Control Inteligente

Unidad 3: Redes neuronales artificiales y modelos de entrenamiento

SubTemas

2

Tema 3.1 Redes neuronales artificiales

- **Introducción**

Se estima que el sistema nervioso de un ser humano contiene alrededor de cien mil millones de neuronas, organizadas mediante una red compleja en la que las neuronas individuales pueden estar conectadas a varios miles de neuronas distintas.

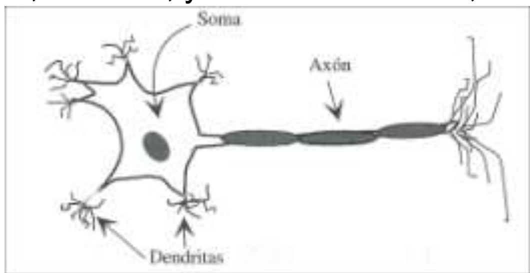
Se calcula que una neurona del córtex cerebral recibe información, por término medio, de unas 10,000 neuronas, y envía impulsos a varios cientos de ellas.

3

Tema 3.1 Redes neuronales artificiales

- **Estructura**

Desde un punto de vista funcional, las neuronas constituyen procesadores de información sencillos. Como todo sistema de este tipo, posee un canal de entrada de información, las **dendritas**; un órgano de cómputo, el **soma**; y un canal de salida, el **axón**.



4

Tema 3.1 Redes neuronales artificiales

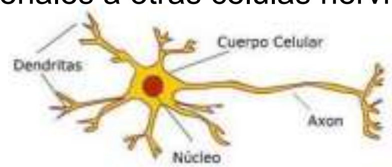
- **Sinapsis**

La mayor parte de las neuronas poseen una estructura de árbol llamada desdirías de tal forma que las neuronas se unen a través de uniones denominadas sinapsis. El cuerpo de la neurona o soma contiene el núcleo. Se encarga de todas las actividades metabólicas de la neurona y recibe la información de otras neuronas vecinas a través de las conexiones sinápticas, de hecho algunas neuronas se comunican sólo con las cercanas, mientras que otras se conectan con miles.

Tema 3.1 Redes neuronales artificiales

- **Función de elementos de una neurona**

Las dendritas, parten del soma y tienen ramificaciones. Se encargan de la recepción de señales de las otras células a través de conexiones sinápticas. Si pensamos en términos electrónicos se puede decir que las dendritas son las conexiones de *entrada* de la neurona. Por su parte el axón es la *salida* de la neurona y se utiliza para enviar impulsos o señales a otras células nerviosas.



Tema 3.1 Redes neuronales artificiales

- **Transmisión de información**

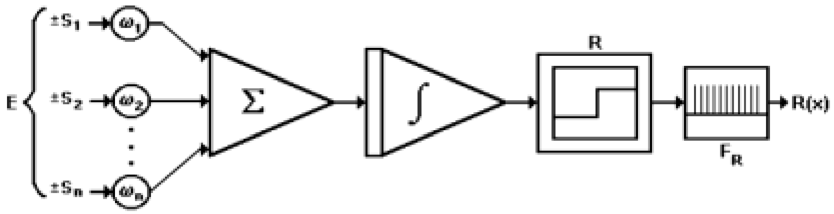
La transmisión de una señal de una célula a otra por medio de la sinapsis es un proceso químico. El efecto es elevar o disminuir el **potencial eléctrico** dentro del cuerpo de la célula receptora. Si su potencial alcanza el umbral se envía un pulso por el axón. Se dice, entonces, que la célula se disparó.

7

Tema 3.1 Redes neuronales artificiales

- **Modelo electrónico de una neurona**

Un modelo de neurona electrónica que permite trasladar el concepto de las neuronas biológicas en neuronas artificiales presenta tres componentes principales: un módulo sumador, un módulo integrador y otro de generación de respuesta.



