Máster Universitario en Lógica, Computación e Inteligencia Artificial.

Aprendizaje Automático

Diego Enrique Morales Polanco

Cuestionario 3

Ejercicio 1. Sea N_1 un perceptrón con dos neuronas en la capa de salida que llamaremos A y B, sea *softmax* la función de activación de esa capa de salida. A partir de N_1 vamos a construir un nuevo perceptrón multicapa N_2 haciendo los siguientes cambios.

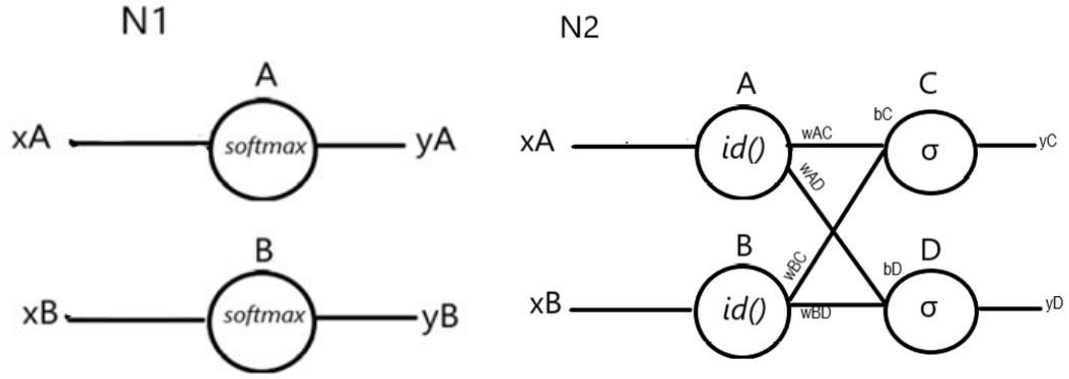
1. En N_2 la función de activación de A y B será la función identidad. $(\forall x \in \mathbb{R}) id(x) = x$.

2. Tras la capa formada por A y B, en N_2 añadimos una nueva capa formada por dos neuronas, que llamaremos C y D. La función de activación de C y D es la función sigmoide.

3. Los pesos asociados a las entradas ficticias de C y D, $\omega_{AC}, \omega_{AD}, \omega_{BC}$ y ω_{BD} a los pesos de las conexiones entre las neuronas indicadas. Supondremos también que $\omega_{AC} = 1$ y $\omega_{BD} = -1$.

Supongamos que, para cualquier entrada, las redes N_1 y N_2 proporcionan la misma salida (esto es, la salida de la neurona A en N_1 es la misma salida que la neurona C en N_2 después de aplicar las respectivas funciones de activación y, análogamente, las salidas de B en N_1 y D en N_2 después de aplicar las funciones de activación también coinciden). Calcular ω_{AD} y ω_{BC}.

Primero dibujamos ambos perceptrones descritos para tener una visualización de las redes:



En N2, la salida de A y B es x_A y x_B respectivamente debido a la función identidad.

Según el enunciado, N1 y N2 proporcionan las mismas salidas. Por lo que:

$$y_A = y_C$$

$$y_B = y_D$$

Aplicando la función softmax para N1, tendríamos las siguientes expresiones de las salidas:

$$y_A = \frac{e^{x_A}}{e^{x_A} + e^{x_B}}$$

$$y_B = \frac{e^{x_B}}{e^{x_A} + e^{x_B}}$$

Expandiendo N2 obtenemos lo siguiente:

$$y_C = \sigma(x_A w_{AC} + x_B w_{BC} + b_C)$$

$$y_D = \sigma(x_A w_{AD} + x_B w_{BD} + b_D)$$

Expandiendo la función sigmoide:

$$y_C = \frac{1}{1 + e^{-(x_A w_{AC} + x_B w_{BC} + b_C)}}$$

$$y_D = \frac{1}{1 + e^{-(x_A w_{AD} + x_B w_{BD} + b_D)}}$$

Sabiendo que $\omega_{AC} = 1$ y $\omega_{BD} = -1$y los bias son 0.

