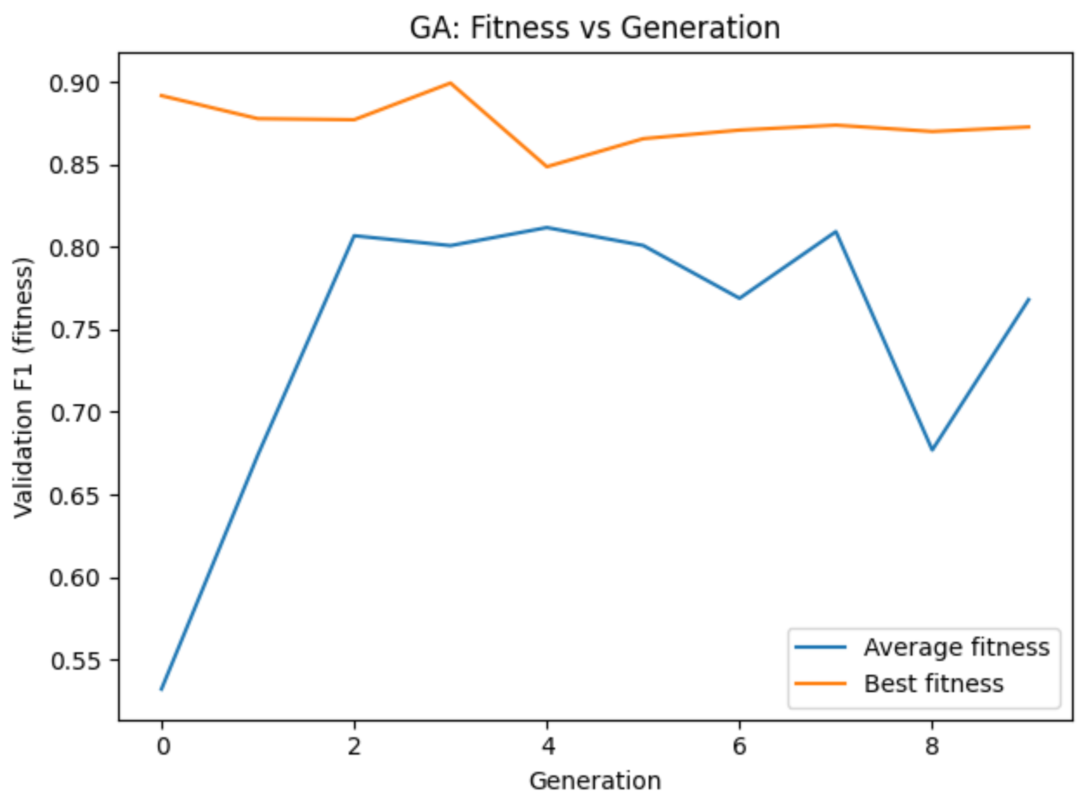
HW4 Report

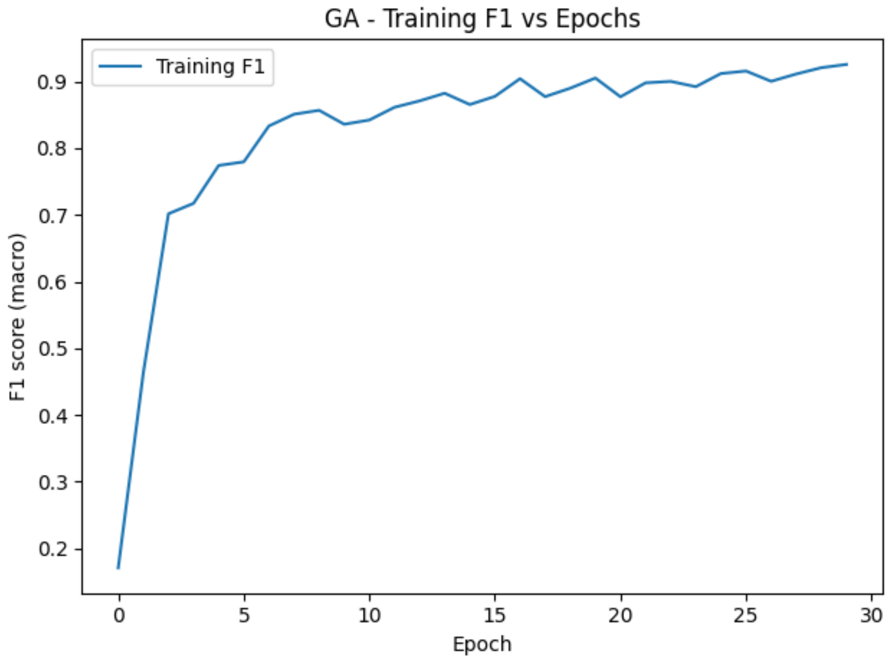
hmsun0813
https://github.com/hmsun0813/IEMS469/tree/main/hw4

Part 1 — Genetic Algorithm



For GA, the highest fitness score appears when mini batch size is 64, and the activation function is relu. The test F1 score is 0.9249.

Below is the plot of the training F1 score versus the training epochs.

