1 Key Takeaways from this Week

The goal of ML is to learn a function $f$ parameterized by $w$ s.t. $f_w(x)$ is very close to $y$. Each algorithm is a triplet of three design decisions:

1. **Hypothesis class** – How will I write down my prediction for $y$ as a function of $x$? Which parameters $w$ do I need to learn?

2. **Loss function** – How do I measure how far my prediction is from the real $y$?

3. **Optimization algorithm** – What algorithm will I use to minimize my loss function?

<table>
<thead>
<tr>
<th>$y \in \mathbb{R}$</th>
<th>Linear regression $f_w(x) := w \cdot \phi(x)$</th>
<th>Squared loss: $(f_w(x) - y)^2$</th>
<th>GD or SGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y \in {-1, 1}$</td>
<td>(Binary) linear classification $f_w(x) := \text{sign}(w \cdot \phi(x))$</td>
<td>0-1 loss: $1[</td>
<td>f_w(x)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hinge loss: $\max{1 - (w \cdot \phi(x))y, 0}$</td>
<td>GD or SGD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logistic loss: $\log(1 + e^{-y(w \cdot \phi(x))})$</td>
<td>GD or SGD</td>
</tr>
</tbody>
</table>

**Dimension check.** Above, $w, \phi(x) \in \mathbb{R}^d$, while $y$ is a scalar.
2 Practice Problems

1) Problem 1: Gradient computation

(i) Let \( \phi(x): \mathbb{R} \rightarrow \mathbb{R}^d, \ w \in \mathbb{R}^d, \) and \( f(x, w) = w \cdot \phi(x) \). Consider the following loss function.

\[
\text{Loss}(x, y, w) = \frac{1}{2} \max \{2 - (w \cdot \phi(x))y, 0\}^2. \tag{1}
\]

Compute its gradient \( \nabla_w \text{Loss}(x, y, w) \).

Solution  Note that \( \text{Loss}(x, y, w) \) can be written as the following piecewise defined function using the definition of max.

\[
\text{Loss}(x, y, w) = \begin{cases} 
\frac{1}{2}(2 - (w \cdot \phi(x)y))^2 & \text{if } 2 - (w \cdot \phi(x))y \geq 0 \\
0 & \text{otherwise.} 
\end{cases} \tag{2}
\]

Using the chain rule, we get that the gradient is:

\[
\nabla_w \text{Loss}(x, y, w) = \begin{cases} 
-(2 - w \cdot \phi(x)y)\phi(x)y & \text{if } 2 - w \cdot \phi(x)y \geq 0 \\
0 & \text{otherwise.} 
\end{cases} \tag{3}
\]
2) Problem 2: More gradient computations

(i) Compute the gradient of the loss function below.

\[ \text{Loss}(x, y, w) = \sigma(-(w \cdot \phi(x))y), \]  

where \( \sigma(z) = (1 + \exp(-z))^{-1} \) is the logistic function.

Solution Let \( z = (-w \cdot \phi(x))y \), then \( \text{Loss}(x, y, w) = \sigma(z) = (1 + \exp(-z))^{-1} \).

Applying the chain rule, we get

\[
\nabla_w \text{Loss}(x, y, w) = \frac{\partial \sigma(z)}{\partial z} \nabla_w z
\]

(5)

\[
= -(1 + \exp(-z))^{-2} \exp(-z) y \phi(x)
\]

(6)

\[
= -(1 + \exp(-z))^{-1} \left( \frac{\exp(-z)}{1 + \exp(-z)} \right) y \phi(x)
\]

(7)

\[
= -\sigma(z)(1 - \sigma(z)) y \phi(x).
\]

(8)

Plugging in the expression for \( z \) gives us the final expression.

\[
\nabla_w \text{Loss}(x, y, w) = -\sigma(-(w \cdot \phi(x))y)(1 - \sigma(-(w \cdot \phi(x))y)) y \phi(x).
\]

(9)

(ii) Suppose we have the following loss function.

\[ \text{Loss}(x, y, w) = \max\{1 - \lfloor (w \cdot \phi(x))y \rfloor, 0\}, \]  

where \( \lfloor a \rfloor \) returns \( a \) rounded down to the nearest integer. Determine what the gradient of this function looks like, and whether gradient descent is suitable to optimize this loss function.

