Clase 15: Redes Neuronales

Mauricio Castro C.
mcastro@mat.uc.cl

Departamento de Estadística, Pontificia Universidad Católica de Chile

TÓPICOS APLICADOS EN ESTADÍSTICA

Segundo Semestre 2022

Redes Neuronales

El cerebro como una red neuronal

- Corteza cerebral: parte más grande del cerebro consistente en una red interconectada de células llamadas neuronas.

El cerebro como una red neuronal

- Corteza cerebral: parte más grande del cerebro consistente en una red interconectada de células llamadas neuronas.

- Neuronas: células nerviosas elementales que forman bloques del sistema nervioso.

El cerebro como una red neuronal

- Corteza cerebral: parte más grande del cerebro consistente en una red interconectada de células llamadas neuronas.

- Neuronas: células nerviosas elementales que forman bloques del sistema nervioso.

- Soma: contiene el núcleo dos tipos de proyecciones, dendritas y axones.

El cerebro como una red neuronal

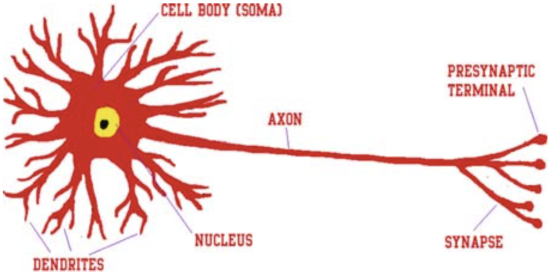
- **Corteza cerebral**: parte más grande del cerebro consistente en una red interconectada de células llamadas **neuronas**.

- **Neuronas**: células nerviosas elementales que forman bloques del sistema nervioso.

- **Soma**: contiene el núcleo dos tipos de proyecciones, **dendritas** y **axones**.

- Cada neurona tiene un axón, el cual termina en una **sinapsis**.

El cerebro como una red neuronal



El cerebro como una red neuronal

- Las neuronas envían señales a otras a través de procesos electroquímicos.

El cerebro como una red neuronal

- Las neuronas envían señales a otras a través de procesos electroquímicos.
- Bajo ciertas condiciones la neurona envía un pulso eléctrico llamado spike.

- Las neuronas envían señales a otras a través de procesos electroquímicos.
- Bajo ciertas condiciones la neurona envía un pulso eléctrico llamado spike.
- Sinapsis inhibitoria: evita que la neurona dispare el impulso.

El cerebro como una red neuronal

- Las neuronas envían señales a otras a través de procesos electroquímicos.
- Bajo ciertas condiciones la neurona envía un pulso eléctrico llamado spike.
- Sinapsis inhibitoria: evita que la neurona dispare el impulso.
- Sinapsis excitatoria: empuja a la neurona a que dispare el impulso.

Modelo McCulloch-Pitts

- Modelo propuesto por McCullogh y Pitts (1943).

- Modelo propuesto por McCullogh y Pitts (1943).

- Multiples inputs $X_1, X_2, \ldots, X_r$ (dendritas) con valores 1 o 0 ("On" o "Off").

Modelo McCulloch-Pitts

- Modelo propuesto por McCullogh y Pitts (1943).
- Multiples inputs $X_1, X_2, \ldots, X_r$ (dendritas) con valores 1 o 0 ("On" o "Off").
- Un solo output y (axón).

Modelo McCulloch-Pitts

- Modelo propuesto por McCullogh y Pitts (1943).

- Multiples inputs $X_1, X_2, \ldots, X_r$ (dendritas) con valores 1 o 0 ("On" o "Off").

- Un solo output y (axón).

- Idea: Replicar de manera artificial y simplificada el comportamiento de las neuronas en el cerebro.

Modelo McCulloch-Pitts

- Excitación total $u = \sum_j x_j$.

Modelo McCulloch-Pitts

- Excitación total $u = \sum_j x_j$.

- Si la sinapsis es no inhibitoria, u es comparado con un valor θ.

Modelo McCulloch-Pitts

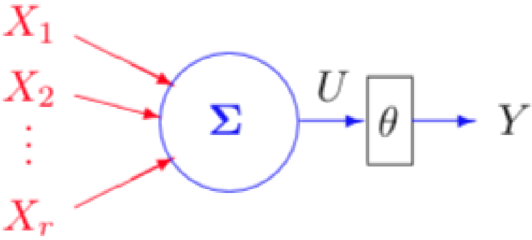
- Excitación total $U = \sum_j X_j$.

- Si la sinapsis es no inhibitoria, U es comparado con un valor θ.

- Si $U \geqslant \theta$, $Y = 1$, es decir, la neurona dispara y transmite una nueva señal.

- Excitación total $U = \sum_j X_j$.
- Si la sinapsis es no inhibitoria, U es comparado con un valor θ.
- Si $U \geqslant \theta$, $Y = 1$, es decir, la neurona dispara y transmite una nueva señal.
- Aqui los inputs son binarios por construcción.

Modelo McCulloch-Pitts



Modelo McCulloch-Pitts

- Si $\theta > r$, la neurona nunca se dispara.

Modelo McCulloch-Pitts

- Si $\theta > r$, la neurona nunca se dispara.
- Esto pues si $X_j = 1$, $\forall j = 1, \ldots, r$, entonces $U = \sum_{j=1}^{r} X_j = r < \theta$.

Modelo McCulloch-Pitts

- Si $\theta > r$, la neurona nunca se dispara.
- Esto pues si $X_j = 1$, $\forall j = 1, \ldots, r$, entonces $U = \sum_{j=1}^{r} X_j = r < \theta$.
- Si $\theta = 0$ entonces $Y = 1$.

