CMPS242 Homework #4

Zayd Hammoudeh

October 30, 2017

Problem #1

Consider 1-dimensional linear regression.

First compute the optimum solution, w^*, for a batch of examples, (x_i, y_i), $1 \leq i \leq n$, ie the weight that minimizes the total loss on examples: $L(w) = \sum_{i=1}^{n}(wx_i - y_i)^2$.

Assume labels are expensive (see lecture 7). You are given only one of the labels y_i. Compute the optimal solution w_i based on a single example (x_i, y_i).

Show that if i is chosen wrt the distribution $\frac{x_i^2}{\sum_j x_j^2}$, then the expected loss of w_i on all examples is twice the optimum ie

$$\mathbf{E}[L(w_i^*)] = 2L(w^*),$$

when all x_i are non-zero.

Hint: First check the above equation on Octave or Matlab on some random data. Make your solution as simple as you can.

For a set of n ordered pairs (x_i, y_i), the squared loss function L is given by

$$L(w) = \sum_j (wx_j - y_j)^2. \tag{1}$$

The optimal weight vector, w^* can be found by taking the derivative and setting it equal to 0. Hence,

$$\frac{\partial L(w)}{\partial w} = 0 = \frac{\partial}{\partial w}\sum_j (w^* x_j - y_j)^2 \tag{2}$$

$$0 = \sum_j 2(w^* x_j - y_j)x_j \tag{3}$$

$$\sum_j x_j y_j = w^* \sum_j x_j^2 \tag{4}$$

$$w^* = \frac{\sum_j x_j y_j}{\|\mathbf{x}\|^2} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2}. \tag{5}$$

For a single example, the weight would just be:

$$w_i^* = \frac{y_i}{x_i} \tag{6}$$

If our selection of x_i was deterministic, then it would perform poorly against an adversary. Hence, it needs to be probabilistic. The probability distribution specified in the problem was

$$P(X = x_i) = \frac{x_i^2}{\sum_j x_j^2} \tag{7}$$

$$= \frac{x_i^2}{\|\mathbf{x}\|^2}. \tag{8}$$

Therefore, the expected loss would be:

$$\mathbf{E}\left[L(w_i^*)\right] = \sum_i \Pr(X = x_i) L(w_i^*) \tag{9}$$

$$= \sum_i \frac{x_i^2}{\|\mathbf{x}\|^2} L(w_i^*) \tag{10}$$

$$= \sum_i \frac{x_i^2}{\|\mathbf{x}\|^2} \sum_j (w_i^* x_j - y_j)^2 \tag{11}$$

$$= \sum_i \frac{x_i^2}{\|\mathbf{x}\|^2} \sum_j \left((w_i^*)^2 x_j^2 - 2 w_i^* x_j y_j + y_j^2 \right) \tag{12}$$

$$= \sum_i \left(\frac{x_i^2}{\|\mathbf{x}\|^2} \sum_j \left((w_i^*)^2 x_j^2 \right) - \frac{2}{\|\mathbf{x}\|^2} \sum_j \left(w_i^* x_i^2 x_j y_j \right) + \frac{x_i^2}{\|\mathbf{x}\|^2} \sum_j y_j^2 \right) \tag{13}$$

$$= \sum_i \left(\frac{x_i^2}{\|\mathbf{x}\|^2} \sum_j \left(\frac{y_i^2}{x_i^2} x_j^2 \right) - \frac{2}{\|\mathbf{x}\|^2} \sum_j \left(\frac{y_i}{x_i} x_i^2 x_j y_j \right) + \frac{x_i^2}{\|\mathbf{x}\|^2} \sum_j y_j^2 \right) \tag{14}$$

$$= \sum_i \left(y_i^2 - \frac{2}{\|\mathbf{x}\|^2} \sum_j \left(y_i x_i x_j y_j \right) + \frac{x_i^2}{\|\mathbf{x}\|^2} \|\mathbf{y}\|^2 \right) \tag{15}$$

$$= \|\mathbf{y}\|^2 + \frac{\|\mathbf{x}\|^2}{\|\mathbf{x}\|^2} \|\mathbf{y}\|^2 - \sum_i \frac{2}{\|\mathbf{x}\|^2} \sum_j \left(y_i x_i x_j y_j \right) \tag{16}$$

$$= 2\|\mathbf{y}\|^2 - \frac{2}{\|\mathbf{x}\|^2} \sum_i \sum_j \left(y_i y_j x_i x_j \right) \tag{17}$$

We can then find the similar relation for w^*.