Solution

\[
\text{Loss}(x, y, w) = \begin{cases} 
1 - \lfloor (w \cdot \phi(x))y \rfloor & \text{if } \lfloor (w \cdot \phi(x))y \rfloor \leq 1, \\
0 & \text{otherwise}
\end{cases}
\]

(11)

If we draw the plot for the floor function, we can see that its derivative is 0 (the lines are flat and the slope is 0) almost everywhere.
Thus, when applying chain rule to find the gradient of Loss($x, y, w$), the computed gradient will also be 0 almost everywhere, so gradient descent is not suitable to optimize this function as the iterates would not move from the point of initialization.
3) Problem 3: Gradient and Gradient Descent

(i) Let \( \phi(x) : \mathbb{R} \mapsto \mathbb{R}^d, \ w \in \mathbb{R}^d \). Consider the following objective function (a.k.a. loss function).

\[
\text{Loss}(x, y, w) = \begin{cases} 
1 - 2(w \cdot \phi(x))y & \text{if } (w \cdot \phi(x))y \leq 0 \\
(1 - (w \cdot \phi(x))y)^2 & \text{if } 0 < (w \cdot \phi(x))y \leq 1 \\
0 & \text{if } (w \cdot \phi(x))y > 1,
\end{cases}
\]

where \( y \in \mathbb{R} \). Compute the gradient \( \nabla_w \text{Loss}(x, y, w) \).

**Solution** We apply the rules to compute the gradient for each case separately, leading to the following piece-wise function for the gradient.

\[
\nabla_w \text{Loss}(x, y, w) = \begin{cases} 
-2\phi(x)y & \text{if } (w \cdot \phi(x))y \leq 0 \\
-2(1 - (w \cdot \phi(x))y)\phi(x)y & \text{if } 0 < (w \cdot \phi(x))y \leq 1 \\
0 & \text{if } (w \cdot \phi(x))y > 1
\end{cases}
\] (12)

(ii) Write out the Gradient Descent update rule for some function \( \text{TrainLoss}(w) : \mathbb{R}^d \mapsto \mathbb{R} \).

**Solution** \( w := w - \eta \nabla_w \text{TrainLoss}(w) \), where \( \eta \) is the step size.

(iii) Let \( d = 2 \) and \( \phi(x) = [1, x] \). Consider the following loss function.

\[
\text{TrainLoss}(w) = \frac{1}{2} \left( \text{Loss}(x_1, y_1, w) + \text{Loss}(x_2, y_2, w) \right).
\] (13)

Compute \( \nabla_w \text{TrainLoss}(w) \) for the following values of \( x_1, y_1, x_2, y_2, w \).

\[
w = \begin{bmatrix} 0, 1 \end{bmatrix},
\]

\[
x_1 = -2, \ y_1 = 1,
x_2 = -1, \ y_2 = -1.
\]

**Solution**

\[
\nabla_w \text{TrainLoss}(w) = \frac{1}{2} \nabla_w \left( \text{Loss}(x_1, y_1, w) + \text{Loss}(x_2, y_2, w) \right)
\]

\[
= \frac{1}{2} \nabla_w \text{Loss}(x_1, y_1, w) + \frac{1}{2} \nabla_w \text{Loss}(x_2, y_2, w)
\]

For each of the terms above, we plug in the expression for the gradient computed in part (i) above.
Term one. Note that $\phi(x_1) = [1, -2]$. Since $(w \cdot \phi(x_1))y_1 = -1$, we consider the first piece (Case 1) in the gradient expression (Equation 12). We have

$$
\nabla_w \text{Loss}(x_1, y_1, w) = -2\phi(x_1)y_1 = [-2, 4].
$$

(14)

Term two. Note that $\phi(x_2) = [1, -1]$. Similarly, $(w \cdot \phi(x_2))y_2 = \frac{1}{2}$ taking us to Case 2 so

$$
\nabla_w \text{Loss}(x_2, y_2, w) = -2(1 - (w \cdot \phi(x_2))y_2)\phi(x_2)y_2 = [1, -1].
$$

(15)

Combining the terms,

$$
\nabla_w \text{TrainLoss}(w) = \frac{1}{2}([[-2, 4] + [1, -1]])
= \begin{bmatrix}
-1 & 3 \\
-2 & 2
\end{bmatrix}.
$$

(16)

(iv) Perform two iterations of Gradient Descent to minimize the objective function $\text{TrainLoss}(w) = \frac{1}{2} \left( \text{Loss}(x_1, y_1, w) + \text{Loss}(x_2, y_2, w) \right)$ with values for $x_1, y_1, x_2, y_2$ as above. Use initialization $w^0 = [0, \frac{1}{2}]$ and step size $\eta = \frac{1}{2}$.

