Interpretabilidad del Deep Learning

Explicabilidad específica

Christian Oliva Moya

Contenido del curso

- Segunda semana:
  - **Explicabilidad específica para redes neuronales**
  - Métodos basados en gradientes: *Gradient x Input*
  - Métodos basados en relevancias: Layerwise Relevance Propagation
  - ¿Es LRP equivalente a *Gradient x Input*?
  - Dificultades del Deep Learning

Introducción (I)

- Hasta ahora hemos estado viendo métodos de explicabilidad genéricos, es decir, que se pueden aplicar a cualquier modelo
- Ahora vamos a ver métodos específicos para redes neuronales, que utilizan algunas técnicas que ya conocemos
- Vamos a empezar con la red neuronal más básica: la regresión lineal

Introducción (II)

- ¿Qué es una regresión lineal? Una única neurona

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- Lo que todos conocemos: Se puede saber si un atributo es estadísticamente significativo o no (p-valor de la t-Student, etc, etc)

Introducción (III)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- Desde el punto de vista de la explicabilidad:

¿Cuánto influye cada atributo de entrada a la predicción si X está normalizado?

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_D w_D$$

Introducción (IV)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- Desde el punto de vista de la explicabilidad:

¿Cuánto influye cada atributo de entrada a la predicción si X está normalizado?

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_D w_D$$

- Si wi = 0, da igual lo que valga xi que no modifica la salida
- Si wi > 0, la variación en la salida es directamente proporcional a xi
- Si wi < 0, la variación en la salida es inversamente proporcional a xi

Introducción (V)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- En otras palabras:

¿Qué pasa si aumento un atributo xi?

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_D w_D$$

Introducción (VI)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- En otras palabras:

¿Qué pasa si aumento un atributo xi?

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_D w_D$$

- Si wi = 0, no pasa nada
- Si wi > 0, si aumento el valor de xi, aumenta la predicción
- Si wi < 0, si aumento el valor de xi, disminuye la predicción

Introducción (VII)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- En otras palabras:

Ri = wi →Se comporta como LIME←Dirección de cambio

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_D w_D$$

- Si wi = 0, no pasa nada
- Si wi > 0, si aumento el valor de xi, aumenta la predicción
- Si wi < 0, si aumento el valor de xi, disminuye la predicción

Introducción (VIII)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

- ¿Y si?

Ri = |wi| →Se comporta como Permutación←Explicabilidad global

$$y = x_1 w_1 + x_2 w_2 + \ldots + x_D w_D$$

- Si |wi| = 0, no pasa nada
- Si |wi| > 0, si modifico xi, se modifica la predicción

Introducción (IX)

$$y = \sum_{i=1}^{D} x_i w_i + b$$

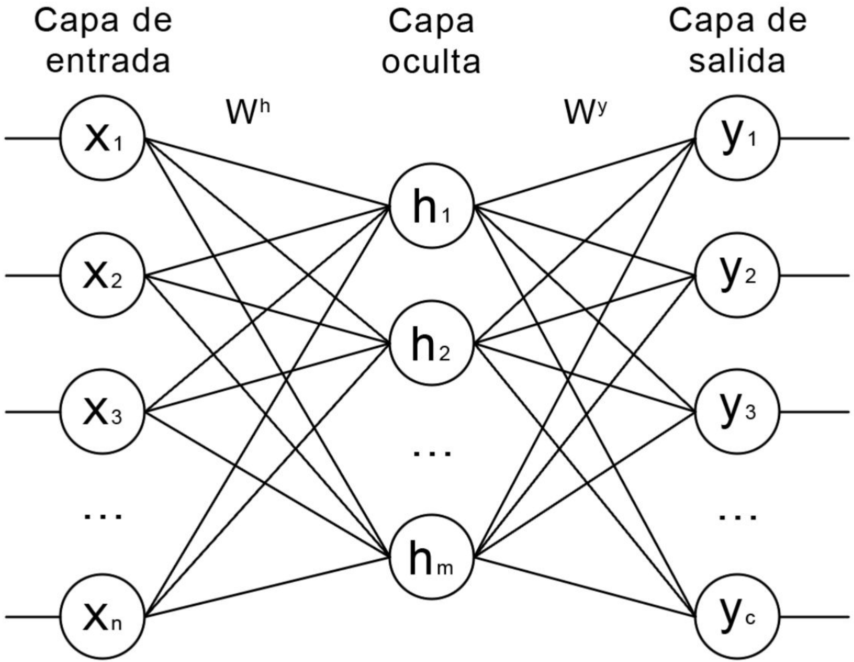
- En una regresión lineal (o logística):
  - **Ri = wi →Se comporta como LIME** ←Explicabilidad local
  - **Ri = |wi| →Se comporta como Permutación**←Explicabilidad global

Introducción (X)

- Vamos a verlo rápidamente en el notebook!
    - Notebook *3.1 Explicabilidad Específica - Regresión Lineal y Logística*

Redes neuronales (I)

