Introduction to deep learning

Assignment2

Batu Kaan Ozen

Jan 28 2022

Project Definition

It is known that algorithm with communication skill is going to play important role in our future life. The aim of the project is that contributing the research of Natural Language Processing algorithms. Our project aims classifying different tweets from different celebrities.

Figure 1: Elon Musk tweet related to AI **2014**

Project Plan:

The plan of the project is firstly applying batchfy operation for reducing variance and having regularization effect afterwards using PyTorch Lighting to solving hyper parameter optimization problem and increasing training efficiency using Lightnings' properties.. The Pytorch Lightning offers a lot of properties such as running your code in different hardware kinds (CPU, GPU, TPU), logging, performance profiler, botleneck profiler, Metrics, Model Checking, Visualization, Early stopping, Running Distributed Training and many more. The main reason using lightning in our project was detecting proper batch size and training rate. Afterwards using Bi direction LSTM for increasing model complexity and accuracy. It is also planned that using Bidirectional long-short term memory(bi-lstm) for training.It is the process of making any neural network to have the sequence information in both directions backwards (future to past) or forward(past to future). For giving an example, we can examine that we can examine this example, "boys go to ....." we can not fill the blank space. Still, when we have a future sentence "boys come out of school", we can easily predict the past blank space the similar thing we want to perform by our model and bidirectional LSTM allows the neural network to perform this. Afterwards we will examine different hardware performance CPU, GPU and TPU. A central processing unit (CPU), also called a central processor, main processor or just processor, is the electronic circuitry that executes instructions comprising a computer program. The CPU performs basic arithmetic, logic, controlling, and input/output (I/O) operations specified by the instructions in the program. A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. The main difference between CPU and GPU is that CPU contains less amount core but the core of cpu is more stronger than gpu, so that means

that gpu can handle more parallel operation compare to cpu and handling a large number of parallel operation cause acceleration in computing. But some task especially which is not appropriate for parallel calculation works in GPU slower. When we come to TPU, Tensor Processing Unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning, particularly using Google's own TensorFlow software.[1] Google began using TPUs internally in 2015, and in 2018 made them available for third party use, both as part of its cloud infrastructure and by offering a smaller version of the chip for sale. My colab pro membership offers an usage of colab TPU. Because of its optimization and precision, it may sometimes cause more accurancy in some tasks. Lastly it is planned to use pretrained encoder.

Hardware :

Google Colab pro offers us:

1. 25 GB RAM
2. TPU v3
3. Intel Xeon(R) CPU 2.30 Hz
4. Tesla P100 PCIE 16GB

The memory size of the GPU is especially help full for our batchfy case, because it offers us to hold more variable on memory during training and it increase our batch size and this cause regularization effect to our training and co lab also offer us to easy training process. All the libraries are installed before training. We just have to install our frame work (Lightning). Using below line : **!pip install git+https://github.com/PyTorchLightning/pytorch-lightning**

Project completed steps:

Firstly, the batcfy is operation completed, it caused great accuracy performance increase and using pre trained encoder is completed but it did not increase the accuracy so it is decided to remove this part. After this step instead of using one directional LSTM, it is changed with its bidirectional version. it also caused performance increase in my training. After that my code is converted to pytorch lightning version and model is trained with TPU and CPU. CPU performance was really slow and TPU performance did not make any contribution to my training process.Then most important property of lightning(Hyper parameter optimization) is applied found its hyper parameters (batch size and proper learning rate). Even tough Pytorch lighting offers us automatic batch number finding algorithm, the number was approximately 2000, but it caused very big regularization effect so, it is decided that instead of using batch size from lightning batchsize, the proper batch size is found by experimentally. it was 32. After finding proper batch size, learning rate is found by using pytorch lightning as you can see in figure 3.

