```python
# Problem 3

import math
import numpy as np
from sklearn.model_selection import train_test_split

# Calculate MSE given y_pred and y
def MSE(y_pred, y):
      a = y.shape[0]
      return (1/(2 * a)) * np.sum(np.square(y_pred - y))

# Calculate the MSE given the prediction parameters and actual y value
def calculate_error(x, w, b, y):
      y_pred = x.dot(w) + b
      return MSE(y_pred, y)

# Loop through the samples based on the batch_size hyperparameter
def batch_loop(x, y, size):
      if(len(x) == len(y)):
            random_x = x[np.random.permutation(x.shape[0])]
            random_y = y[np.random.permutation(y.shape[0])]
            for i in np.arange(0, y.shape[0], size):
                  yield random_x[i:i + size], random_y[i:i + size]

# Perform gradient_descent using the prediction parameters, actual y,
learning rate, and regularized term
def gradient_descent(x, y, b, w, learn_rate, alpha = 0.1):
      a = y.shape[0]
      y_pred = x.dot(w) + b

      # Store the derivatives of w and b
      d_of_w = (1/a) * x.T.dot(y_pred - y)
      d_of_b = (1/a) * np.sum(y_pred - y)
      # Add the regularized term to the derivative of w
      d_of_w += ((alpha/a) * w)
      # Update parameters with gradient descent
      w = w - (learn_rate * d_of_w)
      b = b - (learn_rate * d_of_b)

      return w, b

# Perform SGD
def stochastic_gradient_descent(x, y, b, w, learn_rate: float,
num_of_epoch: int, batch_size: int, dataset_size: int, alpha):

      for n in range(num_of_epoch - 1):
            for mini_batch_x, mini_batch_y in batch_loop(x, y,
batch_size):
                  w, b = gradient_descent(mini_batch_x, mini_batch_y, b,
w, learn_rate, alpha)
```

```
    return w, b

# Find the lowest error by performing SGD
def find_lowest_error(x, y, learn_rate, num_of_epoch, batch_size,
alpha):
    dataset_size = len(y)
    m = np.expand_dims(a=y, axis=-1)
    w = np.random.rand(x.shape[1]) * np.sqrt(1/(x.shape[1] +
m.shape[1]))
    b = np.random.rand(m.shape[1])
    w_trained, b_trained = stochastic_gradient_descent(x, y, b, w,
learn_rate, num_of_epoch, batch_size, dataset_size, alpha)
    return w_trained, b_trained

# Implementation of grid_search to tune our hyperparameters by looping
through the various values we have for each
def grid_search():
    hyperparameters = {
        "learn_rate": [0.1, 0.01, 0.001, 0.0001],
        "num_of_epoch": [5, 10, 15, 20],
        "batch_size": [10, 20, 50, 100],
        "alpha": [0.75, 0.5, 0.25, 0.1]
    }
    for a in range(len(hyperparameters["num_of_epoch"])):
        for b in range(len(hyperparameters["batch_size"])):
            for c in range(len(hyperparameters["learn_rate"])):
                for d in range(len(hyperparameters["alpha"])):
                    yield hyperparameters["num_of_epoch"]
[a], hyperparameters["batch_size"][b], hyperparameters["learn_rate"]
[c], hyperparameters["alpha"][d]

# This function will train the age regressor based on various
hyperparameters provided in the grid_search() function
def train_age_regressor():

    # Load data
    starting_x_tr = np.reshape(np.load("age_regression_Xtr.npy"), (-
1, 48*48))
    starting_y_tr = np.load("age_regression_ytr.npy")
    x_te = np.reshape(np.load("age_regression_Xte.npy"), (-1, 48*48))
    y_te = np.load("age_regression_yte.npy")
    x_tr, x_val, y_tr, y_val = train_test_split(starting_x_tr,
starting_y_tr, train_size=0.8)

    # Initialize best hyperparameter values as the worst they could
be
    best_error = 1000000
    best_num_of_epoch = -1
    best_batch_size = -1
```

```
    best_learn_rate = -1
    best_alpha = -1

    # Loop through each combination of the hyperparameters in 
grid_search() to find the best combination to minimize MSE
    for num_of_epoch, batch_size, learn_rate, alpha in grid_search():

        w_trained, b_trained = find_lowest_error(x_tr, y_tr, 
learn_rate, num_of_epoch, batch_size, alpha)

        # Calculate the MSE from the validation dataset
        error = calculate_error(x_val, w_trained, b_trained, y_val)
        print("train/validation unregularized MSE: ", error)

        # Store the hyperparameters that led to reduced error in 
the following variables
        if error < best_error:
            best_error = error
            best_learn_rate = learn_rate
            best_num_of_epoch = num_of_epoch
            best_batch_size = batch_size
            best_alpha = alpha

    # Finally, calculate the error using the trained weights and 
biases
    error = calculate_error(x_te, w_trained, b_trained, y_te)
    print("\n")
    print("Results of training:")
    print("best error from validation dataset: ", best_error)
    print("best learning rate: ", best_learn_rate)
    print("best number of epochs: ", best_num_of_epoch)
    print("best batch size: ", best_batch_size)
    print("best reg term: ", best_alpha)
    print("unregularized MSE from test dataset: ", error)

    return w_trained, b_trained

def main():

    print("Problem 3 Output:")
    w_output, b_output = train_age_regressor()

if __name__ == '__main__':