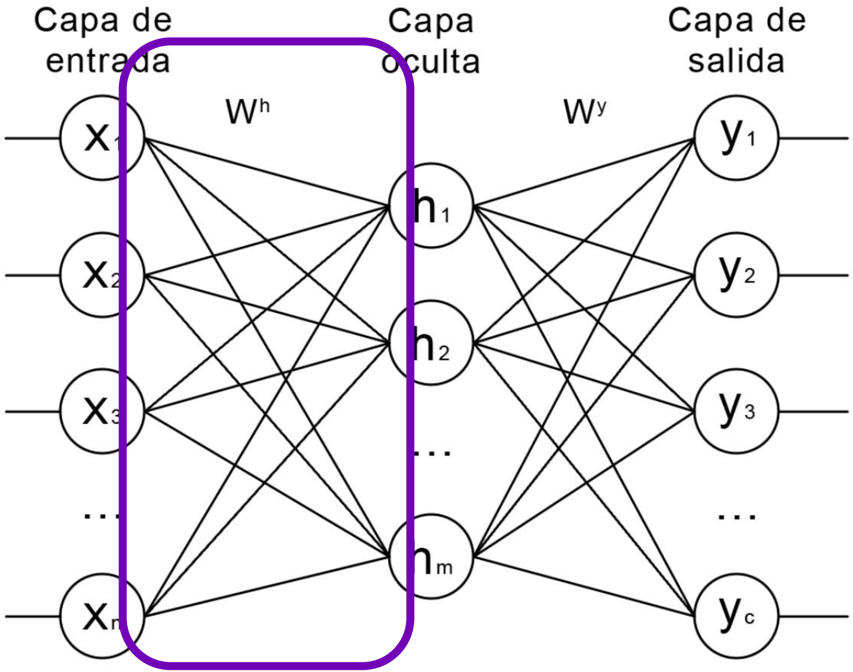
- Por supuesto, tenemos que generalizar a una red neuronal completa



Redes neuronales (II)

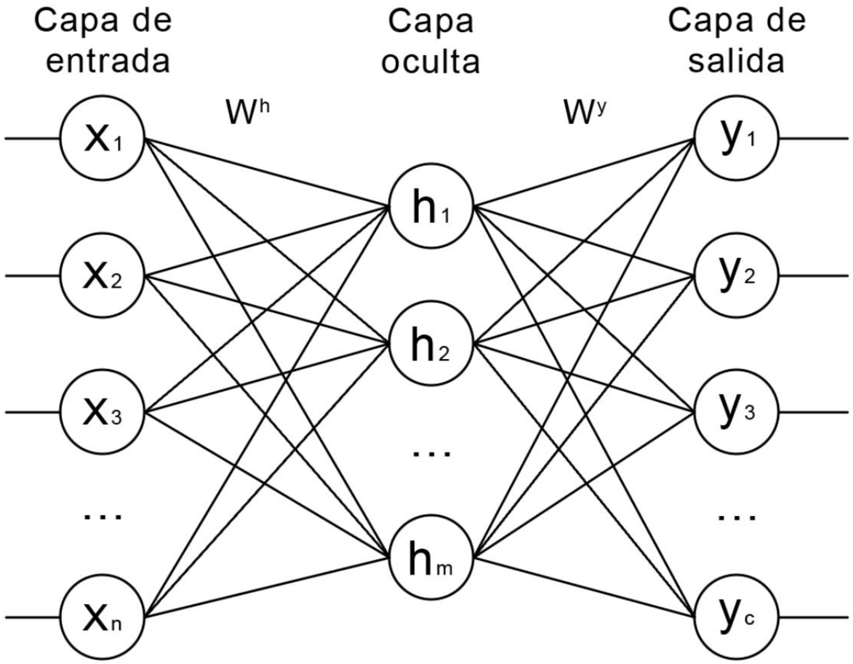
- Ahora no tenemos un único peso por cada entrada, sino una matriz de pesos Wh

  ¿Alguna idea?



Redes neuronales (III)

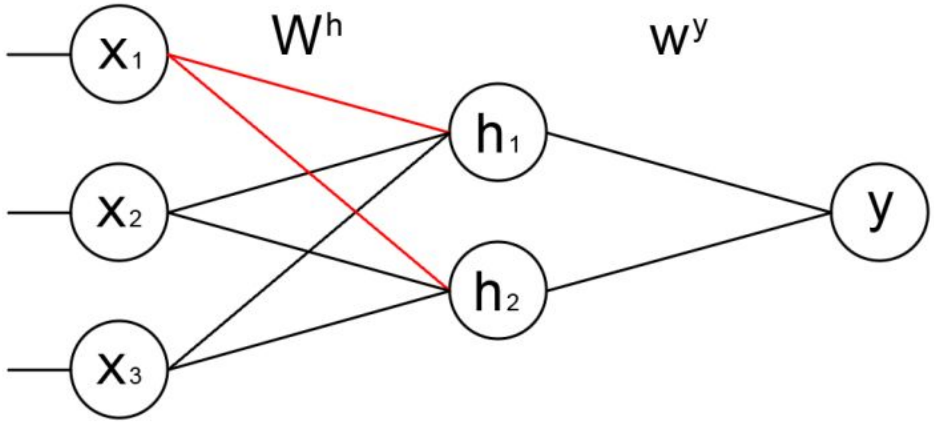
- Tenemos dos opciones:
    1. Analizar de forma global
    2. Analizar de forma local



