Assignment 1 - Intro to DL and NN

CS5388 - Dr. Franklin (Spring 2025)

Part 1: Theoretical Questions

Problem 0 (OPTIONAL FIRST PROBLEM)

Let $\sigma(x)$ is the sigmoid function given by

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Show that

$$\frac{d\sigma}{dx} = \sigma(x) \cdot \big(1 - \sigma(x)\big)$$

NOTE: This problem is not required but counts as extra credit. However, you may use the result in the following problem.

Problem 1- Gradients

Consider the following function $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$:

$$f(\mathbf{x}) = \sigma\left(\log\left(5\left(\max\{x_1, x_2\} \cdot \frac{x_3}{x_4} - (x_5 + x_6)\right)\right) + \frac{1}{2}\right)$$

where σ is the sigmoid function (see above).

Compute the gradient $\nabla_x f(\cdot)$ and evaluate it at $\hat{\mathbf{x}} = (-1, 3, 4, 5, -5, 7)$.

Problem 2 - Proof Practice

(a) Prove that

$$\ln x \leq x - 1, \qquad \forall\, x > 0$$

with equality if and only if $x = 1$

Hint: Consider the derivative of $\ln(x) - (x - 1)$ and think about concavity /convexity and second derivatives.

(b) Consider two discrete probability distributions p and q over k outcomes:

$$\sum_{i=1}^{k} p_i = \sum_{i=1}^{k} q_i = 1$$
$$p_i > 0, q_i > 0, \qquad \forall\, i \in \{1, \dots, k\}$$

The Kullback-Leibler (KL) divergence (also known as relative entropy) between these distributions is given by:

$$KL(p, q) = \sum_{i=1}^{k} p_i \log\left(\frac{p_i}{q_i}\right)$$

It is common to refer to $KL(p, q)$ as a measure of distance (even though it is not a proper metric). Many algorithms in machine learning are based on minimizing KL divergence between two probability distributions. In this question, we will show why this might be a sensible thing to do.

Hint: This question does not require you to know anything more than the definition of $KL(p, q)$ and the identity above.

(i) Using the identity above, show that $KL(p, q)$ is always non-negative.

(ii) When is $KL(p, q) = 0$?

(iii) Provide a counterexample to show that KL divergence is not a symmetric function of its arguments: $KL(p, q) \neq KL(q, p)$

Problem 3 - Calculating Loss for Classification Problems

You are training a multiclass classifier to distinguish between images of different vehicles, {"motorcycle", "car", "truck", "SUV"}. For an image in your training set, you pass it forward through your neural network and compute the following scores for an image containing a motorcycle:

motorcycle score = 2.0
car score = 4.5
truck score = 1.0
bicycle score = 3.5

a) Calculate the Multiclass SVM Loss (hinge loss) for this training example

b) Convert the scores into probabilities using the soft-max function and then calculate the cross-entropy loss for the training example.

Problem 4 - Calculating Loss for Regression Problems

You are training a linear regression model with two features (x_1, x_2) using the model:

$$y = w_1x_1 + w_2x_2 + b$$

You are given the following three training examples:

$$x_1 = 1, x_2 = 2, y = 4$$
$$x_1 = 2, x_2 = 0, y = 3$$
$$x_1 = 3, x_2 = 1, y = 5$$

Your current model parameters are $w_1 = 0.5, w_2 = 1.0, b = 0$

(a) Calculate the L2 loss for each example, then calculate the average L2 loss across all three examples.

(b) Calculate the L2 regularization term using the formula

$$R(w_1, w_2, b) = \lambda \cdot (w_1^2 + w_2^2)$$

where $\lambda = 0.1$ is the regularization strength.

(c) Calculate the total loss which is the sum of the average L2 loss and the L2 regularization term.

Problem 5 - Perceptrons

(a) Design a two-input perceptron (e.g., a single neuron) that implements the Boolean function $A \wedge \neg B$

(b) Design a two layer network of perceptrons that implements $A \oplus B$ (where $\oplus$ is XOR)

Problem 6 - Computational Graph Problem

In practice, writing the closed-form expression of the derivative of a loss function $\mathbf{f}$ with respect to the parameters of a deep neural network is often complex and unnecessary. Instead, we define computation graphs and use automatic differentiation algorithms (like backpropagation) to compute gradients efficiently using the chain rule.

