Assignment 13

Yehonatan Keypur

March 13, 2025

1 Problem 16: Neural Networks

Let V and Q be the number of nodes and weights in a feedforward neural network:

$$V = \sum_{l=0}^{L} d^{(l)}, \quad Q = \sum_{l=1}^{L} d^{(l)} \left(d^{(l-1)} + 1\right),$$

where $d^{(l)}$ is the input vector dimension of layer l.

(a) Forward Propagation Complexity

During forward propagation, the operations consist of matrix-vector multiplications, bias additions, and activation function evaluations.

- **Multiplications:** For each layer l, there are $d^{(l)} \times d^{(l-1)}$ weights. This requires $\sum_{l=1}^{L} d^{(l)} d^{(l-1)}$ multiplications.
- **Additions:** For each layer l, there are $d^{(l)}$ neurons, and each neuron computes a weighted sum over $d^{(l-1)}$ inputs. This results in $\sum_{l=1}^{L} d^{(l)} d^{(l-1)}$ additions.
- **Evaluations of θ:** Each neuron applies the activation function θ. Therefore, there are $\sum_{l=1}^{L} d^{(l)}$ evaluations of θ.

In terms of Q and V:

- Multiplications: $\mathcal{O}(Q)$,
- Additions: $\mathcal{O}(Q)$,
- Evaluations of θ: $\mathcal{O}(V)$.

(b) Backpropagation Complexity

Backpropagation involves calculating the gradient of the loss function with respect to weights and biases using the chain rule. The computations include:

- **Multiplications:** Gradients with respect to weights require multiplications of errors with inputs or activations from the previous layer. This involves Q multiplications.
- **Additions:** Similar to forward propagation, Q additions are required for summing terms.
- **Evaluations of θ:** For logistic sigmoid $\theta(x) = \frac{1}{1+e^{-x}}$, its derivative is $\theta'(x) = \theta(x)(1-\theta(x))$. The value of $\theta(x)$ was already computed during forward propagation. Thus, no additional evaluations of θ are needed in backpropagation.

Thus, the computational complexity for backpropagation matches that of forward propagation:

- Multiplications: $\mathcal{O}(Q)$,
- Additions: $\mathcal{O}(Q)$,
- Evaluations of θ: 0.

(c) Finite Difference Approximation

The finite difference approximation for obtaining the gradient is given by:

$$\frac{\partial h}{\partial w_{ij}^{(l)}} = \frac{h(w_{ij}^{(l)} + \varepsilon) - h(w_{ij}^{(l)} - \varepsilon)}{2\varepsilon} + O(\varepsilon^2),$$

where $h(w_{ij}^{(l)})$ denotes the function output when all weights are fixed except $w_{ij}^{(l)}$, which is perturbed.

To compute the gradient for each weight $w_{ij}^{(l)}$:

- For each perturbation, forward propagation is performed twice (once for $+\varepsilon$ and once for $-\varepsilon$).
- There are Q weights, so 2 forward propagations are required for one gradient computation.

The computational complexity is $O(Q^2)$ because each of the Q weights requires two forward propagations (each $O(Q)$), resulting in $O(Q \cdot Q) = O(Q^2)$.

2 Problem 18: K-Means

(a) Showing that E_{in} is monotonically decreasing

Lloyd's K-means algorithm alternates between two steps: assignment (step 2) and update (step 3). We analyze the effect of each step on E_{in}:

Assignment Step:

In this step, each data point x_n is reassigned to the cluster S_j with the nearest centroid μ_j. Let $r_{nk}^{(t)}$ denote the assignment at iteration t, and $r_{nk}^{(t+1)}$ denote the new assignment. Since $r_{nk}^{(t+1)}$ corresponds to the closest centroid for each x_n, we have:

$$\sum_{k=1}^{K} r_{nk}^{(t+1)} \|x_n - \mu_k^{(t)}\|^2 \leq \sum_{k=1}^{K} r_{nk}^{(t)} \|x_n - \mu_k^{(t)}\|^2 \quad \forall n.$$

Summing over all data points:

$$E'(S_1^{(t+1)}, \dots, S_K^{(t+1)}, \mu_1^{(t)}, \dots, \mu_K^{(t)}) \leq E_{in}^{(t)}.$$

Here, E' is the intermediate error after reassignment but before updating centroids.

