Plan

Introduction

Class Logistics

Math Review

Linear Regression

↳ One variable

↳ Many variables

Neighbor!

↳ Name?

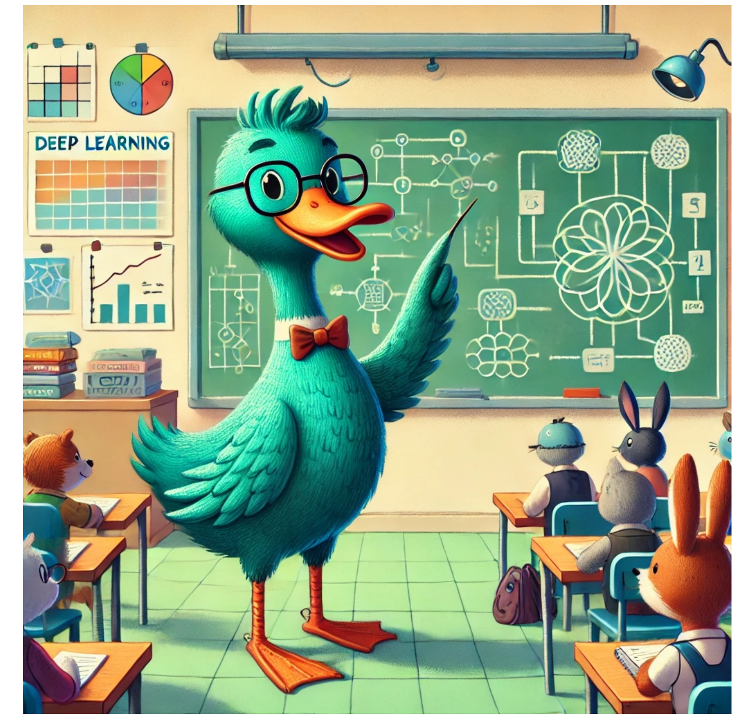
↳ Best food over break?

2023

Deep Learning!!

2025

what do you know about deep learning?



How does deep learning work?

Deep learning algorithms work by training artificial neural networks on a large dataset, using multiple layers of interconnected nodes to learn and make intelligent decisions. These neural networks are inspired by the structure and function of the brain, and are designed to recognize patterns and relationships in the data.

During the training process, the neural network is presented with a large number of examples, and for each example, the input data is passed through the network and the output is compared to the desired output. The errors between the predicted output and the desired output are then used to update the weights and biases of the network, using an optimization algorithm such as gradient descent. This process is repeated for many iterations, and as the network trains, it becomes able to make more accurate predictions on unseen data.

Once the training is complete, the deep learning model can be used to make predictions on new data, using the learned patterns and relationships to make intelligent decisions.

In one paragraph, please explain deep learning.

Deep learning is a subset of machine learning that focuses on using artificial neural networks with multiple layers (often called deep neural networks) to model and understand complex patterns in data. It is inspired by the structure and functioning of the human brain, where interconnected layers of "neurons" process information hierarchically. Deep learning excels at tasks like image and speech recognition, natural language processing, and playing games by automatically extracting and learning features from raw data without the need for manual feature engineering. This capability is enabled by advancements in computational power, large datasets, and optimization techniques like backpropagation, which iteratively adjust the network's weights to minimize errors.

Class

go/csci 1051

Prereqs:
- calculus
- linear algebra
- probability
- python

structure:
- 10am to noon here
- 2 to 4pm here
  (longer for work environment)
- 3 to 4pm office hours

Resources:
- these slides (before + after)
- notes ... that you'll write!
- canvas discussion

LLMs:

a tool (that makes errors)
Point is to learn material

Do
- short blocks of code you check
- simple suggestions you check

Don't
- lots of code
- complicated questions

Grades!!

