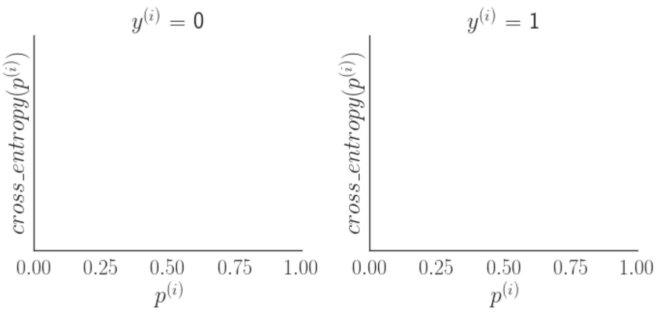
11. Regresión Logística (Clasificación)

Ejercicio 11.1. Para clasificación a través de la técnica de regresión logística, suele utilizarse la función “Binary Cross-Entropy” para definir el error de una predicción en particular. Este error se puede escribir como:

$$\text{Binary\_CE}(\mathbf{y}^{(i)}, \mathbf{p}^{(i)}) = \begin{cases} -\log(\mathbf{p}^{(i)}) & \text{si } \mathbf{y}^{(i)} = 1 \\ -\log(1 - \mathbf{p}^{(i)}) & \text{si } \mathbf{y}^{(i)} = 0 \end{cases}$$

(a) ¿A qué hacen referencias las variables $\mathbf{y}^{(i)}$ y $\mathbf{p}^{(i)}$ en este cálculo?

(b) Completar los gráficos dibujando el costo asociado al error según la clase original de la instancia:



(c) Explicar con sus palabras por qué es bueno que este costo esté cerca de cero y en qué caso se acerca a infinito.

(d) Expresar la fórmula completa del costo asociado a un conjunto de datos $\mathbf{X}$ y sus etiquetas $\mathbf{y}$.

Ejercicio 11.2. ¿Cuál es el valor esperado de predicción de un modelo de regresión logística que utiliza regularización L2 (Ridge) con λ tendiendo a infinito?

Ejercicio 11.3. Los gráficos de la Figura 3 han sido generados mediante la función sigmoidea: $sigm(z^{(i)}) = \frac{1}{1+e^{-z^{(i)}}}$. Para cada uno, $z^{(i)}$ fue calculado utilizando distintos w_0 y w_1 siguiendo la fórmula $z^{(i)} = w_0 + w_1 * x_1^{(i)}$.

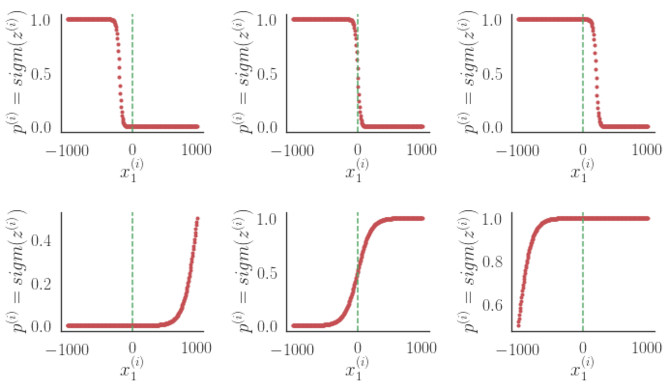


Figura 6: Sigmoideas

1. Determinar en qué dibujos w_1 es positivo y en cuáles negativo. ¿En qué afecta el signo de los distintos w_s en el caso de instancias multidimensionales?
2. Por fila, ordenar los dibujos según su valor de w_0.

Ejercicio 11.4.

1. Escribir el pseudocódigo de la función “Descenso por gradiente” para el caso de regresión logística, comentando brevemente qué se espera de cada argumento (junto a su tipo).
2. Explicar cómo se obtiene el gradiente de la función a minimizar.
3. Escribir el pseudocódigo de mini-batch gradient descent.

12. Redes Neuronales (FNNs)

Ejercicio 12.1. Verdadero o Falso

1. Un perceptrón simple con función de activación sigmoidea es equivalente a un modelo de regresión logística.
2. Un perceptrón simple con función de activación lineal es equivalente a una regresión lineal.
3. En un problema de regresión de $\mathbb{R}^p \to \mathbb{R}^q$, una red sin capas ocultas con q neuronas de salida aprendería los mismos pesos que entrenar q regresiones lineales simples. Ejemplo, predecir no sólo el valor de una casa sino también sus metros cuadrados según p atributos.
4. En un problema de regresión de $\mathbb{R}^p \to \mathbb{R}^q$, una red con capas ocultas con q neuronas de salida aprendería los mismos pesos que entrenar q regresiones lineales (con la misma arquitectura pero sólo 1 neurona de salida).

Ejercicio 12.2. Demostrar que una red neuronal con una neurona de salida, con función de activación lineal en todas las capas salvo la última, y activación sigmoidea en la última capa es equivalente a una regresión logística. ¿Tiene sentido utilizar una red con muchas capas en este caso?