Update Step:

Each centroid $\mu_j^{(t+1)}$ is updated to the mean (centroid) of its assigned points $S_j^{(t+1)}$. The centroid minimizes the sum of squared distances within its cluster:

$$\mu_j^{(t+1)} = \frac{1}{|S_j^{(t+1)}|} \sum_{x_n \in S_j^{(t+1)}} x_n.$$

By the properties of the centroid:

$$\sum_{x_n \in S_j^{(t+1)}} \|x_n - \mu_j^{(t+1)}\|^2 \leq \sum_{x_n \in S_j^{(t+1)}} \|x_n - \mu_j^{(t)}\|^2 \quad \forall j.$$

Summing over all clusters:

$$E_{in}^{(t+1)} \leq E'.$$

Combining both steps:

$$E_{in}^{(t+1)} \leq E' \leq E_{in}^{(t)}.$$

Thus, E_{in} is non-increasing (monotonically decreasing) at each iteration.

(b) Algorithm Termination

Since E_{in} is monotonically decreasing and bounded below by 0 (as it is a sum of non-negative terms), by the Monotone Convergence Theorem, E_{in} must converge to a limit. The algorithm terminates when E_{in} stops decreasing, i.e., when $E_{in}^{(t+1)} = E_{in}^{(t)}$.

To see why termination occurs in finite steps, observe that:

- There are finitely many ways to partition N data points into K clusters (though exponentially many).
- Each distinct partition corresponds to a unique value of E_{in}. If the algorithm revisits a partition, E_{in} would remain unchanged, forcing termination.
- Since E_{in} strictly decreases with each non-convergent iteration, the algorithm cannot cycle indefinitely.

Thus, the algorithm must terminate after a finite number of iterations when E_{in} stabilizes.

□

3 Problem 19: PCA

Base Case (M = 1):
For $M = 1$, the direction $\mathbf{u}_1$ maximizing the variance $\mathbf{u}_1^\top S\mathbf{u}_1$ is the eigenvector of S associated with the largest eigenvalue λ_1. This is proven using Lagrange multipliers to enforce $\mathbf{u}_1^\top \mathbf{u}_1 = 1$.

Inductive Step:
Assume the result holds for an M-dimensional subspace spanned by orthonormal eigenvectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$ corresponding to the M largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_M$. We prove it for $M + 1$.

Objective and Constraints:
Maximize the variance for $\mathbf{u}_{M+1}$ orthogonal to $\{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$ and normalized:

$$J = \mathbf{u}_{M+1}^\top S\mathbf{u}_{M+1},$$

subject to:

$$\mathbf{u}_{M+1}^\top \mathbf{u}_i = 0 \quad \forall i = 1, \ldots, M, \quad \text{and} \quad \mathbf{u}_{M+1}^\top \mathbf{u}_{M+1} = 1.$$

Lagrangian:

$$\mathcal{L} = \mathbf{u}_{M+1}^\top S\mathbf{u}_{M+1} - \lambda(\mathbf{u}_{M+1}^\top \mathbf{u}_{M+1} - 1) - \sum_{i=1}^{M} \mu_i \mathbf{u}_{M+1}^\top \mathbf{u}_i.$$

Optimality Condition:
Setting the derivative of $\mathcal{L}$ w.r.t. $\mathbf{u}_{M+1}$ to zero:

$$2S\mathbf{u}_{M+1} - 2\lambda\mathbf{u}_{M+1} - \sum_{i=1}^{M} \mu_i \mathbf{u}_i = 0 \implies S\mathbf{u}_{M+1} = \lambda\mathbf{u}_{M+1} + \frac{1}{2}\sum_{i=1}^{M} \mu_i \mathbf{u}_i.$$

Enforcing Orthogonality:
Taking inner product with $\mathbf{u}_j$ ($j \leq M$):

$$\mathbf{u}_j^\top S\mathbf{u}_{M+1} = \frac{1}{2}\mu_j.$$

Since $S\mathbf{u}_j = \lambda_j \mathbf{u}_j$ and $\mathbf{u}_j^\top \mathbf{u}_{M+1} = 0$, we find $\mu_j = 0$.

Thus,

$$S\mathbf{u}_{M+1} = \lambda\mathbf{u}_{M+1}.$$

This shows that $\mathbf{u}_{M+1}$ is an eigenvector of S.

Maximizing Variance:
Choose $\mathbf{u}_{M+1}$ as the eigenvector corresponding to the next largest eigenvalue λ_{M+1}. Hence, the optimal $(M + 1)$-dimensional subspace uses the top $M + 1$ eigenvectors.

