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</th>
<th>$f_w(x) := w \cdot \phi(x)$</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</td>
<td>$f_w(x) := \text{sign}(w \cdot \phi(x))$</td>
<td>0-1 loss: $1_{[f_w(x) \neq y]}$</td>
<td>Cannot use GD, SGD</td>
</tr>
<tr>
<td></td>
<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></td>
<td>Logistic loss: $\log(1 + e^{-(w \cdot \phi(x))y})$</td>
<td>GD or SGD</td>
</tr>
</tbody>
</table>

*Dimension check.* Above, $w, \phi(x) \in \mathbb{R}^d$, while $y$ is a scalar.
1) Problem 1: Non-linear features

Consider the following two training datasets of \((x, y)\) pairs:

- \(\mathcal{D}_1 = \{(-1, +1), (0, -1), (1, +1)\}\).
- \(\mathcal{D}_2 = \{(-1, -1), (0, +1), (1, -1)\}\).

Observe that neither dataset is linearly separable if we use \(\phi(x) = x\), so let’s fix that. Define a two-dimensional feature function \(\phi(x)\) such that:

- There exists a weight vector \(w_1\) that classifies \(\mathcal{D}_1\) perfectly (meaning that \(w_1 \cdot \phi(x) > 0\) if \(x\) is labeled \(+1\) and \(w_1 \cdot \phi(x) < 0\) if \(x\) is labeled \(-1\)); and
- There exists a weight vector \(w_2\) that classifies \(\mathcal{D}_2\) perfectly.

Note that the weight vectors can be different for the two datasets, but the features \(\phi(x)\) must be the same.

Some additional food for thought: Is every dataset linearly separable in some feature space? In other words, given pairs \((x_1, y_1), \ldots, (x_n, y_n)\), can we find a feature extractor \(\phi\) such that we can perfectly classify \((\phi(x_1), y_1), \ldots, (\phi(x_n), y_n)\) for some linear model \(w\)? If so, is this a good feature extractor to use?
2) Problem 2: Backpropagation

Consider the following function

\[
\text{Loss}(x, y, z, w) = 2(xy + \max\{w, z\})
\]

Run the backpropagation algorithm to compute the four gradients (each with respect to one of the individual variables) at \(x = 3\), \(y = -4\), \(z = 2\) and \(w = -1\). Use the following nodes: addition, multiplication, max, multiplication by a constant.
Problem 3: K-means

Consider doing ordinary $K$-means clustering with $K = 2$ clusters on the following set of 3 one-dimensional points:

$$\{-2, 0, 10\}. \quad (1)$$

Recall that $K$-means can get stuck in local optima. Describe the precise conditions on the initialization $\mu_1 \in \mathbb{R}$ and $\mu_2 \in \mathbb{R}$ such that running $K$-means will yield the global optimum of the objective function. Notes:

- Assume that $\mu_1 < \mu_2$.
- Assume that if in step 1 of $K$-means, no points are assigned to some cluster $j$, then in step 2, that centroid $\mu_j$ is set to $\infty$.
- Hint: try running $K$-means from various initializations $\mu_1, \mu_2$ to get some intuition; for example, if we initialize $\mu_1 = 1$ and $\mu_2 = 9$, then we converge to $\mu_1 = -1$ and $\mu_2 = 10$. 
4) [optional] Problem 4: Non-linear decision boundaries

Suppose we are performing classification where the input points are of the form \((x_1, x_2) \in \mathbb{R}^2\). We can choose any subset of the following set of features:

\[
\mathcal{F} = \left\{ x_1^2, x_2^2, x_1 x_2, x_1, x_2, \frac{1}{x_1}, \frac{1}{x_2}, 1, 1[x_1 \geq 0], 1[x_2 \geq 0] \right\}
\]  

(2)

For each subset of features \(F \subseteq \mathcal{F}\), let \(D(F)\) be the set of all decision boundaries corresponding to linear classifiers that use features \(F\).

For each of the following sets of decision boundaries \(E\), provide the minimal \(F\) such that \(D(F) \supseteq E\). If no such \(F\) exists, write ‘none’.

For example the set of features \(F = \{x_1^2, x_2\}\) allows the decision boundary of parabolas opening in the \(x_2\) axis, centered at the origin:

- \(E\) is all lines [CA hint]:

- \(E\) is all circles centered at the origin:

- \(E\) is all circles:

- \(E\) is all axis-aligned rectangles:

- \(E\) is all axis-aligned rectangles whose lower-right corner is at \((0, 0)\):