    main()
```

```
Problem 3 Output:
train/validation unregularized MSE:  nan

c:\Users\rakes\anaconda3\Lib\site-packages\numpy\core\
fromnumeric.py:86: RuntimeWarning: overflow encountered in reduce
  return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
```

```
C:\Users\rakes\AppData\Local\Temp\ipykernel_11520\3979377061.py:36:
RuntimeWarning: invalid value encountered in subtract
  w = w - (learn_rate * d_of_w)
C:\Users\rakes\AppData\Local\Temp\ipykernel_11520\3979377061.py:37:
RuntimeWarning: invalid value encountered in subtract
  b = b - (learn_rate * d_of_b)
```

```
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  170.21064791764127
train/validation unregularized MSE:  135.47496519515798
train/validation unregularized MSE:  150.15603253858993
train/validation unregularized MSE:  144.32203983141477
train/validation unregularized MSE:  136.35303705553747
train/validation unregularized MSE:  136.89091361137616
train/validation unregularized MSE:  137.40048043523007
train/validation unregularized MSE:  134.83801409750794
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  137.1889942009483
train/validation unregularized MSE:  147.15633217719605
train/validation unregularized MSE:  133.48721249704104
train/validation unregularized MSE:  135.72377069380403
train/validation unregularized MSE:  136.49086007970536
train/validation unregularized MSE:  136.12489959989404
train/validation unregularized MSE:  135.82999072190552
train/validation unregularized MSE:  135.80759045601127
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
```

```
C:\Users\rakes\AppData\Local\Temp\ipykernel_11520\3979377061.py:10:
RuntimeWarning: overflow encountered in square
  return (1/(2 * a)) * np.sum(np.square(y_pred - y))
```

```
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  135.59989915424987
train/validation unregularized MSE:  135.38186982678764
train/validation unregularized MSE:  135.21050328093162
train/validation unregularized MSE:  135.74104086290356
train/validation unregularized MSE:  137.22904685927196
train/validation unregularized MSE:  136.1779473062348
train/validation unregularized MSE:  136.56620118664006
train/validation unregularized MSE:  136.90164715080323
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  8.752355009211058e+194
train/validation unregularized MSE:  7.300769064454156e+194
train/validation unregularized MSE:  7.94516611004122e+194
train/validation unregularized MSE:  8.312149399119605e+194
train/validation unregularized MSE:  136.12378814775857
train/validation unregularized MSE:  135.47968511186653
train/validation unregularized MSE:  135.64613975138806
train/validation unregularized MSE:  135.45422390889826
train/validation unregularized MSE:  136.16998735741439
train/validation unregularized MSE:  136.29357200099545
train/validation unregularized MSE:  136.47316249859082
train/validation unregularized MSE:  136.15550238088403
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  133.92381277137423
train/validation unregularized MSE:  147.47559982474502
train/validation unregularized MSE:  135.93511482822754
train/validation unregularized MSE:  152.69129790655583
train/validation unregularized MSE:  133.9195372038084
train/validation unregularized MSE:  134.37130363815083
train/validation unregularized MSE:  135.38429598578955
train/validation unregularized MSE:  135.77065957242195
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
```

```
train/validation unregularized MSE:  134.92615134781076
train/validation unregularized MSE:  132.49832879195498
train/validation unregularized MSE:  144.82989083349224
train/validation unregularized MSE:  134.69786031123263
train/validation unregularized MSE:  135.86181118116647
train/validation unregularized MSE:  135.7498864444108
train/validation unregularized MSE:  134.4781895597993
train/validation unregularized MSE:  135.9777643322185
train/validation unregularized MSE:  nan

C:\Users\rakes\AppData\Local\Temp\ipykernel_11520\3979377061.py:28:
RuntimeWarning: overflow encountered in add
  y_pred = x.dot(w) + b

train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  140.804908951711
train/validation unregularized MSE:  136.34494844656788
train/validation unregularized MSE:  135.1280745065044
train/validation unregularized MSE:  134.16497676651386
train/validation unregularized MSE:  136.4499166115252
train/validation unregularized MSE:  136.04411424209536
train/validation unregularized MSE:  136.27094595207345
train/validation unregularized MSE:  135.5147804150093
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  inf
train/validation unregularized MSE:  134.779622656524
train/validation unregularized MSE:  134.98979304035092
train/validation unregularized MSE:  133.74739702730355
train/validation unregularized MSE:  134.10154151546644
train/validation unregularized MSE:  136.94756522456964
train/validation unregularized MSE:  137.13128812204997
train/validation unregularized MSE:  136.12429545868616
train/validation unregularized MSE:  136.97662006236143
