Deep Learning - Homework 3

Matt Isaac

February 5, 2019

Problem 1

1. Provide a geometric interpretation of gradient descent in the one-dimensional case.

This would be a case where we have one dimension in which we can move (call this x), while trying to minimize a function $f(x)$. We can visualize this case on a cartesian coordinate system, with x on the horizontal axis, and $f(x)$ on the vertical axis. We would first pick a value of x at which to begin, call this x_0. We then would calculate the gradient of $f(x)$ at x_0, and find the direction in which the slope of the function is descending, and take a step in that direction. At the new point, call this x_1, would again calculate the gradient of $f(x)$, and take a step in the descending direction. Eventually, when the gradient indicates that the function is not very "steep" anymore, on when it become sufficiently close to 0, we would conclude that we have found a minimum. This minimum will be a global minimum if the function is strictly convex. This could also be visualized as a ball rolling along the function $f(x)$ in a "downhill" direction.

2. An extreme version of gradient descent is to use a mini-batch size of just 1. This procedure is known as online or incremental learning. In online learning, a neural network learns from just one training input at a time (just as human beings do). Name one advantage and one disadvantage of online learning compared to stochastic gradient descent with a mini-batch size of, say, 20.

Advantage: Each step would in this iterative approach would be quite fast.

Disadvantage: Because we have an extremely small minibatch of 1, we are essentially letting one data point determine the direction of each step. We lose the robustness that comes with randomly sampling several points. On average, several data points will lead you in the correct direction. Stepping with information from only one data point will make our steps much "noisier".

3. Create a network that classifies the MNIST data set using only 2 layers: the input layer (784 neurons) and the output layer (10) neurons. Train the network using stochastic gradient descent on the training data. What accuracy do you achieve on the test data? You can adapt the code from the Nielson book, but make sure you understand each step to build up the ntework. Alternatively, you can use Tensorflow, or Pytorch. Please save your code as `prob1.py` and state which library/framework you used. Report the learning rate(s) and mini-batch size(s) you used.

This was done useing code adapted from the tutorial found at https://towardsdatascience.com/@jj1385jeff850527.

PyTorch was used to build the neural net.

Learning rate: 0.001
Minibatch size: 100
Test data accuracy: 92%

Problem 2

1. **Alternate presentation of the equations of backpropagation**:

Show that $\delta^L = \nabla_a C \odot \sigma'(z^L)$ can be rewritten as $\delta^L = \Sigma'(z^L)\nabla_a C$, where $\Sigma'(z^L)$ is a square matrix whose diagonal entries are the values $\sigma'(z^L_j)$ and whose off-diagonal entries are zero.

Our goal is to show that $\delta^L = \nabla_a C \odot \sigma'(z^L) = \delta^L = \Sigma'(z^L)\nabla_a C$

Beginning with the lefthand side (LHS), the entries of $\nabla_a C$ will be $\frac{\partial C}{\partial a_1^L}, \cdots, \frac{\partial C}{\partial a_j^L}$, and the entries of $\sigma'(z^L)$ are $\frac{\partial a_1^L}{\partial z_1^L}, \cdots, \frac{\partial a_j^L}{\partial z_j^L}$. Performing the element-wise multiplication,

$$\nabla_{a^L} C \odot \sigma'(z^L) = \begin{bmatrix} \frac{\partial C}{\partial a_1^L} \\ \vdots \\ \frac{\partial C}{\partial a_j^L} \end{bmatrix} \odot \begin{bmatrix} \frac{\partial a_1^L}{\partial z_1^L} \\ \vdots \\ \frac{\partial a_j^L}{\partial z_j^L} \end{bmatrix} = \begin{bmatrix} \frac{\partial C}{\partial z_1^L} \\ \vdots \\ \frac{\partial C}{\partial z_j^L} \end{bmatrix}.$$

