CS434_HW3-4

Lyon Kee

November 23, 2023

1 Written Exercises: Analyzing Naïve Bayes [5pts]

1.1 Bernoulli Naïve Bayes As A Linear Classifier

1.1.1 Q1 Prove Bernoulli Naïve Bayes has a linear decision boundary [4pts]

Given:

$$P(y=1|x_1,...,x_d) = \frac{P(y=1)\prod_{i=1}^{d} P(x_i|y=1)}{P(x_1,...,x_d)} \propto \theta_1 \prod_{i=1}^{d} \theta_{i1}^{x_i}(1-\theta_{i1})^{1-x_i} \tag{1}$$

$$P(y=0|x_1,...,x_d) = \frac{P(y=0)\prod_{i=1}^{d} P(x_i|y=0)}{P(x_1,...,x_d)} \propto \theta_0 \prod_{i=1}^{d} \theta_{i0}^{x_i}(1-\theta_{i0})^{1-x_i} \tag{2}$$

$$\frac{P(y=1|x_1,...,x_d)}{P(y=0|x_1,...,x_d)} > 1 \tag{3}$$

Prove:

$$\frac{P(y=1|x_1,...,x_d)}{P(y=0|x_1,...,x_d)} > 1 \quad \rightarrow \quad b + \sum_{i=1}^{d} w_i x_i > 0$$

$$\frac{\theta_1 \prod_{i=1}^{d} \theta_{i1}^{x_i}(1-\theta_{i1})^{1-x_i}}{\theta_0 \prod_{i=1}^{d} \theta_{i0}^{x_i}(1-\theta_{i0})^{1-x_i}} > 1$$

$$\ln(\theta_1 \prod_{i=1}^{d} \theta_{i1}^{x_i}(1-\theta_{i1})^{1-x_i}) - \ln(\theta_0 \prod_{i=1}^{d} \theta_{i0}^{x_i}(1-\theta_{i0})^{1-x_i}) > \ln 1$$

$$\ln\theta_1 + \sum_{i=1}^{d} x_i \ln\theta_{i1} + \sum_{i=1}^{d}(1-x_i)\ln(1-\theta_{i1}) - \ln\theta_0 - \sum_{i=1}^{d} x_i \ln\theta_{i0} - \sum_{i=1}^{d}(1-x_i)\ln(1-\theta_{i0}) > 0$$

$$\ln\theta_1 - \ln\theta_0 + \sum_{i=1}^{d} x_i \ln\theta_{i1} - \sum_{i=1}^{d} x_i \ln\theta_{i0} - \sum_{i=1}^{d} x_i \ln(1-\theta_{i1}) + \sum_{i=1}^{d} x_i \ln(1-\theta_{i0}) > 0$$

$$\ln\theta_1 - \ln\theta_0 + \sum_{i=1}^{d} x_i(\ln\theta_{i1} - \ln\theta_{i0} - \ln(1-\theta_{i1}) + \ln(1-\theta_{i0})) > 0$$

Let: $b = \ln\theta_1 - \ln\theta_0; \quad w_i = \ln\theta_{i1} - \ln(1-\theta_{i1}) - \ln\theta_{i0} + \ln(1-\theta_{i0})$

$$b + \sum_{i=1}^{d} w_i x_i > 0$$

1.2 Duplicated Features in Naïve Bayes

1.2.1 Q2 Duplicate Features in Naïve Bayes [1pts]

Given:

$$P(y=0) = P(y=1) \tag{4}$$
$$P(y=1|X_1=x_1) > P(y=0|X_1=x_1) \tag{5}$$
$$P(X_2=x_2|y) = P(X_1=x_1|y) \tag{6}$$

Prove:

$$P(y=1|X_1=x_1) > P(y=0|X_1=x_1)$$
$$\frac{P(X_1=x_1|y=1)P(y=1)}{P(X_1=x_1)} > \frac{P(X_1=x_1|y=0)P(y=0)}{P(X_1=x_1)}$$
$$P(X_1=x_1|y=1) > P(X_1=x_1|y=0)$$