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
```

```
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  130.53625688633375
train/validation unregularized MSE:  129.75050048964476
train/validation unregularized MSE:  154.08044463941076
train/validation unregularized MSE:  133.36050261846475
train/validation unregularized MSE:  135.25848397683106
train/validation unregularized MSE:  133.5039486551229
train/validation unregularized MSE:  134.0113843516638
train/validation unregularized MSE:  134.57761664290004
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  131.06677406447113
train/validation unregularized MSE:  135.4909948104081
train/validation unregularized MSE:  141.96174458495932
train/validation unregularized MSE:  145.7917645609694
train/validation unregularized MSE:  136.58092989720325
train/validation unregularized MSE:  135.66351684090725
train/validation unregularized MSE:  134.7150429282501
train/validation unregularized MSE:  135.2281931603368
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  132.6614593410059
train/validation unregularized MSE:  134.6296907456717
train/validation unregularized MSE:  137.87658069446098
train/validation unregularized MSE:  134.82823754916683
train/validation unregularized MSE:  135.28296979197893
train/validation unregularized MSE:  136.2710570003861
train/validation unregularized MSE:  135.82295170775353
train/validation unregularized MSE:  136.67224377179915
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
```

```
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  135.34874356748114
train/validation unregularized MSE:  134.7489809490512
train/validation unregularized MSE:  134.48031380552717
train/validation unregularized MSE:  133.9500058447134
train/validation unregularized MSE:  136.09522081357562
train/validation unregularized MSE:  136.76843512676024
train/validation unregularized MSE:  136.66260649383534
train/validation unregularized MSE:  136.86499762023251
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  136.36327304526927
train/validation unregularized MSE:  135.7564715276333
train/validation unregularized MSE:  145.9410073456845
train/validation unregularized MSE:  126.77897172963536
train/validation unregularized MSE:  133.1187477090769
train/validation unregularized MSE:  135.98827572568706
train/validation unregularized MSE:  134.441325444631
train/validation unregularized MSE:  134.62321373502994
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  145.00975498709258
train/validation unregularized MSE:  131.65551295581588
train/validation unregularized MSE:  138.07213094585333
train/validation unregularized MSE:  132.40204116730118
train/validation unregularized MSE:  135.42505301291655
train/validation unregularized MSE:  134.99441770355858
train/validation unregularized MSE:  135.3449843332285
train/validation unregularized MSE:  135.49078806498025
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
```

```
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  133.81165885225005
train/validation unregularized MSE:  133.06975983411286
train/validation unregularized MSE:  133.68251944154463
train/validation unregularized MSE:  132.94202141239637
train/validation unregularized MSE:  135.8554379164131
train/validation unregularized MSE:  136.24333632551355
train/validation unregularized MSE:  136.1777849335946
train/validation unregularized MSE:  135.78908971926828
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  nan
train/validation unregularized MSE:  135.37394869615466
train/validation unregularized MSE:  135.9969684254965
train/validation unregularized MSE:  137.21319046108732
train/validation unregularized MSE:  134.94265993858565
train/validation unregularized MSE:  135.7630489204891
train/validation unregularized MSE:  135.75219972793278
train/validation unregularized MSE:  135.44818788652464
train/validation unregularized MSE:  135.8371500736761


Results of training:
best error from validation dataset:  126.77897172963536
best learning rate:  0.001
best number of epochs:  20
best batch size:  10
best reg term:  0.1
unregularized MSE from test dataset:  139.17092912962255
```

```python
# Problem 4c

import numpy as np

def update_b_mae(X, y, w, b):

  # Calculate the predictions.
  predictions = X.dot(w) + b

  # Calculate the signs of the errors.
  errors = np.sign(predictions - y)

  # Update the bias term.
  b -= np.mean(errors)
```

```python
    return b

def update_w_mae(X, y, w, b):

    # Calculate the predictions
    y_pred = X.dot(w) + b

    # Calculate the gradients
    grad_w = X.T.dot(np.sign(y_pred - y))

    # Update the weight vector
    w_new = w - grad_w

    return w_new

# Test both functions with initial values for X, y, w, and b
X = np.array([[1, 2], [3, 4], [5, 6]])
y = np.array([3, 7, 11])
w = np.array([1, 2])
b = 0
# Update both parameters.
b_new = update_b_mae(X, y, w, b)
w_new = update_w_mae(X, y, w, b)
print("Problem 4c Output:")
print(b_new)
print(w_new)
```

```
Problem 4c Output:
-1.0
[ -8 -10]
```

