機器學習概論 Final project

109612041 吳伯諺

Github link: https://github.com/PoYanWu/ML_Final_109612041

Model link: https://drive.google.com/file/d/1FYc74MSekEqXtqDeYGk58zxb8uQRCBTe/view?usp=share_link

Reference

Code參考:

https://github.com/kashyap333/Tabular-Playground-Series---Aug-2022

資料預處理參考:

https://www.kaggle.com/competitions/tabular-playground-series-aug-2022/discussion/342126

https://www.kaggle.com/competitions/tabular-playground-series-aug-2022/discussion/342319

Brief introduction

Python version: 3.9.7

使用 pytorch 的 nn 來完成這份作業

使用到的library:

Torch 1.13.1+cu116

numpy 1.23.5

torchvision 0.14.1+cu116

matplotlib 3.6.2

time

os

copy

csv

random

opencv-python 4.6.0.66

Data preprocessing

參考了幾篇 kaggle 上的做法，以及對資料的觀察，我將 data type 為 float 的資料(除 measurement_17 外)分為一組，計算其平均、標準差作為 feature，並把 attribute2、3 的乘積作為 feature，也看到有人分享 measurement3、5 的遺失是有參考價值的，所以也把 measurement_3 與 measurement_5 的遺失當作 feature，另外根據觀察後 train 跟 test 的 data 在 product_code 這項中完全不重疊因此刪除，而 attribute_0、attribute_1 也有一些不同時存在 train 與 test 的值，因此將其刪掉，而 interger 與 measurement_17 以外的 float 資料也都刪除，最終只留下 measurement_17, m_3_missing, m_5_missing, area, stdev, avg, loading, 等 7 個 feature 作為 NN 的 input。

Model architecture

Linear(in_features=7, out_features=8, bias=True)
ReLU ( )
Dropout(p=0.2)
Linear(in_features=8, out_features=8, bias=True)
Linear(in_features=8, out_features=1, bias=True)

loss_fn = nn.SmoothL1Loss()

Hyperparameters

optimizer = torch.optim.Adam(model.parameters(), lr=0.0003)
batchsize = 1000

Summary

結果截圖:

討論:

比起將所有資料丟進 nn，剔除掉一些可能誤導的資料，或是用一些 function 將不同 columns 間的關係萃取出來，不僅減少計算的量，也提高了正確率。

當複雜的網路沒辦法得到很好的效果可以試著簡化他，說不定會有不錯的結果

心得:

第一次寫 Kaggle 公開的競賽題目，算是一個新的體驗，但如果沒有前面很多人嘗試得出來的建議或觀察，我想我很難自己找到這些方法，但經過這次作業，我知道了這些做法與觀察的方法，這些經驗將成為我往後的能力~