AI and ML Fundamentals

Hasan Isgandarli
Department of Process Automation Engineering
Baku Higher Oil School
Baku, Azerbaijan
hesenisgenderli999@gmail.com

I. Problem Definition

Aim of this laboratory is to build a dataloader for existing dataset class and model architecture in PyTorch. We will prepare each step for the training.

II. Introduction

This report focuses on the provided dataset to find the solution to the problem according to the laboratory work instructions. The most important steps will be included in this report, according to the task.

III. Task

A. Clone the Repository

I've forked the repository given in the task description, made my changes in the repository and pushed them to GitHub.

B. Take the dataset provided

I took the dataset from Kaggle and extracted the three folders named "train", "valid" and "test" to my "datasets/" folder. In the training code, I also changed "val" to "valid", since the folder naming was different than what was given in the provided repository. As an extra pre-processing step, I calculated norms of the training batch, and included in my repository.

C. Examine the Dataset class in your Repository

I edited the dataset retrieval file and made it retrieve the images and labels from the correct directories.

D. Take the ResNet Model

I took the ResNet18 model and changed its classifier to have 53 outputs. I also used pre-trained weights for the ResNet18 model for better results.

E. Set up the tensorboard

I also installed and ran tensorboard while training. After each epoch, the model's F1 score and Loss was plotted.

F. Run the training with 3 configurations

I have run the model with three different configurations and here are the tensorboard plots for Training Loss, Validation F1 Score and Validation Loss.

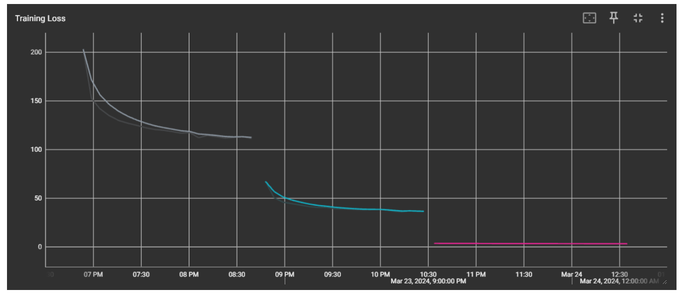

Fig. 1. Training Loss

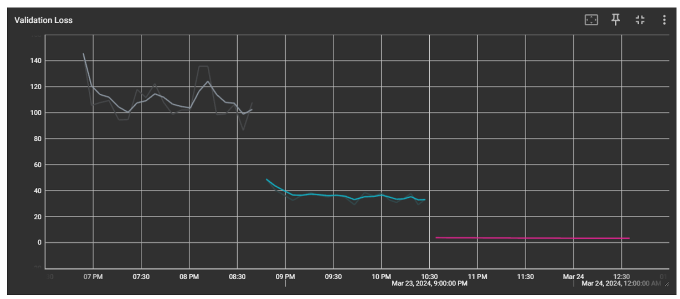

Fig. 2. Validation Loss

G. Explanation of the graphs

Looking at the graphs for Training Losses and Validation Losses, we can see that the model with 1.5 learning rate has a steady decreasing loss in the training set, but the validation does not strictly follow this tendency. The model with 0.5 learning rate is performing a bit better, it is somewhat following the same decreasing trend, but one cannot judge

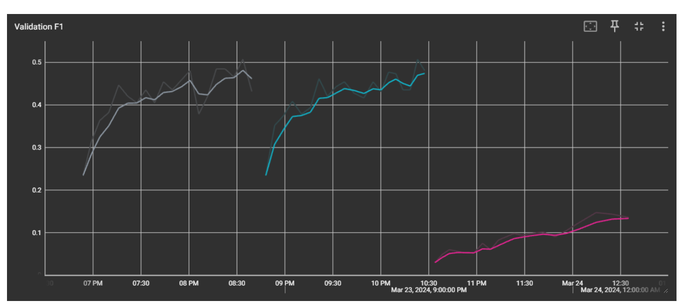

Fig. 3. Validation F1 Score

from only 20 epochs. Finally, the model with 0.00005 seems to have the least loss among them.

Looking at Validation F1 scores, we can see that the model with 1.5 learning rate is not that learning that well, and it may be due to the learning rate. The gradients. The model with 0.5 learning rate is also not consistent in the F1 score, as it happens to have some fluctuations in F1 scores as the epochs go on. This indicates that we need a lower learning rate, which is probably why the model with 0.00005 learning rate is doing so well. However, 20 epochs was not nearly enough for this model to learn all the features in the dataset, as the F1 score at the end, although increasing steadily, did not reach a high value. I believe that if trained for enough epochs (for example, 500) and with a slightly bigger learning rate, like 0.001, the features in the dataset can be learned by the model.

IV. Conclusion

In conclusion, in this laboratory work, Playing Cards Classification dataset and trained a ResNet18 model with three different learning rates and compared the results. This is a good laboratory work, because we learned how to gain insights on the training of the model just from testing different learning rates, and we did it with only 20 epochs. After a sufficient learning rate is determined by this method, the model can be set to train for hundreds of epochs, and we can have a good performing model that did not overfit or underfit.