Redes neuronales (IV)

1. Analizar de forma global

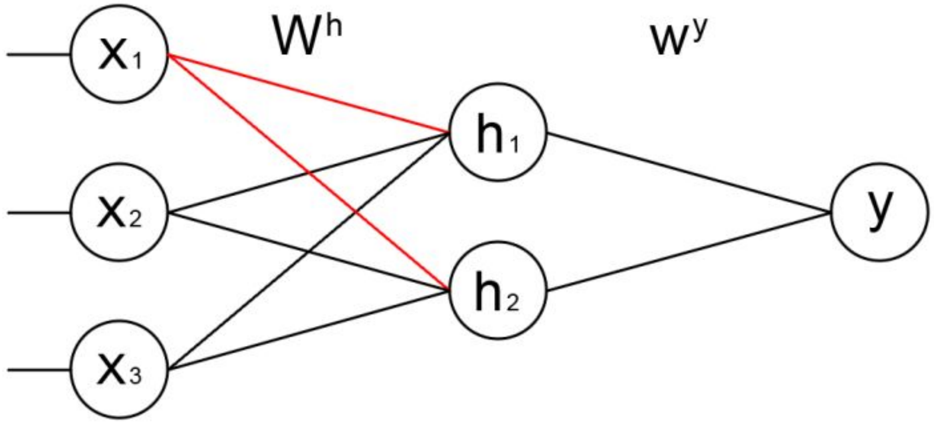
La relevancia de un atributo tiene que verse afectada por todas las conexiones a dicho atributo



Redes neuronales (V)

1. Analizar de forma global

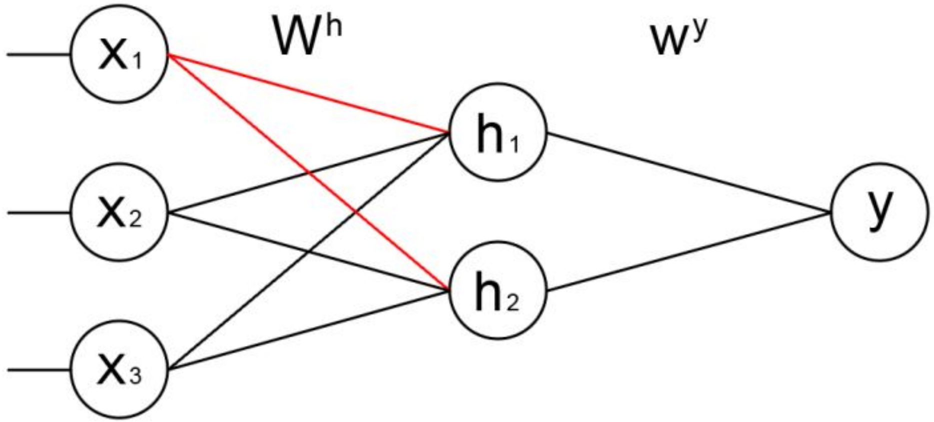
- **Idea**: Si en una regresión lineal Ri = |wi|
  - ¿Ahora tiene sentido hacer Ri = |w11| + |w12| en este ejemplo?



Redes neuronales (VI)

1. Analizar de forma global: **Suma de pesos en valor absoluto**

$$R_i = \sum_{j=1}^{h} |w_{ij}|$$

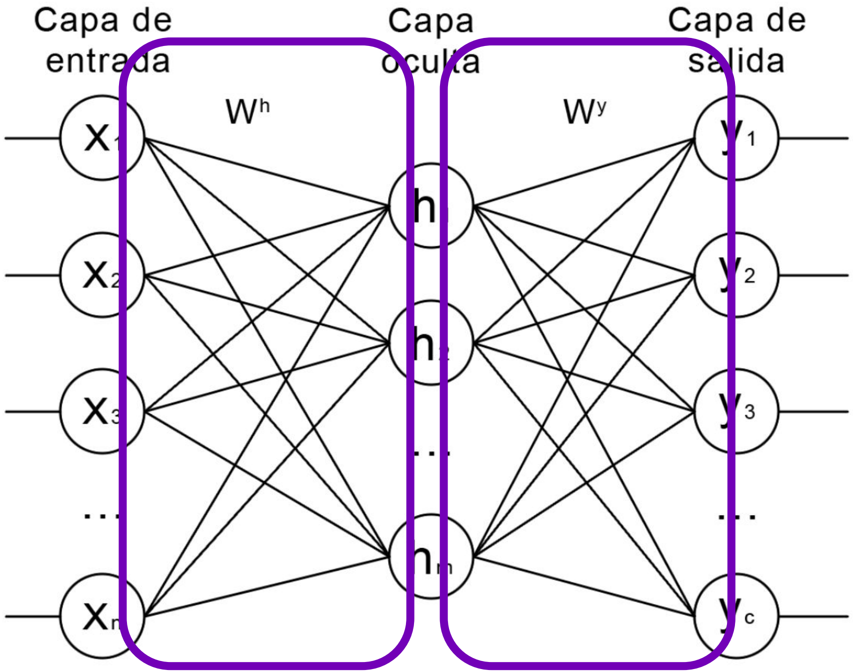


Redes neuronales (VII)

2. Analizar de forma local

- Antes la entrada y la salida estaban conectadas con un peso.
- Ahora hay varias capas.