(1a) If n = # of iterations and we have m = # of observations, for each iteration there's a $\frac{1}{m}$ chance that an observation is sampled.

Probability that it isn't sampled is $1-\frac{1}{m}$

For n iterations: $\left(1-\frac{1}{m}\right)^n$

For one epoch: $\boxed{P_{never} = m\left(1-\frac{1}{m}\right)^n}$

(1b) With $P_{never} = m\left(1-\frac{1}{m}\right)^n$

~~We can directly substitute x for~~

We can do $\ln(P_{never}) = \ln\left(m\left(1-\frac{1}{m}\right)^n\right)$

$$\ln(P_{never}) = \ln m + n\ln\left(1-\frac{1}{m}\right) \rightarrow \ln(1-x) \approx -x \text{ where } x \text{ is small}$$

$$\ln(P_{never}) = \ln m - n\cdot\frac{1}{m}$$

$$P_{never} = e^{\left(\ln m - \frac{n}{m}\right)}$$

$$\boxed{P_{never} = \frac{1}{e} \text{ if } m \text{ is large} \qquad \frac{1}{e} \approx 0.3679}$$

(1c) The variance of the gradient computed over a minibatch requires a minibatch size.

We'll call this n = minibatch size

$$Var(\nabla f(\text{minibatch})) = \frac{Var(\nabla f(x^{(i)}, y^{(i)}))}{n}$$

$$\boxed{Var(\nabla f(\text{minibatch})) = \frac{\sigma^2}{n}}$$

(1d) If n = minibatch size, and we multiply the size by k, the new variance will be:

$$\cancel{Var(\nabla f(\text{minibatch})) = \frac{\sigma^2}{nk}}$$

$$\boxed{Var(\nabla f(\text{minibatch})) = \frac{\sigma^2}{nk}}$$

(1e) The claim is incorrect because simply multiplying the learning rate by k will not keep the gradient noise constant since it will actually increase the gradient noise.