$$y_C = \frac{1}{1 + e^{-(x_A + x_B w_{BC})}}$$

$$y_D = \frac{1}{1 + e^{-(x_A w_{AD} - x_B)}}$$

Recordando que:

$$y_A = y_C$$

$$y_B = y_D$$

Tenemos las siguientes ecuaciones:

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{1}{1 + e^{-(x_A + x_B w_{BC})}}$$

$$\frac{e^{x_B}}{e^{x_A} + e^{x_B}} = \frac{1}{1 + e^{-(x_A w_{AD} - x_B + b_D)}}$$

Resolviendo:

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{1}{1 + e^{-(x_A + x_B w_{BC})}}$$

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{1}{1 + e^{-(x_A + x_B w_{BC})}} * \frac{e^{x_A}}{e^{x_A}}$$

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{e^{x_A}}{e^{x_A} + (e^{-(x_A + x_B w_{BC})} * e^{x_A})}$$

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{e^{x_A}}{e^{x_A} + e^{-x_A - x_B w_{BC} + x_A}}$$

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{e^{x_A}}{e^{x_A} + e^{-x_B w_{BC}}}$$

Aquí vemos que para que el lado izquierdo sea igual que el lado derecho, w_{BC} debe ser igual a -1:

$$\frac{e^{x_A}}{e^{x_A} + e^{x_B}} = \frac{e^{x_A}}{e^{x_A} + e^{-x_B(-1)}}$$

$$\frac{e^{x_A}}{e^{x_A}+e^{x_B}} = \frac{e^{x_A}}{e^{x_A}+e^{x_B}}$$

Exactamente el mismo procedimiento se puede hacer para la otra igualdad, finalmente:

$$\boxed{w_{BC} = -1,\ \ w_{AD} = -1}$$

Ejercicio 2. Supongamos que tenemos un conjunto de datos donde no hay ruido (noise) y un algoritmo de aprendizaje L tal que L(D) siempre es la misma hipótesis, sea cual sea el vector de entrenamiento D. ¿Qué podemos decir sobre el error esperado, el sesgo y la varianza del algoritmo L sobre un conjunto de test extraído de ese conjunto de datos?

En primera instancia, ya que no hay ruido, podemos predecir que el error esperado del algoritmo L dependerá únicamente de el sesgo y la varianza. Ahora procederemos a analizar la influencia del sesgo y la varianza.

Si el resultado del algoritmo de aprendizaje siempre es la misma hipótesis independientemente del vector de entrenamiento, significa que el predictor esperado $\bar{h}$, es igual a la hipótesis que resulta del algoritmo de aprendizaje en cualquier vector de entrenamiento, h^*. Esto significa que:

$$\bar{h}(x) = \sum_{D} h_D(x)\, P(h_D) = h^*(x) * 1 = h^*(x)$$

El valor que el predictor le asigna a cada punto x de cualquier conjunto de entrenamiento es valor que le asignaría el predictor esperado.

Tomando esto en cuenta, podemos afirmar lo siguiente a partir de la definición de varianza:

$$\sum_{(x,y)\in T} \left(h_D(x) - \bar{h}(x)\right)^2 P\big((x,y)\big)$$

$$pero$$

$$h_D(x) = \bar{h}(x) = h^*(x)$$

$$por\ tanto$$

$$\left(h_D(x) - \bar{h}(x)\right)^2 = 0$$

La varianza en cualquier vector de prueba extraído es 0.

