Andrew
Curtis
9/30/22

Homework II

Problem 1

1.1 Calculate bias of $\bar{X}$. $\quad \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

$$E(\bar{X}) = E\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} E(X) \quad \rightarrow \quad E(X_i) = \mu$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mu$$

$$= \frac{n}{n}\mu$$

$$= \mu \qquad \text{Since } E(\bar{X}) = \mu \rightarrow$$

bias of $\bar{X}$ is $E(\bar{X}) - \mu = 0$

$\bar{X}$ is an unbiased estimator of μ.

2.1 Calculate variance of estimator $\bar{X}$.

$$Var(\bar{X}) = Var\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)$$

$$= \frac{1}{n^2}\sum Var(X_i)$$

$$= \frac{1}{n^2}\sum(\sigma^2) \quad \text{since } X_i\text{'s are independent}$$

$$= \frac{1}{n^2}\, n\sigma^2$$

$$= \frac{\sigma^2}{n}$$

The variance of $\bar{X}$ is $Var(\bar{X}) = \frac{\sigma^2}{n}$

3.] What is the MSE of the estimator $\bar{X}$?

$$MSE(\bar{X}) = E[(\bar{X}-\mu)^2] = Var(\bar{X}) + (E(\bar{X})-\mu)^2$$

$$= Var(\bar{X}) + (\text{Bias of } \bar{X})^2$$

$$= \frac{\sigma^2}{n} + 0^2$$

$$= \frac{\sigma^2}{n}$$

The MSE of estimator $\bar{X}$ is

$$MSE(\bar{X}) = \frac{\sigma^2}{n}$$

4.] Calculate bias of $\hat{s}^2$, if it is biased, is there a way to define a new estimator of σ^2 that is unbiased?

$$E(\hat{s}^2) = E\left(\frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X})^2\right) = E\left(\frac{1}{n}\sum\left((X_i-\mu)-(\bar{X}-\mu)\right)^2\right)$$

$$= E\left(\frac{1}{n}\sum\left((X_i-\mu)^2 - 2(\bar{X}-\mu)(X_i-\mu) + (\bar{X}-\mu)^2\right)\right)$$

$$= E\left(\frac{1}{n}\sum(X_i-\mu)^2 - \frac{2}{n}(\bar{X}-\mu)\sum(X_i-\mu) + \frac{1}{n}(\bar{X}-\mu)^2\sum 1\right)$$

$$= E\left(\frac{1}{n}\sum(X_i-\mu)^2 - 2(\bar{X}-\mu)^2 + (\bar{X}-\mu)^2\right)$$

$$= \frac{1}{n}E\left(\sum(X_i-\mu)^2\right) - E\left((\bar{X}-\mu)^2\right)$$

$$= \frac{1}{n}\sum E\left((X_i-\mu)^2\right) - Var(\bar{X})$$

$$= \frac{n}{n}\sigma^2 - \frac{\sigma^2}{n}$$

$$= \sigma^2 - \frac{\sigma^2}{n}$$

So the bias$(\hat{s}^2) = -\frac{\sigma^2}{n}$

Continued on next page.

Problem 1
4 cont.) $Bias(\hat{S}^2) = -\frac{\hat{S}^2}{n}$ try to find estimator of σ^2 that is unbiased?

instead say, $\hat{S}_u^2 = \frac{n}{n-1} S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$

$$E(\hat{S}_u^2) = E\left(\frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2\right)$$

$$= \frac{n}{n-1}\left(E(\hat{S}^2)\right)$$

$$= \frac{n}{n-1}\left(\frac{n-1}{n}\sigma^2\right) = \sigma^2 \rightarrow bias(S_u^2) = \sigma^2 - \sigma^2 = 0$$

So $\hat{S}_u^2 = \frac{n}{n-1}\hat{S}^2$ is an unbiased estimator of σ^2.

