22-06-2021

Aprendizaje semisupervisado

Establecemos el n de centroides.

Trabaja con las m imágenes que mejor representen la totalidad del conjunto de datos. Selecciona las m imágenes en función de la menor distancia a los centroides establecidos.

Se entrena el modelo con las imágenes más representativas. Con el semisupervisado lo que hago es propagar las clases dependiendo de las distancias a cada centroide. Se asigna a las clases que están entorno a cada una de las imágenes más representativa las que están más cercanas que pueden representar la clase de base o no (si la imagen más representativa es un 4 asignará también un 4 a los más cercanos a ese 4. Tiene un margen de error al asignar a imágenes que no son la representativa (4), pero están cercanas a esa imagen representativa, el valor de esa clase. (por ej. si hay un 5 entre las más cercanas, le asignará también un 4).

Percentile\_closet.- Hay otro algoritmo para que cuando dos clúster están muy cercanos incluya en uno de ellos los valores que están más cerca del centroide más el 20% de distancia. Incluirán estos valores en el clúster con el que se aplique en primer lugar este criterio.

Hay que comprobar posteriormente de forma manual que están correctos los dígitos que se incluyen en cada clúster. Si no es así hay que recolocar a mano el clúster en el que se ha de incluir

TSNE.- algoritmo que reduce dimensiones como PCA. Utiliza el mismo método de proyecciones que PCA

DBSCAN

Genera clúster teniendo en cuenta la densidad de los puntos

eps.- distancia que va a considerar como máximo para que sea un vecino

min\_sample.- establece el mínimo de vecinos que han de existir para considerar que es una instancia central

Tiene en cuenta la magnitud de los datos por lo que si tenemos variables con diferentes magnitudes (rangos):

* previamente tenemos que normalizar/standarizar las variables (standarScaler)
* Aplicar Kmeans que no tiene en cuenta las magnitudes

Dentro del clúster puede haber varias instancias centrales

Ventajas respecto a Kmeans:

* Se pueden detectar los outliers (los -1). Valor -1 no pertenece a ningún clúster. Son las anomalías
* Puede recoger distribuciones

Se pueden mostrar las anomalías para ver qué características tienen

No supervisado:

* Clusterización: Kmeans, DBSCAN
* Para reducir la dimensionalidad: PCA, TSNE

24-06-10. Deep learning.- es un subconjunto de machine learning que se refiere a las redes neuronales. A partir de 3 capas ocultas es cuando se considera Deep learning

Data mining.- automatización de trabajo de datos para encontrar patrones repetitivos con lo que podría incluir algoritmos de machine learning

Neural networking

El perceptrón es una red neuronal constituida por una neurona. El bias equivaldría a la secante. Se suman todas las entradas (nuestras X). La neurona tiene una función de activación que transforma las X iniciales en otro valor ajustando el valor de los pesos (pendiente) para hacer la predicción. Los pesos en la primera transformación se pueden asignar de forma aleatoria (los puede establecer el propio algoritmo) o asignarlos quien realiza el estudio. También asigna el propio algoritmo un bias(secante)

Si una X tiene un peso ‘0’ no

Cuantas más capas ocultas tengas más profundo es el aprendizaje.

Normalmente se utiliza para aprendizaje supervisado y refuerzo aunque también se puede utilizar para aprendizaje no supervisado

En el aprendizaje por refuerzo, continua aprendiendo en función del refuerzo ante el resultado.

Tensor.- A partir de 3 dimensiones hablamos de tensor.

Se trabaja con TensorFlow y Keras

Se puede modificar la función de activación en cada capa. Si el problema es de clasificación hay que utilizar la función sigmoide porque da un resultado entre 0 y 1

Early stopping.- para evitar el sobreentrenamiento

Regularization (dropout).-

Neurona recursiva funciona muy bien para predicción de series temporales y

La sigmoide nos da probabilidad

La hiperbólica tangencial para imágenes da valor entre -1 y 1

Cada capa ha de tener una función de activación. La misma o diferente en cada capa

TensorFlow.- Keras es la API de TensofFlow

En RELU (tiene valores entre 0 e infinito), cualquier valor negativo es ‘0’. Para los positivos recoge el valor positivo

La función de activación softmax crea un array cuya suma de valores es 1. Nos dará con un porcentaje la probabilidad de acierto (fiabilidad) de cada una de las neuronas

La capa Dense es una capa de neuronas. La última capa Dense es la que nos da los resultados, uno o varios

La función ‘loss’ minimiza el error