Modelo McCulloch-Pitts

- Si $\theta > r$, la neurona nunca se dispara.
- Esto pues si $X_j = 1, \forall j = 1, \ldots, r$, entonces $U = \sum_{j=1}^{r} X_j = r < \theta$.
- Si $\theta = 0$ entonces $Y = 1$.
- Geométricamente lo anterior se describe así:

Modelo McCulloch-Pitts

- Si $\theta > r$, la neurona nunca se dispara.
- Esto pues si $X_j = 1$, $\forall j = 1, \ldots, r$, entonces $U = \sum_{j=1}^{r} X_j = r < \theta$.
- Si $\theta = 0$ entonces $Y = 1$.
- Geométricamente lo anterior se describe así:
  - $X_1, \ldots, X_r$ es un hipercubo de dimensión r.

Modelo McCulloch-Pitts

- Si $\theta > r$, la neurona nunca se dispara.
- Esto pues si $X_j = 1, \forall j = 1, \dots, r$, entonces $U = \sum_{j=1}^{r} X_j = r < \theta$.
- Si $\theta = 0$ entonces $Y = 1$.
- Geométricamente lo anterior se describe así:
  - $X_1, \dots, X_r$ es un hipercubo de dimensión r.
  - Para un valor de θ, el hipercubo se divide de acuerdo al hiperplano $\sum_{j=1}^{r} X_j = \theta$.

- Si $\theta > r$, la neurona nunca se dispara.
- Esto pues si $X_j = 1$, $\forall j = 1, \ldots, r$, entonces $U = \sum_{j=1}^{r} X_j = r < \theta$.
- Si $\theta = 0$ entonces $Y = 1$.
- Geométricamente lo anterior se describe así:
  - $X_1, \ldots, X_r$ es un hipercubo de dimensión r.
  - Para un valor de θ, el hipercubo se divide de acuerdo al hiperplano $\sum_{j=1}^{r} X_j = \theta$.
  - Los vértices con $Y = 1$ quedan a un lado del hiperplano mientras que los vértices $Y = 0$ al otro.

Modelo McCulloch-Pitts

- El modelo de McCulloch-Pitts también es conocido como unidad lógica del umbral.

Modelo McCulloch-Pitts

- El modelo de McCulloch-Pitts también es conocido como unidad lógica del umbral.

- Es utilizado para calcular funciones lógicas simples, como por ejemplo “Y” o “O”.

Modelo McCulloch-Pitts

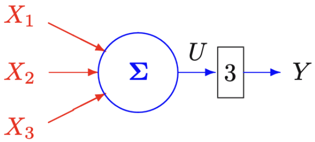
- El modelo de McCulloch-Pitts también es conocido como unidad lógica del umbral.

- Es utilizado para calcular funciones lógicas simples, como por ejemplo "Y" o "O".

- Otras funciones lógicas pueden obtenerse a través de más capas de este modelo.

Modelo McCulloch-Pitts

- El modelo de McCulloch-Pitts también es conocido como unidad lógica del umbral.
- Es utilizado para calcular funciones lógicas simples, como por ejemplo "Y" o "O".
- Otras funciones lógicas pueden obtenerse a través de más capas de este modelo.
- El caso "Y" por ejemplo sugiere que la neurona disparará un impulso si todos los inputs toman el valor 1 por ejemplo.



Teoría de aprendizaje de Hebb

- Donald O. Hebb: The Organization of Behaviour (1949).

Teoría de aprendizaje de Hebb

- Donald O. Hebb: The Organization of Behaviour (1949).

- En este libro se resume como el sistema nervioso central afecta nuestro comportamiento y viceversa.

Teoría de aprendizaje de Hebb

- Donald O. Hebb: The Organization of Behaviour (1949).
- En este libro se resume como el sistema nervioso central afecta nuestro comportamiento y viceversa.
- La teoría de Hebb asume que uno nace con todas las neuronas necesarias para vivir, y que las conexiones iniciales de estas se distribuyen aleatoriamente.

Teoría de aprendizaje de Hebb

- Donald O. Hebb: The Organization of Behaviour (1949).
- En este libro se resume como el sistema nervioso central afecta nuestro comportamiento y viceversa.
- La teoría de Hebb asume que uno nace con todas las neuronas necesarias para vivir, y que las conexiones iniciales de estas se distribuyen aleatoriamente.
- A medida que el ser humano crece, las conexiones neuronales se multiplican y se hacen más fuertes.

Teoría de aprendizaje de Hebb

- La teoría de Hebb también establece que la fuerza de la conexión sináptica entre dos neuronas depende de su historia.

Teoría de aprendizaje de Hebb

- La teoría de Hebb también establece que la fuerza de la conexión sináptica entre dos neuronas depende de su historia.

- Mientras más frecuente sea el disparo entre dos neuronas, mas fuerte será su conexión.

Teoría de aprendizaje de Hebb

- La teoría de Hebb también establece que la fuerza de la conexión sináptica entre dos neuronas depende de su historia.
- Mientras más frecuente sea el disparo entre dos neuronas, mas fuerte será su conexión.
- Lo anterior también se debería cumplir desde el punto de vista de la inhibición.

Teoría de aprendizaje de Hebb

- La teoría de Hebb también establece que la fuerza de la conexión sináptica entre dos neuronas depende de su historia.
- Mientras más frecuente sea el disparo entre dos neuronas, mas fuerte será su conexión.
- Lo anterior también se debería cumplir desde el punto de vista de la inhibición.
- Si una neurona A envia repetidamente una señal a la neurona B y esta no dispara, entonces se reduce las chances que en el futuro A haga que B dispare.

Perceptrones de una capa (single layer)

- El trabajo de Hebb impulsó el trabajo del psicologo Frank Rosenblatt.

Perceptrones de una capa (single layer)

- El trabajo de Hebb impulsó el trabajo del psicologo Frank Rosenblatt.

- Rosenblatt creyó que podía mejorar el trabajo de Hebb construyendo un sistema minimamente restringido llamado perceptrón (1958, 1962).

