Deep learning

DR.Fatemi Zadeh

electrical engineering department

Ahmadreza Majlesara 400101861

computer assignment repository

assignment 1

October 27, 2024

Deep learning

assignment 1

Ahmadreza Majlesara 400101861
computer assignment repository

Question 1

Part 1

In the case of linear separability, if one of the training samples is removed, does the decision boundary move towards the removed point, move away from it, or remain the same? Explain your answer. Now, if we consider the decision boundary for logistic regression, will the decision boundary change or remain the same? Explain your answer. (There is no need to mention the direction of the change.)

solution

in SVM if the point is a support vector, the decision boundary will move away from the removed point. Hence the decision boundary will remain the same if the point is not a support vector. In logistic regression, the boundry will change if any of the points are removed. this is because in logistic regression the boundry is determined by the probability of the points being in a certain class and removing a point will change the probability of the points being in a certain class so the boundry will change.

Part 2

Recall from the class notes that if we allow some of the classifications in the training data to be incorrect, the optimization of the SVM (soft margin) is as follows:

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

subject to:

$$y_i(w^T x_i) \geq 1 - \xi_i \quad \forall i \in \{1,\ldots,n\}$$
$$\xi_i \geq 0, \quad \forall i \in \{1,\ldots,n\}$$

where ξ_i are referred to as slack variables. Suppose $\xi_1,\ldots,\xi_n$ have been optimally computed. Use ξ to obtain an upper bound on the number of misclassified samples.

solution

In the soft-margin SVM formulation:

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

the slack variables ξ_i represent the degree to which each data point fails to meet the margin requirement $y_i(w^T x_i) \geq 1$. Specifically:

1. If $\xi_i = 0$, then the data point x_i lies correctly on the correct side of the margin.
2. If $0 < \xi_i \leq 1$, then x_i is on the correct side of the decision boundary but within the margin.
3. If $\xi_i > 1$, then x_i is misclassified because it fails to meet the condition $y_i(w^T x_i) \geq 0$.

For a data point to be misclassified, ξ_i must be strictly greater than 1. Therefore, we can count the number of data points where $\xi_i > 1$ to determine the misclassified samples.
Now, we know that:

$$\sum_{i=1}^{n} \xi_i$$

is the total penalty added in the objective function due to all slack variables. Since each misclassified sample contributes at least 1 to this total, the upper bound on the number of misclassified samples is simply the total sum of all $\xi_i > 0$ values. so the boundary is:

$$\sum_{i=1}^{n} \mathbb{I}(\xi_i > 1)$$

Part 3

In SVM optimization, what is the role of the coefficient C? Briefly explain your answer by considering two extreme cases, i.e., $C \to 0$ and $C \to \infty$.

solution

The coefficient C in the SVM optimization problem is a regularization parameter that controls the trade-off between the margin width and the training error. It balances the importance of maximizing the margin and minimizing the classification error. The role of C can be understood by considering two extreme cases: When $C \to 0$, the regularization term becomes negligible, and the optimization problem becomes:

$$\min_{w,\xi_i} \frac{1}{2}\|w\|^2$$

In this case, the model will focus solely on maximizing the margin, and the decision boundary will be determined by the support vectors only. The model will be highly sensitive to outliers and noise in the data, potentially leading to overfitting. When $C \to \infty$, the regularization term becomes dominant, and the optimization problem becomes:

$$\min_{w,\xi_i} C \sum_{i=1}^{n} \xi_i$$

In this case, the model will focus on minimizing the classification error, even at the cost of a narrower margin. The decision boundary will be less sensitive to individual data points, leading to a more robust model that generalizes better to unseen data.

Part 4

Compare hard-margin SVM and logistic regression when the two classes are linearly separable. Mention any notable differences.

solution

in the term of objective function hard-margin SVM is as follows:

$$\min_{w} \frac{1}{2}\|w\|^2$$

this approach tries to maximize the margin between the two classes and the decision boundary is determined by the support vectors only. In contrast, logistic regression uses the following objective function:

$$\min_{w} \sum_{i=1}^{n} \log(1 + e^{-y_i(w^T x_i)})$$

this approach tries to maximize the likelihood of the data and the decision boundary is determined by the probability of the points being in a certain class. The main difference between the two approaches is that hard-margin SVM focuses on maximizing the margin, while logistic regression focuses on maximizing the likelihood of the data. output of hard margin SVM is a deterministic boundary without probability estimates. A point's classification depends on which side of the boundary it falls.Hence the output of logistic regression is a probabilistic boundary with probability estimates. A point's classification depends on the probability of it being in a certain class. and in term of robustness, hard-margin SVM is sensitive to outliers and noise in the data, as it tries to find the maximum margin. Logistic regression is more robust to outliers and noise, as it tries to maximize the likelihood of the data.

