

Actividad de investigación sobre JAX

Esta actividad voluntaria tiene como objetivo descubrir y probar JAX, una librería de Google que surge como alternativa a TensorFlow y PyTorch y que se está empleando actualmente en proyectos de gran envergadura como DeepMind.

1. Qué es JAX y cuáles son sus principales características.

JAX es una biblioteca desarrollada por *Google* que proporciona una interfaz similar a *NumPy*, pero con soporte nativo para aceleración en GPU y TPU. Su integración con **XLA** (*Accelerated Linear Algebra*) permite transformar código Python en ejecutables altamente optimizados, mejorando la eficiencia computacional en tareas de aprendizaje automático y modelado numérico.

Características Clave de JAX:

- Autodiferenciación (*grad*): permite calcular derivadas de cualquier orden de manera eficiente, lo que resulta fundamental en la optimización de modelos de machine learning mediante descenso de gradiente. Su flexibilidad lo hace ideal para calcular gradientes en redes neuronales y otros modelos de optimización matemática.
- Compilación Just-In-Time (*JIT*): a través de XLA, compila dinámicamente las funciones para maximizar su rendimiento en hardware acelerado. Esto reduce significativamente los tiempos de ejecución y permite ejecutar código en GPU/TPU sin modificaciones adicionales.
- Autovectorización (*vmap*): transforma automáticamente funciones escalares en operaciones vectorizadas, eliminando la necesidad de escribir bucles explícitos y optimizando el procesamiento de datos en paralelo.
- Auto-paralelización (*pmap*): distribuye automáticamente la ejecución en múltiples dispositivos (CPU, GPU, TPU), facilitando el entrenamiento de modelos en entornos de computación distribuida y reduciendo los cuellos de botella computacionales.

En conjunto, estas características hacen de JAX una herramienta poderosa para investigadores y profesionales en IA y Big Data que buscan aprovechar la computación numérica optimizada sin perder la flexibilidad de Python.

2. Comparación de JAX con TensorFlow y PyTorch

JAX: desarrollado por *Google*, conocido por su **eficiencia** y capacidad para acelerar el código mediante la **compilación en tiempo real (JIT)**. Admite operaciones de **GPU** y **CPU** y utiliza un paradigma de **programación funcional**, que es más intuitivo y menos propenso a errores en comparación con el paradigma imperativo de TensorFlow y PyTorch. JAX es particularmente útil para investigadores y desarrolladores que necesitan **alto rendimiento y flexibilidad**.

TensorFlow: también desarrollado por *Google*, es reconocido por su **amplia infraestructura y escalabilidad**, lo que lo hace adecuado para implementaciones a gran escala. Ofrece una amplia gama de herramientas y bibliotecas, como **TensorFlow Serving** para la implementación de modelos y **TensorFlow Lite** para dispositivos móviles y de borde. TensorFlow es **más complejo y menos flexible** que JAX y PyTorch, pero brinda un soporte sólido para entornos de producción.

PyTorch: desarrollado por *Facebook*, destaca por su **simplicidad y flexibilidad**, lo que lo convierte en una opción popular entre investigadores y desarrolladores. Tiene una **comunidad más grande** y más modelos preentrenados disponibles en comparación con TensorFlow. El gráfico computacional dinámico de PyTorch permite una depuración y experimentación más sencillas, lo que resulta beneficioso para fines de investigación.

3. Ecosistema: librerías implementadas sobre JAX y otras herramientas que se integran bien con esta tecnología.

- Haiku: Biblioteca de alto nivel similar a **Keras**. Permite programar con un enfoque orientado a objetos mediante la función *transform()*, que convierte funciones en *init()* y *apply()*.

- Optax: Biblioteca de optimización con **funciones de pérdida comunes** (*L2*, *BCE*), **optimizadores** (*SGD*, *Adam*, *RMSProp*), **privacidad diferencial** y **transformaciones avanzadas** (*clipping*, *ruido*, *escalado*).

- Jraph: Librería para **redes neuronales** en grafos (*GNNs*) que usa la estructura *GraphsTuple* para representar múltiples grafos. También incluye un modelo zoo.

- RLax: Biblioteca para **aprendizaje por refuerzo** (*RL*) que proporciona implementaciones matemáticas clave como valores, funciones de retorno y políticas.

- Chex: Herramienta de **depuración**, con funcionalidades de **aserciones**, pruebas en distintas arquitecturas y fakes para emular **paralelización**.