Introduction to Deep Learning Theory

Exercises

Aleksandr Petiushko and Nesterova Maria

February 2025

Lecture 1

1.1 Exercise

Derive the condition of the steepest gradient descent in the form of $\nabla R(w^{(t+1)}) \cdot \nabla R(w^{(t)}) = 0$. Provide an example of the optimal learning rate for the quadratic R (like mean squared error).

1.2 Exercise

Prove. Consider multi-class case $|Y| > 2$. Then linear classifier:

$$a(x, w) = \arg\max_{c \in Y} g(x, w^c), \quad \text{where} \quad g(x, y) = \langle w, y \rangle. \tag{1}$$

1.3 Exercise

Question. What is the maximum value of Cross Entropy (CE)? Or KL-divergence?

1.4 Exercise

Prove. Cross Entropy can be expressed as

$$H(p, q) = H(p) + D_{KL}(p||q).$$

Lecture 2

2.5 Exercise

Consider the case of no bias term: $\arg\max_{i \in \{0,\dots,K-1\}} \langle W_i, h(x) \rangle$.

Prove proposition: $\forall s > 1$,

$$\max_{i \in \{0,\dots,K-1\}} \frac{\exp \langle W_i, sh(x) \rangle}{\sum_{j=0}^{K-1} \exp \langle W_j, sh(x) \rangle} \geq \max_{i \in \{0,\dots,K-1\}} \frac{\exp \langle W_i, h(x) \rangle}{\sum_{j=0}^{K-1} \exp \langle W_j, h(x) \rangle} \tag{2}$$

Hint: refer to the article.

2.6 Exercise

Prove proposition: for the Neural Collapse case, the lower bound on SoftMax loss with normalized projections and representations is

$$\log\left(1+(K-1)\exp\left(-\frac{K}{K-1}l^2\right)\right) \tag{3}$$

Hint: refer to the article.

2.7 Exercise

Assuming that $h'(x) = T*h(x)$, provide the reasoning and corresponding derivations behind the TPE loss:

$$L_{TPE} = -\frac{1}{m}\sum_{i=0}^{m-1}\log\frac{e^{\langle h'(x_i^a),h'(x_i^p)\rangle}}{e^{\langle h'(x_i^a),h'(x_i^p)\rangle}+e^{\langle h'(x_i^a),h'(x_i^n)\rangle}}$$

Hint: refer to the article.

2.8 Exercise

A-SoftMax. Prove: For multi-class case and usage of $\cos(m\theta)$ with normalization of both W and x, the lower bound is $m \geq 3$.

Note that you need to use the assumption of W_i uniform distribution.

Hint: refer to the article.

2.9 Exercise

CosFace (AM-softMax). Prove: Bounds for cosine margin $\cos(\theta) - m$: $0 \leq m \leq \frac{K}{K-1}$.

Hint: use $0 \leq m \leq 1 - \max_{i\neq j} W_i W_j$.

Hint 2: refer to the article.

Articles to consider

- V. Papyan et al. "Prevalence of neural collapse during the terminal phase of deep learning training." 2020
- X. Y. Han et al. "Neural collapse under mse loss: Proximity to and dynamics on the central path." 2021
- M. Wang et al. "Deep face recognition: A survey." 2018
- S. Sankaranarayanan et al. "Triplet probabilistic embedding for face verification and clustering." 2016.
- F. Wang et al. Normface: L2 hypersphere embedding for face verification. 2017
- W. Liu et al. "Sphereface: Deep hypersphere embedding for face recognition." 2017.
- H. Wang et al. "Cosface: Large margin cosine loss for deep face recognition." 2018.

Lecture 3

3.10 Exercise

Suppose that $\forall(x, y) \in P : x = 0, y \sim U(0, 1)$.
$\forall(x, y) \in Q : x = \theta, \theta \in [0, 1], y \sim U(0, 1)$.
When $\theta \neq 0$:

$$D_{KL}(P\|Q) = D_{KL}(Q\|P) = +\infty, D_{JS}(P\|Q) = \log 2, W_1(P, Q) = \theta.$$

When $\theta = 0$:

$$D_{KL}(P\|Q) = D_{KL}(Q\|P) = D_{JS}(P\|Q) = 0, W_1(P, Q) = \theta = 0.$$

Prove the following:

1. D_{KL} gives infinity on disjoint supports,
2. D_{JS} gives discontinuity at $\theta = 0$, and vanishing gradients outside,
3. W_1 gives a smooth value.

