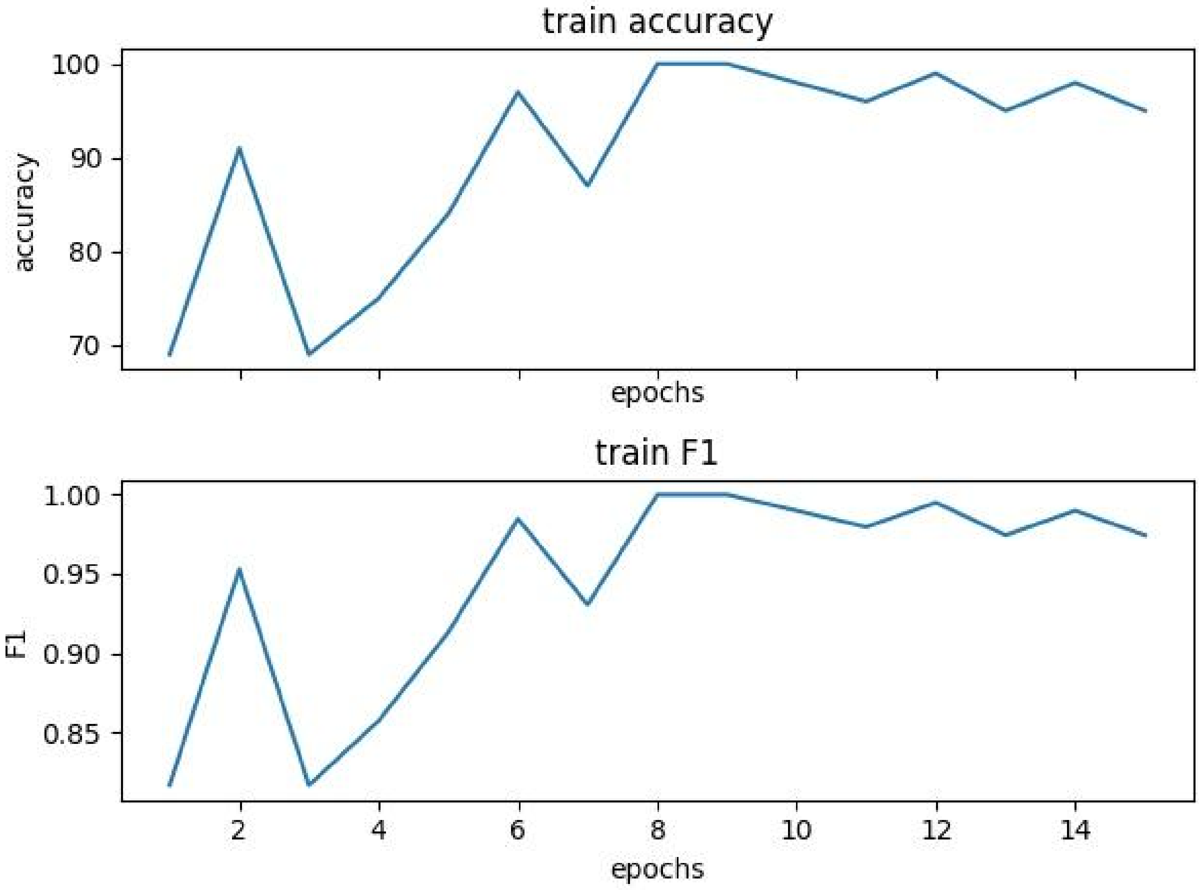
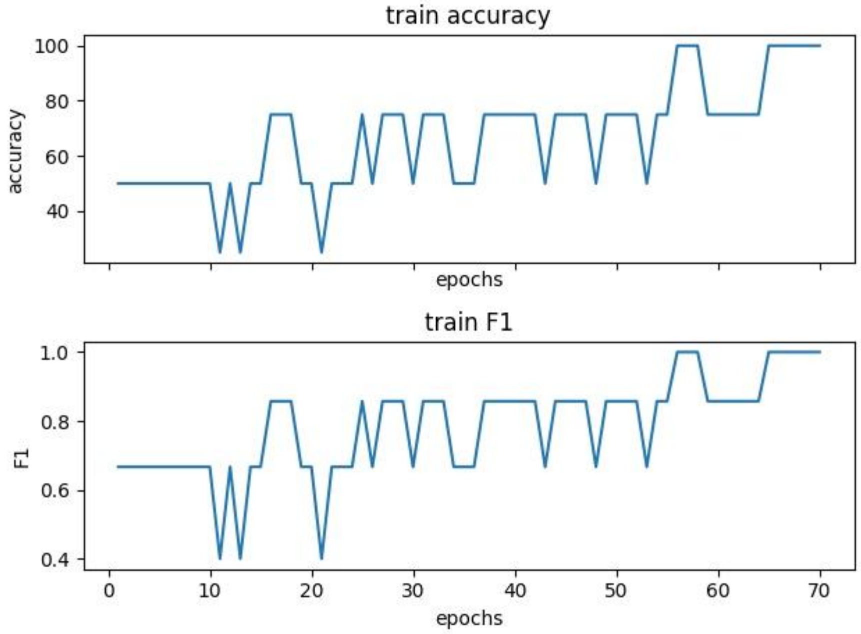
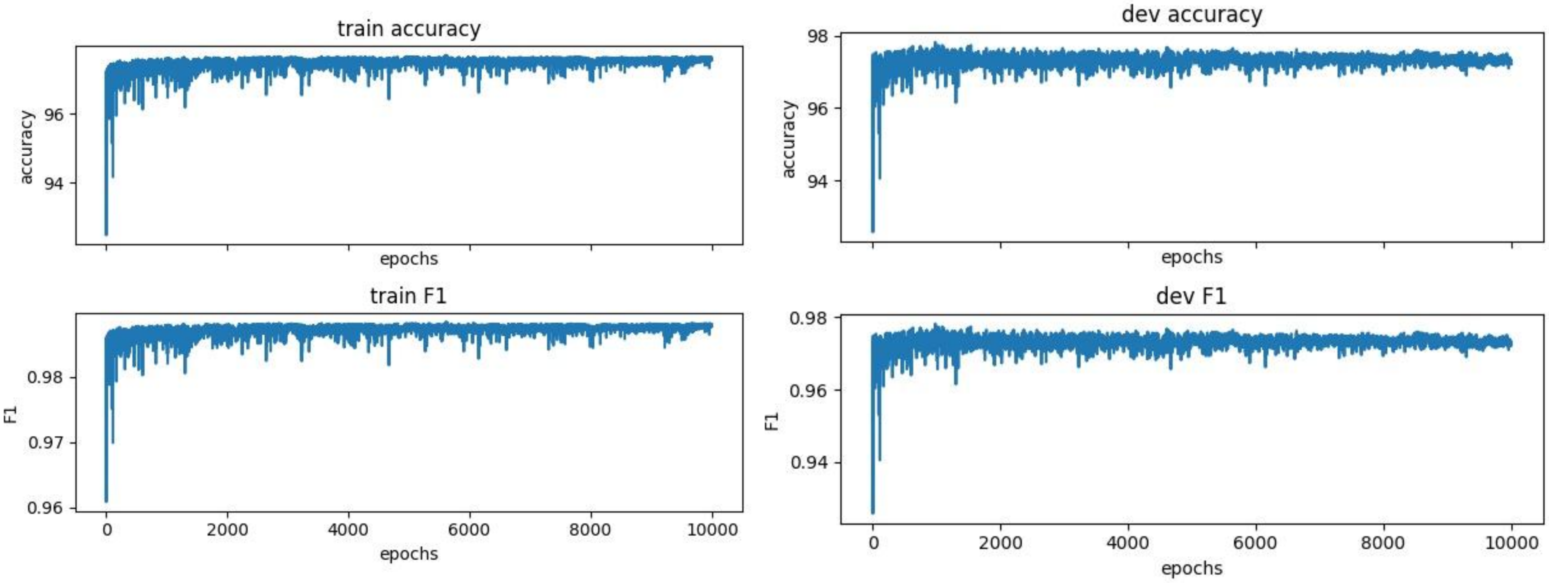
linearSmoke.dat
Number of hidden units: 1
Learning rate:1
Number of epochs: 15
Batch size:1



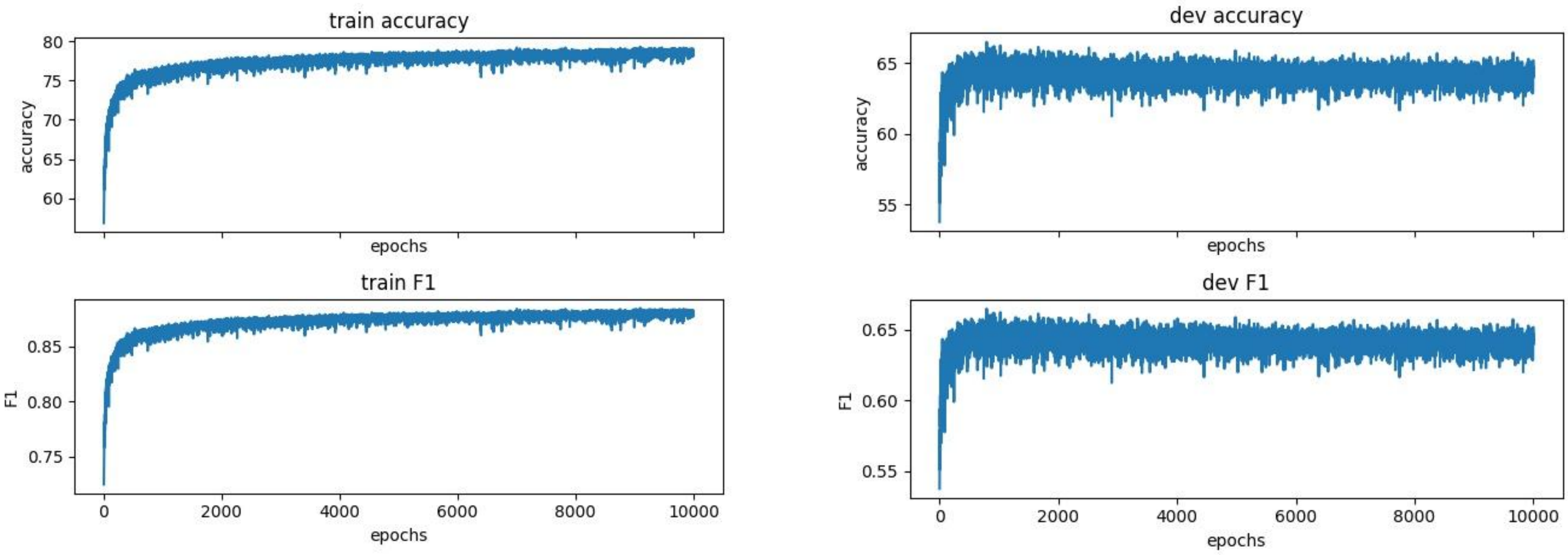
xorSmoke.dat
Number of hidden units: 2
Learning rate: 1
Number of epochs: 70
Batch size: 1



htru2
Number of hidden units: 15
Learning rate: 0.001
Number of epochs: 10000
Batch size: 5



higgs
Number of hidden units: 20
Learning rate: 0.005
Number of epochs: 10000
Batch size: 5



What I’ve learned:
I have learned the effect of learning rate on the model. Initially when I was testing my model, I used 0.01 for all dataset to get the feel of it. It turns out that linearSmoke.dat takes 20-30 epochs to reach 100% accuracy, xorSmoke.dat takes 100-120 epochs. I remembered that the perceptron algorithm basically has a learning rate of 1, so I tried to change the learning rate to 1 on both dataset and they converge much quicker, with the number of epochs specified above. What’s particularly intriguing is that the htru2 dataset just stayed fixed on 91.267% percent and doesn’t fluctuate at all. After asking the professor, I found out that it is because the learning rate is too high. I changed the learning rate to 0.001 and it works fine. Learning rate is also different from dataset to dataset. I tried 0.0001 for higgs.train first, and the algorithm learns really slowly and just doesn’t perform very well. Then I tried 0.001 and 0.005, each learns faster than before and 0.005 shows slightly better performance. I also tried 0.01, but it seems to be unable to learn after a certain amount of epochs, so it’s better to keep the learning rate smaller than 0.01.