Consider the following vector function $\mathbf{f}(\mathbf{w})$ where $\mathbf{w} = (w_1, w_2)$ and

$$f_1(w_1, w_2) = \ln(w_1^2 + w_2^2) + w_1\, e^{w_2}$$
$$f_2(w_1, w_2) = w_1 \sin w_2 + \cos\ w_1$$

(a) Draw the computation graph for the forward pass through f and label the intermediate variables. Let the nodes in your graph represent the simple functions:

$$\{+,\ \square^2,\ \ln,\ \ *\ ,\ \sin,\ \ \cos\ )$$

(b) Compute the value of $\mathbf{f}$ for the inputs $(2, 1)$. Show the values of all the intermediate variables on the computation graph.

(c) Compute the Jacobian matrix of $\mathbf{f}$ with respect to w_1 and w_2 using automatic differentiation and give its value at (2, 1).

(d) Confirm your results in part (c) by performing manual differentiation to compute the Jacobian at (2, 1)

(e) Estimate the Jacobian matrix at the point (2,1) using a forward difference approximation with $\Delta w_1 = \Delta w_2 = 0.001$

(f) Briefly explain the advantages of using automatic differentiation compared to manual differentiation or numerical differentiation, especially in the context of neural networks.

Problem 7 - Gradient Descent

You are training a linear regression model with n features and provide the training data: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$ where $\mathbf{x}_i$ is an n-dimensional real-valued vector and y_i is a real number for each i.

Derive the gradient descent training rule for a single neuron with output f where

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_1 x_1^2 + w_2 x_2 + w_2 x_2^2 + \cdots + w_n x_n + w_n x_n^2$$

For the training rule, use the error function,

$$L = \frac{1}{2} \sum_{i=1}^{m} (y_i - f(\mathbf{x}_i))^2$$

Part 2: Paper Review

The purpose of this section of each homework assignment is to connect you with papers that propose intriguing new ideas related to the topics of our class. I found it very useful in my own graduate-level Deep Learning courses, and I hope you will, too.

You must choose **one** of the papers and complete the following:

1. Write a brief review of the paper,
2. Answer paper-specific question,

Guidelines: The paper review is limited to 350 words and the answer to the paper-specific question is limited to 350 words.

The review should cover the following:

- Summarize the main contribution of the paper in terms of its key insights. What are the strengths and weaknesses?
- Respond with your personal takeaway from this paper. What is noteworthy to you about this paper? Is there a future direction of research in an area the authors haven't addressed?

Paper Choice 1:

Adam Gaier and David Ha’s spotlight presentation at NeurIPS 2019 on “Weight Agnostic Neural Networks” challenges conventional notions about neural networks. Through experiments, the paper re-examines the comparative importance of architectures and weights in predicting performance. The paper can be viewed here. The authors have also written a blog post with intuitive visualizations to help understand its key concepts better.

Question: One of the fundamental aspects of deep learning is that, given a parameterized function, we can determine weights to represent any function if it possesses sufficient depth and complexity. This paper delves into the representational capabilities of architectures when a fixed method is employed for determining weights. Does the method for weight determination play a significant role in determining the representational power of architectures? In your opinion, do you believe these two architectures possess equal representational power? Provide reasons to support your answer.

Paper Choice 2:

The second paper is again one that questions conventional wisdom and shows that large neural networks can actually fit random labels, that is labels that remain fixed but, for example, have no relationship to what is actually in the image. The paper title is “Understanding deep learning requires rethinking generalization” and can be found here.

Question: If neural networks can “memorize” the data, which is the only thing they can do for random label assignments that don’t correlate with patterns in the data, why do you think neural networks learn more meaningful, generalizable representations when there are meaningful patterns in the data?

Part 3: Coding Portion

coming soon…