

¿Qué es PyTorch?

Es un paquete de Python diseñado para realizar cálculos numéricos haciendo uso de la programación de tensores. Además permite su ejecución en GPU para acelerar los cálculos.

Normalmente PyTorch es usado tanto para sustituir numpy y procesar los cálculos en GPU como para la investigación y desarrollo en el campo del machine learning, centrado principalmente en el desarrollo de redes neuronales.

Proporciona módulos y clases elegantemente diseñados como `torch.nn`, `torch.optim`, `Dataset` y `DataLoader` para ayudarnos a crear y entrenar redes neuronales.

En nuestro trabajo utilizamos algunos como:

- **`torch.nn.Sequential`**: para crear perceptrón multicapa.
- **`Torch.nn.MSELoss`**: para definir función de pérdida.
- **`torch.optim.Adam`**: para crear optimizador.
- **`torch.autograd.grad`**: para calcular derivadas de las salidas del perceptrón con respecto a entradas.
- **`torch.optim.Adam.zero_grad`**: Establece los gradientes de todos los tensores optimizados a cero.

Alternativas a PyTorch.

En la actualidad disponemos de varias alternativas a PyTorch en su aplicación al machine learning, algunas de las más conocidas son:

- **Tensorflow**: fue desarrollado por *Google Brain Team*. Es software libre diseñado para computación numérica mediante grafos.
- **Caffe**: es un framework de machine learning diseñado con objetivo de ser usado en visión por computador o clasificación de imágenes. Caffe es popular por su librería de modelos ya entrenados (Model Zoo) que no requieren de ninguna implementación extra.

- **Microsoft CNTK:** es el framework de software libre desarrollado por Microsoft. Es muy popular en el área de reconocimiento del habla aunque también puede ser usado para otros campos como texto e imágenes.
- **Theano:** es una librería de python que te permite definir, optimizar y evaluar expresiones matemáticas que implican cálculos con arrays multidimensionales de forma eficiente.
- **Keras:** es una API de alto nivel para el desarrollo de redes neuronales escrita en Python. Utiliza otras librerías de forma interna como son Tensorflow, CNTK y Theano. Fue desarrollado con el propósito de facilitar y agilizar el desarrollo y la experimentación con redes neuronales.

¿Por qué usar PyTorch?

PyTorch es una librería muy reciente y pese a ello dispone de una gran cantidad de manuales y tutoriales donde encontrar ejemplos. Además de una comunidad que crece a pasos agigantados.

PyTorch dispone una interfaz muy sencilla para la creación de redes neuronales pese a trabajar de forma directa con tensores sin la necesidad de una librería a un nivel superior como pueda ser Keras para Theano o Tensorflow.

Al contrario que otros paquetes como Tensorflow, PyTorch trabaja con grafos dinámicos en vez de estáticos. Esto significa que en tiempo de ejecución se pueden ir modificando las funciones y el cálculo del gradiente variará con ellas. En cambio en Tensorflow debemos definir primero el grafo de computación y después utilizar la *session* para calcular los resultados de los tensores, esto dificulta la depuración de código y hace más tediosa su implementación.

PyTorch dispone de soporte para su ejecución en tarjetas gráficas (GPU), utiliza internamente CUDA, una API que conecta la CPU con la GPU que ha sido desarrollado por NVIDIA.