Part 5

Compare soft-margin SVM and logistic regression when the two classes are not linearly separable. Mention any notable differences.

solution

In terms of objective function, soft-margin SVM is as follows:

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

where ξ_i are slack variables that allow some points to fall on the wrong side of the margin. This approach tries to maximize the margin while allowing some classification errors, which makes it suitable for non-linearly separable data. In contrast, logistic regression uses the following objective function:

$$\min_{w} \sum_{i=1}^{n} \log(1 + e^{-y_i(w^T x_i)})$$

This approach tries to maximize the likelihood of the data, fitting a decision boundary that reflects the probability of each point belonging to a particular class. Logistic regression does not attempt to maximize the margin; instead, it optimizes for a probability-based decision boundary.

The main difference between these approaches is that soft-margin SVM focuses on a balance between maximizing the margin and minimizing classification errors, while logistic regression focuses entirely on maximizing the likelihood of the data without directly optimizing for a margin.

The output of soft-margin SVM is still a deterministic boundary, but it allows some flexibility for points that do not fit perfectly on either side of the margin. Logistic regression, on the other hand, provides a probabilistic boundary with probability estimates, meaning that each point is assigned a probability of being in a certain class.

In terms of robustness, soft-margin SVM is more robust than hard-margin SVM because it allows some misclassifications, but it can still be affected by outliers, especially when C is large. Logistic regression is generally more robust to outliers since it maximizes the likelihood across all points and does not enforce a strict margin.

Question 2

Part 1

Suppose in PCA we project each point x_i to $z_i = V_k^T x_i$, where $V_k = [v_1, \ldots, v_k]$, i.e., the first k principal components. We can reconstruct x_i from z_i as follows:

$$\hat{x}_i = V_k z_i$$

Show that

$$\|x_i - \hat{x}_i\| = \|z_i - z_j\|$$

holds.

solution

from the given information, we have:

$$\hat{x}_i = V_{1:k} z_i$$

$$\hat{x}_j = V_{1:k} z_j$$

so we can write:

$$\hat{x}_i - \hat{x}_j = V_{1:k} z_i - V_{1:k} z_j$$

$$\hat{x}_i - \hat{x}_j = V_{1:k}(z_i - z_j)$$

thus we can conclude that:

$$\|\hat{x}_i - \hat{x}_j\|^2 = \|V_{1:k}(z_i - z_j)\|^2$$

so if we expand the right side of the equation by the fact that $\|x\|^2 = x^T x$ we get:

$$\|V_{1:k}(z_i - z_j)\|^2 = (V_{1:k}(z_i - z_j))^T (V_{1:k}(z_i - z_j))$$

$$= (z_i - z_j)^T V_{1:k}^T V_{1:k}(z_i - z_j)$$

Since $V_{1:k}$ consists of the first k principal components, its columns are orthonormal.

$$\Rightarrow V_{1:k}^T V_{1:k} = I$$

$$\Rightarrow (z_i - z_j)^T V_{1:k}^T V_{1:k}(z_i - z_j) = (z_i - z_j)^T I (z_i - z_j)$$

$$= (z_i - z_j)^T (z_i - z_j)$$

$$= \|z_i - z_j\|^2$$

thus we approve that:

$$\Rightarrow \|\hat{x}_i - \hat{x}_j\|^2 = \|z_i - z_j\|^2$$

Part 2

Show that the reconstruction error is equal to:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = (n-1) \sum_{j=k+1}^{p} \lambda_j$$

where $\lambda_{k+1}, \ldots, \lambda_p$ are the smallest eigenvalues. Therefore, the more principal components we use for reconstruction, the better the accuracy.

