IML Assignment 2A

Aman K. Foujdar

May 12, 2025

Question 1

Showing that $\phi(x) \cdot \phi(x') = e^{-\frac{(x-x')^2}{2}}$.

We know that: A kernel $K(x, x')$ is valid (i.e., a positive semi-definite kernel) if:

$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

for some feature map ϕ into a (possibly infinite-dimensional) Hilbert space.

We are given the infinite-dimensional feature map:

$$\phi_i(x) = \frac{1}{\sqrt{i!}} e^{-x^2/2} x^i, \quad \text{for } i = 0, 1, 2, \dots$$

We want to compute the dot product between $\phi(x)$ and $\phi(x')$:

$$\phi(x) \cdot \phi(x') = \sum_{i=0}^{\infty} \phi_i(x) \cdot \phi_i(x')$$

Substituting the definition of ϕ_i, we get:

$$\phi(x) \cdot \phi(x') = \sum_{i=0}^{\infty} \left(\frac{1}{\sqrt{i!}} e^{-x^2/2} x^i \right) \left(\frac{1}{\sqrt{i!}} e^{-x'^2/2} x'^i \right)$$

$$= e^{-x^2/2} e^{-x'^2/2} \sum_{i=0}^{\infty} \frac{(xx')^i}{i!}$$

The sum is the Taylor expansion of the exponential function:

$$\sum_{i=0}^{\infty} \frac{(xx')^i}{i!} = e^{xx'}$$

Therefore:

$$\phi(x) \cdot \phi(x') = e^{-x^2/2} e^{-x'^2/2} e^{xx'} = e^{-\frac{x^2+x'^2}{2} + xx'}$$

Now simplify the exponent:

$$-\frac{x^2 + x'^2}{2} + xx' = -\frac{1}{2}(x^2 - 2xx' + x'^2) = -\frac{(x-x')^2}{2}$$

Hence, we obtain:

$$\phi(x) \cdot \phi(x') = e^{-\frac{(x-x')^2}{2}} = K(x, x')$$

Conclusion: What we basically did was, we explicitly constructed a feature map $\phi(x)$ and we showed that the inner product of feature vectors $\phi(x).\phi(x')$ equals the given kernel function.

The Gaussian (RBF) kernel is a valid kernel corresponding to the dot product of the infinite-dimensional feature maps:

$$\boxed{K(x, x') = \phi(x) \cdot \phi(x') = e^{-\frac{(x-x')^2}{2}}}$$

Question 2

The goal of a perceptron in this case is to find any separating hyperplane that correctly classifies all training data points. If the data is linearly separable, the Perceptron algorithm is guaranteed to find a separating hyperplane in a finite number of iterations (Perceptron Convergence Theorem). However, there can be infinitely many such hyperplanes.

Whereas, the goal of a Hard Margin Linear SVM is to find the separating hyperplane that maximizes the geometric margin. This margin is defined as the smallest distance from any data point to the hyperplane. This yields a unique optimal separating hyperplane (assuming the data is strictly linearly separable)

We have two major advantages because fo this:

- Better Generalization: A larger margin intuitively means that the decision boundary is "further away" from the data points. This makes the classifier less sensitive to small variations or noise in the training data and is more likely to correctly classify new, unseen data points.
- Uniqueness of the Solution: For a strictly linearly separable dataset, the hard-margin SVM provides a unique optimal solution (the maximum-margin hyperplane).

Equivalence of Hard-Margin SVM Formulations

We consider the following two formulations of the hard-margin linear SVM:

Formulation A (Standard Primal Form):

$$\min_{w,b} \|w\|^2 \quad \text{subject to } y_i(w^\top x_i + b) \geq 1 \quad \forall i$$

Formulation B (Normalized Margin Form):

$$\min_{w,b} \|w\|^2 \quad \text{subject to } y_i(w^\top x_i + b) \geq 0 \quad \forall i, \quad \text{and } \min_i y_i|w^\top x_i + b| = 1$$

Note: There is some latex error that I'm getting. So I'm not able to put a transpose over w^*. So please assume that my w^* basically means 'w^* transposed'.

