Redes Neuronales Monocapa I

Modelos de la computación (Aprendizaje supervisado)

Francisco Fernández Navarro

Departamento de Lenguajes y Ciencias de la Computación
Área: Ciencias de la Computación e Inteligencia Artificial

Índice de contenidos

¿Qué es un Perceptrón Simple?

- El perceptrón simple es un tipo de modelo de aprendizaje automático.
- Fue propuesto por Frank Rosenblatt en 1957.
- Es la unidad básica de una red neuronal.
- Se utiliza para problemas de clasificación binaria.
- Inspirado en la forma en que funcionan las neuronas biológicas.

Funcionamiento y limitaciones

¿Cómo funciona?

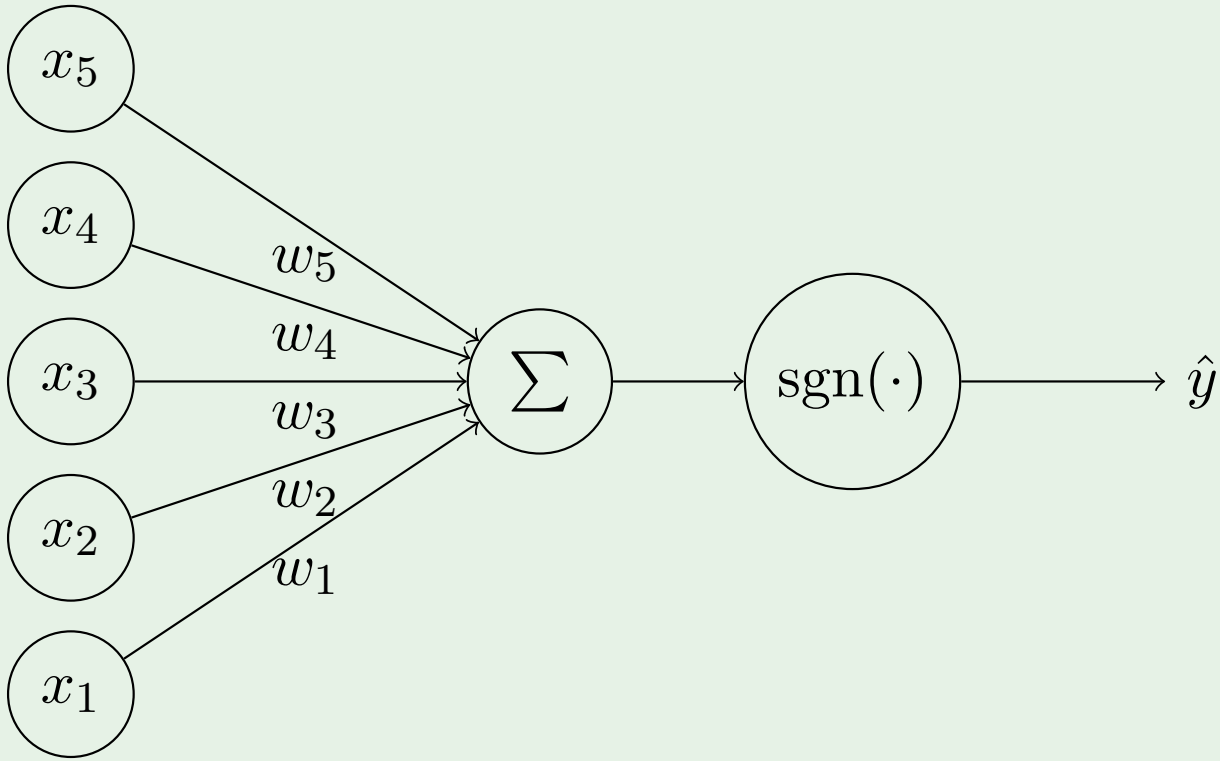
- El perceptrón toma entradas ponderadas y las suma.
- Aplica una función de activación (generalmente una función escalón) a la suma.
- Produce una salida binaria (0 o 1) en función del resultado de la función de activación.