¿Cómo combino todas las capas para tener una relevancia Rx?



Redes neuronales (VIII)

2. Analizar de forma local

- Pensemos de otra manera:

¿Tenemos alguna herramienta diferente que nos diga algo parecido a SHAP?

¿o a LIME?

- SHAP: Variación del modelo al anular atributos
- LIME: Dirección de cambio de un atributo

Contenido del curso

- Segunda semana:
    - Explicabilidad específica para redes neuronales
    - **Métodos basados en gradientes: *Gradient x Input***
    - Métodos basados en relevancias: Layerwise Relevance Propagation
    - ¿Es LRP equivalente a *Gradient x Input*?
    - Dificultades del Deep Learning

Métodos basados en gradientes (I)

2. Analizar de forma local

- ¿Qué representa la derivada de una función respecto a x?

$$\frac{\partial f(x)}{\partial x}$$

- La pendiente, es decir, la dirección de cambio de f(x) cuando cambia x

Métodos basados en gradientes (II)

2. Analizar de forma local

- El **gradiente** de la salida del modelo respecto a la entrada mide la tasa de cambio de la salida del modelo cuando se modifica la entrada

$$Ri = \frac{\partial y}{\partial x_i}$$

¡¡Se comporta parecido a LIME!!

Métodos basados en gradientes (III)

2. Analizar de forma local

- En una regresión lineal, ¿cuál es el gradiente?

$$y = \sum_{i=1}^{D} x_i w_i + b$$

$$Ri = \frac{\partial y}{\partial x_i}$$

Métodos basados en gradientes (IV)

2. Analizar de forma local

- En una regresión lineal, ¿cuál es el gradiente?

$$y = \sum_{i=1}^{D} x_i w_i + b$$

$$Ri = \frac{\partial y}{\partial x_i} = w_i$$

Coincide con la relevancia que habíamos dicho antes!!

Métodos basados en gradientes (V)

2. Analizar de forma local

$$Ri = \frac{\partial y}{\partial x_i}$$

- Esta técnica se llama **Saliency Maps** [1]
- El gradiente respecto a la entrada se puede calcular **sea cual sea** la red neuronal que estamos intentando explicar
- En una red neuronal, tenemos la suerte de que los gradientes se calculan mediante **diferenciación automática** con Tensorflow o Pytorch

[1] K. Simonyan, A. Vedaldi, and A. Zisserman. “Deep Inside Convolutional Networks: Visualizing Image Classification Models and Saliency Maps”, 2013.

Métodos basados en gradientes (VI)

- Si Saliency Maps se comporta como LIME, **Gradient x Input** se comporta como SHAP

$$Ri = x_i \frac{\partial y}{\partial x_i}$$

- Esta técnica es la más conocida de las basadas en descenso por gradiente. Hay variaciones más complejas, pero siguen la misma idea

Métodos basados en gradientes (VII)

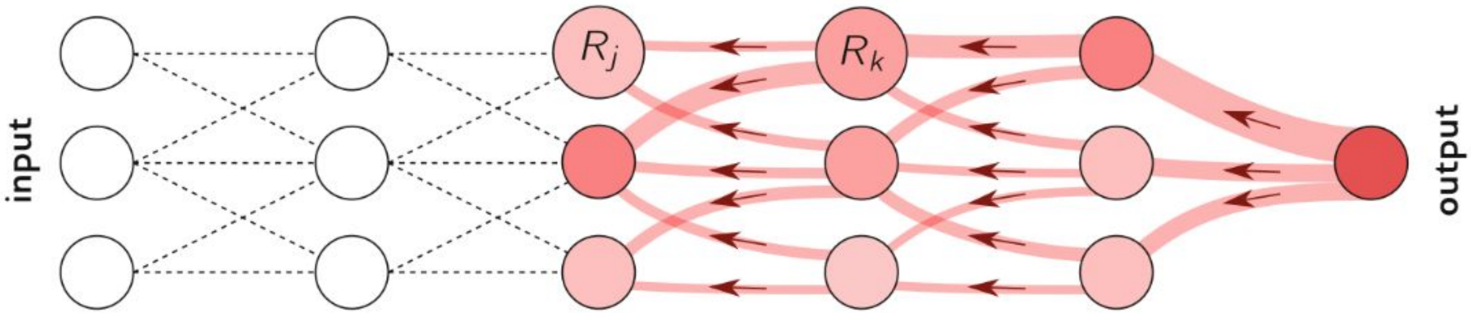
- Vamos a verlo rápidamente en el notebook!
  - Notebook *3.2 Explicabilidad Específica - Redes*