Next, moving to the right hand side (RHS),

$$\Sigma'(z^L)\nabla_{a^L} C = \begin{bmatrix} \frac{\partial a_1^L}{\partial z_1^L} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial a_j^L}{\partial z_j^L} \end{bmatrix} \begin{bmatrix} \frac{\partial C}{\partial a_1^L} \\ \vdots \\ \frac{\partial C}{\partial a_j^L} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_1^L}{\partial z_1^L} \frac{\partial C}{\partial a_1^L} \\ \vdots \\ \frac{\partial a_j^L}{\partial z_j^L} \frac{\partial C}{\partial a_j^L} \end{bmatrix} = \begin{bmatrix} \frac{\partial C}{\partial z_1^L} \\ \vdots \\ \frac{\partial C}{\partial z_j^L} \end{bmatrix}$$

Thus, we have shown that

$$\nabla_{a^L} C \odot \sigma'(z^L) = \Sigma'(z^L)\nabla_{a^L} C = \begin{bmatrix} \frac{\partial C}{\partial z_1^L} \\ \vdots \\ \frac{\partial C}{\partial z_j^L} \end{bmatrix}.$$

In other words, $\nabla_{a^L} C \odot \sigma'(z^L)$ can be written as $\Sigma'(z^L)\nabla_{a^L} C$.

2. Show that $\delta^L = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$ can be rewritten as $\delta^l = \Sigma'(z^l)(w^{l+1})^T \delta^{l+1}$.

Our goal is to show that $((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) = \delta^l = \Sigma'(z^l)(w^{l+1})^T \delta^{l+1}$.

Beginning with the left-hand side (LHS), let us first define some of the components:

$$(w^{l+1})^T = \begin{bmatrix} w_{1,1}^{l+1} & w_{2,1}^{l+1} & \cdots & w_{j,1}^{l+1} \\ w_{1,2}^{l+1} & w_{2,2}^{l+1} & \cdots & w_{j,2}^{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,k}^{l+1} & w_{2,k}^{l+1} & \cdots & w_{j,k}^{l+1} \end{bmatrix}$$

$$\delta^{l+1} = \begin{bmatrix} \delta_1^{l+1} \\ \delta_2^{l+1} \\ \vdots \\ \delta_j^{l+1} \end{bmatrix}$$

Next,

$$((w^{l+1})^T \delta^{l+1}) = \begin{bmatrix} w_{1,1}^{l+1} & w_{2,1}^{l+1} & \cdots & w_{j,1}^{l+1} \\ w_{1,2}^{l+1} & w_{2,2}^{l+1} & \cdots & w_{j,2}^{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,k}^{l+1} & w_{2,k}^{l+1} & \cdots & w_{j,k}^{l+1} \end{bmatrix} \begin{bmatrix} \delta_1^{l+1} \\ \delta_2^{l+1} \\ \vdots \\ \delta_j^{l+1} \end{bmatrix} = \begin{bmatrix} \sum_j (w_{j,1}^{l+1})(\delta_j^{l+1}) \\ \sum_j (w_{j,2}^{l+1})(\delta_j^{l+1}) \\ \vdots \\ \sum_j (w_{j,k}^{l+1})(\delta_j^{l+1}) \end{bmatrix}.$$

Now, putting this together to find the entire LHS:

$$((w^{l+1})^T\delta^{l+1}) \odot \sigma'(z^l) = \begin{bmatrix} \sum_j (w^{l+1}_{j,1})(\delta^{l+1}_j) \\ \sum_j (w^{l+1}_{j,2})(\delta^{l+1}_j) \\ \vdots \\ \sum_j (w^{l+1}_{j,k})(\delta^{l+1}_j) \end{bmatrix} \odot \begin{bmatrix} \frac{\partial a^L_1}{\partial z^L_1} \\ \frac{\partial a^L_2}{\partial z^L_2} \\ \vdots \\ \frac{\partial a^L_j}{\partial z^L_j} \end{bmatrix} = \begin{bmatrix} \left(\sum_j (w^{l+1}_{j,1})(\delta^{l+1}_j)\right) \frac{\partial a^L_1}{\partial z^L_1} \\ \left(\sum_j (w^{l+1}_{j,2})(\delta^{l+1}_j)\right) \frac{\partial a^L_2}{\partial z^L_2} \\ \vdots \\ \left(\sum_j (w^{l+1}_{j,k})(\delta^{l+1}_j)\right) \frac{\partial a^L_j}{\partial z^L_j} \end{bmatrix}.$$