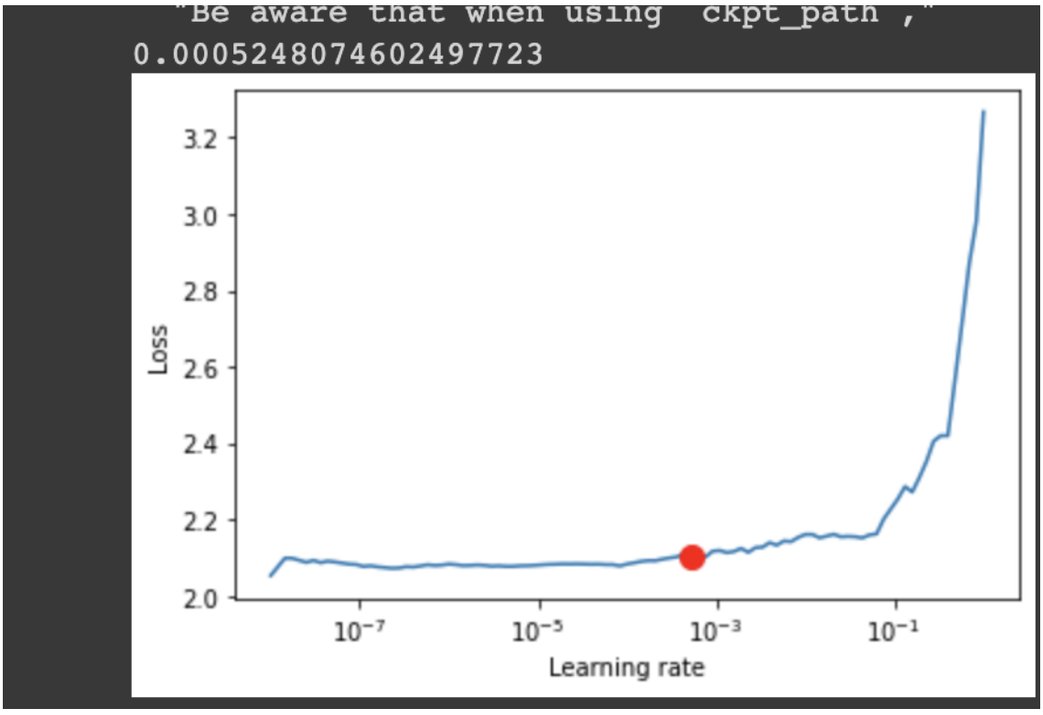

Figure 2: Found learning rate from Lightning

After finding learning rate model is trained with 100 epoch-es and applied early stepping in 10k iteration by inception for that tensorboard is used. .You can see training loss in figure 4, training accuracy in figure 5, validation accuracy in figure 6 and validation loss in figure 7

