CS 188 Spring 2016 Section 11: Perceptrons/Neural Networks

1 Perceptron

We would like to use a perceptron to train a classifier for datasets with 2 features per point and labels +1 or -1. Consider the following labeled training data:

<table>
<thead>
<tr>
<th>Features</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(x_1, x_2)$</td>
<td>$y^*$</td>
</tr>
<tr>
<td>(-1,2)</td>
<td>1</td>
</tr>
<tr>
<td>(3,-1)</td>
<td>-1</td>
</tr>
<tr>
<td>(1,2)</td>
<td>-1</td>
</tr>
<tr>
<td>(3,1)</td>
<td>1</td>
</tr>
</tbody>
</table>

1. Our two perceptron weights have been initialized to $w_1 = 2$ and $w_2 = -2$. After processing the first point with the perceptron algorithm, what will be the updated values for these weights?

   For the first point, $y = g(w_1 x_1 + w_2 x_2) = g(2 \cdot -1 + -2 \cdot 2) = g(-5) = -1$, which is incorrectly classified. To update the weights, we add the first data point: $w_1 = 2 + (-1) = 1$ and $w_2 = -2 + 2 = 0$.

2. After how many steps will the perceptron algorithm converge? Write “never” if it will never converge.

   Note: one steps means processing one point. Points are processed in order and then repeated, until convergence.

   The data is not separable, so it will never converge.

Instead of the standard perceptron algorithm, we decide to treat the perceptron as a single node neural network and update the weights using gradient descent on the loss function.

The loss function for one data point is $Loss(y, y^*) = (y - y^*)^2$, where $y^*$ is the training label for a given point and $y$ is the output of our single node network for that point.

3. Given a general activation function $g(z)$ and its derivative $g'(z)$, what is the derivative of the loss function with respect to $w_1$ in terms of $g$, $g'$, $y^*$, $x_1$, $x_2$, $w_1$, and $w_2$?

   $$\frac{\partial Loss}{\partial w_1} = 2(g(w_1 x_1 + w_2 x_2) - y^*)g'(w_1 x_1 + w_2 x_2)x_1$$
4. For this question, the specific activation function that we will use is:

\[ g(z) = 1 \text{ if } z \geq 0 \text{ and } = -1 \text{ if } z < 0 \]

Given the following gradient descent equation to update the weights given a single data point. With initial weights of \( w_1 = 2 \) and \( w_2 = -2 \), what are the updated weights after processing the first point?

Gradient descent update equation: \( w_i = w_i - \alpha \frac{\partial \text{Loss}}{\partial w_i} \)

Because the gradient of \( g \) is zero, the weights will stay \( w_1 = 2 \) and \( w_2 = -2 \).

5. What is the most critical problem with this gradient descent training process with that activation function?

The gradient of that activation function is zero, so the weights will not update.
2 Neural Nets

Consider the following two-neuron network for binary classification:

Here $x$ is a single real-valued input (not a vector) with an associated class $y$ (0 or 1). There are two neurons, with input weights $w_1$ and $w_2$, and activation functions $g_1$ and $g_2$. The output

$$h_w(x) = a_2$$

is a value between 0 and 1, representing the probability of being in class 1. We will be using a real-valued loss function $\text{Loss}_w(x, y)$.

(a) Let $z_1$ and $z_2$ refer to the pre-activation values at neuron 1 and neuron 2, respectively. Write $z_1$, $a_1$, $z_2$, and $a_2$ in terms of the previous values of the neural network.

$$z_1 = xw_1$$
$$a_1 = g_1(z_1)$$
$$z_2 = a_1w_2$$
$$a_2 = g_2(z_2)$$

(b) Write the output $a_2$ in terms of the input $x$, weights $w_i$, and activation functions $g_i$.

$$h_w(x) = g_2(w_2g_1(w_1x))$$

(c) Use the chain rule to derive $\delta \text{Loss} \delta w_2$. Write your expression as a product of partial derivatives that can be directly computed – you don’t have to directly compute them. (Hint: the series of expressions you wrote in part 1 will be very useful; you may use any of those variables.)

$$\frac{\delta \text{Loss}}{\delta w_2} = \frac{\delta \text{Loss}}{\delta a_2} \frac{\delta a_2}{\delta z_2} \frac{\delta z_2}{\delta w_2}$$

(d) Now use the chain rule to derive $\frac{\delta \text{Loss}}{\delta w_1}$ in terms of the same quantities as part 3.

$$\frac{\delta \text{Loss}}{\delta w_1} = \frac{\delta \text{Loss}}{\delta a_2} \frac{\delta a_2}{\delta z_2} \frac{\delta z_2}{\delta a_1} \frac{\delta a_1}{\delta z_1} \frac{\delta z_1}{\delta w_1}$$
(e) Suppose the loss function is quadratic \(\text{Loss}_w(x, y) = (y - a_2)^2\) and \(g_1\) and \(g_2\) were both sigmoid functions \(\frac{1}{1 + e^{-z}}\). Using the fact that \(\frac{\delta g}{\delta z} = g_i(z_i)(1 - g_i(z_i))\), write \(\frac{\delta \text{Loss}}{\delta w_2}\) and \(\frac{\delta \text{Loss}}{\delta w_1}\) in terms of \(x, y, w, a, z\):

\[
\frac{\delta \text{Loss}}{\delta w_2} = -2(y - a_2)g_2(z_2)(1 - g_2(z_2))a_1 \\
= -2(y - a_2)a_1 \frac{1}{1 + e^{-z_2}} \left( 1 - \frac{1}{1 + e^{-z_2}} \right)
\]

\[
\frac{\delta \text{Loss}}{\delta w_1} = 2(y - a_2)g_2(z_2)(1 - g_2(z_2))w_2g_1(z_1)(1 - g_2(z_2))x \\
= -2(y - a_2)a_1 a_2 w_2 x \frac{1}{1 + e^{-z_1}} \left( 1 - \frac{1}{1 + e^{-z_1}} \right) \frac{1}{1 + e^{-z_2}} \left( 1 - \frac{1}{1 + e^{-z_2}} \right)
\]

(f) Write the stochastic gradient descent update for \(w_1\) in terms of the step size \(\alpha\) and the values computed above:

\[
w_1 \leftarrow w_1 + \alpha(y - a_2)a_1 a_2 w_2 x \frac{1}{1 + e^{-z_1}} \left( 1 - \frac{1}{1 + e^{-z_1}} \right) \frac{1}{1 + e^{-z_2}} \left( 1 - \frac{1}{1 + e^{-z_2}} \right)
\]