solution

again from the given information, we have:

$$\hat{x}_i = V_{1:k} V_{1:k}^T x_i$$

so

$$x_i - \hat{x}_i = x_i - V_{1:k} V_{1:k}^T x_i$$

this can be written as:

$$x_i - \hat{x}_i = (I - V_{1:k} V_{1:k}^T) x_i$$

and if we get norm of the above equation we get:

$$\|x_i - \hat{x}_i\|^2 = \|(I - V_{1:k} V_{1:k}^T) x_i\|^2$$

Sum of Squared Reconstruction Errors can be written as:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = \sum_{i=1}^{n} \|(I - V_{1:k} V_{1:k}^T) x_i\|^2$$

$$\sum_{i=1}^{n} \|(I - V_{1:k} V_{1:k}^T) x_i\|^2 = \sum_{i=1}^{n} \|V_{k+1:p} V_{k+1:p}^T x_i\|^2$$

$$= \sum_{i=1}^{N} (V_{k+1:p} V_{k+1:p}^T x_i)^T (V_{k+1:p} V_{k+1:p}^T x_i)$$

$$= \sum_{i=1}^{N} x_i^T V_{k+1:p} V_{k+1:p}^T V_{k+1:p} V_{k+1:p}^T x_i$$

so cause $V_{k+1:p}$ is orthogonal matrix, we have:

$$= \sum_{i=1}^{N} x_i^T V_{k+1:p} V_{k+1:p}^T x_i$$

$$= \sum_{i=1}^{N} Tr(x_i^T V_{k+1:p} V_{k+1:p}^T x_i)$$

$$= \sum_{i=1}^{N} Tr(V_{k+1:p}^T x_i x_i^T V_{k+1:p})$$

so as we know we can define $S = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$ and as $\bar{x} = 0$ this can be written as:

$$S = \frac{1}{N-1}\sum_{i=1}^{N} x_i x_i^T$$

so we can write:

$$\sum_{i=1}^{N} Tr(V_{k+1:p}^T x_i x_i^T V_{k+1:p}) = Tr(V_{k+1:p}^T S V_{k+1:p})$$

which is equal to:

$$(n-1)Tr(V_{k+1:p}^T S V_{k+1:p}) = (n-1)\sum_{i=k+1}^{p} \lambda_i$$

so the sum of squared reconstruction errors is equal to:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 = (n-1)\sum_{i=k+1}^{p} \lambda_i$$

The equation $\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 = (n-1)\sum_{i=k+1}^{p} \lambda_i$ provides insights into how PCA minimizes reconstruction error by focusing on the largest eigenvalues (principal components) that capture the most variance in the data. It highlights the trade-offs involved in dimensionality reduction and reconstruction accuracy in PCA.The reconstruction error is directly related to the eigenvalues that correspond to the discarded principal components. The larger these eigenvalues, the greater the reconstruction error. If we use more principal components (larger k) to reconstruct the data, the reconstruction error will decrease.

Question 3

Part 1

Consider the equation $Xw = y$, where $X \in \mathbb{R}^{m \times n}$ is a non-square data matrix, w is a weight vector, and y is a vector of labels corresponding to each data point in each row of X.

Assume $X = U\Sigma V^T$ (full SVD of X). Here, U and V are square and orthogonal matrices, and Σ is an $m \times n$ matrix with non-zero singular values (σ_i) on the diagonal.

For this problem, $\Sigma^\dagger$ is defined as an $n \times m$ matrix with the inverse of singular values $\left(\frac{1}{\sigma_i}\right)$ along the diagonal.

First, consider the case where $m > n$, meaning the data matrix X has more rows than columns (tall matrix) and the system is overdetermined. How do we find the weights w that minimize the error between Xw and y? In other words, we want to solve $\min \|Xw - y\|^2$.

solution

we can rewrite the equation as:

$$min\|Xw - y\|^2 = min(Xw - y)^T(Xw - y)$$

$$\Rightarrow min\|Xw - y\|^2 = minw^T X^T Xw - w^T X^T y - y^T Xw + y^T y$$

$$\Rightarrow min\|Xw - y\|^2 = minw^T X^T Xw - 2w^T X^T y + y^T y$$

now if we take the derivative of the equation with respect to w and set it to zero, we get:

$$\frac{d}{dw}(w^T X^T Xw - 2w^T X^T y + y^T y) = 0$$

$$\Rightarrow 2X^T Xw - 2X^T y = 0$$

$$\Rightarrow X^T Xw = X^T y$$

$$\Rightarrow w = (X^T X)^{-1} X^T y$$

Part 2

Use the SVD $X = U\Sigma V^T$ and simplify the solution.

