MIT 9.520/6.860: Statistical Learning Theory and Applications **Fall 2022**

Problem Set 3

Due Date: **Tue. Nov 15 2022, 10:59 am** *(Canvas submission)*

Instructions: There are **3 problems** in total in this problem set. The breakdown of individual scores per sub-problem are provided. Use the provided LaTeX template to typeset your report. Provide sufficient explanations in all solutions.

What to submit: Submit your report **online through Canvas** by the due date/time. Submission must be a single pdf in LaTeX format. Please report your final answers in **bold** font.

Policies: Collaborative reports are not allowed. Even if you discuss problems with classmates, you are expected to write and submit **individual reports**.

Problem 1 [30 points] In this problem we will study the training of deep ReLU networks for binary classification using the exponential loss. A deep network with K layers is a function $f : \mathbb{R}^d \to \mathbb{R}$, $f(W; \mathbf{x}) = W_K \sigma(W_{K-1} \dots W_2 \sigma(W_1 \mathbf{x}) \dots)$, where $\sigma(z) = \max(0, z)$ is the ReLU activation function that is applied coordinate-wise to its inputs. We will find it helpful to consider parameterizing the deep network weights W as $W_k = \rho_k V_k$ where $||V_k|| = 1$ for $k = 1, \dots, K$. Since ReLU is a positively-homogeneous activation function we can write $f(W; \mathbf{x}) = \rho f(V; \mathbf{x})$, with $\rho = \prod_{k=1}^{K} \rho_k$. The norm used in this problem is the Frobenius norm.

We will consider the exponential loss function in this problem. The loss incurred by a deep network $f(W; \cdot)$ on a training dataset $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is $L(W) = \sum_{i=1}^n e^{-y_i f(W; \mathbf{x}_i)}$. As in the lecture, we are interested in the evolution of the network parameters under *gradient flow*, which is $\dot{W}_k = \frac{\partial W_k}{\partial t} = -\frac{\partial L}{\partial W_k}$.

A structural property of ReLU networks that may be useful in analyzing gradient flow is:

$$\sum_{i,j} W_k^{ij} \frac{\partial f(W; \mathbf{x})}{\partial W_k^{ij}} = f(W; \mathbf{x})$$

Problem 1.1 [15 points] We would like to estimate the growth/decay rate of the norms of the weights under gradient flow. This is in general hard to do, but can be done if we assume that all the layers have the same norm ($||W_1|| = ||W_2|| = \dots = ||W_K||$). Show that if separability on $\mathcal{S}$ has been achieved then the norms of the weights keep growing, and that the growth rate is the same for all layers (ie, independent of k).

Hint: it might be easier to show that the evolution of the squared norm $\frac{\partial ||W_k||^2}{\partial t}$ *is independent*

of k.

Now let us consider a related version of this problem, the minimization of $L(V, \rho)$ under the constraint that $||V_k||^2 = 1$. The lagrangian for this problem is:

$$\mathcal{L} = \sum_{i=1}^{n} e^{-y_i \rho f(V;\mathbf{x}_i)} + \sum_{k=1}^{K} \lambda_k ||V_k||^2 \tag{3.1}$$

This gives us the gradient flow equations $\frac{\partial \rho_k}{\partial t} = -\frac{\partial \mathcal{L}}{\partial \rho_k}$, $\frac{\partial V_k}{\partial t} = -\frac{\partial \mathcal{L}}{\partial V_k}$.

Problem 1.2 [15 points] The *normalized margin* of the deep network $f(W;\cdot)$ for a (correctly classified) sample $(\mathbf{x}, y)$ is $yf(V;\mathbf{x})$, and tracks how far the sample is from the decision boundary. Assume that after a long time in the evolution of the gradient flow the training point (or group of points) with the smallest normalized margin dominates the loss, which means we can approximate the first term in (3.1) with just $e^{-y_1 \rho f(V;\mathbf{x}_1)}$ (we call $f(V;\mathbf{x}_1)$ the smallest margin). Show that after this point in time, the normalized margin of this sample $\nu_1 = y_1 f(V;\mathbf{x}_1)$ is non-decreasing (ie, $\frac{\partial \nu_1}{\partial t} \geq 0$).

