PlaidML

PlaidML es un sustituto a TensorFlow para el cálculo con
tensores. Su principal ventaja con respecto a TensorFlow es que se pueden usar GPUs tanto de AMD como de NVIDIA,
obteniendo un muy buen resultado.

Keras

Keras es una API de alto nivel para redes neuronales escrita en Python.
Puede correr sobre TensorFlow, CNTK o Theano.
Aunque no lo ponga en su documentación, también se puede usar sobre PlaidML, como explicaremos en esta guía.

PRERREQUISITOS

- Python (recomendado 3.7 o superior)
- Anaconda 3
- Pip
- OpenCL 1.2 o superior

Anaconda

Anaconda es un package manager, un environment manager, una distribución de Python/R
para data science y una colección de más de 7500 paquetes open-source.

Instalando Anaconda

1. Entramos a esta página y descargamos la última versión del instalador.
2. Abrimos una terminal y nos vamos a la carpeta donde hayamos descargado el instalador. Ejecutamos `bash archivo.sh` donde archivo es el nombre del archivo descargado.
3. Iniciará el instalador. Aceptamos y le damos sí a todo.
4. Cuando acabe reiniciamos la terminal para que haya surtido efecto la instalación.
5. Para poder usar conda desde cualquier sitio sin tener el entorno base activado ejecutamos `conda config --set auto_activate_base False`

OpenCL

OpenCL (Open Computing Language, en español lenguaje

de computación abierto) consta de una interfaz de programación de aplicaciones y de un lenguaje de programación. Juntos permiten crear aplicaciones con paralelismo a nivel de datos y de tareas que pueden ejecutarse tanto en unidades centrales de procesamiento como con unidades de procesamiento gráfico.

Instalando OpenCL en AMD

1. Abrimos una terminal y escribimos `sudo pacman -S opencl-amd`
2. Para comprobar que todo vaya bien después de la instalación ejecutamos `clinfo` y debería aparecernos nuestra tarjeta gráfica (activando la opción de dispositivos experimentales, por si acaso).

Instalando OpenCL en NVIDIA

1. Abrimos una terminal y escribimos `sudo pacman -S opencl-nvidia`
2. Para comprobar que todo vaya bien después de la instalación ejecutamos `clinfo` y debería aparecernos nuestra tarjeta gráfica (activando la opción de dispositivos experimentales, por si acaso).

Instalando OpenCL en Intel

1. Abrimos una terminal y escribimos `sudo pacman -S intel-compute-runtime`
2. Para comprobar que todo vaya bien después de la instalación ejecutamos `clinfo` y debería aparecernos nuestra tarjeta gráfica (activando la opción de dispositivos experimentales, por si acaso).

Pip

Pip es el package-management system por defecto de python.

Instalando pip

1. Abrimos una terminal y ejecutamos `sudo pacman -S python-pip`
2. Para comprobar si la instalación ha sido exitosa ejecutamos `pip` en la terminal y debería salirnos las opciones posibles que se le pueden añadir.

INSTALACIÓN Y PREPARACIÓN DEL ENTORNO

1. Creamos el entorno virtual con anaconda ejecutando en la terminal `conda create -n nombredelentorno`
2. Activamos nuestro nuevo entorno virtual con `conda activate nombredelentorno`
3. Instalamos Keras ejecutando `conda install -c conda-forge keras`. Esto instalará

keras solo en el entorno virtual que tenemos activo.

4. Instalamos PlaidML con pip ejecutando `pip install -U plaidml-keras`. Esto instalará plaidml solo en el entorno virtual que tenemos activo.
5. Ejecutamos en la terminal `plaidml-setup` para configurar qué cpu o gpu queremos usar. Guardamos la configuración.
6. Para comprobar que todo ha ido bien, instalamos plaidbench cor `pip install plaidbench` y, luego, ejecutamos `plaidbench keras mobilenet`. Si funciona habríamos completado la instalación.

POSTINSTALACIÓN

PlaidML como backend por defecto en Keras.

Opción 1: Modificar keras.json

1. Ejecutamos `nano ~/.keras/keras.json` en la terminal para editar el archivo. Si no tenemos dicha ruta
   o dicho archivo, lo creamos (si no necesitamos nada especial) con el siguiente contenido:

```
{
   "epsilon": 1e-07,
   "floatx": "float32",
   "image_data_format": "channels_last",
   "backend": "plaidml.keras.backend"
}
```

2. Para comprobar que todo ha ido bien, ejecutamos `python -c "from keras import backend"` y debería darnos una salida del tipo *Using plaidml.keras.backend backend.*

Opción 2: Modificar la variable global KERAS_BACKEND

1. Abrimos una terminal y ejecutamos `KERAS_BACKEND=plaidml.keras.backend`
2. Para comprobar que todo ha ido bien, ejecutamos `python -c "from keras import backend"` y debería darnos una salida tipo *Using plaidml.keras.backend backend.*

Ejemplo de código para probar nuestra instalación al completo

En esta página encontraremos un ejemplo
de código para entrenar una red convolucional simple con el dataset MNIST.

Enlaces de interés

- GitHub de PlaidML
- Gestionando entornos con conda
- Cambiar el backend de Keras
- Usando Pip en un Conda Environment

- Conda Forge
- Buena forma de instalar Pip en Arch Linux