Participation (13)

- 1 pt per lecture
- check in same day

Problem Sets (52)

- 4 pt per class
- practice to build skill

Project (30)

- Apply idea and tinker!
- Fast (like the class)

Scribed Notes (9)

- resource + familiarity
- sign up early!
- in lieu of problem that day
- can get 95 w/o

$[93, \infty)$	A
$[90, 93)$	A-
$[87, 90)$	B+
$[83, 87)$	B
⋮	

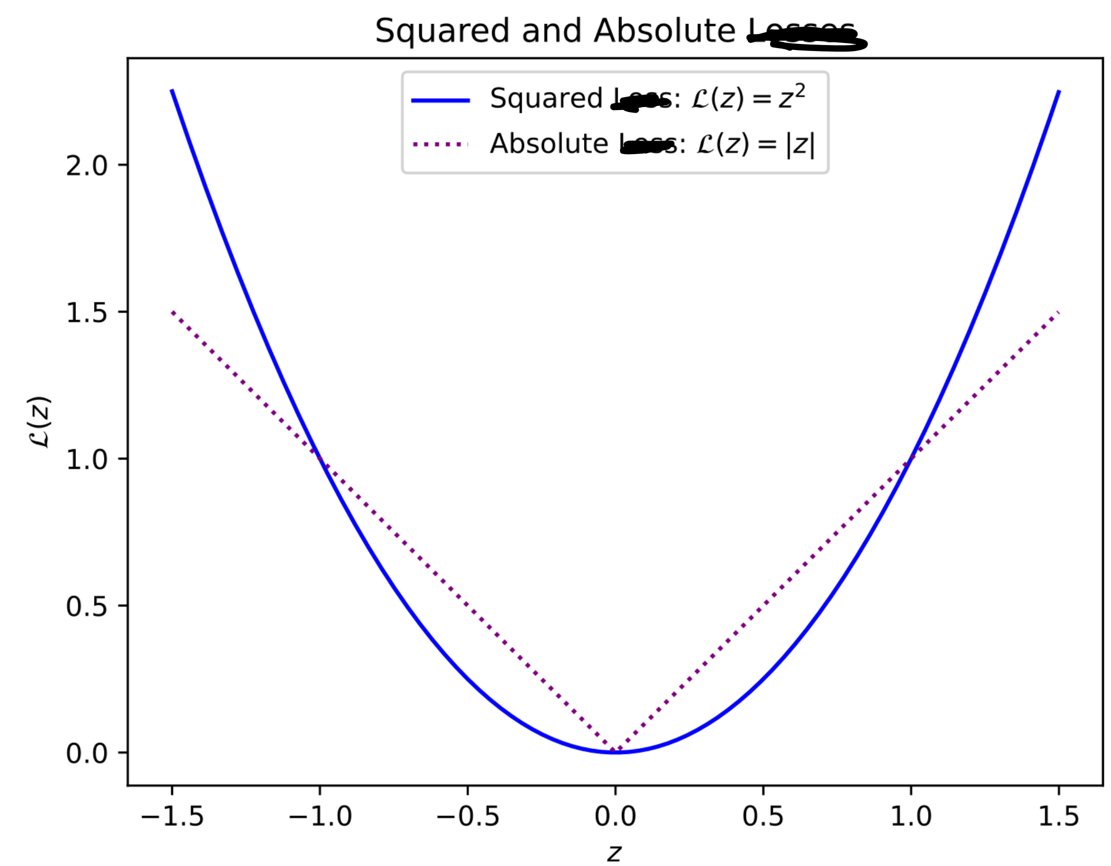
Math Review

Machine learning = math

$$\mathcal{L} : \mathbb{R} \to \mathbb{R}$$

e.g. $\mathcal{L}(z) = z^2$

$\mathcal{L}(z) = |z|$



$$\frac{\partial \mathcal{L}(z)}{\partial z} = \lim_{h \to 0} \frac{\mathcal{L}(z+h) - \mathcal{L}(z)}{h}$$

e.g. $\frac{\partial}{\partial z}\left(z^2\right) = 2z$

$$\frac{\partial}{\partial z}\left(z^a\right) = a z^{a-1}$$

$$\frac{\partial}{\partial z}\left(\mathcal{L}(z) + g(z)\right) = \frac{\partial \mathcal{L}}{\partial z} + \frac{\partial g}{\partial z}$$

$$\frac{\partial}{\partial z}\left(\ln(z)\right) = \frac{1}{z}$$

Chain Rule:

$$\frac{\partial}{\partial z} g(\mathcal{L}(z))$$
$$= \frac{\partial g}{\partial z}(\mathcal{L}(z)) \frac{\partial \mathcal{L}}{\partial z}(z)$$

Product Rule:

$$\frac{\partial}{\partial z}[g(z) \cdot \mathcal{L}(z)]$$
$$= g(z)\frac{\partial \mathcal{L}}{\partial z}(z) + \mathcal{L}(z)\frac{\partial g}{\partial z}(z)$$

