CARNEGIE MELLON UNIVERSITY 07-280
HOMEWORK 6

DUE: Thursday, Feb. 26, 2026

https://www.cs.cmu.edu/~07280

INSTRUCTIONS

- **Format:** Use the provided LaTeX template to write your answers in the appropriate locations within the *.tex files and then compile a pdf for submission. We try to mark these areas with STUDENT SOLUTION HERE comments. Make sure that you don't change the size or location of any of the answer boxes and that your answers are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.

  You may also digitally annotate the pdf. Illegible handwriting will lead to lost points. However, we suggest that try to do at least some of your work directly in LaTeX.

- **How to submit written component:** Submit to Gradescope a pdf with your answers. Again, make sure your answer boxes are aligned with the original pdf template.
- **How to submit programming component:** See Programming combonent for details on how to submit to the Gradescope autograder.
- **Policy:** See the course website for homework policies, including late policy, and academic integrity policies.
- **Proofs and Derivations:** For full credit, you must clearly show all nontrivial steps. Explicit reasons for each step are not required but can be helpful, especially if the step is not obvious.

Name	
Andrew ID	
Hours to complete all components (nearest hour)	

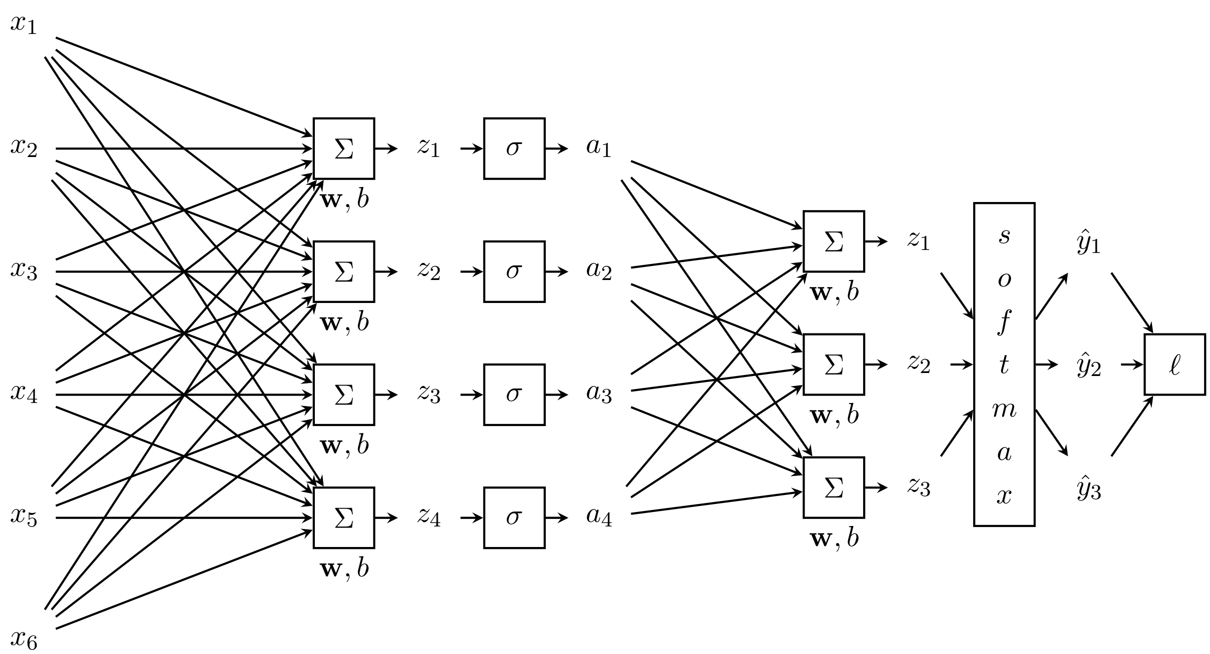
1 Backpropagation in Neural Network layers

In this part, you will derive the necessary backpropagation operations to efficiently implement a simple neural network. You will begin by deriving the formulas for the gradients of each operation in a neural network, including the linear layer, the sigmoid activation layer, and the softmax operator. In part 2, you will use these formulas to derive the gradients for a small neural network.

