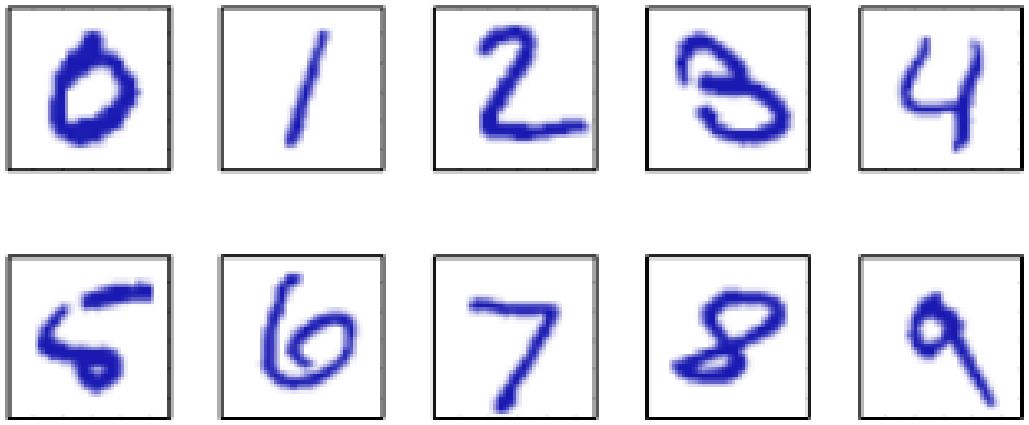
Clasificación

03 de Junio del 2020

Clasificación




HOMERO ESTUVISTE TODA LA NOCHE
CLASIFICANDO TWEETS?
CREO QUE ESTOY CIEGO

Aprendizaje Automático o Machine Learning

El objetivo del machine learning es crear un algoritmo que aprenda un modelo que nos permita resolver una tarea dada. Este modelo es entrenado usando gran cantidad de datos. El modelo aprende de estos datos y es capaz de hacer predicciones sobre datos nuevos.

Problemas más comunes:

- Clasificación
- Clustering
- Regresión

Tipos de Clasificación

Supervisada : se aprende desde datos etiquetados. Ej: Clasificar. dígitos.

No Supervisada : se aprende desde datos no etiquetados. Ej: Agrupar libros similares.

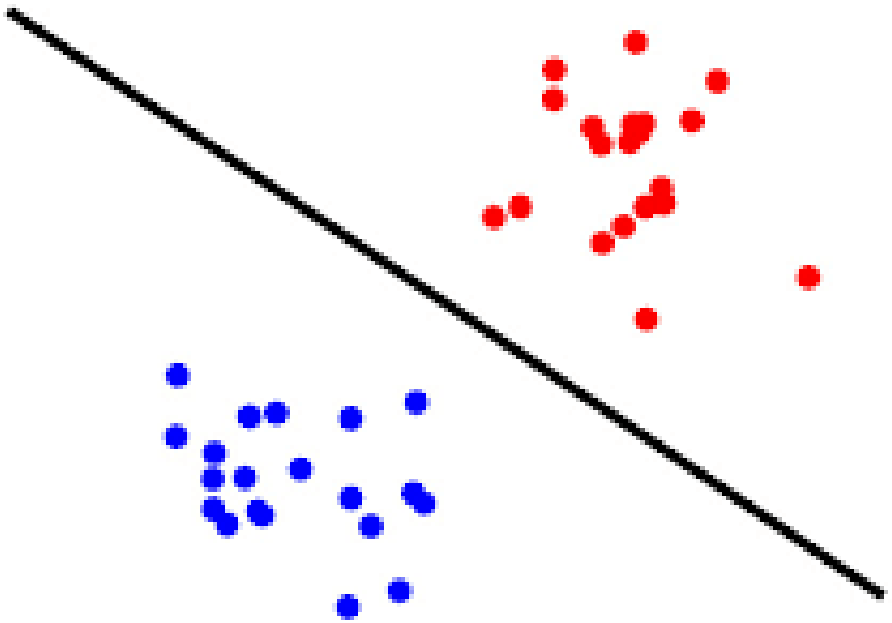
Por refuerzo : el objetivo es maximizar una recompensa. No tenemos datos etiquetados porque hay muchos caminos posibles. Ej: IA para juegos, sistemas de control.