8

Tema 3.1 Redes neuronales artificiales

- **Estado del arte (Artículos)**

Inspirados en los sistemas nerviosos, muchos investigadores, especialmente aquellos dedicados a modelar el cerebro, han estado explorando diferentes alternativas para modelar las redes neuronales artificiales.

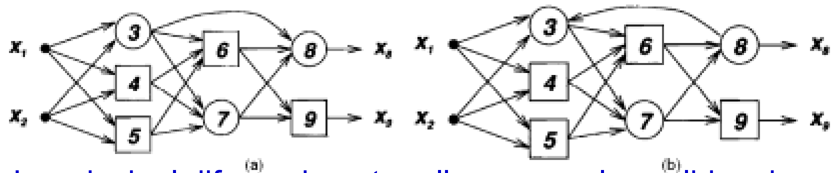
Se ha modelado el cerebro como un sistema dinámico no lineal, continuo en el tiempo, cuyas diferentes arquitecturas buscan emular los mecanismos del cerebro para simular un comportamiento inteligente.

TAREA1: En equipos de 3, hacer un resumen en español de un artículo relacionado a las RNA

Tema 3.1 Redes neuronales artificiales

- **Clasificación de las RNA**

Por lo tanto, existe una gran variedad de enfoques de arquitectura, una de las clasificaciones de las redes neuronales es acerca de su organización, por un lado existen las redes neuronales hacia adelante y las redes neuronales recursivas.

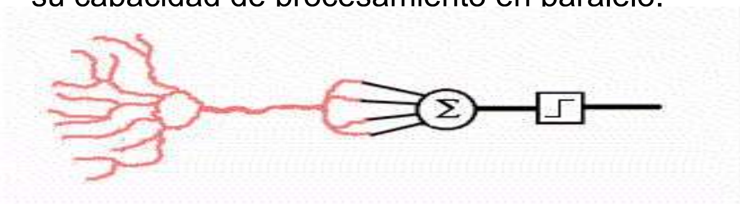


La principal diferencia entre ellas es que las salidas de las primeras no regresan a una capa anterior (a); mientras que en las recursivas si lo hacen (b).

Tema 3.1 Redes neuronales artificiales

- **Otra clasificación**

Por otro lado, cuando las redes neuronales son desarrolladas a nivel software o en sistemas basados en un microcontrolador, se clasifican como digitales; así mismo, cuando la implementación se lleva a cabo con circuitos amplificadores o discretos, se dice que son analógicas. La ventaja de estas últimas radica en su capacidad de procesamiento en paralelo.



11

Tema 3.1 Redes neuronales artificiales

- **Mas estado del arte**

En la literatura se pueden encontrar diferentes modelos matemáticos de redes neuronales, como las redes neuronales con retropropagación a través del tiempo, perceptrón multicapa (MLP, por sus siglas en inglés) Redes neuronales **Nv** [de Mark Tilden], Redes neuronales celulares [CNN], de estas últimas se tiene la ventaja de que pueden ser implementadas usando amplificadores operacionales lo que permite que su implementación sea de bajo costo.

12

Tema 3.1 Redes neuronales artificiales

- **Redes Neuronales Celulares (CNN)**

Las Redes Neuronales Celulares son concebidas en el Laboratorio de la Universidad de California en Berkeley por Leon Chua en el año de 1987; pero es hasta el año de 1988 cuando se publica su trabajo de manera conjunta con Lin Yang con un par de artículos en los que presentan la teoría así como las primeras aplicaciones.

El principal campo de aplicación de las CNN ha sido, desde su definición, el procesamiento de imágenes y el reconocimiento de patrones; pero también se han utilizado para el control de postura y movimiento de robots biológicamente inspirados.