Conclusion:
By induction, PCA maximizes variance by projecting onto the eigenvectors of S with the largest eigenvalues.

4 Problem 10: SVM

Question a

The dual SVM optimization problem is:

$$\max_{\alpha} -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m K_{nm} + \sum_{n=1}^{N} \alpha_n,$$

subject to:

$$\sum_{n=1}^{N} y_n \alpha_n = 0, \quad 0 \leq \alpha_n \leq C \quad \text{for } n = 1, \ldots, N.$$

Here, $K_{nm} = x_n^T x_m$ is the kernel matrix.

When optimizing over α_1 and α_2, the terms involving $\alpha_{n\geq 3}$ are constant. The objective function simplifies to:

$$\psi_1(\alpha_1, \alpha_2) = -\frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}\alpha_1\alpha_2 - y_1\alpha_1 v_1 - y_2\alpha_2 v_2 + \alpha_1 + \alpha_2,$$

where:

$$s = y_1 y_2 \in \{-1, +1\}, \quad v_i = \sum_{j=3}^{N} y_j \alpha_j K_{ij} \quad \text{for } i = 1, 2.$$

The equality constraint $\sum_{n=1}^{N} y_n \alpha_n = 0$ becomes:

$$y_1\alpha_1 + y_2\alpha_2 = -\sum_{n=3}^{N} y_n \alpha_n.$$

Multiply both sides by y_1 and use $s = y_1 y_2$:

$$\alpha_1 + s\alpha_2 = \gamma,$$

where:

$$\gamma = -y_1 \sum_{n=3}^{N} y_n \alpha_n.$$

Question b

Substitute $\alpha_1 = \gamma - s\alpha_2$ into $\psi_1(\alpha_1, \alpha_2)$ to obtain $\psi_2(\alpha_2)$:

$$\psi_2(\alpha_2) = -\frac{1}{2}K_{11}(\gamma - s\alpha_2)^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}(\gamma - s\alpha_2)\alpha_2 - y_1(\gamma - s\alpha_2)v_1 - y_2\alpha_2 v_2 + (\gamma - s\alpha_2) + \alpha_2.$$

Expand the quadratic terms:

$$\psi_2(\alpha_2) = -\frac{1}{2}K_{11}(\gamma^2 - 2s\gamma\alpha_2 + \alpha_2^2) - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}(\gamma\alpha_2 - s\alpha_2^2) - y_1\gamma v_1 + y_1 s\alpha_2 v_1 - y_2\alpha_2 v_2 + \gamma - s\alpha_2 + \alpha_2.$$

Collect terms involving α_2:

$$\psi_2(\alpha_2) = -\frac{1}{2}K_{11}\gamma^2 + sK_{11}\gamma\alpha_2 - \frac{1}{2}K_{11}\alpha_2^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}\gamma\alpha_2 + K_{12}\alpha_2^2 - y_1\gamma v_1 + y_1 s\alpha_2 v_1 - y_2\alpha_2 v_2 + \gamma + (1-s)\alpha_2.$$

Take the derivative with respect to α_2 and set it to zero:

$$\frac{d\psi_2}{d\alpha_2} = sK_{11}\gamma - K_{11}\alpha_2 - K_{22}\alpha_2 - sK_{12}\gamma + 2K_{12}\alpha_2 + y_1 s v_1 - y_2 v_2 + (1-s) = 0.$$

yielding:

$$\frac{d\psi_2}{d\alpha_2} = -\eta\alpha_2 + s\gamma(K_{11} - K_{12}) + y_2(v_1 - v_2) + (1-s) = 0,$$

where $\eta = K_{11} + K_{22} - 2K_{12}$.

Solve for α_2:

$$\alpha_2 = \frac{s\gamma(K_{11} - K_{12}) + y_2(v_1 - v_2) + (1-s)}{\eta}.$$

Question c

The decision function is:

$$f(x_i) = \sum_{n=1}^{N} y_n \alpha_n^* K_{ni} + b^*,$$

where α_n^* are the Lagrange multipliers before optimization.