Clasificación: Formulación Matemática

- El objetivo es: dado un vector $\vec{x}$ asignarlo a una de K clases discretas C_k donde $k = 1, 2 \ldots, K$.
- Se espera que las clases sean disjuntas, para que cada entrada sea solo asiganada a una clase.
- El espacio de entrada es dividido en regiones de decisión y las divisiones entre regiones se llaman fronteras de decisión o superficies de decisión.
- El vector objetivo $\vec{t}$ tiene K dimensiones y todos sus dimensiones son 0, excepto la dimensión a la que perteneca la entrada x. Dicha dimesión vale 1. Se lo conoce como one-hot-encoded.

Clasificación Binaria

Encontrar un hiperplano que divida las muestras en dos, las rojas y las azules.



Conjunto de Entrenamiento o Dataset

Para entrenar los modelos necesitamos muestras o datos de entrenamiento.
Normalmente dividimos los datos de entrenamiento en tres:

- Train o entrenamiento
- Validation o validación
- Test o prueba

El conjunto de entrenamiento de utiliza para ajustar los parámetros de configuración del algoritmo de aprendizaje (además del modelo). Estos parámetros se ajustan para lograr el mejor comportamiento en el conjunto de validación. Una vez ajustados estos parámetros de configuración del algoritmo, son utilizados para aprender el modelo final el cual se aplica sobre los datos de prueba.

Función discriminante

- Para resolver el problema vamos a encontrar una función discriminante.
- El caso más simple, cuando las muestras son linealmente separables tiene la siguiente forma:

$$y(\vec{x}) = \vec{w}^T \vec{x} + \omega_0,$$

donde $\vec{w}$ es el peso y ω_0 el bias.

- Muchas veces necesitamos usar funciones no lineales para mejorar la clasificación. Por ejemplo que la salida represente probabilidades.
- Por lo tanto usamos un modelo lineal generalizado:

$$y(\vec{x}) = f\left(\vec{w}^T \vec{x} + \omega_0\right),$$

donde $f(.)$ es conocida como función de activación.

- La superficie de decisión sigue siendo lineal, porque tiene que cumplir que $y(\vec{x}) = cte$ o $\vec{w}^T \vec{x} + \omega_0$.

Función discriminante

Para clasificar entre 2 clases C_1 y C_2, definimos una función de activación $f(.)$, como por ejemplo el signo:

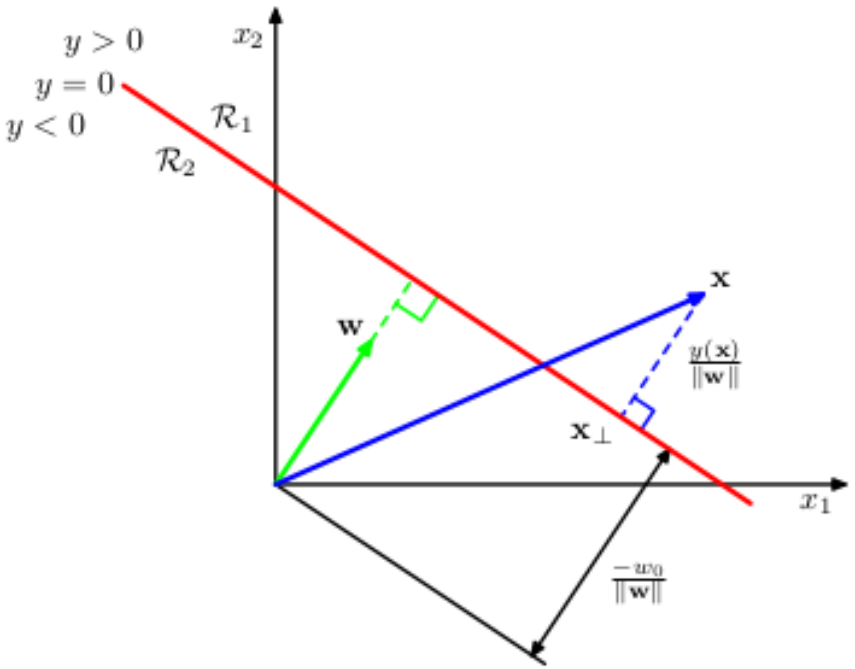
$$\vec{x} \in C_1 \text{ if } y(\vec{x}) \geq 0 \text{ else } \vec{x} \in C_2$$