(i) The optimal solution to A is feasible for B

Let (w^*, b^*) be the optimal solution to Formulation A. Then for all i:

$$y_i(w^* \cdot x_i + b^*) \geq 1$$

In particular,

$$\min_i y_i|w^* \cdot x_i + b^*| \geq 1$$

This means all constraints of Formulation B are satisfied:

$$y_i(w^* \cdot x_i + b^*) \geq 0 \quad \text{and} \quad \min_i y_i(w^* \cdot x_i + b^*) \geq 1$$

So (w^*, b^*) is a feasible solution for B.

(ii) The optimal solution to B is feasible for A

Let (w^*, b^*) be the optimal solution to Formulation B. Then:

$$y_i(w^* \cdot x_i + b^*) \geq 0 \quad \forall i, \quad \text{and} \quad \min_i y_i(w^* \cdot x_i + b^*) = 1$$

Hence, for all i:

$$y_i(w^* \cdot x_i + b^*) \geq 1$$

So (w^*, b^*) satisfies all constraints of Formulation A and is thus feasible for A.

(iii) Optimal solution of A equals optimal solution of B

We have shown that the optimal solution of A is feasible in B, and the optimal solution of B is feasible in A. Since both formulations minimize the same objective function $\|w\|^2$:

$$\|w_A\|^2 \leq \|w_B\|^2 \quad \text{and} \quad \|w_B\|^2 \leq \|w_A\|^2$$

This implies:

$$\|w_A\|^2 = \|w_B\|^2$$

Hence, the solutions must be the same:

$$(w_A, b_A) = (w_B, b_B)$$

Conclusion: The optimal solutions to Formulations A and B are equivalent.

Question 3

Convergence of Hard-Margin and Soft-Margin SVMs

Hard-margin SVM: The hard-margin SVM requires that the data be perfectly linearly separable, i.e., there must exist a hyperplane such that:

$$y_i(w^T x_i + b) \geq 1 \quad \forall i$$

If the data is **not linearly separable** (as we can see in the fugure in the asisgnment), this constraint cannot be satisfied for all i, and thus, the optimization problem becomes infeasible. Therefore, the hard-margin SVM does **not converge** for non-separable data.

Soft-margin SVM: The soft-margin SVM introduces slack variables $\xi_i \geq 0$ to allow for constraint violations:

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

The objective now becomes:

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C \sum_i \xi_i$$

This formulation is almost always feasible, even when the data is not linearly separable. It allows some misclassifications, depending on the choice of the hyperparameter C.

Effect of Hyperparameter C in Soft-Margin SVM

As $C \to \infty$:

- The penalty for misclassification becomes very high and the optimizer strongly discourages violations of the margin.
- The soft-margin SVM behaves increasingly like the hard-margin SVM. In short, if the data is linearly separable, it will converge to the same solution as the hard-margin SVM.

As $C \to 0^+$:

- The penalty for margin violations is negligible. In this case, the optimizer may allow many violations to minimize $\|w\|^2$.
- The resulting classifier may have a very wide margin but poor classification performance.

Proof

Given the optimal solution (w, b) of the hard-margin SVM, the constraints are:

$$y_i(w^T x_i + b) \geq 1 \quad \forall i$$

Suppose, for contradiction, that $y_i(w^T x_i + b) > 1$ for all i. Then we can scale w and b by a factor $\alpha \in (0, 1)$:

$$\tilde{w} = \alpha w, \quad \tilde{b} = \alpha b$$

This scaling would yield:

$$y_i(\tilde{w}^T x_i + \tilde{b}) = \alpha y_i(w^T x_i + b) > \alpha \cdot 1 = 1$$

Hence, all constraints are still satisfied, and we have:

$$\|\tilde{w}\|^2 = \alpha^2 \|w\|^2 < \|w\|^2$$

This contradicts the optimality of (w, b), since $(\tilde{w}, \tilde{b})$ achieves a lower objective value while satisfying all constraints.

