CSC246 Machine Learning Assignment 2 Write-up

Qingjie Lu

Netid: qlu7

Files

This submission contains the following files:

analysis.py

mlp.py

dataproc.py

modelFile

train_mlp.py

test_mlp.py

CSC246 Machine Learning Assignment 2 Writeup.pdf

analysis.py is the original file with the implementation of f score added.

dataproc.py is unchanged.

mlp.py is the main implementation of the neural network with all three requirement functions

modelFile is the save model file.

train_mlp.py is modified with f score and plotting.

test_mlp.py id unchanged

CSC246 Machine Learning Assignment 2 Writeup.pdf is this file and contains all the experimentation data and additional information.

Usage

The usage of the code is unchanged, and a example command is provided below:

python .\train_mlp.py --train_file ../data/htru2.train --dev_file ../data/htru2.dev --hidden_units 11 --epochs 20 --batch 5

Extra credit note: I did implement the f score part.

Recommendations of the model parameters

Thoughts on Epoch: in general, definitely the more epochs, the better in terms of getting the best results, but more epochs on more hidden units could be more time consuming and unnecessary. In this case, 100 epoch does give a better result than 50, but the improvement is small. I would recommend 100 for the best result.

Thoughts on learn_rate: seems like 0.1 is the best one as well. A smaller learn rate could not achieve improvements. A bigger learn rate actually makes the results worse after more epoch.

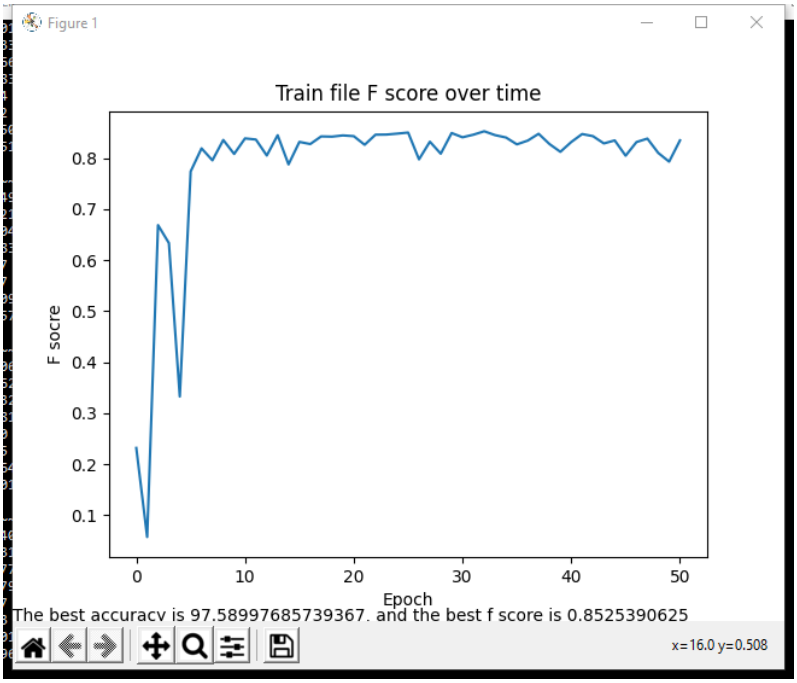
Thoughts on batch: batch seems to be mixed bag. A high or low batch would not yield a higher accuracy, a batch of 5 is appropriate for this particular problem.