In order to change the learning rate to keep the noise constant, the learning rate must be divided by $\sqrt{k}$ to calculate the new learning rate.

$$\boxed{\text{So, new learning rate} = \frac{\text{old learning rate}}{\sqrt{k}}}$$

Need to use $J(\theta) = \frac{1}{4} \sum_{x \in X} (f^*(x) - f(x;\theta))^2$

(2) $J(w_1, w_2, b) = \frac{1}{4} \sum (x^{(i)T} w + b - y^{(i)})^2$

For XOR problem: X could be $\overset{x_1}{[0,0]^T}, \overset{x_2}{[1,1]^T}, \overset{x_3}{[0,1]^T}, \overset{x_4}{[1,0]^T}$

y could be $\underset{y_1}{0}, \underset{y_2}{0}, \underset{y_3}{1}, \underset{y_4}{1}$

and use $w^T = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ for the equation

Substitute X, y, and w in the above formula to get:

$$J(w_1, w_2, b) = \frac{1}{4}\left((x_1 w^T + b - y_1)^2 + (x_2 w^T + b - y_2)^2 + (x_3 w^T + b - y_3)^2 + (x_4 w^T + b - y_4)^2\right)$$

$$= \frac{1}{4}\left((0 + b - 0)^2 + (w_1 + w_2 + b - 0)^2 + (w_1 + b - 1)^2 + (w_2 + b - 1)^2\right)$$

$$J(w_1, w_2, b) = \frac{1}{4}\left(b^2 + (w_1 + w_2 + b)^2 + (w_1 + b - 1)^2 + (w_2 + b - 1)^2\right)$$

Now, Find the derivatives:

$$\frac{dJ}{dw_1} = \frac{1}{4}\left(0 + 2(w_1 + w_2 + b) + 2(w_1 + b - 1) + 0\right)$$

$$= \frac{1}{2}(2w_1 + w_2 + 2b - 1)$$

$$\frac{dJ}{dw_2} = \frac{1}{4}\left(0 + 2(w_1 + w_2 + b) + 0 + 2(w_2 + b - 1)\right)$$

$$= \frac{1}{2}(w_1 + 2w_2 + 2b - 1)$$

$$\frac{dJ}{db} = \frac{1}{4}\left(2b + 2(w_1 + w_2 + b) + 2(w_1 + b - 1) + 2(w_2 + b - 1)\right)$$

$$= \frac{1}{2}(2w_1 + 2w_2 + 4b - 2)$$

Each derivative can be set to equal 0, and therefore, equal each other.

Continue on next page

(2) (continued)

Now, we have a system of equations with:

$$\frac{1}{2}(2w_1 + w_2 + 2b - 1) = 0$$
$$\frac{1}{2}(w_1 + 2w_2 + 2b - 1) = 0$$
$$\frac{1}{2}(2w_1 + 2w_2 + 4b - 2) = 0$$

So, we have $\frac{1}{2}(2w_1 + 2w_2 + 4b - 2) = 0$

$$= w_1 + w_2 + 2b - 1 = 0$$

We can take $\frac{1}{2}(2w_1 + w_2 + 2b - 1) = 0$

$$= 2w_1 + w_2 + 2b - 1 = 0$$

If we subtract the first from the second, we get:

$$w_1 = 0$$

If we subtract $\frac{1}{2}(2w_1 + w_2 + 2b - 1)$ from $\frac{1}{2}(w_1 + 2w_2 + 2b - 1)$, we will get $w_1 = w_2$.

