Machine, Data and Learning

Assignment-2

Report

By: Shivam Singh

2022115003

Task: 1,2 – Pics attached

Task: 3,4,5 – Have been done in this report with screenshots and graphs

Other graphs have been attached at the end of the report.

Task 1: Gradient Descent

Task 1) Gradient Descent - It is an optimization algo used to find the coefficients (slope & intercept) in a linear regression model.

(a) For one independent & one ~~indepen~~ dependent variable,

$$\hat{y} = \underbrace{w_1}_{\text{parameter}} x + \underbrace{w_0}_{\text{bias}}$$

x: independent var

$\hat{y}$: dependent variable

w_0 & w_1 $\propto$ fitness ~~model of~~ of model

Thus we define loss funcⁿ to measure fitness of model with ~~least~~ least sq. error.

$$J(w_0, w_1) = \left(\sum_{n=1}^{N} y^n - f(x^{(n)}; w_0, w_1) \right) \frac{1}{N}$$

Minimizing loss functionⁿ $\propto$ model fit goodness

Thus target is to minimize loss functionⁿ

$\Rightarrow$ $\exists$ w_0, w_1 for which $J(w_0, w_1)$ is least.

Now using gradient descent to find w_0, w_1

i) Lets assume $w_0 = 0$ & $w_1 = 0$

ii) Calculate descent of loss function w.r.t. w_0, w_1

$$\frac{\partial J}{\partial w_0} = \frac{1}{N} \left(\sum_{n=1}^{N} f(x^{(n)} - y^{(n)}) \right)$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{N} \left(\sum_{n=1}^{N} \left(f(x^{(n)}) - y^{(n)} \right) x^{(n)} \right)$$

iii) The algo now takes a learning rate say α.

iv) This learning rate switches function in opp. to gradient by updating parameter in following way.

$$w_{1(n+1)} = w_{(n)} - \alpha \frac{\partial L}{\partial w_1}$$

$$w_0(n+1) = w_{0(n)} - \alpha \frac{\partial L}{\partial w_0}$$

v) The algo stops by comparing the value & previous value of loss function & if

↳ value decreases less than that for a set no. of iteration

↳ ~~val~~ val increase for a set no. of iteration

Q1

v) The algo can also stop after a desired no. of iterations are completed.

At the end of algo, we have our desired value of ω_0, ω_1

(4) For multiple independent variables & single dependent variable,

$$\hat{y} = \omega_0 + \sum_{n=1}^{D} (\omega_i x_n)$$

$$\hat{y} = f(x_1, x_2, \ldots x_D; \omega_0, \omega_1, \ldots \omega_D) \quad \text{where } D > 1$$

let $x_0 = 1$

$\Rightarrow$ ω_0 becomes bias

$$\underline{\omega}' = [\omega_0 \ldots \omega_D]^T \quad \underline{x} = [1 \; x_1 \ldots x_D]^T$$

let's write the loss of function of above model -

$$J(\underline{\omega}) = \left(\sum_{n=1}^{N} \left(y^{(n)} - f(x^{(n)})\right)^2 \right) \cdot 1/N$$

$$= \frac{1}{N} \left(\sum_{n=1}^{N} \left(y^{(n)} - (\omega_0 + \omega_1 x_1^{(n)} + \ldots \omega_D x_D^{(n)}) \right)^2 \right)$$

to optimize, $J(\underline{\omega})$ we need to $\left(\frac{\partial J(\underline{\omega}')}{\partial \omega_0}\right), \left(\frac{\partial J(\underline{\omega}')}{\partial \omega_1}\right)$, $\ldots \frac{\partial J(\omega')}{\partial \omega_D}$ minimize them.

Repeat for D dimension,

$$\omega_0 \leftarrow \omega_0 - \frac{\partial J}{\partial \omega_0}$$

$$\vdots$$

$$\omega_D \leftarrow \omega_D - \frac{\partial J}{\partial \omega_D}$$

$\Rightarrow$ can be written in matrix

$$\boxed{\omega' \leftarrow \omega' - \frac{\partial J}{\omega'}}$$

$\Rightarrow$ $[\omega_0, \omega_1 \ldots \omega_D]^T = \omega'$ becomes similar as above case, then applying partial derivatives give above sys. of eq.

Task 2: Numerical on Bias and Variance

Task 2 $x: [-2, -1, 0, 1, 2, 3]$
$y: [5, 0, 1, 4, 9, 21]$ → True val.

Model-1 $f(x_1) = 2x^2 + 3x + 1$
$\hat{y}_1 : [3, 0, 1, 6, 15, 28]$

Model-2 $f(x_2) = x^2 + 3x$
$\hat{y}_2 : [-2, -2, 0, 4, 10, 18]$

Model-3 $f(x_3) = 2x^2 + 2x + 1$
$\hat{y}_3 : [5, 1, 1, 5, 13, 25]$

Bias^2 calculate → $Bias = E_e[f(x)] - f(x)$ for all models
$Bias^2 = E_x[Bias^2]$ for all value of x.