3.11 Exercise

Let us $x \sim p_{data}, y = G(z) \sim p_g, s = tx + (1 - t)y, t \in [0, 1]$. Provide the reasoning and corresponding derivations: why we use s instead of x for WGAN-GP loss:

$$\max_G \min_D \left(\mathbb{E}_{z\sim p_z}[D(G(z))] - \mathbb{E}_{x\sim p_{data}}[D(x)] + \lambda \mathbb{E}_{s\sim p_s}[(||\nabla_s D(s)|| - 1)^2]\right)$$

Hint: refer to this article.

Articles to consider

- Gulrajani I. et al. "Improved training of wasserstein gans". 2017.

Lecture 4

4.12 Exercise

Provide the algorithm and corresponding derivations behind *training* the Bayesian Neural Net by backpropagation.

Hint: refer to this article.

4.13 Exercise

Using $p_\theta(z) = N(0,1)$, $q_\phi(z|x) = N(\mu_\phi(x), \sigma^2_\phi(x))$, and $p_\theta(x|z) = N(g(z), c^2 I)$, derive the VAE loss in the following form:

$$L = \frac{1}{2}\sum_{j=1}^{K}\left(\mu_{j,\phi}(x)^2 + \sigma^2_{j,\phi}(x) - 1 - \log\sigma^2_{j,\phi}(x)\right) + E_{q_\phi(z|x)}\frac{(x-g(z))^2}{2c^2}$$

Hint: refer to this article.

Articles to consider

- Blundell, Charles, et al. "Weight uncertainty in neural network." 2015.
- Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." 2013.

Lecture 5

5.14 Exercise

Let us have N as the cardinality of the whole dataset and n as the mini-batch size. Provide the reasoning on why we have the normalizing factor $\frac{N}{n}$ for the SGLD update rule:

$$\theta^{(i+1)} = \theta^{(i)} + \eta(\nabla \log p(\theta^{(i)}) + \frac{N}{n}\sum_{j=1}^{n}\nabla \log p(x_{k_j}|\theta^{(i)})) + \sqrt{2\eta}\epsilon^{(i)}$$

Hint: refer to this article.

5.15 Exercise

Assume that you are given a Markov chain with state space Ω and transition matrix T, which is defined for all $x, y \in \Omega$ and $t \geq 0$ as $T(x,y) := P(X_{t+1} = y|X_t = x)$. Furthermore, let π be the stationary distribution of the chain.

Show that, if for some t the current state X_t is distributed according to the stationary distribution and additionally the chain satisfies the detailed balance equations

$$\pi(x)T(x,y) = \pi(y)T(y,x), \text{ for all } x, y \in \Omega,$$

then the following holds for all $k \geq 0$ and $x_0, \ldots, x_k \in \Omega$:

$$P(X_t = x_0, \ldots, X_{t+k} = x_k) = P(X_t = x_k, \ldots, X_{t+k} = x_0).$$

5.16 Exercise

Design a Gibbs sampler to simulate from a bivariate Normal distribution:

$$X = (X_1, X_2) \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right),$$

i.e. the pdf of the target distribution is

$$\pi(x_1, x_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{x_1^2 - 2\rho x_1 x_2 + x_2^2}{2(1-\rho^2)}\right\}.$$

.

Hint: you need to provide the conditional pdfs of $x_1|x_2$ and $x_2|x_1$

Articles to consider

- Welling, Max, and Yee W. Teh. "Bayesian learning via stochastic gradient Langevin dynamics." 2011

Lecture 6

6.17 Exercise

Lets work for Markovian case, where decoding each latent z_t is dependent only on the previous latent z_{t+1}. In case of Markovian Hierarchical VAE we can rewrite the ELBO:

$$\mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p(x, z_{1:T})}{q_\phi(z_{1:T}|x)}\right] = \mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p(z_T)p_\theta(x|z_1)\prod_{t=2}^{T} p_\theta(z_{t-1}|z_t)}{q_\phi(z_1|x)\prod_{t=2}^{T} q_\phi(z_t|z_{t-1})}\right]. \tag{4}$$

Prove it.