Si se toma un conjunto de entrenamiento con todos los puntos $x \in X$, el predictor resultante será el mismo que tomando cualquier subconjunto de X con menor tamaño que X. Esto significa que, si el algoritmo de aprendizaje es consistente con el conjunto de entrenamiento que tiene todos los elementos de X, pues clasificará correctamente todos los elementos $x \in X$, independientemente de el conjunto de entrenamiento que se le proporcione para ser entrenado y el conjunto de prueba que se extraiga. Por lo que el sesgo sería 0.

Si el algoritmo de aprendizaje no es consistente, todo el error esperado dependerá únicamente del sesgo introducido por el algoritmo.

Se puede afirmar que el algoritmo L tiende a hacer underfitting al vector de entrenamiento D.

Ejercicio 3. Supongamos que la asignación de etiquetas a los elementos de cualquier conjunto de entrenamiento es determinista, esto es, al mismo $x \in U$ siempre le asignamos el mismo valor de clasificación y. Consideremos un algoritmo de aprendizaje tal que a cada conjunto de entrenamiento D le asigna f_D, esto es, $L(D) = f_D$ con la propiedad de que el valor esperado de $f_D(x)$ al variar D es la misma etiqueta y que tenemos para x en el conjunto de entrenamiento. ¿Qué podemos decir sobre el sesgo y la varianza de ese algoritmo de aprendizaje sobre un conjunto de test extraído de ese conjunto de datos?

Primero, el hecho de que al mismo $x \in U$ siempre se le asigne el mismo valor de clasificación y, significa que no hay ruido introducido por el conjunto de entrenamiento, ya que la etiqueta esperada de cada dato ($\bar{y}$) es la misma que la etiqueta real del dato (y).

Por otro lado, si para cada f_D que se obtenga del algoritmo de aprendizaje al variar D, el valor de clasificación asignado por f_D a cada elemento del conjunto de entrenamiento es igual al valor real de clasificación, significa que L es un algoritmo consistente con su conjunto de entrenamiento y lo suficientemente expresivo para, valga la redundancia, expresar todos los conceptos en U.

En base a esto, del sesgo se puede concluir lo siguiente:

- Todo x del conjunto de entrenamiento que también se encuentre en el conjunto de test será clasificado correctamente por f_D.

- Que el algoritmo sea consistente con el conjunto de entrenamiento no significa que lo será con el conjunto de test, por lo que habrá sesgo introducido por el algoritmo de aprendizaje.

- Si todos los elementos del conjunto de test aparecen en el conjunto de entrenamiento el sesgo será 0. Por lo que en este caso la distribución entre el conjunto de entrenamiento y test tiene un efecto en el sesgo.

De la varianza se puede concluir lo siguiente:

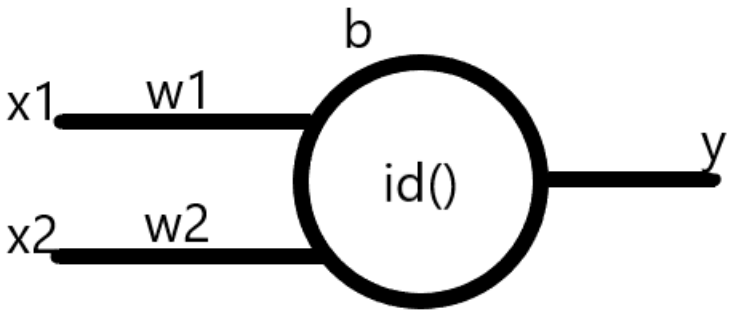
- El predictor esperado ($\bar{h}$) depende del conjunto de conjuntos de entrenamiento elegido $\mathfrak{D}$. Si un x elemento del conjunto de prueba se encuentra en el conjunto $\mathfrak{D}$, será clasificado correctamente ya que $\bar{h}(x)$ será el valor de la etiqueta de x.

- Si x se encuentra en el conjunto de conjunto de entrenamientos posibles $\mathfrak{D}$ pero no en el conjunto de entrenamiento D elegido, pues habrá una varianza debido a la elección de un conjunto de entrenamiento u otro ya que $\bar{h}(x)$ será distinta a $h_D(x)$.