Contenido del curso

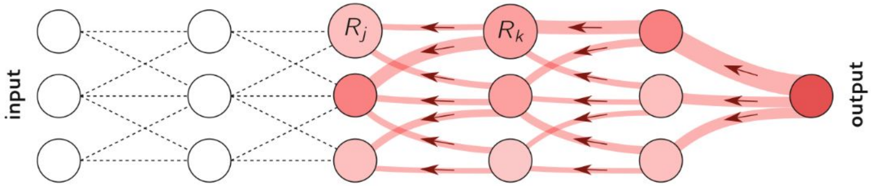
- Segunda semana:
  - Explicabilidad específica para redes neuronales
  - Métodos basados en gradientes: *Gradient x Input*
  - **Métodos basados en relevancias: Layerwise Relevance Propagation**
  - ¿Es LRP equivalente a *Gradient x Input*?
  - Dificultades del Deep Learning

LRP (I)

- LRP es una técnica **alternativa** al cálculo de relevancias basadas en gradientes que aporta una explicabilidad **local** y **específica** para redes
- Se propaga la relevancia por la red de forma similar a la retropropagación



LRP (II)

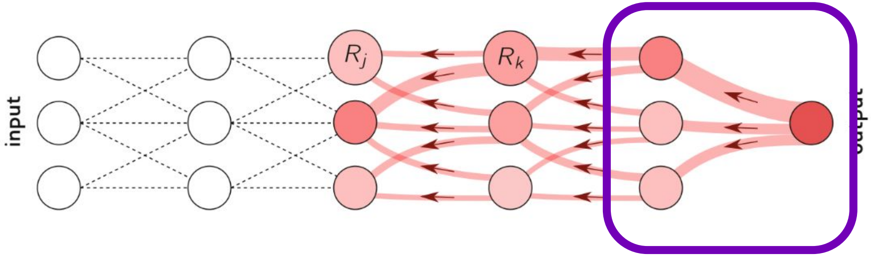


- Comenzando en la capa de salida, LRP propaga la relevancia hacia atrás
- La regla básica de propagación, conocida como LRP-0, es la siguiente:

$$R_i = \sum_k \frac{a_i w_{ik} R_k}{\sum_j a_j w_{jk}}$$

Vamos a desglosarla poco a poco para entenderla

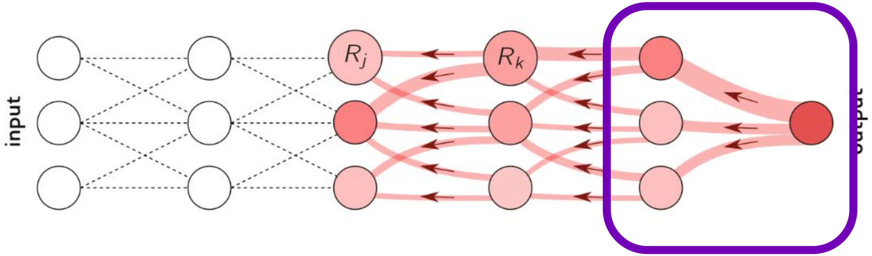
LRP (III)



- En la capa de salida:
  - **Definimos R=y** de la clase real
  - a_i w_i define la conexión entre la neurona i y la salida
  - El sumatorio del denominador es la preactivación de la salida

$$R_i = \frac{a_i w_i y}{\sum_j a_j w_j}$$

LRP (IV)

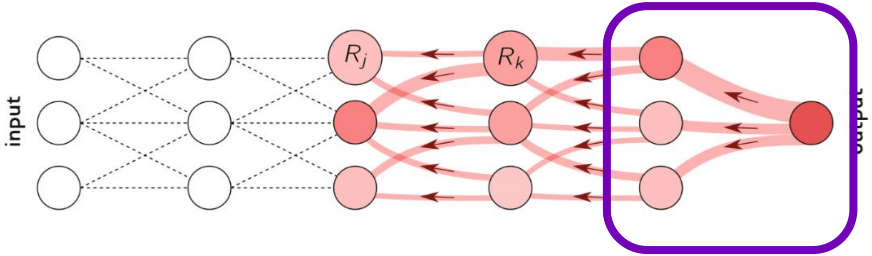


- Dejadme reescribirlo de otra manera:

$$R_i = y \frac{a_i w_i}{\sum_j a_j w_j}$$

- ¿Qué representa la fracción? **Es la normalización de las componentes de la preactivación**

LRP (V)



- **Cada R_i se multiplica luego por y**

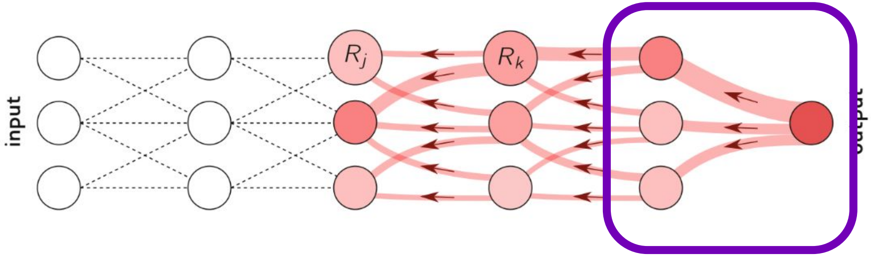
$$R_i = y\frac{a_i w_i}{\sum_j a_j w_j}$$

$$R_1 = \frac{a_1 w_1}{a_1 w_1 + a_2 w_2 + a_3 w_3}$$

$$R_2 = \frac{a_2 w_2}{a_1 w_1 + a_2 w_2 + a_3 w_3}$$

$$R_3 = \frac{a_3 w_3}{a_1 w_1 + a_2 w_2 + a_3 w_3}$$

LRP (VI)