13

Tema 3.1 Redes neuronales artificiales

- **Modelo matemático de las CNN**

La unidad básica de un red neuronal celular es llamada célula (o celda). Contiene elementos lineales (capacitor, resistencias) y no lineales (fuentes de corriente) y fuentes independientes. La Red Neuronal Celular definida por Chua y Yang consiste en usar circuitos dinámicos no lineales, localmente interconectados e idénticos. Usando un arreglo de 2 dimensiones, en una capa simple, entonces matemáticamente esta definición se escribe como:

$$\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{kl\in N_r(ij)} \hat{A}_{ij;kl}(y_{kl}(t), y_{ij}(t)) + \sum_{kl\in N_r(ij)} \hat{B}_{ij;kl}(u_{kl}(t), u_{ij}(t)) + I_{ij}$$

14

Tema 3.1 Redes neuronales artificiales

- **Modelo matemático de las CNN**

La función de salida es:

$$y_{ij} = f(x_{ij})$$

Donde $f(x_{ij})$,es una función de salida no lineal, llamada también función de umbral. Existen diferentes funciones de salida no lineales: sigmoide, unitaria, gaussiana, etc. Se dice que si para cada neurona B=0, entonces la CNN es autónoma.

Una de las aplicaciones de las CNN es el llevar a cabo la función de un Generador Central de Patrones (GCP). En la siguiente sección se explora este concepto.

15

Tema 3.1 Redes neuronales artificiales

- **Generador Central de Patrones**

Los movimientos realizados por animales cuando realizan actividades tales como caminar, volar, correr, nadar, etc.; se llevan a cabo empleando patrones periódicos en sus extremidades.

La hipótesis relacionada propone la existencia de un Generador Central de Patrones (GCP) que se encarga de realizar estos patrones.

Estudios realizados sobre cómo los animales realizan sus movimientos, revelan que el patrón de actividad locomotora se debe a un patrón de actividad neuronal

16

Tema 3.1 Redes neuronales artificiales

- **Generador Central de Patrones**

El principal componente del sistema motriz es el GCP, un circuito neuronal que produce un patrón motriz rítmico sin necesidad de sensores que retroalimenten o controlen. Se localiza generalmente en la espina cordal en los vertebrados o en los ganglios en los invertebrados.

Tema 3.1 Redes neuronales artificiales

- **Generador Central de Patrones**

El GCP incluye el mecanismo neuronal necesario para la generación rítmica coordinada neuro-motora y por lo tanto de la salida del sistema, es decir los músculos. La salida del GCP controla directamente los órganos efectores (piernas, brazos, dedos, etc.) mientras que las señales que recibe del control neuronal superior sólo son necesarias para iniciar el movimiento, pero no para generar el patrón correcto de movimiento.

Tema 3.1 Redes neuronales artificiales

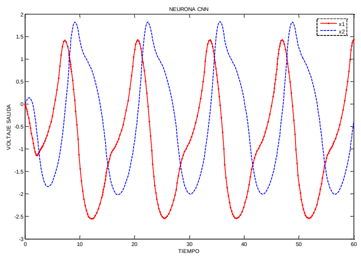
- **Modelo matemático simplificado**

Del modelo de Chua, Paolo Arena y Luigi Fortuna, agregan las siguientes condiciones, con el fin de simplificar las ecuaciones manejando menor número de variables:

$$\frac{dx_1}{dt} = -x_1 + \alpha y_1 - \beta y_2 + i_1$$

$$\frac{dx_2}{dt} = -x_2 + \alpha y_2 + \beta y_1 + i_2$$

$$y_i = \frac{1}{2}\left(\left|x_i + 1\right| - \left|x_i - 1\right|\right)$$



19

Tema 3.1 Redes neuronales artificiales

- **Inteligencia artificial**

La Inteligencia Artificial es la disciplina que estudia la forma de diseñar procesos que exhiban características que comúnmente se asocian con el comportamiento humano inteligente.

Uno de los modelos que ha surgido para emular el proceso de aprendizaje es la red neuronal artificial.

Las RNA son modelos matemáticos que intentan reproducir el funcionamiento del sistema nervioso. Como todo modelo, realizan una simplificación del sistema real que simulan y toman las características principales del mismo para la resolución de una tarea determinada.

20

Tema 3.1 Redes neuronales artificiales

- **Topología**

La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

Una vez determinada la topología de la red neuronal **es necesario entrenarla**. En la etapa de entrenamiento la red es capaz de aprender relaciones complejas entre entradas y salidas mediante el ajuste de los pesos de las conexiones entre neuronas.