Gradients:

$\mathcal{L}: \mathbb{R}^d \to \mathbb{R}$ now $z \in \mathbb{R}^d$

e.g. $\mathcal{L}(z) = \sum_{i=1}^{d} z_i^2$

$2z_j$

$$\frac{\partial \mathcal{L}}{\partial z_j}(z) =$$

treat all but z_j as constant

$$\nabla_z \mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial z_1} \\ \frac{\partial \mathcal{L}}{\partial z_2} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial z_d} \end{bmatrix} \in \mathbb{R}^d$$

$d \times 1$

Vectors and Matrices:

$u, v \in \mathbb{R}^d$

$$\begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix}$$

$d \times 1$

$$u \cdot v = \sum_{i=1}^{d} u_i v_i$$

$$\|v\|_2 = \sqrt{\sum_{i=1}^{d} v_i^2} = \sqrt{v \cdot v}$$

$A \in \mathbb{R}^{d \times m}$ $\quad$ $B \in \mathbb{R}^{m \times n}$

$$\begin{array}{cc} & \\ i \text{ row} & \begin{bmatrix} & & \\ \hline & & \\ & & \end{bmatrix} \end{array} \qquad \begin{array}{c} j \text{ col} \\ \begin{bmatrix} & \vert & \\ & \vert & \\ & & \end{bmatrix} \end{array}$$

$d \times m$ $\quad$ $m \times n$

$$\underset{\substack{d \times m \;\; m \times n \\ d \times n}}{AB} = \begin{array}{cc} & j \\ i & \begin{bmatrix} & & \\ & \blacksquare & \\ & & \end{bmatrix} \end{array}$$

BA ?

Inverse Matrix:

$$Ax = b$$

$$A \in \mathbb{R}^{d \times d} \quad x \in \mathbb{R}^{d} \quad b \in \mathbb{R}^{d}$$

$$A^{-1}A = I = \begin{bmatrix} 1 & 0 & 0 & \\ & 1 & & 0 \\ 0 & & 1 & \\ & & & 1 \\ & & & & 1 \end{bmatrix}$$

$$A^{-1}Ax = A^{-1}b$$

$$x = Ix = A^{-1}b$$

2

Supervised Learning

n labelled observations

$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$

For now, $x^{(i)}, y^{(i)} \in \mathbb{R}$

e.g. (temp today, temp tomorrow)

