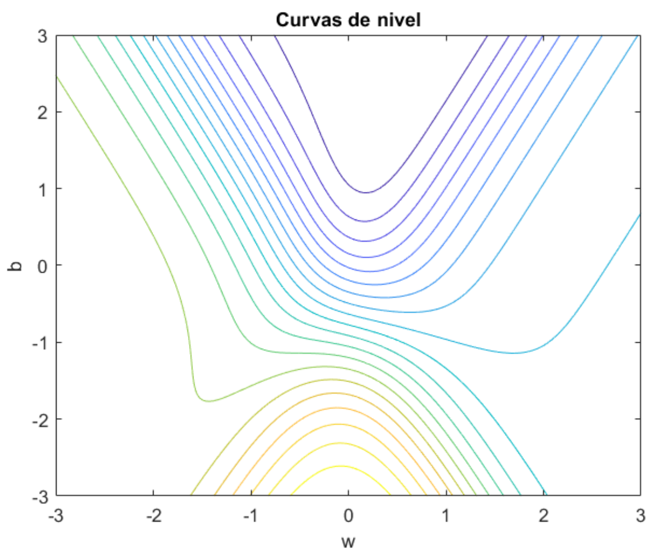


Figura N°2, Curvas de nivel para el índice de rendimiento de error. Obtenida mediante el comando contour

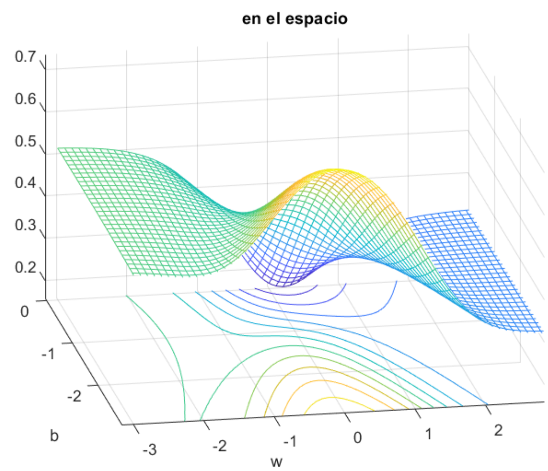


Figura N°3, En el espacio para el índice de rendimiento de error. Obtenida mediante el comando meshc

A diferencia del problema anterior, al tener una función de activación no lineal, como logsig, las curvas de contorno no convergen a un mínimo identificable de manera grafica.

2.2. Escriba las expresiones del algoritmo de retropropagación del error para este problema. (0.25)

Las expresiones del algoritmo son las siguientes:

$$1.\ a^0 = p_1, p_2,\ \ t = t_1, t_2 \quad (12)$$

$$2.\ a^1 = logsig(w^1 a^0 + b^1) \quad (13)$$

$$3.\ a = a^1$$

$$4.\ s^1 = -2(1 - a^1)(a^1)(t - a^1) \quad (15)$$

$$5.\ \begin{array}{c} w^1(k+1) = w^1(k) + 2\alpha((1 - a^1)(a^1)(t - a^1))(a^0) \\ b^1(k+1) = b^1(k) + 2\alpha((1 - a^1)(a^1)(t - a^1)) \end{array} \quad (16)$$

2.3. Demuestre el efecto del procesamiento batch en la dirección de búsqueda para el algoritmo de retropropagación basado en el gradiente con y sin procesamiento batch, comenzando desde la estimación inicial w(0) = 0, b(0) = 0.5 . (0.25)

El efecto del procesamiento batch, en lo que se refiere en la dirección de búsqueda, se puede ver de la siguiente manera:

La dirección del parámetro W es el siguiente:

$$-s^1 a^0 \quad (17)$$

Y la dirección del parámetro b la siguiente:

$$-s^1 \quad (18)$$

Y los valores que se obtuvieron son los siguientes:

Dirección	W	b
P1	-0.17	0.08
P2	0.36	0.18
Batch	0.09	0.13

Figura N°4, Valores de dirección.