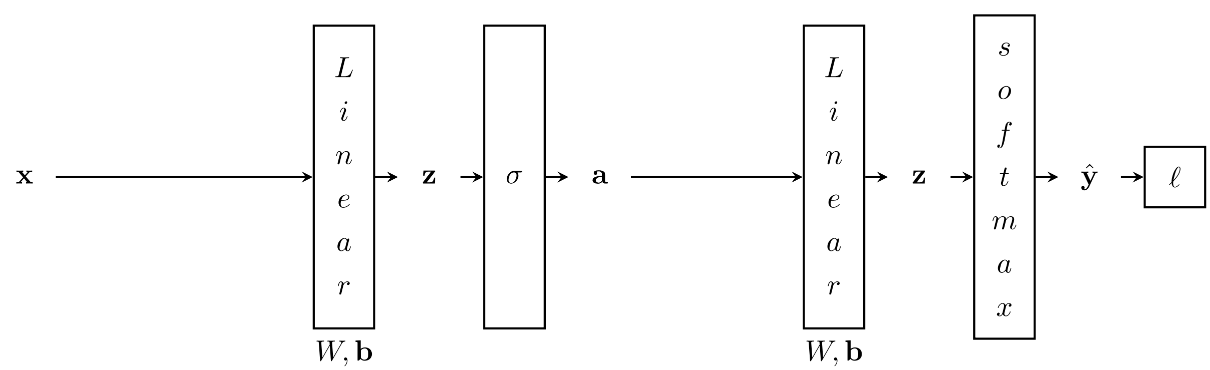
Note: In this assignment, your final answer should **not** contain any summations. You may have summations in the steps leading up to your final answer, but not in the final answer itself. This will be crucial for your implementations in the programming section of the assignment.

Example fully connected classification network

Network diagram (neuron view)



Network diagram (layer view)



Some notes on notation

In this assignment, we'll be using **denominator layout**.

Recall that the denominator layout for the gradient of a scalar y with respect to a N dimensional vector $\mathbf{x}$ is given by:

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_N} \end{bmatrix}$$

where $\mathbf{x}$ is a N dimensional vector.

If we have a scalar y and a $N \times M$ matrix X, the partial derivative of y with respect to X, $\frac{\partial y}{\partial X}$, in denominator format is:

$$\frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial y}{\partial X_{1,1}} & \cdots & \frac{\partial y}{\partial X_{1,M}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial X_{N,1}} & \cdots & \frac{\partial y}{\partial X_{N,M}} \end{bmatrix}$$

1. **[42 pts] Backpropagation for a linear layer**

   Recall the formula for a linear layer that maps from M-dimensional space to N-dimensional space:

   $$\mathbf{z} = W\mathbf{x} + \mathbf{b}$$

   - $\mathbf{x}$ is the M-dimensional input vector (to this layer; not necessarily the input to the full network)
   - $\mathbf{z}$ is the K-dimensional output vector,
   - W is the weight matrix of size $K \times M$, and
   - $\mathbf{b}$ is the bias vector of dimension K.
   - $\hat{\mathbf{y}}$ is a function of the output vector $\mathbf{z}$. $y^{(i)}$ is a one-hot vector $\mathbf{y}$ in K-class classification.
   - $J = \ell(y^{(i)}, \hat{y}^{(i)})$ is the loss function for a single data example ($N = 1$). In this assignment, we are referring to J with respect to one training example. In other words, you can think of $J = \ell(\hat{\mathbf{y}}, \mathbf{y})$.

   When we perform backpropagation on a linear layer, we are given $\frac{\partial J}{\partial \mathbf{z}}$, and we wish to use this to compute $\frac{\partial J}{\partial \mathbf{x}}$, $\frac{\partial J}{\partial W}$, and $\frac{\partial J}{\partial \mathbf{b}}$.

   Continued on next page.

(a) **[14 pts]** Derive $\partial J/\partial \mathbf{x}$ in terms of $\partial J/\partial \mathbf{z}$, $\mathbf{x}$, $\mathbf{z}$, W, and $\mathbf{b}$. Note that you will need some but not all of these variables for your derivation.

Your derivation must include the derivation for a single entry in $\partial J/\partial \mathbf{x}$, i.e., $\partial J/\partial x_j$.

