CS 4644/7643: Deep Learning
Spring 2024
HW2 Solutions

Jadon Co

Due: 11:59pm, Feb 19, 2024

1 Collaborators [0.5 points]

I worked with Elias Cho and Keigo Hayashi for this assignment.

2 Activation Function

1. **[2 points]**

   Given the context and definition above, consider a zero-centered activation function $g(x)$. Show that if its derivative $g'(x)$ exists, then $g'(0) = 0$.

   So, $g(x)$ is a zero-centered activation function. If we know that the derivative of this function exists, $g'(x)$, then we must show that g'(0) = 0.

To do this, we will use the definition of a derivative at the point x = 0:

$$g'(x) = \lim_{h \to 0} \frac{g(x+h) - g(x)}{h}$$

Then, we are analyzing the gradient at point x = 0, then we can substitute it into the limit equation:

$$g'(0) = \lim_{h \to 0} \frac{g(h) - g(0)}{h}$$

Additionally, we also are given that the function is symmetric about the y-axis, so we can analyze the limit as h approaches 0 from the left (negative) side:

$$g'(0) = \lim_{-h \to 0} \frac{g(-h) - g(0)}{-h}$$

Next, we know that in order for the function to be both zero-centered (output centered around 0) and symmetric (about the y-axis), the value of g(0) must be 0. Moreover, we can substitute g(-x) = -g(x), as that is the property:

$$g'(0) = \lim_{h \to 0} \frac{g(h)}{h} = \lim_{-h \to 0} \frac{-g(h)}{-h}$$

Finally, for the derivative to exist and be consistent from both the positive and negative direction, the slope or rate of change must be flat and approaching 0. The only conclusion that makes this statement work is that the slop is 0 at point x = 0 or that g'(0) = 0.

3 Gradient Descent

1. **[3 points]**

   Given:

$$f(w^t) + \langle w - w^t, \nabla f(w^t) \rangle + \frac{\lambda}{2} \|w - w^t\|^2$$

   First, we take the gradient with respect to dw:

$$\frac{d}{dw}(f(w^t) + \langle w - w^t, \nabla f(w^t) \rangle + \frac{\lambda}{2} \|w - w^t\|^2)$$

$$\frac{d}{dw}(f(w^t)) = 0$$

$$\frac{d}{dw}(\langle w - w^t, \nabla f(w^t) \rangle) = \nabla f(w^t)$$

$$\frac{d}{dw}(\frac{\lambda}{2} \|w - w^t\|^2) = \lambda(w - w^t)$$

   So, we get the resulting equation:

$$0 + \nabla f(w^t) + \lambda(w - w^t) = 0$$

   Then, we rearrange the equation in terms of w (which is the same as w* in the question):

$$w* = w^t - \frac{1}{\lambda} \nabla f(w^t)$$

   This equation shows us that the gradient descent update rule can be viewed as a minimizing Taylor approximation to a given loss function f(w) with the

$$l_2$$

regularization term. This term ensures that the solution remains fairly close to the estimated value while addressing problems like the affine approx. being unbounded. Additionally, having the

$$l_2$$

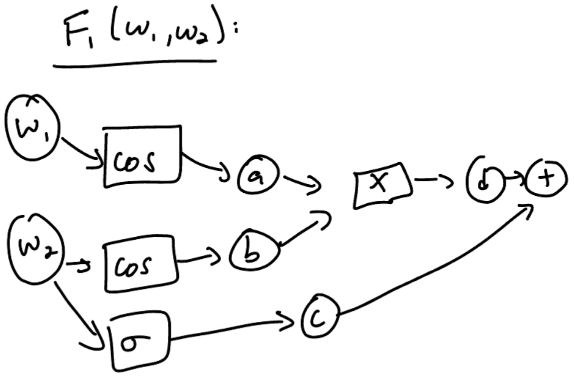
term is important for controlling the step size for each update.