Perceptrones de una capa (single layer)

- El trabajo de Hebb impulsó el trabajo del psicologo Frank Rosenblatt.
- Rosenblatt creyó que podía mejorar el trabajo de Hebb construyendo un sistema minimamente restringido llamado perceptrón (1958, 1962).
- El perceptrón mejora el modelo de McCulloch-Pitts introduciendo pesos.

- El trabajo de Hebb impulsó el trabajo del psicologo Frank Rosenblatt.

- Rosenblatt creyó que podía mejorar el trabajo de Hebb construyendo un sistema minimamente restringido llamado perceptrón (1958, 1962).

- El perceptrón mejora el modelo de McCulloch-Pitts introduciendo pesos.

- Aquí X_j está asociado a un peso de conección β_j, $j = 1, \ldots, r$.

Perceptrones de una capa (single layer)

- El trabajo de Hebb impulsó el trabajo del psicologo Frank Rosenblatt.
- Rosenblatt creyó que podía mejorar el trabajo de Hebb construyendo un sistema minimamente restringido llamado perceptrón (1958, 1962).
- El perceptrón mejora el modelo de McCulloch-Pitts introduciendo pesos.
- Aquí X_j está asociado a un peso de conección β_j, $j = 1, \ldots, r$.
- Pesos positivos ($\beta_j > 0$) reflejan sinapsis excitatorias y pesos negativos ($\beta_j < 0$) inhibitorias.

Perceptrones de una capa (single layer)

- La magnitud de β_j muestra la fuerza de la conección.

Perceptrones de una capa (single layer)

- La magnitud de β_j muestra la fuerza de la conección.

- Ahora, $U = \sum_{j=1}^{r} \beta_j X_j$, con X_j binaria o real-valorada.

Perceptrones de una capa (single layer)

- La magnitud de β_j muestra la fuerza de la conección.
- Ahora, $U = \sum_{j=1}^{r} \beta_j X_j$, con X_j binaria o real-valorada.
- Al igual que en el modelo de McCulloch-Pitts, $Y = 1$ si $U \geqslant \theta$, donde θ es un valor determinado (umbral), y $Y = 0$ en caso contrario.

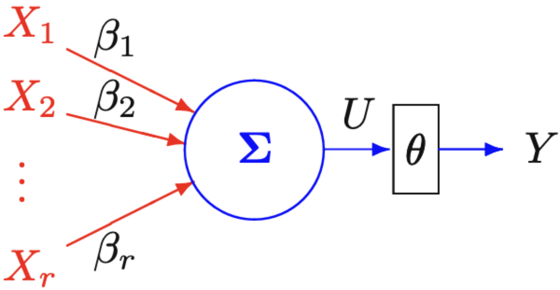
Perceptrones de una capa (single layer)

- La magnitud de β_j muestra la fuerza de la conección.
- Ahora, $U = \sum_{j=1}^{r} \beta_j X_j$, con X_j binaria o real-valorada.
- Al igual que en el modelo de McCulloch-Pitts, $Y = 1$ si $U \geqslant \theta$, donde θ es un valor determinado (umbral), y $Y = 0$ en caso contrario.
- Note que, si $\beta_0 = -\theta$, entonces $U = \sum_{j=0}^{r} \beta_j X_j$ puede ser comparado con 0, donde $X_0 = 1$.

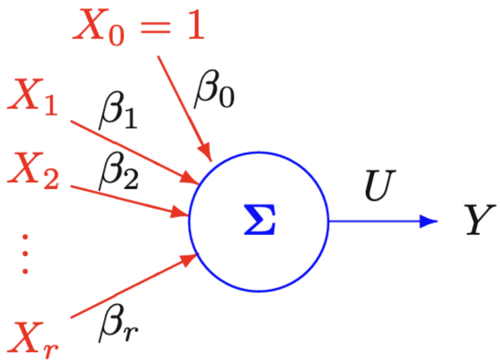
Perceptrones de una capa (single layer)

- La magnitud de β_j muestra la fuerza de la conección.
- Ahora, $U = \sum_{j=1}^{r} \beta_j X_j$, con X_j binaria o real-valorada.
- Al igual que en el modelo de McCulloch-Pitts, $Y = 1$ si $U \geqslant \theta$, donde θ es un valor determinado (umbral), y $Y = 0$ en caso contrario.
- Note que, si $\beta_0 = -\theta$, entonces $U = \sum_{j=0}^{r} \beta_j X_j$ puede ser comparado con 0, donde $X_0 = 1$.
- Aquí si $U \geqslant 0$, $Y = 1$ y $Y = 0$ en otro caso. β_0 se llama elemento de sesgo.

Perceptrones de una capa (single layer)



Perceptrones de una capa (single layer)



Perceptrones de una capa (single layer)

- La función $Y \in \{0, 1\}$ se llama perceptron-calculable.

Perceptrones de una capa (single layer)

- La función $Y \in \{0, 1\}$ se llama perceptron-calculable.

- Para un valor de θ existe un hiperplano que divide el espacio de los inputs en dos, R_0 y R_1, para los que $Y = 0$ y $Y = 1$.

Perceptrones de una capa (single layer)

- La función $Y \in \{0, 1\}$ se llama perceptron-calculable.
- Para un valor de θ existe un hiperplano que divide el espacio de los inputs en dos, R_0 y R_1, para los que $Y = 0$ y $Y = 1$.
- Si los puntos en R_0 pueden ser separados de los de R_1 se dice que el conjunto de puntos es linealmente separable.

Perceptrones de una capa (single layer)

- La función $Y \in \{0, 1\}$ se llama perceptron-calculable.
- Para un valor de θ existe un hiperplano que divide el espacio de los inputs en dos, R_0 y R_1, para los que $Y = 0$ y $Y = 1$.
- Si los puntos en R_0 pueden ser separados de los de R_1 se dice que el conjunto de puntos es linealmente separable.
- Esta partición permite al perceptrón predecir una clase dada.

