

---

# CSE517A – HOMEWORK 4

---

M. Neumann

**11 Apr 2025**

- Please keep your written answers brief and to the point. Incorrect or rambling statements can hurt your score on a question.
- If your hand writing is not readable, we **cannot give you credit**. We recommend you type your solutions in L<sup>A</sup>T<sub>E</sub>X and compile a .pdf for each answer. **Start every problem on a new page!**
- This will be due **11 Apr 2025** at **11:59pm** with an automatic 3-day extension.
- You may work in groups of at most 2 students.
- Submission instructions:
  - Start every problem on a **new page**.
  - Submissions will be exclusively accepted via **Gradescope**. Find instructions on how to get your Gradescope account and submit your work on the course webpage.

Note, that if you use **any resources** outside the course materials to derive (part of) your solution, you will need to cite the source in your homework submission. This also holds for **online sources**. If you collaborate with anyone other than your partner, it is your responsibility to indicate this in your submission. Course materials are the lecture notes, course books, and resources linked therein or on Canvas, plus course materials of any prerequisite course officially listed as such in the course listing.

Citing the source(s) does **not** legitimate the *copying* of existing solutions to any given problem, neither does it legitimate that another person (that is not you or your partner) directly solves any (part of the) problems for you. Please, refer to the **course syllabus** for more details.

**Problem 1 (5 points) Warm-up: Derivatives of vectors and with respect to vectors**

What is  $\frac{\partial y}{\partial \mathbf{x}}$ ,  $\frac{\partial \mathbf{y}}{\partial x}$ , and  $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ ? Provide the vector or matrix of partial derivatives.

**Problem 2 (15 points)** Neural Networks are the basis for many modern-day AI systems. Find out more about your favorite AI system and its underlying NN.

To answer the following questions, you may use *any resource* (including AI systems). However, you are required to **provide the reference to the source** for each piece of information. The reference/citation has to be the actual source document/webpage – providing that the answer was generated by an AI system is not enough – we want to practice fact-checking our findings!

**(basic info)** What is the purpose, name, version, and release/creation date of the AI system/NN?

**(size)** How big is the NN? What is its *architecture* and its *number of parameters*?

**(training data)** On how much and what data was it trained?

**(training time)** How long did it take to train it?

**(hardware)** On what hardware was it trained? And what is the acquisition cost (in USD) of this hardware? If you can't find a definite answer, give your best estimate and justify/explain your calculation.

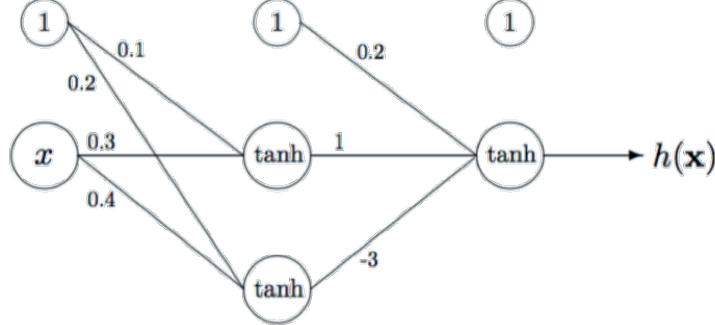
**(energy cost)** How much energy was used to train it? Think *energy cost* (in USD), but also *kilowatt-hours* (kWh) and *CO<sub>2</sub> emissions/carbon footprint*. Compare emissions to something you can relate to (flight from STL to ORD, emissions of average consumer household per month, operation of a large hospital/university (for a week), etc.). If you can't find a definite answer, give your best estimate and justify/explain your calculation.

**(cost of operation)** What does it take to run it? Think hardware and energy costs of running the AI system on a day-to-day basis. If you can't find a definite answer, give your best estimate and justify/explain your calculation.

**(interesting find)** Report on some other surprising or interesting fact or piece of information you learned from your investigation.

### Problem 3 (10 points) Some Practice with nns

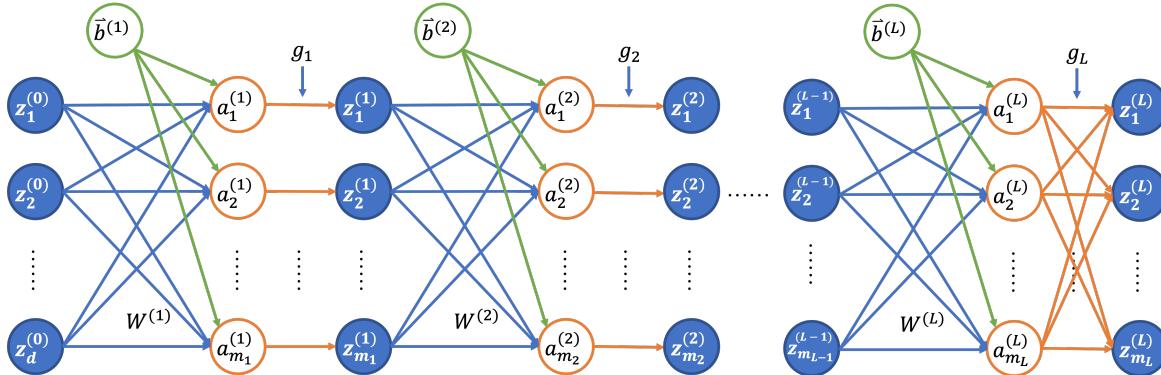
Consider the following neural network:



- (a) (5 pts) Compute one forward propagation pass for  $x = 2$ . Clearly state the results for all  $a^{(\ell)}$ 's and  $z^{(\ell)}$ 's.
- (b) (5 pts) Assume  $y = 1$ , compute one backpropagation pass using the squared loss. Clearly state the results for all  $\delta^{(\ell)}$  and weight updates  $\Delta W^{(\ell)}$  and  $\Delta b^{(\ell)}$ .

### Problem 4 (30 points) NNs for Multi-class Classification

Consider the following fully-connected **feed-forward neural network** for multi-class classification with  $L > 2$  layers and  $m_i > 2, i = \{1,..L\}$  units per layer:



- (a) (5 pts) [Output Layer] How many classes does your multi-class classification task have? Define the outputs  $z_i^{(L)}$  using an appropriate output activation function  $g_L$  and argue whether  $\vec{z}^{(L)}$  is a vector of probabilities.
- (b) (5 pts) [Forward Pass & Predictions] Write  $\vec{z}^{(\ell)}$  in terms of  $\vec{z}^{(\ell-1)}, W^{(\ell)}, \vec{b}^{(\ell)}$  and  $g_\ell$ . After completing the forward pass, we get the outputs  $\vec{z}^{(L)}$  for a given input  $\vec{x}_i$ . How do you determine the predicted class label  $\hat{y}_i$ .

(c) (5 pts) [Loss Function] For multi-class classification we use the *cross-entropy loss*. Provide its definition and simplify this loss function to only include entries in  $\vec{z}^{(L)}$  that are relevant for an observed  $y$ . Sketch the loss function for its relevant input range.

(d) (10 pts) [Back Propagation – Output Layer] Using the *simplified cross-entropy loss*  $\mathcal{L}(\vec{x}, y)$  derived in the previous part, derive the weight update rule for *all* entries in  $W^{(L)}$ , i.e.,  $w_{ij}^{(L)}$ , in terms of the appropriate variables  $z_y^{(L)}$  and  $z_i^{(L-1)}$ .

HINT: Include the derivative of the *output activation function* (cf. part (a)). That is, evaluate this expression:  $g'_L(a_i^{(L)}) = \frac{\partial g_L}{\partial a_i^{(L)}}$  based on  $z_y^{(L)}$  and  $z_i^{(L)}$ .

(e) (5 pts) [More Back Propagation] Given  $\delta_k^{(L)} = \frac{\partial \mathcal{L}}{\partial a_k^{(L)}}$   $\forall k$  (error contribution(s) of the last layer), derive  $\delta_j^{(L-1)} = \frac{\partial \mathcal{L}}{\partial a_j^{(L-1)}}$  (error contributions of the second last layer).

### Problem 5 (30 points) Activation Functions

(a) (5 pts) One of the most popular activation functions is the *rectified linear unit* (RELU). State and draw the function and its derivative. Carefully label the axes and mark the units.

(b) (10 pts) Explain the *vanishing gradient problem* and argue whether RELU suffers from this issue.

(c) (10 pts) Explain the *dying neurons problem* and argue whether RELU suffers from this issue. Include an explanation on what a *dead neuron* is.

(d) (5 pts) How can we overcome the dying neurons problem? Make sure you briefly explain how your suggested fix solves the problem.

### Problem 6 (10 points) Debugging Backpropagation

The *backpropagation algorithm* can be difficult to implement and debug. A simple trick to debug the algorithm is to compare the partial derivative computed by backpropagation algorithm (known as the *analytical derivative*) and its numerical approximation.

If we have a function  $f(x)$ , the *derivative of  $f$  at  $x$*  can be *approximated numerically* by using the finite difference approximation:

$$\frac{\delta f}{\delta x} = \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon) - f(x - \epsilon)}{2\epsilon}$$

(a) (5 pts) Compute that analytic derivative for  $f(x) = x^3$  and its numerical approximation at  $x = 2$  for  $\epsilon = 0.001$ .

(b) (5 pts) For NNS we will need this for functions of many variables  $w$  and  $b$ . To compute its derivative, we can randomly generate the parameters  $w$ 's and  $b$ 's, then iterate through each parameter at a time, vary each value by  $\epsilon$ . Provide the equation for the *numerical partial derivative* for a function  $f(w_{11}^{(\ell)}, \dots, w_{m_{\ell-1}m_\ell}^{(\ell)}, b)$ .