La región de decisión queda definida por $y(\vec{x}) = 0$, pero puede ser modificada con el término de bias ω_0.

¿Por qué usamos $y(\vec{x})$ como *score* de clasificación?
Vamos a calcular la distancia de un punto a la recta.

Ejemplo en 2 dimensiones

Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.



Dirección de w

Sean dos puntos $\vec{x_A}$ y $\vec{x_B}$ que pertenecen a la recta:

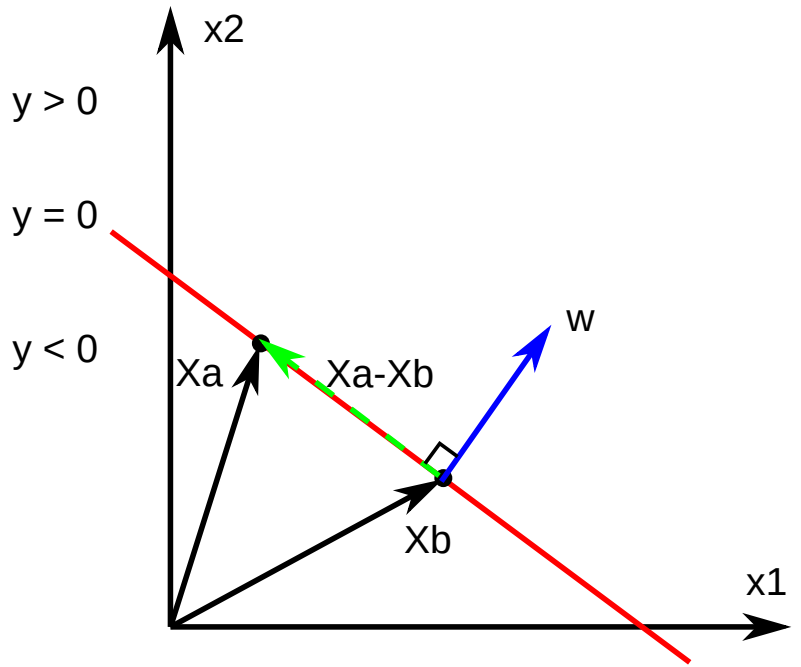
$$y(\vec{x_A}) = y(\vec{x_B}) = 0$$

$$\vec{w}^T \vec{x_A} + \omega_0 = \vec{w^T} \vec{x_B} + \omega_0$$

$$\vec{w}^T (\vec{x_A} - \vec{x_B}) = 0,$$

esto implica que:

$$\vec{w}^T \perp recta$$



Distancia de la recta al origen

Supongamos un punto $\vec{x}$ que pertenece a la recta:

