**Visión de Conjunto**

En esta sección se va a describir la solución completa de Huawei con respecto a la IA y se va a introducir el lugar de MindSpore en esta solución. Los desarrolladores que estén interesados en MindSpore puede visitar la comunidad MindSpore y seguir el repositorio en git.

**Introducción a la solución completa respecto a la IA con Huawei Ascend**

Ascend es una infraestructura de computación tanto de back-end como de front-end y una aplicación basada en la serie de procesadores Ascend. Incluye los chips de la serie Ascend, hardware de la serie Atlas, habilitación de chips de CANN, el framework IA de MindSpore, ModelArts y la habilitación de aplicación de MindX.

Huawei Atlas IA es una solución de computación basada en la serie de procesadores IA Ascend y utiliza distintos productos como módulos, cartas, estaciones de Edge, servidores y clústeres para construir un escenario completo de una solución infraestructural de una IA orientada a dispositivos, Edge y Cloud. Se encarga de los datos centrales y las soluciones inteligentes de Edge, así como la inferencia completa y el procesamiento del entrenamiento en la fase de Deep learning.

* **Serie Atlas:** Provee entrenamiento para la IA, inferencia de cartas y entrenamiento de servidores (Saber más).
* **Arquitectura heterogénea de computación CANN:** Es una capa conductora que habilita los chips (Saber más).
* **MindSpore:** Contine los framework de los escenarios completos (Saber más).
* **MindX SDK:** Ascend SDK que provee soluciones de aplicación (Saber más).
* **ModelArts:** Cloud de Huawei para el desarrollo de plataformas IA (Saber más).
* **MindStudio:** Cadena de herramientas para el desarrollo de E2E que proviene del IDE para el desarrollo de IA (Saber más).

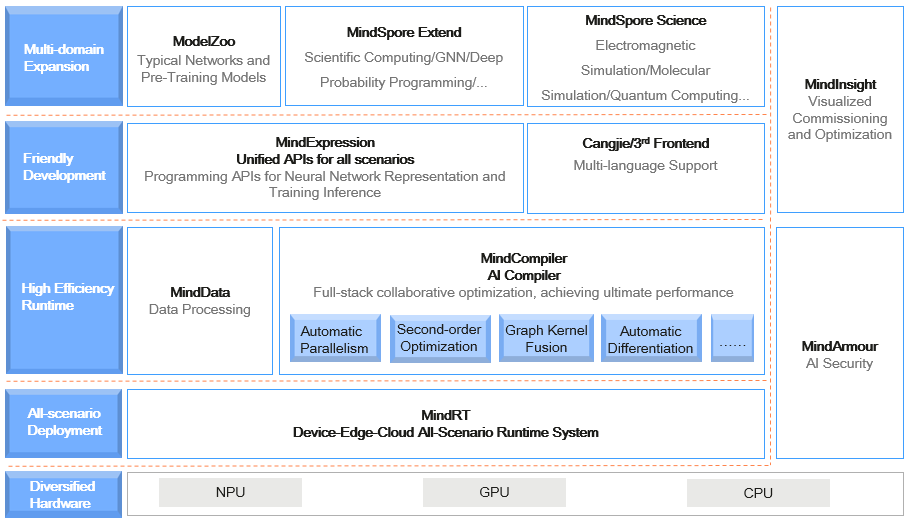
Para más detalles, visita web oficial de Huawei Ascend.

**Introducción a MindSpore**

MindSpore es un framework de Deep learning para todos los escenarios, con el objetivo de conseguir un desarrollo sencillo, una ejecución eficiente y una cobertura completa de los escenarios. Desarrollo de nuevas características mediante el uso de APIs sencillas y un sencillo revisor de código. La ejecución eficiente se refleja en la computación, el preprocesamiento de los datos y el entrenamiento distribuido. LA cobertura de los escenarios completos indica que el framework da soporte a Cloud, Edge y los escenarios de los dispositivos.

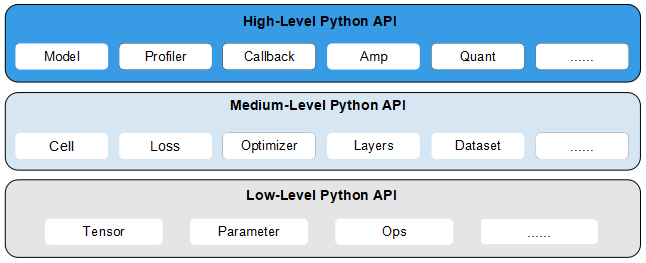
La siguiente figura muestra la arquitectura completa de MindSpore:

* **ModelZoo:** Provee algoritmos de aprendizaje disponibles en la red, y los desarrolladores también pueden contribuir en la creación de nuevas redes.
* **Extensión MindSpore:** El paquete de expansión de MindSpore añade nuevas funcionalidades para el soporte de nuevas áreas, como GNN/programación probabilística profunda/reforzamiento del aprendizaje, etc.
* **MindScience:** Es un paquete de computación científica diseñado para varias industrias basado en la convergencia del framework de MindSpore. Contiene conjuntos de datos punteros en la industria, estructuras básicas de red, modelos entrenados previamente de alta precisión y herramientas de pre y post procesamiento para acelerar el desarrollo de las aplicaciones relacionadas con la computación científica.
* **MindExpression:** Programa para front-end basado en Python y para el desarrollo de interfaces. En el futuro, se planea que se desarrollen interfaces para C/C++. Cangjie, desarrollador de Huawei, esta trabajando en un lenguaje de front-end, que actualmente se encuentra en la primera fase del desarrollo. Además, Huawei está trabajando en la interconexión con front-end de terceros para introducir más ecosistemas externos.
* **MindData:** Provee funciones e interfaces de programación, así como procesadores de datos eficientes, conjuntos de datos comúnmente utilizados, y soporte de usuario para el procesamiento de registros y una tubería de optimización paralela.
* **MindCompiler:** El compilador de la capa central, que se encarga de la implementación de las 3 mayores funciones basada en la unificación del Cloud del dispositivo MindIR, incluyendo toda la independencia de un hardware optimizador (derivación de tipos, diferenciación automática y simplificación de expresiones), optimización de componentes relativos al hardware (paralelismo automático, optimización de memoria, fusión de grafos Kernel y ejecución en tubería) y la optimización relativa al despliegue y la inferencia (cuantificación y optimización de código).
* **MindRT:** Sistema en tiempo de ejecución de MindSpore, que da cobertura la parte relacionada con el tiempo de ejecución en el sistema de Cloud y de usuario, en el lado del dispositivo y en la parte de menor peso de los pequeños sistemas de con Internet of Things.
* **MindInsight:** Provee optimización en tiempo real de códigoy especificación de ciertas herramientas. También proporciona soporte de usuarios en la optimización de código y mejora de las herramientas en tiempo de entrenamiento en red.
* **MindArmour:** Diseñado como aplicación a nivel de empresa, relacionada con los criterios de seguridad y privacidad, como puede ser falta de robustez, modelos de seguridad en las pruebas, entrenamiento individual y privado, privacidad en las tareas de riesgo, detección de datos derivados, etc.



**API a nivel de estructura**

Con el fin de dar soporte en la construcción de la red, en la ejecución de grafos y subgrafos completos y la ejecución de operadores simples de ejecución, MindSpore proporciona a los usuarios 3 niveles distintos de API. En orden ascendente, son las API de bajo nivel en Python, las API de medio nivel en Python y las API de alto nivel en Python.



* API de alto nivel en Python

Son la primera capa. Se basa en las API de medio nivel en Python, añadiendo avances en la API que incluyen entrenamiento y manejo de inferencia, mezclado con la precisión en el entrenamiento y en la optimización y mejora del código, permitiendo a los usuarios controlar la ejecución de los procesos de la red completa e implementar el entrenamiento del modelo. También hay mejoras en la inferencia y la optimización de las redes neuronales. Por ejemplo, utilizando Model API, para que los usuarios puedan especificar el modelo de red neuronal que van a entrenar, así como los ajustes relacionados con el entrenamiento, entrenar el propio modelo de red neuronal, y realizar optimizaciones de código en tiempo real sobre el rendimiento de la red neuronal a través del Proffiler API.

* API de medio nivel en Python

Se trata de la segunda capa, que encapsula API de bajo nivel y proporciona módulos como la capa de red, el optimizador y la función de perdida. Los usuarios pueden construir de forma flexible redes neuronales y controlar los procesos de ejecución a través de estas APIs, con la intención de implementar de forma rápido la lógica del modelo. Por ejemplo, los usuarios pueden llamar a Cell API para construir modelos de redes neuronales y lógica computacional, añadir la función de perdida y optimizar los métodos al modelo de la red utilizando el módulo de perdida y Optimizar API, y usar el módulo del conjunto de datos para procesar la información para el entrenamiento y la derivación.

* API de najo nivel en Python

Se trata de la tercera capa, incluyen la definición de los tensores, las operaciones básicas y un diferenciador automático de módulos, habilitando a los usuarios a definir de forma sencilla los tensores y llevar a cabo computación derivada. Por ejemplo, los usuarios pueden personalizar los tensores usando Tensor API, y utilizar el operador GradOperation en el módulo ops.composite para calcular la derivada de la función de una posición específica.