Limitaciones

- Limitado a problemas de clasificación linealmente separables.
- No puede aprender problemas no lineales (XOR).
- Superado por modelos más complejos como las redes neuronales multicapa.

Representación gráfica

Modelo perceptrón simple con 5 entradas



Índice de contenidos

Formulación base y objetivo

Formulación del problema

Los parámetros de los modelos de aprendizaje se estiman a partir de un conjunto de entrenamiento $\mathcal{D} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, donde $\mathbf{x}_n = (x_{n1}, x_{n2}, \ldots, x_{nK}) \in \mathbb{R}^K$ es el vector de atributos del n-ésimo patrón, K es la dimensión del espacio de entrada (número de atributos en el problema), $y_n \in \{0, 1\}$ es la etiqueta de clase. La función final, $f : \mathbb{R}^K \rightarrow \{0, 1\}$.

Objetivo

Estimar los parámetros de la función f a partir del conjunto de datos $\mathcal{D}$ con el objetivo de reducir el error de predicción.

Unidad de proceso bipolar

Asumimos una codificación $\{-1, 1\}$ de la variable dependiente para el problema de clasificación, y por ello, la función a estimar realizará un *mapping* del tipo $f : \mathbb{R}^K \to \{-1, 1\}$

Formulación de la unidad

La salida de la unidad para un patrón n-ésimo, $\mathbf{x}_n \in \mathbb{R}^K$ es:

$$\hat{y}(\mathbf{x}_n) = f(\mathbf{x}_n) = \begin{cases} 1 & \text{si} \quad w_1 x_{n1} + w_2 x_{n2} + \ldots + w_K x_{nK} \geq \theta \\ -1 & \text{si} \quad w_1 x_{n1} + w_2 x_{n2} + \ldots + w_K x_{nK} < \theta \end{cases} \tag{1}$$

donde $\mathbf{w} \in \mathbb{R}^K$ son los pesos con los que se ponderan los valores de entrada (conocidos como **pesos sinápticos**) y el parámetro θ es el **umbral o sesgo** de la unidad de procesamiento.

Unidad de proceso bipolar. Aspectos a tener en cuenta

- A la suma ponderada $u_n = w_1 x_{n1} + w_2 x_{n2} + \ldots + w_K x_{nK} \in \mathbb{R}$ se le llama en el área **potencial sináptico**.
- La función asociada a la unidad de proceso bipolar también puede definirse a través de la función signo como:

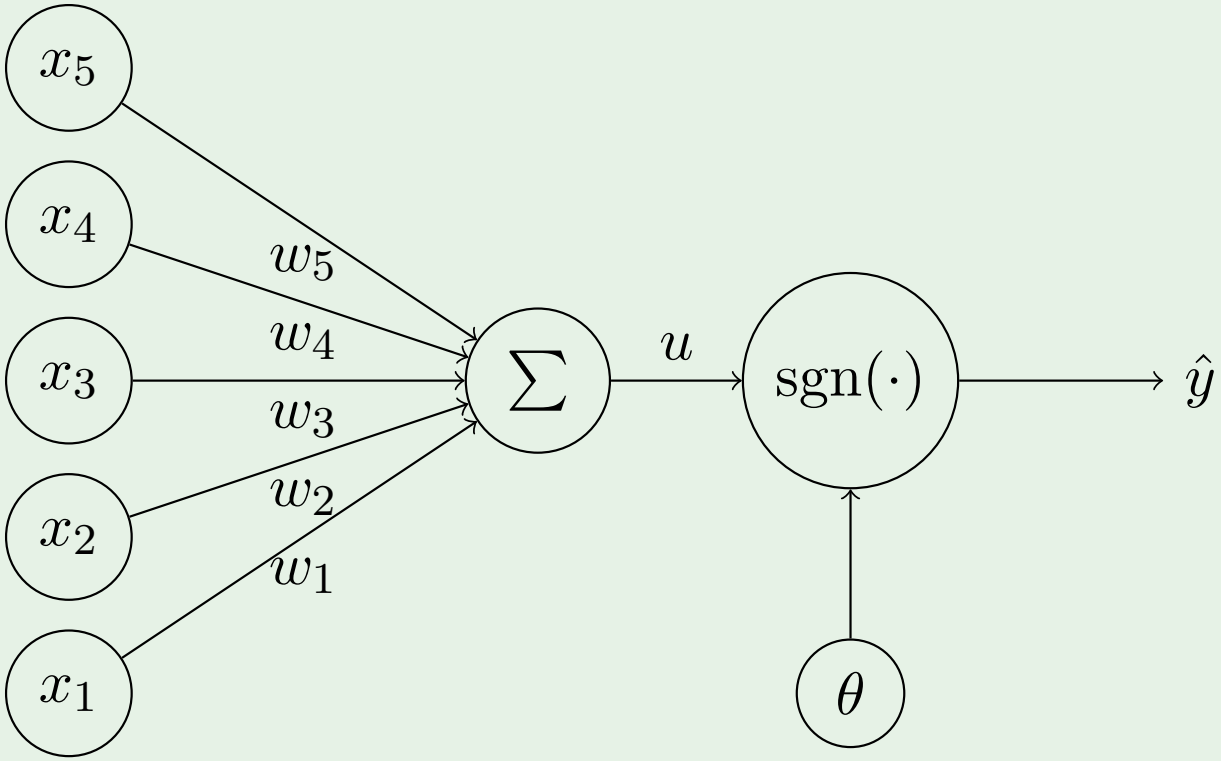
$$\hat{y}(\mathbf{x}_n) = f(\mathbf{x}_n) = \text{sgn}(u_n - \theta), \tag{2}$$

siendo la función signo

$$\text{sgn}(x) = \begin{cases} 1 & \text{si} \quad x \geq 0 \\ -1 & \text{si} \quad x < 0 \end{cases}$$

Representación gráfica

Modelo perceptrón simple con 5 entradas. Unidad de proceso bipolar



Unidad de proceso binaria

En esta unidad de proceso se asume una codificación $\{0, 1\}$ para la variable dependiente del modelo. Consecuentemente, la función a estimar realizará un *mapping* del tipo $f : \mathbb{R}^K \rightarrow \{0, 1\}$

Formulación de la unidad

La salida de la unidad para un patrón n-ésimo, $\mathbf{x}_n \in \mathbb{R}^K$ es:

$$\hat{y}(\mathbf{x}_n) = f(\mathbf{x}_n) = \begin{cases} 1 & \text{si} \quad w_1 x_{n1} + w_2 x_{n2} + \ldots + w_K x_{nK} \geq \theta \\ 0 & \text{si} \quad w_1 x_{n1} + w_2 x_{n2} + \ldots + w_K x_{nK} < \theta \end{cases} \tag{3}$$

Índice de contenidos

Regla de aprendizaje del perceptrón simple

Idea principal

- Comenzar con unos valores iniciales aleatorios e ir modificándolos iterativamente cuando la salida de la unidad no coincide con la salida deseada.
- Vamos a suponer que las iteraciones del algoritmo van de 1 hasta I, $i = 1, 2, \ldots, I$.
- Iremos modificando los parámetros además patrón a patrón, $n = 1, \ldots, N$.

Regla de aprendizaje del perceptrón simple

Regla de aprendizaje

Los pesos sinápticos son modificados en función de la siguiente función:

$$w_k(i, n+1) = w_k(i, n) + \Delta w_k(i, n), \tag{4}$$

donde $w_k(i, n)$ es el valor del vector de pesos en su dimensión k-ésima, en la iteración i-ésima, cuando recibe por entrada el patrón n-ésimo ($\mathbf{x}_n$), y $\Delta w_k(i, n)$ es la variación de dicha componente:

$$\Delta w_k(i, n) = \eta(i)(y_n - \hat{y}_n(i))x_{nk} \tag{5}$$

siendo $\eta(i)$ la **tasa de aprendizaje** en la iteración i-ésima y $\hat{y}_n(i) \in \{-1, 1\}$ la predicción del modelo en la iteración i-ésima (con los pesos sinápticos asociados a esa iteración) para el patrón n-ésimo ($\mathbf{x}_n$).