solution

if we multiply the equation $Xw = y$ by X^T from the left, we get:

$$X^T Xw = X^T y$$

now if we substitute $X = U\Sigma V^T$ in the equation, we get:

$$V\Sigma^T U^T U\Sigma V^T w = V\Sigma^T U^T y$$

$$\Rightarrow V\Sigma^T\Sigma V^T w = V\Sigma^T U^T y$$
$$\Rightarrow w = V(\Sigma^T\Sigma)^{-1}\Sigma^T U^T y$$
$$\Rightarrow w = V\Sigma^\dagger U^T y$$

Part 3

You will notice that the least squares solution is of the form $w^* = Ay$. What happens if we multiply X from the left by matrix A? For this reason, matrix A is called the left inverse least squares.

solution

in the previous part we found that the least-squares solution has the form $w^* = Ay$, which $A = V\Sigma^\dagger U^T$. if we multiply X from the left by the matrix A, we get:

$$XA = U\Sigma V^T V\Sigma^\dagger U^T = U\Sigma\Sigma^\dagger U^T$$

as it is mentioned the matrix Σ is a $m \times n$ matrix with non-zero singular values on the diagonal, so the product of $\Sigma\Sigma^\dagger$ is a $m \times m$ diagonal matrix with 1 on the first n diagonal and 0 on the rest. so the product of $U\Sigma\Sigma^\dagger U^T$ is equal to UI_nU^T which I_n is an $m \times m$ matrix that has 1 on the first n diagonal and 0 on the rest. hence the product of XA is equal to I_n. so the matrix A is called the left pseudoinverse of the least-squares solution.

Part 4

Now consider the case where $m < n$, meaning the data matrix X has more columns than rows, and the system is underdetermined. There are infinitely many solutions for w. However, we are looking for the minimum-norm solution, meaning we want to solve $\min \|w\|_2^2$ subject to $Xw = y$. What is the minimum-norm solution?

solution

we have an optimization problem in the following form:

$$min\|w\|_2^2$$

$$s.t\ Xw = y$$

we can write the lagraingian of the problem as:

$$L(w, \lambda) = \|w\|_2^2 + \lambda^T(Xw - y)$$

and we can rewrite the equation as:

$$L(w,\lambda) = w^T w + \lambda^T X w - \lambda^T y$$

now if we take the derivative of the equation with respect to w and set it to zero, we get:

$$\frac{\partial}{\partial w}(w^T w + \lambda^T X w - \lambda^T y) = 0$$

$$\Rightarrow 2w + X^T \lambda = 0$$

$$\Rightarrow w = -\frac{1}{2} X^T \lambda$$

we can calculate λ by substituting the value of w in the equation $Xw = y$:

$$Xw = y$$

$$\Rightarrow X(-\frac{1}{2} X^T \lambda) = y$$

$$\Rightarrow -\frac{1}{2} X X^T \lambda = y$$

$$\Rightarrow \lambda = -2(XX^T)^{-1} y$$

$$\Rightarrow w = X^T (XX^T)^{-1} y$$

Part 5

Use the SVD $X = U\Sigma V^T$ and simplify the solution.

solution

if we substitute the value of X in the equation $w = X^T (XX^T)^{-1} y$, we get:

$$w = (U\Sigma V^T)^T ((U\Sigma V^T)(U\Sigma V^T)^T)^{-1} y$$

$$w = V\Sigma^T U^T (U\Sigma V^T V\Sigma^T U^T)^{-1} y$$

$$w = V\Sigma^T U^T (U\Sigma\Sigma^T U^T)^{-1} y$$

$$w = V\Sigma^\dagger U^T y$$

$$w = V\Sigma^\dagger U^T y$$

Part 6

You will notice that the minimum norm solution is of the form $w^* = By$. What happens if we multiply X from the right by matrix B? For this reason, matrix B is called the right inverse minimum norm.

solution

in the previous part we found that the minimum-norm solution has the form $w^* = By$, which $B = V\Sigma^{\dagger}U^T$. if we multiply X from the right by the matrix B, we get:

$$BX = V\Sigma^{\dagger}U^T U\Sigma V^T = V\Sigma^{\dagger}\Sigma V^T$$

as it is mentioned the matrix Σ is a $m \times n$ matrix with non-zero singular values on the diagonal, so the product of $\Sigma^{\dagger}\Sigma$ is a $n \times n$ diagonal matrix with 1 on the first m diagonal and 0 on the rest. so the product of $V\Sigma^{\dagger}\Sigma V^T$ is equal to $VI_m V^T$ which I_m is an $n \times n$ matrix that has 1 on the first m diagonal and 0 on the rest. hence the product of BX is equal to I_m. so the matrix B is called the right pseudoinverse of the minimum-norm solution.