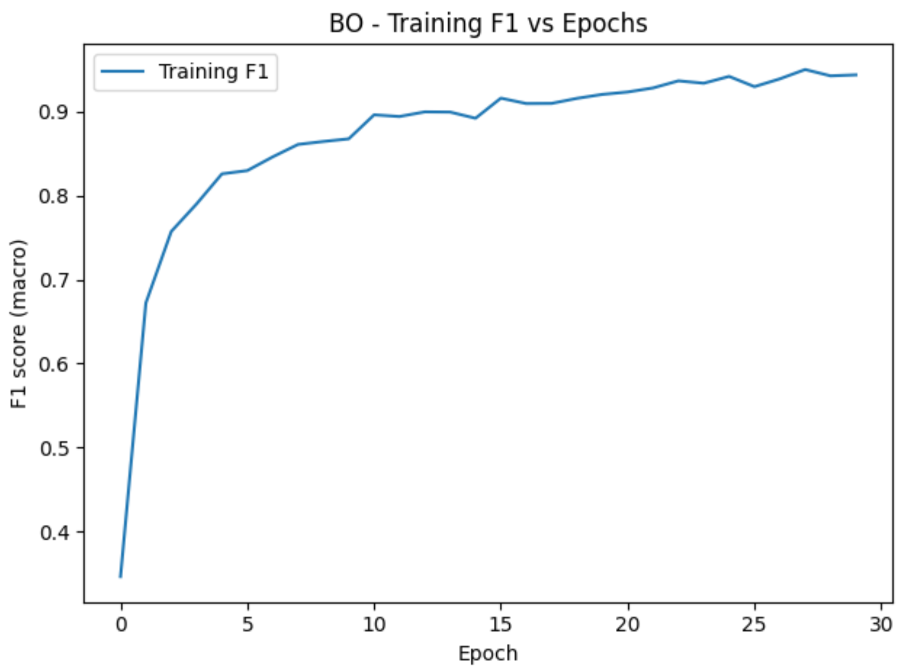


Part 2 — Bayesian Optimization

```
=== Running Bayesian Optimization ===
|   iter    |  target   | batch_... | act_id... |
-------------------------------------------------
| 1         | 0.8667021 | 2.2472407 | 1.9014286 |
| 2         | 0.0638124 | 4.3919636 | 1.1973169 |
| 3         | 0.8241266 | 0.9361118 | 0.3119890 |
| 4         | 0.6018039 | 0.3485016 | 1.7323522 |
| 5         | 0.0290435 | 3.6066900 | 1.4161451 |
| 6         | 0.0834647 | 1.6598045 | 1.1011250 |
| 7         | 0.8722232 | 1.9659301 | 1.9027724 |
| 8         | 0.8597878 | 2.2470423 | 1.9028473 |
| 9         | 0.8700089 | 2.1431429 | 1.8068443 |
| 10        | 0.8462176 | 2.3272513 | 1.7510713 |
| 11        | 0.8752310 | 1.8792881 | 1.7245162 |
| 12        | 0.8894317 | 1.7496184 | 1.8929915 |
| 13        | 0.8741961 | 1.6407195 | 1.6974295 |
| 14        | 0.8550579 | 2.4012951 | 0.4068968 |
| 15        | 0.8575644 | 1.5082056 | 1.8902374 |
| 16        | 0.8730458 | 1.3898542 | 1.6760570 |
| 17        | 0.8629409 | 1.2459003 | 1.8696850 |
| 18        | 0.8601993 | 1.1390889 | 1.6450631 |
| 19        | 0.8904832 | 0.9868003 | 1.8493057 |
| 20        | 0.8778509 | 0.8768141 | 1.6393418 |
=================================================
BO best hyperparameters: B=32, activation=tanh
```

Above is the progress output for BO. The best hyper parameters found by BO is mini batch size 32 and tanh as the activation function. The final test F1 score is 0.9439.

Below is the plot of the training F1 score versus the training epochs.



As the assignment results show, GA explored a broad range of hyperparameter combinations and produce a final test F1 of 0.9249. GA's evolutionary nature makes it especially effective at irregular or discrete search spaces, such as the categorical activation functions used in this assignment. It does not rely on any assumptions about smoothness in the objective function, which makes it robust and broadly applicable. However, this advantage comes at the cost of computational efficiency. Because GA requires evaluating many individuals over multiple generations, it tends to be slower to converge and requires more total model training runs.

In contrast, BO achieved a higher test F1 score of 0.9439 while using fewer model evaluations. BO builds a probabilistic surrogate model of the objective function and strategically chooses the next hyperparameters to evaluate using an acquisition function. This allows BO to balance exploration and exploitation So that it is significantly more sample-efficient than GA. BO converges more quickly because it uses information from all previous evaluations to guide the search. However, BO performs best when the hyperparameter space is low-dimensional and contains continuous variables. In this assignment, the discrete activation functions are not ideal for BO's underlying assumptions. BO can also struggle when the objective function is highly irregular or very high-dimensional, since the Gaussian Process–based surrogate model becomes harder to optimize and more computationally expensive.