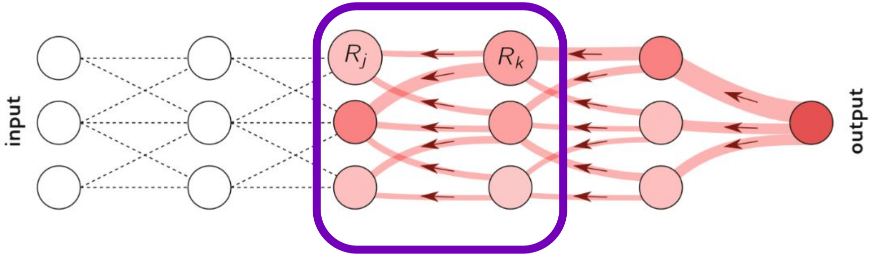


- En circuitos eléctricos, esto se conoce como **divisor de corriente**. Se distribuye por cada conexión una parte de la corriente total de entrada.

$$R_1 = \frac{a_1 w_1}{a_1 w_1 + a_2 w_2 + a_3 w_3} \qquad R_2 = \frac{a_2 w_2}{a_1 w_1 + a_2 w_2 + a_3 w_3}$$

$$R_3 = \frac{a_3 w_3}{a_1 w_1 + a_2 w_2 + a_3 w_3}$$

LRP (VII)



- En el resto de capas sucede lo mismo

$$R_i = \sum_k \frac{a_i w_{ik} R_k}{\sum_j a_j w_{jk}}$$

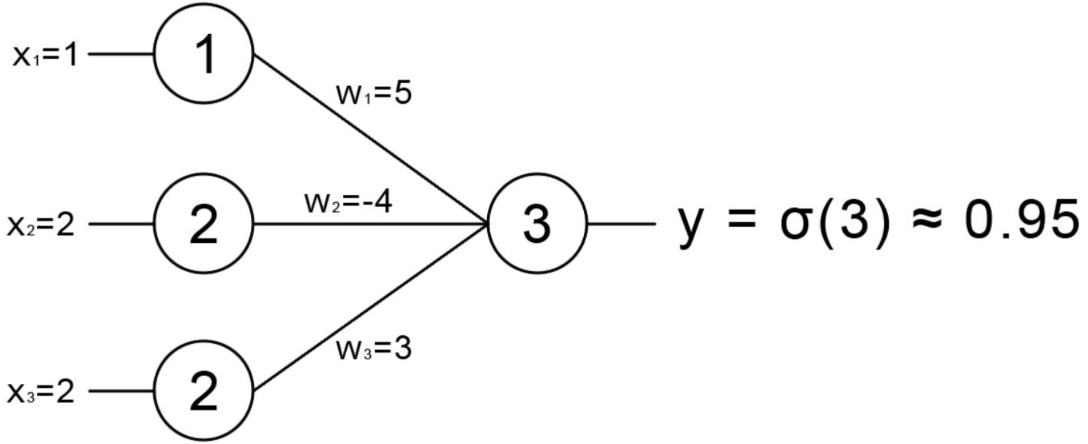
Las gallinas que entran por las que salen

Al final, si entraron 10 gallinas, tienen que salir 10 gallinas

Ley de conservación de relevancia

LRP (VIII)