Question 4

Part 1

Consider a linear regression problem that includes n data points and d features. When $n = d$, the matrix $F \in \mathbb{R}^{n \times n}$ has the biggest eigenvalue α and the smallest eigenvalue with a very small value. We have $y = Fw^* + \varepsilon$. If we calculate $\hat{w}_{inv} = F^{-1}y$, cause of small singular value of F and having noise we see that $\|\hat{w}_{inv} - w^*\| = 10^{10}$.

Instead of inverting F, assume we use gradient descent. We repeat gradient descent k times starting from $w = 0$ with a loss function $\ell(w) = \frac{1}{2}\|y - Fw\|^2$. We assume that the learning rate η is small enough to ensure the stability of gradient descent for the given problem (this is an important point).

The gradient descent update formula for $t > 0$ is as follows:

$$w_t = w_{t-1} - \eta \left(F^T (Fw_{t-1} - y)\right)$$

We are looking for the error $\|w_k - w^*\|_2$. We want to show that, in the worst case, this error can be bounded by the following:

$$\|w_k - w^*\|_2 \leq k\eta\alpha\|y\|_2 + \|\hat{w}_l\|_2$$

In other words, the error cannot go out of bounds, at least not too quickly.

To complete this task, we only need to prove the key idea using the triangle inequality and the norm properties, as the result will follow naturally.

Show that for $t > 0$:

$$\|w_t\|_2 \leq \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$$

solution

The gradient descent update rule is:

$$w_t = w_{t-1} - \eta F^T (Fw_{t-1} - y)$$

This can be expanded as:

$$w_t = w_{t-1} - \eta F^T F w_{t-1} + \eta F^T y$$

if we factorize w_{t-1} from the equation, we can simplify the equation as:

$$w_t = (I - \eta F^T F) w_{t-1} + \eta F^T y$$

now we take the norm of both sides:

$$\|w_t\|_2 = \|(I - \eta F^T F) w_{t-1} + \eta F^T y\|_2$$

if we use the triangle inequality, we can bound the above equation as:

$$\|w_t\|_2 \leq \left\|(I-\eta F^T F)w_{t-1}\right\|_2 + \left\|\eta F^T y\right\|_2$$

now we have 2 terms in the right side of the equation, we can bound each term separately:

$$\left\|(I-\eta F^T F)w_{t-1}\right\|_2 \leq \left\|I-\eta F^T F\right\|_2 \|w_{t-1}\|_2$$

The spectral norm $\left\|I-\eta F^T F\right\|_2$ can be bounded using the eigenvalues of $F^T F$. Let $\lambda_{\min}$ and $\lambda_{\max}$ be the smallest and largest eigenvalues of $F^T F$, respectively. Since η is small enough to ensure stability, and the smallest eigenvalue is very small ($\lambda_{\min} \approx 0$), we have:

$$\left\|I-\eta F^T F\right\|_2 = \max_i |1-\eta\lambda_i| = 1-\eta\lambda_{\min} \approx 1$$

so we can bound the first term as:

$$\left\|(I-\eta F^T F)w_{t-1}\right\|_2 \leq \|w_{t-1}\|_2 \quad (1)$$

Now we bound the second term:

$$\left\|\eta F^T y\right\|_2 = \eta \left\|F^T y\right\|_2 \leq \eta \left\|F^T\right\|_2 \|y\|_2$$

Since $\left\|F^T\right\|_2 = \|F\|_2$, and the largest eigenvalue of F is α:

$$\left\|\eta F^T y\right\|_2 \leq \eta\alpha \|y\|_2 \quad (2)$$

Combining equations (1) and (2), we get:

$$\|w_t\|_2 \leq \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$$

and this completes the proof.