- Sea el vector aleatorio $\mathbf{X} = (X_1, \ldots, X_r)^\top$ de entradas (inputs).

Funciones de activación

- Sea el vector aleatorio $\mathbf{X} = (X_1, \ldots, X_r)^\top$ de entradas (inputs).
- Dado $\mathbf{X}$, para la ℓ-ésima neurona, se tiene la ℓ-ésima función de activación lineal

Funciones de activación

- Sea el vector aleatorio $\mathbf{X} = (X_1, \ldots, X_r)^\top$ de entradas (inputs).

- Dado $\mathbf{X}$, para la ℓ-ésima neurona, se tiene la ℓ-ésima función de activación lineal

$$U_\ell = \beta_{0\ell} + \mathbf{X}^\top \boldsymbol{\beta}_\ell, \quad \ell = 1, \ldots, s,$$

donde $\beta_{0\ell}$ es la constante o sesgo relacionado al umbral para que la neurona dispare y $\boldsymbol{\beta}_\ell = (\beta_{1\ell}, \cdots, \beta_{r\ell})^\top$ es el vector r-dimensional de pesos.

Funciones de activación

- Sea el vector aleatorio $\mathbf{X} = (X_1, \ldots, X_r)^\top$ de entradas (inputs).

- Dado $\mathbf{X}$, para la ℓ-ésima neurona, se tiene la ℓ-ésima función de activación lineal

$$U_\ell = \beta_{0\ell} + \mathbf{X}^\top \boldsymbol{\beta}_\ell, \quad \ell = 1, \ldots, s,$$

  donde $\beta_{0\ell}$ es la constante o sesgo relacionado al umbral para que la neurona dispare y $\boldsymbol{\beta}_\ell = (\beta_{1\ell}, \cdots, \beta_{r\ell})^\top$ es el vector r-dimensional de pesos.

- De forma matricial, se tiene $\mathbf{U} = \beta_0 + \mathbf{B}\mathbf{X}$, con $\mathbf{U} = (U_1, \cdots, U_s)^\top$, $\boldsymbol{\beta}_0 = (\beta_{01}, \cdots, \beta_{0s})^\top$ un vector s-dimensional de sesgos y $\mathbf{B} = (\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_s)^\top$ una matrix de conexiones de dimensión $(s \times r)$.

Funciones de activación

- La función de activación no lineal será $f(\mathbf{U}) = f(\beta_0 + \mathbf{BX})$, donde $\mathbf{f} = (f, \cdots, f)^\top$ es un vector s-dimensional de funciones cuyos elementos son la función f y $\mathbf{f}(\mathbf{U}) = (f(U_1), \cdots, f(U_s))^\top$.

Funciones de activación

- La función de activación no lineal será $f(\mathbf{U}) = f(\beta_0 + \mathbf{BX})$, donde $\mathbf{f} = (f, \cdots, f)^\top$ es un vector s-dimensional de funciones cuyos elementos son la función f y $\mathbf{f}(\mathbf{U}) = (f(U_1), \cdots, f(U_s))^\top$.

- La función más simple es la función identidad.

Funciones de activación

- La función de activación no lineal será $f(\mathbf{U}) = f(\beta_0 + \mathbf{B}\mathbf{X})$, donde $\mathbf{f} = (f, \cdots, f)^\top$ es un vector s-dimensional de funciones cuyos elementos son la función f y $\mathbf{f}(\mathbf{U}) = (f(U_1), \cdots, f(U_s))^\top$.

- La función más simple es la función identidad.

- Sin embargo, otras funciones pueden ser consideradas dependiendo del caso al cual nos enfrentemos.

Funciones de activación

- La función de activación no lineal será $f(\mathbf{U}) = f(\beta_0 + \mathbf{B}\mathbf{X})$, donde $\mathbf{f} = (f, \cdots, f)^\top$ es un vector s-dimensional de funciones cuyos elementos son la función f y $\mathbf{f}(\mathbf{U}) = (f(U_1), \cdots, f(U_s))^\top$.

- La función más simple es la función identidad.

- Sin embargo, otras funciones pueden ser consideradas dependiendo del caso al cual nos enfrentemos.

- Existe evidencia empírica de que la función hiperbólica tangente converge más rápidamente que la función logística.

Funciones de activación

Activation Function	$f(u)$	Range of Values
Identity, linear	u	$\Re$
Hard-limiter	$\text{sign}(u)$	$\{-1,+1\}$
Heaviside, step, threshold	$I_{[u\geq 0]}$	$\{0,1\}$
Gaussian radial basis function	$(2\pi)^{-1/2}e^{-u^2/2}$	$\Re$
Cumulative Gaussian (sigmoid)	$\sqrt{2/\pi}\int_0^u e^{-z^2/2}dz$	$(0,1)$
Logistic (sigmoid)	$(1+e^{-u})^{-1}$	$(0,1)$
Hyperbolic tangent (sigmoid)	$(e^u - e^{-u})/(e^u + e^{-u})$	$(-1,+1)$

- En clasificación, la idea es usar $\mathbf{X}_1, \ldots, \mathbf{X}_n$ n vectores de entrada, o copias independientes de $\mathbf{X}$.

- En clasificación, la idea es usar $\mathbf{X}_1, \ldots, \mathbf{X}_n$ n vectores de entrada, o copias independientes de $\mathbf{X}$.

- Clasifica cada vector en dos clases Π_1 y Π_2.

Perceptron de Rosenblatt Unidad Simple

- En clasificación, la idea es usar $\mathbf{X}_1, \ldots, \mathbf{X}_n$ n vectores de entrada, o copias independientes de $\mathbf{X}$.

- Clasifica cada vector en dos clases Π_1 y Π_2.