- Se puede afirmar que el algoritmo L tiende a hacer sobreajuste al vector de entrenamiento D.

Ejercicio 4. Si tomamos la función identidad ($\forall x \in R, id(x) = x$) como función de activación, entonces el perceptrón simple (esto es, una única neurona) es capaz de aprender la función booleana binaria f_{AND} ($f_{AND}(1,1) = 1, f_{AND}(0,1) = 0, f_{AND}(1,0) = 0, f_{AND}(0,0) = 0$). Si la respuesta es **Verdadero** tienes que dar un perceptrón que caracterice a f_{AND}. Si la respuesta es **Falso**, tienes que dar las razones de por qué no puede existir tal perceptrón.

La respuesta es **Falso.**



De la forma general de las neuronas:

$$y = g\left(\sum w_{ij} a_j\right)$$

$$y = id(w_1 x_1 + w_2 x_2 + b)$$

$$y = w_1 x_1 + w_2 x_2 + b$$

De la tabla de verdad de la compuerta AND:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Nos quedan las siguientes ecuaciones:

$$0 = w_1 * 0 + x_2 * 0 + b$$

$$\boxed{b = 0} \quad Ecuación\ 1$$

$$0 = w_1 * 0 + w_2 * 1 + b$$

$$\boxed{w_2 + b = 0} \quad Ecuación\ 2$$

$$0 = w_1 * 1 + w_2 * 0 + b$$

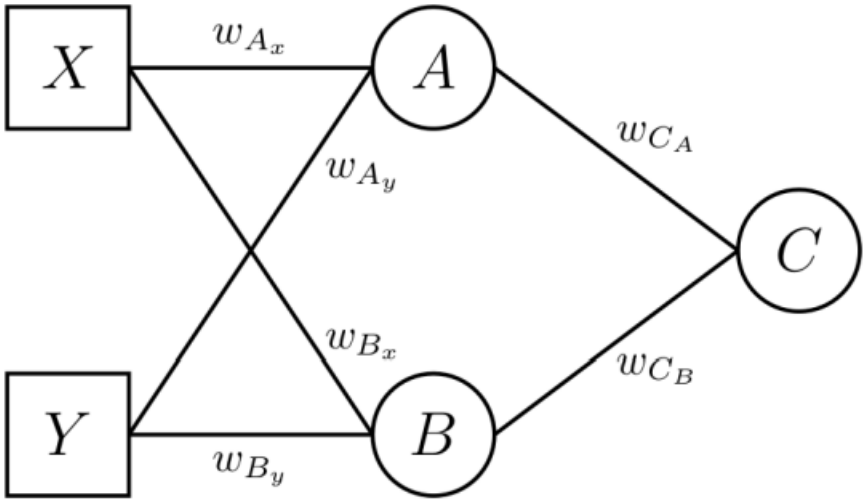
$$\boxed{w_1 + b = 0} \quad Ecuación\ 3$$

$$1 = w_1 * 1 + w_2 * 1 + b$$

$$\boxed{w_1 + w_2 + b = 1} \quad Ecuación\ 4$$

Se puede apreciar que para las primeras 3 ecuaciones, los valores que la satisfacen ($b = 0, w_2 = 0, w_1 = 0$) no pueden satisfacer la ecuación 4. Por esto, no se puede usar un perceptrón simple con la función identidad como función de activación para simular el comportamiento de una puerta lógica AND.

Ejercicio 5. Consideremos la red neuronal de la siguiente figura donde tomamos la función umbral como función de activación y considerando la entrada ficticia $a_0 = 1$ en todos los casos:



Determinar los valores de los pesos de la red neuronal para que clasifique como 1 la siguiente región geométrica:

(a) $\{(x,y) \in \mathbb{R}^2 \mid y \leq 1 + x \, o \, y \leq 1 - x\}$

(b) $\{(x,y) \in \mathbb{R}^2 \mid -1 \leq x - y \leq 1\}$

En ambos casos, explica con detalle el razonamiento que te ha llevado a la conclusión.