If gradient descent cannot diverge, what can be said about the eigenvalues of $(I-\eta F^T F)$, what shape do the eigenvalues take?

solution

Let λ_i be the eigenvalues of $F^T F$. Since $F^T F$ is a symmetric positive semi-definite matrix, all its eigenvalues are non-negative:

$$\lambda_i \geq 0 \quad \text{for all } i.$$

The eigenvalues of the matrix $I-\eta F^T F$ are given by:

$$\mu_i = 1-\eta\lambda_i.$$

For the gradient descent algorithm to **not diverge**, the magnitude of each eigenvalue μ_i must be less than or equal to 1:

$$|\mu_i| \leq 1.$$

This condition ensures that the iterative updates do not amplify the errors, keeping the algorithm stable. now if we substitute the expression for μ_i in the above inequality, we get:

$$|1 - \eta\lambda_i| \leq 1 \implies -1 \leq 1 - \eta\lambda_i \leq 1$$

which simplifies to:

$$0 \leq \eta\lambda_i \leq 2.$$

so we can say that $0 \leq \eta \leq \frac{2}{\lambda_i}$, and this can simplify to:

$$0 \leq \eta \leq \frac{2}{\lambda_{max}}$$

If gradient descent cannot diverge, the eigenvalues of $(I - \eta F^T F)$ must be real numbers within the interval $[-1, 1]$. They take the shape of values that, when applied iteratively, do not cause the weight vector to grow unboundedly. This ensures the stability and convergence of the gradient descent algorithm.

Question 5

Part 1

Show that the expected squared error can be decomposed into three parts: bias, variance, and irreducible error σ^2:

$$Error = Bias^2 + Variance + \sigma^2$$

Formally, assume we have a randomly sampled training set D (independently drawn from the test data), and we compute an estimator $\hat{\theta}(D)$ (for example, using empirical risk minimization). The expected squared error for a test input x is decomposed as follows:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(Y-\hat{f}_{\hat{\theta}(D)}(x))^2\right] = Bias\left(\hat{f}_{\hat{\theta}(D)}(x)\right)^2 + Var\left(\hat{f}_{\hat{\theta}(D)}(x)\right) + \sigma^2$$

Recall the formulaic history of variance and bias that may be useful:

$$Bias\left(\hat{f}_{\hat{\theta}(D)}(x)\right) = \mathbb{E}_{Y\sim p(y|x),D}\left[\hat{f}_{\hat{\theta}(D)}(x) - Y\right]$$

$$Var\left(\hat{f}_{\hat{\theta}(D)}(x)\right) = \mathbb{E}_D\left[\left(\hat{f}_{\hat{\theta}(D)}(x) - \mathbb{E}_D\left[\hat{f}_{\hat{\theta}(D)}(x)\right]\right)^2\right]$$

solution

we know that D is sampled independently from test data. if we define $f(x)$ as the true function, then we can write the expected squared error as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(Y-\hat{f}_{\hat{\theta}(D)}(x))^2\right] = \mathbb{E}_{Y\sim p(y|x),D}\left[(Y-f(x)+f(x)-\hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

using this we can expand the above equation as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(Y-f(x))^2\right] + \mathbb{E}_{Y\sim p(y|x),D}\left[(f(x)-\hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

$$+2\mathbb{E}_{Y\sim p(y|x),D}\left[(Y-f(x))(f(x)-\hat{f}_{\hat{\theta}(D)}(x))\right]$$

now we can expand the last term of above equation using the fact that $y = f(x) + \varepsilon$ where ε is the noise term as follows:

$$2\mathbb{E}_{Y\sim p(y|x),D}\left[(f(x)+\varepsilon)f(x)\right] - 2\mathbb{E}_{Y\sim p(y|x),D}\left[f(x)f(x)\right]$$

$$-2\mathbb{E}_{Y\sim p(y|x),D}\left[(f(x)+\varepsilon)\hat{f}_{\hat{\theta}(D)}(x)\right] + 2\mathbb{E}_{Y\sim p(y|x),D}\left[f(x)\hat{f}_{\hat{\theta}(D)}(x)\right]$$

and this can be written as:

$$2f^2(x) + 2\mathbb{E}_{Y\sim p(y|x),D}\left[\varepsilon\right]f(x) - 2f^2(x) - 2f(x)\mathbb{E}_{Y\sim p(y|x),D}\left[f_{\hat{\theta}(D)}(x)\right]$$