Hint: As shown in class for the square loss, use the constraint $||V_k|| = 1$ to eliminate the Lagrange multiplier λ_k from the gradient flow equation $\frac{\partial V_k}{\partial t} = -\frac{\partial \mathcal{L}}{\partial V_k}$.

Problem 2 [40 points] In this problem we will explore the connection between deep networks and kernel machines. Recall that kernel functions induce a nonlinear feature mapping $\mathbf{\Phi}(\mathbf{x})$ to a high-dimensional space, with the kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y}) \rangle$ computing the inner product between two inputs in the induced feature space. We consider the kernels induced by repeated application of this induced feature mapping:

$$k^{(\ell)}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{\Phi}(\dots \mathbf{\Phi}(\mathbf{x}) \dots)), \mathbf{\Phi}(\mathbf{\Phi}(\dots \mathbf{\Phi}(\mathbf{y}) \dots)) \rangle$$

with the feature map repeated ℓ times.

Problem 2.1 [10 points] Consider the polynomial kernel $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^d$. Find the composition kernel $k^{(2)}(\mathbf{x}, \mathbf{y})$. How does it differ from the original polynomial kernel?

Problem 2.2 [10 points] Consider the RBF kernel $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x}-\mathbf{y}||^2}{\gamma^2}\right)$. Find the composition kernel $k^{(2)}(\mathbf{x}, \mathbf{y})$. Compare $k(\mathbf{x}, \mathbf{y})$ and $k^{(2)}(\mathbf{x}, \mathbf{y})$ in the limit where $||\mathbf{x} - \mathbf{y}||$ is large and small.

Now we will consider a parameteric feature map $\mathbf{\Phi}(\mathbf{x}) = \sigma(W\mathbf{x}), W \in \mathbb{R}^{h \times d}$, which is similar to a single layer of a deep network. Let us use a rectified polynomial nonlinearity: $\sigma(z) = \Theta(z) z^n$ (where $\Theta(z) = \mathbf{1}_{z>0}$). If we consider the kernel defined by the feature map $\mathbf{\Phi}$, we have:

$$k_n(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y}) \rangle = \sum_{i=1}^{h} \Theta(\langle \mathbf{w}_i, \mathbf{x} \rangle) \Theta(\langle \mathbf{w}_i, \mathbf{y} \rangle) \langle \mathbf{w}_i, \mathbf{x} \rangle^n \langle \mathbf{w}_i, \mathbf{y} \rangle^n$$

Let us assume that the rows of W are drawn from a normal distribution, ie $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, and we consider the inner product in the limit of $h \to \infty$. This means the kernel defined by the feature map is:

$$k_n(\mathbf{x}, \mathbf{y}) = \int d\mathbf{w} \frac{e^{-\frac{||\mathbf{w}||^2}{2}}}{(2\pi)^{d/2}} \Theta(\langle \mathbf{w}, \mathbf{x} \rangle) \Theta(\langle \mathbf{w}, \mathbf{y} \rangle) \langle \mathbf{w}, \mathbf{x} \rangle^n \langle \mathbf{w}, \mathbf{y} \rangle^n \tag{3.2}$$

This integral can be shown to take the following form:

$$k_n(\mathbf{x}, \mathbf{y}) = \frac{1}{\pi} ||\mathbf{x}||^n ||\mathbf{y}||^n J_n(\theta) \tag{3.3}$$

where $\theta = \cos^{-1}\left(\frac{\langle \mathbf{x}, \mathbf{y} \rangle}{||\mathbf{x}|| ||\mathbf{y}||}\right)$, and:

$$J_n(\theta) = (-1)^n (\sin\theta)^{2n+1} \left(\frac{1}{\sin\theta} \frac{\partial}{\partial \theta}\right)^n \left(\frac{\pi - \theta}{\sin\theta}\right)$$

Problem 2.3 [20 points] Evaluate the integral in (3.2) to compute the kernel defined by one layer of a deep network. We will focus on the special case where $n = 0$, and observe that the kernel for $n > 0$ can be derived from this case. You may choose to follow the following steps in your evaluation of the integral

1. Without loss of generality, we can assume $\mathbf{x}$ lies along the w_1 direction and that $\mathbf{y}$ lies in the w_1, w_2 plane. Integrate out the other orthogonal directions to obtain a double integral for $J_n(\theta)$ as defined in (3.3):

$$J_n(\theta) = \int \int dw_1 dw_2 e^{-\frac{w_1^2 + w_2^2}{2}} g(w_1, w_2, \theta)$$

2. Make the following change of variables $u = w_1$, $v = w_1 \cos\theta + w_2 \sin\theta$ to obtain an integral in u, v. This integral should be in the first quadrant of the u, v plane.

