Background

You will use a real-world manufacturing dataset collected from the injection molding dataset (the file has a .mat extension) .

Fig. 1 below exhibits the injection molding machine. The injection molding machine is usually used to produce plastic trinkets, toys, water bottles, containers, etc. The raw material, plastic, is injected through a nozzle into a mold cavity cooled and hardened as per the layout of the cavity. In our case, the injection molding machine is used to produce a transparent mold where the goal of this assignment is to build a neural network classifier to monitor the quality of transparent mold.

The monitoring process is performed using 48 input attributes representing different machine parameters. 48 input attributes are outlined in Table 1. There exist two common defects in the transparent mold: weaving and short-forming. This forms a three-class classification problem. Fig. 2, 3, 4 respectively show the three product conditions: normal, weaving and short forming. The challenge of this problem lies in its non-stationary characteristic because two machine parameters, namely holding pressure and injection speed, are varied during the process runs. Please check the reference for further details.

Fig. 1 Injection Molding Machine

Fig. 2 Normal Case

Fig. 3 Weaving Case

Fig. 4 Short forming Case

Table 1. 48 Features of the injection molding machine

Number	Description
1	Screw volume, actual value
2	Material cushion, actual value
3	Dosage time, actual value
4	Cycle time, actual value
5	Mould heating circuit 1, actual value
6	Mould heating circuit 2, actual value
7	Mould heating circuit 3, actual value
8	Mould heating circuit 4, actual value
9	Mould heating circuit 5, actual value
10	Mould heating circuit 6, actual value
11	Plotting point 2 (Holding Pressure)*
12	Injection Speed*
13	Injection flow, actual value
14	Switch-over volume, actual value
15	Maximum injection pressure, actual value
16	Injection time, actual value
17	Cylinder heating zone 1, actual value
18	Cylinder heating zone 2, actual value
19	Cylinder heating zone 3, actual value
20	Cylinder heating zone 4, actual value
21	Cylinder heating zone 5, actual value
22	Opening force, actual value
23	Opening speed, actual value
24	Oil temperature, actual value
25	Mould temperature control unit 1, actual value
26	Nozzle stroke, actual value
27	Closing speed, actual value
28	Advancement speed, actual value
29	Retraction speed, actual value
30	Mould protection force, actual value
31	Temperature of support housing, actual value
32	Circumferential speed, actual value
33	Ejector pressure, nominal value
34	Ejector pressure, actual value
35	Nozzle 1 flow, nominal value
36	Nozzle 1 pressure, actual value
37	Injection torque, actual value
38	Injection rotational speed, actual value
39	Injection force of screw 1, actual value
40	Dosage torque, actual value
41	Dosage rotational speed, actual value
42	Hydraulic accumulator pressure, actual value
43	Charge pressure of accumulator, measured value
44	Mould-entry time, actual value
45	Part removal time, actual value
46	Maximum injection pressure, actual value
47	Back Pressure, Actual
48	Clamping Force, Actual

<u>**Task 1**</u>

Build a **three-layer feed-forward neural network** to solve the monitoring problem of injection molding machine. Use Pytorch and make sure it is executable in Google Colab environments so that I can test. The **proportion of training and testing samples is 70:30** where your model must deliver the **smallest testing error possible**. In that case, you need to select the number of nodes of hidden layers, the number of epochs, the learning rates, the mini-batch size, etc. that lead to the smallest testing error. You have to use the SGD optimizer. The evaluation metric here is the classification error. Do not use feature selection here.

Task 2

Study the **effect of network structure**: hidden nodes, hidden layers to the classification performance. That is, you try different network configurations and document the patterns in your comments for me to understand. It has to cover different aspects of network configurations such as shallow network, wide network, deep network etc.

Task 3

Study the **effect of learning rates**. Please document/comment your experiments well. You need to give correct conclusions and give suggestions on how learning rates should be set. This includes possible adaptive learning rates where the value increases or decreases as the increase of epochs.

Task 4

Study the **effect of mini-batch size**. You can set mini-batch size to be 1 (stochastic gradient descent), N (batch gradient descent) or any other size. The most important aspect is to be conclusive with your finding so that I can present the decisions well.

Notes:

- For task 2-4, the analysis should touch the issue of accuracy as well as complexity as this is what my research is focusing on. The complexity can be defined in the context of execution time.
- Do comment/document such that I can understand and present the findings
- .ipynb file
- Please use Pytorch and not any other different deep learning libraries. You can combine with other supporting libraries but the main code must be in Pytorch
- Codes must be executable under google colab environments