$$-\mathbb{E}_{Y\sim p(y|x),D}\left[\varepsilon\right]\mathbb{E}_{Y\sim p(y|x),D}\left[f_{\hat{\theta}(D)}(x)\right] + f(x)\mathbb{E}_{Y\sim p(y|x),D}\left[f_{\hat{\theta}(D)}(x)\right]$$

as we know that $\mathbb{E}_{Y\sim p(y|x),D}[\varepsilon] = 0$ so the above equation is equal to 0 and we can rewrite the first equation as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(Y - f(x))^2\right] + \mathbb{E}_{Y\sim p(y|x),D}\left[(f(x) - \hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

we know that the first term is the noise term so we focus on writing the second term. if we define $\mathbb{E}_{Y\sim p(y|x),D}\left[f_{\hat{\theta}(D)}(x)\right]$ as $\bar{f}(x)$ then we can write the second term as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(f(x) - \hat{f}_{\hat{\theta}(D)}(x))^2\right] = \mathbb{E}_{Y\sim p(y|x),D}\left[(f(x) - \bar{f}(x) + \bar{f}(x) - \hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

now we can write this equation as follows:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(f(x) - \bar{f}(x))^2\right] + \mathbb{E}_{Y\sim p(y|x),D}\left[(\bar{f}(x) - \hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

$$+2\mathbb{E}_{Y\sim p(y|x),D}\left[(f(x) - \bar{f}(x))(\bar{f}(x) - \hat{f}_{\hat{\theta}(D)}(x))\right]$$

we can expand the last term as follows:

$$2\mathbb{E}_{Y\sim p(y|x),D}\left[f(x)\bar{f}(x)\right] - 2\mathbb{E}_{Y\sim p(y|x),D}\left[f(x)\hat{f}_{\hat{\theta}(D)}(x)\right]$$

$$-2\mathbb{E}_{Y\sim p(y|x),D}\left[\bar{f}(x)\bar{f}(x)\right] + 2\mathbb{E}_{Y\sim p(y|x),D}\left[\bar{f}(x)\hat{f}_{\hat{\theta}(D)}(x)\right]$$

this equals to the following:

$$2f(x)\bar{f}(x) - 2f(x)\bar{f}(x) - 2\bar{f}(x)^2 + 2\bar{f}(x)^2 = 0$$

so we can write the second term as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(f(x) - \bar{f}(x))^2\right] + \mathbb{E}_{Y\sim p(y|x),D}\left[(\bar{f}(x) - \hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

the first term is expectation of the squared bias and can be written as $(f(x) - \bar{f}(x))^2$ so we can write the expected squared error as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(Y - \hat{f}_{\hat{\theta}(D)}(x))^2\right] = \mathbb{E}_{Y\sim p(y|x),D}\left[(Y - f(x))^2\right] + (f(x) - \bar{f}(x))^2$$

$$+\mathbb{E}_{Y\sim p(y|x),D}\left[(\bar{f}(x) - \hat{f}_{\hat{\theta}(D)}(x))^2\right]$$

as i mentioned before the first term is the noise term. the second is the squared bias and the third term is the variance term. so we can write the expected squared error as:

$$\mathbb{E}_{Y\sim p(y|x),D}\left[(Y - \hat{f}_{\hat{\theta}(D)}(x))^2\right] = Bias\left(\hat{f}_{\hat{\theta}(D)}(x)\right)^2 + Var\left(\hat{f}_{\hat{\theta}(D)}(x)\right) + \sigma^2$$

Part 2

Suppose our training set consists of $D = \{(x_i, y_i)\}_{i=1}^n$, where the only randomness comes from the noise vector ε. $Y = X\theta^* + \varepsilon$, where θ^* is the true linear model and each noise variable ε_i is independently and identically distributed with zero mean and variance 1. We use ordinary least squares (OLS) to estimate $\hat{\theta}$ from this data.

