Additional Exercises on Gradient Descent and Stochastic Gradient Descent
(Problems are from previous years' exams)
CS-526 Learning Theory

Short problems

1. [Several correct answers possible.] Let $(x_i, y_i) \in \mathbb{R} \times \{0, 1\}$ for $i \in \{1, \ldots, n\}$. Let $\hat{y}_i(w) = 1/\left(1 + e^{-wx_i}\right)$. Define

$$f : w \in \mathbb{R} \mapsto -\sum_{i=1}^{n} \left[y_i \log\left(\hat{y}_i(w)\right) + (1 - y_i) \log\left(1 - \hat{y}_i(w)\right)\right] + \lambda |w|,$$

   where $\lambda > 0$. The function f is:

   (a) convex.

   (b) differentiable everywhere.

   (c) subdifferentiable everywhere.

   (d) Lipschitzian.

2. You have lots and lots and lots of data. You use a neural net with a single hidden layer and run stochastic gradient descent. What theoretical framework(s) will likely give you meaningful insights for this situation? Explain why.

   (a) NTK as discussed in the course

   (b) mean field model as discussed in the course

   (c) basic learning theory generalization bounds

3. Assume that you are in a scenario where the mean field model that we discussed in the course applies. Can you use the machinery discussed in the paper by Montanari to optimize and predict the performance of an actual system? What, if any, are the remaining problems. Write down a few (and we mean a few) sentences to discuss.

4. (3 pts) Consider the function

$$f(x) = x^2 + 2.5 \cos x + |x|,$$

   defined on the real line $\mathbb{R}$. Which of the following statements is correct and why/why not? The function f is:

   (a) convex

   (b) differentiable everywhere

   (c) subdifferentiable everywhere

5. Let $g : \mathbb{R} \mapsto \mathbb{R}$ be a differentiable Lipschitz function with constant ρ. Define $h_\alpha : \mathbb{R}^d \mapsto \mathbb{R}$, with $h_\alpha(x) = g(\|x\|^\alpha)$ where $\alpha > 0$. For which values of $\alpha > 0$ can we conclude that h_α a Lipschitz function without further information on g ? Give a Lipschitz constant when this is the case.

A Conservation Law For Neural Networks

Consider a neural network (NN). To keep things as simple as possible assume that the activation functions have weights but no bias terms. Assume that we train the NN using gradient descent (GD). Let $\boldsymbol{w}$ denote the vector of weights. Let $L(\boldsymbol{w})$ denote our cost function (which depends of course on the given samples; but we suppress this dependence in our notation). Then, starting with an initial value $\boldsymbol{w}_0$, we proceed by computing the sequence $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta \nabla L(\boldsymbol{w}_{t-1})$ for a given step size η for a certain number of steps. This is our GD algorithm. As we already explored in the class, it is often easier to look at the continous-time version of this algorithm. This is called gradient flow (GF). The corresponding continuous-time version is the differential equation $\dot{\boldsymbol{w}}(t) = -\nabla L(\boldsymbol{w})$. GD (GF) is not the only possible algorithm. There are several variants that are used in practice, e.g., Nesterov's algorithm. For our purpose it is easiest to consider the so-called Newton dynamics (ND). The corresponding continuous-time version reads $\ddot{\boldsymbol{w}}(t) = -\nabla L(\boldsymbol{w}(t))$. For various reasons it is not used in practice but it is mathematically easier.

1. (6pts) Show that when we apply the ND to our system then $\frac{1}{2}\|\dot{\boldsymbol{w}}(t)\|^2 + L(\boldsymbol{w}(t))$ stays constant during the learning process, i.e., along the trajectory of the differential equation given by the ND. Why is this important? This says that the sum of the squares of the weights can grow by at most $L(\boldsymbol{w}(t=0))$, the initial loss. In particular, the weights cannot grow to infinity. This observation is important for the analysis.

2. (6pts) Assume further that $L(\boldsymbol{w}) = f(\|\boldsymbol{w}\|)$. Show that the (order-two antisymmetric) tensor $A = \boldsymbol{w}\dot{\boldsymbol{w}}^T - \dot{\boldsymbol{w}}\boldsymbol{w}^T$ (with $\boldsymbol{w}$ viewed as a column vector) stays constant during the learning process.

HINT: The proof is VERY easy. You have a function of time and want to show that it is constant.

P.S.: If we do not think of NNs but mechanics then $\|\dot{\boldsymbol{w}}(t)\|^2$ is the kinetic energy and $L(\boldsymbol{w}(t))$ is the potential energy. The statement is then the usual conservation law of energy. The second case corresponds to the conservation of the rotational momentum in case the potential is radially symmetric.

P.P.S.: We derived this conservation law for the ND. But similar expressions can be derived for other dynamics, such as GF.

P.P.P.S: In NN there are many other symmetries that stem e.g. from symmetries of the data or the activiation functions. It can be shown that each such symmetry leads to a conserved quantity.

Gradient Descent for Positive Semi-definite Matrices

Let $X, Y \in \mathbb{R}^{n\times n}$ be $n \times n$ real matrices and $A, B \in \mathbb{R}^{n\times n}$ be $n \times n$ real symmetric and positive definite matrices. Let $F : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ the function $F(X) = \frac{1}{2}\mathrm{Tr}X^T BX$.

1. (4 pts) Show that $F(X) \geq 0$ for any X.

2. (4 pts) Compute the second derivative of

$$f(s) = \mathrm{Tr}(sX^T + (1-s)Y^T)B(sX + (1-s)Y)$$

for $s \in [0, 1]$ and deduce that F is a convex function.

3. (4 pts) Deduce the inequality $F(Y) - F(X) \geq \mathrm{Tr}X^T B(Y - X)$. Is F Lipschitz ?

4. (4 pts) Consider now the function $G : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ with $G(X) = \frac{1}{2}\mathrm{Tr}(X - I)^T A(X - I)$ where I is the identity matrix. Define $L(X) = F(X) + G(X)$.

   (a) (2 pts) Write down the gradient descent algorithm for L. Call X_t the updated matrix at time t.

   (b) (2 pts) Assume that the operator norm $\|X_t\| \leq M$ stays bounded uniformly in n. Show that

$$\|\frac{1}{T}\sum_{t=1}^{T} X_t - (B + A)^{-1}A\| \leq \frac{2M}{\eta T}\|(B + A)^{-1}\|$$