Ejercicio 12.3. Dada una red neuronal con la siguiente arquitectura, en donde X hace referencia al tamaño de la entrada; $W^{[l]}$ a la matriz de pesos que conecta la capa l con la capa $l-1$; $A^{[l]}$ a las activaciones de la capa l

$$X \in \mathbb{R}^{5\times 3}, \quad W^{[1]} \in \mathbb{R}^{*\times 2}, \quad A^{[1]} \in \mathbb{R}^{*\times *}, \quad W^{[2]} \in \mathbb{R}^{*\times 1}, \quad A^{[2]} \in \mathbb{R}^{*\times *}, \quad W^{[3]} \in \mathbb{R}^{*\times 2}, \quad Y \in \mathbb{R}^{*\times *}$$

1. Escribir la fórmula denotada por esta red. Es decir, $Y = \ldots$ suponiendo funciones de activación g_i para toda capa intermedia, y g_o para la salida. Escribir dos versiones, una en la que los términos de bias están explícitos, una en la que no. Para lo segundo, utilizar la notación $ext(M)$ que simboliza agregar una columna de unos en primer lugar en la matriz M.
2. Completar los valores faltantes denotado con asteriscos (siempre refiriéndose a las versiones no extendidas).
3. Dibujar el esquema de la red neuronal.

Ejercicio 12.4. Construir a mano un perceptrón simple que resuelva el operador lógico AND: dadas dos variables X_1 y X_2, devuelve *True* o *False*. En las variables de entrada, interpretar $X_i = 1$ como *True* y $X_i = 0$ como *False*. Ídem para los operadores OR, NOR y NAND.

Ejercicio 12.5. Cantidad de parámetros de una red neuronal

1. Sea una red neuronal densa con *biases* con una capa de entrada con 3 neuronas, una capa oculta con 4 neuronas y una capa de salida con 2 neuronas. Realizar un diagrama de dicha red y calcular la cantidad de pesos en esta red neuronal.
2. Sea una red neuronal con densa M capas ocultas, donde la i-ésima capa oculta tiene N_i neuronas ($i = 1, 2, \ldots, M$), una capa de entrada con I neuronas y una capa de salida con O neuronas. Calcular la cantidad total de pesos en esta red neuronal.
3. Suponga que cada capa oculta tiene una cantidad fija de N_h neuronas para todas las capas. Determinar cómo crece la cantidad total de pesos en la red en términos de M, es decir, determinar la '*complejidad del modelo*' en términos de la cantidad de capas.

Ejercicio 12.6. En caso de estar resolviendo un problema de regresión:

- ¿Cuándo tiene sentido utilizar una función de activación lineal en la última capa?
- ¿Cuándo tiene sentido utilizar una función de activación ReLu en la última capa?

Ejercicio 12.7. Backpropagation
En este ejercicio, consideraremos una red neuronal con las siguientes definiciones:

- La entrada z de una neurona j en la capa l está dada por:

$$z_j^{[l]} = \sum_i w_{i,j}^{[l]} a_i^{[l-1]} + b_j^{[l]}$$

  donde $a_i^{[l]} = \sigma(z_i^{[l]})$, y σ es la función de activación. Aquí, i representa los índices de las neuronas de la capa anterior. Además $w_{i,j}^{[l]}$ representa el peso desde la neurona i de la capa $l-1$ a la neurona j de la capa l.

- La función de costo C para la red neuronal está definida como:

$$C = \sum_i \frac{1}{2}(y_i^{[L]} - a_i^{[L]})^2$$

donde L es la última capa de la red neuronal.

- Definimos el siguiente término para la última capa:

$$\frac{\partial C}{\partial z_j^{[L]}} = \delta_j^{[L]}$$

- Para la última capa, se tiene que:

$$\frac{\partial C}{\partial z_j^{[L]}} = \delta_j^{[L]}$$

Utilizando estas definiciones, resuelve los siguientes problemas:

1. Demuestra que:

$$\delta_j^{[L]} = (a_j^{[L]} - y_j^{[L]})\sigma'(z_j^{[L]})$$

2. Demuestra que:

$$\delta_j^{[l]} = \sigma'(z_j^{[l]}) \sum_k \delta_k^{[l+1]} w_{jk}^{[l+1]}$$

   Sugerencia: usa la siguiente igualdad:

$$\delta_j^{[l]} = \sum_k \frac{\partial C}{\partial z_k^{[l+1]}} \frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}}$$

   donde k es el número de neuronas de la capa $l+1$. Backpropagation

3. Demuestra que:

$$\frac{\partial C}{\partial w_{ij}} = \delta_j^{[l]} a_i^{[l-1]}$$

4. Demuestra que:

$$\frac{\partial C}{\partial b_j^{[l]}} = \delta_j^{[l]}$$

5. Conceptualmente, ¿qué representan los δ?

Para resolver los siguientes ejercicios, usar el playground de TensorFlow disponible en `http://playground.tensorflow.org`.

Ejercicio 12.8. Elegir el tercer dataset en el playground, que tiene dos grupos de puntos bien separados. Experimentar con diferentes configuraciones de capas ocultas, nodos y atributos, y estudiar el comportamiento de cada parte de la red durante el entrenamiento. Por ejemplo, usar:

- un solo atributo (X_1) y una capa con un nodo;
- un solo atributo (X_1) y dos o más capas con dos o más nodos;
- dos atributos (X_1, X_2) y una sola capa con un nodo; etc.

Ejercicio 12.9. Usando sólo los atributos X_1 y X_2, construir una configuración mínima (en cantidad de capas y de nodos por capa) de un perceptrón multicapa que resuelva el operador lógico XOR. Usar el segundo dataset del playground.

Ejercicio 12.10. Usando sólo los atributos X_1 y X_2, construir una perceptrón multicapa que pueda aprender los otros dos problemas no linealmente separables incluidos en el playground: el círculo azul rodeado de amarillo (fácil) y la doble espiral (difícil).

Ejercicio 12.11. Estudiar cómo impacta en el aprendizaje de los dos ejercicios anteriores la inclusión de otros atributos (por ejemplo, $sin(X_1)$), así como la elección de distintas funciones de activación (lineal, tanh, sigmoid).