Calculate the error and variance of the estimate $\hat{\theta}$, and use it to calculate the error and variance of predictions on specific test inputs. For simplicity, assume $X^T X$ is diagonal.

$$\hat{\theta} = (X^T X)^{-1} X^T Y$$

Assume our data matrix is non-random and $Y \in \mathbb{R}^n$ is a random vector representing the noisy training targets. For simplicity, assume $X^T X$ is diagonal.

solution

first we calculate the error and variance of the estimate $\hat{\theta}$. we know that $\hat{\theta} = (X^T X)^{-1} X^T Y$ so we can write the error as:

$$\begin{aligned}
\text{Error} = \mathbb{E}\left[\hat{\theta}\right] &= \mathbb{E}\left[(X^T X)^{-1} X^T Y\right] \\
&= \mathbb{E}\left[(X^T X)^{-1} X^T (X\theta^* + \varepsilon)\right] \\
&= \mathbb{E}\left[(X^T X)^{-1} X^T X\theta^* + (X^T X)^{-1} X^T \varepsilon\right] \\
&= \mathbb{E}\left[\theta^* + (X^T X)^{-1} X^T \varepsilon\right] \\
&= \theta^* + (X^T X)^{-1} X^T \mathbb{E}[\varepsilon] \\
&= \theta^*
\end{aligned}$$

so we can say that $\mathbb{E}\left[\hat{\theta}\right] = \theta^*$. now we calculate the covariance of the estimate $\hat{\theta}$ as follows:

$$\begin{aligned}
Cov(\hat{\theta}) &= \mathbb{E}\left[(\hat{\theta} - \mathbb{E}\left[\hat{\theta}\right])(\hat{\theta} - \mathbb{E}\left[\hat{\theta}\right])^T\right] \\
&= \mathbb{E}\left[(\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T\right] \\
&= \mathbb{E}\left[((X^T X)^{-1} X^T Y - \theta^*)((X^T X)^{-1} X^T Y - \theta^*)^T\right] \\
&= \mathbb{E}\left[((X^T X)^{-1} X^T (X\theta^* + \varepsilon) - \theta^*)((X^T X)^{-1} X^T (X\theta^* + \varepsilon) - \theta^*)^T\right] \\
&= \mathbb{E}\left[((X^T X)^{-1} X^T \varepsilon)((X^T X)^{-1} X^T \varepsilon)^T\right] \\
&= \mathbb{E}\left[(X^T X)^{-1} X^T \varepsilon\varepsilon^T X (X^T X)^{-1}\right] \\
&= (X^T X)^{-1} X^T \mathbb{E}\left[\varepsilon\varepsilon^T\right] X (X^T X)^{-1} \\
&= (X^T X)^{-1} X^T I X (X^T X)^{-1} \\
&= (X^T X)^{-1} X^T X (X^T X)^{-1} \\
&= (X^T X)^{-1}
\end{aligned}$$

so we can say that $Cov(\hat{\theta}) = (X^T X)^{-1}$. now we calculate the error and variance of predictions on specific test inputs. we can write the error as:

$$\text{Error} = \mathbb{E}\left[(X\hat{\theta} - Y)\right]$$

$$= \mathbb{E}\left[X\hat{\theta}\right] - \mathbb{E}[X\theta^* + \varepsilon]$$

$$= X\mathbb{E}\left[\hat{\theta}\right] - X\theta^* - \mathbb{E}[\varepsilon]$$

$$= X\theta^* - X\theta^* - 0 = 0$$

so we can say that $\text{Error} = 0$. now we calculate the variance of predictions on specific test inputs as follows:

$$\text{Var} = \mathbb{E}\left[(X\hat{\theta} - \mathbb{E}\left[X\hat{\theta}\right])(X\hat{\theta} - \mathbb{E}\left[X\hat{\theta}\right])^T\right]$$

$$= \mathbb{E}\left[(X(X^T X)^{-1}X^T Y - X\theta^*)(X(X^T X)^{-1}X^T Y - X\theta^*)^T\right]$$

$$= \mathbb{E}\left[(X(X^T X)^{-1}X^T (X\theta^* + \varepsilon) - X\theta^*)(X(X^T X)^{-1}X^T (X\theta^* + \varepsilon) - X\theta^*)^T\right]$$

$$= \mathbb{E}\left[(X(X^T X)^{-1}X^T \varepsilon)(X(X^T X)^{-1}X^T \varepsilon)^T\right]$$

$$= \mathbb{E}\left[X(X^T X)^{-1}X^T \varepsilon\varepsilon^T X(X^T X)^{-1}X^T\right]$$

$$= X(X^T X)^{-1}X^T \mathbb{E}\left[\varepsilon\varepsilon^T\right] X(X^T X)^{-1}X^T$$

$$= X(X^T X)^{-1}X^T I X(X^T X)^{-1}X^T$$

$$= X(X^T X)^{-1}X^T$$