Homework 3: Mathematics of Deep Learning (EN 580.745)

Instructor: René Vidal, Biomedical Engineering, Johns Hopkins University

Due Date: 12/17/2019, 11:59PM Eastern Time

Instructions.

- You can discuss the problems with your peers at a high level, but you must write your own solutions.
- You can consult relevant background material, but don't seek out solutions to the problems themselves. Cite the outside material use.
- You are free to use any of the well-known libraries (e.g. Tensorflow, PyTorch, Scikit-Learn etc.).
- Submit a zip file containing a single scanned PDF writeup along with your code and figures. For code, a Jupyter notebook is preferred. Typed writeups are nice, but clearly handwritten solutions are also fine.

1. Variant of matrix factorization dropout. Recall from lecture that dropout applied to matrix factorization,

$$\underset{\boldsymbol{U},\boldsymbol{V}}{\text{minimize}} \underset{\boldsymbol{z}\sim\text{Ber}(\theta)^r}{\mathbb{E}} \|\boldsymbol{X} - \boldsymbol{U}\,\text{diag}(\boldsymbol{z})\boldsymbol{V}^\top\|_F^2, \tag{1}$$

introduces regularization on the product $\boldsymbol{U}\boldsymbol{V}^\top$ whose *lower convex envelope* is the squared nuclear norm $\|\boldsymbol{U}\boldsymbol{V}^\top\|_*^2$. Furthermore, we showed that the dropout-induced regularizer is in fact equal to the squared nuclear norm, since the factors $(\boldsymbol{U},\boldsymbol{V})$ can always be balanced.

In this exercise we will examine a variant of matrix factorization dropout that, perhaps surprisingly, induces squared nuclear norm regularization more directly. First, let $\boldsymbol{U}\in\mathbb{R}^{D\times r}$, $\boldsymbol{V}\in\mathbb{R}^{N\times r}$. Assume without loss of generality that all rank-1 factors $\boldsymbol{u}_i\boldsymbol{v}_i^\top$ are unique[1]. We'll now define a distribution $\mathcal{D}_{(\boldsymbol{U},\boldsymbol{V})}$ that samples normalized rank-1 factors $(1/\|\boldsymbol{u}_i\boldsymbol{v}_i^\top\|_F)\boldsymbol{u}_i\boldsymbol{v}_i^\top$ with probability proportional to their contribution to $\boldsymbol{U}\boldsymbol{V}^\top$.

Formally, let $\mathcal{A} = \{\bar{\boldsymbol{u}}\bar{\boldsymbol{v}}^\top \mid \|\bar{\boldsymbol{u}}\bar{\boldsymbol{v}}^\top\|_F = 1\}$ be the set of unit norm rank-1 matrices, and define a probability mass $p:\mathcal{A}\to[0,1]$ for $\mathcal{D}_{(\boldsymbol{U},\boldsymbol{V})}$ as follows

$$p(\bar{\boldsymbol{u}}\bar{\boldsymbol{v}}^\top) = \begin{cases} \frac{\|\boldsymbol{u}_i\boldsymbol{v}_i^\top\|_F}{\sum_{i=1}^r \|\boldsymbol{u}_i\boldsymbol{v}_i^\top\|_F} & \text{if } \bar{\boldsymbol{u}}\bar{\boldsymbol{v}}^\top = \frac{\boldsymbol{u}_i\boldsymbol{v}_i^\top}{\|\boldsymbol{u}_i\boldsymbol{v}_i^\top\|_F} \text{ for some } i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Now let $\Theta(\boldsymbol{U},\boldsymbol{V}) \triangleq \sum_{i=1}^r \|\boldsymbol{u}_i\boldsymbol{v}_i^\top\|_F$, and consider the stochastic optimization problem

$$\underset{\boldsymbol{U},\boldsymbol{V}}{\text{minimize}} \underset{\{\bar{\boldsymbol{u}}_j\bar{\boldsymbol{v}}_j^\top\}_{j=1}^m\sim\mathcal{D}_{(\boldsymbol{U},\boldsymbol{V})}^m}{\mathbb{E}} \Big\|\boldsymbol{X} - \frac{\Theta(\boldsymbol{U},\boldsymbol{V})}{m}\sum_{j=1}^m \bar{\boldsymbol{u}}_j\bar{\boldsymbol{v}}_j^\top\Big\|_F^2, \tag{3}$$

where $m\geq 2$. According to this formulation, the data $\boldsymbol{X}$ are approximated by a random sample average of unit norm rank-one matrices, drawn independently from the distribution $\mathcal{D}_{(\boldsymbol{U},\boldsymbol{V})}$, which in turn is determined by the current factors $(\boldsymbol{U},\boldsymbol{V})$.

(a) **(15 points)** Compute the expectation in (3) to show that it is equivalent to

$$\underset{\boldsymbol{U},\boldsymbol{V}}{\text{minimize}}\ \|\tilde{\boldsymbol{X}} - \boldsymbol{U}\boldsymbol{V}^\top\|_F^2 + \frac{1}{m-1}\Big(\sum_{i=1}^r \|\boldsymbol{u}_i\boldsymbol{v}_i^\top\|_F\Big)^2, \tag{4}$$

where $\tilde{\boldsymbol{X}} \triangleq \frac{m}{m-1}\boldsymbol{X}$.

(b) **(5 points)** As we've seen, the regularizer in (4) is in fact the variational form of the squared nuclear norm. We have therefore established a more direct connection between (3) and the squared nuclear norm, compared with (1). Despite this direct equivalence however, (3) is not as straightforward to optimize by SGD as is (1). Explain why not in 1-3 sentences.