If we substitute $w_1 = w_2 = 0$ to the equation $\frac{1}{2}(2w_1 + 2w_2 + 4b - 2) = 0$, we will get that $b = \frac{1}{2}$

So $w_1 = 0$, $w_2 = 0$, $b = 0.5$

(4a) $f_{MAE}(w,b) = \frac{1}{n} \sum_{i=1}^{n} |x^{(i)T} w + b - y^{(i)}|$

$\nabla_b f_{MAE}(w,b) = \nabla_b \left(\frac{1}{n} \sum_{i=1}^{n} |x^T w + b - y| \right)$

The gradient with respect to b will indicate how much the bias must be increased or decreased to bring $\hat{y}$ closer to y where $\hat{y} = x^T w + b$.

When $\nabla_b f_{MAE}(w,b) > 0$, $n_+ > n_-$

When $\nabla_b f_{MAE}(w,b) < 0$, $n_- > n_+$

When $\nabla_b f_{MAE}(w,b) = 0$, $n_+ = n_-$

(4b) $\nabla_w f_{MAE}(w,b) = \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} |x^T w + b - y| \right)$

In this case, the expression should derive to: $\nabla_w f_{MAE}(w,b) = \frac{1}{n} \sum_{i=1}^{n} x^T | y - x^T w + b |$

$$\boxed{\nabla_w f_{MAE}(w,b) = \frac{x^T}{n} (n_+ - n_-)}$$

(4c) Can be seen in Jupyter Notebook.

(5.) We're given $P(y|x) = \mathcal{N}(\mu = x^T w), \sigma^2)$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - x^T w)^2}{2\sigma^2}\right)$$

Need to find $P(D \mid \mu = x^T w, \sigma^2)$ where

$D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$. So, this should be

$$P(D \mid \mu = x^T w, \sigma^2) = \log \prod_{i=1}^n P(y^{(i)} \mid x^{(i)}, w, \sigma^2)$$

$$= \sum_{i=1}^n \log P(y^{(i)} \mid x^{(i)}, w, \sigma^2)$$

$$= \sum_{i=1}^n \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - x^T w)^2}{2\sigma^2}\right)\right)$$

$$= \sum_{i=1}^n -\frac{(y - x^T w)^2}{2\sigma^2} + \log\left(\frac{1}{\sqrt{2\pi\sigma}}\right)$$

$$= \frac{-1}{2\sigma^2} \sum_{i=1}^n (y - x^T w)^2 - \log \sigma + C$$

Now, we can compute the gradient of $P(D \mid \mu = x^T w, \sigma^2)$ and set it equal to 0 after

$$\nabla_w P(D \mid x^T w, \sigma^2) = \nabla_w \left(\frac{-1}{2\sigma^2} \sum_{i=1} (y - x^T w)^2 - \log \sigma + C\right)$$

$$\nabla_w P(D \mid x^T w, \sigma^2) = \frac{-1}{2\sigma^2} \sum_{i=1}^n -2x(y - x^T w)^2$$

$$\nabla_w P(D \mid x^T w, \sigma^2) = \frac{1}{\sigma^2} \sum_{i=1}^n x(y - x^T w)$$

Continue on next page.

(5) (continued)

$$\nabla_w P(D \mid x^T w, \sigma^2) = \frac{1}{\sigma^2} \sum_{i=1}^{n} x(y - x^T w) = 0$$

$$0 = \sum_{i=1}^{n} xy - xx^T w$$

$$\sum_{i=1}^{n} xx^T w = \sum_{i=1}^{n} xy$$

$$\boxed{w = \left(\sum_{i=1}^{n} xx^T \right)^{-1} \left(\sum_{i=1}^{n} xy \right)}$$

Now, we compute $\nabla_\sigma P(D \mid x^T w, \sigma^2) = \nabla_\sigma \left(\frac{-1}{2\sigma^2} \sum_{i=1}^{n} (y - x^T w)^2 - \log \sigma + c \right)$

$$0 = \frac{-1}{\sigma^3} \sum_{i=1}^{n} (y - x^T w)^2 + \frac{n}{\sigma}$$

$$\frac{-n}{\sigma} = \frac{-1}{\sigma^3} \sum_{i=1}^{n} (y - x^T w)^2$$

$$\boxed{\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x^T w - y)^2}$$