6.18 Exercise

Let's work with Variational Diffusion Models (VDM). The forward diffusion process adds noise by $q(x_t|x_{t-1})$. Let's use the notation for both the latents and data as $x_t: \ x_0 = x, \ x_{t,t>0} = z_t$.

The posterior distribution: $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$. The encoder has distribution $q(x_t|x_{t-1}) = N(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$.

Let us roll-out the equations for $q(x_t|x_0)$ based on linearity of Gaussians, where any $\epsilon_t \sim N(0, I)$.

$$
\begin{aligned}
x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} = \\
= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}\right) + \sqrt{1-\alpha_t}\epsilon_{t-1} = \\
= \sqrt{\alpha_t\alpha_{t-1}} x_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} = \\
= \sqrt{\alpha_t\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon_{t-2} = \cdots = \sqrt{\prod_{i=1}^{t}\alpha_i} x_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\epsilon_0. \quad (5)
\end{aligned}
$$

Prove the transition between lines 3 and 4 of equations above.

6.19 Exercise

The denoising step is described by $p_\theta(x_{t-1}|x_t)$. We can approximate it with some ground-truth distribution $q(x_{t-1}|x_t)$. If we add a conditioning on x_0, the $q(x_{t-1}|x_t, x_0)$ is obtained in analytical form.

Prove:

$$
\begin{gathered}
q\left(x_{t-1} \mid x_t, x_0\right) \propto N\left(\mu_q\left(x_t, x_0\right), \Sigma_q(t)\right), \\
\mu_q\left(x_t, x_0\right) = \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right) x_t + \sqrt{\bar{\alpha}_{t-1}}\left(1-\alpha_t\right) x_0}{1-\bar{\alpha}_t}, \\
\Sigma_q(t) = \frac{\left(1-\alpha_t\right)\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} I.
\end{gathered} \quad (6)
$$

6.20 Exercise

Prove: KL-divergence between two Gaussians for data of dimension d :

$$
\begin{gathered}
D_{KL}\left(N\left(\mu_q, \Sigma_q\right) || N\left(\mu_\theta, \Sigma_\theta\right)\right) = \\
= \frac{1}{2}\left[\log\frac{\left|\Sigma_\theta\right|}{\left|\Sigma_q\right|} - d + \operatorname{tr}\left(\Sigma_\theta^{-1}\Sigma_q\right) + \left(\mu_\theta - \mu_q\right)\Sigma_\theta^{-1}\left(\mu_\theta - \mu_q\right)\right].
\end{gathered} \quad (7)
$$

6.21 Exercise

Based on Tweede's Formula, we get $x_0 = \frac{x_t + (1-\bar{\alpha}_t)\nabla \log p(x_t)}{\sqrt{\bar{\alpha}_t}}$, and then substitute into

$$
\mu_q\left(x_t, x_0\right) = \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right) x_t + \sqrt{\bar{\alpha}_{t-1}}\left(1-\alpha_t\right) x_0}{1-\bar{\alpha}_t} = \frac{1}{\sqrt{\alpha_t}} x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla \log p\left(x_t\right) \quad (8)
$$

Prove it.

Hint: Tweede's formula refer to the article.

Articles to consider

- C. Luo. "Understanding diffusion models: A unified perspective", 2022.
- Efron, Bradley. "Tweedies formula and selection bias", 2011.

Lecture 7

7.22 Exercise

ℓ_p is a norm for $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. For $0 < p < 1$ the functional ℓ_p such as $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$ is not a norm. Prove it.

7.23 Exercise

DeepFool. Consider the linear binary classifier $sign(f(x))$, where $f(x) = w^T x + b$. x_0 is a data point, x_1 is the projection of the point x_0 onto the class-separating surface, and $r = x_1 - x_0$ is a perturbation.