Problem 2

1.1 Suppose we take all the weights and biases in a network of perceptrons and multiply by a constant $c > 0$. Show that the behavior of the network doesn't change.

for a single perceptron, $$\text{output} = \begin{cases} 0 \text{ if } w \cdot x + b \leq 0 \\ 1 \text{ if } w \cdot x + b > 0 \end{cases}$$

if we multiply the weight and bias by c then,

$$\text{output} = \begin{cases} 0 \text{ if } wc \cdot x + bc \leq 0 \\ 1 \text{ if } wc \cdot x + bc > 0 \end{cases}$$

→ Since x inputs are binary, the weights and bias (threshold) will scale the same and output will be unaffected.

Example, say for 1 perceptron $w = 2$ and bias is -1

if $x_1 = 1$ $w_1 \cdot x_1 + (b_1) = 2 \cdot 1 + (-1) = 1 > 0$ output $= 1$
if $x_1 = 0$ $w_1 \cdot x_1 + (b_1) = 2 \cdot 0 + (-1) = -1 \leq 0$ output $= 0$

now multiply w_1 and b_1 by constant $c = 2$.

if $x_1 = 1$ $cw_1 \cdot x_1 + (cb_1) = 2 \cdot 2 \cdot 1 + (-1 \cdot 2) = 4 - 2 = 2 > 0$ output $= 1$
if $x_1 = 0$ $cw_1 \cdot x_1 + (cb_1) = 2 \cdot 2 \cdot 0 + (-1 \cdot 2) = -2 \leq 0$ output $= 0$

thus it holds true that multiplying weights and biases of a perceptron network by a positive constant does not affect behavior of the network

2.1 Show that in the limit $c \to \infty$ the behavior of this network of sigmoid neurons is exactly the same as the network of perceptrons. How can this fail when $w \cdot x + b = 0$ for one of the perceptrons?

output of sigmoid neuron: $\sigma(z) = \frac{1}{1 + \exp(-\sum_j w_j x_j + b)}$

output of sigmoid neuron multiplied by constant $c > 0$: $\sigma(c(w \cdot x + cb)) = \frac{1}{1 + e^{-(c(w \cdot x + cb))}}$

given that $w \cdot x + b \neq 0$ for each,

- if $w \cdot x + b > 0$, $\lim c \to \infty$, $\frac{1}{1 + e^{-(c(w \cdot x + cb))}} \to \frac{1}{1 + 0} = 1$
- if $w \cdot x + b < 0$, $\lim c \to \infty$, $\frac{1}{1 + e^{-(c(w \cdot x + cb))}} \to \frac{1}{1 + \infty} = 0$

So the sigmoid neuron will act like (give output) of a perceptron.

this holds true for all neurons, so the sigmoid network will act like a network of perceptrons in the case of $\lim c \to \infty$

if $w \cdot x + b = 0$, output of the sigmoid neuron is $\sigma(0) = \frac{1}{1 + e^0} = \frac{1}{2}$.

In this case it does not matter what c is, the sigmoid will not output a value of 0 or 1 like a perceptron.

3. For each of the possible input of the MLP in the figure, calculate the output. (perceptrons)

input	Output
0, 0, 0	$(0 \cdot .6 + 0 \cdot .5 + 0 \cdot -0.6 + (-0.4)) \cdot 1 + (0 \cdot (-.7) + 0(.4) + 0(.8) + (-.5)) \cdot 1 + (-0.5) = 0$
0, 0, 1	$(-.6 + -.4) + (-.3) \rightarrow 1$
0, 1, 0	$(.5 + -.4) + (0) = 1$
1, 0, 0	$(1) + 0 = 1$
0, 1, 1	$(.5 - .6 - .4)\ (1) = 1$
1, 0, 1	$(0)\ (0) = 0$
1, 1, 0	$(1)\ (0) = 1$
1, 1, 1	$(1)\ (0) = 1$

