

Aprendizaje profundo y redes neuronales convolucionales

Los algoritmos de retro-propagación en las Redes Neuronales han permitido el uso de un gran número de capas ocultas en una sola red. Estas redes neuronales pueden ser consideradas profundas, donde cada capa puede interpretar y combinar varios conceptos abstractos más simples que fueron aprendidos por los nodos de la capa anterior y producir conceptos de nivel superior.

Las redes neuronales convolucionales pueden reducir el recuento de nodos en redes neuronales profundas que procesan información bidimensional (como imágenes) al tratar las entradas cercanas de manera diferente en comparación con las entradas que están muy separadas. Como resultado, estos modelos han demostrado ser especialmente exitosos cuando se trata de tareas de procesamiento de imágenes y videos. Además de las capas totalmente conectadas, similares a las capas ocultas en el Perceptrón Multicapa (MLP), estas redes utilizan capas de agrupación (muestreo descendente), que agregan salidas de neuronas de capas anteriores, y capas convolucionales, que pueden usarse como filtros para detectar ciertas características (como un borde en una orientación particular).

El entrenamiento de modelos de aprendizaje profundo puede ser computacionalmente intensivo y, a menudo, se realiza con la ayuda de unidades de procesamiento de gráficos (GPU), que son más eficientes que las CPU normales para implementar el algoritmo de retro-propagación. Las bibliotecas especializadas de aprendizaje profundo, como TensorFlow, son capaces de utilizar plataformas informáticas basadas en GPU. Sin embargo, en aras de la simplicidad, utilizaremos la implementación de MLP ofrecida por la biblioteca y un conjunto de datos simple. Los principios que se utilizarán, sin embargo, todavía se aplican a redes y conjuntos de datos más complejos.

Representación de la configuración de capa oculta

Dado que la arquitectura de MLP está determinada por la configuración de capa oculta, exploremos cómo se puede representar esta configuración en nuestra solución. La configuración de capa oculta del modelo Multilayer Perceptron (https://scikit-learn.org/stable/modules/neural_networks_supervised.html) se transmite a través de la tupla `hidden_layer_sizes`, que se envía como parámetro al constructor del modelo. De forma predeterminada, el valor de esta tupla es (100,), lo que significa una sola capa oculta de 100 nodos. Si quisiéramos, por ejemplo, configurar el MLP con tres capas ocultas de 20 nodos cada una, el valor de este parámetro sería (20, 20, 20). Antes de implementar nuestro optimizador basado en [algoritmos genéticos](#) para la configuración de la capa oculta, necesitamos definir un cromosoma que pueda traducirse en este patrón.sklearn

Para lograr esto, necesitamos encontrar un cromosoma que pueda expresar tanto el número de capas como el número de nodos en cada capa. Un cromosoma de longitud variable que se puede traducir directamente en la tupla de longitud variable que se utiliza como parámetro del modelo, es una opción; Sin embargo, este enfoque requeriría operadores genéticos personalizados, posiblemente engorrosos. Para poder usar nuestros operadores genéticos estándar, usaremos una representación de longitud fija. Cuando se utiliza este enfoque, el número máximo de capas se decide de antemano, y todas las capas siempre están

representadas, pero no necesariamente expresadas en la solución. Por ejemplo, si decidimos limitar la red a cuatro capas ocultas, el cromosoma se verá de la siguiente manera: $[n_1, n_2, n_3, n_4]$

Sin embargo, para controlar el número real de capas ocultas en la red, algunos de estos valores pueden ser cero o negativos. Tal valor significa que no se agregarán más capas a la red. Los siguientes ejemplos ilustran este método:

El cromosoma $[10, 20, -5, 15]$ se traduce en la tupla $(10, 20)$ ya que el -5 termina el recuento de capas.

El cromosoma $[10, 0, -5, 15]$ se traduce en la tupla $(10,)$ ya que el 0 termina el recuento de capas.

El cromosoma $[10, 20, 5, -15]$ se traduce en la tupla $(10, 20, 5)$ ya que el -15 termina el recuento de capas.

El cromosoma $[10, 20, 5, 15]$ se traduce en la tupla $(10, 20, 5, 15)$.

Taller - Problemas de Aprendizaje Automático

1. Ir al link: <https://macti-meia.unam.mx/> y seleccionar JUPYTERHUB y acceder con su cuenta asignada.
2. Iniciar una nueva Terminal y asegurarse de estar dentro de la carpeta "taller_deap"
3. Desde la terminal ejecutar: **"git clone <https://github.com/aoaunam/mlpDEAP.git>"**