Introduction to Deep Learning M2177.0043 April 4, 2020

Due on April 11, 2020
Seoul National University

Homework 2

INSTRUCTIONS

- The homework is due at 9:00am on April 11, 2020. Anything that is received after that time will be considered to be late and we do not receive late homeworks. We do however ignore your lowest homework grade.
- Homeworks need to be submitted electronically on ETL. Only PDF generated from LaTex is accepted.
- Make sure you prepare the answers to each question separately. This helps us dispatch the problems to different graders.
- Collaboration on solving the homework is allowed. Discussions are encouraged but you should think about the problems on your own.
- If you do collaborate with someone or use a book or website, you are expected to write up your solution independently. That is, close the book and all of your notes before starting to write up your solution.

1 Learning a binary classifier with gradient descent [40 points]

In this exercise, we will train a binary label classifier with the hinge loss. We wish to solve the following problem with $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$ by utilizing the hinge loss. Concretely,

$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{w}^\intercal \mathbf{x}_i) + \frac{\lambda}{2}||\mathbf{w}||^2$$

1. Derive the subgradient of the loss function.

2. Use $\lambda = 0.1, n = 1000, d = 100$, and use fixed step size of 0.01. Also, use the following code to generate the data set $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\intercal \in \mathbb{R}^{n \times d}$, the label vector $\mathbf{y} = [y_1, \ldots, y_n] \in \mathbb{R}^n$, and the initial value $\mathbf{w}^{(0)} \in \mathbb{R}^d$.

```
X = np.vstack([np.random.normal(0.1, 1, (n//2, d)),
               np.random.normal(-0.1, 1, (n//2, d))])
y = np.hstack([np.ones(n//2), -1.*np.ones(n//2)])
w0 = np.random.normal(0, 1, d)
```

   Then, solve the optimization problem of finding the optimal separating hyperplane $\mathbf{w}^*$ with gradient descent (run for 100 iterations). Attach the a) source code, b) iteration vs function value plot (function value in log scale), and c) iteration vs classification accuracy plot. Accuracy is defined by the fraction of data points your prediction $y_{\text{prediction}} = \text{sign}\,(\mathbf{w}^\intercal \mathbf{x}_i)$ matches the label y_i. Fix the random seed with `np.random.seed(1337)` in your code to make the experiment deterministic.

2 Matrix estimation with positive semidefinite constraint [60 points]

We seek to solve the following constrained optimization problem,

$$\underset{X \in \mathbb{R}^{n \times n}, X \succeq 0}{\text{minimize}} \ \langle S, X \rangle - \log\det(X) + \alpha \|X\|_1,$$

where $\|X\|_1 = \sum_{i,j} |X_{ij}|$.

Introduction to Deep Learning M2177.0043 April 4, 2020

Due on April 11, 2020

Seoul National University

Homework 2

In this exercise, we will using automatic differentiation utility called *autograd*. In the near future, we will understand exactly how auto diff works but for now, just treat it as a black-box which gives us the gradients. To install, run the following, run `pip install autograd`. Then, to import,

```
import autograd.numpy as np
from autograd import grad
np.random.seed(1337)
```

Using *autograd*, implement projected gradient descent (PGD) given the following setting

```
n = 5
A = np.random.normal(0, 1, (n, n))
S = A.dot(A.T)
alpha = 0.1
A0 = np.random.normal(0, 1, (n,n))
X0 = A0.dot(A0.T)
```

Use fixed step size of 0.01 and run PGD for 500 steps. Attach the a) source code and b) iteration vs function value after projection (function value in log scale) plot. In case it is not obvious, you need to research how to perform optimal projection of a matrix onto the PSD cone for the projection step.