Regla de aprendizaje del perceptrón simple

A tener en cuenta

A mayor valor $\eta(i)$ mayor modificación del peso sináptico y viceversa. Cuando se toma constante en todas las iteraciones, $\eta(i) = \eta > 0$ tendremos la regla de adaptación con incremento fijo.

Cuestión

¿Como configuraríais el parámetro η?

Regla de aprendizaje del perceptrón simple

Cuando la función de transferencia usada es la función signo (valores bipolares) la regla de aprendizaje se puede escribir de la forma:

$$w_k(i, n+1) = \begin{cases} w_k(i,n) + 2\eta(i)x_{nk} & \text{si} \quad y_n = 1, \hat{y}_n(i) = -1 \\ w_k(i,n) & \text{si} \quad y_n = \hat{y}_n(i) \\ w_k(i,n) - 2\eta(i)x_{nk} & \text{si} \quad y_n = -1, \hat{y}_n(i) = 1 \end{cases} \tag{6}$$

Modificación del sesgo

Debemos tener en cuenta que

Para un patrón n-ésimo, $\mathbf{x}_n$:

$$w_1 x_{n1} + \ldots + w_K x_{nK} \geq \theta \Leftrightarrow w_1 x_{n1} + \ldots + w_K x_{nK} + w_{K+1} x_{nK+1} \geq 0,$$

con $x_{nK+1} = -1$ y $w_{K+1} = \theta$. Por ello:

$$\Delta\theta(i,n) = -\eta(i)(y_n - \hat{y}_n(i)) \tag{7}$$

Flujo algorítmico

Perceptrón simple $(\mathcal{D}, \eta(i) = \eta > 0)$:

1: $\mathbf{X} \leftarrow (\mathbf{X} \ -\mathbf{1}_N) \in \mathbb{R}^{N\times(K+1)}$
2: **for** $k = 1$ until $K + 1$ **do**
3: $\quad w_k(1,0) \leftarrow 2 \times \text{rand}() - 1, 0 \leq \text{rand}() < 1$
$\quad (\mathbf{w}(1,0) = (w_1(1,0), \ldots, w_{K+1}(1,0)))$.
4: **end for**
5: **for** $i = 1$ until I **do**
6: $\quad$ **for** $n = 1$ until N **do**
7: $\quad\quad \hat{y}_n(i) \leftarrow \text{sgn}((\mathbf{w}(i,n))'\mathbf{x}_n)$
8: $\quad\quad$ **for** $k = 1$ until $K + 1$ **do**
9: $\quad\quad\quad \Delta w_k(i,n) \leftarrow \eta(i)(y_n - \hat{y}_n(i))x_{nk}$
10: $\quad\quad\quad w_k(i,n) \leftarrow w_k(i,n-1) + \Delta w_k(i,n)$.
11: $\quad\quad$ **end for**
12: $\quad$ **end for**
13: **end for**

Criterios de parada

¿Cuando paramos?

- Número máximo de iteraciones.
- Si el error de entrenamiento cae por debajo de este umbral, se detiene el entrenamiento.
- Puedes detener el entrenamiento cuando el cambio en el error (por ejemplo, el error cuadrático medio) entre épocas consecutivas es menor que un cierto umbral predefinido. Esto indica que la red ha convergido lo suficiente.