21

Tema 3.1 Redes neuronales artificiales

- **Ajuste de pesos sinápticos**

Widrow y Lehr identifican una cantidad significativa de algoritmos de entrenamiento. La mayoría de éstos utilizan información del **gradiente** de una función de error para ajustar los pesos de las conexiones, y se les llama: algoritmos de gradiente descendente.

Dada una topología fija, el entrenamiento de una red neuronal puede ser visto como un **proceso de optimización** cuyo objetivo es encontrar un conjunto de pesos que minimice el error que produce la red sobre el conjunto de datos de entrenamiento.

22

Tema 3.1 Redes neuronales artificiales

- **Aplicaciones**

Las redes neuronales artificiales han sido aplicadas con éxito en gran cantidad de problemas como por ejemplo reconocimiento de patrones, clasificación, visión, control, predicción, etc.

23

Tema 3.1 Redes neuronales artificiales

- **REGRESIÓN Y OPTIMIZACIÓN**

Antes de iniciar con los conceptos de aplicación de las RNA y sus métodos de aprendizaje, es necesario conocer dos temas: la identificación de sistemas y la optimización de funciones.

Aunque, se aclara que para propósitos de aprendizaje de este curso, éstos temas se repasaran superficialmente.

El método de mínimos cuadrados (LSE, Least Square Estimator) es utilizando para la identificación de sistemas.

24

Tema 3.1 Redes neuronales artificiales

- **Identificación de sistemas**

El problema de determinar un modelo matemático para un sistema desconocido (llamado también target model) por la observación de sus pares de datos de entrada-salida, es llamado identificación de sistemas.

25

Tema 3.1 Redes neuronales artificiales

- **¿Para qué hacer identificación de sistemas?**

Predecir el comportamiento de un sistema (ejemplo: pronóstico del clima)

Explicar interacciones y relaciones entre las entradas y salidas de un sistema (ejemplo: paridad peso – dólar)

Diseñar un controlador basado en el modelo de un sistema (ejemplo: control de una aeronave)

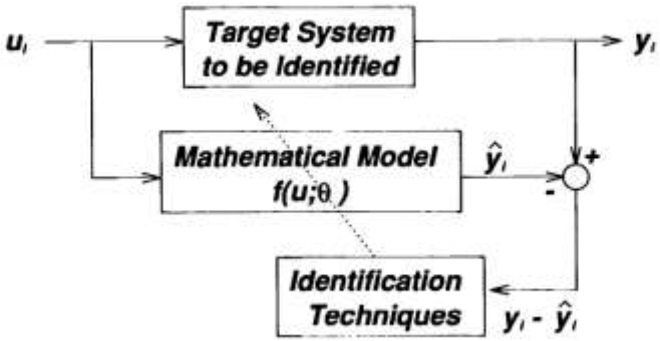
26

Tema 3.1 Redes neuronales artificiales

- **Pasos generales para la identificación de sist.**

a) Identificación estructural

b) Identificación de parámetros



27

Tema 3.1 Redes neuronales artificiales

- **Least Square Estimator**

En el problema general de mínimos cuadrados, la salida de un modelo lineal y, esta dado por la expresión parametrizada linealmente:

$$y = \theta_1 f_1(\mathbf{u}) + \theta_2 f_2(\mathbf{u}) + \cdots + \theta_n f_n(\mathbf{u}),$$

Donde

$\mathbf{u} = [u_1, \cdots, u_p]^T$ es el vector de entrada del modelo

$f_1, \cdots, f_n$ son funciones conocidas de u

$\theta_1, \cdots, \theta_n$ son los parámetros a ser estimados

28

Tema 3.1 Redes neuronales artificiales

- **Regresión lineal (1/3)**

En el estudio de la estadística, el ajuste de curvas utilizando un modelo lineal se le conoce como regresión lineal. Así a la ecuación anterior se le conoce como función de regresión, y a los parámetros θ_i desconocidos, se les llama coeficientes de regresión.