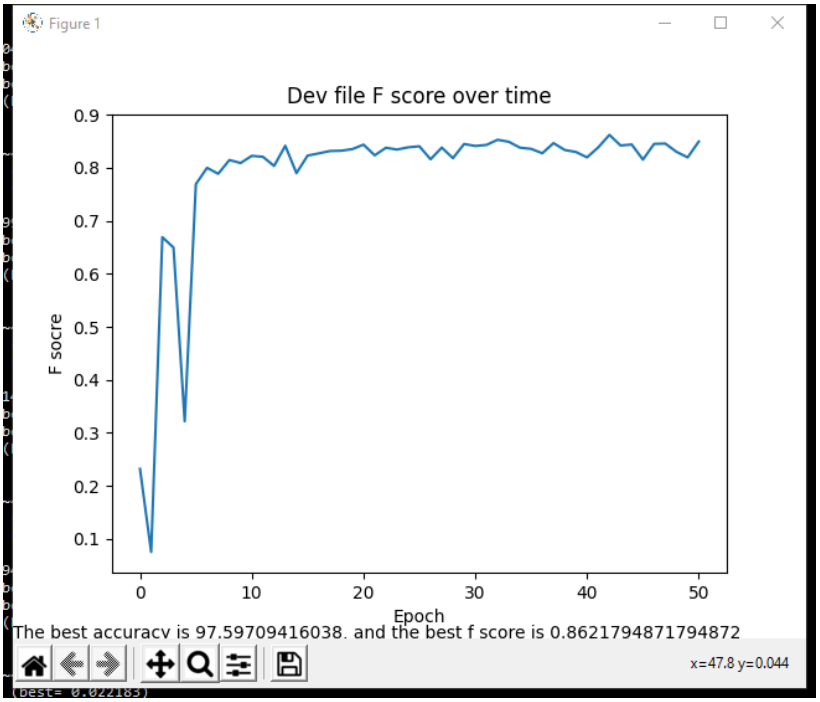
Thoughts on hidden_units: after experimenting on a couple of hidden units, 10 seems to be the best number. The reason why I don't believe I am overfitting is that the dev data follows the similar pattern as the f score does not get worse.

Experimentations

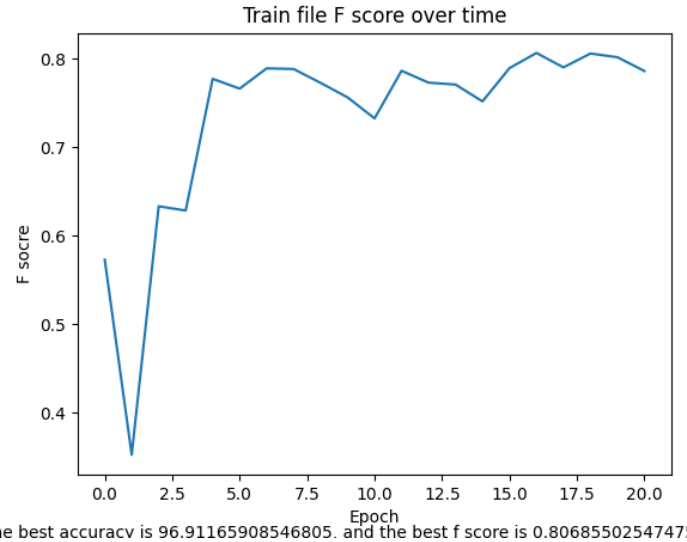
The following records a couple of selected results of the experiments.