For full credit on this and **all following derivations and proofs**, you must clearly show all nontrivial steps. Explicit reasons for each step are not required but can be helpful, especially if the step is not obvious.

$\partial J/\partial \mathbf{x}$

(b) **[14 pts]** Derive $\partial J/\partial W$ in terms of $\partial J/\partial \mathbf{z}$, $\mathbf{x}$, $\mathbf{z}$, W, and $\mathbf{b}$. Note that you will need some but not all of these variables for your derivation.

Your derivation must include the derivation for a single entry in $\partial J/\partial W$; i.e. $\partial J/\partial W_{k,j}$.

Reminder that your final answer should **not** contain any summations and should be expressed using only matrix operations.

$\partial J/\partial W$

(c) **[14 pts]** Derive $\partial J/\partial \mathbf{b}$ in terms of $\partial J/\partial \mathbf{z}$, $\mathbf{x}$, $\mathbf{z}$, W, and $\mathbf{b}$. Note that you will need some but not all of these variables for your derivation.

Your derivation must include the derivation for a single entry in $\partial J/\partial \mathbf{b}$, i.e., $\partial J/\partial b_k$.

$\partial J/\partial \mathbf{b}$

2. **[16 pts] Backpropagation for sigmoid activation**

Recall the vector version of logistic (sigmoid) activation function $\mathbf{a} = \sigma(\mathbf{z})$, where:

$$a_k = \frac{1}{1 + e^{-z_k}}$$

Also, recall the derivative for the scalar logistic function, $d\sigma/dz = \sigma(z)\,(1 - \sigma(z))$.

Given $\partial J/\partial \mathbf{a}$, derive $\partial J/\partial \mathbf{z}$ in terms of $\partial J/\partial \mathbf{a}$, $\mathbf{a}$, and $\mathbf{z}$. Note that you will need some but not all of these variables for your derivation. You may use $\circ$ to denote element-wise product in your answer (the Hadamard Product).

Your derivation must include the derivation for a specific entry in $\partial J/\partial \mathbf{z}$, i.e., $\partial J/\partial z_k$.

$\partial J/\partial \mathbf{z}$

3. **[42 pts] Backpropagation for softmax and cross-entropy loss**

Cross-entropy loss is often used when a network ends with a softmax layer. In this question, we'll see that it can be convenient to combine these two into a single layer because the gradient simplifies quite nicely.

Let's start with the softmax function, $\hat{\mathbf{y}} = g_{softmax}(\mathbf{z})$, where

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

where z_k is the input to the softmax for class k, and K is the total number of classes.

Looking at the dimensions of the partial derivative for the output of the softmax function with respect to its input, $\partial \hat{\mathbf{y}} / \partial \mathbf{z}$, we can see that it is a $K \times K$ matrix, where the diagonal elements are $\partial \hat{y}_k / \partial z_k$ and the off-diagonal elements are $\partial \hat{y}_k / \partial z_j$ for all $j \neq k$.

Before we combine the softmax with cross-entropy, derive the off-diagonal and diagonal entries of the softmax partial derivative. Note: You may choose to simplify these expressions here or wait until the following cross-entropy part.

Off-diagonal: $\partial \hat{y}_k / \partial z_j$

Diagonal: $\partial \hat{y}_k / \partial z_k$

Recall the cross-entropy loss:

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

where $\mathbf{y}$ is the true label (as a one-hot vector) and $\hat{y}_k$ is the predicted probability for class k, and K is the total number of classes.

Building on your answers to the previous part, prove that $\partial J/\partial \mathbf{z} = \hat{\mathbf{y}} - \mathbf{y}$.

Your derivation must include the derivation for a specific entry in $\partial J/\partial \mathbf{z}$, i.e., $\partial J/\partial z_j$.

$\partial J/\partial \mathbf{z}$

2 Collaboration Policy

After you have completed all other components of this assignment, report your answers to the following collaboration questions.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details including names of people who helped you and the exact nature of help you received.

   Your Answer

2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details including names of people you helped and the exact nature of help you offered.

   Your Answer

3. Did you find or come across code that implements any part of this assignment? If so, include full details including the source of the code and how you used it in the assignment.

   Your Answer