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

28-06-2021

Posible solución de la competición

Profilereports para obtener de forma rápida, missing, duplicados,…

Utilizar el archivo kill Python para que me elimine los archivos Python en el sistema que me haya creado y no tienen utilidad pero sí están ocupando memoria. Cada vez que se reinicia el ordenador estos ficheros desaparecen pero para no tener que estar saliendo yo entrando durante la ejecución de algún proceso.

Una vez entrenado el modelo con una muestra de los datos aplicamos el modelo seleccionado con todos los datos

Utilizamos nuestro modelo con las mejores características que nos ha dado. Una vez que lo tenemos lo entrenamos con cross validation

Cuando tenemos un número de datos enorme puede que haya problemas de memoria en el ordenador por lo que se sugiere utilizar un cros validation tipo II dividiendo el grupo de validación en dos. Una parte hará de train y otra de validación y después al revés, el grupo de validación de esta submuestra será el de train y el de train será el de validación. En un segundo momento se hace lo mismo pero con los datos de train de la muestra inicial extraída de train y validation.

Está encontrando un patrón cuando en una gráfica vamos viendo que se va manteniendo en un score similar. Si a partir del más alto vemos que posteriormente el conjunto de train o de validation baja puede ser que haya sobreentrenamiento

**Redes neuronales. Funciones de optimización y regularización**

Optimización para que el error sea el mínimo posible. Hay que buscar el algoritmo de optimización que será mejor para nuestro modelo.

RMSpro solo se puede utilizar para regresión

Regularización.- Pueden desactivar capas enteras o varias neuronas de una capa. Se pueden utilizar varios regularizadores a la vez.

El regularizador dropout va apagando algunas neuronas para que las demás puedan detectar sin las que están muy especializadas en la detección de algún patrón

**Red neuronal convolucional.-**

En cada una de las capas, se van a realizar, a partir de las entradas (normalmente son imágenes) una operación diferente (un kernel diferente. Kernel es una matriz que puede ser de dos dimensiones o de más).

Varios tipos de convolución

Valid padding.- No crea números fuera de la matriz principal

Con padding same (crea números fuera de la matriz principal) mantenemos la resolución de la imagen original

Con stride 2 ( se va pasando el kernel de dos en dos) la resolución es menor. Sa va pasando de dos pixeles en dos pixeles. Así aunque tenga menos resolución puede haber patrones más reconocibles que se puedan ver.

Max pooling, min pooling y average pooling.

Nos quedamos con un imagen de menor resolución pero con el valor más alto (pixel mas cercano al color X por ej.)

Max, nos quedamos con el valor más alto. Se queda con los pixeles que aportan mejor resolución

Min nos quedamos con el valor más bajo

Average, la media de los valores

La finalidad de Deep learning con imágenes es tener el mayor número posible de imágenes de la imagen original para encontrar patrones. Cada neurona se centra en una imagen modificada específica (convoluciones). Cada convolución es una capa. El Relu es la capa de activación. Es el resultado de la convolución

Con el optimizador evitamos que varias neuronas se especialicen en un mismo aspecto. Se mantendrán aquellas neuronas que realmente aporten valor para identificar el patrón

Filter.- número de neuronas

Kernel\_size.- tamaño del kernel

Input\_shape.- (ej. 32,32,3), el 3 hace referencia a que las imágenes son en color

Padding = same

Por defecto el stride es 1. Stride (1,1) se mueve una columna hacia la derecha y 1 fila al bajar. Stride (2,2), se mueve dos columnas a la derecha y 2 filas al bajar

La función softmax te da una salida de un array

El sparse nos da un vector con el que se identifica la imagen (ej. [0,0,1]). 3 valores en el vector porque el Dense es 3 (3 neuronas)