--hidden_units 10 --epochs 50 --batch 5 --learn_rate 1e-1

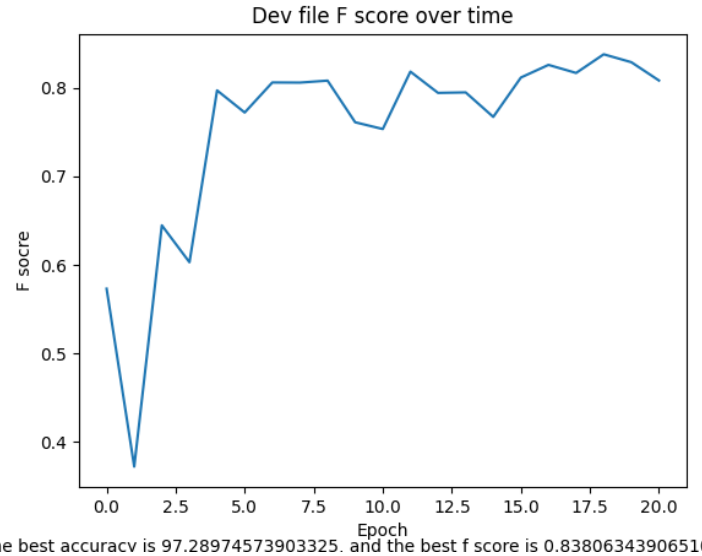




--hidden_units 10 --epochs 20 --batch 5 --learn_rate 1e-1

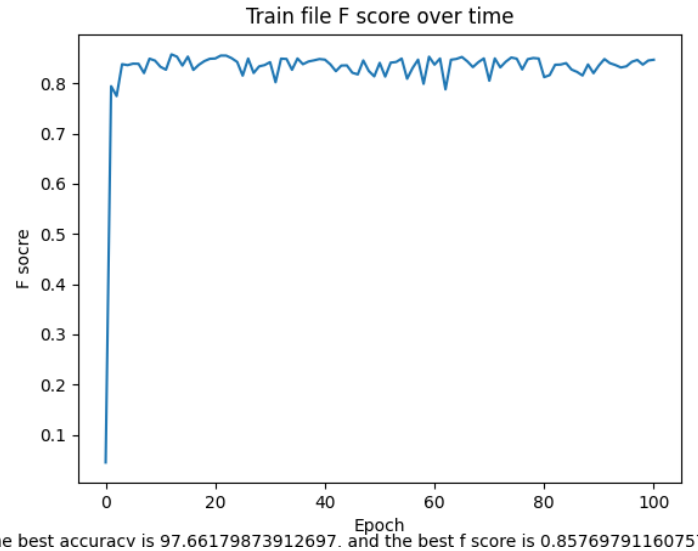


The best accuracy is 96.91165908546805. and the best f score is 0.8068550254747568

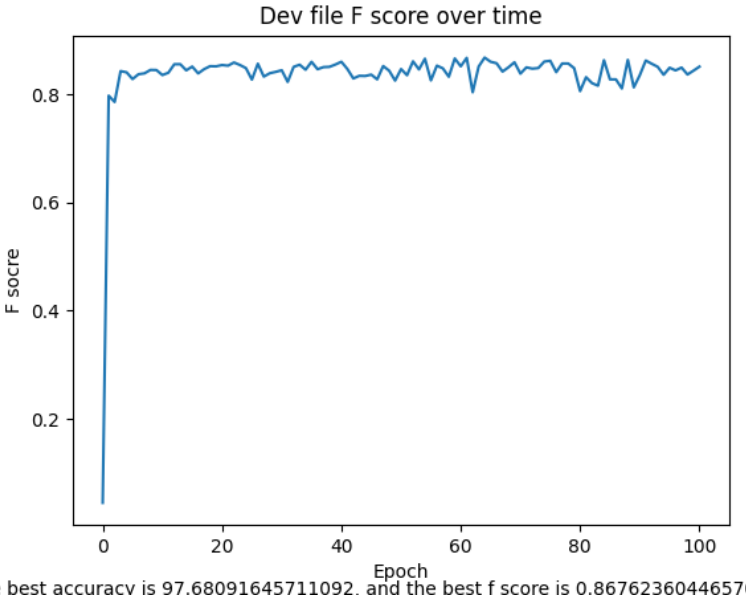


The best accuracy is 97.28974573903325. and the best f score is 0.8380634390651085