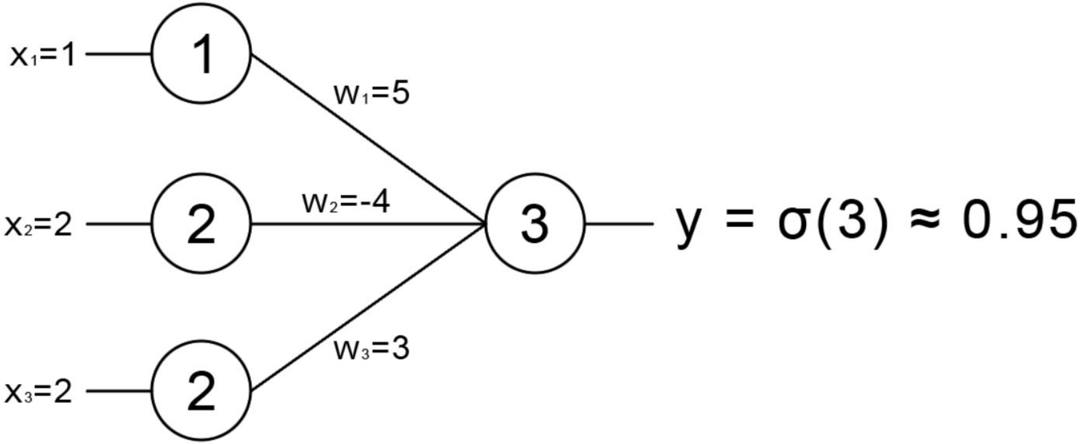
- Ejemplo numérico



$$
\begin{aligned}
R_1 &= \frac{1 \times 5}{1 \times 5 - 2 \times 4 + 2 \times 3} \times 0{,}95 = \frac{5}{3} \times 0{,}95 = 1{,}67 \times 0{,}95 = 1{,}58 \\
R_2 &= \frac{2 \times (-4)}{1 \times 5 - 2 \times 4 + 2 \times 3} \times 0{,}95 = \frac{-8}{3} \times 0{,}95 = -2{,}67 \times 0{,}95 = -2{,}53 \\
R_3 &= \frac{2 \times 3}{1 \times 5 - 2 \times 4 + 2 \times 3} \times 0{,}95 = \frac{6}{3} \times 0{,}95 = 2 \times 0{,}95 = 1{,}9
\end{aligned}
$$

LRP (IX)

- Ejemplo numérico

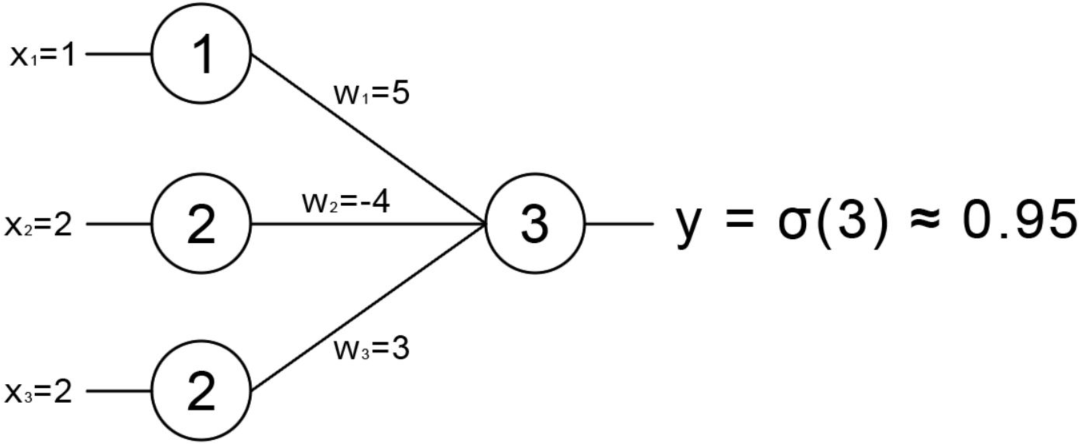


$$
\begin{aligned}
R_1 &= \frac{1 \times 5}{1 \times 5 - 2 \times 4 + 2 \times 3} \times 0{,}95 = \frac{5}{3} \times 0{,}95 = 1{,}67 \times 0{,}95 = 1{,}58 \\
R_2 &= \frac{2 \times (-4)}{1 \times 5 - 2 \times 4 + 2 \times 3} \times 0{,}95 = \frac{-8}{3} \times 0{,}95 = -2{,}67 \times 0{,}95 \boxed{= -2{,}53} \\
R_3 &= \frac{2 \times 3}{1 \times 5 - 2 \times 4 + 2 \times 3} \times 0{,}95 = \frac{6}{3} \times 0{,}95 = 2 \times 0{,}95 = 1{,}9
\end{aligned}
$$

R2 Negativamente relevante

LRP (X)

- Ejemplo numérico



$$
\begin{aligned}
R_1 &= \frac{1\times 5}{1\times 5 - 2\times 4 + 2\times 3} \times 0{,}95 = \frac{5}{3} \times 0{,}95 = 1{,}67 \times 0{,}95 \boxed{= 1{,}58} \\
R_2 &= \frac{2\times(-4)}{1\times 5 - 2\times 4 + 2\times 3} \times 0{,}95 = \frac{-8}{3} \times 0{,}95 = -2{,}67 \times 0{,}95 = -2{,}53 \\
R_3 &= \frac{2\times 3}{1\times 5 - 2\times 4 + 2\times 3} \times 0{,}95 = \frac{6}{3} \times 0{,}95 = 2 \times 0{,}95 \boxed{= 1{,}9}
\end{aligned}
$$

R1 y R3 Positivamente relevantes

LRP (XI)

- Hemos visto la regla básica LRP-0.
- Otras reglas LRP:
    - **LRP-ε**
    - LRP-αβ
    - LRP-γ

$$\boxed{R_i = \sum_k \frac{a_i w_{ik} R_k}{\epsilon + \sum_j a_j w_{jk}}}$$

LRP (XII)

- Hemos visto la regla básica LRP-0.
- Otras reglas LRP:
  - LRP-ε
  - **LRP-αβ**
  - LRP-γ

$$\boxed{R_i = \sum_k \left(\alpha \frac{(a_i w_{ik})^+}{\sum_j (a_j w_{jk})^+} - \beta \frac{(a_i w_{ik})^-}{\sum_j (a_j w_{jk})^-} \right) R_k}$$

LRP (XIII)