Índice de contenidos

Teorema de convergencia

Teorema

Si el conjunto de patrones de entrenamiento con sus salidas deseadas, es linealmente separable entonces el perceptrón simple encuentra una solución en un número finito de iteraciones

Demostración

Como los patrones son linealmente separables existirán unos valores $w_1^\star, w_2^\star, \ldots, w_{K+1}^\star$ tales que:

$$\begin{array}{ll} \sum_{k=1}^{K} w_k^\star x_{nk} \geq w_{K+1}^\star & \text{si} \quad \mathbf{x}_n \in \mathcal{C}_1 \\ \sum_{k=1}^{K} w_k^\star x_{nk} < w_{K+1}^\star & \text{si} \quad \mathbf{x}_n \in \mathcal{C}_2 \end{array} \tag{8}$$

Teorema de convergencia

Demostración

Supongamos que en la iteración i y para el patrón n, la red tiene que modificar los pesos sinápticos según la regla de aprendizaje, puesto que la salida de la red $\hat{y}_n(i)$ no coincide con la salida deseada y_n, entonces:

$$\begin{aligned}\sum_{k=1}^{K+1}(w_k(i,n+1)-w_k^\star)^2 &= \sum_{k=1}^{K+1}(w_k(i,n)+\eta(y_n-\hat{y}_n(i))x_{nk}-w_k^\star)^2 \\ &= \sum_{k=1}^{K+1}(w_k(i,n)-w_k^\star)^2+\eta^2(y_n-\hat{y}_n(i))^2\sum_{k=1}^{K+1}(x_{nk})^2 \\ &\quad +2\eta(y_n-\hat{y}_n(i))\sum_{k=1}^{K+1}(w_k(i,n)-w_k^\star)(x_{nk})\end{aligned}$$

El último término puede expresarse como:

$$2\eta(y_n-\hat{y}_n(i))\sum_{k=1}^{K+1}w_k(i,n)(x_{nk})-2\eta(y_n-\hat{y}_n(i))\sum_{k=1}^{K+1}w_k^\star(x_{nk})$$

Teorema de convergencia

Demostración

- Obsérvese que $(y_n - \hat{y}_n(i)) \sum_{k=1}^{K+1} w_k(i,n)(x_{nk}) < 0$, ya que si $\sum_{k=1}^{K+1} w_k(i,n)(x_{nk}) > 0$, entonces $\hat{y}_n(i) = 1$ y al estar en un error, forzosamente $y_n = -1$. Dicho término puede expresarse como: $-2 \left| \sum_{k=1}^{K+1} w_k(i,n)(x_{nk}) \right|$.
- Asimismo, el término $(y_n - \hat{y}_n(i)) \sum_{k=1}^{K+1} w_k^\star(x_{nk}) > 0$, puesto que si $\sum_{k=1}^{K+1} w_k^\star(x_{nk}) > 0$, entonces $y_n = 1$ (ya que el modelo acertaría en su punto final), y consecuentemente $\hat{y}_n(i) = -1$ (en esa iteración asumimos que falla). Por ello, el término puede definirse como: $2 \left| \sum_{k=1}^{K+1} w_k^\star(x_{nk}) \right|$
- $\eta^2 (y_n - \hat{y}_n(i))^2 \sum_{k=1}^{K+1} (x_{nk})^2 = 4\eta^2 \sum_{k=1}^{K+1} (x_{nk})^2$, ya que $(y_n - \hat{y}_n(i))$ al tener un error será o 2 o -2 (al cuadrado 4).

Teorema de convergencia

Demostración

Por lo tanto tenemos que

$$\sum_{k=1}^{K+1}(w_k(i,n+1)-w_k^\star)^2 \leq \sum_{k=1}^{K+1}(w_k(i,n)-w_k^\star)^2+4\eta^2\sum_{k=1}^{K+1}(x_{nk})^2 -4\eta\left|\sum_{k=1}^{K+1}w_k^\star(x_{nk})\right|$$

puesto que hemos prescindido (de forma arbitraria) de un término negativo en la derecha de la expresión $\left(-4\eta\left|\sum_{k=1}^{K+1}w_k(i,n)(x_{nk})\right|\right)$. De esta forma

$$\sum_{k=1}^{K+1}(w_k(i,n+1)-w_k^\star)^2 \leq \sum_{k=1}^{K+1}(w_k(i,n)-w_k^\star)^2 +4\eta\left(\eta\sum_{k=1}^{K+1}(x_{nk})^2-\left|\sum_{k=1}^{K+1}w_k^\star(x_{nk})\right|\right)$$