--hidden_units 10 --epochs 100 --batch 5 --learn_rate 1e-1

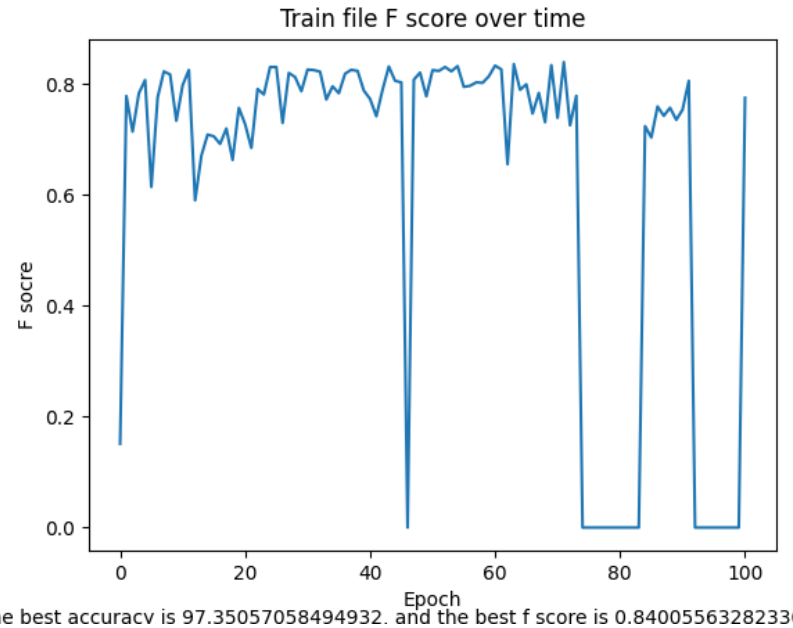


The best accuracy is 97.66179873912697. and the best f score is 0.8576979116075765

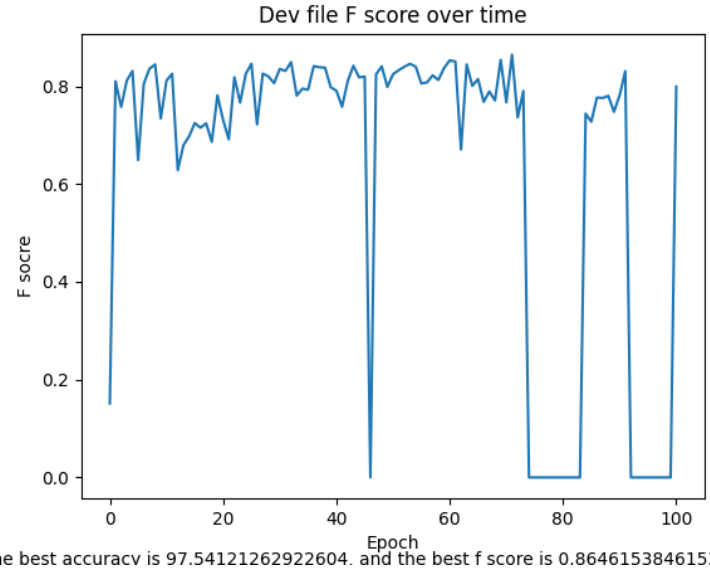


The best accuracy is 97.68091645711092. and the best f score is 0.8676236044657096

--hidden_units 10 --epochs 100 --batch 2 --learn_rate 1e-1

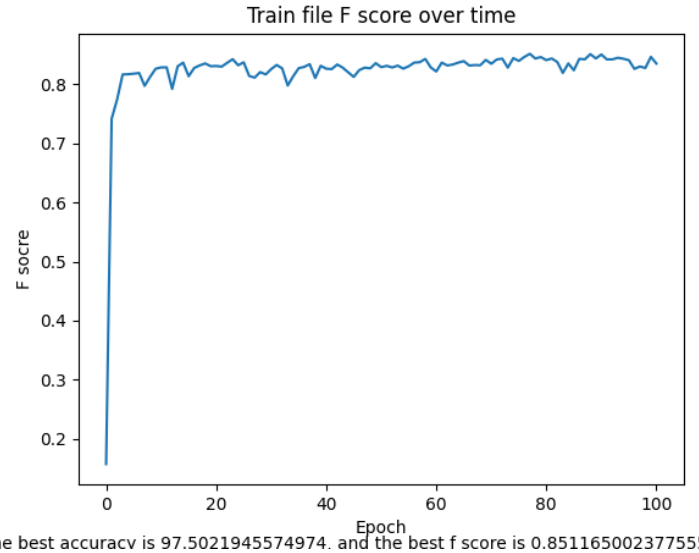


The best accuracy is 97.35057058494932. and the best f score is 0.8400556328233658