Las direcciones p1 y p2 no son cercanas entre sí, yendo a direcciones distintas, por lo cual puede no facilitar la convergencia del algoritmo, mientras que, con procesamiento batch, es el promedio simple entre las dos direcciones, lo cual puede llevar a una convergencia más rápido del algoritmo. Las direcciones se pueden ver de manera gráfica:

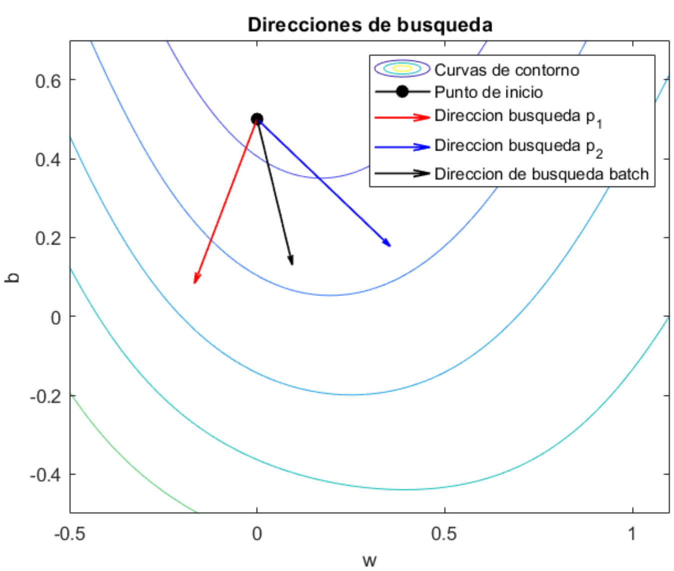


Figura N°5, Direcciones de búsqueda.

Problema 3

Considere la siguiente función cuadrática

F(x) = 0.5x^T [10 −6; −6 10] x + [4 4] x

3.1. Determine la condición que debe cumplir la tasa de aprendizaje y el coeficiente de momentum para que el algoritmo de descenso más momentum pueda minimizar esta función. (0.25)

La condición que debe cumplir la tasa de aprendizaje y el coeficiente de momentum para que el algoritmo para que pueda minimizar la función es la siguiente:

$$|(1+\gamma)-(1-\gamma)\alpha\lambda_i| < 2\sqrt{\gamma} \quad (19)$$

Con las siguientes condiciones de los parámetros:

$$\alpha > 0, \qquad \gamma > 0, \qquad \lambda_i > 0$$

Siendo λ_i los valores propios de la matriz A, la cual proviene de la función cuadrática del enunciado.

La expresión (19) proviene del libro Neural Network Design.

De la ecuación (19) se despeja el coeficiente de momentum, con las condiciones ya existentes, por lo que se tiene esto:

$$\frac{\alpha^2\lambda_i^2 - 2\alpha\lambda_i + 1}{(\alpha\lambda_i + 1)^2} < \gamma < 1 \quad (20)$$

Considere la siguiente función cuadrática

F(x) = 0.5x^T [10 −6; −6 10] x + [4 4] x

3.2. Suponga que la tasa de aprendizaje es α = 0.2. Encuentre un valor para el coeficiente de momentum γ tal que el algoritmo será estable. (0.25)

Para poder obtener algún valor para el coeficiente de momentum, se tiene que obtener los valores propios de la matriz A, los cuales son estos:

$$\lambda_{1,2} = 4 \quad ,16 \quad (21)$$

Y, según el enunciado, se tiene el siguiente valor de tasa de aprendizaje:

$$\alpha = 0.2 \quad (22)$$

Entonces, aplicando la ecuación (20), se obtiene el siguiente rango de gamma:

$$0.5313 < \gamma < 0.7363$$

De este rango, se elige un valor de gamma de 0.64

Considere la siguiente función cuadrática

F(x) = 0.5x^T [10 −6; −6 10] x + [4 4] x

3.3. Suponga que la tasa de aprendizaje es α = 20. Encuentre un valor para el coeficiente de momentum γ tal que el algoritmo será estable. (0.25)

Ya que los valores propios son los mismo, el único cambio es la tasa de aprendizaje:

Y, según el enunciado, se tiene el siguiente valor de tasa de aprendizaje:

$$\alpha = 0.2 \quad (23)$$

Entonces, aplicando la ecuación (20), se obtiene lo siguiente:

$$0.9876 < \gamma < 0.9969$$

De este rango, se elige un valor de gamma de 0.991

Considere la siguiente función cuadrática

F(x) = 0.5x^T [10 −6; −6 10] x + [4 4] x

3.4. Dibuje las trayectorias del algoritmo para los valores y tanto de la parte (b) como de la parte c) en la gráfica de contorno de F(x), comenzando desde la suposición inicial x0 = [−1 − 2.5]^T (0.25)

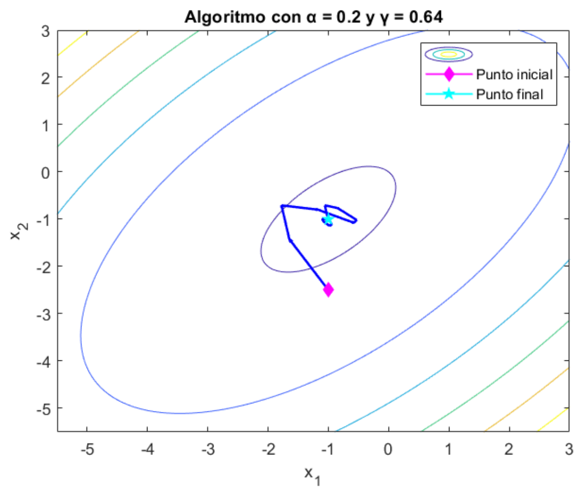


Figura N°6, Trayectoria algoritmo para α = 0.2 y γ= 0.64. obtenido mediante comando contour.

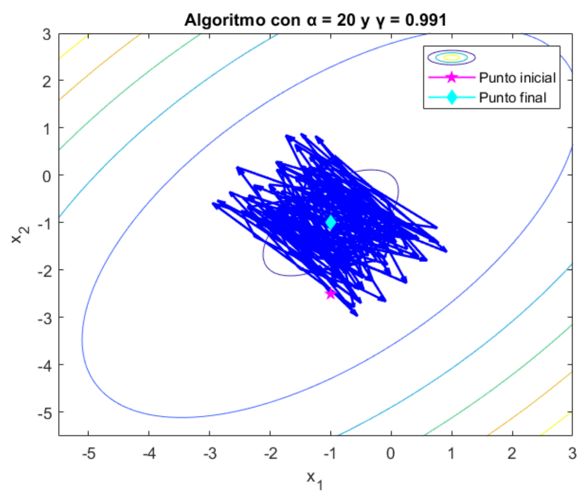


Figura N°1, Trayectoria algoritmo para α = 20 y γ= 0.991. obtenido mediante comando contour.

Los comentarios que se pueden realizar de esto es que el valor de la tasa de aprendizaje afecta mucho en la convergencia del algoritmo. Por ejemplo, si la tasa de aprendizaje es 0.2, la convergencia es mucho más ordenada y más rápida en comparación a una tasa de aprendizaje de 20.

Problema 4

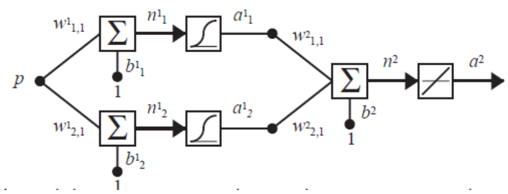
Para un modelo multicapa con una estructura 1 - S - 1, con una función lineal en la capa de salida, escriba un programa que implemente el algoritmo de retropropagación del error usando operaciones matriciales. Elija los pesos y sesgos iniciales para que sean números aleatorios distribuidos uniformemente entre -0,5 y 0,5 (utilizando la función MATLAB rand) y entrene la red para aproximar la función (4.1)

Utilice S1 = 2 y S1 = 10. Experimente con varios valores diferentes para la tasa de aprendizaje y use varias condiciones iniciales diferentes, pero el mismo conjunto de entrenamiento. Analice las propiedades de convergencia del algoritmo a medida que cambia la tasa de aprendizaje. (1)

Al realizar experimentos con este algoritmo, se pudo analizar varias propiedades de este:

1. Al tener la tasa de aprendizaje mayor o igual a 1, el algoritmo, sin importar cuantas veces se probó, nunca pudo converger a un punto, por lo que no se puede tener una tasa mayor a ese valor: 1.
2. Al usar valores alto de tasa de aprendizaje, en el caso con S1 = 10, mayores a 0.9, en general, el algoritmo salta de un punto a otro, cuando se encuentra con una curva de contorno, saltándose de un mínimo a otro mínimo. Mientras se encuentra dentro de esos mínimos, el sistema converge a ese mínimo, cambiando muy poco de valor, pero, si por alguna razón, encuentra un salto hacia otro mínimo, lo va a realizar, y se repite el proceso. En el caso de S1 = 2, el sistema converge realizando saltos grandes que van disminuyendo en magnitud cuando se acerca al mínimo, y tiene una trayectoria mas suave en comparación al caso anterior. Y, al llegar al mínimo, empieza a converger a ese mínimo, sin saltarse de un mínimo a otro, como con S1 = 10.
3. En general, con valores menores a 0.9, mayores a 0.5, y no valores debajo de 0.1, el sistema converge realizando grandes saltos de manera escalonada, los cuales se van disminuyendo en magnitud hasta llegar al mínimo, y después de llegar cerca de un mínimo, procede a converger a ese mínimo.
4. Al tener valores menores a 0.5, pero mayores a 0.1, el sistema converge realizando, primero, como saltos grandes, los cuales van bajando en magnitud, pero no de forma escalonada, sino que crean una curva suave de trayectoria. Y al llegar al mínimo, este procede a
5. Al tener un valor debajo de 0.1, el sistema converge lentamente hacia un mínimo, realizando una trayectoria suave al llegar al mínimo, convergiendo sin problema a ese punto.
6. En todos los casos anteriores, excepto en el 2, no hubo diferencia entre S1 = 2 y S1 = 10.

Se desea usar la red de la Figura



para aproximar la función (4.1) Los parámetros de red iniciales se eligen para ser

w1(0) = [−0.27 − 0.41]T **w2(0) = [0.09 − 0.17]T**

b1(0) = [−0.48 − 0.13]T **b2(0) = [−0.48]**

Para crear el conjunto de entrenamiento, se muestrea la función en los puntos p = 1 y p = 0. Encuentre la matriz Jacobiana y su valor para el primer paso del algoritmo de Levenberger Marquandt. (0.25) 4.3. Escriba un programa que implemente la version de Levenberger-Marquardt. Experimente con las mismas condiciones iniciales y el mismo conjunto de entrenamiento de 4.1). Analice las propiedades de convergencia del algoritmo y compare con lo obtenido en 4.1) (1) Primero, se tiene que el vector x este compuesto por lo siguiente:

$$x^T = [-0.27 \quad -0.41 \quad -0.48 \quad 0.13 \quad 0.09 \quad -0.17 \quad -0.48] \ \ (24)$$

Y el error cuadrático del primer paso es el siguiente:

$$Error_1 = 1.3001 \quad (25)$$

Ya que no se alcanza a mostrar todo el cálculo en una página, la matriz Jacobiana del primero paso es la siguiente (en este caso, se utilizó un uk de 0.01):

$$J = \begin{bmatrix} -0.0063 & -0.0081 & -0.0196 & -0.0213 & 0.5137 & 0.5251 & -1 \\ 0.0146 & 0.0198 & 0.0395 & 0.0423 & 0.5137 & 0.5251 & 0.0423 \end{bmatrix} \ \ (26)$$

Y el delta x que se obtiene es el siguiente:

$$\Delta x^T = [-0.27 \quad -0.41 \quad -0.48 \quad -0.13 \quad 0.09 \quad -0.17 \quad -0.48] \ \ (27)$$

Y el nuevo vector x es el siguiente:

$$x_{nuevo}^T = [-0.2911 \quad -0.4388 \quad -0.5388 \quad -0.1905 \quad -0.8748 \quad -1.1561 \quad -0.2488] \ \ (28)$$

Y el error cuadrático del nuevo paso, con este nuevo vector x, es el siguiente:

$$Error_2 = 2.6292 \quad (29)$$

Y la diferencia entre los dos errores es el siguiente:

$$\Delta Error = 1.3291 \quad (30)$$

El cual significa que el error es más grande en comparación al primer error. Entonces uk tendría que ser multiplicado por algún factor y realizar de nuevo el cálculo del jacobiano.