Homework III

STAT 6910/7810-002 - Spring semester 2019

Due: Friday, February 15, 2019 - 5:00 PM

Please put all relevant files & solutions into a single folder titled `<lastname and initials>_assignment3` and then zip that folder into a single zip file titled `<lastname and initials>_assignment3.zip`, e.g. for a student named Tom Marvolo Riddle, `riddletm_assignment3.zip`. Include a single PDF titled `<lastname and initals>_assignment3.pdf` and any Python scripts specified. Any requested plots should be sufficiently labeled for full points.

Unless otherwise stated, programming assignments should use built-in functions in Python, Tensorflow, and PyTorch. In general, you may use the `scipy` stack [1]; however, exercises are designed to emphasize the nuances of machine learning and deep learning algorithms - if a function exists that trivially solves an entire problem, please consult with the TA before using it.

Problem 1

1. Provide a geometric interpretation of gradient descent in the one-dimensional case. (Adapted from the Nielsen book, chapter 1)

2. An extreme version of gradient descent is to use a mini-batch size of just 1. This procedure is known as online or incremental learning. In online learning, a neural network learns from just one training input at a time (just as human beings do). Name one advantage and one disadvantage of online learning compared to stochastic gradient descent with a mini-batch size of, say, 20. (Adapted from the Nielsen book, chapter 1)

3. Create a network that classifies the MNIST data set using only 2 layers: the input layer (784 neurons) and the output layer (10 neurons). Train the network using stochastic gradient descent on the training data. What accuracy do you achieve on the test data? You can adapt the code from the Nielson book, but make sure you understand each step to build up the network. Alternatively, you can use Tensorflow, or Pytorch. Please save your code as `prob1.py` and state which library/framework you

used. Report the learning rate(s) and mini-batch size(s) you used. (Adapted from the Nielsen book, chapter 1)

Problem 2

1. Alternate presentation of the equations of backpropagation (Nielsen book, chapter 2)
   Show that $\delta^L = \nabla_a C \odot \sigma'(z^L)$ can be written as $\delta^L = \sum'(z^L)\nabla_a C$, where $\Sigma'(z^L)$ is a square matrix whose diagonal entries are the values $\sigma'(z_j^L)$ and whose off-diagonal entries are zero.
2. Show that $\delta^l = ((w^{l+1})^T\delta^{l+1}) \odot \sigma'(z^l)$ can be rewritten as $\delta^l = \Sigma'(z^l)(w^{l+1})^T\delta^{l+1}$.
3. By combining the results from problems 2.1 and 2.2, show that $\delta^l = \Sigma'(z^l)(w^{l+1})^T \dots \Sigma'(z^{L-1})(w^L)^T\Sigma'(z^L)\nabla_a C$.
4. Backpropagation with linear neurons (Nielsen book, chapter 2)
   Suppose we replace the usual non-linear σ function (*sigmoid*) with $\sigma(z) = z$ throughout the network. Rewrite the backpropagation algorithm for this case.

Problem 3

1. Let $f : \mathbb{R}^d \to \mathbb{R}$. Show that if f is strictly convex, then f has at most one global minimizer.
2. Use the Hessian to give a simple proof that the sum of two convex functions is convex. You may assume that the two functions are twice continuously differentiable.
   Hint: You may use the fact that the sum of two PSD matrices is also PSD.
3. Consider the function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c$ where A is a symmetric $d \times d$ matrix. Derive the Hessian of f. Under what conditions on A is f convex? Strictly convex?
4. Using the definition of convexity, prove that the function $f(x) = x^3$ is not convex.
5. Using the fact that $f(x) = x^3$ is twice continuously differentiable, prove that f is not convex.
6. A function f is concave if $-f$ is convex. Prove that for all $x > 0$, the function $f(x) = \ln(x)$ is concave.
7. Let $f(x) = ax + b$ where $a, b \in \mathbb{R}$. f is called an *affine function* (a linear function plus an offset). Prove that f is both convex and concave. Does f have a global minimum or a global maximum? Are there any other functions that are twice continuously differentiable and both convex and concave that do not have the form of an affine function? Explain your reasoning.

References

[1] "The scipy stack specification." [Online]. Available: https://www.scipy.org/stackspec.html