$$\begin{cases} f_1(\mathbf{u}_1)\theta_1 + f_2(\mathbf{u}_1)\theta_2 + \cdots + f_n(\mathbf{u}_1)\theta_n & = & y_1, \\ f_1(\mathbf{u}_2)\theta_1 + f_2(\mathbf{u}_2)\theta_2 + \cdots + f_n(\mathbf{u}_2)\theta_n & = & y_2, \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{u}_m)\theta_1 + f_2(\mathbf{u}_m)\theta_2 + \cdots + f_n(\mathbf{u}_m)\theta_n & = & y_m. \end{cases}$$

Tema 3.1 Redes neuronales artificiales

- **Regresión lineal (2/3)**

Usando notación matricial, podemos reescribir las ecuaciones anteriores en:

$$A\,\theta = y$$

Donde

A es una matriz *mxn* llamada matriz de diseño

θ es un vector *nx1* de parámetros desconocidos

y es un vector *mx1* de salida

Así que la solución debería ser (si A es cuadrada)

$$\theta = A^{-1}\,y$$

Tema 3.1 Redes neuronales artificiales

- **Regresión lineal (3/3)**

Pero normalmente se tienen mas m experimentos que n parámetros a ajustar (“encontrar”); además, existe ruido en las mediciones o el modelo puede no ser apropiado para describir el sistema target. Así que incorporando un vector de error ***e***, para tomar en cuenta ruido aleatorio o errores de modelado:

$$A\,\theta + e = y$$

Deseamos buscar $\theta = \hat{\theta}$ que minimice la suma de los errores al cuadrado:

$$E(\boldsymbol{\theta}) = \sum_{i=1}^{m} (y_i - \mathbf{a}_i^T \boldsymbol{\theta})^2 = \mathbf{e}^T \mathbf{e} = (\mathbf{y} - \mathbf{A}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{A}\boldsymbol{\theta})$$

Tema 3.1 Redes neuronales artificiales

- **Teorema de estimador de mínimos cuadrados**

El error al cuadrado de la ecuación anterior, es el mínimo cuando $\theta = \hat{\theta}$, a esto se le llama LSE (abreviando), el cual satisface la siguiente ecuación

$$A^T A\,\theta = A^T y$$

Si $A^T A$ es no-singular, $\hat{\theta}$ es único, y esta dado por:

$$\hat{\theta} = \underbrace{(A^T A)^{-1} A^T}_{pseudoinversa}\, y$$

32

Tema 3.1 Redes neuronales artificiales

- **Demostración del Teorema de LSE (1/3)**

Para iniciar con el proceso es necesario manejar algunas propiedades de matrices y gradiente de vectores.

$(AB)^T = B^T A^T$ ……(3.1)

$(AB)^{-1} = B^{-1} A^{-1}$ …….(3.2)

$A = A^t$ …….(3.3) Solo si A es simétrica

$\nabla(x^t y) = \nabla(y^t x) = y$ …….(3.4)

$\nabla(x^t A x) = (A + A^t)x$ ……(3.5)

$a = a^T$ ……(3.6) Donde a es un escalar

33

Tema 3.1 Redes neuronales artificiales

- **Demostración del Teorema de LSE (2/3)**

Iniciamos con la definición de la función de Error

$$E(\theta) = (y - A\theta)^T (y - A\theta)$$

$$E(\theta) = y^T - (A\theta)^T (y - A\theta)$$

$$E(\theta) = y^T - \theta^T A^T (y - A\theta)$$

$$E(\theta) = y^T y - y^T A\theta - \theta^T A^T y + \theta^T A^T A\theta$$

El segundo término es un escalar, que debe cumplir con 3.6

$$E(\theta) = y^T y - \theta^T A^T y - \theta^T A^T y + \theta^T A^T A\theta$$

$$E(\theta) = y^T y - 2\theta^T A^T y + \theta^T A^T A\theta$$

Encontraremos ahora la derivada de la función de Error, el primer término es cero