- Solo se considera un output $(s = 1)$

- En clasificación, la idea es usar $\mathbf{X}_1, \ldots, \mathbf{X}_n$ n vectores de entrada, o copias independientes de $\mathbf{X}$.

- Clasifica cada vector en dos clases Π_1 y Π_2.

- Solo se considera un output $(s = 1)$

- En la versión lineal, se tiene que $sign\{\beta_0 + \mathbf{X}^\top \beta\}$. Es la es la función *hard-limiter*.

- El output es conocido como unidad de umbral lineal.

Perceptron de Rosenblatt Unidad Simple

- El output es conocido como unidad de umbral lineal.

- El perceptron de Rosenblatt es básicamente el de McCullog y Pitts (1943) pero con pesos.

Perceptron de Rosenblatt Unidad Simple

- El output es conocido como unidad de umbral lineal.

- El perceptron de Rosenblatt es básicamente el de McCullog y Pitts (1943) pero con pesos.

- Una versión más general está dada por $f(\beta_0 + \mathbf{X}^\top \boldsymbol{\beta})$, donde $f(\cdot)$ es una función de activación.

Perceptron de Rosenblatt Unidad Simple

- El output es conocido como unidad de umbral lineal.

- El perceptron de Rosenblatt es básicamente el de McCullog y Pitts (1943) pero con pesos.

- Una versión más general está dada por $f(\beta_0 + \mathbf{X}^\top \boldsymbol{\beta})$, donde $f(\cdot)$ es una función de activación.

- En clasificación, generalmente se considera $f(\cdot)$ como la sigmoidea.

Regla de aprendizaje

- Sea $Y = +1$ si $\mathbf{X} \in \Pi_1$ y $Y = -1$ si $\mathbf{X} \in \Pi_2$.

Regla de aprendizaje

- Sea $Y = +1$ si $\mathbf{X} \in \Pi_1$ y $Y = -1$ si $\mathbf{X} \in \Pi_2$.

- Aquí, $\mathbf{X}^\top \boldsymbol{\beta} \geqslant 0$ y $\mathbf{X}^\top \boldsymbol{\beta} < 0$.

Regla de aprendizaje

- Sea $Y = +1$ si $\mathbf{X} \in \Pi_1$ y $Y = -1$ si $\mathbf{X} \in \Pi_2$.
- Aquí, $\mathbf{X}^\top \boldsymbol{\beta} \geqslant 0$ y $\mathbf{X}^\top \boldsymbol{\beta} < 0$.
- Las clases se asumen linealmente separables.

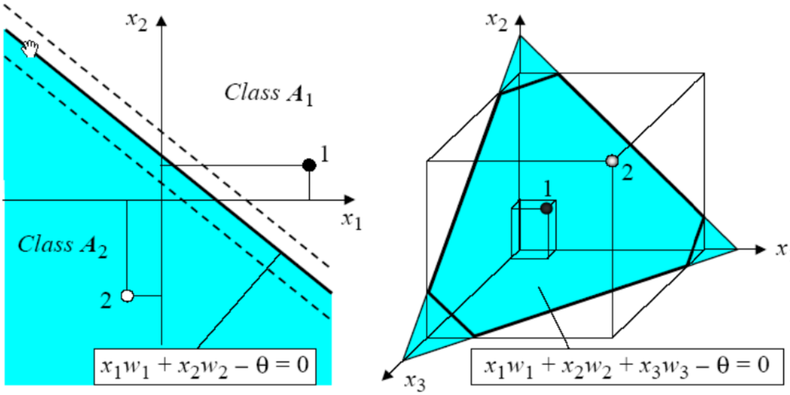
Regla de aprendizaje

- Sea $Y = +1$ si $\mathbf{X} \in \Pi_1$ y $Y = -1$ si $\mathbf{X} \in \Pi_2$.
- Aquí, $\mathbf{X}^\top \boldsymbol{\beta} \geqslant 0$ y $\mathbf{X}^\top \boldsymbol{\beta} < 0$.
- Las clases se asumen linealmente separables.
- El algoritmo utilizado es el algoritmo de aprendizaje on-line.

Regla de aprendizaje

- Sea $Y = +1$ si $\mathbf{X} \in \Pi_1$ y $Y = -1$ si $\mathbf{X} \in \Pi_2$.
- Aquí, $\mathbf{X}^\top \boldsymbol{\beta} \geqslant 0$ y $\mathbf{X}^\top \boldsymbol{\beta} < 0$.
- Las clases se asumen linealmente separables.
- El algoritmo utilizado es el algoritmo de aprendizaje on-line.
- Este algoritmo re-etiqueta $\{\mathbf{X}_i\}$ uno a la vez.

Perceptron de Rosenblatt Unidad Simple



Regla de aprendizaje

- En la iteración h, consideramos $\mathbf{X}_h$, $h = 1, 2, \ldots$

Regla de aprendizaje

- En la iteración h, consideramos $\mathbf{X}_h$, $h = 1, 2, \ldots$

- El algoritmo calcula una secuencia $\{\boldsymbol{\beta}_h\}$ de pesos usando un valor inicial $\boldsymbol{\beta}_0 = 0$.

Regla de aprendizaje

- En la iteración h, consideramos $\mathbf{X}_h$, $h = 1, 2, \ldots$

- El algoritmo calcula una secuencia $\{\boldsymbol{\beta}_h\}$ de pesos usando un valor inicial $\boldsymbol{\beta}_0 = 0$.

- Si en la h-ésima iteración, la versión actual de $\boldsymbol{\beta}_h$ clasifica $\mathbf{X}_h$ correctamente, entonces $\boldsymbol{\beta}_{h+1} = \boldsymbol{\beta}_h$. Note que si $\mathbf{X}_h^\top \boldsymbol{\beta}_h \geqslant 0$ entonces $\mathbf{X}_h \in \Pi_1$ y $\mathbf{X}_h^\top \boldsymbol{\beta}_h < 0$ entonces $\mathbf{X}_h \in \Pi_2$.

