The University of Alabama

College of Engineering · Computer Science

Assignment 5

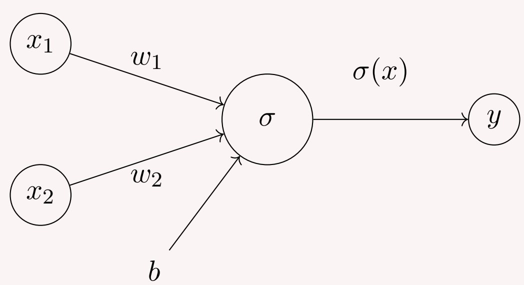
Written Assignment — Past Exam Problems

Submission Instructions

- Submit your solutions as a **separate PDF file**.
- Clearly label each problem number and part (e.g., Problem 3b).
- For any Python-related problems, include your code and the output (or state clearly if a line would error).
- Problem 5 includes a built-in figure in the PDF. If your instructor gives a separate scan of the original exam figure, follow their instructions.
- Ensure your work is legible and well organized. Show all steps for full credit.

Problem 1: Perceptron gates (14 Points)

Use the perceptron classifier to design the following gates.



- **NAND:** Write down the truth table and compute the weights and bias **(7 points)**.
- **NOR:** Write down the truth table and compute the weights and bias **(7 points)**.

Solution:

∎

Problem 2: XOR and the perceptron (6 Points)

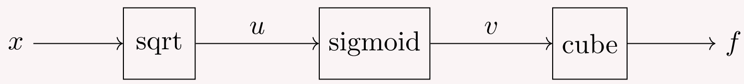
Explain why the XOR function cannot be computed using a single perceptron. Use graphs to support your argument **(6 points)**.

Solution:

■

Problem 3: Computational graph (20 Points)

For the following computational graph:



(a) Write the nested function in math notation $f(x) = ?$ **(4 points)**.

(b) Assume $x = 2$, compute u, v, and f **(4 points)**.

(c) Compute $\frac{\partial f}{\partial x}\Big|_{x=2}$ using the finite difference rule and the nested function notation you wrote **(6 points)**:

$$\frac{df}{dx}\Big|_{x=a} = \frac{f(a+\triangle) - f(a-\triangle)}{2\triangle}$$

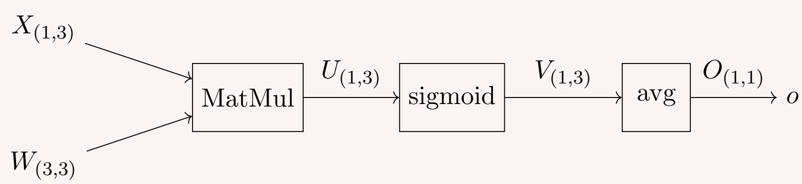
(d) Write the formula for the chain rule $\frac{\partial f}{\partial x}\Big|_{x=2}$ in terms of u, v and use the partial derivatives $\frac{\partial u}{\partial x}$, $\frac{\partial v}{\partial u}$, and $\frac{\partial f}{\partial v}$ to compute the derivative again and verify you get the same answer **(6 points)**.

Solution:

■

Problem 4: Small neural network (one sample) (25 Points)

For the following neural network we have only one sample, so the input is a vector of size 1×3.



Given

$$X_{(1,3)} = \begin{bmatrix} 0.5 & 0.1 & 0.6 \end{bmatrix}, \qquad W_{(3,3)} = \begin{bmatrix} 0.1 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.2 \\ 0.3 & 0.7 & 0.4 \end{bmatrix}.$$

(a) Compute the intermediate matrix $U_{(1,3)}$, $V_{(1,3)}$ and $O_{(1,1)}$ **(7 points)**.

(b) Write down the chain rule for $\frac{\partial O}{\partial W}$ and write down the rules to compute the partial derivatives $\frac{\partial U}{\partial W}$, $\frac{\partial V}{\partial U}$ and $\frac{\partial O}{\partial V}$ **(13 points)**.

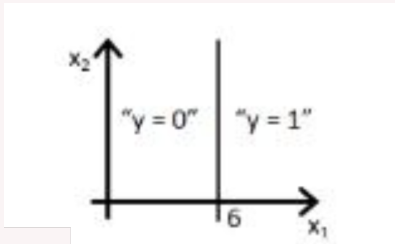
(c) Use numeric differentiation to verify your result for only one element of W, say $w_{1,1}$, where you can recompute O at $w_{1,1} + \epsilon$ and then use the difference to approximate the derivative **(5 points)**.

Solution:

■

Problem 5: Logistic decision boundary (5 Points)

For the logistic regression represented by the decision boundary in the figure below and the equation $y' = \sigma(w_2 x_2 + w_1 x_1 + w_0)$. The boundary is linear; in the schematic, it is the **vertical line** $x_1 = 6$.