$$E'(\theta) = -2A^T y + \theta^T K\theta$$

Aplicando 3.5 al segundo término (que aún no estaba derivado)

$$E'(\theta) = -2A^T y + [\ A^T A + (A^T A)^T\ ]\theta$$

$$E'(\theta) = -2A^T y + 2A^T A\theta$$

34

Tema 3.1 Redes neuronales artificiales

- **Demostración del Teorema de LSE (3/3)**

Así la derivada del error re-ordenada queda:

$$\frac{\partial E}{\partial \theta} = 2A^T A\theta - 2A^T y$$

Haciendo $\frac{\partial E}{\partial \theta} = 0$ en $\theta = \hat{\theta}$

Se obtiene la ecuación normal

$$A^T A\theta = A^T y$$

Que se resuelve así:

$$\theta = \left(A^T A\right)^{-1} A^T y$$

35

Tema 3.1 Redes neuronales artificiales

- **Tarea**

Obtenga el estimador de mínimo cuadrado ponderado de la ecuación 3.8, directamente haciendo cero la derivada del error ponderado (ecuación 3.9).

$$\hat{\theta}_W = \left(A^T W A\right)^{-1} A^T W y$$

Donde *W*, es una matriz simétrica y definida positiva.

$$E_W(\theta) = (y - A\theta)^T W (y - A\theta)$$

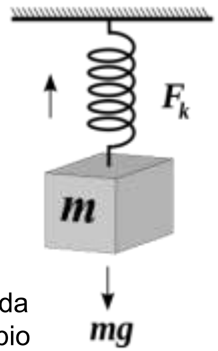
Su justificación radica en que las ecuaciones de las diapositivas anteriores implican que el vector de error tiene el mismo valor, pero W nos ayuda a absorber esta diferencia, cuando W es la matriz Identidad, volvemos a tener las mismas ecuaciones anteriores.

36

Tema 3.1 Redes neuronales artificiales

- **EJEMLO: Ley de Hooke**

Por la ley de Hooke, sabemos que cuando una fuerza es aplicada a un resorte, el cambio en la longitud es proporcional a la fuerza aplicada.



Así tenemos la siguiente relación, entre la longitud l del resorte y la fuerza f aplicada:

$$l = k_0 + k_1 f$$

k_o es la longitud del resorte sin fuerza aplicada
k_1 (constante del resorte) representa el cambio de longitud cuando se aplica una fuerza

37

Tema 3.1 Redes neuronales artificiales

- **Identificando k_0 y k_1**

Aplicamos varios fuerzas diferentes y medimos las correspondientes longitudes del resorte, ver tabla:

Experiment	Force (newtons)	Length of Spring (inches)
1	1.1	1.5
2	1.9	2.1
3	3.2	2.5
4	4.4	3.3
5	5.9	4.1
6	7.4	4.6
7	9.2	5.0

Intenta colocar la información en la forma

$A\,\theta + e = y$

38

Tema 3.1 Redes neuronales artificiales

- **Aplicando el LSE**

Aplicamos

$$\hat{\theta} = (A^T A)^{-1} A^T y$$

$$\begin{bmatrix} \hat{k}_0 \\ \hat{k}_1 \end{bmatrix} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y} = \begin{bmatrix} 1.20 \\ 0.44 \end{bmatrix}$$

39

Tema 3.1 Redes neuronales artificiales

- **Regresión con Matlab**

Comando polyfit (Ajuste de curva polinomial) - ver help

```
x = 0: 0.1: 2.5;
y = erf(x);
p = polyfit(x,y,6);
x = 0: 0.1: 5;
y = erf(x);
f = polyval(p,x);
plot(x,y,'o',x,f,'-')
axis([0 5 0 2])
```