Regla de aprendizaje

- En la iteración h, consideramos $\mathbf{X}_h$, $h = 1, 2, \ldots$
- El algoritmo calcula una secuencia $\{\boldsymbol{\beta}_h\}$ de pesos usando un valor inicial $\boldsymbol{\beta}_0 = 0$.
- Si en la h-ésima iteración, la versión actual de $\boldsymbol{\beta}_h$ clasifica $\mathbf{X}_h$ correctamente, entonces $\boldsymbol{\beta}_{h+1} = \boldsymbol{\beta}_h$. Note que si $\mathbf{X}_h^\top \boldsymbol{\beta}_h \geqslant 0$ entonces $\mathbf{X}_h \in \Pi_1$ y $\mathbf{X}_h^\top \boldsymbol{\beta}_h < 0$ entonces $\mathbf{X}_h \in \Pi_2$.
- Si $\boldsymbol{\beta}_h$ no clasifica bien $\mathbf{X}_h$, entonces se actualiza el peso: si $\mathbf{X}_h^\top \boldsymbol{\beta}_h \geqslant 0$ pero $\mathbf{X}_h \in \Pi_2$, entonces $\boldsymbol{\beta}_{h+1} = \boldsymbol{\beta}_h - \eta \mathbf{X}_h$. Por el contrario, si $\mathbf{X}_h^\top \boldsymbol{\beta}_h < 0$ pero $\mathbf{X}_h \in \Pi_1$, entonces $\boldsymbol{\beta}_{h+1} = \boldsymbol{\beta}_h + \eta \mathbf{X}_h$.

- Las reglas anteriores pueden escribirse como:

- Las reglas anteriores pueden escribirse como:

$$\beta_h = \beta_{h-1} + \eta Y_h \mathbf{X}_h \ \psi(-Y_h \mathbf{X}_h^\top \beta_{h-1}),$$

donde $\psi(z) = 0$, si $z < 0$ y $\psi(z) = 1$, si $z > 0$ (función step).

- η se conoce como el parámetro de aprendizaje.

Regla de aprendizaje

- Las reglas anteriores pueden escribirse como:

$$\beta_h = \beta_{h-1} + \eta Y_h \mathbf{X}_h \ \psi(-Y_h \mathbf{X}_h^\top \beta_{h-1}),$$

  donde $\psi(z) = 0$, si $z < 0$ y $\psi(z) = 1$, si $z > 0$ (función step).

- η se conoce como el parámetro de aprendizaje.

- Generalmente se asume $\eta = 1$.

Regla de aprendizaje

- Las reglas anteriores pueden escribirse como:

$$\beta_h = \beta_{h-1} + \eta Y_h \mathbf{X}_h \ \psi(-Y_h \mathbf{X}_h^\top \beta_{h-1}),$$

donde $\psi(z) = 0$, si $z < 0$ y $\psi(z) = 1$, si $z > 0$ (función step).

- η se conoce como el parámetro de aprendizaje.

- Generalmente se asume $\eta = 1$.

- Notar que producto del procedimiento anterior, $\beta_{h+1} = \sum_{i=1}^{h} \mathbf{X}_i$.

- Supongamos que existe $\boldsymbol{\beta}^*$ tal que

Regla de aprendizaje

- Supongamos que existe $\boldsymbol{\beta}^*$ tal que

$$A = \min_{\mathbf{X}_i \in \Pi_1} \mathbf{X}_i^\top \boldsymbol{\beta}^*, \quad B = \max_{\mathbf{X}_i \in \Pi_1} \|\mathbf{X}_i\|^2 .$$

- Supongamos que existe $\boldsymbol{\beta}^*$ tal que

$$A = \min_{\mathbf{X}_i \in \Pi_1} \mathbf{X}_i^\top \boldsymbol{\beta}^*, \quad B = \max_{\mathbf{X}_i \in \Pi_1} \|\mathbf{X}_i\|^2 .$$

- Tenemos que, $\boldsymbol{\beta}_{h+1}^\top \boldsymbol{\beta}^* = \sum_{i=1}^h \mathbf{X}_i^\top \boldsymbol{\beta}^*$. Lo anterior, es siempre mayor o igual que h veces A, es decir, $\boldsymbol{\beta}_{h+1}^\top \boldsymbol{\beta}^* \geqslant hA$.

Regla de aprendizaje

- Supongamos que existe $\boldsymbol{\beta}^*$ tal que

$$A = \min_{\mathbf{X}_i \in \Pi_1} \mathbf{X}_i^\top \boldsymbol{\beta}^*, \quad B = \max_{\mathbf{X}_i \in \Pi_1} \|\mathbf{X}_i\|^2 .$$

- Tenemos que, $\boldsymbol{\beta}_{h+1}^\top \boldsymbol{\beta}^* = \sum_{i=1}^{h} \mathbf{X}_i^\top \boldsymbol{\beta}^*$. Lo anterior, es siempre mayor o igual que h veces A, es decir, $\boldsymbol{\beta}_{h+1}^\top \boldsymbol{\beta}^* \geqslant hA$.

- Usando la desigualdad de Cauchy-Schwarz ($\|xy\|^2 \leqslant \|x\|^2 \|y\|^2$), se tiene que

Regla de aprendizaje

- Supongamos que existe $\boldsymbol{\beta}^*$ tal que

$$A = \min_{\mathbf{X}_i \in \Pi_1} \mathbf{X}_i^\top \boldsymbol{\beta}^*, \quad B = \max_{\mathbf{X}_i \in \Pi_1} \|\mathbf{X}_i\|^2 .$$

- Tenemos que, $\boldsymbol{\beta}_{h+1}^\top \boldsymbol{\beta}^* = \sum_{i=1}^{h} \mathbf{X}_i^\top \boldsymbol{\beta}^*$. Lo anterior, es siempre mayor o igual que h veces A, es decir, $\boldsymbol{\beta}_{h+1}^\top \boldsymbol{\beta}^* \geqslant hA$.