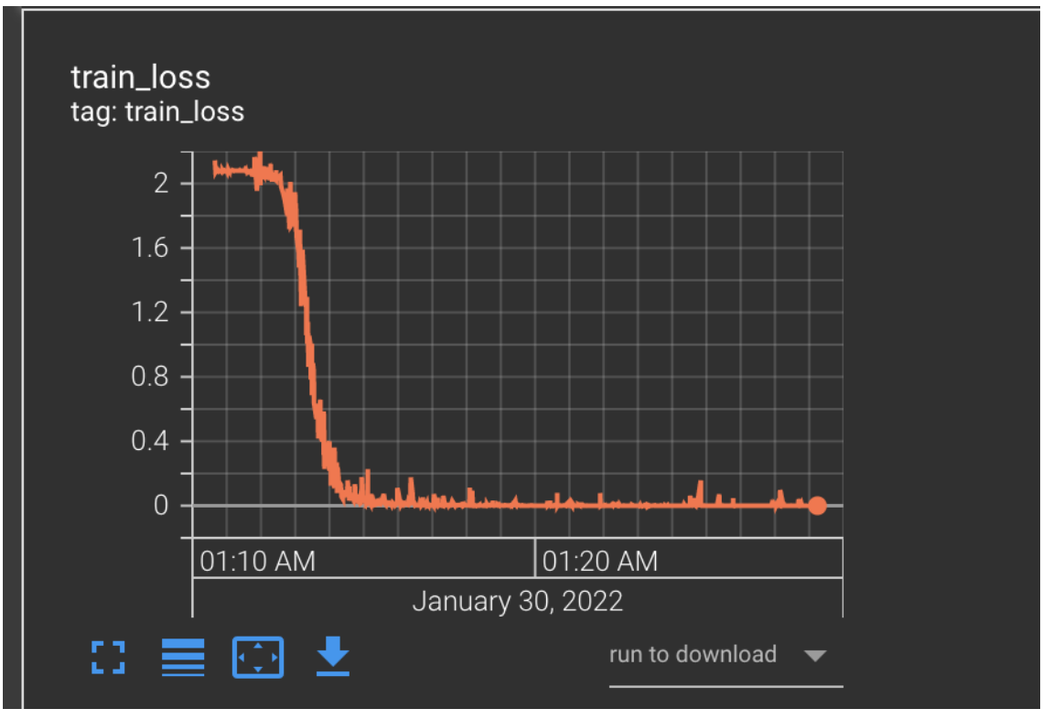

Figure 3: Training loss

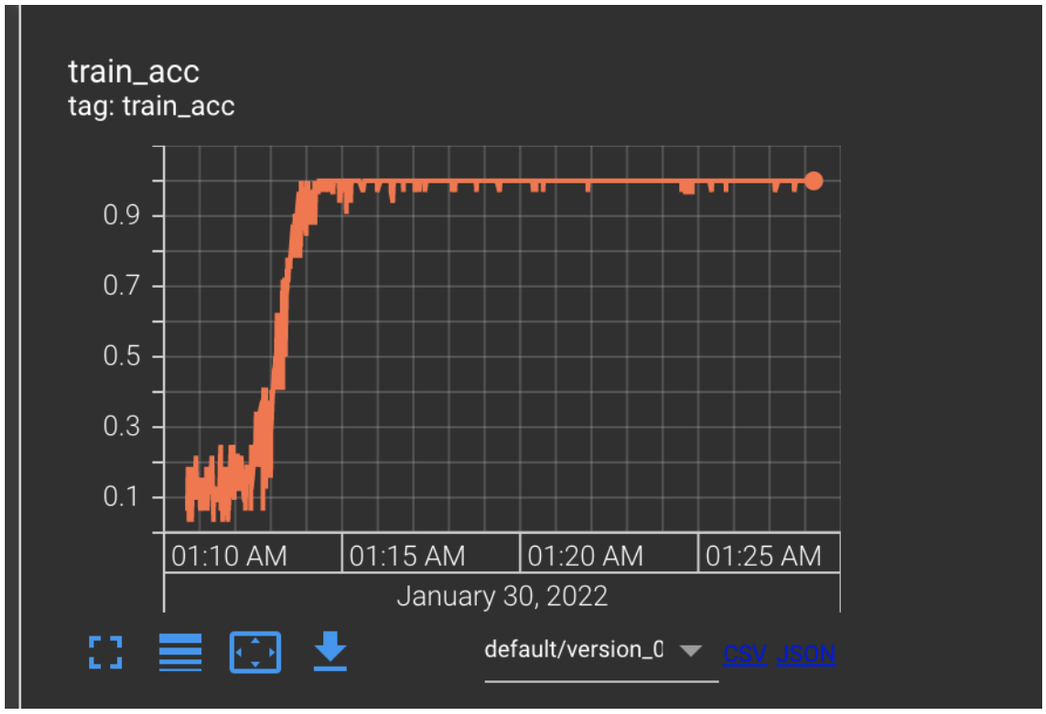

Figure 4: Training accuracy

