Anil Poonai

Problem 1: Part A

F(x) = $||X||_2^2$

F(x) = $((\sum_{i=1}^{n} x_i^2)^{1/2})2$

F(x) = $\sum_{i=1}^{n} x_i^2$

Can ignore the summation as n is 1.

$\frac{\partial}{\partial x}$ = 2x

Problem 1: Part B

$\sum_{i=1}^{n} ||x_i - \mu||_2^2$

$\frac{\partial}{\partial x}$ = 2(x- μ) Got the derivative, could just put it in the answer for Part A but accounting for μ. This works because this is the derivative at each point, and we have the summation of all of the points already in the expression.

Now we set the derivative with summation $= 0$

$\sum_{i=1}^{n} 2(x_i - \mu) = 0$

Do some algebra and we end up with:

$\mu = (\sum_{i=1}^{n} x_i)/n$

Problem 2: Part A

L(w) = $||x||_1$

L(w) = $||Xw - y||_1$

Sizes: X(data) -> n*c, w(weights) -> c*1, y(labels) -> n*1

Problem 2: Part B

No, because L1 norm isn't differentiable at zero therefore gradient optimization cannot be used. The explanation is partly explained in Part C as well.

Problem 2: Part C

Part A was straightforward since we just need the difference from the estimate and actual value. Since L1 is the summation of ||x||, we can just replace Xw-y for x. For part B, it wouldn't have a value at zero if we took the derivative as it's non-convex there, there would be no unique global minimum so we can't minimize the loss function. This would mostly be concerned with the loss function part of the 3-step recipe as we are calculating the losses from the prediction regarding the label.

Problem 3: Part A

I do not have an answer for this question:

I first tried to do the math by having it all listed out in vectors and matrices but there's just a lot of numbers I would have to brute force. I did:

$[X][W_{HI}]+[B_{HI}] = Z_I$ where X is the input data and W_{HI} and B_{HI} is the weights and biases for each neuron in the hidden layer and Z_I is the output the ReLu activation function will take in. I then wrapped the ReLu function around that so that it would either be zero or Z_I, I then did the following:

$[B_O]+\sum_{i=1}^{n} [W_{OI}][Z_I] = y$, where B_O is the bias for the output layer, W_{OI} is the weights of the output layer, and Z_I is either o or the Z_I from before and y is the predicted output.

The problem I keep running into is that this just leads to a method where I have to brute force numbers for a solution, and this is a lot of trial and error.

I also tried just calculating it out straightforward for each x value listed on the datasets. This doesn't work out because that is even more brute forcing then the previous method.

I found it easier to setup an optimization using excel solver to find a solution than to do the math or code so I did that first.
My computer actually ran out of RAM for both datasets before finding a solution and I have 128 GB of RAM.

I also tried to code a solution in python, and I let it run for a few hours and nothing came from it.

The closest I got was just trying random numbers and I got 7/11 for dataset 2:

A	B	C	G	I	J	K	L	N	O	P	Q	S	T	U	V	W	X	Y	Z
		Parameters	Input	Z1	Z2	Z3	Z4	ReLu1	ReLu2	ReLu3	ReLu4	Prediction		Value for D1	Value for D2		Grade for D1		Grade for D2
w1	=	1.00	0	-10	-10	-10	-4	0	0	0	0	0	=	8	0		0		1
w2	=	1.00	1	-9	-9	-9	-3	0	0	0	0	0	=	6	1		0		0
w3	=	1.00	2	-8	-8	-8	-2	0	0	0	0	0	=	4	2		0		0
w4	=	1.00	3	-7	-7	-7	-1	0	0	0	0	0	=	2	3		0		0
b1	=	-10.00	4	-6	-6	-6	0	0	0	0	0	0	=	0	2		1		0
b2	=	-10.00	5	-5	-5	-5	1	0	0	0	1	1	=	2	1		0		1
b3	=	-10.00	6	-4	-4	-4	2	0	0	0	2	2	=	4	2		0		1
b4	=	-4.00	7	-3	-3	-3	3	0	0	0	3	3	=	2	3		0		1
w5	=	1.00	8	-2	-2	-2	4	0	0	0	4	4	=	0	4		0		1
w6	=	1.00	9	-1	-1	-1	5	0	0	0	5	5	=	2	5		0		1
w7	=	1.00	10	0	0	0	6	0	0	0	6	6	=	4	6		0		1
w8	=	1.00	11	1	1	1	7	1	1	1	7	10	=	6			0	Sum for D2	7
b0	=	0.00	12	2	2	2	8	2	2	2	8	14	=	8			0		
																Sum for D1	1		

That was with w1= 1, w2 = 1 w3 = 1, w4 = 1, b1 = -10, b2 = -10, b3 = -10, b4 = -4, w5 = 1, w6 = 1, w7 = 1, w8 = 1, and b0 = 0.

The closest I for was dataset 1 was 2/13:

Parameters			Input	Z1	Z2	Z3	Z4	ReLu1	ReLu2	ReLu3	ReLu4	Prediction		Value for D1	Value for D2		Grade for D1		Grade for D2
w1	=	-1.00	0	0	0	0	0	0	0	0	0	8	=	8	0		1		0
w2	=	-1.00	1	-1	-1	-1	-1	0	0	0	0	8	=	6	1		0		0
w3	=	-1.00	2	-2	-2	-2	-2	0	0	0	0	8	=	4	2		0		0
w4	=	-1.00	3	-3	-3	-3	-3	0	0	0	0	8	=	2	3		0		0
b1	=	0.00	4	-4	-4	-4	-4	0	0	0	0	8	=	0	2		0		0
b2	=	0.00	5	-5	-5	-5	-5	0	0	0	0	8	=	2	1		0		0
b3	=	0.00	6	-6	-6	-6	-6	0	0	0	0	8	=	4	2		0		0
b4	=	0.00	7	-7	-7	-7	-7	0	0	0	0	8	=	2	3		0		0
w5	=	1.00	8	-8	-8	-8	-8	0	0	0	0	8	=	0	4		0		0
w6	=	1.00	9	-9	-9	-9	-9	0	0	0	0	8	=	2	5		0		0
w7	=	1.00	10	-10	-10	-10	-10	0	0	0	0	8	=	4	6		0		0
w8	=	1.00	11	-11	-11	-11	-11	0	0	0	0	8	=	6			0	Sum for D2	0
b0	=	8.00	12	-12	-12	-12	-12	0	0	0	0	8	=	8			1		
																Sum for D1	2		

