Problem Set 3

June 20, 2022

DUE: Monday June 27 @ 23:59 EST

1. **Written Problems (50pts)** *Bishop 5.3*

   (a) (20pts) Consider a regression problem with multiple target variables (i.e. the output is a vector). For this problem, we will be using a Neural Network where we have unrolled the parameters into a single parametere vector θ (i.e. $\hat{\mathbf{y}} = f_\theta(\mathbf{x})$. Let us assume that the noise in the target variables is drawn from a Gaussian centered at the predictions with covariance $\mathbf{\Sigma}$ (i.e. $\mathbf{y}_{gt} \sim \mathcal{N}(\hat{\mathbf{y}}, \mathbf{\Sigma})$.

   Given a set of independent observations $D = \{(\mathbf{x}, \mathbf{y}_{gt})\}_{i=1}^N$:

   i. Let us assume that $\mathbf{\Sigma}$ is constant. Derive the error function that must be optimized in order to find the MLE θ^*

   ii. Now assume that $\mathbf{\Sigma}$ must also be learned from the data. Derive the MLE solution $\mathbf{\Sigma}^*$.

   iii. Note that θ^* and $\mathbf{\Sigma}^*$ are coupled (i.e. depend on each other). How can we find these solutions? Give a rough outline of an algorithm to find these solutions.

   (b) (30 points) Consider a feed-forward neural network with K layers and arbitrary error function $e(\hat{\mathbf{y}}, \mathbf{y}_{gt})$. Since the neural network has K layers, we can define the network with the following recursion:

$$\mathbf{a}_0 = \mathbf{x} \tag{1}$$
$$\mathbf{a}_k = f_k(\mathbf{W}_k \mathbf{a}_{k-1} + \mathbf{b}_k) \qquad k \in \{1, 2, \cdots, K\} \tag{2}$$
$$\hat{\mathbf{y}} = \mathbf{a}_K \tag{3}$$

   where $f_k(\cdot)$ is the activation function for layer k and is not necessarily the same as the activation function for other layers. Assume that the activation functions are all element-wise independent. When given a batch of data, we process the batch directly by vectorizing the feed-forward equations:

$$\mathbf{A}_0 = \mathbf{X}_{\text{batch}} \tag{4}$$
$$\mathbf{A}_k = f_k(\mathbf{A}_{k-1} \mathbf{W}_k + \mathbf{b}_k) \qquad k \in \{1, 2, \cdots, K\} \tag{5}$$
$$\hat{\mathbf{Y}} = \mathbf{A}_K \tag{6}$$

In this problem, derive the vectorized backpropagation equations for $\frac{\partial \mathbb{E}[e]}{\partial \mathbf{W}_k}$, $\frac{\partial \mathbb{E}[e]}{\partial \mathbf{b}_k}$, and $\frac{\partial \mathbb{E}[e]}{\partial \mathbf{A}_{k-1}}$. You should get the following equations:

$$\frac{\partial \mathbb{E}[e]}{\partial \mathbf{W}_k} = \mathbf{A}_{k-1}^T f_k'(\mathbf{A}_{k-1}\mathbf{W}_k + \mathbf{b}_k)\frac{\partial \mathbb{E}[e]}{\partial A_k} \tag{7}$$

$$\frac{\partial \mathbb{E}[e]}{\partial \mathbf{b}_k} = \mathbf{1}^T f_k'(\mathbf{A}_{k-1}\mathbf{W}_k + \mathbf{b}_k)\frac{\partial \mathbb{E}[e]}{\partial A_k} \tag{8}$$

$$\frac{\partial \mathbb{E}[e]}{\partial \mathbf{A}_{k-1}} = f_k'(\mathbf{A}_{k-1}\mathbf{W}_k + \mathbf{b}_k)\frac{\partial \mathbb{E}[e]}{\partial A_k}\mathbf{W}_k^T \tag{9}$$

2. **Programming Problem (50pts)** Please go to the following link and make a colab (`https://colab.research.google.com/drive/10gPho6gogdSAkp62Lri2M68QMBpH7Qk9?usp=sharing`) and create a copy for youself. In this question you will be implementing your own Neural Network package by completing the skeleton code I have provided. Specifically, you are applying the vectorized gradient rules that you derived above.