- Usando la desigualdad de Cauchy-Schwarz ($\|xy\|^2 \leqslant \|x\|^2\|y\|^2$), se tiene que

$$\left(\beta_{h+1}^\top \beta^*\right)^2 \leqslant \left\|\beta_{h+1}^\top\right\|^2 \|\beta^*\|^2$$

Regla de aprendizaje

- Dado lo anterior, se puede concluir que

Regla de aprendizaje

- Dado lo anterior, se puede concluir que

$$\|\beta_{h+1}\|^2 \geqslant \frac{h^2 A^2}{\|\beta^*\|^2}$$

- Dado lo anterior, se puede concluir que

$$\|\beta_{h+1}\|^2 \geqslant \frac{h^2 A^2}{\|\beta^*\|^2}$$

- ¿Qué significa lo anterior?: la norma al cuadrado de β crece al menos cuadráticamente con el número de iteraciones.

- Dado lo anterior, se puede concluir que

$$\|\beta_{h+1}\|^2 \geqslant \frac{h^2 A^2}{\|\beta^*\|^2}$$

- ¿Qué significa lo anterior?: la norma al cuadrado de β crece al menos cuadráticamente con el número de iteraciones.

- Pensemos ahora en la regla de actualización, dada por $\beta_{k+1} = \beta_k + X_k$, donde $X_k \in \Pi_1$, $k = 1, \ldots, h$.

Regla de aprendizaje

- De lo anterior, se tiene que $\|\beta_{k+1}\|^2 = \|\beta_k\|^2 + \|\mathbf{X}_k\|^2 + 2\mathbf{X}_k^\top \beta_k$.

Regla de aprendizaje

- De lo anterior, se tiene que $\|\beta_{k+1}\|^2 = \|\beta_k\|^2 + \|\mathbf{X}_k\|^2 + 2\mathbf{X}_k^\top \beta_k$.

- Cuando $\mathbf{X}_k$ es clasificado incorrectamente, $\mathbf{X}_k^\top \boldsymbol{\beta}_k < 0$. Entonces,

$$\|\boldsymbol{\beta}_{k+1}\|^2 - \|\boldsymbol{\beta}_k\|^2 \leqslant \|\mathbf{X}_k\|^2.$$

Regla de aprendizaje

- De lo anterior, se tiene que $\|\boldsymbol{\beta}_{k+1}\|^2 = \|\boldsymbol{\beta}_k\|^2 + \|\mathbf{X}_k\|^2 + 2\mathbf{X}_k^\top \boldsymbol{\beta}_k$.

- Cuando $\mathbf{X}_k$ es clasificado incorrectamente, $\mathbf{X}_k^\top \boldsymbol{\beta}_k < 0$. Entonces,

$$\|\boldsymbol{\beta}_{k+1}\|^2 - \|\boldsymbol{\beta}_k\|^2 \leqslant \|\mathbf{X}_k\|^2.$$

- Observemos que $\|\boldsymbol{\beta}_{h+1}\|^2 \leqslant \sum_{k=1}^{h} \|\mathbf{X}_k\|^2 \leqslant hB$.

Regla de aprendizaje

- De lo anterior, se tiene que $\|\beta_{k+1}\|^2 = \|\beta_k\|^2 + \|\mathbf{X}_k\|^2 + 2\mathbf{X}_k^\top \beta_k$.

- Cuando $\mathbf{X}_k$ es clasificado incorrectamente, $\mathbf{X}_k^\top \boldsymbol{\beta}_k < 0$. Entonces,

$$\|\boldsymbol{\beta}_{k+1}\|^2 - \|\boldsymbol{\beta}_k\|^2 \leqslant \|\mathbf{X}_k\|^2.$$

- Observemos que $\|\beta_{h+1}\|^2 \leqslant \sum_{k=1}^{h} \|\mathbf{X}_k\|^2 \leqslant hB$.

- ¿Qué significa lo anterior?: la norma al cuadrado de $\boldsymbol{\beta}$ crece a lo mas linealmente con número de iteraciones.

Regla de aprendizaje

- Si h crece, lo anterior parece ser contradictorio pues por un lado $||\beta_{k+1}||^2||$ crece al menos cuadráticamente con h pero a lo más linealmente con h.

Regla de aprendizaje

- Si h crece, lo anterior parece ser contradictorio pues por un lado $||\beta_{k+1}||^2||$ crece al menos cuadráticamente con h pero a lo más linealmente con h.

- Entonces, h no puede crecer sin tener una cota.

Regla de aprendizaje

- Si h crece, lo anterior parece ser contradictorio pues por un lado $||\beta_{k+1}||^2||$ crece al menos cuadráticamente con h pero a lo más linealmente con h.
- Entonces, h no puede crecer sin tener una cota.
- Lo ideal es encontrar un h_{max}.

Regla de aprendizaje

- Si h crece, lo anterior parece ser contradictorio pues por un lado $||\beta_{k+1}||^2||$ crece al menos cuadráticamente con h pero a lo más linealmente con h.

- Entonces, h no puede crecer sin tener una cota.

- Lo ideal es encontrar un h_{max}.

- En este caso, $h_{\text{max}} = \frac{B||\beta^*||^2}{A^2}$.

Regla de aprendizaje

- Si h crece, lo anterior parece ser contradictorio pues por un lado $||\beta_{k+1}||^2||$ crece al menos cuadráticamente con h pero a lo más linealmente con h.
- Entonces, h no puede crecer sin tener una cota.
- Lo ideal es encontrar un h_{max}.
- En este caso, $h_{\text{max}} = \frac{B||\beta^*||^2}{A^2}$.
- Por lo tanto, se establece que el algoritmo encontrará una solución en un número finito de iteraciones.