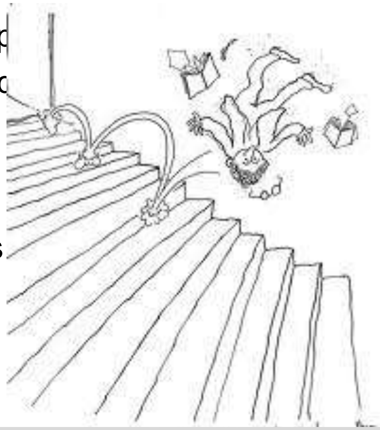
40

Tema 3.1 Redes neuronales artificiales

- **Optimización de funciones**

Podemos clasificarla en dos tip

- Métodos basados en derivac
  - Gradiente descendente
  - Newton - Raphson
  - Gauss-Newton
- Métodos libres de derivadas
  - Algoritmos genéticos
  - Simulated Annealing
  - Downhill simplex search
  - Random Search

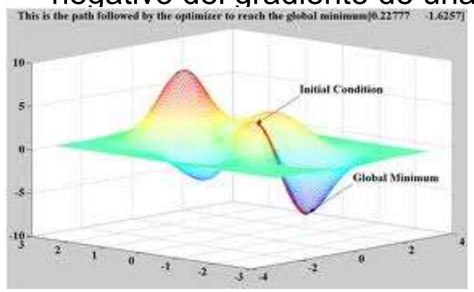


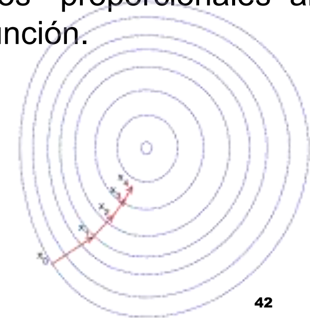
Just after learning the "Steepest Descent" method in optimization class …(Peña class)

Tema 3.1 Redes neuronales artificiales

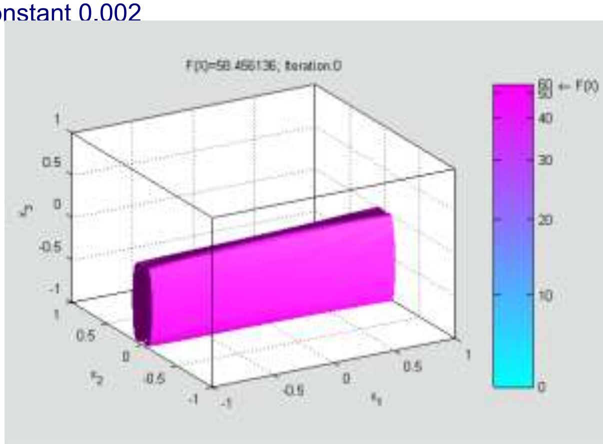
- **Método del descenso más rápido**

El Steepest descent method, también llamado gradiente descendente, es un algoritmo de optimización de primer orden. Para localizar el mínimo local de una función, se deben tomar "pasos" proporcionales al negativo del gradiente de una función.





This animation shows the first 83 iterations of a gradient decent algorithm for a 3-variable function F(X). Each arrow shows the current direction of the gradient, and each surface is an isosurface (cross-section) of F(X) at its current value. The step-length gamma is a constant 0.002



43

Tema 3.1 Redes neuronales artificiales

- **Contexto con las RNA**

El aprendizaje puede ser visto como el proceso de ajuste de los parámetros libres de la red. La mayoría de los métodos de entrenamiento consisten en proponer una función de error que mida el rendimiento actual de la red en función de los pesos sinápticos (los cuales inician con valores aleatorios).

El proceso de aprendizaje es un proceso iterativo, en el cual se va refinando la solución hasta alcanzar un nivel de operación suficientemente bueno.

44

Tema 3.1 Redes neuronales artificiales

- **Método del gradiente descendente**

El método de entrenamiento más utilizado es el método del gradiente descendente. Este método define una función E(W) que proporciona el error que comete la red en función del conjunto de pesos sinápticos W.

El objetivo del aprendizaje será encontrar la configuración de pesos que corresponda al mínimo global de la función de error, aunque en muchos casos es suficiente encontrar un mínimo local lo suficientemente bueno.