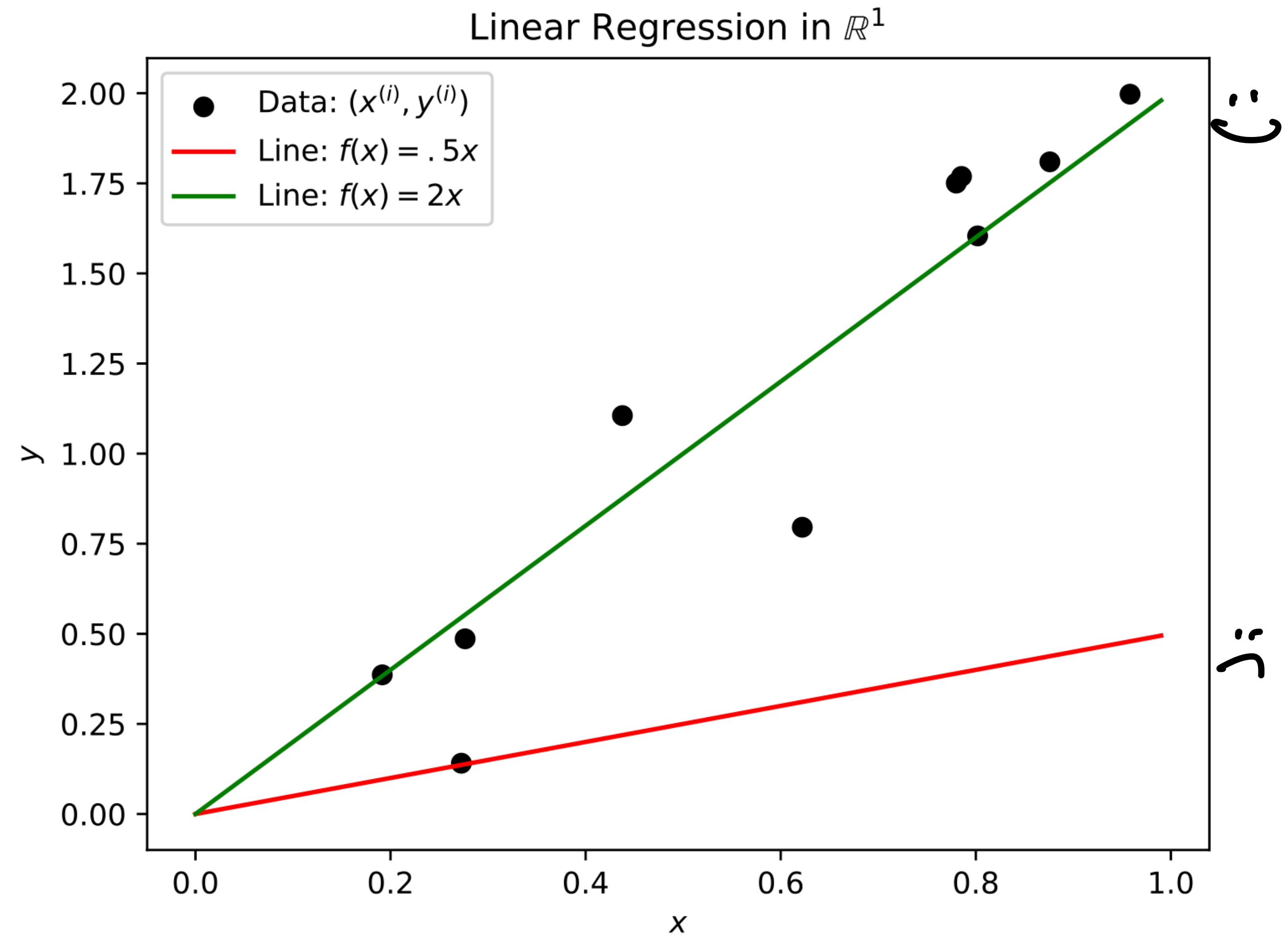
Goal: Learn $f: \mathbb{R} \to \mathbb{R}$
so that $f(x^{(i)}) \approx y^{(i)}$

3-Step Framework

1. Model turns data into output
2. Loss measures quality of model ($\downarrow$ = ☺)
3. Optimizer improves model to reduce loss

Linear Model

$$f(x) = wx$$



Mean Squared Error Loss

↳ depend on $f(x^{(i)}) - y^{(i)}$

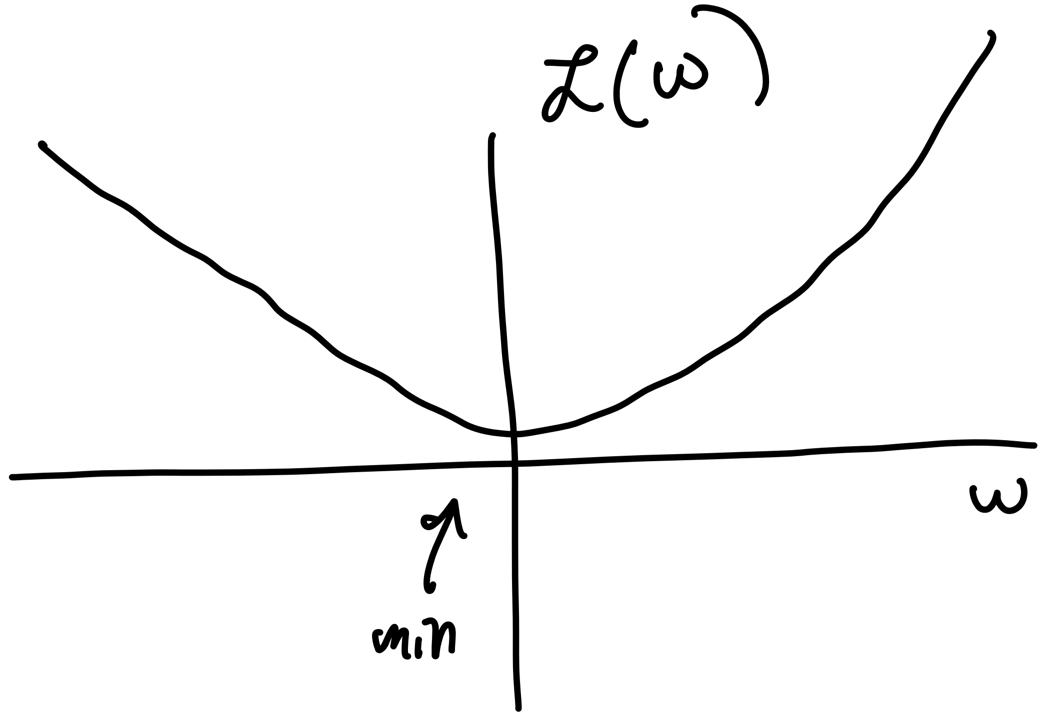
↳ always positive e.g. $|f(x)^{(i)} - y^{(i)}|$