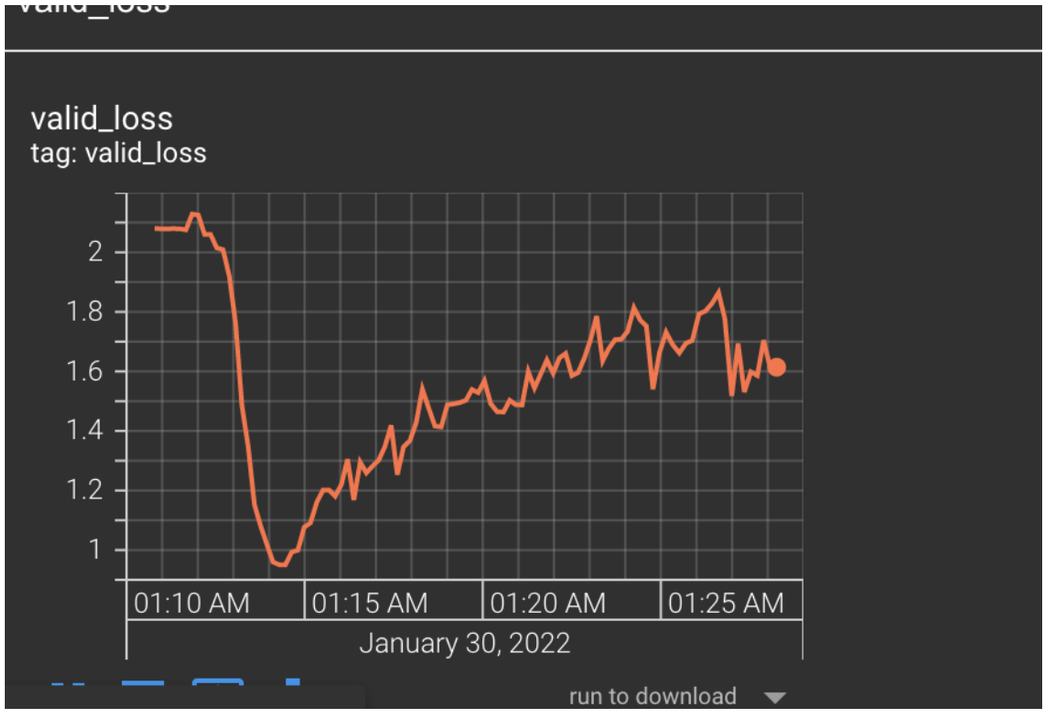

Figure 5: Validation loss

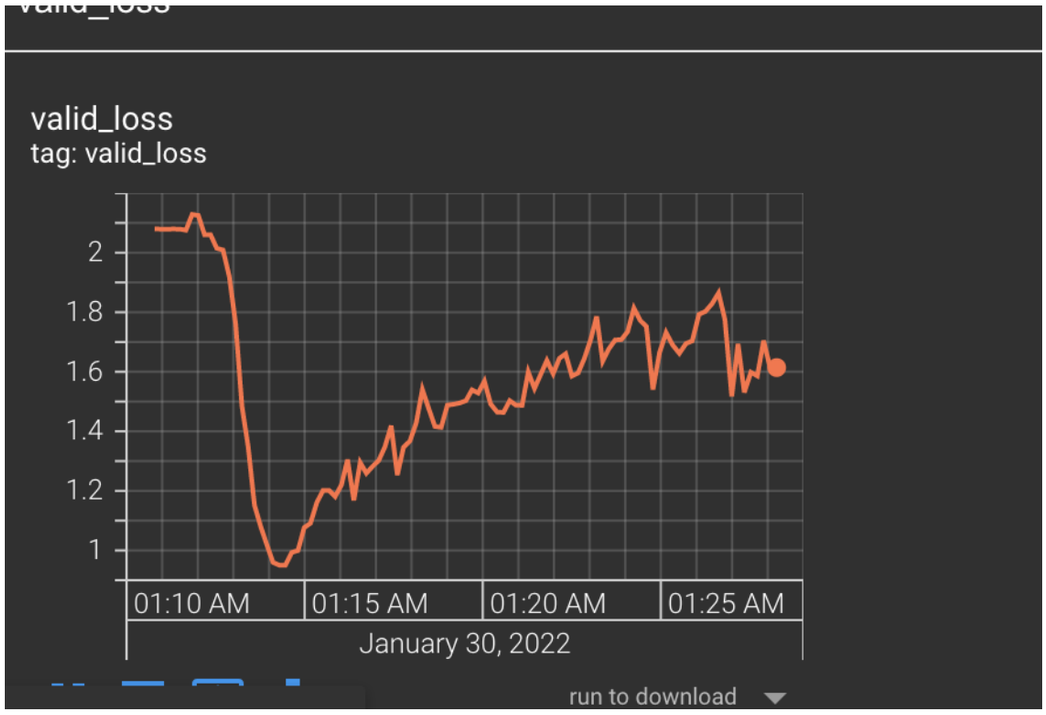

Figure 6: Validation loss