Deep Learning HW2

Cody Grogan
A02313514

September 2023

Problem 1

For section 2, the author writes about how to plan and what to think about before deploying a model. The first is to understand the limitations of your data. This is important because it can prevent one from looking foolish in publication and allowing for a model to be as generalizable as possible. The second point is to make sure you have enough data to properly characterize the problem you model is trying to solve. Meaning that if model needs to pick up on small changes a significant amount of data may be needed to extrapolate them in training. The third point is to do your research on the topic you are trying to solve. Mainly looking at how others have attempted the problem and consulting experts for advice. The final point is to always think beyond prediction performance and think about the constraints of how the model will be used.

For section 4 the author writes about how to evaluate a model once it is being trained on data. The first section emphasizes the importance of a independent and representative test set to the actual problem being solved. The second section explains why optimization of hyperparameters on the test set will not always produce a generalized model. The third section shows the importance of evaluating a model multiple times on different test sets. This not only gives a better idea of the average performance but also improves the repeatability of model results. The final section explains how to use metrics to compare model performance. Most important is knowing the distribution of your data so your performance metric isn't skewed.

Problem 2

1.

Starting from assuming that f has 2 global minima x_1 and x_2 where $f(x_1) = f(x_2)$ but the minimizers are not equal to one another. Thus we can use these in the definition of strict convexity.

$$f(tx_1 + (1-t)x_2) < tf(x_1) + (1-t)f(x_2) \tag{1}$$

So if we can replace x_1 on the right with x_2 because we know $f(x_1) = f(x_2)$

$$f(tx_1 + (1-t)x_2) < tf(x_2) + (1-t)f(x_2) \tag{2}$$
$$f(tx_1 + (1-t)x_2) < f(x_2) \tag{3}$$
$$\tag{4}$$

Then taking t = 0 we see,

$$f(x_2) < f(x_2) \tag{5}$$

This then breaks the inequality because a point cannot be less than itself contradicting the assumption.

2.

If 2 convex functions are twice continuously differentiable then we know that their Hessians $\nabla^2 f_1$ and $\nabla^2 f_2$ are PSD matrices. Then we also know the sum of 2 PSD matrices is another PSD matrix. Thus proving that the sum of 2 convex functions is also convex.

3.

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c \tag{6}$$

Taking the first gradient with respect to x

$$\nabla_x f(x) = \nabla_x \frac{1}{2}x^T Ax + \nabla_x b^T x + c \tag{7}$$
$$= \frac{1}{2}(A + A^T)x + b^T \tag{8}$$
$$= Ax + b \tag{9}$$

Now taking the next gradient,

$$\nabla_x^2 f(x) = \nabla_x Ax + b^T \tag{10}$$
$$= A^T + 0 \tag{11}$$
$$= A \tag{12}$$

For convex, A must be PSD and for strictly convex A must be PD.

4.

For a convex function we know the following is true,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \tag{13}$$
$$(tx + (1-t)y)^3 \leq tx^3 + (1-t)y^3 \tag{14}$$
$$\tag{15}$$

Now choosing points y = 0 and x = -1 we get,

$$(-t)^3 \leq -t \tag{16}$$
$$\tag{17}$$

Now selecting t = 0.5 we see,

$$(-0.5)^3 \leq -0.5 \tag{18}$$
$$-0.125 \leq -0.5 \tag{19}$$

Showing x^3 is not convex

5.

Differentiating x^3 twice we find,

$$\frac{d^2 f}{dx^2} = 6x \tag{20}$$

From this we see we can put in a value of x = -6 showing that it is not positive definite.

6.

First taking the second derivative of -ln(x) we find,

$$\frac{d^2}{dx^2} f = \frac{1}{x^2} \tag{21}$$

Thus showing that for any values of x > 0 second derivative is positive and proving convexity.

7.

a)

Using the definition of convexity we see,

$$a(tx + (1-t)y) + b \leq t(ax+b) + (1-t)(ay+b) \tag{22}$$
$$a(tx + (1-t)y) + b \leq atx + tb + (1-t)ay + (1-t)b \tag{23}$$
$$a(tx + (1-t)y) + b \leq a(tx + (1-t)y) + b \tag{24}$$
$$0 \leq 0 \tag{25}$$

This then shows that the chord between x and y is always equal to any point between x and y. Thus the inequality can be changed and the same is true proving it is convex and concave.

b)

Lets assume there exists a global minimum x_1 such that $f(x_1) < f(x_2)$ for all $x_2 \in R$.

$$ax_1 + b < ax_2 + b \tag{26}$$
$$x_1 < x_2 \tag{27}$$

From this we see that f cannot have global minima from $(-\infty, \infty)$ because we can always choose a point such that $x_2 < x_1$. However when f is bounded and a is positive we know the lower bound is the global minimum and the upper bound is the global maxima. This is opposite when a is a negative number.

c)

Yes because a scalar function f = a is also concave and convex but does not take the form of an affine function.

