Homework 1

Guilherme Albertini

September 22, 2022

Theory

Let $Linear_1 \to f \to Linear_2 \to g$ be a be a two-layer neural net architecture whereby $Linear_i(x) = \boldsymbol{W}^{(i)}\boldsymbol{x} + \boldsymbol{b}^{(i)}$ is the i^{th} affine transformation and f, g are element-wise nonlinear activation functions. When an input $\boldsymbol{x} \in \mathbb{R}^n$ is fed into the network, $\hat{\boldsymbol{y}} \in \mathbb{R}^K$ is obtained as output.

Problem 1: Regression Task

We would like to perform regression task. We choose $f(\cdot) = 5(\cdot)^+ = 5ReLU(\cdot)$ and g to be the identity function. To train, we choose MSE loss function, $\ell_{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = ||(\hat{\boldsymbol{y}} - \boldsymbol{y})||^2$.

1. Name and mathematically describe the 5 programming steps you would take to train this model with PyTorch using SGD on a single batch of data.

   (a) First, we initialize the parameters to random values (and can optionally save the original parameters in another variable), and tell PyTorch that we want to track their gradients.

   (b) Second, we calculate the predicted values and visualise the actual and predicted values.

   (c) Third,we calculate the average loss of the model using backpropagation, seeking to reduce MSE of our existing model. To do that, we compute the current gradients to approximate how to change update the existing parameters.

   (d) Fourth, we update the paramaters via the chosen learning rate.

   (e) Finally, iterate until early stopping or another criterion (i.e certain number of epochs, target MSE threshold, etc.) is applied. IIt is best to monitor the training and validation losses and chosen metrics when deciding to stop.

2. For a single data point (x, y), write down all inputs and outputs for forward pass of each layer. You can only use variables and mechanics specified prior in your answer.

3. $Linear_1$

4. Six

For the negative log-liklihood:

$$\frac{\partial L(c\hat{w})}{\partial c} = \frac{\partial(\log(1+\exp(-cy_i\hat{w}^T x_i)))}{\partial c} = \frac{-y_i\hat{w}^T x}{1+\exp(cy_i\hat{w}^T x_i)}$$

Both $y_i\hat{w}^T x_i$ and $1+\exp(-cy_i\hat{w}^T x_i)$ are non-negative terms: if all points are classified correctly, then this means that $-y_i w^T x_i$ is negative which in turn implies the partial derivative is less than 0, decreasing without bound as we increase c (with a non-negative c) – we never get an optimizer for this case as there is always another $c > 1$ (and thus $c\hat{w}$) that forcing the (non-negative) liklihood to increase without bound.

Problem 2

Given...

Proof. Let $\epsilon > 0$. If you have a shorter statement that you still want centered, use two $$ on either side.

$$\exists \text{ some } \delta > 0 \mid ...$$

□

Problem 3

Proof. □

Section 2.2

Problem 6

Blah

Problem 7

Blah

Problem 10

Blah