Mathematical Foundations of Deep Neural Networks, M1407.001200
E. Ryu
Fall 2021

Homework 9
Due 5pm, Friday, December 10, 2021

Problem 1: *Projected gradient method.* Consider the optimization problem

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & x \in C, \end{array}$$

where $C \subset \mathbb{R}^n$. Constrained optimization problems of this type can be solved with the *projected gradient method*

$$x^{k+1} = \Pi_C(x^k - \alpha \nabla f(x^k)),$$

where Π_C is the projection onto C. The projection of $y \in \mathbb{R}^n$ onto $C \subseteq \mathbb{R}^n$ is defined as the point in C that is closest to y:

$$\Pi_C(y) = \underset{x\in C}{\text{argmin}}\, \|x-y\|^2.$$

For the particular set

$$C = \{x \in \mathbb{R}^2 \,|\, x_1 = a,\, 0 \le x_2 \le 1\},$$

where $a \in \mathbb{R}$, show that

$$\Pi_C(y) = \begin{bmatrix} a \\ \min\{\max\{y_2, 0\}, 1\} \end{bmatrix},$$

where $y = (y_1, y_2)$.

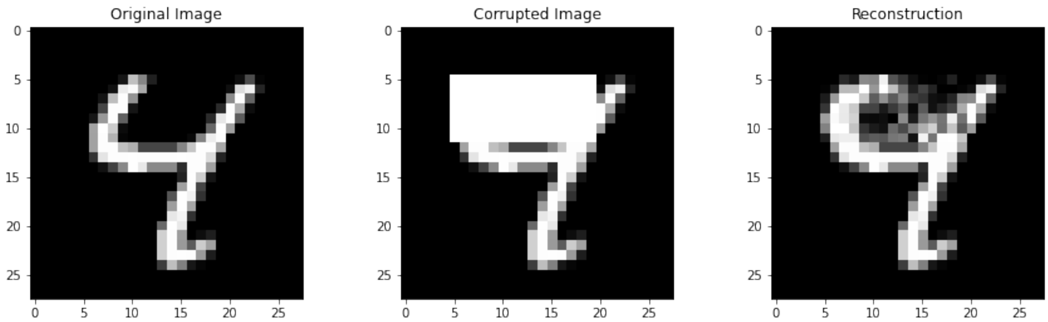

Figure 1: The original, corrupted, and inpainted MNIST image.

Problem 2: *Image inpainting with flow models.* Assume we have a trained flow model that we use to evaluate the likelihood function p. (Since we will not further train or update the flow model, we supress the network parameter θ and write p rather than p_θ.) The starter code `flow_inpainting.py` loads a NICE flow model pre-trained on the MNIST dataset saved in `nice.pt`. Let $X_{\text{true}} \in \mathbb{R}^{28\times 28}$ be an MNIST image with pixel intensities normalized to be in $[0, 1]$. Let $M = \{0, 1\}^{28\times 28}$ be a binary mask. We measure $M \odot X_{\text{true}}$, where $\odot$ denotes elementwise multiplication, and the goal is to inpaint the missing information $(1 - M) \odot X_{\text{true}}$, where $1 - M \in \{0, 1\}^{28\times 28}$ is the inverted mask. (See Figure 1.) Perform inpainting by solving the following constrained maximum likelihood estimation problem

$$\begin{array}{ll} \underset{X\in\mathbb{R}^{28\times 28}}{\text{minimize}} & -\log p(X) \\ \text{subject to} & M \odot X = M \odot X_{\text{true}} \\ & 0 \le X \le 1, \end{array}$$

where $0 \le X \le 1$ is enforced elementwise. Use the projected gradient method with learning rate 10^{-3} and 300 iterations.

Hint. Represent the optimization variable with

```
X = image.clone().requires_grad_(True)
```

while preserving `image`, the tensor containing the corrupted image. When manipulating `X` in the projection step, manipulate `X.data` rather than `X` itself so that the computation graph is not altered by the projection step. Use `clamp(...)` to enforce the $0 \le X \le 1$ constraint.