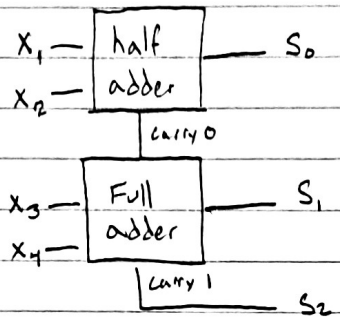
input	output
0 0 0	0
0 0 1	1
0 1 0	1
1 0 0	1
0 1 1	1
1 0 1	0
1 1 0	1
1 1 1	1

4] repeat 3. with sigmod neurons. $\frac{1}{1+e^{(wx+b)}}$

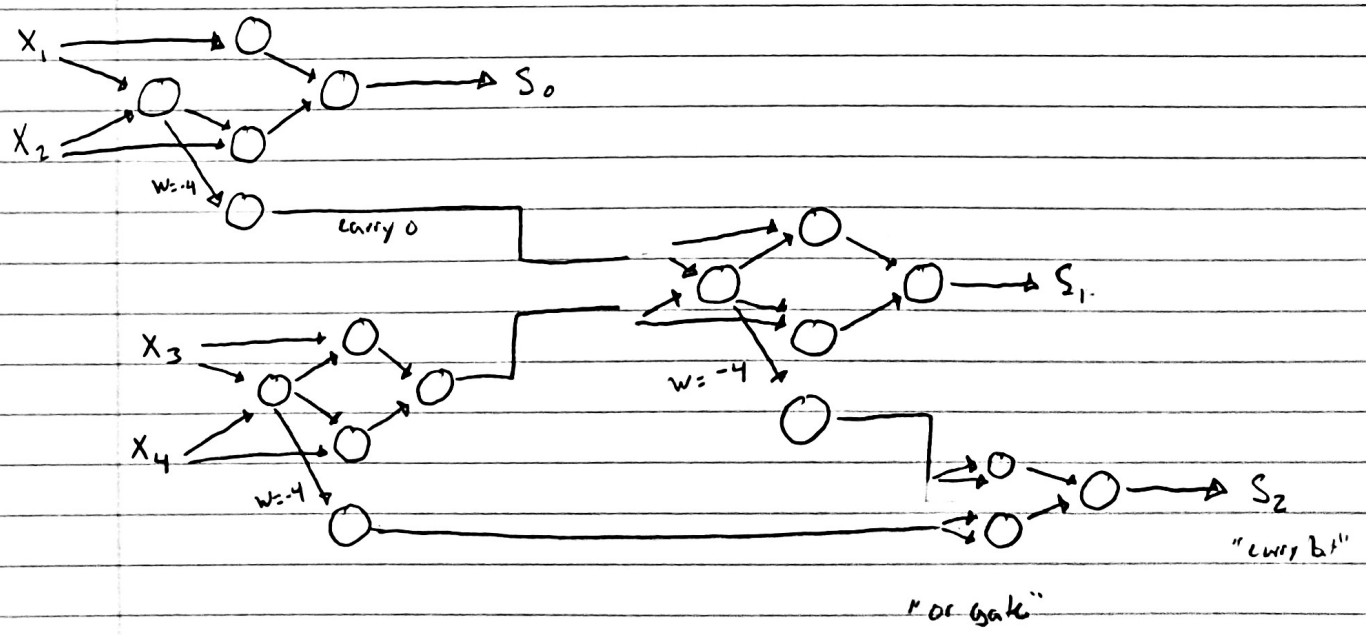
input	output
0 0 0	0.327
0 0 1	0.341
0 1 0	0.378
1 0 0	0.328
0 1 1	0.388
1 0 1	0.322
1 1 0	0.372
1 1 1	0.383

5. design an adder that does 2-bit binary addition.

* higher level diagram



Below is 2-bit binary adder using perceptrons. Weights and biases are -2 and 3 respectively, unless otherwise marked. This is to have perceptrons act as NAND Gates. Full adder can be constructed from two halfadders and an "or" gate.



Problem 3

for each scenario, describe whether it can be solved as a regression or classification problem or both.

1] Determine which kind of tree is in a picture.

