Deep Learning

Perceptrones altualmente no se utilizan

ejecuta la funcion umbral, si supera lo esperado da como resultado 1 sino lo supera, da 0

pesos psinacticos

nos inventamos una neuroma que siempre va a ser 1 y la vamos a multiplicar por b (esta es inventada)

en total la neurona de lsalida recive sumatorio de las XiWi + b

si el resultado es menor que 0, de como resultado 0, y al contrario dará 1

a super k es0 si es un gato y 1 si no lo es

google tenia 10 millones de fotos⇒ a super k 10 millones

b es el sesgo

explicacion:

premisas

$$b = w_0 \text{ and } x_0 = +1.$$

si me equivoco mucho, tendré que cambiar mucho,

$$(a^k - \Phi_{b,w}(x^k))$$

Si me he equivocado mucho, tengo yo mucha culpa

pero si no tengo peso, no es mi culpa el que no salga el resultado esperado

$$x^k{}_i$$

$$\Delta w_i = (a^k - \Phi_{b,w}(x^k))\, x^k{}_i$$

salida deseada menos salida producida por valor de (X cuadrado de i)

Feedforward Networks I

tiene una o varias capas intermedias

cada una de las neuronas las vamos a poner un sesgo

las capas intermedias no tienen porque tener el mismo numero de neuronas

solo tiene conexiones hacia las capas de delante, ni a si mismas ni hacia atrás

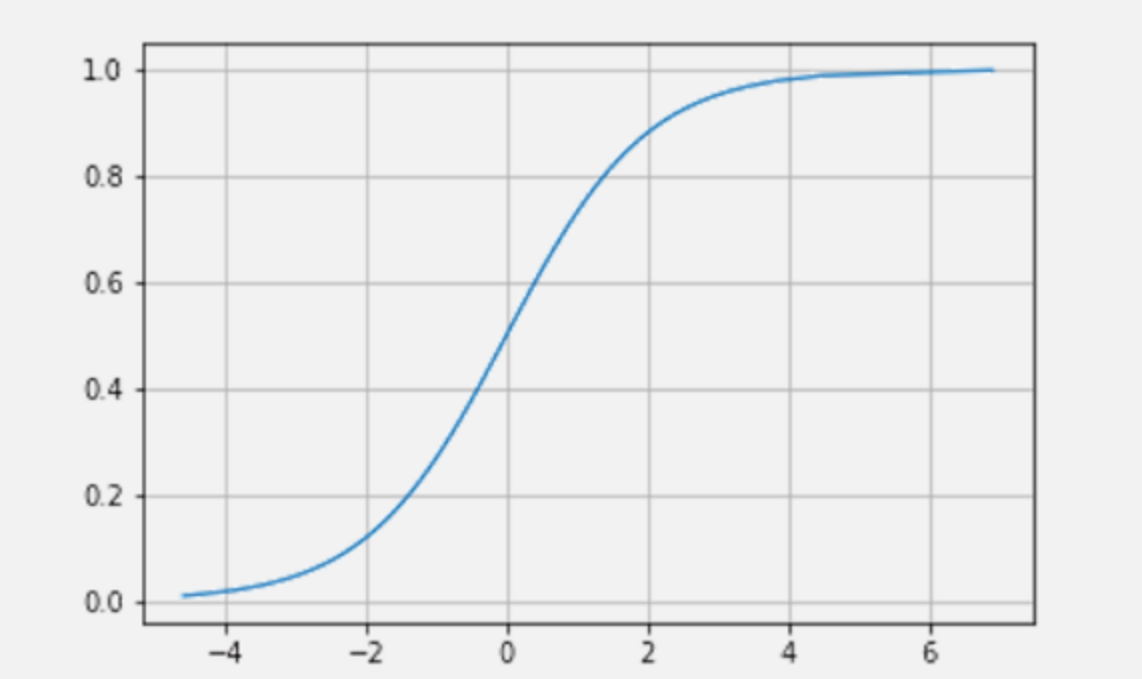
mas de una salida