$$P(y=1|X_1=x_1, X_2=x_2)$$
$$= \frac{P(X_1=x_1, X_2=x_2|y=1)P(y=1)}{P(X_1=x_1, X_2=x_2)}$$
$$= \frac{P(X_1=x_1, X_2=x_2|y=1)P(y=1)}{P(X_1=x_1, X_2=x_2|y=1)P(y=1) + P(X_1=x_1, X_2=x_2|y=0)P(y=0)}$$
$$= \frac{P(X_1=x_1, X_2=x_2|y=1)}{P(X_1=x_1, X_2=x_2|y=1) + P(X_1=x_1, X_2=x_2|y=0)}$$
$$= \frac{P(X_1=x_1|y=1)P(X_2=x_2|y=1)}{P(X_1=x_1|y=1)P(X_2=x_2|y=1) + P(X_1=x_1|y=0)P(X_2=x_2|y=0)}$$
$$= \frac{P(X_1=x_1|y=1)^2}{P(X_1=x_1|y=1)^2 + P(X_1=x_1|y=0)^2}$$
$$= \frac{P(X_1=x_1|y=1)^2}{P(X_1=x_1|y=1)^2 + P(X_1=x_1|y=0)^2}$$

$$P(y=1|X_1=x_1)$$

$$= \frac{P(X_1 = x_1|y = 1)P(y = 1)}{P(X_1 = x_1|y = 1)P(y = 1) + P(X_1 = x_1|y = 0)P(y = 0)}$$
$$= \frac{P(X_1 = x_1|y = 1)}{P(X_1 = x_1|y = 1) + P(X_1 = x_1|y = 0)}$$
$$= \frac{P(X_1 = x_1|y = 1)^2}{P(X_1 = x_1|y = 1)^2 + P(X_1 = x_1|y = 1)P(X_1 = x_1|y = 0)}$$

We know:

$$\frac{P(X_1 = x_1|y = 1)^2}{P(X_1 = x_1|y = 1)^2 + P(X_1 = x_1|y = 0)^2} > \frac{P(X_1 = x_1|y = 1)^2}{P(X_1 = x_1|y = 1)^2 + P(X_1 = x_1|y = 1)P(X_1 = x_1|y = 0}$$

By looking at the difference in the denominator with:

$P(X_1 = x_1|y = 1) > P(X_1 = x_1|y = 0)$

Therefore:

$P(y = 1|X_1 = x_1, X_2 = x_2) > P(y = 1|X_1 = x_1)$

2 Implementing a Neural Network For Digit Identification [20pts]

2.1 Cross-Entropy Loss for Multiclass Classification

2.2 Implementing Backpropagation for Feed-forward Neural Network

2.2.1 Q3 Implementing the Backward Pass for a Linear Layer [6pt]

2.3 Analyzing Hyperparmeter Choices

2.3.1 Q4 Learning Rate [2pts]

A. Compare and contrast

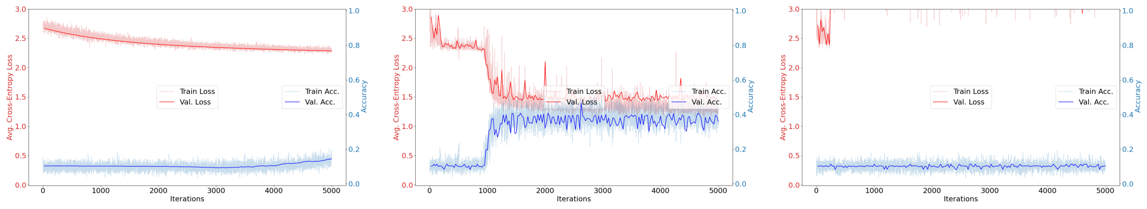

Figure 1: Left to right: plot for step sizes 0.0001, 5, 10

We observe that on stepsize 0.0001, our learning curve is still converging and it is doing so very slowly and smoothly for both the cross entropy loss and accuracy, reaching a Train Acc of 13.62% & Val Acc of 14.4%.
We observe that on stepsize 5, our learning curve is no longer converging, it is doing so quite quickly but it is jumping back and forth which tells us that the step size is too big for it to go deeper into the local minima. The cross-entropy loss and accuracy seem to both be stagnant, reaching a Train Acc of 39.78% & Val Acc of 36.0%.
We observe that on stepsize 10, our learning curve is not converging, it is like that starting from the very beginning but it is jumping back and forth which tells us that the step size is too big for it to go deeper into any local minima and it is easy for it to leave a local minima. The cross-entropy loss and accuracy seem to both be flat on average with entropy loss actually getting worse, reaching a Train Acc of 10.46% & Val Acc of 10.2%.

B. if the max epochs were increased
on stepsize 0.0001: it would continue to converge to the current local minima that it is in.
on stepsize 5: it would be unlikely for it to improve on performance as it is the best the model could learn given the large step size, however if we were to run multiple times there could be a chance for the performance to increase to 80% because it seems like the current model is stuck in a local minima or between a few that isn't great for the learning.
on stepsize 10: It is unlikely for it to improve, if lucky, we might arrive at a better accuracy but only for a split second and it is most likely to lose that accuracy in the next epoch due to the high learning step size.

