**Andrés Orrego Pérez - C.C 1216725315**

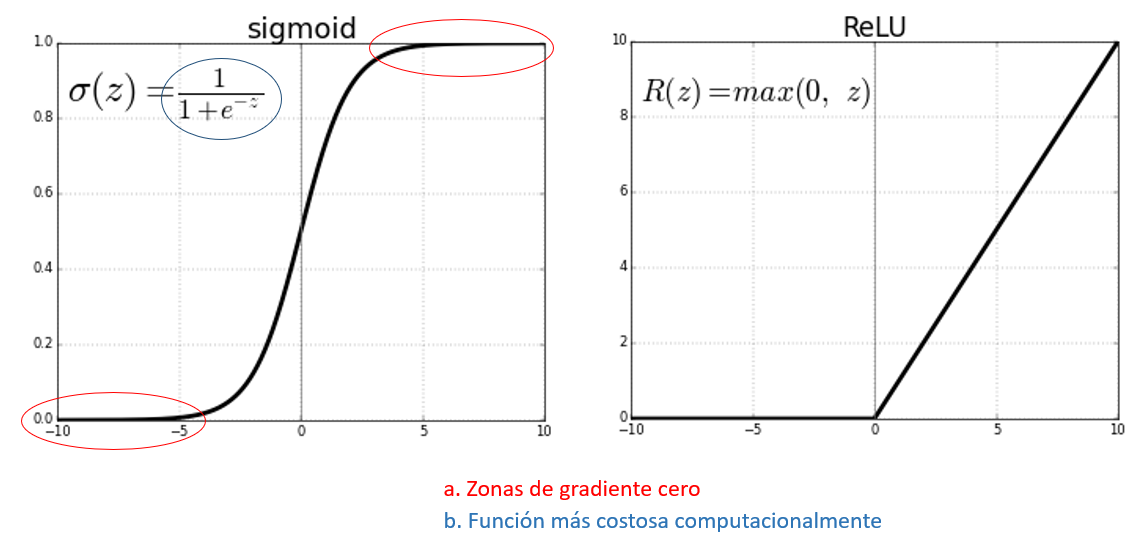
**Diego Eusse Naranjo - C.C 1037655233**

**Daniel García Alzate - C.C 1036955423**

**FOLLOW UP 3 (Activation Functions)**

Lea los artículos propuestos, busque más información si es necesario y debatir:

1. **¿Por qué la función de transferencia o de activación famosa llamada ReLu es mejor que la Sigmoide?**

****

R/ Existen diferentes funciones de activación en las redes neuronales, dos de ellas son la función ReLu y Sigmoide. Algunas características de estas funciones son:

**Función Sigmoide**

****

* Problema del **‘Desvanecimiento del Gradiente’**
* Convergencia lenta.
* No está centrada en el cero.
* Acotada entre 0 y 1.
* Buen rendimiento en la última capa.

**Función ReLu**



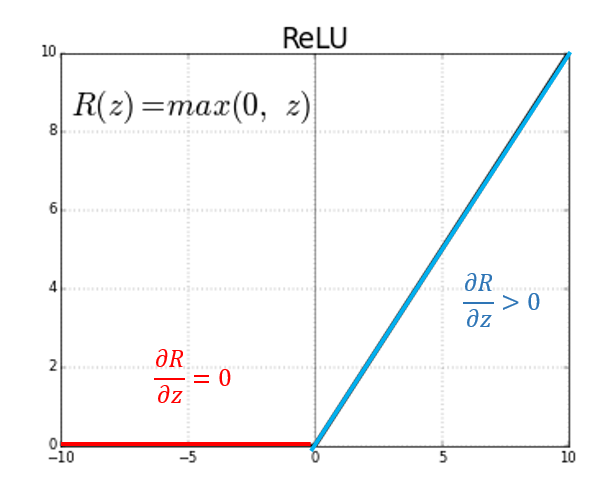
* Activación Sparse: Se activa ante entradas positivas.
* No acotada.
* Genera muchas neuronas con valor de 0, ante posibles entradas negativas.
* Buen desempeño para algoritmos basados en imágenes.
* Ideal para redes convolucionales.