Remark. The optimization problem can be interpreted as finding the most likely reconstruction consistent with the measurements.

Remark. The NICE paper [2] obtains better inpainting results by using a learning rate scheduler (iteration-dependent stepsize) and adding noise to escape from local minima.

Problem 3: *VLB for IWAE.* The standard variational lower bound (VLB) of VAE is

$$\log(p_\theta(x)) \geq \mathrm{VLB}_{\theta,\phi}(x) = \mathbb{E}_{Z\sim q(z|x)}\left[\log\left(\frac{p_\theta(x \mid Z)p_Z(Z)}{q_\phi(Z \mid x)}\right)\right],$$

where $p_\theta(z \mid x)$ is the true posterior and $q_\phi(z \mid x)$ is the approximate posterior. Define

$$\mathrm{VLB}^{(K)}_{\theta,\phi}(x) = \mathbb{E}_{Z_1,\dots,Z_K\sim q_\phi(z|x)}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(x \mid Z_k)p_Z(Z_k)}{q_\phi(Z_k \mid x)}\right],$$

to be the VLB for importance weighted autoencoders (IWAE) [1]. To clarify, $Z_1,\dots,Z_K$ are sampled independently from $q_\phi(z \mid x)$. Note that IWAE with $K = 1$ coincides with the standard VAE, and $\mathrm{VLB}^{(1)}_{\theta,\phi} = \mathrm{VLB}_{\theta,\phi}$. Show:

(a) $\log p_\theta(x) \geq \mathrm{VLB}^{(K)}_{\theta,\phi}(x)$ for all x and $K \geq 1$.

(b) If $K \geq M$, then $\mathrm{VLB}^{(K)}_{\theta,\phi}(x) \geq \mathrm{VLB}^{(M)}_{\theta,\phi}(x)$ for all x.

(c) Let $X_1,\dots,X_N$ be data for training the IWAE. Show that if q_ϕ is "powerful enough", then

$$\underset{\theta\in\Theta}{\text{maximize}} \sum_{i=1}^{N}\log p_\theta(X_i) \;=\; \underset{\theta\in\Theta,\phi\in\Phi}{\text{maximize}} \sum_{i=1}^{N}\mathrm{VLB}^{(K)}_{\theta,\phi}(X_i).$$

What should be the precise meaning of "powerful enough"?

Hint. For (a), use the Jensen's inequality. For (b), let $I \subset \{1,\dots,K\}$ with $|I| = M$ be a uniformly distributed subset of distinct indices from $\{1,\dots,K\}$. Then, $\mathbb{E}_{I=\{i_1,\dots,i_M\}}\left[\frac{a_{i_1}+\cdots+a_{i_M}}{M}\right] = \frac{a_1+\cdots+a_K}{K}$ for any sequence of numbers $a_1,\dots,a_K$.

Remark. This analysis shows that $\mathrm{VLB}^{(K)}_{\theta,\phi}$ provides a tighter approximation of the log likelihood than $\mathrm{VLB}_{\theta,\phi}$. However, using $\mathrm{VLB}^{(K)}_{\theta,\phi}$ requires more computation than $\mathrm{VLB}_{\theta,\phi}$.

Problem 4: *Gradient ascent-descent for robust logistic regression.* Consider the minimax optimization problem

$$\underset{\theta\in\mathbb{R}^p}{\text{minimize}}\;\underset{\phi\in\mathbb{R}^p}{\text{maximize}}\; L(\theta,\phi),$$

where

$$L(\theta,\phi) = \frac{1}{N}\sum_{i=1}^{N}\log(1+\exp(-Y_i(X_i-\phi)^\intercal\theta)) - \frac{\lambda}{2}\|\phi\|^2,$$