$$y(\vec{x}) = \vec{w}^T \vec{x} + \omega_0 = 0,$$

multiplicando por $\vec{w}$

$$\|\vec{w}\|^2 \vec{x} + \vec{w}\omega_0 = 0$$

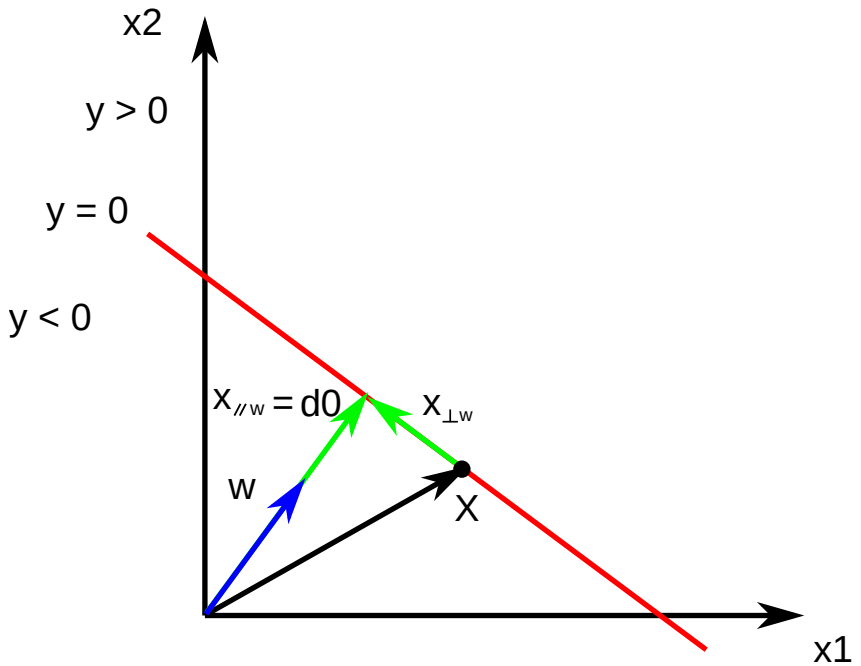
$$\vec{x} = \frac{-\vec{w}\omega_0}{\|\vec{w}\|^2}$$

$$d_0 = proy\ \vec{x}\ en\ \vec{w}$$

$$d_0 = \frac{\vec{w}^T}{\|\vec{w}\|}\vec{x}$$

$$d_0 = \frac{\vec{w}^T}{\|\vec{w}\|}\frac{-\vec{w}\omega_0}{\|\vec{w}\|^2}$$

$$d_0 = \frac{-\omega_0}{\|\vec{w}\|}$$



Distancia de un punto a la recta

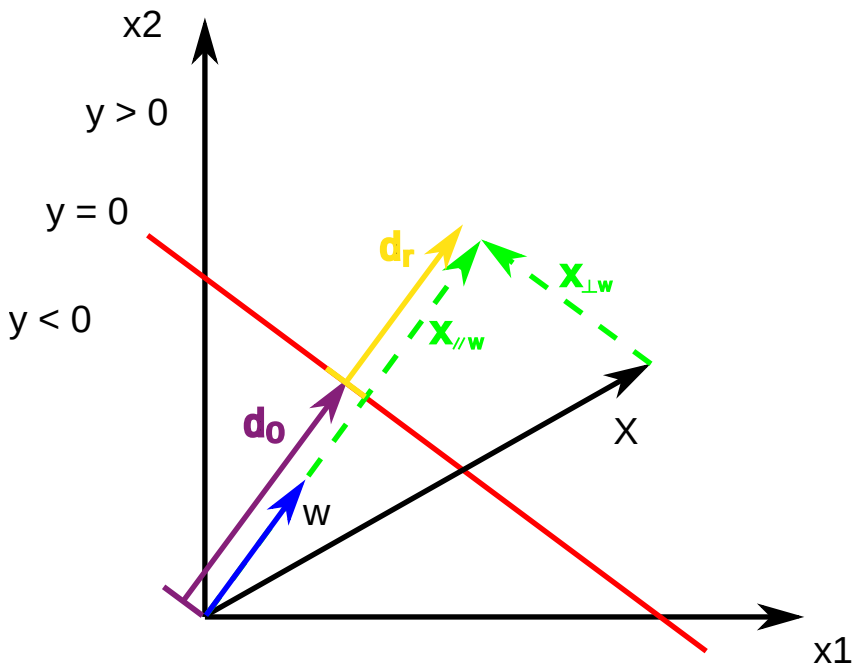
La distancia de un punto $\vec{x}$ a la recta es igual a la proyección del punto $\vec{x}$ sobre $\vec{w}$ menos la distancia de la recta al origen:

$$d_r = \vec{x}_{\|\vec{w}} - d_0$$

$$d_r = \frac{\vec{w}^T}{\|\vec{w}\|}\vec{x} - d_0$$

$$d_r = \frac{\vec{w}^T}{\|\vec{w}\|}\vec{x} - \frac{-\omega_0}{\|\vec{w}\|}$$

$$d_r = \frac{y(\vec{x})}{\|\vec{w}\|},$$



por lo tanto $y(\vec{x})$ es proporcional a la distancia a la recta de un punto determinado y **puede ser usado para clasificar una muestra**.

Regresión Logística

Regresión logística

- Muchas veces necesitamos una salida probabilística, que esté entre 0 y 1.
- Para eso podemos una función de activación conocida como sigmoide.
- Este modelo de clasificación recibe el nombre de regresión logística.

Sigmoide

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

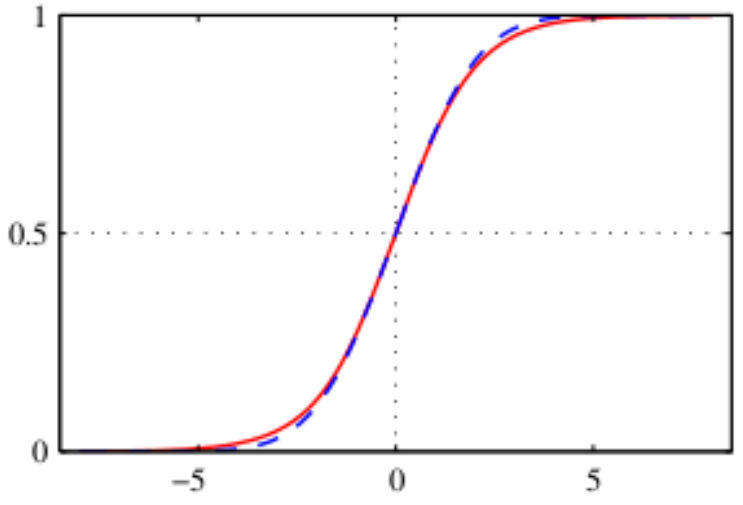


Figure 4.9 Plot of the logistic sigmoid function $\sigma(a)$ defined by (4.59), shown in red, together with the scaled probit function $\Phi(\lambda a)$, for $\lambda^2 = \pi/8$, shown in dashed blue, where $\Phi(a)$ is defined by (4.114). The scaling factor $\pi/8$ is chosen so that the derivatives of the two curves are equal for $a = 0$.

Un pequeño cambio de notación

Expresemos nuestro discriminante inicial usando $\vec{w}'$ y ω_0':

$$\vec{w}'^T \vec{x} + \omega_0'$$

y definamos nuevos vectores $\vec{x}$ y $\vec{w}$ ampliados:

$$\vec{x}^T = [\vec{x}', 1]$$
$$\vec{w}^T = [\vec{w}', \omega_0'].$$

Ahora el discriminante nos queda:

$$\vec{w}'^T \vec{x} + \omega_0' = \vec{w}^T \vec{x}$$

Modelo de regresión

Supongamos que queremos clasificar entre dos categorías C_1 y C_2 (dicotomizador). La probabilidad de que una muestra $\vec{x}$ pertenezca a la clase 1 la podemos expresar como:

$$p(C_1|\vec{x}) = y(\vec{x}) = \sigma(\vec{w}^T\vec{x}) = \frac{1}{1+e^{-\vec{w}^T\vec{x}}}$$

y de que pertenezca a la clase 2:

$$p(C_2|\vec{x}) = 1 - p(C_1|\vec{x}) = 1 - y(\vec{x}) = 1 - \sigma(\vec{w}^T\vec{x}) = \frac{1}{1+e^{\vec{w}^T\vec{x}}}$$