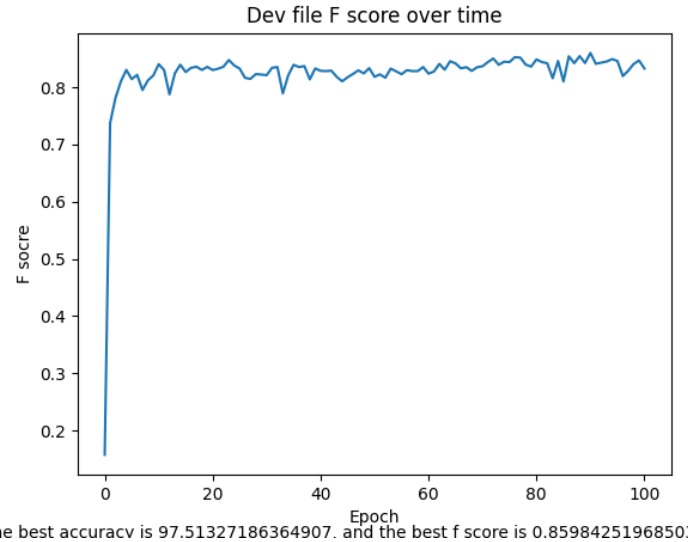


The best accuracy is 97.54121262922604. and the best f score is 0.8646153846153847

--hidden_units 10 --epochs 100 --batch 10 --learn_rate 1e-1

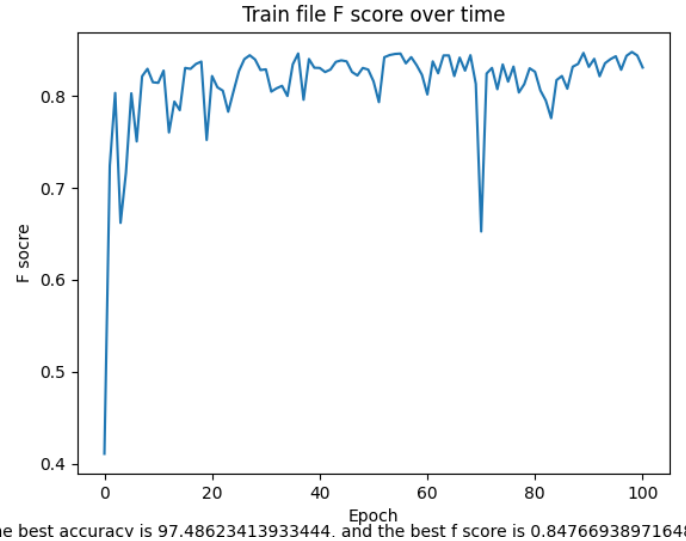


The best accuracy is 97.5021945574974. and the best f score is 0.8511650023775559

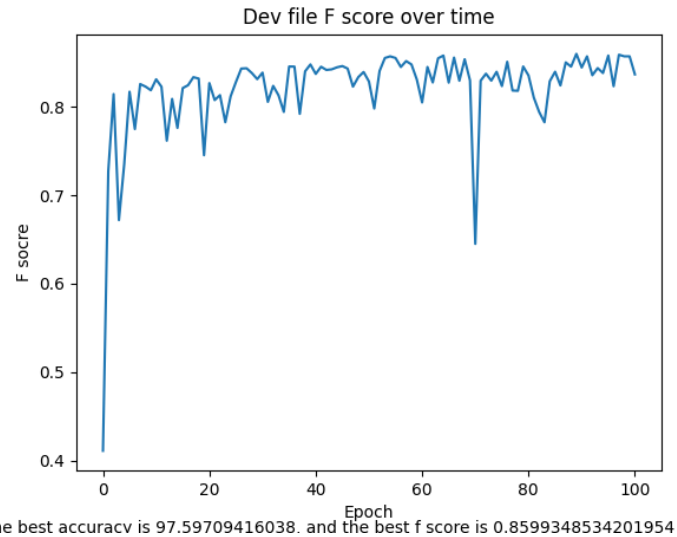


The best accuracy is 97.51327186364907. and the best f score is 0.8598425196850394

