APML - Ex 1

Rhea Chowers, 204150643

November 21, 2019

1 Theoretical Questions

1. The function defined will not define a larger hypothesis class when incorperated into a network. Assume the parametrized relu comes after some linear layer. We'll show that the bias term of the linear layer controls the relu's parameter, making it unnecessary: let $t = Wx + b$ since we're after a linear layer. Thus

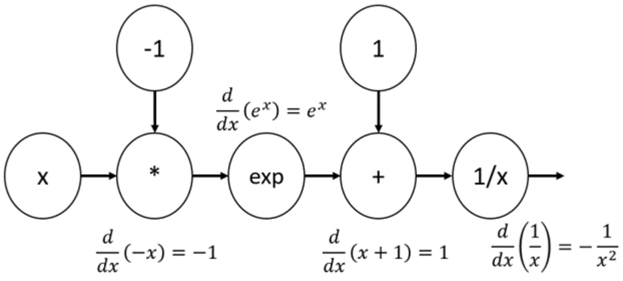
$$f_i(o;t) = \max t, o_i = \max Wx + b, o_i = \max Wx + b - o_i, 0 = \max Wx + b', 0$$

for $b' = b - o_i$. The linear layer can learn the parameter b' instead of b, making the combination of the linear layer followed by a relu as expressive as a linear layer followed by a parameterized relu. Thus parameterizing the relu function increases allows the network to define a larger hypothesis class only when there are no layers with added bias before it.

2. Derivative of the function:

$$\frac{d}{dx}\sigma(x) = -\left(\frac{d}{dx}\left(1+e^{-x}\right)\right)\cdot\left(\frac{1}{1+e^{-x}}\right)^2 = e^{-x}\left(\frac{1}{1+e^{-x}}\right)^2 = \sigma(x)\cdot\frac{e^{-x}}{1+e^{-x}} = \sigma(x)\cdot\left(\frac{1+e^{-x}-1}{1+e^{-x}}\right) = \sigma(x)\cdot(1-\sigma(x))$$

The computational graph:

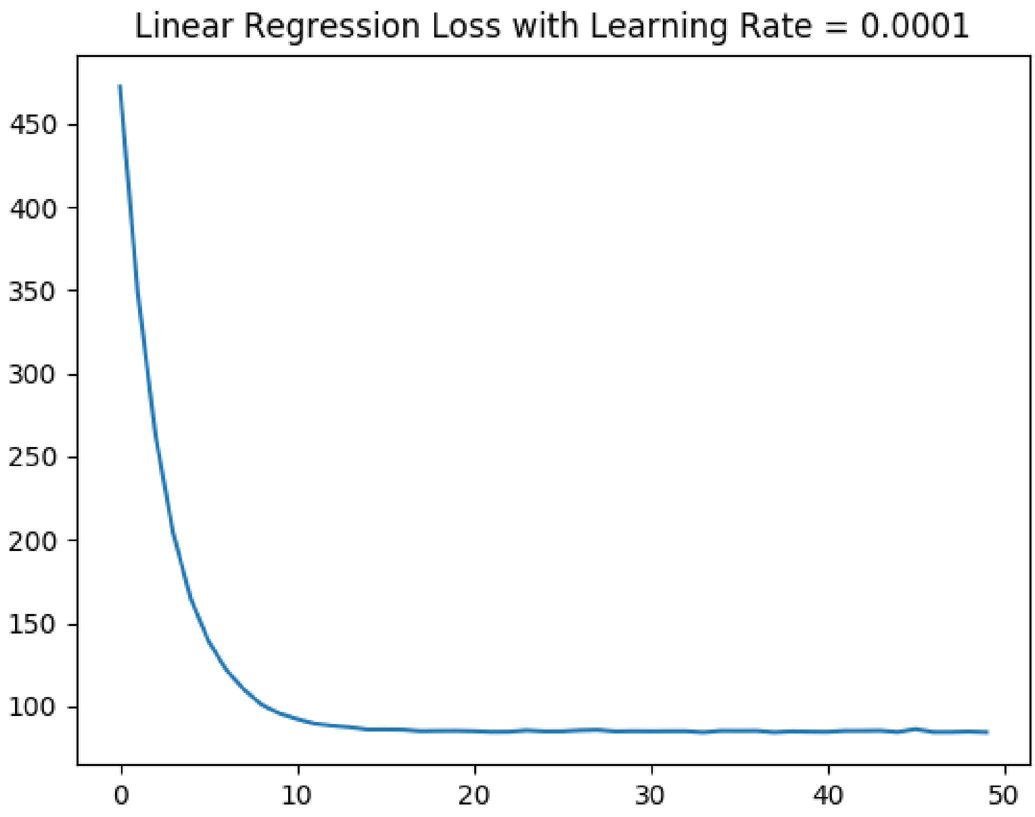


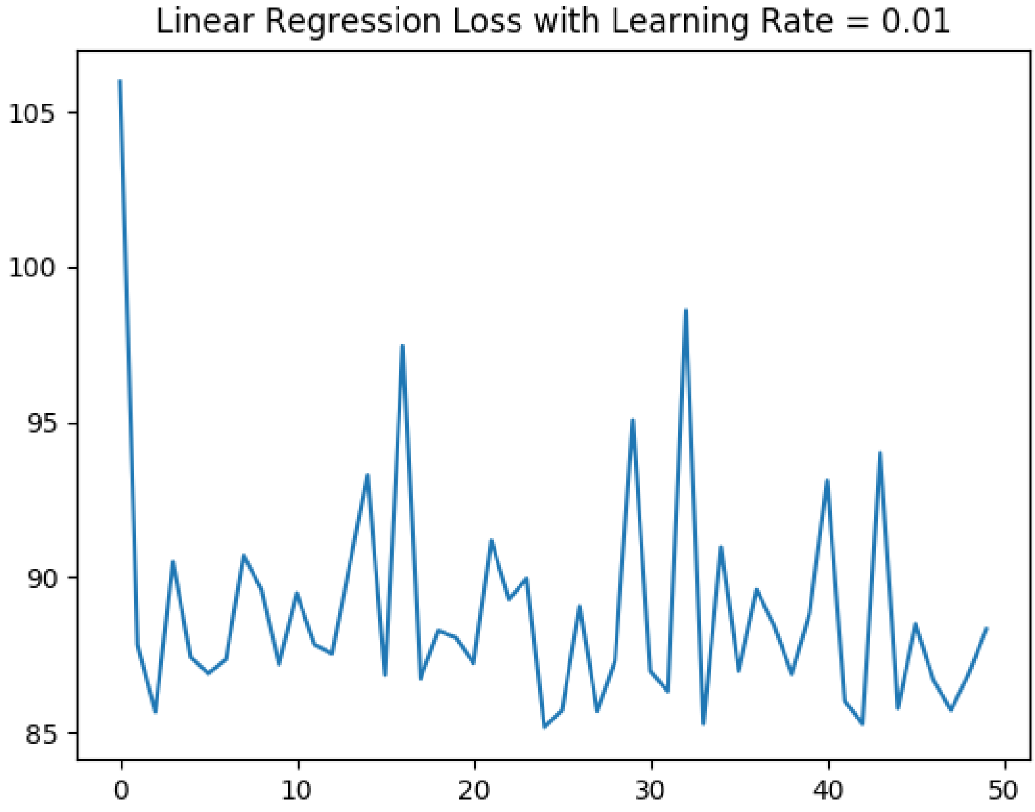
-1
1
$\frac{d}{dx}(e^x) = e^x$
x
*
exp
+
1/x
$\frac{d}{dx}(-x) = -1$
$\frac{d}{dx}(x+1) = 1$
$\frac{d}{dx}\left(\frac{1}{x}\right) = -\frac{1}{x^2}$

$$\frac{d}{dx}(\sigma(x)) = -1\cdot e^{-x}\cdot 1\cdot\left(-\frac{1}{(1+e^{-x})^2}\right) = \sigma(x)\big(1-\sigma(x)\big)$$

