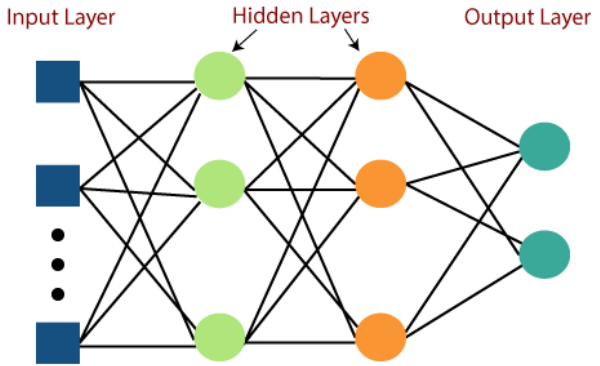
CS 4644/7643: Deep Learning

Assignment 1 Overview

Coding Overview

- Due **January 31st**
- Deliverables:
  - Code (Gradescope Autograder)
  - Report (Gradescope)

- Implement 2 MLPs: Softmax Regression and 2-Layer Neural Network
- Activation functions
  - Sigmoid, ReLU
- Cross-entropy Loss
- Optimizer (SGD with regularization)



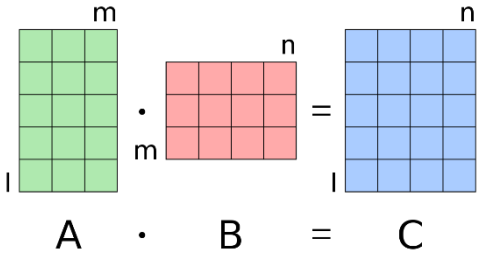
Tips & things to watch out for

- Include all plots generated in PDF report, add pages if necessary
- When uploading your report to Gradescope, **please tag the relevant question for each slide**; there will be a penalty of -1 point for each incorrectly tagged question
- Please avoid tagging the title slides; only tag slides that are directly related to each question.

Model Implementation: Softmax Regression

X: (N, 784)

y: (N,) → each is 1 of 10 possible labels



Forward	**Dimensions**
Z = XW	X: (N, 784), W: (784, 10)
R = ReLU(Z)	R: (N, 10) **(called “Logits”)**
S = softmax(R)	S: (N, 10)
L = CE(S, y)	Scalar (loss)

Model Implementation: Softmax Regression

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial R} \cdot \frac{\partial R}{\partial Z} \cdot \frac{\partial Z}{\partial W}$$

Backward	**Dimensions**
`L = CE(softmax(R), y)`	`Scalar (loss)`
`∂L/∂R = ∂CE(softmax(R), y)/∂R`	`Same as R`
`∂L/∂Z = ∂L/∂R * ∂R/∂Z`	`Same as Z`
`∂L/∂W = ∂L/∂Z * ∂Z/∂W`	`Same as W`

Convenient Derivation for: $\frac{\partial L}{\partial R}$

- dL/dR represents the derivative of the cross-entropy loss function wrt the INPUTS to the softmax function (logits).
- Read this article for a detailed explanation.

Model Implementation: Softmax Regression

Tips

- Reminder: check the matrix calculus office hours for various math refreshers
- If you are confused by working with batches, try working through an example with a single sample; then generalize to batches
- When performing matrix operations, think about the dimensions of the desired output and how you can arrive at that given the dimensions of the inputs (e.g. if transposes are needed, which matrix comes first)
- For CE Loss, be sure to take the average across the batch, not the sum

Problem Set Overview

Reminders:

- LaTeX submissions are strongly encouraged, while scanned handwritten copies are acceptable.
- Hard copy submissions **NOT** acceptable

- **Solutions to each problem/sub-problem must be on a separate page.**
- **Mark pages for every sub-problem accurately**

Important Mathematical Concepts for PS1

- How can we show that 2 vectors are orthogonal?

- Brush up on mathematical definition of convexity, local minimuma, global minima

- Computing gradients.
  - Derivative of scalar wrt to a vector: vector
  - Derivative of vector wrt to a vector: Jacobian Matrix

- Proofs
  - Proof by Induction
  - Proof by Contradiction

$$f(x_1, x_2, \ldots, x_n) = (f_1, f_2, \ldots, f_m)$$

$$J_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$