EECS 182 Deep Neural Networks
Fall 2025 Anant Sahai and Gireeja Ranade

Homework 0

This homework is due on Fri, Sep 5, 2025, at 10:59PM. A failure to complete this homework will be deemed "insufficient engagement" and will be used to make room in the class for students who are actually committed to doing the work. The non-programming problems must be done by hand and scanned in and the programming problem should have a pdf of the completed notebook attached. Combine your scan and the pdf into a single large pdf for submission.

1. Reflection on your learning goals at the start of the semester

Deep learning is a particularly challenging subject *culturally* given the state of our understanding as well as the rapid advancements and economic/cultural/intellectual impacts of this emergent technology. You've come into this class with your own individual background. Think about it and do a little online exploration and concisely write up your response.

(a) **Before doing any further reading or exploration and just based on what you know, please briefly describe what you think your learning goals are.**

(b) Open up any modern top-tier LLM-based chat system — you can use the Berkeley promotion of Perplexity for example, or just use OpenAI's ChatGPT, Anthropic's Claude, Google's Gemini, etc. Interact with the system to have a conversation about Deep Learning. **After this interaction, please describe how — if at all — this has modified your learning goals.**

(c) Now, the approach taken in this course is a reasonably intellectually conservative point of view that tries to approach Deep Learning through a lens of understanding that leverages mathematical intuition and models, as well as connections to the larger Machine Learning tradition, to the extent possible. However, it is important to understand that this is not the only possible perspective. At least skim through David Donoho's 2024 Paper "Data Science at the Singularity" available at https://doi.org/10.1162/99608f92.b91339ef. **After reading that paper, how have you thought about the role of this class in your learning?**

(d) **Finally, comment briefly about how your own understanding of deep learning can benefit from activities that complement what we are doing in this course? What kind of guidance from your peers and course staff do you think would be helpful?**

2. Vector Calculus Review

Let $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. For the following parts, before taking any derivatives, identify what the derivative looks like (is it a scalar, vector, or matrix?) and how we calculate each term in the derivative. Then carefully solve for an arbitrary entry of the derivative, then stack/arrange all of them to get the final result. Note that the convention we will use going forward is that vector derivatives of a scalar (with respect to a column vector) are expressed as a row vector, i.e. $\frac{\partial f}{\partial \mathbf{x}} = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n}]$ since a row acting on a column gives a scalar. You may have seen alternative conventions before, but the important thing is that you need to understand the types of objects and how they map to the shapes of the multidimensional arrays we use to represent those types.

(a) Show $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T\mathbf{c}) = \mathbf{c}^T$

(b) Show $\frac{\partial}{\partial \mathbf{x}}||\mathbf{x}||_2^2 = 2\mathbf{x}^T$

(c) Show $\frac{\partial}{\partial \mathbf{x}}(A\mathbf{x}) = A$

(d) Show $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A\mathbf{x}) = \mathbf{x}^T(A + A^T)$

(e) Under what condition is the previous derivative equal to $2\mathbf{x}^T A$?

3. Least Squares and the Min-norm problem from the Perspective of SVD

Consider the equation $X\mathbf{w} = \mathbf{y}$, where $X \in \mathbb{R}^{m\times n}$ is a non-square data matrix, w is a weight vector, and y is vector of labels corresponding to the datapoints in each row of X.

Let's say that $X = U\Sigma V^T$ is the (full) SVD of X. Here, U and V are orthonormal square matrices, and Σ is an $m \times n$ matrix with non-zero singular values (σ_i) on the "diagonal".

For this problem, we define $\Sigma^\dagger$ an $n \times m$ matrix with the reciprocals of the singular values ($\frac{1}{\sigma_i}$) along the "diagonal".

(a) First, consider the case where $m > n$, i.e. our data matrix X has more rows than columns (tall matrix) and the system is overdetermined. **How do we find the weights w that minimizes the error between $X\mathbf{w}$ and y?** In other words, we want to solve $\min_{\mathbf{w}} \|X\mathbf{w} - \mathbf{y}\|^2$.

(b) **Plug in the SVD $X = U\Sigma V^T$ and simplify.** Be careful with dimensions!

(c) You'll notice that the least-squares solution is in the form $\mathbf{w}^* = A\mathbf{y}$. **What happens if we left-multiply X by our matrix A?** This is why the matrix A of the least-squares solution is called the left-inverse.

(d) Now, let's consider the case where $m < n$, i.e. the data matrix X has more columns than rows and the system is underdetermined. There exist infinitely many solutions for w, but we seek the minimum-norm solution, ie. we want to solve $\min \|\mathbf{w}\|^2 \text{s.t.} X\mathbf{w} = \mathbf{y}$. **What is the minimum norm solution?**

(e) **Plug in the SVD $X = U\Sigma V^T$ and simplify.** Be careful with dimensions!

