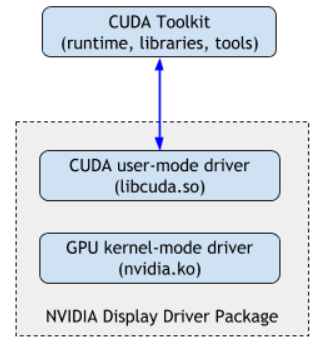
**From GPU to PyTorch: the version configuration and compatibility consideration**

The pipeline of the whole process of using PyTorch with GPU is: check (update) GPU driver 🡪 install CUDA toolkit and cudnn 🡪 create an environment and install the PyTorch properly.

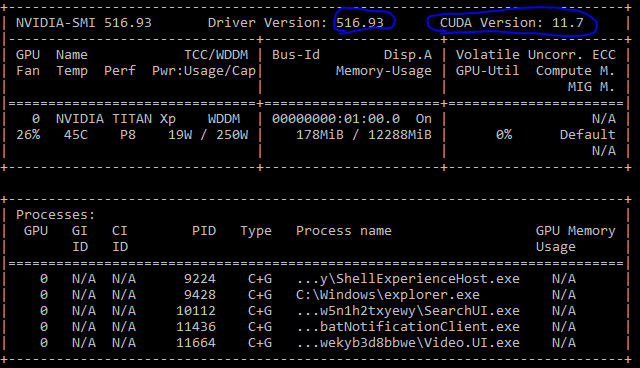
1. Check (update) GPU driver

For a GPU hardware, it can work only when a driver (software) is installed. This driver is called NVIDIA display driver (see the dash-line window in the figure [<https://docs.nvidia.com/deploy/cuda-compatibility/index.html>] below ) which includes CUDA user-mode driver (the first time we see CUDA). The version of the driver determines which version of CUDA Toolkit (see the top of this figure) is applicable, since the driver is closer to the hardware and the CUDA Toolkit is based on it. The driver is downward compatible, meaning that the higher/latter version can replace the older version and can support with the older CUDA Toolkit. Therefore, updating the driver brings more choices.

Driver website: <https://www.nvidia.com/Download/index.aspx?lang=en-us>



In command prompt, type in ***nvidia-smi*** to check the GPU driver version (NVIDIA Display Driver Package) and the version of its component: CUDA user-mode driver. For our TITAN Xp, see the following figure.

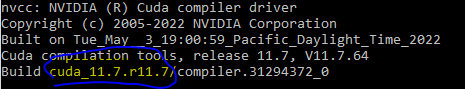


1. Download and install CUDA Toolkit with version lower than the CUDA user-mode driver in the driver. Website: <https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64>

This is the second time we see CUDA.

Download and install cudnn whose version is determined by the CUDA Toolkit. Website: <https://developer.nvidia.com/rdp/cudnn-download>

In command prompt, type in ***nvcc --version*** to check the installed CUDA Toolkit version.



1. Install PyTorch in an environment. The official website is <https://pytorch.org/> . Here we need to choose CUDA version again to install a cudatoolkit and this is the third time we see a CUDA thing. This cudatoolkit version is limited to this specific environment and works as a replacement of the CUDA Toolkit at step 2. Therefore, you need to make sure its version is lower than the CUDA driver at step 1 and is compatible with cudnn at step 2. Usually, choose one lower and closer to the CUDA version at step 2.

Finally, the version flow is: GPU hardware 🡪 driver version 🡪 version of CUDA Toolkit and cudnn 🡪 PyTorch installation (cudatoolkit version).