3. Make another variable substitution to adopt polar coordinates $u = \cos\left(\frac{\psi}{2} + \frac{\pi}{4}\right)$, $v = \sin\left(\frac{\psi}{2} + \frac{\pi}{4}\right)$, and then integrate out the radius coordinate r. As a check point - the r integral will evaluate to $n! \times \sin^{2n+1}\theta$, and the remaining integral in ψ is in the first quadrant.

4. Now plugin $n = 0$. In this case we can use the following formula (obtained through contour integration) to evaluate the integral

$$\int_0^\alpha \frac{d\psi}{a + b\cos\psi} = \frac{1}{\sqrt{a^2 - b^2}} \tan^{-1} \frac{\sin\alpha \sqrt{a^2 - b^2}}{b + a\cos\alpha}$$

The result of the integral is $J_0(\theta) = \pi - \theta$

Problem 3 [30 points] In this problem we will provide a simplified derivation of the Neural Collapse phenomenon. To recall, Neural Collapse occurs on a deep network $f(\mathbf{x}) = \mathbf{W}\mathbf{h}(\mathbf{x}) + \mathbf{b}$ in the Terminal Phase of Training. Before we recall the conditions of Neural Collapse let us define the first and second order statistics of the last layer features:

- Mean class features: $\mu_c = \frac{1}{C}\sum_{i=1}^{N} \mathbf{h}_{i,c}$
- Within class covariance: $\Sigma_W = \frac{1}{NC}\sum_{c=1}^{C}\sum_{i=1}^{N}(\mathbf{h}_{i,c} - \mu_c)(\mathbf{h}_{i,c} - \mu_c)^\top$
- Between class covariance: $\Sigma_B = \frac{1}{C}\sum_{c=1}^{C}(\mu_c - \mu_G)(\mu_c - \mu_G)^\top$
- Total covariance: $\Sigma_T = \frac{1}{NC}\sum_{c=1}^{C}\sum_{i=1}^{N}(\mathbf{h}_{i,c} - \mu_G)(\mathbf{h}_{i,c} - \mu_G)^\top$

Neural Collapse (in an idealized scenario) is defined by the following 4 conditions:

- **NC1 (Variability collapse)** $\Sigma_W = 0$
- **NC2 (Convergence to Simplex ETF)** $\|\mu_c - \mu_G\|_2 = \|\mu_{c'} - \mu_G\|_2 \quad \forall c, c'$. Moreover, if we define $\tilde{\mu}_c = \frac{\mu_c - \mu_G}{\|\mu_c - \mu_G\|_2}$, then we have $\langle \tilde{\mu}_c, \tilde{\mu}_{c'} \rangle = \frac{C}{C-1}\delta_{c,c'} - \frac{1}{C-1}$.
- **NC3 (Self-Duality)** Collect the centered class means into a matrix $\mathbf{M} = [\mu_c - \mu_G]$, we have $\frac{\mathbf{W}^\top}{\|\mathbf{W}\|_F} = \frac{\mathbf{M}}{\|\mathbf{M}\|_F}$
- **NC4 (Nearest Center Classification)** $\text{argmax}_c \langle \mathbf{w}_c, \mathbf{h}(\mathbf{x}) \rangle + b_c = \text{argmin}_c \|\mathbf{h}(\mathbf{x}) - \mu_c\|_2$

We will now show how these conditions arise from the minimization of the squared error loss. Due to a result from Webb and Lowe (1990), we can write the solution of least squares as:

$$\mathbf{W} = \frac{1}{C}\mathbf{M}\Sigma_T^\dagger$$
$$\mathbf{b} = \frac{1}{C}\mathbf{1}_C - \frac{1}{C}\mathbf{M}\Sigma_T^\dagger\mu_G$$

Using the above solution, show that NC1+NC2 $\implies$ NC3+NC4