CS 615 - Deep Learning

Assignment 2 - Objectives, Gradients, and Backpropagation
Winter 2024

Alec Peterson (ap3842@drexel.edu)

1 Theory

1. (10 points) Given $H = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ as an input, compute the gradients of the output with respect to this input for the following activation layers. Show your answer in **tensor form** by having a Jacobian matrix for each observation.

   (a) A ReLU layer

$$Gradient = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix}$$

   (b) A Softmax layer

$$Gradient = \begin{bmatrix} \begin{bmatrix} 0.0819 & -0.0599 & -0.4425 \\ -0.0081 & 0.1848 & -0.4425 \\ -0.0081 & -0.0599 & 0.2227 \end{bmatrix} \\ \\ \begin{bmatrix} 0.0819 & -0.0599 & -0.4425 \\ -0.0081 & 0.1848 & -0.4425 \\ -0.0081 & -0.0599 & 0.2227 \end{bmatrix} \end{bmatrix}$$

   (c) A Logistic Sigmoid Layer

$$Gradient = \begin{bmatrix} \begin{bmatrix} 0.1966 & 0 & 0 \\ 0 & 0.1050 & 0 \\ 0 & 0 & 0.0452 \end{bmatrix} \\ \\ \begin{bmatrix} 0.0177 & 0 & 0 \\ 0 & 0.0066 & 0 \\ 0 & 0 & 0.0025 \end{bmatrix} \end{bmatrix}$$

   (d) A Tanh Layer

$$Gradient = \begin{bmatrix} \begin{bmatrix} 0.18157 & 0 & 0 \\ 0 & 0.03468 & 0 \\ 0 & 0 & 0.00492 \end{bmatrix} \\ \\ \begin{bmatrix} 0.000670 & 0 & 0 \\ 0 & 0.000091 & 0 \\ 0 & 0 & 0.000012 \end{bmatrix} \end{bmatrix}$$

(e) A Linear Layer

$$Gradient = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix}$$

2. (2 points) Given $H = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ as an input, compute the gradient of the output a fully connected layer with regards to this input if the fully connected layer has weights of $W = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ as biases $b = \begin{bmatrix} -1 & 2 \end{bmatrix}$.

$$Gradient = \begin{bmatrix} \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \\ \\ \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \end{bmatrix}$$

3. (2 points) Given target values of $Y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and estimated values of $\hat{Y} = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$ compute the loss for:

(a) A squared error objective function

$J = mean((Y - \hat{Y}) * (Y - \hat{Y})) = 0.265$ (rounded)

(b) A log loss (negative log likelihood) objective function)

$J = mean(-(Y log(\hat{Y}) + (1 - Y) log(1 - \hat{Y})) = 0.7136$ (rounded)

4. (1 point) Given target *distributions* of $Y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and estimated distributions of $\hat{Y} = \begin{bmatrix} 0.2 & 0.2 & 0.6 \\ 0.2 & 0.7 & 0.1 \end{bmatrix}$ compute the cross entropy loss.

$J = mean(\sum_{k=1}^{K} y_k ln(\hat{y}_k)) = 0.6554$ (rounded)

5. (4 points) Given target values of $Y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and estimated values of $\hat{Y} = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$ compute the gradient of the following objective functions with regards to their input, $\hat{Y}$:

(a) A squared error objective function

$$Gradient = -2(Y - \hat{Y}) = \begin{bmatrix} 0.4 \\ -1.4 \end{bmatrix}$$

(b) A log loss (negative log likelihood) objective function)

$$Gradient = \frac{Y - \hat{Y}}{\hat{Y}(1-\hat{Y})} = \begin{bmatrix} -1.25 \\ 3.33 \end{bmatrix}$$

6. (1 point) Given target *distributions* of $Y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and estimated distributions of $\hat{Y} = \begin{bmatrix} 0.2 & 0.2 & 0.6 \\ 0.2 & 0.7 & 0.1 \end{bmatrix}$ compute the gradient of the cross entropy loss function, with regard to the input distributions $\hat{Y}$.

$$Gradient = \frac{-Y}{\hat{Y}} = \begin{bmatrix} -5, 0, 0 \\ 0, -1.429, 0 \end{bmatrix}$$

2 Update Your Codebase

See submitted code.

3 Forwards-Backwards Propagate a Dataset

In HW1 you implemented forwards propagation for the Kid Creative dataset with the following architecture (note that I have added on a LogLoss layer):

Input→FC (1 output)→Logistic Sigmoid→LogLoss

Now let's do forwards-backwards propagation. Using the code shown in the *Objectives and Gradients* slides, perform one forwards-backwards pass. As you go backwards through each layer, report the *mean* gradient, as averaged over the observations. For this particular architecture you should report the mean gradient coming backwards out of:

See submitted code.

1. Log Loss
   Average gradient across observations= -1.2569

2. Logistic Sigmoid Layer
   Average gradient across observations= -0.3142

3. Fully-Connected Layer
   (Result is a 16-element vector)

$$\textbf{Average gradient across observations} = \begin{bmatrix} 5.2147 * 10^{-5} \\ -1.3846 * 10^{-4} \\ 3.1415 * 10^{-4} \\ 1.2422 * 10^{-4} \\ 2.2199 * 10^{-4} \\ 2.5619 * 10^{-4} \\ 1.9717 * 10^{-4} \\ 9.7056 * 10^{-5} \\ 6.4876 * 10^{-5} \\ -2.4394 * 10^{-5} \\ 5.0782 * 10^{-5} \\ -1.1640 * 10^{-4} \\ 1.8573 * 10^{-4} \\ -2.3763 * 10^{-4} \\ 2.9701 * 10^{-4} \\ -1.0713 * 10^{-4} \end{bmatrix}$$