Problem 3

1.

For a single perceptron we see the following if we multiply the weights and bias by c,

$$x\bar{W} + \bar{b} < 0 \tag{28}$$
$$xcW + cb < 0 \tag{29}$$
$$c(xW + b) < 0 \tag{30}$$
$$xW + b < 0 \tag{31}$$

From this we see that the equation for the perceptron is unchanged. Thus if we replaced W and b with larger matrices of weights and biases (like in a multilayer perceptron) the result would be the same.

2.

Having the same equation for a perceptron but with a sigmoid neuron we see,

$$\sigma(x\bar{W} + \bar{b}) = \frac{1}{1 + e^{-(x\bar{W}+\bar{b})}} \tag{32}$$

$$\sigma(x\bar{W} + \bar{b}) = \frac{1}{1 + e^{-(xcW+cb)}} \tag{33}$$

$$\sigma(x\bar{W} + \bar{b}) = \frac{1}{1 + e^{-c(xW+b)}} \tag{34}$$

$$\tag{35}$$

So as c approaches infinity we see that the sign of the original perceptron dictates the output of the neuron. Furthermore we see that the function switches between 1 or 0 instantaneously at about the point where the perceptron output is 0. This then extends to a network of sigmoid neurons because the sigmoid function is applied element-wise to each $xW + b$ for each neuron.
This assumption fails however when $xW + b$ is equal to zero because $e^0 = 1$ and will always result in an output of 0.5 and not 0 or 1.

3.

x_1	x_2	x_3	z_1	z_2	Final Out
0	0	0	-.4, -.5	-.5	0
0	0	1	-1, .3	.5	1
0	1	0	.1, -.1	.5	1
0	1	1	-.5, .7	.5	1
1	0	0	.2, -1.2	0.5	1
1	0	1	-.4, -.4	-.5	0
1	1	0	.7, -.8	.5	1
1	1	1	.1, 0	.5	1

4.

x_1	x_2	x_3	z_1	z_2	Final Out
0	0	0	-.4, -.5	.279	.569
0	0	1	-1, .3	.343	.585
0	1	0	.1, -.1	.5	0.622
0	1	1	-.5, .7	.546	.633
1	0	0	.2, -1.2	.281	.569
1	0	1	-.4, -.4	.302	.575
1	1	0	.7, -.8	.478	.617
1	1	1	.1, 0	.525	.628

Problem 4

1.

This is a classification problem because our output would be discrete in nature. Meaning that we are trying to put a tree into a bin with other trees of the same species with no inbetween cases.

2.

This is a regression problem because we are trying to pick a continuous number that represents the number of years they have left.

3.

Because of the threshold condition of this question we know that we are trying to bin people into dying within a year or more than a year. Thus this is a classification problem with discrete outputs.

4.

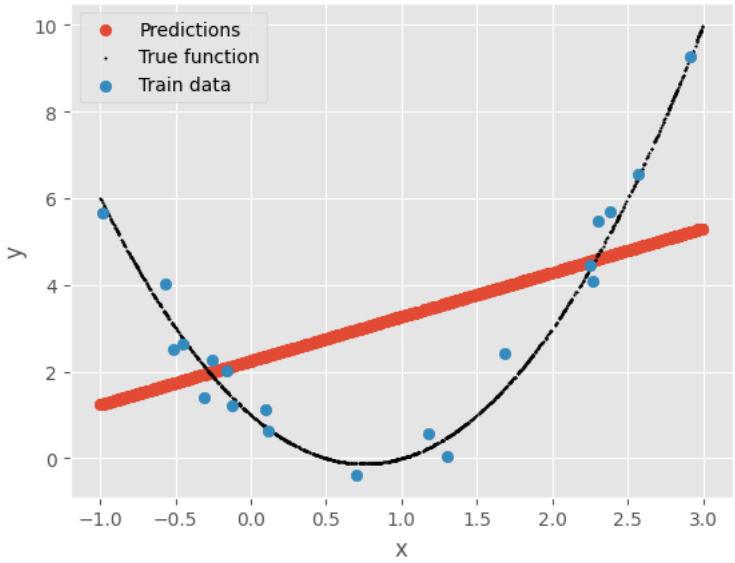
This problem can be both continuous or discrete in formulation. If we are trying to predict whether a person will give a 1, 2, 3, 4, or 5 star review the output will be discrete. However, if the rating system is continuous we will have a regression problem because the output will be continuous.