Therefore, there must exist at least one point x_i for which:

$$y_i(w^T x_i + b) = 1$$

This implies:

- For some i with $y_i = 1$, we have $w^T x_i + b = 1$
- For some j with $y_j = -1$, we have $w^T x_j + b = -1$

Thus, at least one data point lies on each of the margin hyperplanes:

$$\{x : w^T x + b = 1\} \quad \text{and} \quad \{x : w^T x + b = -1\}$$

Question 4

ℓ_2-Norm Soft Margin SVM

We are given the optimization problem for the ℓ_2-norm soft margin SVM:

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i^2$$

$$\text{subject to} \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \text{for } i = 1, 2, \ldots, n$$

Why Can We Drop the Constraint $\xi_i \geq 0$?
Assume for contradiction that $\xi_i < 0$ for some i. Then the constraint becomes

$$y_i(w^T x_i + b) \geq 1 - \xi_i > 1,$$

which is easier to satisfy. However, the term ξ_i^2 in the objective will increase as ξ_i becomes more negative. Therefore, we can always increase ξ_i to zero (or a small positive value), satisfying the constraint while reducing the objective value.

In other words, what i've understood is that at the optimum, $\xi_i \geq 0$ must hold naturally, so explicitly including the constraint $\xi_i \geq 0$ is unnecessary. The optimal solution remains unchanged whether or not the constraint is included.

Lagrangian of the Problem

We'll now introduce Lagrange multipliers $\alpha_i \geq 0$ for each constraint. The Lagrangian is:

$$\mathcal{L}(w, b, \xi, \alpha) = \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\left[y_i(w^T x_i + b) - 1 + \xi_i\right]$$

Gradient Conditions

We compute the gradients with respect to w, b, and ξ_i, and set them to zero.

Gradient with respect to w:

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n}\alpha_i y_i x_i$$

Gradient with respect to b:

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

Gradient with respect to ξ_i:

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = \frac{2C}{n}\xi_i - \alpha_i = 0 \quad \Rightarrow \quad \xi_i = \frac{n}{2C}\alpha_i$$

Since, in the question we're given that $\xi = (\xi_1, \ldots, \xi_n)^T$, we can write:

$$\xi = \frac{n}{2C}\alpha$$

Question 5

Ridge Regression (Closed-form Solution)

We are given the cost function for ridge regression as:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(y_i - \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$$

In vector/matrix notation, this becomes:

$$J(\boldsymbol{\theta}) = \|\boldsymbol{y} - X\boldsymbol{\theta}\|^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$$

where:

- $X \in \mathbb{R}^{n\times d}$ is the matrix whose rows are $\boldsymbol{x}_i^\top$,
- $\boldsymbol{y} \in \mathbb{R}^n$ is the output vector,
- $\boldsymbol{\theta} \in \mathbb{R}^d$ is the weight vector.

Taking the gradient with respect to $\boldsymbol{\theta}$ and setting it to zero:

$$\nabla_{\boldsymbol{\theta}} J = -2X^\top(\boldsymbol{y} - X\boldsymbol{\theta}) + \lambda\boldsymbol{\theta} = 0$$

$$\Rightarrow (X^\top X + \lambda I)\boldsymbol{\theta} = X^\top \boldsymbol{y}$$

Hence, the closed-form solution is:

$$\boxed{\boldsymbol{\theta} = (X^\top X + \lambda I)^{-1} X^\top \boldsymbol{y}}$$

Kernel Ridge Regression and the Kernel Trick

Suppose we define a feature map ϕ, and we want to work in the feature space without computing ϕ explicitly.

As we're mentioned in the question, we can assume that $\boldsymbol{\theta}$ lies in the span of the input features:

$$\boldsymbol{\theta} = \sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i) = \Phi^\top \boldsymbol{\alpha}$$

where $\Phi \in \mathbb{R}^{D\times n}$ is the matrix whose columns are $\phi(\boldsymbol{x}_i)$, and $\boldsymbol{\alpha} \in \mathbb{R}^n$.

