16-385 Computer Vision, Spring 2020

Take-home Quiz 9

Due Date: Sunday April 12, 2020 23:59

Question 1

As we discussed in class, neural networks use non-linear activation functions to connect their hidden layers.

1. Prove that if we have a *linear* activation function, then the number of hidden layers has no effect on the actual network.
2. A common activation function is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. Derive the gradient of the sigmoid function with respect to x, and show that it can be written as a function of $\sigma(x)$ itself—that is, if we know the value of the sigmoid, we can also compute its gradient without having access to x directly.
3. How does the gradient of the sigmoid activation function behaves as the absolute value of x increases? Can you think of any problems this behavior may create for the gradient descent algorithm, when the sigmoid is used as the activation function for many layers?
4. Often, the sigmoid function is replaced with the hyperbolic tangent function, $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$. Show that this function and its gradient can be written in terms of $\sigma(x)$.
5. What are the output ranges of the sigmoid and the hyperbolic tangent functions? When would we prefer to use each of these functions?

Question 2

Consider a loss function $L(\mathbf{x})$ defined for all vectors $\mathbf{x} \in \mathbb{R}^d$. As we say in class, a standard algorithm for minimizing this loss function is to perform a gradient descent iteration:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta(t)\nabla L(\mathbf{x}^t), \tag{1}$$

where $\nabla L(\mathbf{x})$ is the $d \times 1$ vector of the partial derivatives of L with respect to every coordinate of $\mathbf{x}$, that is, $\nabla L_i(\mathbf{x}) = \frac{\partial L(\mathbf{x})}{\partial x_i}$. There are many algorithms for setting the *step size* $\eta(t)$. Here, we will derive two such algorithms involving second-order derivatives of L.

1. Let $\mathbf{H}(\mathbf{x})$ be the $d \times d$ *Hessian* matrix of L, where $H_{ij}(\mathbf{x})$ equals $\frac{\partial^2 L(\mathbf{x})}{\partial x_i \partial x_j}$. Use $\nabla L(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ to write out the second-order expansion of L around a point $\mathbf{x}^t$.

2. Combine Equation (1) with the Taylor expansion above, and show that the resulting expression is minimized by selecting:

$$\eta^t = \frac{\|\nabla L(\mathbf{x}^t)\|^2}{(\nabla L(\mathbf{x}^t))^\top \mathbf{H}(\mathbf{x}^t)(\nabla L(\mathbf{x}^t))}. \tag{2}$$

3. Alternative, show that the Taylor expansion can be directly minimized by setting:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \mathbf{H}^{-1}(\mathbf{x}^t)\nabla L(\mathbf{x}^t). \tag{3}$$

   This alternative gradient-descent procedure is often called *Newton's algorithm*. How does it compare to the standard gradient-descent algorithm of Equations (1)-(2)?

4. Both the gradient-descent algorithm of Equations (1)-(2) and Newton's algorithm of Equation (3) suggest using the Hessian matrix to perform gradient descent. Can you think of any advantages or disadvantages of using the Hessian for optimization?

Instructions

1. **Integrity and collaboration:** Students are encouraged to work in groups but each student must submit their own work. If you work as a group, include the names of your collaborators in your write up. Plagiarism is strongly prohibited and may lead to failure of this course.

2. **Questions:** If you have any questions, please look at Piazza first. Other students may have encountered the same problem, and it may be solved already. If not, post your question on the discussion board. Teaching staff will respond as soon as possible.

3. **Write-up:** Your write-up should be typese in LaTeXand should consist of your answers to the theory questions. Please note that we **DO NOT** accept handwritten scans for your write-up in quizzes.

4. **Submission:** Your submission for this assignment should be a PDF file, `<andrew-id.pdf>`, composed of your write-up. **Please do not submit ZIP files.**