5.

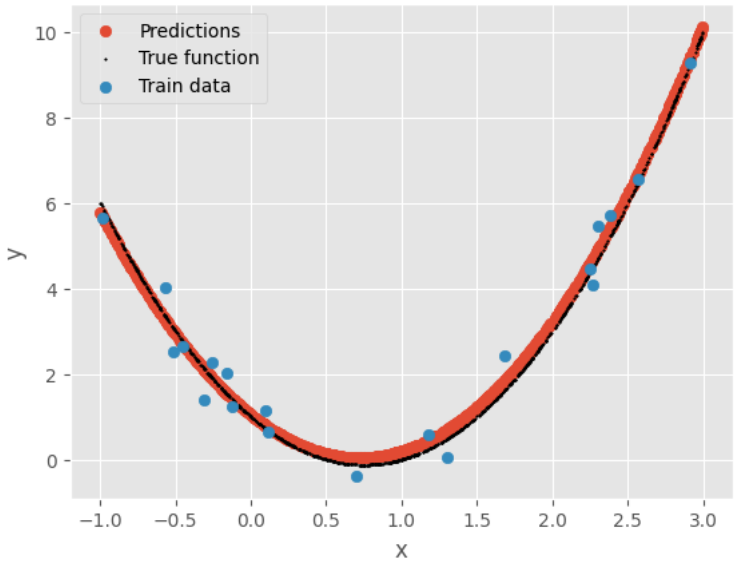
This is a classification problem in the sense that it is trying to find the word with the highest likelihood of fitting into the sentence. In addition, letters and thus words are discrete values showing this problem can be solved with classification and not regression.

Problem 5

1.



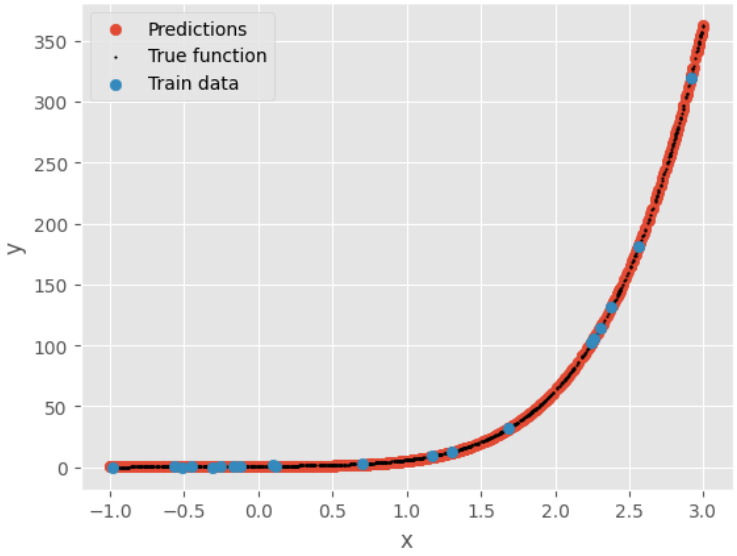
2.



The parameters for the predicted curve are, 1.94, -2.79 and bias 1.04, with true parameters of 2, -3, and bias 1. Yes the SGD found parameters very close

to the true parameters with some small differences.

3.



Yes visually the result matches the polynomial fairly well. However looking at the parameters shows a different story. For the polynomial I chose all coefficients of 1 but the model found parameters of 0.8958412, 1.3859569, 0.8578167, 0.9271538, 0.44738206, 0.6447907. These are moderately close to the correct coefficients but testing error outside of the generated data would most likely suffer.

4.

Using the average absolute error between the output of the network and the test values I get,

Sigma 0.1

Samples	Weight Decay	Mean Absolute Error
15	0	78.975
15	.2	78.587
15	.5	79.009
100	0	78.479
100	.2	78.209
100	.5	78.309

With a best at N = 100 and Weight decay of 0.2

Sigma 0.5

Samples	Weight Decay	Mean Absolute Error
15	0	78.722
15	.2	78.700
15	.5	78.704
100	0	78.452
100	.2	78.315
100	.5	78.249

With the best at N = 100 and Weight Decay of 0.5

Sigma 0.5

Samples	Weight Decay	Mean Absolute Error
15	0	78.546
15	.2	78.600
15	.5	78.486
100	0	78.377
100	.2	78.225
100	.5	78.295

With the best at N = 100 and Weight Decay of 0.2