↳ differentiable e.g. $(f(x^{(i)}) - y^{(i)})^2$

↳ independent of n e.g. $\frac{1}{n}\sum_{i=1}^{n} (f(x^{(i)}) - y^{(i)})^2$

Optimization

$$\mathcal{L}(w) = \frac{1}{n}\sum_{i=1}^{n}\left(f(x^{(i)}) - y^{(i)}\right)^2$$



weight

convex because sum of convex functions

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial}{\partial w}\frac{1}{n}\sum_{i=1}^{n}\left(f(x^{(i)}) - y^{(i)}\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w}\left(w x^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} 2\left(w x^{(i)} - y^{(i)}\right)\frac{\partial w x^{(i)}}{\partial w}$$

$$= \frac{2}{n}\sum_{i=1}^{n}\left(w x^{(i)} - y^{(i)}\right) x^{(i)} = 0$$

$$\sum_{i=1}^{n} w\left(x^{(i)}\right)^2 - \sum_{i=1}^{n} y^{(i)} x^{(i)} = 0$$

$$w \sum_{i=1}^{n}\left(x^{(i)}\right)^2 = \sum_{i=1}^{n} y^{(i)} x^{(i)}$$

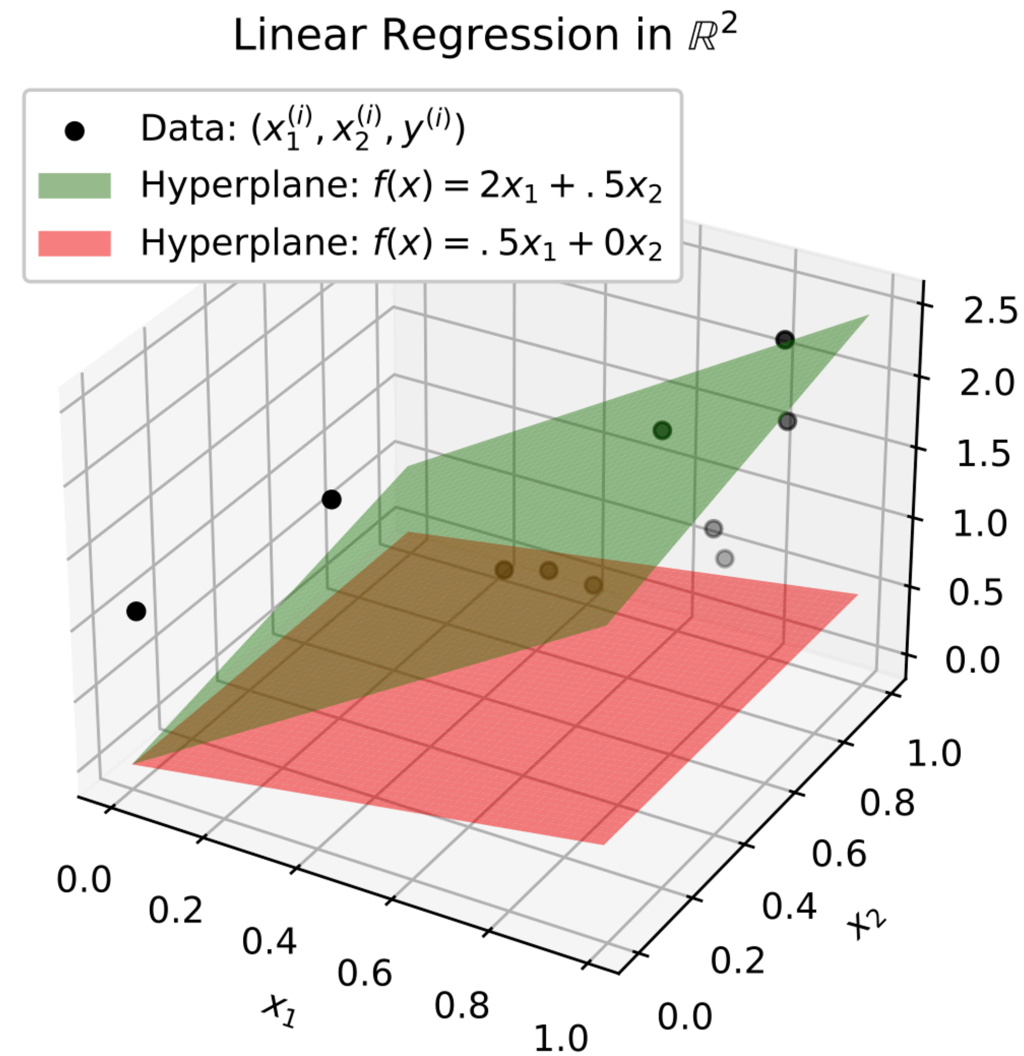
$$w = \frac{\sum_{i=1}^{n} y^{(i)} x^{(i)}}{\sum_{i=1}^{n}\left(x^{(i)}\right)^2}$$

