Collab Worksheet 4

CS440/ECE448, Spring 2021

Week of 2/26 - 3/3, 2021

Question 1

You are a Hollywood producer. You have a script in your hand and you want to make a movie. Before starting, however, you want to predict if the film you want to make will rake in huge profits, or utterly fail at the box office. You hire two critics A and B to read the script and rate it on a scale of 1 to 5 (assume only integer scores). Each critic reads it independently and announces their verdict. Of course, the critics might be biased and/or not perfect, therefore you may not be able to simply average their scores. Instead, you decide to use a perceptron to classify your data. There are three features: a constant bias, and the two reviewer scores. Thus $f_0 = 1$ (a constant bias), $f_1 =$ score given by reviewer A, and $f_2 =$ score given by reviewer B.

Movie Name	A	B	Profit
Pellet Power	1	1	No
Ghosts!	3	2	Yes
Pac is bac	4	5	No
Not a Pizza	3	4	Yes
Endless Maze	2	3	Yes

(a) (5 points) Train the perceptron to generate $\hat{y} = 1$ if the movie returns a profit, $\hat{y} = -1$ otherwise. The initial weights are $w_0 = -1$,$w_1 = 0$,$w_2 = 0$. Present each row of the table as a training token, and update the perceptron weights before moving on to the next row. Use a learning rate of $\alpha = 1$. After each of the training examples has been presented once (one epoch), what are the weights?

Solution: The first row of the table is correctly classified, therefore the weights are not changed. The second row is incorrectly classified, therefore the weights are updated as $w = w + yf = [0, 3, 2]$. Using these weights results in misclassification of the third row, therefore the weights are updated again to $[-1, -1, -3]$. Using these weights results in misclassification of the fourth row, therefore the weights are updated again to $[0, 2, 1]$. These weights correctly classify the fifth row.

(b) (3 points) Suppose that, instead of learning whether or not the movie is profitable, you want to learn a perceptron that will always output $\hat{y} = +1$ when the total of the two reviewer scores is more than 8, and $\hat{y} = -1$ otherwise. Is this possible? If so, what are the weights w_0, w_1, and w_2 that will make this possible?

Solution: Yes, a perceptron can learn this function. Any weights such that $w_1 = w_2$ and $w_0 < -8w_1$ are correct; for example, the weights $[-8.1, 1, 1]$.

(c) (2 points) Instead of either part (a) or part (b), suppose you want to learn a perceptron that will always output $\hat{y} = +1$ when the two reviewers agree (when their scores are exactly the same), and will output $\hat{y} = -1$ otherwise. Is this possible? If so, what are the weights w_0, w_1 and w_2 that will make this possible?

Solution: This problem is the logical complement of the XOR problem, therefore it is not linearly separable, and cannot be learned by a perceptron.

Question 2

An image classification algorithm is being trained using the multiclass perceptron learning rule. There are 10 classes, each parameterized by a weight vector w_k, for $0 \leq k \leq 9$. During the last round of training, all of the training tokens were correctly classified. Which of the weight vectors were updated, and why?

Solution: None. The perceptron learning rule updates the weight vectors only if the classifier makes a mistake.

Question 3

Logistic regression is trained using gradient descent, with the goal of achieving the Bayes error rate (the lowest possible error rate) on testing data. There are many reasons why gradient descent might not successfully minimize the number of test-corpus errors. List at least three.

Solution: Here are a few:

1. **Wrong criterion:** The number of errors is not a differentiable criterion, so gradient descent has to minimize a differentiable approximation. Minimizing the differentiable approximation might not actually minimize the number of errors.
2. **Generalization error:** Minimizing error on the training corpus might not minimize error on the test corpus.
3. **Approximation error:** The Bayes error rate might not be achievable by a linear classifier. Since logistic regression learns a linear classifier, it might not be possible to achieve the Bayes error rate.
4. **Local optimum:** Gradient descent converges to a local minimum of the training criterion, not a global minimum.
5. **Computational limitations:** The amount of computation available for training might not be enough for gradient descent to fully converge.

Question 4

The softmax function is defined as

$$\hat{y}_k = \frac{\exp(e_k)}{\sum_j \exp(e_j)}$$

Find $d\hat{y}_5/de_3$ in terms of $\hat{y}_3$, $\hat{y}_5$, e_3 and/or e_5.

Solution:

$$\frac{d\hat{y}_5}{de_3} = -\frac{\exp(e_5)}{(\sum_j \exp(e_j))^2}\frac{d\sum_j \exp(e_j)}{de_3} = -\frac{\exp(e_5)\exp(e_3)}{(\sum_j \exp(e_j))^2} = -\hat{y}_5\hat{y}_3$$