Estimando $\vec{w}$ desde un dataset

Nuestro objetivo es obtener un w que nos permita separar entre ambas clases. Supongamos que tenemos un conjunto de N muestras de entrenamiento $\vec{x_n}$ con sus correspondientes etiquetas o anotaciones t_n.

$$DS = \{(\vec{x}_n, t_n), \quad n = 1, \ldots, N\}, \quad t_n \in \{0, 1\}$$

Para eso definamos una función de verosimilitud, o sea una medida que indique cuan probable son los datos en función de las etiquetas t_n y el peso $\vec{w}$. Luego con esa función, vamos a definir un costo o medida de error y por último usando el gradiente vamos a encontrar un w que minimize ese error.

$$p(\bar{t}, \vec{w}) = \prod_{n=1}^{N} y_n^{t_n} \left(1 - y_n\right)^{1-t_n}$$

Función de verosimilitud

$$p(\bar{t}, \vec{w}) = \prod_{n=1}^{N} y_n^{t_n} \left(1 - y_n\right)^{1-t_n}$$

La función de verosimilitud aumenta si una muestra está bien clasificada y disminuye si está mal clasificada.

Supongamos una muestra cuyo t_n es 1:

$$y_n^{t_n} \left(1 - y_n\right)^{1-t_n} = y_n^1 \left(1 - y_n\right)^{1-1} = y_n$$

para que la verosimilitud se incremente necesitamos que y_n sea alto (cercano a 1).

Supongamos una muestra cuyo t_n es 0:

$$y_n^{t_n} \left(1 - y_n\right)^{1-t_n} = y_n^0 \left(1 - y_n\right)^{1-0} = 1 - y_n$$

para que la verosimilitud se incremente necesitamos que y_n sea bajo (cercano a 0).

Función de error

No se trabaja con funciones de verosimilitud, sino que se trabaja con funciones de error.

Como función de error elegimos el negativo del logaritmo de la verosimilitud, también llamada entropía cruzada:

$$E(\vec{w}) = -\log\left[p(\bar{t}, \vec{w})\right] = -\sum_{n=1}^{N}\left[t_n \log(y_n) + (1 - t_n)\log\left(1 - y_n\right)\right]$$

$$E(\vec{w}) = -\sum_{n=1}^{N}\left\{t_n \log\left[\sigma\left(\vec{w}^T\vec{x}_n\right)\right] + (1 - t_n)\log\left[1 - \sigma\left(\vec{w}^T\vec{x}_n\right)\right]\right\}$$