Classification. Being visual and the inputs that are being searched for can be broken into classes, Discrete outcome.

2] Predict when someone will die based on health conditions.

Regression. This would likely take some numerical parameter inputs that we try to use to guess a continuous outcome

3] Predict the probability that a person will survive for more than 1 year after a cancer diagnosis based on their current health conditions.

Both. The reason why is because you could describe the probability as continuous or discrete. (i.e. classification if you want a yes or a no if they will die before a year or regression for a continuous probability over time).

4] Predict how many stars out of 5 a person will rate a product on Amazon based on previous reviews and history.

Both. the amount of stars could be used for classification. Or regression because it does follow an order and parameters could be used to predict continuously.

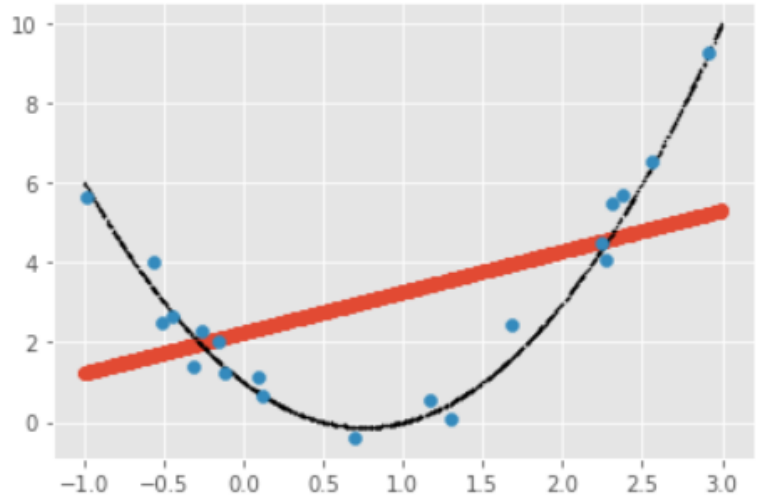
5] Text prediction.

Classification. A distinct value is being predicted

Problem 4.
Part 1.

- Implement and fit a linear regression model using pytorch
    - See attached code.

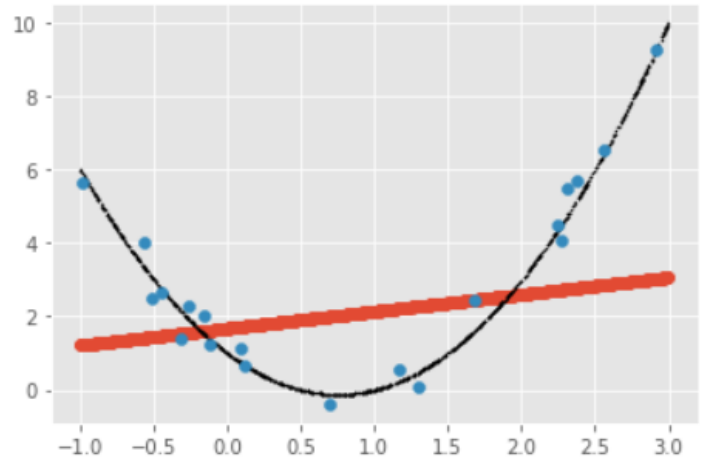
- Complete training routine.
    - See attached code.
- Plot the training data and test function using the test data
    - Included is the output of the model parameters for comparison in later parts.




```
[Parameter containing:
tensor([[1.0186]], requires_grad=True), Parameter containing:
tensor([2.2506], requires_grad=True)]
```

Part 2.