Find r: minimize $\|r\|_2$ subject to constraint $w^T x_1 + b = 0$.

7.24 Exercise

FGSM. Consider the linear binary classifier $sign(f(x))$, where $f(x) = w^T x + b = \langle w, x \rangle + b$. $r = \langle w, \delta \rangle$ is a perturbation.

Find δ: maximize $\langle w, \delta \rangle$ subject to constraint $\|\delta\|_\infty \leq \epsilon,\ \epsilon > 0$.

Articles to consider

- Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. 2015.
- Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2014.

Lecture 8

Lecture 9

9.25 Exercise

Consider the linear binary classifier $f(x) = sign(w^T x + b)$.

Prove: if g is a smoothed version of f with any σ, then $f(x) = g(x)$.

9.26 Exercise

Consider the linear binary classifier $f(x) = sign(w^T x + b)$ and g is a smoothed version of f with any σ.

Prove: if $\underline{p_A} = p_A$ and $\overline{p_B} = p_B$, then the certification radius is

$$R = \frac{|w^T x + b|}{\|w\|}.$$

9.27 Exercise

Consider a simple neural network with a single hidden layer and ReLU activation functions. Let the output of the network be defined as:

$$f(x) = W_2 \cdot \text{ReLU}(W_1 x + b_1)$$

where W_1 and W_2 are weight matrices, and b_1 is a bias vector.

Find a Lipschitz constant for this neural network.

Hint: refer to this article.

9.28 Exercise

Suppose f classifies $N(x, \sigma^2)$ as c_A with probability $\geq p_A$. Suppose $p_A \in (0.5, 1]$ satisfies $P(f(x + \epsilon) = c_A) \geq p_A$.

New classifier is:

$$g(x) = \arg\max_{c \in \{0,1\}} P(f(x + \epsilon) = c),\ \epsilon \sim N(0, \sigma^2).$$

Prove: $g(x + \delta) = c_A$ for all $\|\delta\|_2 < \sigma \Phi^{-1}(p_A)$, where Φ is CDF of standard Gaussian.

$$\Phi(x) = \frac{1}{2\pi} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt.$$

Hint: refer to this article (Appendix).

Articles to consider

- Cohen, Jeremy, Elan Rosenfeld, and Zico Kolter. "Certified adversarial robustness via randomized smoothing." 2019. and Appendix.
- Huang, Yujia, Huan Zhang, Yuanyuan Shi, J. Zico Kolter, and Anima Anandkumar. "Training certifiably robust neural networks with efficient local lipschitz bounds." 2021.

Lecture 10

10.29 Exercise

Let functional $f : \mathbb{R}^d \to \mathbb{R}$. Smoothed version $g : \mathbb{R}^d \to \mathbb{R}$:

$$g(x) = \mathbb{E}_{\epsilon \sim N(0,\sigma^2)}[f(x+\epsilon)].$$

Φ^{-1} is the inverse of the standard Gaussian CDF:

$$\Phi(x) = \frac{1}{2\pi} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt.$$

For any bounded $f : \mathbb{R}^d \to [l, u]$, the map $\eta(x) = \sigma\Phi^{-1}\left(\frac{g(x)-l}{u-l}\right)$ is 1-Lipschitz, implying

$$l + (u-l)\Phi\left(\frac{\eta(x) - \|\delta\|_2}{\sigma}\right) \leq g(x+\delta) \leq l + (u-l)\Phi\left(\frac{\eta(x) + \|\delta\|_2}{\sigma}\right). \quad (9)$$

Prove it.

Hint: refer to this article.

10.30 Exercise

Fix $f : R_n \to [0, 1]$ and define g by:

$$g = E[f(x+\epsilon)],\ \epsilon \sim N(0, 1).$$

Find a Lipschitz constant L of g.

Hint: refer to this article.

Articles to consider

- Chiang, Ping-yeh, et al. "Detection as regression: Certified object detection with median smoothing." 2020.
- 1Salman, Hadi, et al. Provably robust deep learning via adversarially trained smoothed classifiers. 2019.