Multivariate

$(x^{(1)}, y^{(1)}) \dots (x^{(n)}, y^{(n)})$

$x^{(i)} \in \mathbb{R}^d \qquad y^{(i)} \in \mathbb{R}$

$w \in \mathbb{R}^d$



① $f(x) = w \cdot x = \sum_{j=1}^{d} w_j x_j$

② $\mathcal{L}(w) = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} \cdot w - y^{(i)})^2$

$= \frac{1}{n} \| \underset{n \times d}{X} \underset{d \times 1}{w} - \underset{n \times 1}{y} \|_2^2$

$X \in \mathbb{R}^{n \times d} \qquad y \in \mathbb{R}^n$

Optimization

$\nabla_w \mathcal{L}(w) = 0$

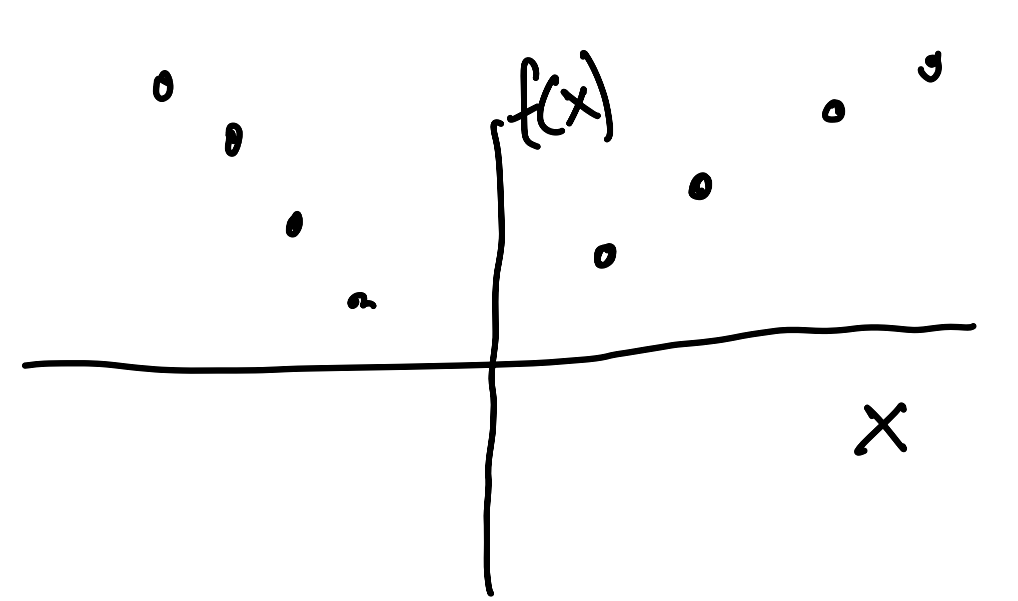
and solve for w

Pset :)

check in form!

Nagging Questions

1. What if data doesn't have linear pattern?



2. What if loss isn't convex?