***Assignment for Digilytics.ai***

In this assignment I will discuss some methods of Fine – tuning a BERT model for a Spam classification project.

We will discuss the following methods:-

1. Using Google Colab with a GPU
2. On Windows machine without a GPU
3. On Windows machine which has an NVIDIA GPU installed internally (specify any one GPU model)
4. On Windows machine which has an NVIDIA GPU attached externally (specify any one GPU model)
5. By connecting the windows machine to a linux cloud instance on Azure or GCP or AWS, with a connected GPU
6. Using google collaborator
   1. Open the ipynb notebook give in the assignment folder (Fine –tuning BERT)
   2. Select change runtime from Runtime menu and and select GPU as accelerator.
   3. In colab you do not need to install the libraries but only need to import them as most of them are already preinstalled on colab cloud by google
7. On windows without gpu
   1. Open the ipynb notebook on your own compyter by installing jupyter
   2. And then install the other libraries by typing !pip install torch
   3. Torch numpy pandas and sklearn(and install allother libraries which are not preinstalled
   4. In the second code cell od the ipynb notebook change (‘device = torch.device(“CUDA”)) to (‘device= torch.device(“CPU”) as we do not want to run our model on GPU (This will take more time)
8. 3.On Windows machine which has an NVIDIA GPU installed internally (specify any one GPU model)
   1. You need to install Jupyter notebook instance on your local device to run the ipynb notebook
   2. And then install the other libraries by typing !pip install torch
   3. Torch numpy pandas and sklearn and other libraries which are not present on your local computer
   4. And Then you need to install the CUDA and CuDNN to use internal GPU of the PC (A separate document is provided which contains detailed steps of how to set up CUDA on your local machine
   5. After installing all the libraries and preparing the environment you can now run the ipynb notebook on your local windows computer with internal GPU
9. On Windows machine which has an NVIDIA GPU attached externally (specify any one GPU model)
   1. For externally attaching a NVEDIA GPU you need to have a video card docker and cables to connect it
   2. After connecting the GPU externally to the computer you need to download NVEDIA Driver for the particular GPU
   3. And then install the other libraries by typing !pip install torch
   4. Torch numpy pandas and sklearn(and install allother libraries which are not preinstalled
   5. And Then you need to install the CUDA and CuDNN to use internal GPU of the PC (A separate document is provided which contains detailed steps of how to set up CUDA on your local machine
   6. After installing all the libraries and preparing the environment you can now run the ipynb notebook on your local windows computer with internal GPU
10. By connecting the windows machine to a Linux cloud instance on Azure or GCP or AWS, with a connected GPU(I will be explaining for AWS particularly)
    1. To use the ipynb notebook first step you need to take is to sign up on AWS website and make a root account
    2. After setting up the account you need to make an S3 bucket (you can search for S3 bucket service on find services) . This S3 bucket will be used to store all the model parameters , architecture and data itself.
    3. After that make a sagemaker instance( you can search sage maker in the find services search bar) sagemaker is the AWS which launches an instance with jupyter , which comes very handy helpful in implementing data science projects.

Note- turn off the sagemaker instance when you are done working otherwise the standard AWS charges may include

* 1. After opening the jupyter notebook you need to upload the ipynb notebook in it.
  2. After you have the notebook in your jupyter instance run it.

Note – Run the notebook instance on t2.medium or t2.large which come in free tier to exclude any additional charges from amazon

Made by:-

Yatin Arora