Recta Tangente o Dirección del Gradiente

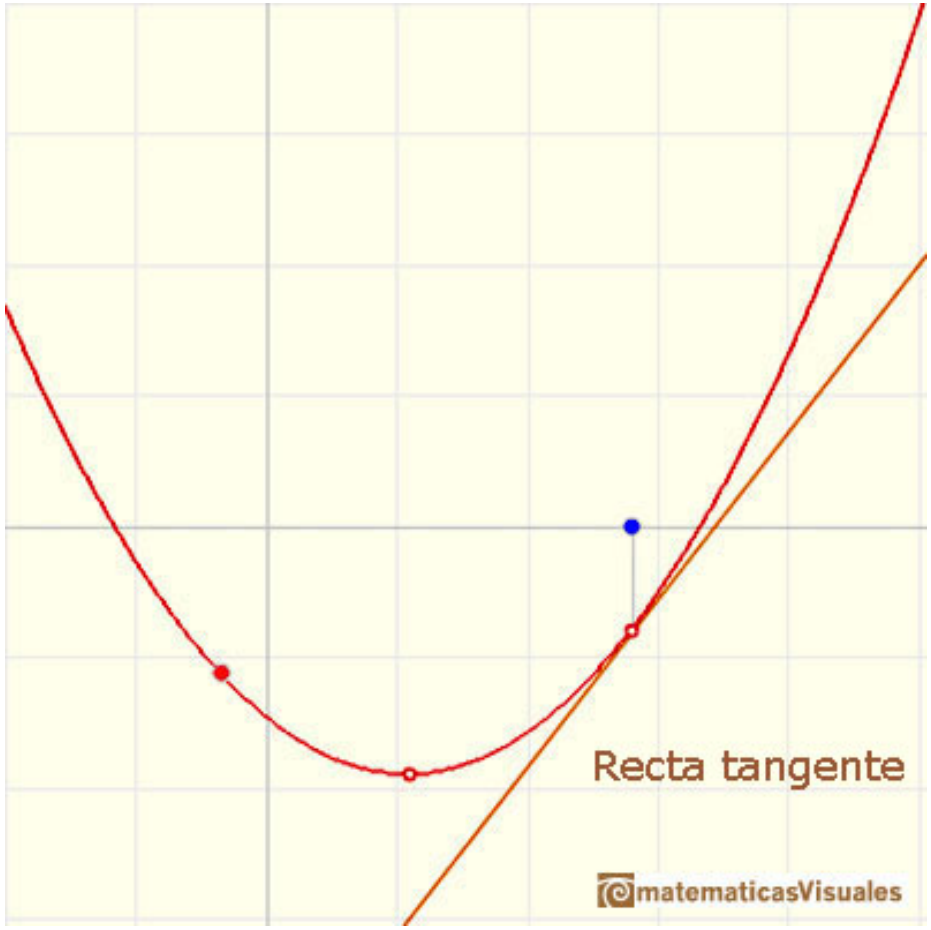
$$y = x^2$$

$$\nabla_x y = \frac{dy}{dx}$$

$$\nabla_x y = 2x$$

$$\nabla_x y|_{x=1} = 2$$

$$\nabla_x y|_{x=-1} = -2$$



Cómputo del gradiente

Ahora computamos el gradiente de E con respecto a $\vec{w}$:

$$\nabla_w E(\vec{w}) = -\sum_{n=1}^{N} \left[t_n \frac{\sigma\,(1-\sigma)}{\sigma} \vec{x}_n - (1-t_n)\,\frac{\sigma\,(1-\sigma)}{1-\sigma} \vec{x}_n \right]$$