--hidden_units 6 --epochs 100 --batch 5 --learn_rate 1e-1

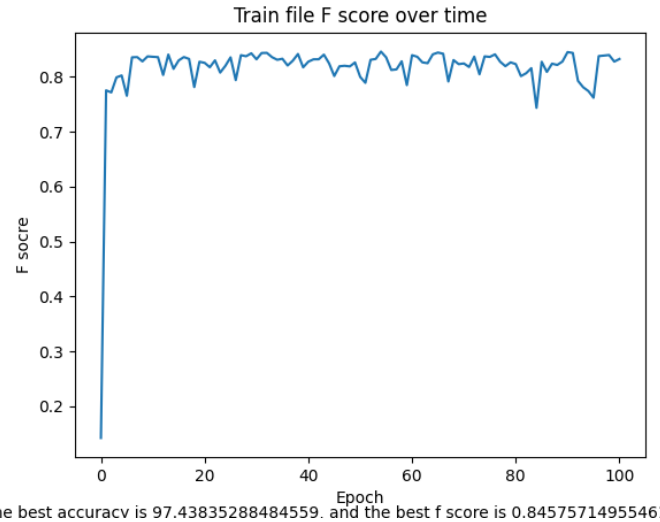


The best accuracy is 97.48623413933444. and the best f score is 0.8476693897164823

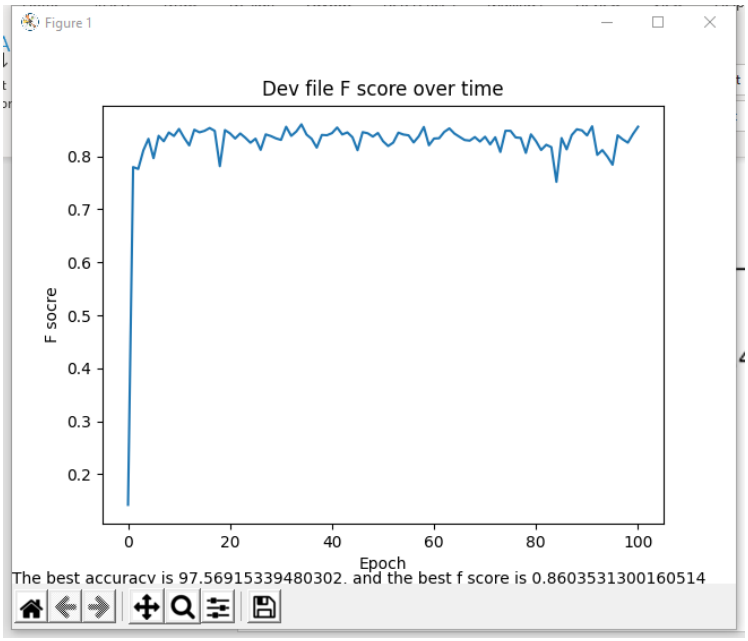


The best accuracy is 97.59709416038. and the best f score is 0.8599348534201954

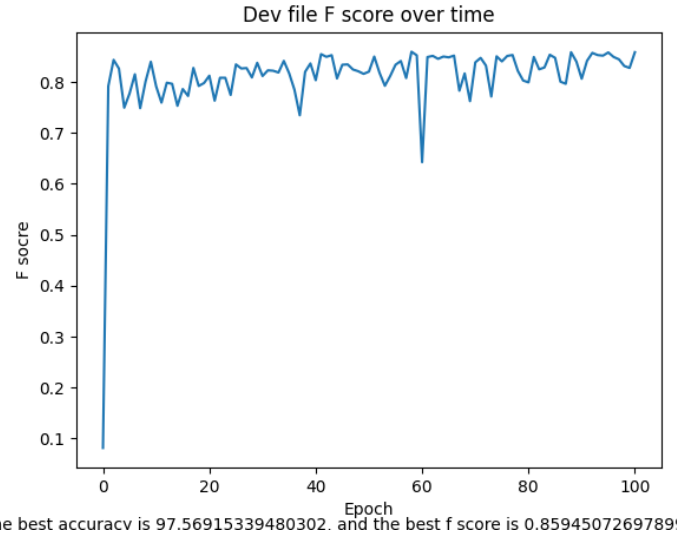
--hidden_units 16 --epochs 100 --batch 5 --learn_rate 1e-1



