00137130/00101755: Deep Learning: Algorithms and Applications
Homework 1 *Due*: March 19, 2020

Note: Unless otherwise noted, section and equation numbers refer to those in the book by Goodfellow, Bengio, and Courville.

1. Consider the XOR learning problem described in Section 6.1.
   (a) For the MSE loss and linear output unit, verify that the solution is $\boldsymbol{w} = \mathbf{0}$ and $b = 1/2$.
   (b) Find the solution for the cross-entropy loss and sigmoid output unit.

2. Prove that the solutions to optimization problems (6.14) and (6.16) are the conditional mean and median of $\boldsymbol{y}$ given $\boldsymbol{x}$, respectively.

3. Numerical differentiation is an alternative approach to back-propagation for computing the gradient. This can be done, for example, by applying the central difference approximation

$$\frac{\partial J}{\partial \theta} = \frac{J(\theta + \varepsilon) - J(\theta - \varepsilon)}{2\varepsilon} + \text{remainder}$$

   to each parameter of the network.
   (a) Show that the remainder term is $O(\varepsilon^2)$.
   (b) Determine the time complexity of this algorithm and compare it with that of back-propagation.

4. It is mentioned in Section 7.5 that, "For some models, the addition of noise with infinitesimal variance at the input of the model is equivalent to imposing a penalty on the norm of the weights." State this formally for a feedforward network with MSE loss and prove your claim.

5. Consider a feedforward network with one hidden layer $\boldsymbol{h}$ and regularized loss (7.48), where $\Omega(h) = \|\boldsymbol{h}\|_1$. Devise a back-propagation algorithm to solve this problem.

6. Prove that the weight scaling inference rule is exact for regression networks with conditionally normal outputs.

7. State and prove a convergence theorem for stochastic gradient descent under conditions (8.12) and (8.13). *Hint*: See Robbins and Monro (1951).

8. In this exercise, we establish a convergence result for gradient descent with Polyak averaging.
   (a) Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T$ be an arbitrary sequence of vectors. Any algorithm with initialization $\boldsymbol{w}^{(1)} = \mathbf{0}$ and update rule

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \boldsymbol{v}_t$$

   satisfies

$$\sum_{t=1}^{T} \langle \boldsymbol{w}^{(t)} - \boldsymbol{w}^*, \boldsymbol{v}_t \rangle \leq \frac{\|\boldsymbol{w}^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{v}_t\|^2.$$

   (b) Let f be a convex, ρ-Lipschitz function, $\boldsymbol{w}^* = \arg\min_{\|\boldsymbol{w}\| \leq B} f(\boldsymbol{w})$, and $\bar{\boldsymbol{w}} = \sum_{t=1}^{T} \boldsymbol{w}^{(t)}/T$. Use part (a) to show that the gradient descent algorithm for minimizing f with $\eta = B/(\rho\sqrt{T})$ satisfies

$$f(\bar{\boldsymbol{w}}) - f(\boldsymbol{w}^*) \leq \frac{B\rho}{\sqrt{T}}.$$