$X_1,\dots,X_N \in \mathbb{R}^p$, $Y_1,\dots,Y_N \in \{-1,1\}$, and $\lambda = 30$. Use the data

```
N, p = 30, 20
np.random.seed(0)
X = np.random.randn(N,p)
Y = 2*np.random.randint(2, size = N) - 1
lamda = 30
```

where $X_1^\intercal,\dots,X_N^\intercal$ are the rows of `X`. Implement stochastic gradient ascent-descent with starting points θ^0 and ϕ^0 randomly initialized to be zero-mean IID Gaussians with standard deviation 0.1, descent and ascent stepsizes $\alpha = 3\times10^{-1}$ and $\beta = 10^{-4}$, and 1000 epochs. You may find the starter code `minimax_logistic.py` helpful.

Remark. We can interpret this problem as performing robust logistic regression where there is uncertaintly in the data $X_1, \ldots, X_N$.

Problem 5: *Rock paper scissors and minimiax optimization.* Consider a game of rock paper scissors between players A and B. Players A and B play randomized strategies with

$$p_A = \begin{bmatrix} \mathbb{P}(A \text{ plays rock}) \\ \mathbb{P}(A \text{ plays paper}) \\ \mathbb{P}(A \text{ plays scissors}) \end{bmatrix}, \qquad p_B = \begin{bmatrix} \mathbb{P}(B \text{ plays rock}) \\ \mathbb{P}(B \text{ plays paper}) \\ \mathbb{P}(B \text{ plays scissors}) \end{bmatrix}.$$

Define

$$\Delta^3 = \{p = (p_1, p_2, p_3) \in \mathbb{R}^3 \,|\, p_1, p_2, p_3 \geq 0,\, p_1 + p_2 + p_3 = 1\}$$

so that $p_A, p_B \in \Delta^3$. In the game, a player receives 1 point for a win, -1 points for a loss, and 0 points for a draw. Consider the minimax problem

$$\underset{p_A \in \Delta^3}{\text{minimize}} \quad \underset{p_B \in \Delta^3}{\text{maximize}} \quad \mathbb{E}_{p_A, p_B}[\text{points for } B].$$

(a) Show that

$$p_A^\star = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \qquad p_B^\star = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

is the unique solution to the minimax problem.

(b) Note that if $p_B = (1/3, 1/3, 1/3)$, then $\mathbb{E}_{p_A, p_B}[\text{points for } B] = 0$ regardless of how A plays. Does this mean any strategy $p_A \in \Delta^3$ is optimal for player A? (Here, the word "optimal" is used informally. Think about whether any $p_A \in \Delta^3$ is a best strategy for A.)

Clarification. We say $(\theta^\star, \phi^\star)$ is a solution to the minimax problem

$$\underset{\theta \in \Theta}{\text{minimize}} \quad \underset{\phi \in \Phi}{\text{maximize}} \quad L(\theta, \phi)$$

if $\theta^\star \in \Theta$, $\phi^\star \in \Phi$, and

$$L(\theta^\star, \phi) \leq L(\theta^\star, \phi^\star) \leq L(\theta, \phi^\star)$$

for all $\theta \in \Theta$ and $\phi \in \Phi$, i.e., unilaterally deviating from $\theta^\star$ increases the value of $L(\theta, \phi)$ and unilaterally deviating from $\phi^\star$ decreases the value of $L(\theta, \phi)$.

Remark. In the setup of GANs (which is what this problem is intended to prepare you for), if the generator is perfect, the discriminator cannot do better than a 50-50 guess in detecting fakes. However, the discriminator is still forced to learn to distinguish imperfect fakes, as otherwise, the generator can take advantage of the discriminator.

References

[1] Y. Burda, R. Grosse, and R. Salakhutdinov, Importance weighted autoencoders, *ICLR*, 2016.

[2] L. Dinh, D. Krueger, and Y. Bengio, NICE: Non-linear independent components estimation, *ICLR Workshop*, 2015.