45

Tema 3.1 Redes neuronales artificiales

- **Paso de iteración**

Debido a la complejidad de E, se utilizan algoritmos iterativos, en los métodos descendentes el siguiente punto θ_{next} se determina por la siguiente ecuación:

$$\theta_{next} = \theta_{now} + \eta d$$

η = velocidad de aprendizaje ó tamaño de paso (positivo).

d = vector de dirección

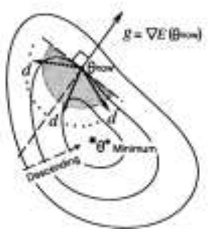
46

Tema 3.1 Redes neuronales artificiales

- **Métodos basados en el gradiente**

Cuando la dirección directa cuesta abajo **d** se determina en la base del gradiente (g) de una función objetivo E, tales métodos se llaman “basados en el gradiente”.

$$g(\theta)(=\nabla E(\theta)) \stackrel{def}{=} \left[\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}, \ldots, \frac{\partial E(\theta)}{\partial \theta_n} \right]^T$$



Cuando d=-g, entonces d es la dirección del descenso más rápido.

47

Tema 3.1 Redes neuronales artificiales

- **Método del descenso más rápido**

La ecuación queda:

$$\theta_{next} = \theta_{now} - \eta g$$

Donde g es la evaluación de la derivada de E, en el punto θ_{now}

$$g = \nabla E(\theta_{now})$$

48

Tema 3.1 Redes neuronales artificiales

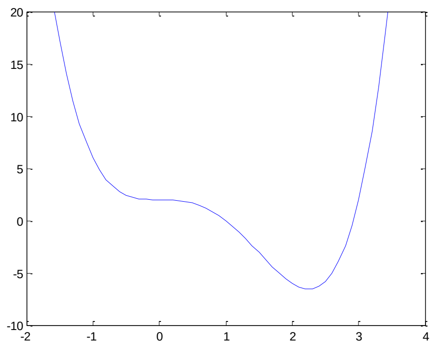
- **Ejemplo**

Encontrar el mínimo de la función $y = x^4 - 3x^3 + 2$

$$E = x^4 - 3x^3 + 2$$

$$\nabla E = 4x^3 - 9x^2$$

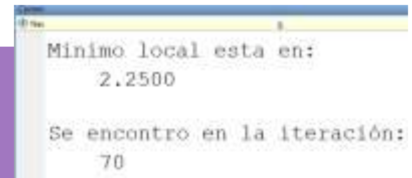
```
x=-5:0.1:5;
y=x.^4-3*x.^3+2;
plot(x,y)
axis([-2 4 -10 20])
```



49

Tema 3.1 Redes neuronales artificiales

- **Código en matlab**



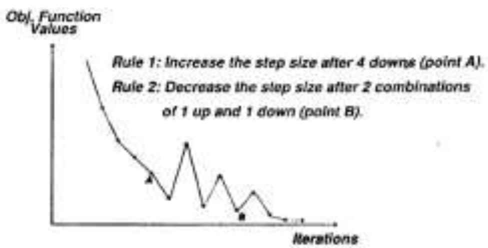

```
Tnow=0;
Tnext=6;%iniciamos en 6
n=0.01; %step size
precision=0.00001;
iteraciones=0;
while abs(Tnext-Tnow) > precision
  Tnow=Tnext;
  DeltaE= 4*Tnow^3 - 9*Tnow^2;
  Tnext=Tnow - n * DeltaE;
  iteraciones=iteraciones+1;
end
disp('Minimo local esta en: ');
disp(Tnext)
disp('Se encontro en la iteración: ');
disp(iteraciones)
```

50

Tema 3.1 Redes neuronales artificiales

- **Limitaciones**

En algunos casos, el método del gradiente descendente es relativamente lento, entre mas se acerca al mínimo: técnicamente su razón de convergencia asintótica es inferior a muchos otros métodos, como los de **Newton**. Así mismo, tiende a tener zigzags.



51