Technical Report: Disruption Prediction Model

ludian

November 28, 2023

1 Introduction

In future tokamaks, major disruptions can be hazardous to the devices. Thus studies are made on the prediction algorithms to enable the future devices to adopt early safety measures in disruptive discharges. This set of code is composed under the competition of Multi-Machine Disruption Prediction Challenge for Fusion Energy by ITU. The main task for the competition is to sort out the disruptive discharges from the Alcator C-Mod discharges. The training set is a large number of discharges from the HL-2A tokamak and the J-TEXT tokamak, and a small number of discharges from the C-Mod tokamak. The test set is the discharges from the C-Mod tokamak, while the end of the discharges are chopped off by 40ms.

This Disruption Prediction Model is based on the machine learning algorithm. It classifies the discharges into normal discharges and the disruptive ones based on the training set. This model consists of a number of subnetworks, each of which processes a signal channel of the discharges (i.e. the plasma current (ip) channel). Then the results are summerized by a final voting network, which collect the outputs of the subnetworks and gives the final results. This simple and robust model performs fairly well and got F1 scores of above 0.9 in the competition.

2 The Models

In this set of code, the PyTorch module is applied. The subnetworks in the model are small-scale Convolutional Neural Networks (ssCNN). These ssCNNs are very easy to train, and their performances are very stable. The configuration of the ssCNNs are presented in Tab 1. Firstly, the signal is downsampled by an Average Pooling Layer. This network is dedicated to extract the most general and long-term features. This downsample layer can override the peripheral perturbations and make the network easier to extract the most important features. Then the rest of the network is of a classical CNN structure, only that the scale of the network is very small. The network consists only of 2 small convolutional layers and a small FC layer. A network of this size is already enough to extract all the key features, and a small size makes the network very robust.

Table 1: ssCNN Network Configurations

Layer	Parameters
Avg Pooling	kernel = 125
Convolution 1	in = 1, out = 4, kernel = 3
Leaky ReLU	slope = 0.01
Convolution 2	in = 4, out = 4, kernel = 3
Max Pooling	kernel = 4
Linear	in = 12 (156 for c3), out = 1
Sigmoid	–

The signals chosen to be processed by the networks are as follows:

- Plasma Current
- Horizontal and Vertical Displacements
- Plasma Density
- C3 Radiation
- Loop Voltage
- n = 1 Norm
- q95
- IP Error

These signals are chosen because that they have significant differences in disruptive discharges than in the normal discharges. The signals are proceeded with the ssCNN networks that having almost the same configuration (except for the C3 signal, because it has a higher sampling rate). The subnetworks are trained seperately with the different signal channels. The Models, the parameters, and the training codes for different signal channels are put in seperated directories: ip/, dx/, dy/, ne/, c3/, vl/, n1n/, q95/, and ipe/. The Adam optimizers and the BCELoss loss functions are adopted. The Multistep Learning Rate Scheduler is also used. For each channel the training curve is like FIG 1.

With the predictions from a single subnetwork, the performance on the test database can get a F1 score of 0.79 (which is my first submission). To make use of the results from multiple channels, we need a final network to combine them together. In this model, we adopt a voting network. It is also a very small network, consisting mainly of a FC layer. The main purpose of this network is to add weights and correlations among the subnetworks. The configuration of the voting network is shown in Tab 2. After connecting all the subnetworks with the voting network, the results can get a F1 score of 0.90.

