Math 5490 – Homework 6: Due May 10 by 11:59pm

Instructions:

- Complete the problems below, and submit your solutions and Python code by uploading them to the Google form: `https://forms.gle/B3EfYpSU8jQPYhiS9`
- Submit all your Python code in a single .py file using the function templates given in each problem. I will import your functions from this file and test your code.
- If you use LaTeX to write up your solutions, upload them as a pdf file. Students who use LaTeX to write up their solutions will receive bonus points on the homework assignment (equivalent to 1/3 of a letter grade bump).
- If you choose to handwrite your solutions and scan them, please either use a real scanner, or use a smartphone app that allows scanning with you smartphone camera. It is not acceptable to submit photos of your solutions, as these can be hard to read.

Problems:

1. Complete the following exercises.

   (i) Let $g(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, where $\mathbf{x} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$. Show that the Jacobian matrix of g is $Jg(\mathbf{x}) = A$.

   (ii) Let $h(\mathbf{x}) = \sigma(\mathbf{x})$, where $\sigma : \mathbb{R} \to \mathbb{R}$ acts componentwise on $\mathbf{x}$, so that $\sigma(\mathbf{x})_i = \sigma(x_i)$. Show that $Jh(\mathbf{x})$ is the diagonal matrix $D = \operatorname{diag} \sigma'(\mathbf{x})$.

   (iii) Use the first two parts and the chain rule to show that the Jacobian of $g(\mathbf{x}) = A\sigma(B\mathbf{x} + \mathbf{b})$ is $Jg(\mathbf{x}) = ADB$, where $D = \operatorname{diag} \sigma'(B\mathbf{x} + \mathbf{b})$.

2. The traditional neural network architecture

$$\mathbf{F}_k = \sigma_k(W_k\mathbf{F}_{k-1} + \mathbf{b}_k), \quad k = 1, \ldots, L,$$

often yields worse performance for deeper networks with more layers compared to shallower networks. The main issue it that training is difficult, due to vanishing gradients or gradient blowup (where the gradients either become very small and training does not progress, or become very large and training is unstable). To understand why, consider the case where $\sigma_k(t) = t$ is the identity and the biases $\mathbf{b}_k = 0$ all vanish. Then

$$\mathbf{F}_L(\mathbf{x}) = W_L W_{L-1} \cdots W_2 W_1 \mathbf{x}.$$

The L-fold product is very sensitive to the spectral norms of the matrices; when the eigenvalues are larger than one in magnitude it blows up exponentially with L, while when they are less than one it will decay exponentially.

The *Residual Neural Network (ResNet)* architecture solves this problem by changing the architecture to

$$\mathbf{F}_k = \mathbf{F}_{k-1} + V_k\sigma_k(\mathbf{z}_k), \quad \text{where} \quad \mathbf{z}_k = W_k\mathbf{F}_{k-1} + \mathbf{b}_k, \quad k = 1, \ldots, L. \tag{1}$$

The idea is to have each layer learn the *residual* $\mathbf{F}_k - \mathbf{F}_{k-1}$, which allows the network to easily skip layers, by setting $\mathbf{F}_k = \mathbf{F}_{k-1}$. Thus, a deeper network with ResNet architecture should not perform worse than a shallower network. Note that in ResNet the dimensions of each layer must be the same.

Follow an argument similar to that in the proof of Theorem 10.1 to establish the ResNet back propagation equations

$$\begin{aligned}\nabla_{\mathbf{F}_{k-1}}\mathcal{L} &= \nabla_{\mathbf{F}_k}\mathcal{L} + W_k^T D_k V_k^T \nabla_{\mathbf{F}_k}\mathcal{L}\\ \nabla_{V_k}\mathcal{L} &= (\nabla_{\mathbf{F}_k}\mathcal{L})\,\sigma(\mathbf{z}_k)^T\\ \nabla_{\mathbf{z}_k}\mathcal{L} &= D_k V_k^T \nabla_{\mathbf{F}_k}\mathcal{L}\end{aligned}$$

and as in Theorem 10.1 $\nabla_{W_k}\mathcal{L} = (\nabla_{\mathbf{z}_k}\mathcal{L})\,\mathbf{F}_{k-1}^T$ and $\nabla_{\mathbf{b}_k}\mathcal{L} = \nabla_{\mathbf{z}_k}\mathcal{L}$. Hint: For the first identity, let $h_k : \mathbb{R}^n \to \mathbb{R}$ denote the loss $\mathcal{L}$ as a function $\mathbf{F}_k$, so that $h_k(\mathbf{F}_k) = \mathcal{L}$ and $\nabla h_k(\mathbf{F}_k) = \nabla_{\mathbf{F}_k}\mathcal{L}$. Then show that

$$h_{k-1}(\mathbf{F}_{k-1}) = h_k(\mathbf{F}_{k-1} + V_k\sigma_k(W_k\mathbf{F}_{k-1} + \mathbf{b}_k)),$$

and use the chain rule.