2 Linear Regression

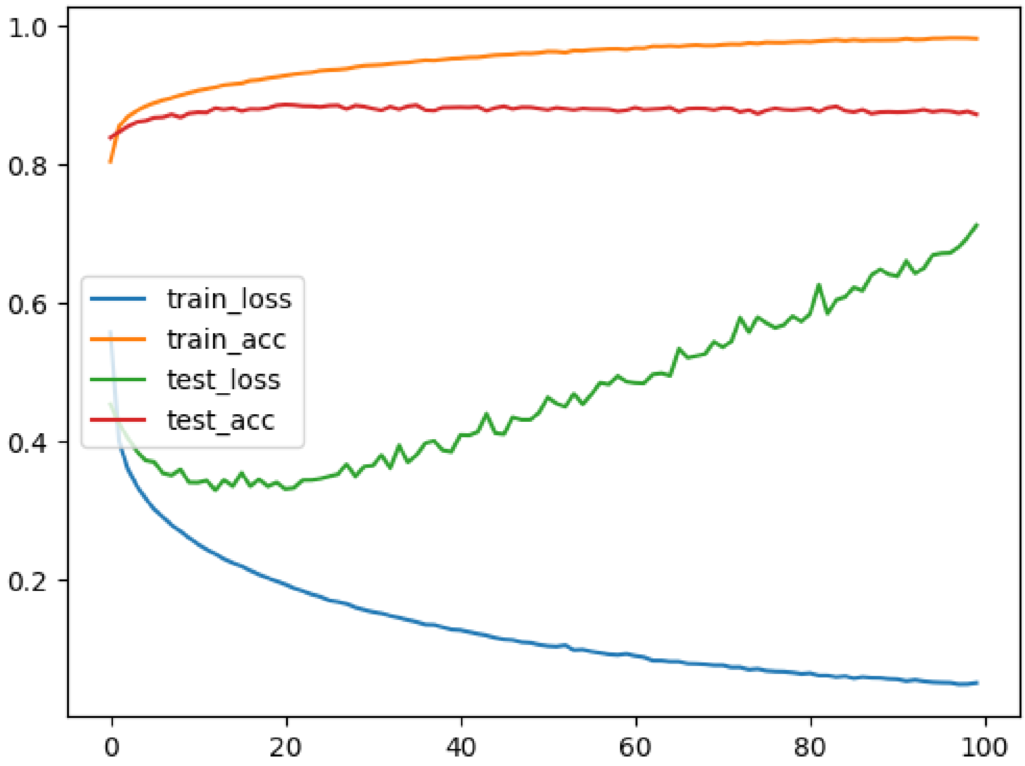
In the following graph we can see the loss at the end of each epoch for a learning rate of lr=0.0001. This creates a smooth curve. In the following graph we can see what happens with a significantly higher learning rate of 0.01:


Linear Regression Loss with Learning Rate = 0.0001
450
400
350
300
250
200
150
100
0
10
20
30
40
50