That was with w1= -1, w2 = -1 w3 = -1, w4 = -1, b1 = 0, b2 = 0, b3 = 0, b4 = 0, w5 = 1, w6 = 1, w7 = 1, w8 = 1, and b0 = 8.

Problem 3: Part B

This is just mean squares, so the derivative is straightforward:

$L(\vec{\theta}) = \sum_{i=1}^{n} \quad (y_i - f(x_i,\vec{\Theta}))^2$

$\nabla L(\vec{\theta}) = -2\sum_{i=1}^{n} \quad ((y_i - f(x_i,\vec{\Theta})) * \nabla f(x_i, \vec{\Theta}))$

I left it in terms of $\nabla f(x_i, \vec{\Theta})$ since that's what the question asked for.

Problem 3: Part C

First layer values after plugging in the input(x) and parameters.

Z_1 = -1

Z_2 = 3

Z_3 = 1

Z_4 = -1

After ReLu

Z_1 = 0

Z_2 = 3

Z_3 = 1

Z_4 = 0

After summation in output layer

-4

After adding final bias

-3

Did this both manually and on the excel calculator I made:

P3. C:

$$\bar{z}_i = w_{hi}\, x + b_{hi}$$

relu

$$Y = b_0 + \sum_{i=1}^{4} w_{0i}\, z_i$$

$x = 2$

$$\bar{z}_1 = -1 \to 0 \to 0$$
$$\bar{z}_2 = 3 \to 3 \to -3$$
$$\bar{z}_3 = 1 \to 1 \to -1$$
$$\bar{z}_4 = -1 \to 0 \to 0$$

-4

$\boxed{-3}$

Parameters		Input	Z1	Z2	Z3	Z4	ReLu1	ReLu2	ReLu3	ReLu4	Prediction
w1 =	-1	0	1	1	-1	1	1	1	0	1	0
w2 =	1	1	0	2	0	0	0	2	0	0	-1
w3 =	1	2	-1	3	1	-1	0	3	1	0	-3
w4 =	-1	3	-2	4	2	-2	0	4	2	0	-5

Problem 3: Part D

I have all the formulas written out but am not sure what the parameters are supposed to be. But once I'm given those, I can figure out what the derivative at x=2 is with no problem.

$$\frac{\partial L}{\partial w0} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial w0}$$

$$\frac{\partial y}{\partial w0} = \sum_{i=1}^{4} z_i$$

$$\frac{\partial L}{\partial y} = 2 \sum_{i=1}^{n} (y0 - f(x, \theta))$$

$\frac{\partial L}{\partial \mathrm{w0}}$ = 2 * $\sum_{i=1}^{4} z$ * $\sum_{i=1}^{n}(y0 - f(x,\theta))$

$\frac{\partial L}{\partial b0}$ = $\frac{\partial L}{\partial \mathrm{y}}$ * $\frac{\partial y}{\partial \mathrm{b0}}$

$\frac{\partial y}{\partial \mathrm{b0}}$ = 1

$\frac{\partial L}{\partial \mathrm{y}}$ = 2 $\sum_{i=1}^{n}(y0 - f(x,\theta))$

$\frac{\partial L}{\partial b0}$ = 2 $\sum_{i=1}^{n}(y0 - f(x,\theta))$

$\frac{\partial L}{\partial \mathrm{w1}}$ = $\frac{\partial L}{\partial \mathrm{y}}$ * $\frac{\partial y}{\partial \mathrm{z}}$ * max(0,z) * $\frac{\partial z}{\partial \mathrm{wo}}$

$\frac{\partial z}{\partial \mathrm{wo}}$ = x -> max(0,z) = max(0,x)

$\frac{\partial y}{\partial \mathrm{z}}$ = $\sum_{i=1}^{4} wo_{\mathrm{i}}$

$\frac{\partial L}{\partial \mathrm{y}}$ = 2 $\sum_{i=1}^{n}(y0 - f(x,\theta))$

$\frac{\partial L}{\partial \mathrm{w1}}$ 2 * $\sum_{i=1}^{4} wo$ * max(0,x) * $\sum_{i=1}^{n}(y0 - f(x,\theta))$

$\frac{\partial L}{\partial \mathrm{b1}}$ = $\frac{\partial L}{\partial \mathrm{y}}$ * $\frac{\partial y}{\partial \mathrm{z}}$ * max(0,z) * $\frac{\partial z}{\partial \mathrm{bo}}$

$\frac{\partial z}{\partial \mathrm{bo}}$ = 1 -> max(0,z) = 1

$\frac{\partial y}{\partial \mathrm{z}}$ = $\sum_{i=1}^{4} wo_{\mathrm{i}}$

$\frac{\partial L}{\partial \mathrm{y}}$ = 2 $\sum_{i=1}^{n}(y0 - f(x,\theta))$

$\frac{\partial L}{\partial \mathrm{w1}}$ 2 * $\sum_{i=1}^{4} wo$ * $\sum_{i=1}^{n}(y0 - f(x,\theta))$