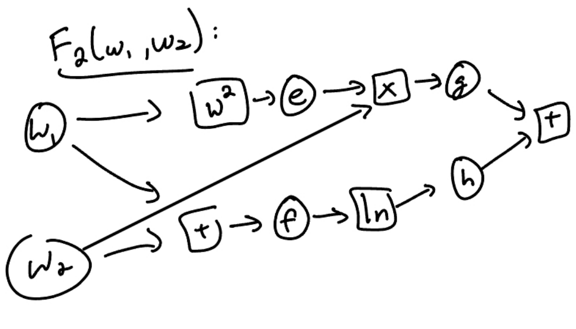
When comparing the solved equation of w* with that from gradient descent (given in the beginning of the problem), we see that:

$$\eta = \frac{1}{\lambda}$$

4 Automatic Differentiation

1. **[4 points]**





$$f(1,2) = (cos(1)cos(2) + o(2)) + (ln(1+2) + 2) = (0.66, 3.10)$$

2.

$$\frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} \end{bmatrix} = \begin{bmatrix} \frac{f_1(w_1+\Delta w,2)-f_1(w_1,w_2)}{\Delta w} & \frac{f_1(w_1,w_2+\Delta w)-f_1(w_1,w_2)}{\Delta w} \\ \frac{f_2(w_1+\Delta w,w_2)-f_2(w_1,w_2)}{\Delta w} & \frac{f_2(w_1,w_2+\Delta w)-f_2(w_1,w_2)}{\Delta w} \end{bmatrix} = \begin{bmatrix} 0.35 & -0.387 \\ 4.35 & 1.33 \end{bmatrix}$$

3. Forward mode differentiation, Didn't complete

4. Backward mode differentiation, Didn't complete

5 Convolutions

[5 points]

1. So, our goal is to prove that any circulant matrix is commutative with a shift matrix. To do this, we must first establish some terms:

   Let C_a be a circulant matrix and S be a shift matrix (thus, we are proving that CS = SC)

   The circulant matrix is a convolution that transforms the vector it is multiplied by. Also, the shift matrix S does an operation that shifts all elements of a vector to the right by a single position and then the last element of the vector becomes the first element.

   Given these definitions, let's consider C_aS and SC_a.

   1. C_aS means that first we are going to apply the shift to the vector and perform convolution on it after. 2. SC_a means that we are going to first do the convolution and then shift the results.

   So, if we have a vector a = $(a_0, a_1..., a_n - 1)$, if we shift the vector first it would become $(a_n - 1, a_0, a_1..., a_n - 2)$. Then if we create a circulant matrix, the matrix would result in:

$$C = \begin{bmatrix} a_{n-1} & a_0 & a_1 & \cdots & a_{n-3} & a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 & \cdots & a_{n-4} & a_{n-3} \\ a_{n-3} & a_{n-2} & a_{n-1} & \cdots & a_{n-5} & a_{n-4} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_0 \\ a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \end{bmatrix}$$

   .

   But if we are first to create the circulant matrix from a, it would look like this:

$$C_a = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_{n-1} & \ddots & a_2 \\ \vdots & a_1 & a_0 & \ddots & \vdots \\ a_{n-2} & \ddots & \ddots & \ddots & a_{n-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix}$$

   .

   Then if we perform a shift in the matrix, we will get the exact same result as above. This is because the circulant matrix C_a is inherently performing a circular form of convolution (this means applying shift before or after doesn't matter).

   Therefore, $CaS = SCa$, proving that any circulant matrix is commutative with a shift matrix.

2. Next, we will prove that the a (circular) convolution is the only linear operation with shift equivariance. (Hint: how do you prove a bidirectional implication?

To show this, we need to demonstrate that for any linear operation L that is shift-equivariant, L must be a circular convolution.

First, let us consider a linear operation L acting on a vector a, such that this equation is satisfied:

$L(S(x)) = S(L(x))$

Since, L is a linear operation, we can represent the linear transformation of vector x as a matrix M where Lx = Mx. But because L is shift equivalent, M must preserve the structure of x under shifts. The only way to do this is for M to be a circulant matrix.