Solution Note that we have already computed $\nabla_w \text{TrainLoss}(w)$ at the initialization point $w^0$ in the question above.

$$
\begin{align*}
\mathbf{w}^1 &= \mathbf{w}^0 - \eta \nabla_w \text{TrainLoss}(w) \text{ at } \mathbf{w}^0 \\
&= \begin{bmatrix} 0, \frac{1}{2} \end{bmatrix} - \left( \frac{1}{2} \right) \begin{bmatrix} 1, 3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -1 \end{bmatrix}.
\end{align*}
$$

From part (iii) above

Now we need to compute $\nabla_w \text{Loss}(x_1, y_1, w)$ and $\nabla_w \text{Loss}(x_2, y_2, w)$ at the new iterate $\mathbf{w}^1$.

We repeat the process we did for (iii) by applying the piece-wise defined gradient (Equation 12) to the two points, this time setting $\mathbf{w} = \mathbf{w}^1$. 

6
Term one. Since \((w^1 \cdot \phi(x_1))y_1 = \frac{3}{4}\), we have \(\nabla_w \text{Loss}(x_1, y_1, w) = -2(1 - (w^1 \cdot \phi(x_1))y_1)\phi(x_1)y_1 = [-\frac{1}{2}, 1]\). Note that we are now in Case 2 with respect to the piecewise definition of the gradient (Equation 12). When computing \(\nabla_w \text{Loss}(x_1, y_1, w)\) at \(w^0\), we were in Case 1.

Term two. \((w^1 \cdot \phi(x_2))y_2 = -\frac{1}{2}\) taking us to Case 1, so \(\nabla_w \text{Loss}(x_2, y_2, w) = -2\phi(x_2)y_2 = [2, -2]\).

Hence,

\[
\begin{align*}
w^2 &= w^1 - \eta \nabla_w \text{TrainLoss}(w) \text{ at } w^1 \\
&= \begin{bmatrix} \frac{1}{4}, -\frac{1}{4} \end{bmatrix} - \left( \frac{1}{2} \right) \left( \frac{1}{2} \right) \left( \begin{bmatrix} -\frac{1}{2}, 1 \end{bmatrix} + [2, -2] \right) \\
&= \begin{bmatrix} -\frac{1}{8}, 0 \end{bmatrix}.
\end{align*}
\]
4) **Problem 4 (Extra): Vector visualization**

Recall that we can visualize a vector \( \mathbf{w} \in \mathbb{R}^d \) as a point in \( d \)-dimensional space. Let us now visualize some vectors in 2 dimensions on pen and paper.

(i) Consider \( \mathbf{x} \in \mathbb{R}^2 \). Draw the line (i.e. the “decision boundary”) that separates between vectors having a positive dot product with weights \( \mathbf{w} = [3, -2] \) and those having a negative dot product. Shade the part of the 2D plane that contains vectors satisfying \( \mathbf{w} \cdot \mathbf{x} > 0 \).

Hint: It might help to write out the expression for the dot product and seeing the relation between \( x_1 \) and \( x_2 \) that leads to a positive dot product. You could also use the geometric interpretation of the dot product.

**Solution** \( \mathbf{w} \cdot \mathbf{x} = 3x_1 - 2x_2 > 0 \)
(ii) Repeat the above for $w = [2, 0]$ and $w = [0, 2]$.

Solution When $w = [2, 0]$, $w \cdot x = 2x_1 > 0$

When $w = [0, 2]$, $w \cdot x = 2x_2 > 0$
(iii) A small twist: visualize the set of vectors where $w \cdot x \geq 1$ for $w = [3, -2]$.

Solution \[ w \cdot x = 3x_1 - 2x_2 \geq 1, \text{ so } 3x_1 - 2x_2 - 1 \geq 0 \]

Note that we get a line that is parallel to the one in (i) but shifted by a certain amount.
(iv) Consider the following element-wise inequality notation. For two vectors \( \mathbf{a}, \mathbf{b} \in \mathbb{R}^d \),

\[
\mathbf{a} \leq \mathbf{b} \iff a_i \leq b_i \quad \forall i = 1, 2, \ldots d.
\] (17)

Suppose we have a matrix \( A \in \mathbb{R}^{2 \times 2} \) and a vector \( \mathbf{b} \in \mathbb{R}^2 \) as follows.

\[
A = \begin{bmatrix} 3 & -2 \\ 2 & 0 \end{bmatrix}, \quad \mathbf{b} = [1, 0].
\] (18)

Visualize the set of vectors where \( A\mathbf{x} \geq \mathbf{b} \). Hint: A matrix vector product is a collection of dot products, and the above set can be obtained by the intersection of two of the sets constructed in the previous questions.

**Solution** \( A\mathbf{x} = [3x_1 - 2x_2, 2x_1] \geq [1, 0] \), so it’s the intersection of \( 3x_1 - 2x_2 \geq 1 \) and \( x_1 \geq 0 \).