

TFG - Cap 3 NN

lidiacon

December 2018

1 Introduction

Redes Neuronales

En el que estudiaremos la estructura y las utilidades de las redes neuronales.

2 Definición y elementos

En vistas a imitar el funcionamiento de un cerebro humano, en cuanto lo que a procesamiento se refiere, se crearon las primeras redes neuronales. (Meter historia y patatas)

2.1 Neuronas

La neurona es el elemento básico de una red neuronal. Al igual que las existentes en el cerebro humano, estas neuronas son células de procesamiento; reciben varias entradas de datos y devuelven una única salida. En la mayoría de los casos, esta salida consiste en la suma de la multiplicación de cada entrada por un peso asignado por la propia neurona, de forma que cada neurona devuelve una salida distinta a pesar de que las entradas sean las mismas.

(Foto: func de neurona) Sin embargo, dentro de estas neuronas existen varios tipos, siendo las especializadas en no linealidad las que nos interesan. Éstas también se dividen en función de su algoritmo de activación. A continuación, mencionaremos las más utilizadas.

- Sigmoides: $f(z) = \frac{1}{(1+e^{-z})}$
Esta función logit marca una diferenciación clara entre valores pequeños y grandes agrupándolos en 0 o 1 respectivamente, entre estos dos extremos las neuronas adoptarán una forma de S. (Foto: función sigmoide)
- Tanh: $f(z) = \tanh(z)$
Este tipo de neurona tiene una salida muy parecida a la tipo sigmoide, manteniendo la forma de S, pero con la diferencia de que el marco está entre -1 y 1. Suele ser usada con más asiduidad debido a que está centrada en 0. (Foto: función tanh)
- ReLU: $f(z) = \max(0, z)$
La neurona de linealidad restringida o ReLU (restricted linear unit) utiliza un distinto tipo de no linealidad. A pesar de sus variadas desventajas, en los últimos años se ha posicionado como la más usada en los problemas de visión computacional. Y será la que más usaremos nosotros en nuestros ejemplos. (Foto: función ReLU)

2.2 Estructura y funcionamiento

Ya que las funcionalidades de una neurona son limitadas, nos resultaría imposible resolver un problema de Machine Learning con ella. Para ello, de nuevo basándose en el modelo de nuestro cerebro, las neuronas se encuentran agrupadas por capas, formando así una red neuronal. Cada capa consta de un

número finito de neuronas del mismo tipo, recibiendo todas el mismo volumen de datos pero, como hemos mencionado antes, devolviendo resultados diferentes cada una.

Las redes neuronales están compuestas generalmente por varias capas, de forma que las salidas de la anterior forman las entradas de la siguiente. Entre ellas, podemos distinguir la capa de entrada, que es la que recibe los datos a analizar, la capa de salida, que es la que devuelve los resultados en el formato que nos interesa en función del problema, y las capas intermedias u ocultas. Éstas son las encargadas de abstraer todos los datos recibidos por la capa anterior para que sea más fácil procesarlos y analizarlos. Un ejemplo sería el ser capaz de reconocer formas y contornos: Nuestros ojos reciben sólo un grupo de colores que el cerebro ayuda a procesar añadiendo bordes y separaciones para poder interpretar las imágenes.

(explicar back-propagation)

3 Tipos de problemas

Como ya hemos visto, las redes neuronales están diseñadas para poder adaptarse y usarse en múltiples problemas de Machine Learning. Para aprender a manejarlos con ellas, hemos desarrollado ejemplos de redes neuronales en los siguientes tipos de problemas.

3.1 Clasificación

Los problemas de clasificación consisten, a grandes rasgos, en asignar un dato recibido a una categoría de un conjunto ya establecido. El ejemplo más claro de esta categoría sería proveer de imágenes de perros y gatos a una red neuronal, y que sea ésta la que decida de qué animal se trata en cada caso.

En nuestro ejemplo de aprendizaje sobre clasificación, hemos elegido otro dataset conocido: Reconocimiento de cifras escritas a mano. El dataset consiste en un conjunto de imágenes de cada número escrito a mano de distintas formas, incluyendo la respuesta correcta para comprobar los resultados de nuestro clasificador.

(añadir estructura del clasificador: EN REVISIÓN)

A modo de salida, hemos utilizado un tipo especial de capa llamado *softmax*. En ella, se da la condición de que la suma de todas las salidas sea igual a 1. De esta forma, las probabilidades que reflejan las distintas salidas se encuentran normalizadas y son excluyentes entre sí, dejando que haya una probabilidad que destaque por encima de las demás en su conjunto; ésa será nuestra predicción.

3.2 Regresión

Los problemas de regresión se distinguen de los de clasificación en que la respuesta de la red neuronal no pertenece a un grupo, sino que forma parte de un conjunto continuo de posibles resultados. La forma de entender este tipo de problemas sería tomando como ejemplo una red neuronal que sea capaz de predecir el precio de una casa en función de su tamaño, número de habitaciones, etc.

Para este ejemplo, hemos utilizado dicho dataset, ya que está incluido directamente en uno de los paquetes de Keras. (completar ejemplo regresión)