(a) $\{(x,y) \in \mathbb{R}^2 \mid y \leq 1 + x \, o \, y \leq 1 - x\}$

Se hará que la neurona C simule el funcionamiento de una puerta lógica OR. Así, se puede trabajar por separado la salida de la neurona A y la B.

En la neurona A:

$$y_A = umbral(w_{Ax}x + w_{Ay}y + b_A)$$

$$y_A = \begin{cases} 1 & si \quad w_{Ax}x + w_{Ay}y + b_A \geq 0 \\ 0 & si \quad w_{Ax}x + w_{Ay}y + b_A < 0 \end{cases}$$

Asignemos que la neurona A debe cumplir con la condición:

$$y \leq 1 + x$$

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

$$x - y + 1 \geq 0$$

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

$$x - y + 1 \geq 0$$

$$w_{Ax}x + w_{Ay}y + b_A \geq 0$$

$$1 * x + (-1) * y + 1 \geq 0$$

Así:

$$w_{Ax} = 1, \qquad w_{Ay} = -1, \qquad b_A = 1$$

En la neurona B:

$$y_B = umbral(w_{Bx}x + w_{By}y + b_B)$$

$$y_B = \begin{cases} 1 & si \quad w_{Bx}x + w_{By}y + b_B \geq 0 \\ 0 & si \quad w_{Bx}x + w_{By}y + b_B < 0 \end{cases}$$

Asignemos que la neurona B debe cumplir con la condición:

$$y \leq 1 - x$$

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

$$-x - y + 1 \geq 0$$

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

$$-x - y + 1 \geq 0$$

$$w_{Bx}x + w_{By}y + b_B \geq 0$$

$$(-1) * x + (-1) * y + 1 \geq 0$$

Así:

$$w_{Bx} = -1, \qquad w_{By} = -1, \qquad b_B = 1$$

Finalmente queremos que la neurona C simula una compuerta lógica OR.

$$y_C = umbral(w_{AC}y_A + w_{BC}y_B + b_C)$$

$$y_C = \begin{cases} 1 & si \quad w_{AC}y_A + w_{BC}y_B + b_C \geq 0 \\ 0 & si \quad w_{AC}y_A + w_{BC}y_B + b_C < 0 \end{cases}$$

De la tabla de verdad de la compuerta OR:

y_A	y_B	y_C
0	0	0
0	1	1
1	0	1
1	1	1

Nos quedan las siguientes desigualdades:

$$w_{AC} * 0 + w_{BC} * 0 + b_C < 0$$

$$\boxed{b_C < 0} \quad Desigualdad\ 1$$

$$w_{AC} * 0 + w_{BC} * 1 + b_C \geq 0$$

$$\boxed{w_{BC} + b_C \geq 0} \quad Desigualdad\ 2$$

$$w_{AC} * 1 + w_{BC} * 0 + b_C \geq 0$$

$$\boxed{w_{AC} + b_C \geq 0} \quad Desigualdad\ 3$$

$$w_{AC} * 1 + w_{BC} * 1 + b_C \geq 0$$

$$\boxed{w_{AC} + w_{BC} + b_C \geq 0} \quad Desigualdad\ 4$$

Los valores:

$$b_C = -1, \qquad w_{BC} = 1, \qquad w_{AC} = 1$$

Cumplen con todas las desigualdades.

Por lo que los valores finales de la red son:

$$w_{Ax} = 1, \qquad w_{Ay} = -1, \qquad b_A = 1$$

$$w_{Bx} = -1, \qquad w_{By} = -1, \qquad b_B = 1$$

$$w_{BC} = 1, \qquad w_{AC} = 1, \qquad b_C = -1$$

(b) $\{(x, y) \in \mathbb{R}^2 | -1 \leq x - y \leq 1\}$

Se hará que la neurona C simule el funcionamiento de una puerta lógica AND. Así, se puede trabajar por separado la salida de la neurona A y la B.