Now, looking at the RHS,

$$\Sigma'(z^l)((w^{l+1})^T\delta^{l+1}) = \begin{bmatrix} \frac{\partial a^L_1}{\partial z^L_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial a^L_j}{\partial z^L_j} \end{bmatrix} \begin{bmatrix} \sum_j (w^{l+1}_{j,1})(\delta^{l+1}_j) \\ \sum_j (w^{l+1}_{j,2})(\delta^{l+1}_j) \\ \vdots \\ \sum_j (w^{l+1}_{j,k})(\delta^{l+1}_j) \end{bmatrix} = \begin{bmatrix} \left(\sum_j (w^{l+1}_{j,1})(\delta^{l+1}_j)\right) \frac{\partial a^L_1}{\partial z^L_1} \\ \left(\sum_j (w^{l+1}_{j,2})(\delta^{l+1}_j)\right) \frac{\partial a^L_2}{\partial z^L_2} \\ \vdots \\ \left(\sum_j (w^{l+1}_{j,k})(\delta^{l+1}_j)\right) \frac{\partial a^L_j}{\partial z^L_j} \end{bmatrix}$$

These results show that

$$((w^{l+1})^T\delta^{l+1}) \odot \sigma'(z^l) = \Sigma'(z^l)((w^{l+1})^T\delta^{l+1}) = \begin{bmatrix} \left(\sum_j (w^{l+1}_{j,1})(\delta^{l+1}_j)\right) \frac{\partial a^L_1}{\partial z^L_1} \\ \left(\sum_j (w^{l+1}_{j,2})(\delta^{l+1}_j)\right) \frac{\partial a^L_2}{\partial z^L_2} \\ \vdots \\ \left(\sum_j (w^{l+1}_{j,k})(\delta^{l+1}_j)\right) \frac{\partial a^L_j}{\partial z^L_j} \end{bmatrix}.$$

In other words, we have shown that $((w^{l+1})^T\delta^{l+1}) \odot \sigma'(z^l)$ can be written as $\Sigma'(z^l)((w^{l+1})^T\delta^{l+1})$.

3. By combining the results from problems 2.1 and 2.2, show that $\delta^l = \Sigma'(z^l)(w^{l+1})^T \cdots \Sigma'(z^{L-1})(w^L)^T\Sigma'(z^L)\nabla_a C$

In problem 2.2, it was given that $\delta^l = \left[\Sigma'(z^l)(w^{l+1})^T\right]\delta^{l+1}$. For ease of notation, let $\theta^i = \left[\Sigma'(z^i)(w^{i+1})^T\right]$. That is, $\delta^l = \theta^l \cdot \delta^{l+1}$. Incrementing indices, we see the following relationships should also hold:

$$\begin{aligned} \delta^l &= \theta^l \cdot \delta^{l+1} \\ \delta^{l+1} &= \theta^{l+1} \cdot \delta^{l+2} \\ \delta^{l+2} &= \theta^{l+2} \cdot \delta^{l+3} \\ &\vdots \\ \delta^{L-2} &= \theta^{L-2} \cdot \delta^{L-1} \\ \delta^{L-1} &= \theta^{L-1} \cdot \delta^L \end{aligned}$$

From problem 2.1, we also have the follwoing relationship:

$$\delta^L = \left[\Sigma'(z^L)\nabla_{a^L} C\right].$$

Now, using the relationships defined above, we can recursively substitue for δ^{l+i}. Beginning with $\delta^l = \theta^l \cdot \delta^{l+1}$, we can plug $\theta^{l+1} \cdot \delta^{l+2}$ in for δ^{l+1} since, from the relationships above, $\delta^{l+1} = \theta^{l+1} \cdot \delta^{l+2}$. This continues recursively, substituting the next values in for δ^{l+i}.

$$\begin{aligned}
&\delta^l = \theta^l \cdot \delta^{l+1} \\
\implies &\delta^l = \theta^l \cdot \theta^{l+1} \cdot \delta^{l+2} \\
\implies &\delta^l = \theta^l \cdot \theta^{l+1} \cdot \theta^{l+2} \cdot \delta^{l+3} \\
&\vdots \\
\implies &\delta^l = \theta^l \cdot \theta^{l+1} \cdot \theta^{l+2} \cdots \theta^{L-2} \cdot \delta^{L-1} \\
\implies &\delta^l = \theta^l \cdot \theta^{l+1} \cdot \theta^{l+2} \cdots \theta^{L-2} \cdot \theta^{L-1} \cdot \delta^{L}
\end{aligned}$$