Then the prediction on a new data point $\boldsymbol{x}_{\text{new}}$ is:

$$\hat{y}_{\text{new}} = \boldsymbol{\theta}^\top \phi(\boldsymbol{x}_{\text{new}}) = \left(\sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i)\right)^\top \phi(\boldsymbol{x}_{\text{new}}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_{\text{new}})$$

We now solve for $\boldsymbol{\alpha}$ by minimizing the kernelized ridge cost:

$$J(\boldsymbol{\alpha}) = \left\|\boldsymbol{y} - \Phi^\top \boldsymbol{\alpha}\right\|^2 + \frac{\lambda}{2}\|\Phi^\top \boldsymbol{\alpha}\|^2$$

Define the kernel matrix $K \in \mathbb{R}^{n\times n}$ as:

$$K_{ij} = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$$

Then $K = \Phi^T \Phi$, and the cost function becomes:

$$J(\boldsymbol{\alpha}) = (\boldsymbol{y} - K\boldsymbol{\alpha})^T(\boldsymbol{y} - K\boldsymbol{\alpha}) + \frac{\lambda}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha}$$

Taking the gradient and setting it to zero:

$$\nabla_{\boldsymbol{\alpha}} J = -2K(\boldsymbol{y} - K\boldsymbol{\alpha}) + \lambda K \boldsymbol{\alpha} = 0$$

Expanding and rearranging:

$$-2K\boldsymbol{y} + 2K^2\boldsymbol{\alpha} + \lambda K\boldsymbol{\alpha} = 0$$

$$K(2K\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha} - 2\boldsymbol{y}) = 0$$

Assuming K is positive semidefinite and invertible:

$$2K\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha} = 2\boldsymbol{y}$$

$$(K + \frac{\lambda}{2}I)\boldsymbol{\alpha} = \boldsymbol{y}$$

Thus, the solution is:

$$\boxed{\boldsymbol{\alpha} = \left(K + \frac{\lambda}{2}I\right)^{-1}\boldsymbol{y}}$$

One thing to note is that if the regularization term does not have the factor of $\frac{1}{2}$, the standard solution becomes:

$$\boxed{\boldsymbol{\alpha} = (K + \lambda I)^{-1}\boldsymbol{y}}$$

So, the prediction becomes:

$$\boxed{\hat{y}_{\text{new}} = \boldsymbol{k}_{\text{new}}^\top (K + \lambda I)^{-1}\boldsymbol{y}}$$

where:

$$\boldsymbol{k}_{\text{new}} = [K(\boldsymbol{x}_1, \boldsymbol{x}_{\text{new}}), \ldots, K(\boldsymbol{x}_n, \boldsymbol{x}_{\text{new}})]^\top \in \mathbb{R}^n$$

This is how we can use the "kernel trick" to obtain a closed-form expression for the prediction on the new input without ever explicitly computing ϕ_{new}.

Kernel Trick and Prediction

Once $\boldsymbol{\alpha}$ is known, the prediction on a new input $\boldsymbol{x}_{\text{new}}$ is:

$$\hat{y}_{\text{new}} = \boldsymbol{\theta}^T\phi(\boldsymbol{x}_{\text{new}}) = \sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_{\text{new}}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_{\text{new}})$$

Question 6:

Decision Boundary

We are given:

$$W = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

The first-layer pre-activations are:

$$h_1 = x_1 - x_2, \quad h_2 = -x_1 - x_2$$

After applying the ReLU activation:

$$f(h_1) = \max(0, x_1 - x_2), \quad f(h_2) = \max(0, -x_1 - x_2)$$

The second-layer input:

$$z = v_1 f(h_1) + v_2 f(h_2) + v_3 \cdot 1 = f(h_1) + f(h_2) - 2$$

Set the decision boundary:

$$\sigma(z) = \frac{1}{1 + e^{-z}} = 0.5 \Rightarrow z = 0 \Rightarrow f(h_1) + f(h_2) = 2$$

We want to find the boundary where $\sigma(z) = 0.5$, which means $z = 0$. Recall that:

$$z = \max(0, x_1 - x_2) + \max(0, -x_1 - x_2) - 2$$

We consider four cases:

Region A: Both ReLUs active

This means:

$$x_1 - x_2 > 0 \quad \text{and} \quad -x_1 - x_2 > 0 \quad \Rightarrow \quad x_1 > x_2 \quad \text{and} \quad x_1 < -x_2$$

So:

$$x_2 < x_1 < -x_2 \quad \Rightarrow \quad \text{This region exists only if} \quad x_2 < 0$$