En la neurona A:

$$y_A = umbral(w_{Ax}x + w_{Ay}y + b_A)$$

$$y_A = \begin{cases} 1 & si \quad w_{Ax}x + w_{Ay}y + b_A \geq 0 \\ 0 & si \quad w_{Ax}x + w_{Ay}y + b_A < 0 \end{cases}$$

Asignemos que la neurona A debe cumplir con la condición:

$$x - y \geq -1$$

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

$$x - y + 1 \geq 0$$

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

$$x - y + 1 \geq 0$$

$$w_{Ax}x + w_{Ay}y + b_A \geq 0$$

$$1 * x + (-1) * y + 1 \geq 0$$

Así:

$$w_{Ax} = 1, \qquad w_{Ay} = -1, \qquad b_A = 1$$

En la neurona B:

$$y_B = umbral(w_{Bx}x + w_{By}y + b_B)$$

$$y_B = \begin{cases} 1 & si \quad w_{Bx}x + w_{By}y + b_B \geq 0 \\ 0 & si \quad w_{Bx}x + w_{By}y + b_B < 0 \end{cases}$$

Asignemos que la neurona B debe cumplir con la condición:

$$x - y \leq 1$$

Esta condición representa una recta en el plano, todos los puntos por debajo o coincidentes de esta recta serán clasificados como 1. Podemos reescribir esta condición de la forma en que están escritas las condiciones de la función umbral.

$$-x + y + 1 \geq 0$$

Vemos que esta forma tiene implícitamente escritos los valores que deben tener los pesos y el bias para que se cumpla la condición:

$$-x + y + 1 \geq 0$$

$$w_{Bx}x + w_{By}y + b_B \geq 0$$

$$(-1) * x + 1 * y + 1 \geq 0$$

Así:

$$w_{Bx} = -1, \qquad w_{By} = 1, \qquad b_B = 1$$

Finalmente queremos que la neurona C simule una compuerta lógica OR.

$$y_C = umbral(w_{AC}y_A + w_{BC}y_B + b_C)$$

$$y_C = \begin{cases} 1 & si \quad w_{AC}y_A + w_{BC}y_B + b_C \geq 0 \\ 0 & si \quad w_{AC}y_A + w_{BC}y_B + b_C < 0 \end{cases}$$

De la tabla de verdad de la compuerta AND:

y_A	y_B	y_C
0	0	0
0	1	0
1	0	0
1	1	1

Nos quedan las siguientes desigualdades:

$$w_{AC} * 0 + w_{BC} * 0 + b_C < 0$$

$$\boxed{b_C < 0} \quad Desigualdad\ 1$$

$$w_{AC} * 0 + w_{BC} * 1 + b_C < 0$$

$$\boxed{w_{BC} + b_C < 0} \quad Desigualdad\ 2$$

$$w_{AC} * 1 + w_{BC} * 0 + b_C < 0$$

$$\boxed{w_{AC} + b_C < 0} \quad Desigualdad\ 3$$

$$w_{AC} * 1 + w_{BC} * 1 + b_C \geq 0$$

$$\boxed{w_{AC} + w_{BC} + b_C \geq 0} \quad Desigualdad\ 4$$

Los valores:

$$b_C = -2, \qquad w_{BC} = 1, \qquad w_{AC} = 1$$

Cumplen con todas las desigualdades.

Por lo que los valores finales de la red son:

$$w_{Ax} = 1, \qquad w_{Ay} = -1, \qquad b_A = 1$$

$$w_{Bx} = -1, \qquad w_{By} = 1, \qquad b_B = 1$$

$$b_C = -2, \qquad w_{BC} = 1, \qquad w_{AC} = 1$$