Teorema de convergencia

Definición de los límites superiores e inferiores de η

Teniendo en cuenta que:

$$\sum_{k=1}^{K+1}(w_k(i,n+1)-w_k^\star)^2 \leq \sum_{k=1}^{K+1}(w_k(i,n)-w_k^\star)^2 + 4\eta\left(\eta\sum_{k=1}^{K+1}(x_{nk})^2-\left|\sum_{k=1}^{K+1}w_k^\star(x_{nk})\right|\right)$$

Lo que nos lleva a la restricción de que $\left(\eta\sum_{k=1}^{K+1}(x_{nk})^2-\left|\sum_{k=1}^{K+1}w_k^\star(x_{nk})\right|\right)<0$, y de esta forma:

$$0<\eta<\frac{\left|\sum_{k=1}^{K+1}w_k^\star(x_{nk})\right|}{\sum_{k=1}^{K+1}(x_{nk})^2}$$

Índice de contenidos

Estimación tasa de aprendizaje

Selección del valor óptimo de η

Queremos elegir un valor óptimo de η, $\eta^\star$, con el que $D(i,n+1) = \sum_{k=1}^{K+1}(w_k(i,n+1) - w_k^\star)^2$ sea mínimo (consiguiendo un mayor acercamiento de los pesos de la red a la solución). Recordemos que:

$$D(i,n+1) = D(i,n) + 4\eta^2 \sum_{k=1}^{K+1}(x_{nk})^2 - 4\eta \left|\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})\right| - 4\eta \left|\sum_{k=1}^{K+1} w_k^\star(x_{nk})\right|$$

La derivada con respecto a η

$$\frac{\partial D(i,n+1)}{\partial \eta} = 8\eta \sum_{k=1}^{K+1}(x_{nk})^2 - 4\left|\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})\right| - 4\left|\sum_{k=1}^{K+1} w_k^\star(x_{nk})\right| = 0$$

y asi

$$\eta^\star = \frac{\left|\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})\right| + \left|\sum_{k=1}^{K+1} w_k^\star(x_{nk})\right|}{2\sum_{k=1}^{K+1}(x_{nk})^2}$$

Estimación tasa de aprendizaje

Selección del valor óptimo de η

Problema: Desconocemos el valor de $w_k^\star$, por lo que no conocemos el segundo término del numerador. Si aproximamos dicho término por el anterior tenemos:

$$\eta^\star = \frac{\left|\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})\right| + \left|\sum_{k=1}^{K+1} w_k^\star(x_{nk})\right|}{2\sum_{k=1}^{K+1}(x_{nk})^2} = \frac{\left|\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})\right|}{\sum_{k=1}^{K+1}(x_{nk})^2}$$

Teniendo en cuenta que
$-2\left|\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})\right| = (y_n - \hat{y}_n(i))\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})$, entonces:

$$\eta^\star = -\frac{(y_n - \hat{y}_n(i))\sum_{k=1}^{K+1} w_k(i,n)(x_{nk})}{2\sum_{k=1}^{K+1}(x_{nk})^2}$$

Estimación tasa de aprendizaje

Selección del valor óptimo de η

Sustituyendo el parámetro η en la regla de aprendizaje::

$$w_k(i, n+1) = w_k(i, n) - \frac{(y_n - \hat{y}_n(i)) \sum_{k=1}^{K+1} w_k(i, n)(x_{nk})}{2 \sum_{k=1}^{K+1} (x_{nk})^2} (y_n - \hat{y}_n(i)) x_{nk}$$

y teniendo en cuenta que $(y_n - \hat{y}_n(i))(y_n - \hat{y}_n(i)) = 4$, entonces:

$$w_k(i, n+1) = w_k(i, n) - 2 \frac{\sum_{k=1}^{K+1} w_k(i, n)(x_{nk})}{\sum_{k=1}^{K+1} (x_{nk})^2} x_{nk}$$

¡Gracias por vuestra atención!