The best accuracy is 97.43835288484559. and the best f score is 0.8457571495546178

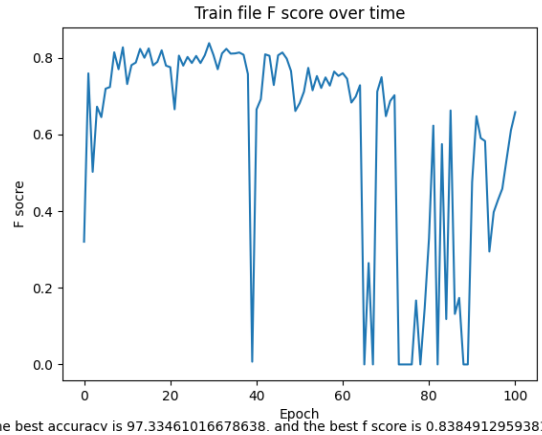


--hidden_units 10 --epochs 100 --batch 5 --learn_rate 2e-1

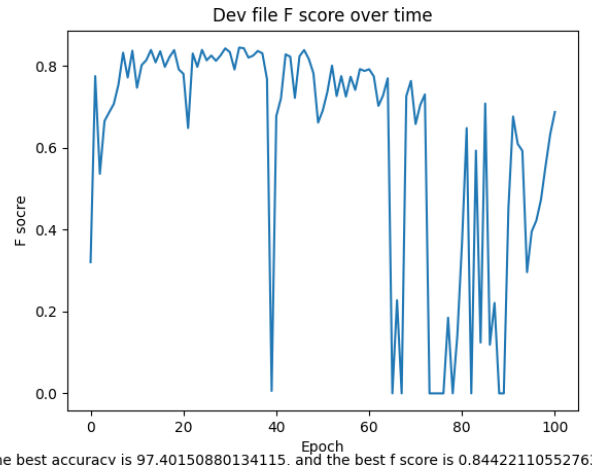


The best accuracv is 97.56915339480302. and the best f score is 0.8594507269789984

--hidden_units 10 --epochs 100 --batch 5 --learn_rate 3e-1

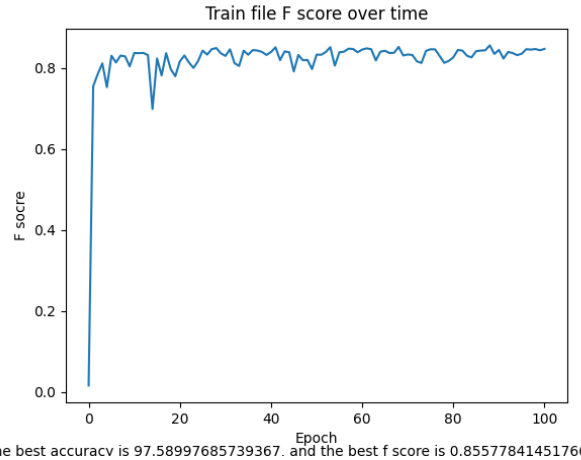


The best accuracv is 97.33461016678638. and the best f score is 0.8384912959381046