[1] If not, we can consolidate the redundant factors without affecting $\boldsymbol{U}\boldsymbol{V}^\top$.

2. Implicit regularization for under-determined linear regression. Consider the least-squares problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \ f(\boldsymbol{x}) \triangleq \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2, \tag{5}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^n$, and $\operatorname{rank}(\boldsymbol{A}) = n < m$. Furthermore, assume $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$ for some $\boldsymbol{x}^*$.

(a) **(5 points)** Derive a closed-form expression for the solution(s) to (5).

(b) **(5 points)** Prove that (5) satisfies the following smoothness condition

$$f(\boldsymbol{w}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{w} - \boldsymbol{x}\rangle + \frac{1}{2\eta}\|\boldsymbol{w} - \boldsymbol{x}\|_2^2, \tag{6}$$

for some explicit $\eta > 0$ depending on $\boldsymbol{A}$.

(c) **(5 points)** Prove that as a consequence, gradient descent with this fixed step size η converges to a critical point (hence a global minimizer) of (5). Recall that the gradient descent update can be written

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \eta\nabla f(\boldsymbol{x}^{(k)}). \tag{7}$$

(d) **(5 points)** Prove that if we initialize $\boldsymbol{x}^{(0)} = \boldsymbol{0}$, then gradient descent converges to the minimum ℓ_2-norm solution

$$\hat{\boldsymbol{x}}_{\ell_2} \triangleq \underset{\boldsymbol{x}}{\arg\min} \ \|\boldsymbol{x}\|_2 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{8}$$

(e) **(10 points)** Now, suppose instead that we optimize a non-convex "factorized" variant of (5)

$$\underset{\boldsymbol{x}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{A}(\boldsymbol{x} \odot \boldsymbol{x}) - \boldsymbol{y}\|_2^2, \tag{9}$$

where $\odot$ denotes the entrywise product. Interestingly, Gunasekar and co-authors proved in [1, Corollary 2] that this seemingly small change in the optimization problem significantly affects the induced regularization. Specifically, they showed that gradient descent with very small step size and initialization very close to zero converges to the **minimum ℓ_1-norm solution**

$$\hat{\boldsymbol{x}}_{\ell_1} \triangleq \underset{\boldsymbol{x}}{\arg\min} \ \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{10}$$

Validate this result experimentally. Specifically, using MATLAB or Python, do the following.

i) Generate $\boldsymbol{A} \in \mathbb{R}^{20\times 40}$ by sampling entries from the standard Gaussian distribution, and normalizing the columns to have unit ℓ_2 norm.

ii) Generate a sparse $\boldsymbol{x}^* \in \mathbb{R}^{40}$ with 10 non-zero entries whose locations are sampled uniformly at random. Generate the non-zero entries of $\boldsymbol{x}^*$ by sampling from a Gaussian distribution with standard deviation $1/\sqrt{10}$, and taking absolute value so that all entries are non-negative. Generate $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$.

iii) Implement gradient descent with a fixed step size η for problem (9).

iv) Initialize $\boldsymbol{x}^{(0)}$ near $\boldsymbol{0}$, e.g. $\boldsymbol{x}^{(0)} = 10^{-3}\boldsymbol{1}$, and run gradient descent for 300,000 steps with a small step size, e.g. $\eta = 10^{-3}$. Record the iterate $\boldsymbol{x}^{(k)}$ every 10 steps.

v) Compute the full Lasso solution path, e.g. using Scikit-Learn[2]. Recall that the Lasso problem is

$$\underset{\boldsymbol{x}}{\text{minimize}} \ g_\lambda(\boldsymbol{x}) \triangleq \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1. \tag{11}$$

The solution path then is the continuous curve $\lambda \mapsto \hat{\boldsymbol{x}}(\lambda) \triangleq \arg\min_{\boldsymbol{x}} g_\lambda(\boldsymbol{x})$. Note that in particular, as $\lambda \to 0$, the solution path converges to the minimum ℓ_1-norm solution (10).

vi) Plot the Lasso solution path[3] by plotting each coordinate $(\hat{\boldsymbol{x}}(\lambda))_i$ $(i = 1, \ldots, m)$ as a function of $-\log_{10}(\lambda)$. Similarly, in a separate subplot, plot the gradient descent iterates with iteration on the x-axis and each coordinate $\boldsymbol{x}_i^{(k)}$ $(i = 1, \ldots, m)$ as a separate line. Use a unique color for each coordinate so that they can be easily distinguished, and corresponding coordinates can be easily compared between the Lasso and

[2] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.lasso_path.html

[3] https://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_coordinate_descent_path.html#sphx-glr-auto-examples-linear-model-plot-lasso-coordinate-descent-path-py

gradient descent. How does the Lasso solution path compare with that of gradient descent? Do the two paths seem to converge to the same ultimate solution?

(f) **(Bonus)** In [1], Gunasekar and co-authors are more interested in the more general matrix completion case

$$\underset{\boldsymbol{U}}{\text{minimize}} \ \frac{1}{2} \sum_{(i,j)\in\Omega} ((\boldsymbol{U}\boldsymbol{U}^\top)_{ij} - \boldsymbol{Y}_{ij})^2, \tag{12}$$

where $\boldsymbol{Y} \in \mathbb{R}^{D\times D}$ is a symmetric and positive semi-definite low-rank matrix, Ω is a set of indices for the "observed entries", and $\boldsymbol{U} \in \mathbb{R}^{D\times D}$. The authors' main conjecture is that gradient descent with small step size and near zero initialization converges to the **minimum nuclear norm solution**. Design and carry out an experiment to test this claim.

References

[1] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159, 2017. 2, 3