In this region:

$$f(h_1) = x_1 - x_2, \quad f(h_2) = -x_1 - x_2 \quad \Rightarrow \quad f(h_1) + f(h_2) = -2x_2$$

Set:

$$-2x_2 = 2 \quad \Rightarrow \quad x_2 = -1$$

•Line 1: $x_2 = -1$ in region where both ReLUs are active

Region B: Only $f(h_1)$ active

$$x_1 - x_2 > 0 \quad \text{and} \quad -x_1 - x_2 \leq 0 \quad \Rightarrow \quad x_1 > x_2 \quad \text{and} \quad x_1 \geq -x_2$$

Which simplifies to:

$$x_1 > x_2, \quad x_1 \geq -x_2 \quad \Rightarrow \quad \text{This region exists if} \quad x_1 > \max(x_2, -x_2)$$

In this region:

$$f(h_1) = x_1 - x_2, \quad f(h_2) = 0 \quad \Rightarrow \quad f(h_1) + f(h_2) = x_1 - x_2 \quad \Rightarrow \quad x_1 - x_2 = 2 \quad \Rightarrow \quad x_1 = x_2 + 2$$

•Line 2: $x_1 = x_2 + 2$ in this region

Region C: Only $f(h_2)$ active

$$x_1 - x_2 \leq 0 \quad \text{and} \quad -x_1 - x_2 > 0 \quad \Rightarrow \quad x_1 \leq x_2, \quad x_1 < -x_2$$

Then:

$$f(h_1) = 0, \quad f(h_2) = -x_1 - x_2 \quad \Rightarrow \quad f(h_1) + f(h_2) = -x_1 - x_2 \quad \Rightarrow \quad -x_1 - x_2 = 2 \quad \Rightarrow \quad x_1 + x_2 = -2$$

•Line 3: $x_1 + x_2 = -2$ in this region

Region D: Both ReLUs off

This happens when:

$$x_1 - x_2 \leq 0, \quad -x_1 - x_2 \leq 0 \quad \Rightarrow \quad x_1 \leq x_2, \quad x_1 \geq -x_2 \quad \Rightarrow \quad \text{This region exists if} \quad -x_2 \leq x_1 \leq x_2$$

Then:

$$f(h_1) + f(h_2) = 0 \quad \Rightarrow \quad z = -2 \quad \Rightarrow \quad \sigma(z) < 0.5$$

So no boundary lies in this region.

Final Decision Boundaries

You now have three decision boundary lines based on activation regions:

Region	**Condition**	**Boundary Equation**
Both ReLUs active	$x_1 > x_2, \quad x_1 < -x_2$	$x_2 = -1$
Only $f(h_1)$ active	$x_1 > x_2, \quad x_1 \geq -x_2$	$x_1 = x_2 + 2$
Only $f(h_2)$ active	$x_1 \leq x_2, \quad x_1 < -x_2$	$x_1 + x_2 = -2$
Both ReLUs off	$-x_2 \leq x_1 \leq x_2$	No boundary (z = -2 always)

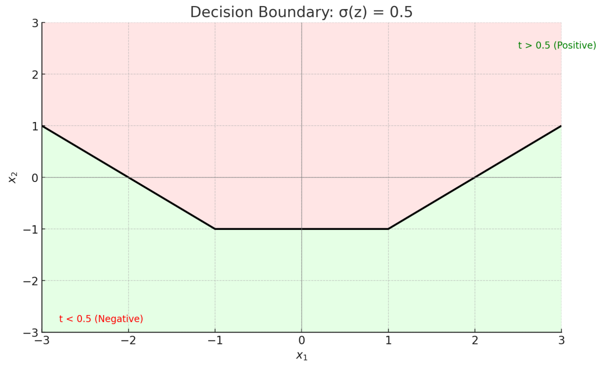

Figure 1: Decision Boundary(Coloured)