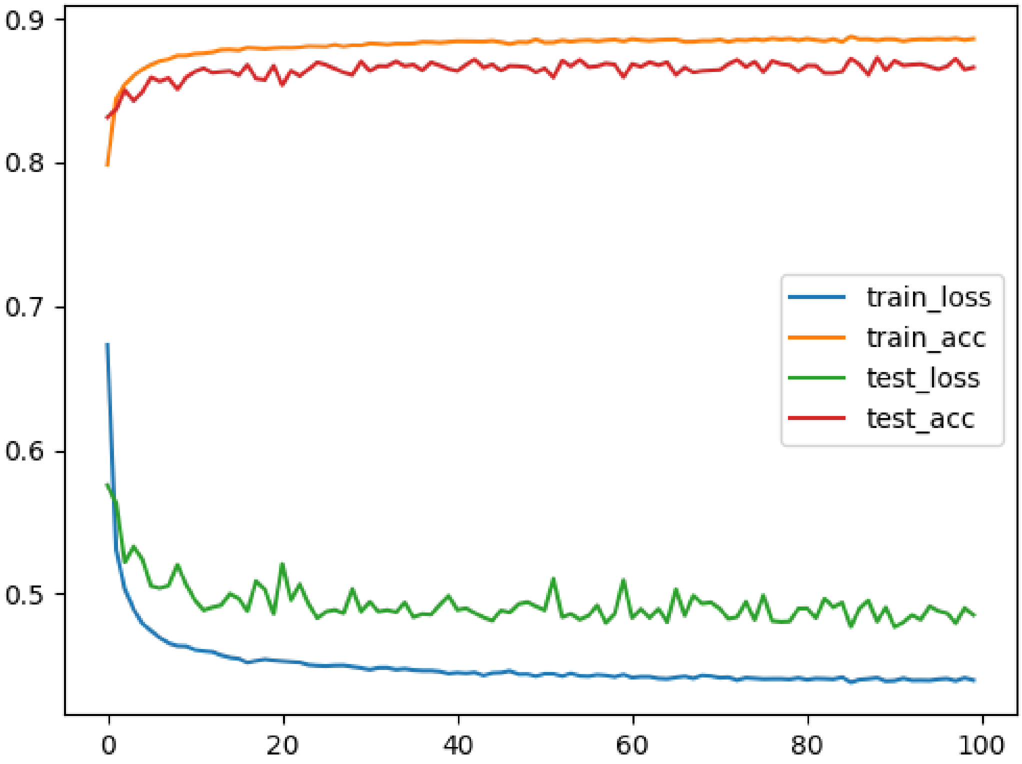


3 MLP and ConvNet

When training the MLP without regularization, we can see that after an initial significant drop in the test loss, we begin to overfit - the training loss continues to decrease but the test loss increases significantly. Thus we can identify a “stage” in learning process of the beginning of overfitting at around 15 epochs.

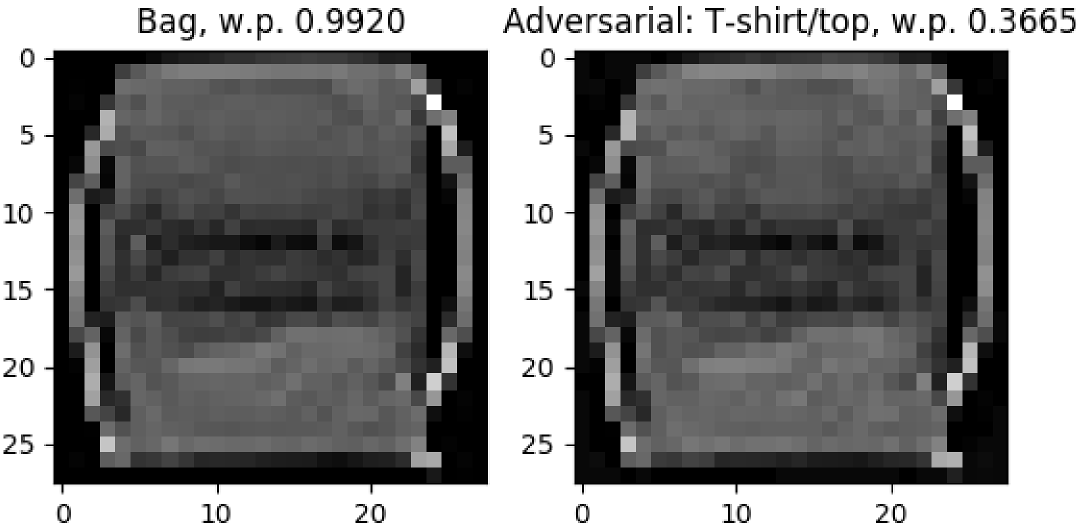


After applying weight regularization to the network, we can see that overfitting is prevented and the test loss and accuracy circle a constant low number:



4 Adversarial Image

In order to create an adversarial image, I took the trained model and fed it images that it classifies correctly. After getting the loss, I calculated its gradient with **respect to the input** (and not the network weights) and then updated the images using the fast gradient sign method. Since the MLP is composed of linear and activational (relu) layers, updating the image to be $image = image + \epsilon \cdot sign\left(\frac{dLoss}{dImage}\right)$ we will 'overactivate' some of the inner layers and destroy the prediction. For example a small change to the bag, almost non-noticeable caused a prediction change (and with fewer confidence):


Bag, w.p. 0.9920
Adversarial: T-shirt/top, w.p. 0.3665
0
5
10
15
20
25
0
10
20