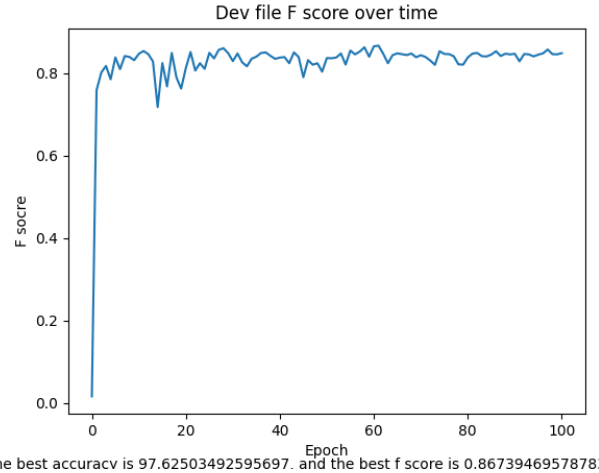


The best accuracv is 97.40150880134115. and the best f score is 0.8442211055276382

--hidden_units 10 --epochs 100 --batch 5 --learn_rate 0.7e-1

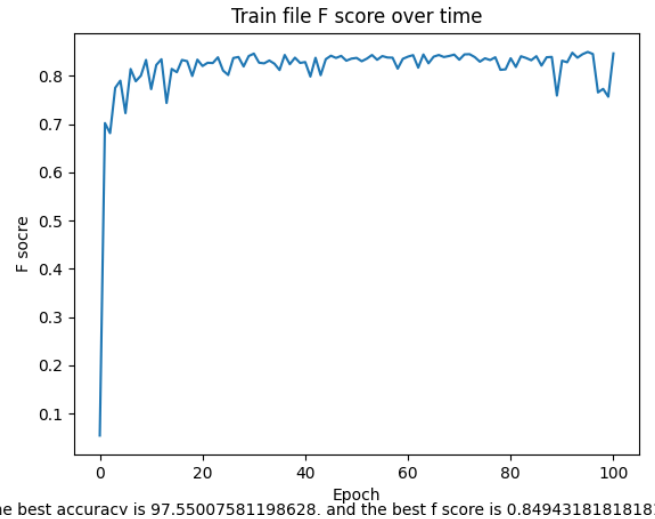


The best accuracy is 97.58997685739367. and the best f score is 0.8557784145176696

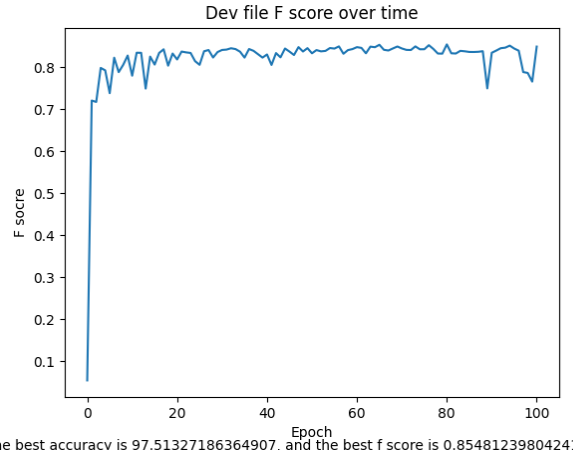


The best accuracy is 97.62503492595697. and the best f score is 0.8673946957878315

--hidden_units 11 --epochs 100 --batch 5 --learn_rate 1e-1



The best accuracy is 97.55007581198628. and the best f score is 0.8494318181818182



The best accuracy is 97.51327186364907. and the best f score is 0.8548123980424144

What I Learned

This project is probably one of the most difficult projects I have done in terms of the internal logic. Other projects can be massive and messy, and you have to figure out the edge cases. This one, however, requires a precise understanding of the math behind the code and the carefulness of walking through the process without a mistake, for a small mistake could ruin the entire neural network. One interesting bug I have had is that I had the yhat and ydata backwards when calculating delta2, and this has resulted mse metric and weights to go to a crazy huge number.

Another huge thing I learned is that I now have a much better understanding of the math behind backprop and softmax. I used to only have a vague understanding of the equations, but this project forced me to spend days grasping the dimensions of the matrices and understanding different parts of the equations.

Model quality

The modelFile that I am submitting here is the result of

--hidden_units 11 --epochs 100 --batch 5 --learn_rate 1e-1

Which has a accuracy rate of 97.736798

And f score of 0.872841