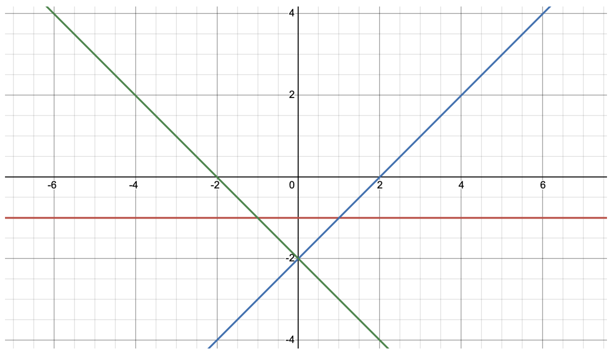

Figure 2: Decision Boundary based off the equations

2. Prediction for $[x_1, x_2]^T = [1, 1]^T$

- $h_1 = 1 \cdot 1 - 1 \cdot 1 + 0 \cdot 1 = 0$
- $h_2 = -1 \cdot 1 - 1 \cdot 1 + 0 \cdot 1 = -2$
- $f(h_1) = \max(0, 0) = 0$
- $f(h_2) = \max(0, -2) = 0$
- $z = 1 \cdot 0 + 1 \cdot 0 - 2 \cdot 1 = -2$
- $t = \sigma(z) = \frac{1}{1+e^{-(-2)}} = \frac{1}{1+e^2} \approx \frac{1}{1+7.389} \approx \frac{1}{8.389} \approx 0.119$

The prediction for $[1, 1]^T$ is approximately 0.119.

3. Gradient Derivation

- Loss function: $l(y, t) = -[y \log(t) + (1 - y) \log(1 - t)]$
- Sigmoid derivative: $\frac{dt}{dz} = t \cdot (1 - t)$
- ReLU derivative: $\frac{df(h_i)}{dh_i} = 1(h_i > 0)$

We use the chain rule:

- $\frac{\partial l}{\partial v_i}$ **$(i \neq 3)$:**
  * $\frac{\partial l}{\partial t} = -\frac{y}{t} + \frac{1-y}{1-t}$ * $\frac{\partial t}{\partial z} = t \cdot (1 - t)$ * $\frac{\partial z}{\partial v_i} = f(h_i)$

$$\frac{\partial l}{\partial v_i} = \left(-\frac{y}{t} + \frac{1-y}{1-t}\right) \cdot t \cdot (1 - t) \cdot f(h_i)$$

$$= [-y \cdot (1-t) + (1-y) \cdot t] \cdot f(h_i)$$
$$= [-y + yt + t - yt] \cdot f(h_i)$$
$$= (t-y) \cdot f(h_i)$$

- $\frac{\partial l}{\partial v_3}$**:**
  * $\frac{\partial z}{\partial v_3} = 1$

$$\frac{\partial l}{\partial v_3} = \left(-\frac{y}{t} + \frac{1-y}{1-t}\right) \cdot t \cdot (1-t) \cdot 1$$
$$= (t-y) \cdot 1$$
$$= t - y$$

- $\frac{\partial l}{\partial w_{ij}}$ **($j \neq 3$):**
  * $\frac{\partial z}{\partial f(h_i)} = v_i$ * $\frac{\partial f(h_i)}{\partial h_i} = 1(h_i > 0)$ * $\frac{\partial h_i}{\partial w_{ij}} = x_j$

$$\frac{\partial l}{\partial w_{ij}} = \left(-\frac{y}{t} + \frac{1-y}{1-t}\right) \cdot t \cdot (1-t) \cdot v_i \cdot 1(h_i > 0) \cdot x_j$$
$$= (t-y) \cdot v_i \cdot 1(h_i > 0) \cdot x_j$$

- $\frac{\partial l}{\partial w_{i3}}$**:** * $\frac{\partial h_i}{\partial w_{i3}} = 1$

$$\frac{\partial l}{\partial w_{i3}} = \left(-\frac{y}{t} + \frac{1-y}{1-t}\right) \cdot t \cdot (1-t) \cdot v_i \cdot 1(h_i > 0) \cdot 1$$
$$= (t-y) \cdot v_i \cdot 1(h_i > 0)$$

Acknowledgment

I would like to acknowledge that I used ChatGPT(for latex erros and syntax correction), Desmos(for graphs), Prof Juneja's class notes and Andrew Ng notes for reference and help during this work.