(f) You'll notice that the min-norm solution is in the form $\mathbf{w}^* = B\mathbf{y}$. **What happens if we right-multiply X by our matrix B?** This is why the matrix B of the min-norm solution is called the right-inverse.

4. The 5 Interpretations of Ridge Regression

(a) *Perspective 1: Optimization Problem.* Ridge regression can be understood as the unconstrained optimization problem

$$\underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{y} - X\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2, \tag{1}$$

where $X \in \mathbb{R}^{n\times d}$ is a data matrix, and $\mathbf{y} \in \mathbb{R}^n$ is the target vector of measurement values. What's new compared to the simple OLS problem is the addition of the $\lambda\|\mathbf{w}\|^2$ term, which can be interpreted as a "penalty" on the weights being too big.

Use vector calculus to expand the objective and solve this optimization problem for w.

(b) *Perspective 2: "Hack" of shifting the Singular Values.* In the previous part, you should have found the optimal $\mathbf{w}$ is given by

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$$

(If you didn't get this, you should check your work for the previous part).

Let $X = U\Sigma V^T$ be the (full) SVD of the X. Recall that U and V are square orthonormal (norm-preserving) matrices, and Σ is a $n \times d$ matrix with singular values σ_i along the "diagonal". **Plug this into the Ridge Regression solution and simplify. What happens to the singular values of $(X^TX + \lambda I)^{-1}X^T$ when $\sigma_i << \lambda$? What about when $\sigma_i >> \lambda$?**

(c) *Perspective 3: Maximum A Posteriori (MAP) estimation.* Ridge Regression can be viewed as finding the MAP estimate when we apply a prior on the (now viewed as random parameters) $\mathbf{W}$. In particular, we can think of the prior for $\mathbf{W}$ as being $\mathcal{N}(\mathbf{0}, I)$ and view the random Y as being generated using $Y = \mathbf{x}^T\mathbf{W} + \sqrt{\lambda}N$ where the noise N is distributed iid (across training samples) as $\mathcal{N}(0, 1)$. At the vector level, we have $\mathbf{Y} = X\mathbf{W} + \sqrt{\lambda}\mathbf{N}$. Note that the X matrix whose rows are the n different training points are not random.

Show that (1) is the MAP estimate for $\mathbf{W}$ given an observation $\mathbf{Y} = \mathbf{y}$.

(d) *Perspective 4: Fake Data.* Another way to interpret "ridge regression" is as the ordinary least squares for an augmented data set — i.e. adding a bunch of fake data points to our data. Consider the following augmented measurement vector $\hat{\mathbf{y}}$ and data matrix $\hat{\mathbf{X}}$:

$$\hat{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_d \end{bmatrix} \quad \hat{\mathbf{X}} = \begin{bmatrix} X \\ \sqrt{\lambda}\mathbf{I}_d \end{bmatrix},$$

where $\mathbf{0}_d$ is the zero vector in $\mathbb{R}^d$ and $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix. **Show that the classical OLS optimization problem $\operatorname{argmin}_{\mathbf{w}} \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\mathbf{w}\|_2^2$ has the same minimizer as (1).**

(e) *Perspective 5: Fake Features.* For this last interpretation, let's instead construct an augmented design matrix in the following way:

$$\check{\mathbf{X}} = [X \ \sqrt{\lambda}\mathbf{I_n}]$$

i.e. we stack X with $\sqrt{\lambda}\mathbf{I}_n$ horizontally. Now our problem is underdetermined: the new dimension $d + n$ is larger than the number of points n. Therefore, there are infinitely many values $\boldsymbol{\eta} \in \mathbb{R}^{d+n}$ for which $\check{\mathbf{X}}\boldsymbol{\eta} = \mathbf{y}$. We are interested in the **min-norm** solution, ie. the solution to

$$\operatorname*{argmin}_{\boldsymbol{\eta}} \|\boldsymbol{\eta}\|_2^2 \text{ s.t. } \check{\mathbf{X}}\boldsymbol{\eta} = \mathbf{y}. \tag{2}$$

Show that this is yet another form of ridge regression and that the first d coordinates of $\boldsymbol{\eta}^*$ form the minimizer of (1).

(f) We know that the Moore-Penrose pseudo-inverse for an underdetermined system (wide matrix) is given by $A^\dagger = A^T(AA^T)^{-1}$, which corresponds to the min-norm solution for $A\boldsymbol{\eta} = \mathbf{z}$. That is, the optimization problem

$$\operatorname{argmin} \|\boldsymbol{\eta}\|^2 \text{s.t.} A\boldsymbol{\eta} = \mathbf{z}$$

is solved by $\boldsymbol{\eta} = A^\dagger\mathbf{z}$. Let $\hat{\mathbf{w}}$ be the minimizer of (1).