función de activación, lo que ejecuta la neurona con la entrada

Función logística

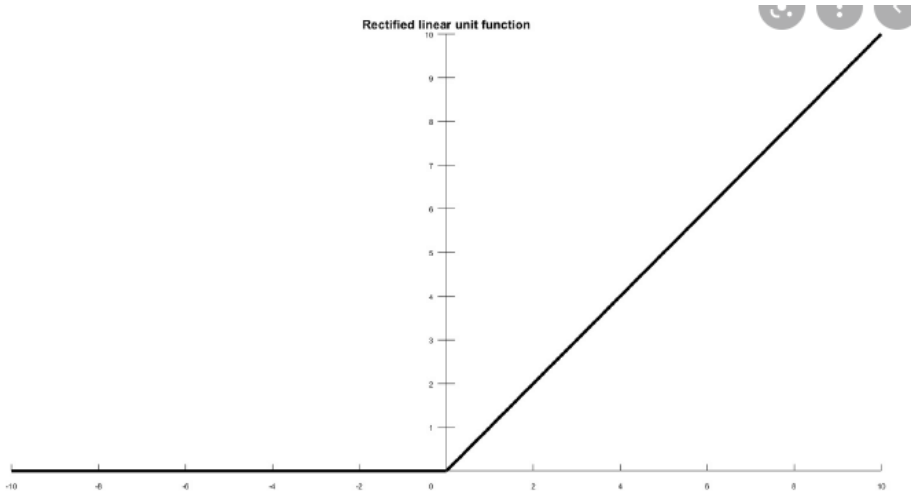
$$f(x) = \frac{1}{1 + e^{-x}}$$

tendrá la siguiente respuesta



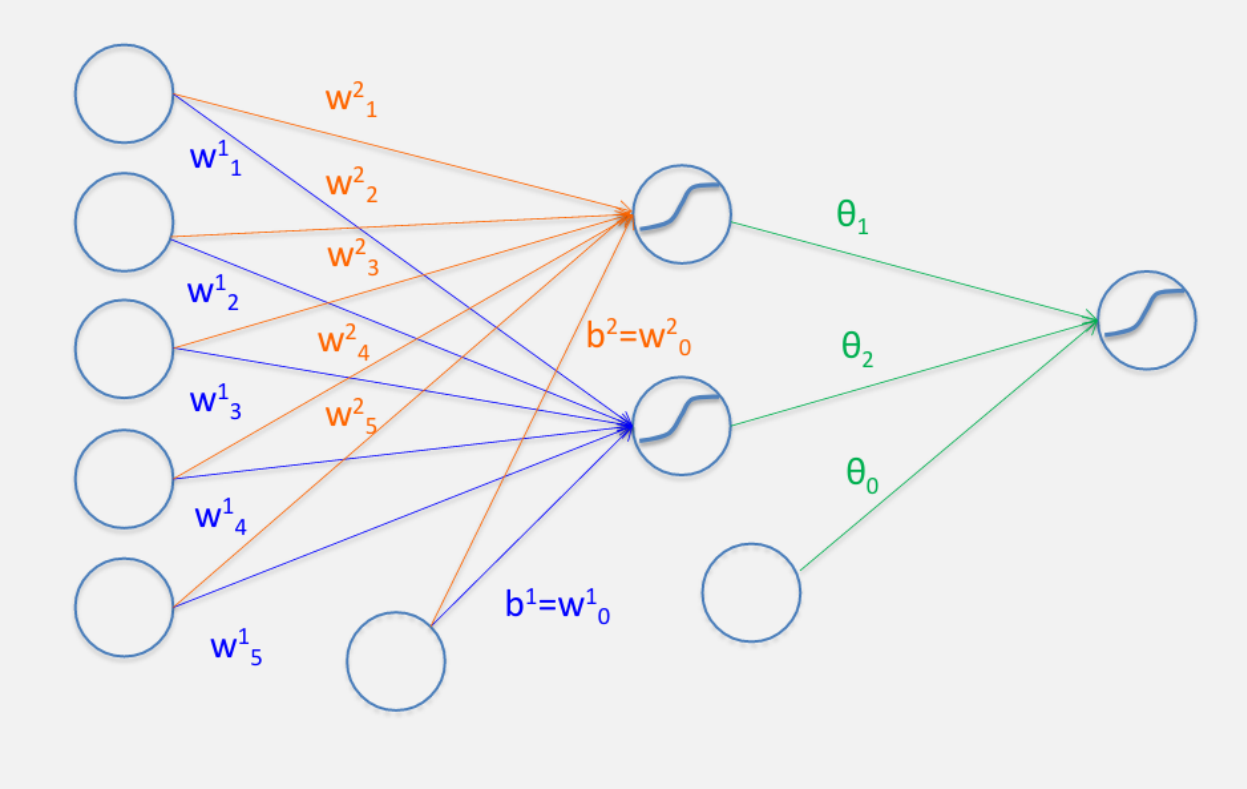
Está acotada

Funciones RELU



En vez de tener salidas 0 o 1, podemos ahora tener valores intermedios

ejemplos de como quedaria la cosa



hasta diapositiva 17 sin incluir

importante las 3 características

muy sensibles a los hiperparámetros

cada vez que se ejecuta, te da un resultado diferente, aun manteniendo los hiperparámetros

algoritmo de retropropagación del error es muy sensible

- es muy importatne

propiedad de aproximación universal (entra fijo)

- podria tener vario homegas
- con una sola capa intermedia y un numero suficiende de neuronas intermedias y una funcion de activacion, es capaz de aprenderse cualquier función
- suficientes neuronas, pero no sabemos el numero exacto

¿Como encontramos estos pesos?

problema de ANN

- Calcular el error, descenso por gradiente cuando la función es convexa (U) es muy sencillo
- En caso de que no sea convexa, es muy dificil calcular la funcion de error

- funciones de error optimas en retropopagación del error (esto se hace por comodidad)

para regresión mse

para clasificación, la entropia negativa

funcion sigmoidal