Subtract $f(x_1)$ and $f(x_2)$:

$$f(x_1) - f(x_2) = \sum_{n=1}^{N} y_n \alpha_n^* (K_{n1} - K_{n2}).$$

Separate the terms involving α_1^* and α_2^*:

$$f(x_1) - f(x_2) = y_1\alpha_1^*(K_{11} - K_{12}) + y_2\alpha_2^*(K_{21} - K_{22}) + \sum_{n=3}^{N} y_n \alpha_n^* (K_{n1} - K_{n2}).$$

Using $v_i = \sum_{n=3}^{N} y_n \alpha_n^* K_{ni}$, we get:

$$f(x_1) - f(x_2) = y_1\alpha_1^*(K_{11} - K_{12}) + y_2\alpha_2^*(K_{12} - K_{22}) + (v_1 - v_2).$$

Solve for $v_1 - v_2$:

$$v_1 - v_2 = f(x_1) - f(x_2) - y_1\alpha_1^*(K_{11} - K_{12}) - y_2\alpha_2^*(K_{12} - K_{22}).$$

Substitute $\alpha_1^* = \gamma - s\alpha_2^*$:

$$v_1 - v_2 = f(x_1) - f(x_2) - y_1(\gamma - s\alpha_2^*)(K_{11} - K_{12}) - y_2\alpha_2^*(K_{12} - K_{22}).$$

Simplify:

$$v_1 - v_2 = f(x_1) - f(x_2) + y_2\alpha_2^*\eta - sy_2\gamma(K_{11} - K_{12}).$$

Question d

Substitute $v_1 - v_2$ into the α_2 expression:

$$\alpha_2 = \alpha_2^* + \frac{y_2\left[(y_2 - f(x_2)) - (y_1 - f(x_1))\right]}{\eta}.$$

Starting from the expression for α_2 derived in Question 2:

$$\alpha_2 = \frac{s(K_{11} - K_{12})\gamma + \mathcal{Y}_2(v_1 - v_2) - s + 1}{\eta}$$

Substitute $v_1 - v_2$ from Question 3:

$$v_1 - v_2 = f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2) + \alpha_2^* \mathcal{Y}_2 \eta - s\mathcal{Y}_2\gamma(K_{11} - K_{12})$$

Plugging this into the numerator:

$$\begin{aligned}
\alpha_2 &= \frac{s(K_{11} - K_{12})\gamma + \mathcal{Y}_2 \left[f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2) + \alpha_2^* \mathcal{Y}_2 \eta - s\mathcal{Y}_2\gamma(K_{11} - K_{12})\right] - s + 1}{\eta} \\
&= \frac{s(K_{11} - K_{12})\gamma + \mathcal{Y}_2(f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)) + \alpha_2^* \eta - s\gamma(K_{11} - K_{12}) - s + 1}{\eta} \\
&= \alpha_2^* + \frac{\mathcal{Y}_2(f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)) - s + 1}{\eta}.
\end{aligned}$$

Recognizing $s = \mathcal{Y}_1\mathcal{Y}_2$ and rearranging terms involving the errors $E_i = f(\boldsymbol{x}_i) - \mathcal{Y}_i$:

$$\alpha_2 = \alpha_2^* + \frac{\mathcal{Y}_2 \left[(\mathcal{Y}_2 - f(\boldsymbol{x}_2)) - (\mathcal{Y}_1 - f(\boldsymbol{x}_1))\right]}{\eta}.$$

This matches the desired update rule.

Question e

Why Clipping is Required:

The unconstrained α_2 may violate the box constraints $0 \leq \alpha_1, \alpha_2 \leq C$ and the linear constraint $\alpha_1 + s\alpha_2 = \gamma$. Clipping ensures α_2 lies within the feasible region defined by these constraints. Moreover, it is required to prevent overfitting and to increase the generalizability of the model by maintaining feasible and well-regularized parameter values.

Deriving L and H for $s = +1$:

When $s = \mathcal{Y}_1\mathcal{Y}_2 = +1$, the linear constraint becomes $\alpha_1 + \alpha_2 = \gamma$. Substituting $\alpha_1 = \gamma - \alpha_2$ into the box constraints:

$$0 \leq \gamma - \alpha_2 \leq C \quad \Rightarrow \quad \gamma - C \leq \alpha_2 \leq \gamma,$$

$$0 \leq \alpha_2 \leq C.$$

Combining these:

- **Lower Bound (L)**: $\max(0, \gamma - C)$ ensures $\alpha_2 \geq 0$ and $\alpha_1 \leq C$.
- **Upper Bound (H)**: $\min(C, \gamma)$ ensures $\alpha_2 \leq C$ and $\alpha_1 \geq 0$.

Thus, for $s = +1$, α_2 is constrained to $[L, H] = [\max(0, \gamma - C), \min(C, \gamma)]$. Clipping α_2 to this interval ensures all constraints are satisfied.

—

Thank you for reviewing this assignment.