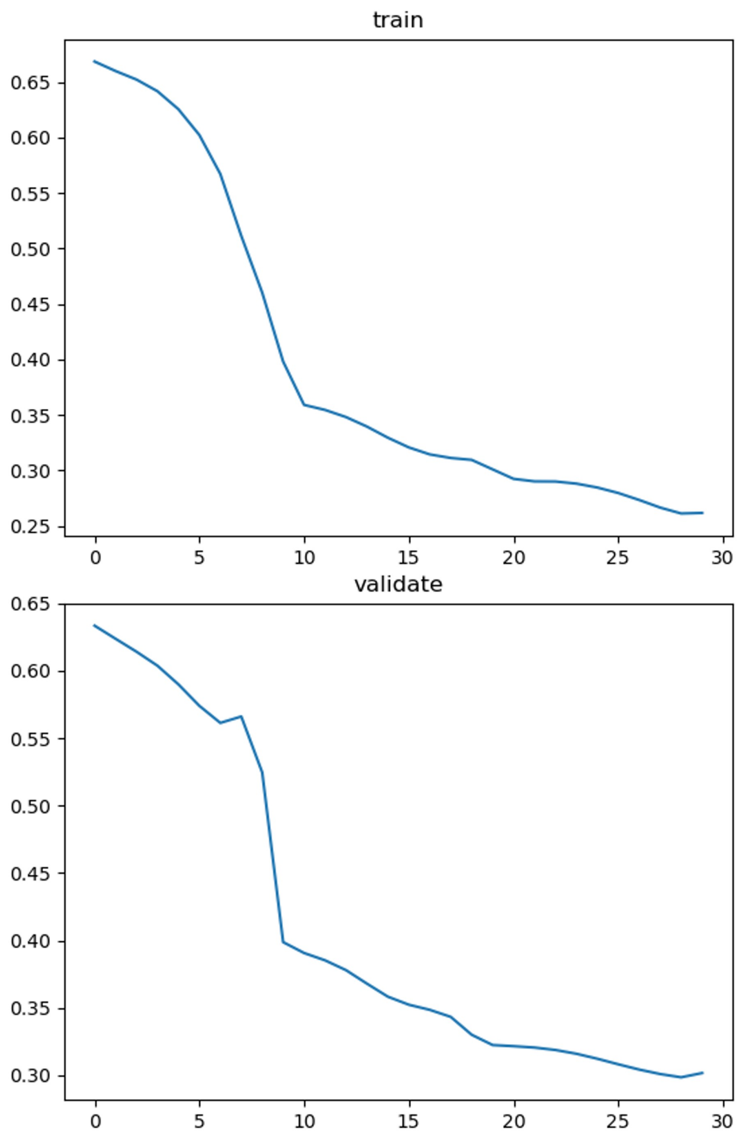

Figure 1: Training Loss and Validate Loss.

Table 2: Voting Network Configurations

Layer	Parameters
Linear	in = 9, out = 3
Max Pooling	kernel = 3
Sigmoid	–

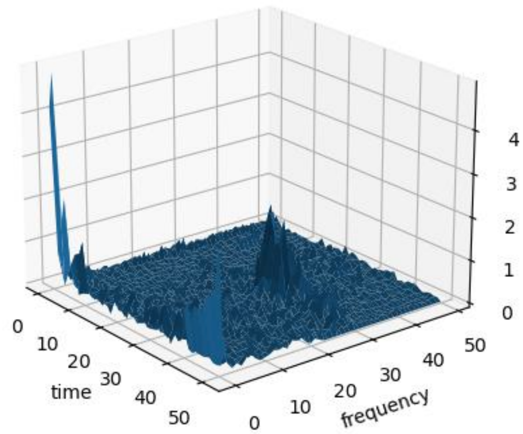

Figure 2: The FTgraph of a Mirnov signal.

3 Abnormal Signals

The signals may have abnormalities, and some of them are strongly correlated with disruptions. Making use of these abnormalities can greatly enhance the classification efficiency.

The most important abnormality is the signal length abnormality. In the test database, the disruptive discharges usually have some signal channels that are exceptionally short. Actually, simply by finding out the short signals can achieve a F1 score of 0.92. Although it seems like playing tricks, it can be useful. If a discharge shows the sign of termination much too earlier than the expecting ending time, it is more likely to be a disruption.

There are also other kinds of abnormalities. In some discharges, some of the signals are missing. In my model, I set the input for the voting network as -1 for those missing signals. However, there's no obvious enhancements of the performances when taking these missing signals into account.

4 Discussion

In this competition, I used a simple and robust model to handle the tasks. This model first proceeds the different signal channels seperately. Then the outputs of the subnetworks are connected together with a voting network. However, we can achieve a even better F1 score simply by finding out the short signals.

There are also some unsuccessful attempts. First, I only used the 20 C-Mod shots as the training database and I didn't use the HL-2A and the J-TEXT data. These tokamaks have significant differences in the system configurations, and I had difficulties in finding the common features within these three tokamaks. Also, I didn't managed to reproduce the sequence forcast method which is mentioned in [1]. And, my network model is not in a recurrent configuration, which means that it is not suitable to act as a real-time monitor of the disruption precursors. And, I've also made attempts to make use of the FFT method to extract features from the frequency domain, as shown in 2 which I also attached the codes in the file *MagneticAnalysis.py*. I tried to extract the features with 2D convolutional network along the temporal domain and the frequency domain. However, it does not make any differences.

References

[1] W Zheng, Q Q Wu, M Zhang, Z Y Chen, Y X Shang, J N Fan, Y Pan, and J-TEXT Team. Disruption predictor based on neural network and anomaly detection on j-text. *Plasma Physics and Controlled Fusion*, 62(4):045012, feb 2020.