Recall that $\delta^L = \Sigma'(z^L)\nabla_{a^L} C$. Making this substituion into the last equation above, and then substituing $\left[\Sigma'(z^i)(w^{i+1})^T\right]$ back in for θ^i we obtain the following:

$$\begin{aligned}
&\delta^l = \theta^l \cdot \theta^{l+1} \cdot \theta^{l+2} \cdots \theta^{L-2} \cdot \theta^{L-1} \cdot \left[\Sigma'(z^L)\nabla_{a^L} C\right] \\
\implies &\delta^l = \left[\Sigma'(z^l)(w^{l+1})^T\right]\left[\Sigma'(z^{l+1})(w^{l+2})^T\right] \cdots \left[\Sigma'(z^{L-2})(w^{L-1})^T\right]\left[\Sigma'(z^{L-1})(w^L)^T\right]\left[\Sigma'(z^L)\nabla_{a^L} C\right]
\end{aligned}$$

Thus, we have shown that $\delta^l = \Sigma'(z^l)(w^{l+1})^T \cdots \Sigma'(z^{L-1})(w^L)^T \Sigma'(z^L)\nabla_a C$.

4. **Backpropagation with linear neurons**

Suppose we replace the usual non-linear σ function (sigmoid) with $\sigma(z) = z$ throughout the network. Rewrite the backpropagation algorithm for this case.

Problem 3

1. Let $f : \mathbb{R}^d \to \mathbb{R}$. Show that if f is strictly convex, then f has at most one global minimizer.

We will begin by supposing that f is strictly convex, and that (by contradiction) f has two global minimums, located at points a and b. In other words, $f(a)$ and $f(b)$ are both global minimum. Let $f(a) = f(b) = z$. Recalling the definition of strict convexity:

$$f(ta + (1-t)b) < tf(a) + (1-t)f(b),$$

where $t \in (0, 1)$, and $a \neq b$. Let $f(tx + (1-t)y) = f(w)$.

Thus,

$$\begin{aligned}
f(w) = f(ta + (1-t)b) &< tf(a) + (1-t)f(b) \\
&\implies f(w) < tz + (1-t)z \\
&\implies f(w) < z
\end{aligned}$$

Thus, we have found a point, w, at which $f(w) < f(a) = f(b) = z$. This is a contradiction because we began the proof assuming $f(a)$ and $f(b)$ were global minimums. Thus, by contradiction, we have shown that if f is strictly convex, it must have at most one global minimum.

2. Use the Hessian to give a simple proof that the sum of two convex functions is convex. You may assume that the two functions are twice continuously differentiable.
   Hint: You may use the fact that the sum of two PSD matrices is also PSD.

As was suggested in the prompt, let f and g be two convex, twice continuously differentiable functions. From the properties of twice continuously differentiable functions, we know that the hessians, $\nabla^2 f(x)$ and $\nabla^2 g(y)$ both exist, and are positive semi-definite matrices. By the definition of positive semi-definite matrices, for a vector $\mathbf{z}$,

$$\mathbf{z}^T(\nabla^2 f(x))\mathbf{z} \geq 0$$

and

$$\mathbf{z}^T(\nabla^2 g(y))\mathbf{z} \geq 0.$$

This also implies that the sum of the two left-hand sides above will also be ≥ 0. That is,

$$\begin{aligned} &\mathbf{z}^T(\nabla^2 f(x))\mathbf{z} + \mathbf{z}^T(\nabla^2 g(y))\mathbf{z} \geq 0 \\ \implies &\mathbf{z}^T\left[\nabla^2 f(x) + \nabla^2 g(y)\right]\mathbf{z} \geq 0 \end{aligned}$$

The last line above indicates that the sum of the two hessians, $[\nabla^2 f(x) + \nabla^2 g(y)]$, is also positive semi-definite. since the hessian of $f(x) + g(y)$ is positive semi-definite, $f(x) + g(y)$ is convex.