If we have a vector x and it is shifted by a shift matrix S resulting in Sx. If we take any circulant matrix N, and apply to the shifted vector Sx the resulting matrix would be NSx or SNx (commutative property). If a non-circulant matrix existed such that this property holds true, then it would create a contradiction. Circulant matrices cannot be replaced by non-circulant matrices in order to shift the structure of any vector x.

Therefore, the only matrices which will satisfy the shift-equivariance property are circulant matrices.

3. What does this tell you about designing deep learning architectures for processing spatial or spatio-temporal data like images and videos?

   This property of shift equivariance in terms of circular convolutions has some interesting applications in terms of designing deep learning architectures. I'd assume that CNN's benefit from this property as can track features without changing and based on input space. Moreover, the spatio-temporal data and this property means that models preserve relationships within the data itself, leading to better analysis.

6 SGD

1. **[3 points]**

   No, SGD is not guaranteed to decrease the overall loss in every iteration. A counter example is provided below:

So, we are given the objective function with N = 2 terms and a batch-size of B = 1. Let's label two functions,

$$f_1(w) = 0.5(w-2)^2, f_2(w) = 0.5(w+1)^2$$

Additionally, the gradients of f_1(w) and f_2(w) are as follows:

$$\nabla f_1(w) = w - 2, \nabla f_2(w) = w + 1$$

Since the batch size is 1, we have to choose one function to update for SGD. When updating the weights we use the equation, $f_{num} : w' = w - \eta \nabla f_{num}$. So we choose to update one of these functions:

$$f_1(w) : w_{new} = w_{old} - \eta(w_{old} - 2)$$
$$f_2(w) : w_{new} = w_{old} - \eta(w_{old} + 1)$$

In our counter example, we make $w_{old} = 0.5$ and we make $\eta = 0.1$. So:

$$f_1(w) : w_{new} = 0.5 - 0.1(0.5 - 2) = 0.65$$
$$f_2(w) : w_{new} = 0.5 - 0.1(0.5 + 1) = 0.35$$

The first function will move away from the minimum while the second function will move toward the minimum. Since the batch chooses one of these functions at random, it might potentially move away from the minimum. The first function will increase loss instead of decreasing like the second function;

so SGD doesn't guarantee that the loss function will decrease on every iteration.

7 Paper Review

1. **[4 points]**

   a) Some key contributions that this paper demonstrates are the true capabilities that neural networks have. Not only are they able to demonstrate their amazing capacity for memorization of training data (with random noise), but they're still able to use understanding to generalize fairly well and the authors argue that networks with two layers can be sufficient given enough parameters (showcasing the power of these models). Some strengths that the paper has includes the amount of in-depth experiments the researchers performed in order to gather strong evidence for their claims regarding neural network generalization. Additionally, the paper not only has tons of empirical evidence, but also makes theoretical claims and presents new ideas that can be expanded further to develop applications of neural networks. I thought that the weaknesses in the paper mainly came from their ideas regarding regularization techniques to impact memorization of data and generalization of results. Even though they show that generalizing after memorizing a dataset is fairly accurate, they don't really explain why this is the case (a lot of papers seem to have a sort of black-box approach to solving problems without fully understanding why).

   b) The findings in the paper have altered my belief about neural networks. We talked about how theoretically we can represent neural networks using two layers and a ton of hyper parameters, but putting empirical data and studies/explanations helped a lot. I found it interesting how little we really understand about neural networks and how much still can be studied based on current research. I found it fascinating how the researchers were able to find ways to extract meaningful data and results (even with noise!!!) and make a generalized model that doesn't fall to problems like over fitting. I think a way for research to move forward would be understanding how neural networks find/determine patterns in real life data to generalize and create effective results. This concept is brushed over in the paper and isn't really explored (more theorized rather than empirically studied), so I'd like to see papers in the future that tackle this problem.