$$L(w^*) = \sum_i (w^* x_i - y_i)^2 \tag{18}$$

$$= \sum_i \left((w^*)^2 x_i^2 - 2 w^* x_i y_i + y_i^2 \right) \tag{19}$$

$$= \|\mathbf{y}\|^2 + \sum_i \left(\frac{\left(\sum_j x_j y_j \right)^2}{\left(\|\mathbf{x}\|^2 \right)^2} x_i^2 - 2 \frac{\left(\sum_j x_j y_j \right)}{\|\mathbf{x}\|^2} x_i y_i \right) \tag{20}$$

$$= \|\mathbf{y}\|^2 + \frac{\left(\sum_j x_j y_j \right)^2}{\left(\|\mathbf{x}\|^2 \right)^2} \|\mathbf{x}\|^2 - \frac{2}{\|\mathbf{x}\|^2} \sum_i \sum_j y_i y_j x_i x_j \tag{21}$$

$$= \|\mathbf{y}\|^2 + \frac{\left(\sum_j x_j y_j \right)\left(\sum_i x_i y_i \right)}{\|\mathbf{x}\|^2} - \frac{2}{\|\mathbf{x}\|^2} \sum_i \sum_j y_i y_j x_i x_j \tag{22}$$

$$= \|\mathbf{y}\|^2 + \frac{1}{\|\mathbf{x}\|^2} \sum_i \sum_j y_i y_j x_i x_j - \frac{2}{\|\mathbf{x}\|^2} \sum_i \sum_j y_i y_j x_i x_j \tag{23}$$

$$= \|\mathbf{y}\|^2 - \frac{1}{\|\mathbf{x}\|^2} \sum_i \sum_j \left(y_i y_j x_i x_j \right) \tag{24}$$

Hence, from Eq. (17) and Eq. (24), it is clear that

$$\mathbf{E}[L(w_i^*)] = 2L(w^*),$$

completing the proof.

Problem #2

Compute all the derivatives using Backpropagation for a 3-layer neural net with one output when the transfer function is the cumulative Gaussian density

$$\Phi(a) = \int_{-\infty}^{a} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz$$

and the output node is the square loss. Assume the node of the hidden layer as well as the output node each have a bias term. Compute the derivatives of the loss wrt the weights between the 2nd and 3rd layer and the 1st and 2nd layer as well as the derivatives of the loss wrt the bias terms.

Hint: First produce a writeup when the transfer function is the sigmoid and then modify it.

Notation:

- $\hat{y}$ – Predicted output value.
- $x_{j,k}^l$ – Input to node j in layer l from node k in layer $l-1$,
- $\mathbf{x}_j^l$ – Vector notation for all neuron inputs from layer $l-1$ to node j in layer l.
- $w_{j,k}^l$ – Weight for connection between the k^{th} neuron in $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer.
- $\mathbf{w}_j^l$ – Vector notation for all connection weights between layer $l-1$ and node j in layer l.
- b_j^l – Bias term for the j^{th} neuron in the l^{th} layer.
- $\Phi(a)$ – Transfer function for the neurons. It is given as:

$$\Phi(a) := \int_{-\infty}^{a} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz. \tag{25}$$

- a_j^l – Input to the transfer function for neuron j in layer l. It is defined as:

$$a_j^l := \mathbf{w}_j^l \mathbf{x}_j^l + b_j^l. \tag{26}$$

- J – Squared loss function. It is defined formally as

$$J = \frac{1}{2}(\hat{y} - y)^2 \tag{27}$$

Additional Notes: By the Fundamental Theorem of Calculus, for given a continuous function f, the derivative of its definite integral whose lower limit is a constant is given by

$$\frac{\partial}{\partial x} \int_a^x f(t)dt = f(x) \tag{28}$$

The transfer function in Eq. 25 is not exactly in this form. However, it could be transformed into that form via a limit as shown below.

$$\frac{\partial}{\partial a}\Phi(a) = \frac{\partial}{\partial a} \int_{-\infty}^{a} f(t)dt = \lim_{b \to -\infty} \frac{\partial}{\partial x} \int_b^x f(t)dt = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{29}$$

Back-propagation for the Weights: Using the chain rule, the derivative of the for all the weights between the output and hidden layer is:

$$\frac{\partial J}{\partial \mathbf{w}_1^3} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1^3} \cdot \frac{\partial a_1^3}{\partial \mathbf{w}_1^3} \tag{30}$$

The loss function for this network is given by the squared loss. Given a single target value y, the derivative of this function is:

$$\frac{\partial J}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}} \left(\frac{1}{2} (\hat{y} - y)^2 \right) = \hat{y} - y \tag{31}$$

The derivative of $\hat{y}$ is then

$$\frac{\partial \hat{y}}{\partial \mathbf{w}_1^3} = \frac{\partial}{\partial \mathbf{w}_1^3} \Phi(a_1^3) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \frac{\partial a_1^3}{\partial \mathbf{w}_1^3} \tag{32}$$

$$= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \mathbf{x}_1^3. \tag{33}$$