- Hemos visto la regla básica LRP-0.
- Otras reglas LRP:
  - LRP-ε
  - LRP-αβ
  - **LRP-γ**

$$\boxed{R_i = \sum_k \frac{a_i(w_{ik} + \gamma w_{ik}^{+}) R_k}{\sum_j a_j (w_{jk} + \gamma w_{jk}^{+})}}$$

Contenido del curso

- Segunda semana:
  - Explicabilidad específica para redes neuronales
  - Métodos basados en gradientes: *Gradient x Input*
  - Métodos basados en relevancias: Layerwise Relevance Propagation
  - **¿Es LRP equivalente a *Gradient x Input*?**
  - Dificultades del Deep Learning

¿LRP es equivalente a Gradient x Input? (I)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = \sum_k \frac{a_i w_{ik}}{\sum_j a_j w_{jk}} R_k$$

Puedo reescribirlo como

$$R_i = a_i \sum_k \frac{w_{ik}}{\sum_j a_j w_{jk}} R_k$$

¿LRP es equivalente a Gradient x Input? (II)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{w_{ik}}{\sum_j a_j w_{jk}} R_k$$

Puedo reescribirlo como

$$R_i = a_i \sum_k \frac{w_{ik}}{z_k} R_k \quad \text{siendo} \quad z_k = \sum_j a_j w_{jk}$$

¿LRP es equivalente a Gradient x Input? (III)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{w_{ik}}{z_k} R_k$$

Puedo reescribirlo como

$$R_i = a_i \sum_k w_{ik} \frac{R_k}{z_k}$$

¿LRP es equivalente a Gradient x Input? (IV)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k w_{ik} \frac{R_k}{z_k}$$

Puedo reescribirlo como

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{R_k}{z_k}$$

¿LRP es equivalente a Gradient x Input? (V)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{R_k}{z_k}$$

Por otro lado, en la capa de salida siempre se asume que $R_k = z_k = y$

$$R_k = h_k \frac{\partial y}{\partial h_k}$$

¿LRP es equivalente a Gradient x Input? (VI)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{R_k}{z_k} \qquad\qquad R_k = h_k \frac{\partial y}{\partial h_k}$$

Si combinamos los dos resultados

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{h_k}{z_k} \frac{\partial y}{\partial h_k}$$

¿LRP es equivalente a Gradient x Input? (VII)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{h_k}{z_k} \frac{\partial y}{\partial h_k}$$

Podemos reescribirlo como

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{f(z_k)}{z_k} \frac{\partial y}{\partial h_k}$$

¿LRP es equivalente a Gradient x Input? (VIII)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{f(z_k)}{z_k} \frac{\partial y}{\partial h_k}$$

¿Qué sucede en el caso f(x) = relu(x)?

$$\frac{h_k}{z_k} = \frac{relu(z_k)}{z_k} = \left\{ \begin{array}{ll} 1, & \text{si } z_k > 0 \\ 0, & \text{si } z_k \leq 0 \end{array} \right\} = \frac{\partial h_k}{\partial z_k}$$

¿LRP es equivalente a Gradient x Input? (IX)

- Se tiene que cumplir una condición para que sean equivalentes

$$R_i = a_i \sum_k \frac{\partial z_k}{\partial a_i} \frac{\partial h_k}{\partial z_k} \frac{\partial y}{\partial h_k}$$

La expresión anterior es equivalente a

$$R_i = a_i \frac{\partial y}{\partial a_i}$$

¿LRP es equivalente a Gradient x Input? (X)

- Se tiene que cumplir una condición para que sean equivalentes

Si tenemos que

$$R_k = h_k \frac{\partial y}{\partial h_k} \qquad R_i = a_i \frac{\partial y}{\partial a_i}$$

Podemos generalizar ya que funciona para cualquier capa!!!

¿LRP es equivalente a Gradient x Input? (XI)

- Se tiene que cumplir una condición para que sean equivalentes

Particularmente

$$R_i = a_i \frac{\partial y}{\partial a_i} \quad \rightarrow \quad R_i = x_i \frac{\partial y}{\partial x_i}$$

Es la misma expresión que Gradient x Input

¿LRP es equivalente a Gradient x Input? (XII)

- Se tiene que cumplir una condición para que sean equivalentes

¿Cuál es la condición?

$$\frac{h_k}{z_k} = \frac{relu(z_k)}{z_k} = \left\{ \begin{array}{ll} 1, & \text{si } z_k > 0 \\ 0, & \text{si } z_k \leq 0 \end{array} \right\} = \frac{\partial h_k}{\partial z_k}$$

Todas las funciones de activación tienen que ser ReLU

LRP mediante gradientes (I)

- Sabiendo esa condición, podemos aprovecharnos y forzar esta situación en el cálculo de las derivadas de Tensorflow

```python
def lrp(f):

    @tf.custom_gradient
    def f_custom_grad(x):
        output = f(x)

        def backward(dy):
            custom_grad = f(x)/x
            return dy * custom_grad

        return output, backward
    return f_custom_grad
```

LRP mediante gradientes (II)

- Vamos a verlo en el notebook!
    - Notebook *3.2 Explicabilidad Específica - Redes*