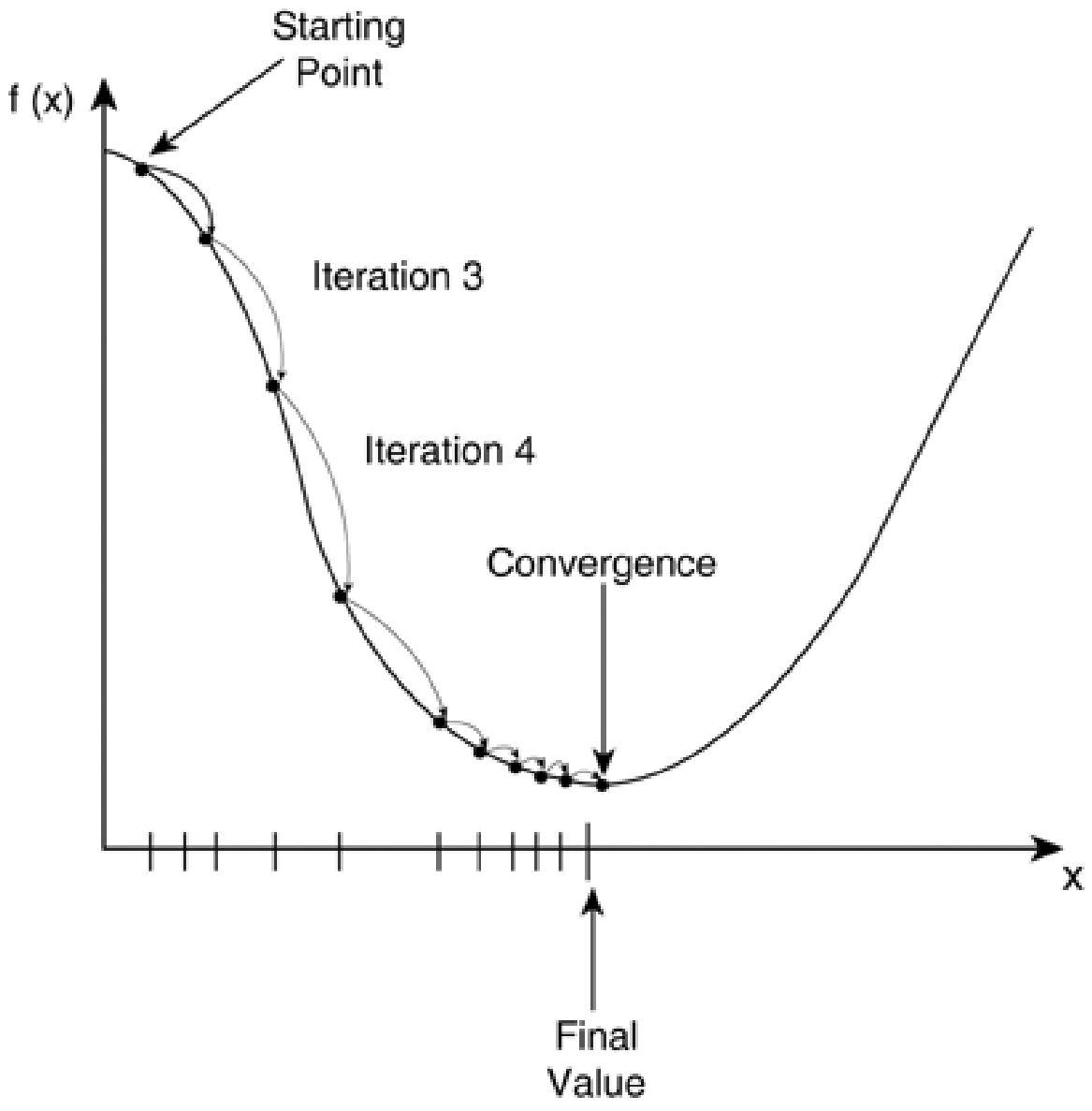
$$\nabla_w E(\vec{w}) = -\sum_{n=1}^{N} \left\{ t_n \left[1 - \sigma\left(\vec{w}^T \vec{x}_n \right) \right] \vec{x}_n - (1-t_n)\,\sigma\left(\vec{w}^T \vec{x}_n \right) \vec{x}_n \right\}$$

$$\nabla_w E(\vec{w}) = -\sum_{n=1}^{N} \left\{ t_n \left[1 - y(\vec{x}_n) \right] \vec{x}_n - (1-t_n)\,y(\vec{x}_n) \vec{x}_n \right\}$$

$$\nabla_w E(\vec{w}) = -\sum_{n=1}^{N} \left\{ t_n \left[1 - y(\vec{x}_n) \right] \vec{x}_n - (1-t_n)\,y(\vec{x}_n) \vec{x}_n \right\}$$

$$\nabla_w E(\vec{w}) = \sum_{n=1}^{N} \left[y(\vec{x}_n) - t_n \right] \vec{x}_n$$

Descenso de gradiente



Descenso de gradiente

Como el gradiente apunta hacia la dirección de máximo crecimiento de la función, en este caso apunta en una dirección en donde se incrementa el error, modificamos w en la dirección opuesta a este gradiente:

$$\vec{w_{s+1}} = \vec{w_s} - \eta \nabla_w E(\vec{w})$$

donde η es conocida como la tasa de aprendizaje. Este algoritmo de optimización es conocido como descenso de gradiente.


That's all Folks!