2.3.2 Q5 ReLU's and Vanishing Gradients [3pts]

A. Compare and contrast

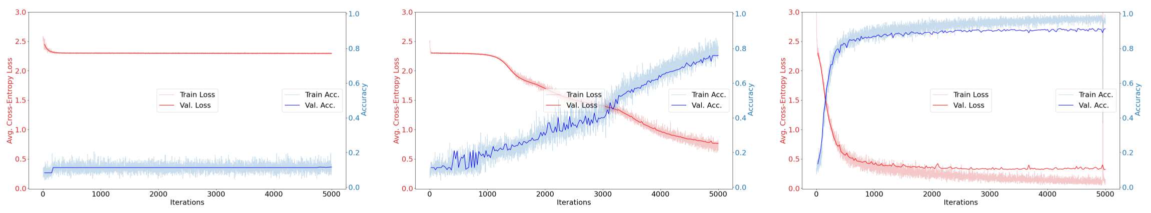

Figure 2: Left to right: 5 layer 0.01 step size, 5 layer 0.1 step size, 5 layer 0.01 step size ReLU

For the 5-layer 0.01 step size, we observe that it is unable to learn after about 200 iterations and has reached a local minima. It started off climbing then it remained stagnant and flat, reaching a Train Acc of 11.92% & Val Acc of 11.6%.
For the 5-layer 0.1 step size, we observe that it is still climbing and will have an improved accuracy if it is running on more epochs. It is climbing the entire time with obvious spikes and drops in certain epochs due to a high learning rate, reaching a Train Acc of 78.66% & Val Acc of 76.0%.
For the 5-layer ReLU, we observe that it has reached a local minima and is not increasing anymore. It climbed for a short time in the beginning first 1000 iterations steeply then it slowly converged to its accuracy. Reaching a Train Acc of 96.52% & Val Acc of 91.4%.
We observe that Train and Validation seems to be very similar for Sigmoid but it is different for ReLU with a slightly bigger difference. The difference was also slightly notable on 5-layer 0.1 step size.

B. If you observed increasing the learning rate in (2) improves over (1), why might that be?
In this case, (1) might have reached a local minima so (1) seems to be stagnant very quickly. but we see that it is unable to learn that quickly with a low learning rate due to the increased amount of hidden layers. Whereas the increased step size allows for bigger learning rates along all layers and it will slowly converge due to the large layers and it's ability to find better local minimas.

C. If (3) outperformed (1), why might that be?
Sigmoid squashes values between 0 and 1 making gradients small when it should be large, but ReLU doesn't squash it resolving the vanishing gradient issue with Sigmoid as values are close to 0 or 1.

2.4 Randomness in Training

2.4.1 Q6 Measuring Randomness [1pt]

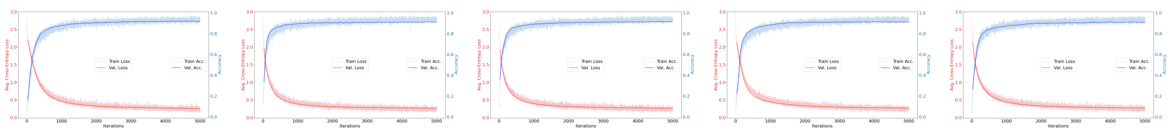
Figure 3: Left to right: seeds 1,2,3,4,5

seed 1: Val Acc: 91.9%
seed 2: Val Acc: 91.4%
seed 3: Val Acc: 91.4%
seed 4: Val Acc: 91.4%
seed 5: Val Acc: 91.0%

This tells us that randomness is a big part of a neural network and that we will arrive at slightly different accuracies. Therefore just like before, our plot might just have been unlucky and have arrived at a very shallow local minima and re-running the model would result in a different accuracy.

2.5 Make Your Kaggle Submission

2.5.1 Q7 Kaggle Submission [8pt]

The current submission has the following parameters:

```
step_size_arr = [0.03]
batch_size_arr = [1000]
max_epochs_arr = [600]
number_of_layers_arr = [5]
activation_arr = ["ReLU" if True else "Sigmoid" ]
```

3 Debriefing (required in your report)

3.1 Approximately how many hours did you spend on this assignment?

12

3.2 Would you rate it as easy, moderate, or difficult?

hard

3.3 Did you work on it mostly alone or did you discuss the problems with others?

Alone

3.4 How deeply do you feel you understand the material it covers (0%–100%)?

60%

3.5 Any other comments?

Q2 worth 1 point is the hardest.