Prof. Dr. Sören Bartels, Tatjana Schreiber July 02th, 2025

Algorithmic Aspects of Data Analytics and Machine Learning

SS 2025 — Sheet 10

`https://aam.uni-freiburg.de/agba/lehre/ss25/algml/index.html`

Due: July 11, 2025, 2 p.m.

Task 1 (2 Points)
Compute the distance of the point $(1, 4, 0)$ from the plane in $\mathbb{R}^3$ defined by the equation $3x + 4y = 4$.

Task 2 (4 Points)
A data set $D = \{(x_i, y_i) \mid i = 1, \dots, n\} \subset \mathbb{R}^d \times \{-1, 1\}$, where not all labels are the same, is linearly separable if and only if the following associated quadratic optimization problem has at least one solution:

$$\min_{\lambda \in \mathbb{R}^n} \quad \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} \lambda_i \quad \text{s.t.} \quad \sum_{i=1}^{n} \lambda_i y_i = 0, \ \lambda_i \geq 0.$$

Consider the dataset

$$D := \left\{ (x_1, y_1) := \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, -1 \right), \ (x_2, y_2) := \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, 1 \right), \ (x_3, y_3) := \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, 1 \right) \right\}$$

consisting of three points in $\mathbb{R}^2$. Write down the quadratic optimization problem and compute a solution by hand. Deduce the separating hyperplane.

Task 3 (4 Points)
Let $D = \{(x_i, y_i) \mid i = 1, \dots, n\} \subset \mathbb{R}^d \times \{-1, 1\}$ be a dataset.
The margin condition requires that the distance of each data point from the decision boundary is at least $1/(2\|w\|)$, which leads to maximizing the margin $1/\|w\|$ under the constraint $y_i(\langle w, x_i \rangle + b) \geq 1$ for all i.
Show that imposing the normalization condition $\|w\| = 1$, instead of the margin condition, leads to the following optimization problem:

$$\operatorname*{arg\,max}_{(w,b) \in \mathbb{S}^{d-1} \times \mathbb{R}} \quad \min_{i=1,\dots,n} |\langle w, x_i \rangle + b| \quad \text{subject to } y_i(\langle w, x_i \rangle + b) > 0 \ \forall i,$$

where $\mathbb{S}^{d-1} = \{w \in \mathbb{R}^d \mid \|w\| = 1\}$ denotes the unit sphere in $\mathbb{R}^d$.

Task 4 (6 Points)
We consider the following one-dimensional dataset

$$D := \{(-3.5, 1), (0.5, 1), (0.75, 1), (-2.5, -1), (-1, -1), (5, -1)\} \subset \mathbb{R} \times \{-1, 1\},$$

where the first entry denotes the feature and the second entry the label.

(i) Sketch the dataset D.

(ii) Find a mapping $\psi : \mathbb{R} \to \mathbb{R}^2$ such that the mapped dataset

$$\hat{D} := \{(\psi(x), y) \mid (x, y) \in D\} \subset \mathbb{R}^2 \times \{-1, 1\}$$

is linearly separable, and sketch $\hat{D}$.

(iii) Determine a classifier of the form $h = \operatorname{sign}(\langle w, \cdot \rangle + b)$ for $\hat{D}$.

(iv) Specify which $x \in \mathbb{R}$ are classified as 1 and which as -1 by the induced classifier $h \circ \psi$.

Practical exercise

The following exercise is not mandatory; the points are bonus points that you can collect. Please submit your solutions as a MATLAB or Python file by July 10, 2 p.m., via email to tatjana.stiefken@mathematik.uni-freiburg.de. Please comment your code and your results.

Project (4* Points)

The following algorithm trains a simple feedforward neural network with one hidden layer to approximate a quadratic function of the form $y = x^2 + 2x + 1$. It uses the sigmoid activation function in the hidden layer and applies gradient descent to minimize the mean squared error between predicted and true values. After training, the model is evaluated on a test set to assess its approximation quality.
Implement the pseudo code in MatLab or Python. Use the sigmoid activation function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

and its derivative $\sigma'(z) = \sigma(z)(1 - \sigma(z))$. Try different values for n_{in} and n_{train}. Plot Y_{test} and Y_{pred}. What do you see?

Algorithm 1 Training a Simple Feedforward Neural Network

```
 1: **Input:** Input size $n_{\text{in}} = 1$, hidden layer size $n_{\text{hid}} = 10$, output size $n_{\text{out}} = 1$
 2: **Hyperparameters:** Learning rate $\eta = 0.01$, number of epochs $N_{\text{epochs}} = 1000$
 3: **Training data:** Generate $X_{\text{train}}$ as $n_{\text{train}}$ discrete values in the interval $[-10, 10]$
 4: **Target values:** $Y_{\text{train}} = x^2 + 2x + 1$ for $x \in X_{\text{train}}$
 5: Initialize $W_1 \in \mathbb{R}^{n_{\text{hid}} \times n_{\text{in}}}$, $W_2 \in \mathbb{R}^{n_{\text{out}} \times n_{\text{hid}}}$, $b_1 \in \mathbb{R}^{n_{\text{hid}}}$, $b_2 \in \mathbb{R}^{n_{\text{out}}}$ randomly
 6: **for** each epoch $e = 1$ to $N_{\text{epochs}}$ **do**
 7:     **Forward pass:**
 8:         $Z_1 \leftarrow W_1 X_{\text{train}} + b_1$,    $A_1 \leftarrow \sigma(Z_1)$
 9:         $Z_2 \leftarrow W_2 A_1 + b_2$,    $A_2 \leftarrow Z_2$
10:     **Compute loss:**
11:         $\mathcal{L} \leftarrow \frac{1}{m} \sum_{i=1}^{m} \|A_2^{(i)} - Y_{\text{train}}^{(i)}\|^2$
12:     **Backpropagation:**
13:         $dZ_2 \leftarrow (A_2 - Y_{\text{train}})$,  $dW_2 \leftarrow \frac{1}{m} dZ_2 A_1^\top$,  $db_2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} dZ_2^{(i)}$
14:         $dZ_1 \leftarrow (W_2^\top dZ_2) \circ \sigma'(Z_1)$,  $dW_1 \leftarrow \frac{1}{m} dZ_1 X_{\text{train}}^\top$,  $db_1 \leftarrow \frac{1}{m} \sum_{i=1}^{m} dZ_1^{(i)}$
15:     **Gradient Descent Step:**
16:         $W_2 \leftarrow W_2 - \eta \cdot dW_2$
17:         $b_2 \leftarrow b_2 - \eta \cdot db_2$
18:         $W_1 \leftarrow W_1 - \eta \cdot dW_1$
19:         $b_1 \leftarrow b_1 - \eta \cdot db_1$
20: **end for**
21: **Testing phase:**
22: Generate test data $X_{\text{test}}$ in $[-15, 15]$ and targets $Y_{\text{test}} = x^2 + 2x + 1$
23: Perform forward pass on test data to obtain predictions $Y_{\text{pred}}$
24: Compute mean squared error: $E = \frac{1}{m} \sum (Y_{\text{pred}} - Y_{\text{test}})^2$.
```