3. Consider the function $f(x) = \frac{1}{2}x^T Ax + b^T x + c$ where A is a symmetric $d \times d$ matrix. Derive the Hessian of f. Under what conditions on A is f convex? Strictly convex?

Since the gradient of a sum of terms is equal to the sum of the gradients of individual terms, we will begin by finding the gradient of each term in the sum individually.

$$\nabla(\frac{1}{2}\mathbf{x}^T A\mathbf{x}) = A\mathbf{x}$$

Next,

$$\nabla(b^T\mathbf{x}) = b.$$

And finally,

$$\nabla c = 0$$

Putting these terms together,

$$\nabla f = A\mathbf{x} + b$$

Next, finding the hessian,

$$\nabla^2 f = A.$$

Thus, since the hessian of f is A, f will be convex when A is PSD and strictly convex when A is PD.

4. Using the definition of convexity, prove that the function $f(x) = x^3$ is not convex.
5. Using the fact that $f(x) = x^3$ is twice continuously differentiable, prove that f is not convex.

First, recall that f is convex if and only if $\nabla^2 f(x)$ is positive semi-definite, or in this 1-dimensional case, ≥ 0 for all values of x.

$$\begin{aligned} f(x) &= x^3 \\ \implies \nabla f(x) &= 3x^2 \\ \implies \nabla^2 f(x) &= 6x \end{aligned}$$

If we pick a value of $x < 0$, for instance, $x = 1$, then $\nabla^2 f(x) = 6(-1) = -6 < 0$. Thus, because $\nabla^2 f(x)$ is not ≥ 0 for all x, the function $f(x) = x^3$ is not convex.

6. A function f is concave if $-f$ is convex. Prove that for all $x > 0$, the function $f(x) = \ln(x)$ is concave.

Let us define $g(x) = -\ln(x)$. By the property stated in the prompt, if $g(x)$ as it has been defined is convex, then $f(x) = \ln(x)$ will be concave.

Since $g(x)$ is twice differntiable, we will find the hessian, and show that it is positive for all values of x.

$$\begin{aligned} g(x) &= -\ln(x) \\ \implies \nabla g(x) &= -\frac{1}{x} \\ \implies \nabla^2 g(x) &= \frac{1}{x^2} \end{aligned}$$

Because x^2 is in the denominator, all values of x, whether postive or negative, will result in $\nabla^2 g(x) = \frac{1}{x^2} > 0$. That is, the hessian of $g(x)$ is ≥ 0 for all x. Therefore, $g(x)$ is convex. As was stated previously, if $g(x) = -\ln(x)$ is convex, then $f(x) = \ln(x)$ is concave.

7. Let $f(x) = ax + b$, where $a, b \in \mathbb{R}$. f is called an *affine function* (a linear function plus an offset). Prove that f is both convex and concave. Does f have a global minimum or a global maximum? Are there any other functions that are twice continuously differentiable and both convex and concave that do not have the form of an affine function? Explain your reasoning.

Proof of Convexity

$f(x) = ax + b$ is twice differentiable.

$$\begin{aligned} f(x) &= ax + b \\ \implies \nabla f(x) &= a \\ \implies \nabla^2 f(x) &= 0 \end{aligned}$$

So $\nabla^2 f(x) = 0 \geq 0$ for all values of x. Thus, $f(x)$ is convex.

Proof of Concavity

Let $g(x) = -f(x)$. If $f(x) = -f(x)$ is convex, then $f(x)$ is concave. In this case, $g(x) = -ax - b$.

$$\begin{aligned} g(x) &= -ax - b \\ \implies \nabla g(x) &= -a \\ \implies \nabla^2 g(x) &= 0 \end{aligned}$$

So $\nabla^2 g(x) = 0 \geq 0$ for all values of x. Thus, $g(x)$ is convex. As was stated above, if $g(x) = -f(x)$ is convex, then $f(x)$ is concave.

NEED TO REWORD: No, f does not have a global minimum or a global maximum. Since the line will continue increasing infinitely in one direction, and continue decreasing infinitely in the other, it will extend to positive and negative infinity, and never have a global maximum or minimum.

There are no other functions that are twice continuously differentiable and both convex and concave. s