CS7650 Problem Set 2 (Spring 2022)

February 8, 2022

Please submit your solutions on Gradescope.

1 Maximum Likelihood Parameter Estimation (25 points)

In binary Naive Bayes, show that the maximum-likelihood estimate for the class prior parameter is

$$P(y = 1) = \frac{c(y = 1)}{m}$$

where $c(y = 1)$ is the number of observations in the data containing the label $y = 1$ and m is the total number of observations.

Recall that Naive Bayes models the likelihood of the training data as follows (see slide 63 in the lecture on binary classification):

$$L(\theta) = \prod_{j=1}^{m} P(y_j, x_j) = \prod_{j=1}^{m} P(y_j) \left[\prod_{i=1}^{n} P(x_{ji}|y_j) \right] \tag{1}$$

$$= \theta_y^{c(y=1)} (1 - \theta_y)^{c(y=0)} \prod_{i=1}^{n} \prod_{y \in \{0,1\}} \theta_{x_i=1|y}^{c(x_i=1,y)} (1 - \theta_{x_i=1|y})^{c(x_i=0,y)} \tag{2}$$

Where the model's parameters are: $\theta_y = P(y = 1)$, and $\theta_{x|y} = P(x|y)$. $c(x_i = 1, y = 1)$ represents the number of training examples where x_i has the value 1 and y has the value 1.

Hint: Take the derivative of the Naive Bayes log-likelihood function with respect to θ_y, set equal and solve to find the value of θ_y that maximizes L(θ).

2 Logistic vs Softmax in Binary Classification (25 points)

Recall the Logistic and Softmax functions

$$P_{Logistic}(y = 1|\mathbf{x}) = \frac{e^{\mathbf{w}^T \mathbf{x}}}{1 + e^{\mathbf{w}^T \mathbf{x}}}$$

$$P_{Softmax}(y|\mathbf{x}) = \frac{e^{\mathbf{w}_y^T \mathbf{x}}}{\sum_{y' \in \mathcal{Y}} e^{\mathbf{w}_{y'}^T \mathbf{x}}}$$

Given $\mathcal{Y} = \{0, 1\}$, what should be the value of $\mathbf{w}$ in the logistic function such that $P_{Logistic}(y|\mathbf{x}) = P_{Softmax}(y|\mathbf{x}) \ \forall \ y \in \mathcal{Y}$? **Show your work**.
Hint: Think about $\mathbf{w}$ in terms of $\mathbf{w}_0$ and $\mathbf{w}_1$.

3 Dead Neurons (25 points)

Given below is the mathematical equation for a two layer-feedforward network with input x and scalar output y with RELU activation function.

$$z_i = \text{ReLU}\left(w_i^1 \cdot \boldsymbol{x} + b_i\right)$$
$$y = \boldsymbol{w^2} \cdot \boldsymbol{z}$$

The ReLU activation function can result in "dead neurons" i.e. neurons that can never be activated on any input. With regards to this information answer the following questions.

1. Under what condition is node z_i "dead"? Make sure to answer in terms of parameters w_i^1 and b_i
2. Let the loss function be l. Gradient of the loss l at a given instance is $\frac{\partial \ell}{\partial y} = 1$. Derive the gradients $\frac{\partial \ell}{\partial b_i}$ and $\frac{\partial \ell}{\partial w_{j,i}^1}$ for such an instance.
3. Using your answers to the previous two parts, explain why a dead neuron can never be brought back to life during gradient-based learning.
4. Suggest some modification to the activation function to overcome the above problem.

4 Back Propagation (25 points)

In lecture, we discussed the backpropagation algorithm in the context of a simple 2-layer feedforward neural network (see slide 96): https://aritter.github.io/CS-7650-sp22/slides/lec6-nn.pdf

$$P(\mathbf{y}|\mathbf{x}) = \text{softmax}(Wg(Vf(x)))$$

with the following conditional log-likelihood objective:

$$\mathcal{L}(\mathbf{x}, i^*) = W\mathbf{z} \cdot e_i^* - \log \sum_j \exp(W\mathbf{z}) \cdot e_j$$

where e_{i^*} is a the one-hot vector representing the gold label, i^*, and activations at the hidden layer, $\mathbf{z}$ are defined as follows:

$$\mathbf{z} = g(Vf(\mathbf{x}))$$

We saw how gradients on the output weight matrix, W, are the same as in multi-class logistic regression using $\mathbf{z}$ as features. We then derived gradients on the input weight matrix V as follows:

$$\frac{\partial \mathcal{L}(\mathbf{x}, i^*)}{\partial V_{ij}} = \frac{\partial \mathcal{L}(\mathbf{x}, i^*)}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial V_{ij}}$$

Your task in this question is to show that the "error at the hidden layer", $\frac{\partial \mathcal{L}(\mathbf{x}, i^*)}{\partial \mathbf{z}}$, can be computed from the "error at the network's output", $err(\text{root}) \stackrel{\text{def}}{=} e_{i^*} - P(\mathbf{y}|\mathbf{x})$, as follows:

$$err(\mathbf{z}) \stackrel{\text{def}}{=} \frac{\partial \mathcal{L}(\mathbf{x}, i^*)}{\partial \mathbf{z}} = W^T[e_{i^*} - P(\mathbf{y}|\mathbf{x})] \stackrel{\text{def}}{=} W^T err(\text{root})$$

Hint: Start with the log-likelihood objective defined above, and compute the derivative with respect to the vector $\mathbf{z}$.