This makes the complete derivative for the output layer is:

$$\frac{\partial J}{\partial \mathbf{w}_1^3} = \boxed{(\hat{y} - y) \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \mathbf{x}_1^3} \tag{34}$$

The derivative of the loss function with respect to the weights between the k^{th} neuron in the hidden layer and the input layer is:

$$\frac{\partial J}{\partial \mathbf{w}_k^2} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1^3} \cdot \frac{\partial a_1^3}{\partial \mathbf{x}_1^3} \cdot \frac{\partial \mathbf{x}_1^3}{\partial a_k^2} \cdot \frac{\partial a_k^2}{\partial \mathbf{w}_k^2} \tag{35}$$

$$= (\hat{y} - y) \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \cdot \frac{\partial a_1^3}{\partial \mathbf{x}_1^3} \cdot \frac{\partial \mathbf{x}_1^3}{\partial a_k^2} \cdot \frac{\partial a_k^2}{\partial \mathbf{w}_k^2} \tag{36}$$

$$= (\hat{y} - y) \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \mathbf{w}_1^3 \cdot \frac{\partial \mathbf{x}_1^3}{\partial a_k^2} \cdot \frac{\partial a_k^2}{\partial \mathbf{w}_k^2} \tag{37}$$

$$= \boxed{(\hat{y} - y) \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \mathbf{w}_1^3 \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_k^2\right)^{\mathrm{T}} \mathbf{x}_k^2 + b_j^2 \right)^2}{2} \right) \mathbf{x}_k^2}. \tag{38}$$

Back-propagation for the Bias Terms: Taking the derivative with respect to the bias is simpler. Eq. 31 applies as it did for the weights. Now, the derivative of the transfer function with respect to b_1^3 is:

$$\frac{\partial}{\partial b_1^3} \hat{y} = \frac{\partial}{\partial b_1^3} \Phi(a_1^3) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3\right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right). \tag{39}$$

This makes the bias term derivative for the output layer is

$$\frac{\partial J}{\partial b_1^3} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1^3} \cdot \frac{\partial a_1^3}{\partial b_1^3} \tag{40}$$

$$= \boxed{(\hat{y} - y) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3 \right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right)}. \tag{41}$$

Similar to what was done in the last equation and Eq. 35, we can find the bias term for the k^{th} neuron in the hidden layer. Hence, it is:

$$\frac{\partial J}{\partial b_k^2} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1^3} \cdot \frac{\partial a_1^3}{\partial \mathbf{x}_1^3} \cdot \frac{\partial \mathbf{x}_1^3}{\partial a_k^2} \cdot \frac{\partial a_k^2}{\partial b_k^2} \tag{42}$$

$$= (\hat{y} - y) \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3 \right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \mathbf{w}_1^3 \cdot \frac{\partial \mathbf{x}_1^3}{\partial a_k^2} \cdot \frac{\partial a_k^2}{\partial b_k^2} \tag{43}$$

$$= \boxed{(\hat{y} - y) \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_1^3 \right)^{\mathrm{T}} \mathbf{x}_1^3 + b_1^3 \right)^2}{2} \right) \mathbf{w}_1^3 \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\left(\left(\mathbf{w}_k^2 \right)^{\mathrm{T}} \mathbf{x}_k^2 + b_k^2 \right)^2}{2} \right)}. \tag{44}$$

Problem #3

Derive the matching loss for the *rectifier* activation/transfer function $f(a) := \max(0, a)$. This function is also known as the ramp function.

Hint: Review how the matching loss is computed when the transfer functions are the sigmoid function and the sign function: $f(a) = \text{sign}(a)$. (See material for lecture 5.)

The max function can be written as a piecewise identity as shown below.

$$f(a) = \max(0, a) = \begin{cases} a & a \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

The integral of the max function is similarly piecewise.

$$\int \max(0, a)da = F(a) = \begin{cases} \frac{a^2}{2} & a \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

$$= \frac{1}{2}\max(0, a)a \tag{47}$$

For a function F, the matching loss is defined as

$$\Delta_H(\mathbf{w} \cdot \mathbf{x}, y) = H(\mathbf{w} \cdot \mathbf{x}) - H(y) - (\mathbf{w} \cdot \mathbf{x} - y)y. \tag{48}$$

Substituting, the matching loss becomes

$$\boxed{\Delta_H(\mathbf{w} \cdot \mathbf{x}, y) = \frac{1}{2}\left(\max(0, \mathbf{w} \cdot \mathbf{x})\mathbf{w} \cdot \mathbf{x} - \max(0, y)y - 2(\mathbf{w} \cdot \mathbf{x} - y)y\right)} \tag{49}$$