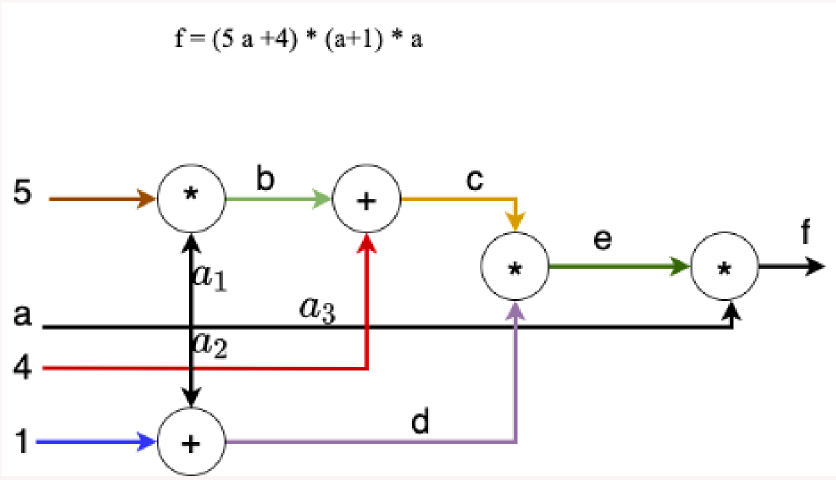
(a) Find the values for w_2, w_1, and w_0 **(5 points)**.

Solution:

■

Problem 6: Computational graph (7 Points)

For the following computational graph of the equation:



Assume $a = 2$ for all numerical values you report in parts (a) and (b).

(a) Calculate the forward information passing through a_1, a_2, a_3, b, c, d, e, and f. **(2 points)**

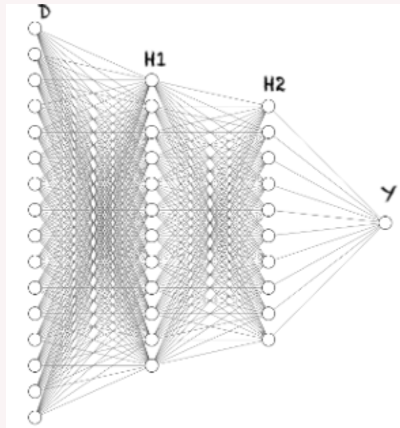
(b) Calculate the backward information passing through a_1, a_2, a_3, b, c, d, e. **(5 points)**

Solution:

■

Problem 7: Feedforward network parameter count (8 Points)

Consider a feedforward neural network for binary classification with two hidden layers. Define the number of neurons in each hidden layer as H_1 and H_2, respectively.



(a) Formulate the mathematical expression to compute the **total number of parameters** in the network, including weights and biases. Express the result in terms of the input dimension D, the hidden dimensions H_1 and H_2, and the number of output neurons ($Y = 1$), assuming a binary classification task with a single output neuron and a sigmoid activation function. **(5 points)**

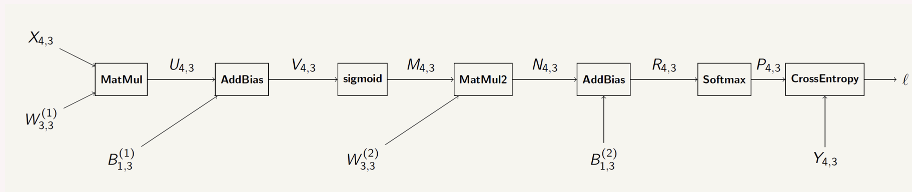
(b) Given $D = 1000$, $H_1 = 256$, $H_2 = 128$, and $N = 1$ (as on the original exam), calculate the total number of parameters in the neural network. **(3 points)**

Solution:

■

Problem 8: Computational graph: gradients for $W^{(1)}$, $W^{(2)}$, $B^{(1)}$, $B^{(2)}$ (15 Points)

Use the graph from Module 07 below (batch 4, 3 features, 3 logits per sample).



Shapes: $X \in \mathbb{R}^{4\times3}$, $W^{(1)}, W^{(2)} \in \mathbb{R}^{3\times3}$, $B^{(1)}, B^{(2)} \in \mathbb{R}^{1\times3}$ (broadcast on rows). Forward: $U = XW^{(1)} + B^{(1)}$, $M = \sigma(V)$ with $\partial V/\partial U = \mathbf{1}_{4,3}$, $N = MW^{(2)} + B^{(2)}$, then logits R and loss ℓ as in the figure. With one-hot Y and predicted P,

$$\frac{\partial \ell}{\partial R} = Y - P.$$

Write each answer using the lecture's $\odot$-chain and block shapes (4×3 where indicated). For $\partial\ell/\partial B^{(1)}$ and $\partial\ell/\partial B^{(2)}$, reduce the 4×3 result to $\mathbb{R}^{1\times3}$ as in class (no need to justify each local Jacobian separately).

(a) $\dfrac{\partial \ell}{\partial W^{(2)}}$. **(4 points)**

(b) $\dfrac{\partial \ell}{\partial B^{(2)}}$. **(4 points)**

(c) $\dfrac{\partial \ell}{\partial W^{(1)}}$. **(4 points)**

(d) $\dfrac{\partial \ell}{\partial B^{(1)}}$. **(3 points)**

Solution:

■