El uso de la función ReLu se ha popularizado más que el de la Sigmoide debido a:

* Al sufrir de “vanishing gradient”, la función Sigmoide reduce considerablemente el gradiente al tener entradas muy grandes (debido a que la derivada de la función Sigmoide en un punto muy alejado de 0 tiende a 0). Este problema se soluciona al implementar la función ReLu.
* La función ReLu es más simple que la función Sigmoide lo que implica que requiere menos recursos computacionales.
* En la práctica, la función ReLu muestra mejor desempeño en la convergencia que la función Sigmoide.

A pesar de esto, la función ReLu también tiene desventajas como la llamada “blow up activation” que se refiere al hecho de que no se puede restringir la salida de la neurona ya que la salida es la misma entrada (para x>0), o que si la entrada es menor que 0, la neurona “muere” ya que la salida sería 0. Sin embargo, estos problemas no son tan significativos como para destronar a la función ReLu sobre la Sigmoide y se han desarrollado modificaciones sobre la función ReLu que han permitido solventarlas.

1. **¿Por qué ReLu es no-lineal, o acaso sí es?**



R/

En matemáticas (álgebra lineal) una función se considera lineal siempre que una función *f* tiene la siguiente propiedad:



Siguiendo con esta propiedad, es posible observar que la función de activación Relu no es lineal porque no cumple con la siguiente comprobación:



Por tanto, por definición, ReLU no es lineal.

1. **¿Cuáles cree usted que son las implicaciones de que la función Sigmoide sufra de “Vanishing Gradient” (Desvanecimiento del Gradiente)?**

Cuando se trabaja con ciertas funciones de activación como la sigmoide, es común el tener que enfrentar el problema de desvanecimiento del gradiente. Este problema se debe a que cada uno de los pesos de la red neuronal recibe una actualización proporcional a la [derivada parcial](https://es.wikipedia.org/wiki/Derivada_parcial) de la función de error con respecto al peso actual en cada iteración de entrenamiento. El problema radica en que, en algunos casos, el gradiente se irá desvaneciendo a valores muy pequeños, impidiendo eficazmente el peso de cambiar su valor.

En el peor de los casos, esto puede impedir que la red neuronal continúe su entrenamiento, debido a que, por ejemplo en el caso de la función sigmoide, su derivada siempre estará acotada entre 0 y 0.25, y cuando se tiene una red con muchas capas, el valor de gradiente cada vez es más cercano a 0 ya que se están multiplicando valores que son más pequeños que 1 en múltiples ocasiones.

Como consecuencia de este problema, las primeras capas de una red neuronal son las más lentas y difíciles de entrenar. Ya que el valor del gradiente que se usa para actualizarlas en cada iteración del entrenamiento es muy pequeño. Y esto causa otro problema adicional: que, si las primeras capas no están bien entrenadas, el problema se arrastra a las capas posteriores.

Análogo con el problema del desvanecimiento de gradiente, cuando se usan funciones de activación cuyas derivadas pueden tomar valores más grandes (mayores a 1), uno de los riesgos es encontrar el denominado **problema de gradiente explosivo**.

Es por ello que debe prestarse especial atención al manejo y parametrización de las funciones de activación, pues como se muestra, pueden impactar de manera drástica al entrenamiento y al desempeño de una red neuronal.

Article 1:

<https://www.datasciencecentral.com/profiles/blogs/deep-learning-advantages-of-relu-over-sigmoid-function-in-deep#:~:text=Advantage%3A,exponential%20operations%20as%20in%20Sigmoids>

Article 2:

https://datascience.stackexchange.com/questions/26475/why-is-relu-used-as-an-activation-function#:~:text=As%20a%20simple%20 definition%2C%20 linear,the%20inputs%20in%20its%20domain.&text=bf(y)-,ReLU%20is%20not%20linear.,consequence%20of%20this%20non%2Dlinearity.

Fuentes adicionales que pueden utilizar:

* <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
* <http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>
* <https://www.youtube.com/watch?v=WTQaox7NKNA>