# ML 1

2. • The reason we subtract the mean before dividing is that we are trying to get a "standard score" or "z-score" of the data. This transformation makes the mean = 0 and the std = 1

→ This wouldn't be achieved if we divided by the standard deviation first

→ Consider:

$a = [105, 101, 95]$

$$\frac{a - \text{mean}(a)}{\text{std}(a)} = [0.7259 \quad 0.4148 \quad -1.106]$$

$$\frac{a}{\text{std}(a)} - \text{mean}(a) = [-67.2470, -67.5581, -69.1135]$$

Not centered around zero

2. We use the pre-computed mean of the training set, to normalize the test set for 2 reasons:
   1. The training set is much larger and thus the sample mean and sample standard deviation are more representative of the population
   2. It is generally best to not derive anything of your model from the test set, so that bench marking to other algorithms is much easier

3 ai) The derivative of a function and the derivative of the log of a function are zero at the same place → Thus minimizing one is like minimizing the other

aii) $$f(x) = \log\left[\prod_{i=1}^{m} \sigma(a_i^T x)^{b_i} (1-\sigma(a_i^T x))^{1-b_i}\right]$$

Log rules: 1) $\log(a \cdot b) = \log(a) + \log(b)$

2) $\log(a^b) = b\log(a)$

$$f(x) = \sum_{i=1}^{m} \log\left(\sigma(a_i^T x)^{b_i} (1-\sigma(a_i^T x))^{1-b_i}\right)$$

$$\sum_{i=1}^{m} \left[\log\left(\sigma(a_i^T x)^{b_i}\right) + \log\left(1-\sigma(a_i^T x)\right)^{1-b_i}\right]$$

$$f(x) = \sum_{i=1}^{m} \left[b_i \log(\sigma(a_i^T x)) + (1-b_i)\log\left(1-\sigma(a_i^T x)\right)\right]$$

$$\frac{\partial}{\partial x_k}\left(a_i^T x\right) = a_i^T \cdot \partial$$

$$f(x) = \sum_{i=1}^{m}\left[b_i \log\left(\sigma\left(a_i^T x\right)\right) + (1-b_i)\log\left(1-\sigma\left(a_i^T x\right)\right)\right]$$

Note: same as homework 1, except missing $\frac{1}{m}$ d -

$$g(h) = b\log(h) + (1-b)\log(1-h)$$

$$h(z) = \frac{1}{1+e^{-z}} \qquad z(x) = a^T x$$

$$\frac{dg}{dx} = \frac{dg}{dh} \cdot \frac{dh}{dz} \cdot \frac{dz}{dx}$$

$$\frac{dg}{dh} = \frac{b}{h} - \frac{1-b}{1-h} = \frac{b(1-h)}{h(1-h)} - \frac{h(1-b)}{h(1-h)} = \frac{b-bh-h+hb}{h(1-h)} = \frac{b-h}{h(1-h)}$$

$$\frac{dh}{dz} = h(z)\left(1-h(z)\right)$$

$$\frac{dz}{dx} = a^T \qquad \frac{dg}{dx} = \frac{b-h}{\cancel{h(1-h)}} \cdot \cancel{h(1-h)} \cdot a^T$$

$$= \left(b_i - \sigma\left(a_i^T x\right)\right) a_i^T \qquad m \times n$$

$$\sum_{i=1}^{m}\left(b_i - \sigma\left(a_i^T x\right)\right) a_i^T$$

$$\nabla f(x) = A^T\left(b - \sigma(Ax)\right)$$

$$\nabla f(x) = A^T(b - \sigma(Ax))$$

$$= A^T b - A^T \sigma(Ax)$$

$$\nabla^2 f(x) = A A^T \sigma(Ax)(1 - \sigma(Ax))$$

- Since Hessian for HW1 is convex then this should be concave

4b iv) Overall, the BTLS version is preferred even though it didn't perform much better.

- I do not have to worry about picking a stepsize as much
- It converges in less iterations
- Although each iteration of BTLS took a longer time

4 b v). Suprisingly, the linear regression model performed much better than the logistic regression by a large margin.

- This is suprising since logistic regression was much harder to implement.
- I would definitely choose linear regression model for this scenario, since I got much better results and it was easier. However, I would potentially use logistic regression if I had to deal with more classes (All numbers 1-9). Assuming I fixed the bugs...