Use the pseudo-inverse to show that solving to the optimization problem in (2) yields

$$\hat{\mathbf{w}} = X^T(XX^T + \lambda\mathbf{I})^{-1}\mathbf{y}$$

Then, show that this is equivalent to the standard formula for Ridge Regression

$$\hat{\mathbf{w}} = (X^TX + \lambda\mathbf{I})^{-1}X^T\mathbf{y}$$

Hint: It may be helpful to review Kernel Ridge Form.

(g) We know that the solution to ridge regression (1) is given by $\hat{\mathbf{w}}_\mathbf{r} = (X^\top X + \lambda \mathbf{I})^{-1} X^\top \mathbf{y}$. **What happens when $\lambda \to \infty$?** It is for this reason that sometimes ridge regularization is referred to as "shrinkage."

(h) **What happens to the solution of ridge regression when you take the limit $\lambda \to 0$?** Consider both the cases when X is wide (underdetermined system) and X is tall (overdetermined system).

5. ReLU Elbow Update under SGD

In this question we will explore the behavior of the ReLU nonlinearity with Stochastic Gradient Descent (SGD) updates. The hope is that this problem should help you build a more intuitive understanding for how SGD works and how it iteratively adjusts the learned function.

We want to model a 1D function $y = f(x)$ using a 1-hidden layer network with ReLU activations and no biases in the linear output layer. Mathematically, our network is

$$\hat{f}(x) = \mathbf{W}^{(2)} \Phi \left(\mathbf{W}^{(1)} x + \mathbf{b} \right)$$

where $x, y \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^d$, $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times 1}$, and $\mathbf{W}^{(2)} \in \mathbb{R}^{1 \times d}$. We define our loss function to be the squared error,

$$\ell \left(x, y, \mathbf{W}^{(1)}, \mathbf{b}, \mathbf{W}^{(2)} \right) = \frac{1}{2} \left\| \hat{f}(x) - y \right\|_2^2.$$

For the purposes of this problem, we define the gradient of a ReLU at 0 to be 0.

(a) Let's start by examining the behavior of a single ReLU with a linear function of x as the input,

$$\phi(x) = \begin{cases} wx + b, & wx + b > 0 \\ 0, & \text{else} \end{cases}.$$

Notice that the slope of $\phi(x)$ is w in the non-zero domain.

We define a loss function $\ell(x, y, \phi) = \frac{1}{2} \|\phi(x) - y\|_2^2$. **Find the following:**

(i) The location of the 'elbow' e of the function, where it transitions from 0 to something else.

(ii) The derivative of the loss w.r.t. $\phi(x)$, namely $\frac{d\ell}{d\phi}$

(iii) The partial derivative of the loss w.r.t. w, namely $\frac{\partial \ell}{\partial w}$

(iv) The partial derivative of the loss w.r.t. b, namely $\frac{\partial \ell}{\partial b}$

(b) Now suppose we have some training point (x, y) such that $\phi(x) - y = 1$. In other words, the prediction $\phi(x)$ is 1 unit above the target y — we are too high and are trying to pull the function downward.

Describe what happpens to the slope and elbow of $\phi(x)$ when we perform gradient descent in the following cases:

(i) $\phi(x) = 0$**.**

(ii) $w > 0$**,** $x > 0$**, and** $\phi(x) > 0$**. It is fine to check the behavior of the elbow numerically in this case.**

(iii) $w > 0$**,** $x < 0$**, and** $\phi(x) > 0$**.**

(iv) $w < 0$**,** $x > 0$**, and** $\phi(x) > 0$**. It is fine to check the behavior of the elbow numerically in this case.**

Additionally, draw and label $\phi(x)$, the elbow, and the qualitative changes to the slope and elbow after a gradient update to w and b. You should label the elbow location and a candidate (x, y) pair. Remember that the update for some parameter vector $\mathbf{p}$ and loss ℓ under SGD is

$$\mathbf{p}' = \mathbf{p} - \lambda \nabla_{\mathbf{p}}(\ell),\ \lambda > 0.$$

(c) Now we return to the full network function $\hat{f}(x)$. **Derive the location e_i of the elbow of the i'th elementwise ReLU activation.**

(d) **Derive the new elbow location e_i' of the i'th elementwise ReLU activation after one stochastic gradient update with learning rate λ.**

6. Coding Fully Connected Networks

In this coding assignment, you will be building a fully-connected neural network from scratch using NumPy.

Download the .zip file with the starter code and get it to work in either Google Colab or a local Conda environment.

Please submit the `.pdf` export of only the jupyter notebook when it is completed as a part of your submission. In addition, please answer the following question:

(a) **Did you notice anything about the comparative difficulty of training the three-layer net vs training the five layer net?**

7. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

(c) **Roughly how many total hours did you work on this homework?**

Contributors:

- Saagar Sanghavi.
- Alexander Tsigler.
- Anant Sahai.
- Jane Yu.
- Philipp Moritz.
- Soroush Nasiriany.
- Josh Sanz.
- Linyuan Gong.
- Luke Jaffe.