- Fit a 2nd degree polynomial and print the model parameters
    - Included also is the graph(see other answers on next page)

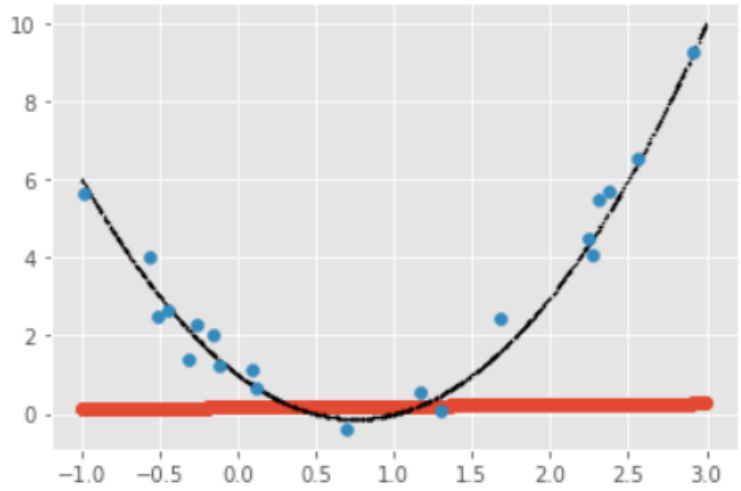



```
[Parameter containing:
tensor([[0.4621]], requires_grad=True), Parameter containing:
tensor([1.6762], requires_grad=True)]
```

- Did you obtain something close to the true parameters?
    - The parameters are 0.4621 and 1.6762, so quite different. However, the loss was a minimum with this 2nd degree fit.

Part 3.

- Fit the data to a 5th degree polynomial




```
[Parameter containing:
tensor([[0.0337]], requires_grad=True), Parameter containing:
tensor([0.1676], requires_grad=True)]
```

- Should the MSE be better than the previous two models?
    - The parameters are 0.0376 and -0.5511, so again quite different. The loss is very close to the 2nd degree polynomial. I think with 5 parameters it should be able to have a lower loss

than the 2nd degree. This is because it could overfit the data. Note, I had to tune the learning rate down quite slow to avoid nan loss results.

Part 4.

Fit another model and compute testing error.

```
testing error: -20772.859375, Noise:  15,  weight decay:  0, sigma:  0.1
testing error: -18980.724609375, Noise:  15,  weight decay:  0.2, sigma:  0.1
testing error: -24143.50390625, Noise:  15,  weight decay:  0.5, sigma:  0.1
testing error: -9206.1181640625, Noise:  100,  weight decay:  0, sigma:  0.1
testing error: -57257.203125, Noise:  100,  weight decay:  0.2, sigma:  0.1
testing error: -187507.59375, Noise:  100,  weight decay:  0.5, sigma:  0.1
testing error: -21530.22265625, Noise:  15,  weight decay:  0, sigma:  0.5
testing error: -23827.064453125, Noise:  15,  weight decay:  0.2, sigma:  0.5
testing error: -27623.6328125, Noise:  15,  weight decay:  0.5, sigma:  0.5
testing error: -114257.78125, Noise:  100,  weight decay:  0, sigma:  0.5
testing error: -99472.671875, Noise:  100,  weight decay:  0.2, sigma:  0.5
testing error: -150161.75, Noise:  100,  weight decay:  0.5, sigma:  0.5
testing error: -21524.748046875, Noise:  15,  weight decay:  0, sigma:  1
testing error: -53.83863067626953, Noise:  15,  weight decay:  0.2, sigma:  1
testing error: -21843.26953125, Noise:  15,  weight decay:  0.5, sigma:  1
testing error: -86995.0390625, Noise:  100,  weight decay:  0, sigma:  1
testing error: -30963.2890625, Noise:  100,  weight decay:  0.2, sigma:  1
testing error: -87532.2734375, Noise:  100,  weight decay:  0.5, sigma:  1
```

- For each sigma and noise value, which weight decay parameter performed the best?
    - For sigma = 0.1 and noise = 15, weight decay of 0.2
    - For sigma = 0.1 and noise = 100, weight decay of 0

    - For sigma = 0.5 and noise = 15, weight decay of 0
    - For sigma = 0.5 and noise = 100, weight decay of 0.2

    - For sigma = 1 and noise = 15, weight decay of 0.2
    - For sigma = 1 and noise = 100, weight decay of 0.2