Regla de aprendizaje

- Existe muchos problemas para los cuales β^* no existe.

- Existe muchos problemas para los cuales β^* no existe.

- Si el algoritmo para, se obtiene una solución.

Regla de aprendizaje

- Existe muchos problemas para los cuales β^* no existe.
- Si el algoritmo para, se obtiene una solución.
- Si el problema no es linealmente separable, el algoritmo iterará por un ciclo indeterminado.

Regla de aprendizaje

- Existe muchos problemas para los cuales β^* no existe.
- Si el algoritmo para, se obtiene una solución.
- Si el problema no es linealmente separable, el algoritmo iterará por un ciclo indeterminado.
- Si el algoritmo se detiene prematuramente, el vector β obtenido puede que no generalice bien los resultados en el conjunto de test.

Descenso por gradiente

- Supongamos que nos interesa minimizar la pérdida entre Y e y (el valor obtenido usando la red neuronal).

Descenso por gradiente

- Supongamos que nos interesa minimizar la pérdida entre Y e y (el valor obtenido usando la red neuronal).

- Sea E una función de pérdida.

Descenso por gradiente

- Supongamos que nos interesa minimizar la pérdida entre Y e y (el valor obtenido usando la red neuronal).

- Sea E una función de pérdida.

- Entonces, buscamos β^* tal que

Descenso por gradiente

- Supongamos que nos interesa minimizar la pérdida entre Y e y (el valor obtenido usando la red neuronal).

- Sea E una función de pérdida.

- Entonces, buscamos $\boldsymbol{\beta}^*$ tal que

$$\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}} E(Y, y).$$

- Considerando la función de activación lineal, tenemos que

- Considerando la función de activación lineal, tenemos que

$$y = \beta_0 + \mathbf{X}^\top \boldsymbol{\beta}$$

- Considerando la función de activación lineal, tenemos que

$$y = \beta_0 + \mathbf{X}^\top \boldsymbol{\beta}$$

- Sea la función de pérdida (error cuadrático medio):

Descenso por gradiente

- Considerando la función de activación lineal, tenemos que

$$y = \beta_0 + \mathbf{X}^\top \boldsymbol{\beta}$$

- Sea la función de pérdida (error cuadrático medio):

$$E(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{y}\|^2$$

- Entonces, buscamos $\boldsymbol{\beta}^*$ tal que

Descenso por gradiente

- Considerando la función de activación lineal, tenemos que

$$y = \beta_0 + \mathbf{X}^\top \boldsymbol{\beta}$$

- Sea la función de pérdida (error cuadrático medio):

$$E(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{y}\|^2$$

- Entonces, buscamos $\boldsymbol{\beta}^*$ tal que

$$\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}} E(Y, y).$$

Descenso por gradiente

- Es posible mostrar que

Descenso por gradiente

- Es posible mostrar que

$$\frac{\partial E}{\partial \beta_i} = \sum_{j=1}^{n} 2(Y_j - y_j)(-\mathbf{X}_{ij})$$

y que $\Delta\beta_i = -\eta \frac{\partial E}{\partial \beta_i}$.

Descenso por gradiente

- Es posible mostrar que

$$\frac{\partial E}{\partial \beta_i} = \sum_{j=1}^{n} 2(Y_j - y_j)(-\mathbf{X}_{ij})$$

y que $\Delta\beta_i = -\eta \frac{\partial E}{\partial \beta_i}$.

- La función gradiente apunta en dirección ascendente y el negativo de la función gradiente, en dirección descendente.

Descenso por gradiente

- Es posible mostrar que

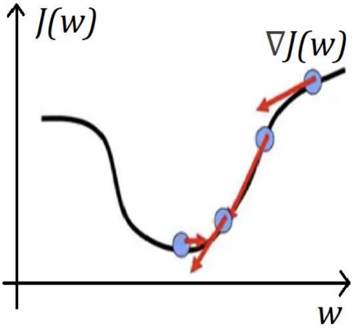
$$\frac{\partial E}{\partial \beta_i} = \sum_{j=1}^{n} 2(Y_j - y_j)(-\mathbf{X}_{ij})$$

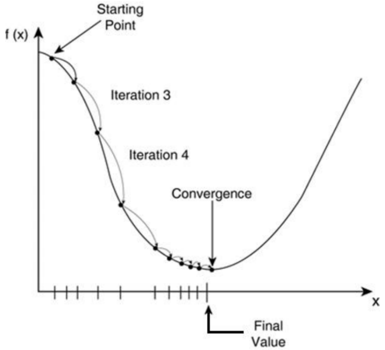
  y que $\Delta\beta_i = -\eta\frac{\partial E}{\partial \beta_i}$.

- La función gradiente apunta en dirección ascendente y el negativo de la función gradiente, en dirección descendente.

- El ir en esta última dirección asegura convergencia el mínimo local de la función de pérdida.

Descenso por gradiente





Descenso por gradiente

- El algoritmo inicia fijando η en algún valor. Algunos autores señalan $\eta = 1$ mientras que otros valores pequeños (0.03 por ejemplo).

Descenso por gradiente

- El algoritmo inicia fijando η en algún valor. Algunos autores señalan $\eta = 1$ mientras que otros valores pequeños (0.03 por ejemplo).

- Inicializa β al azar (desde alguna distribución) mientras que otros autores señalan $\mathbf{0}$.

Descenso por gradiente

- El algoritmo inicia fijando η en algún valor. Algunos autores señalan $\eta = 1$ mientras que otros valores pequeños (0.03 por ejemplo).

- Inicializa β al azar (desde alguna distribución) mientras que otros autores señalan $\mathbf{0}$.

- Luego, se actualiza $\boldsymbol{\beta}_{h+1} = \boldsymbol{\beta}_h - \Delta\beta_i$.