$Bias(-2) = -3$, $Bias(-1) = \cancel{(\ldots)}\ -1/3 = -0.3$
$Bias(0) = -1/3 = -0.33$, $Bias(1) = 1$, $Bias(2) = 1.66$
$Bias(3) = 1.66$ (on calculation).

$E_x[Bias^2] = \left((5/3)^2 + (5/3)^2 + (-1/3)^2 + (-1/3)^2 + 1^2 + (-3)^2\right] \times \frac{1}{6}$
$= \boxed{2.63}$

MSE calculation – $MSE = E_i\,(y - f(x_i))^2$ calculate for all models
$MSE(-2) = \left((5-3)^2 + (5-(-2))^2 + (5-5)^2\right) 1/3 = 17.67$
$MSE(-1) = 1.67$
$MSE(0) = 0.33$

$MSE(1) = 1.67$, $MSE(2) = 7$, $MSE(3) = 20.33$

$MSE = E_x[MSE] \rightarrow$ for all data points

$= 8.11$

Varince calculation: $Var = E_i\left(f(x_i) - E_i(f(x_i))^2\right)$ for all models,

$Var(-2) = \left((3-2)^2 + (-2-2)^2 + (5-2)^2\right)\frac{1}{3} = 8.67$

Similarly,

$Var(-1) = 1.56$, $Var(0) = 0.23$; $Var(1) = 0.67$

$Var(2) = 4.23$, $Var(3) = 17.56$

$Var = E_x[Var]$ for all data point.

$= 5.48$

$MSE = Var + Bias^2$

$\Rightarrow E_x[MSE] = E_x[Bias^2] + E_x[Var]$

$$\boxed{9.11 = 2.63 + 5.48}$$

Hence Proved.

Task 3: Calculating Bias and Variance

Here are the tabulated Model bias and variance for different degree of function classes:

Degree	Bias	Variance
1	0.8198783140094406	0.036492098415394934
2	0.8097155830233209	0.0817271838970644
3	0.07562863446829068	0.06612970576633166
4	0.09529258512650378	0.10953973025472619
5	0.08766792088742666	0.15777251592072294
6	0.09049667217212096	0.1718743527217831
7	0.10687837086680448	0.22107082193244593
8	0.13512728406586505	0.19228827413916486
9	0.20469766512030113	0.18139456328772088
10	0.33922453890619	0.1864299551931788

Bias:

Bias measures how far off the predictions of the model are from the true values. A high bias indicates that the model is too simplistic and cannot capture the underlying patterns in the data.

As we increase the complexity of the model (e.g., by increasing the degree of polynomial functions), the bias generally decreases. This is because more complex models have the flexibility to fit the data more closely, reducing the error due to bias.

However, beyond a certain point, increasing the complexity too much can lead to overfitting, where the model fits the training data too closely and fails to generalize well to unseen data. This can result in an increase in bias for very high degrees of polynomials.

Variance:

Variance measures the variability of the model's predictions for different training sets. A high variance indicates that the model is sensitive to small fluctuations in the training data.

As we increase the complexity of the model, the variance generally increases. This is because more complex models have more parameters and are capable of capturing intricate patterns in the training data.

However, increasing the complexity too much can lead to overfitting, causing the model to learn noise in the training data and resulting in high variance. This means that the model performs well on the training data but poorly on unseen data.

Task 4: Calculating Irreducible Error

Degree	Irreducible Error
1	2.3245294578089216e-17
2	2.480654570646834e-16
3	2.463307335887066e-17
4	8.465450562766818e-17
5	1.4224732503009818e-16
6	3.8857805861880476e-17
7	2.3592239273284576e-16
8	1.9151347174783951e-16
9	3.3584246494910984e-16
10	1.1102230246251566e-17

Irreducible error:

Irreducible error remains constant regardless of how complex the model becomes. It represents the inherent randomness or noise in the data that cannot be eliminated by any model. Increasing the complexity of the model may reduce bias and increase variance, but it cannot reduce the irreducible error. It serves as a fundamental limit on the performance of any model and underscores the importance of understanding and accounting for data uncertainty in predictive modelling.

Task 5: Plotting Bias^2 - Variance Graph

The bias-variance trade-off is a balance between a model's ability to fit data well and its ability to work with new data. A model can be too simple, making it not work well with data (high bias), or too complex, making it only work well with specific data (high variance). The perfect model is balanced between bias and variance, and depends on the type of data.

Underfitting:

When the model is too simple and has high bias, it may underfit the data, and the total error is high. This is shown on the plot as a high bias^2 and low variance. Underfitting occurs when the model is not complex enough to capture the patterns in the data, and it generalizes poorly to new data.

In our data, the underfitting seems to happen at degree 1 and 2 (approximately), as the model is not able to determine correctly the values of prediction (from prediction-original graph for each degree), and there is a high error.

Overfitting:

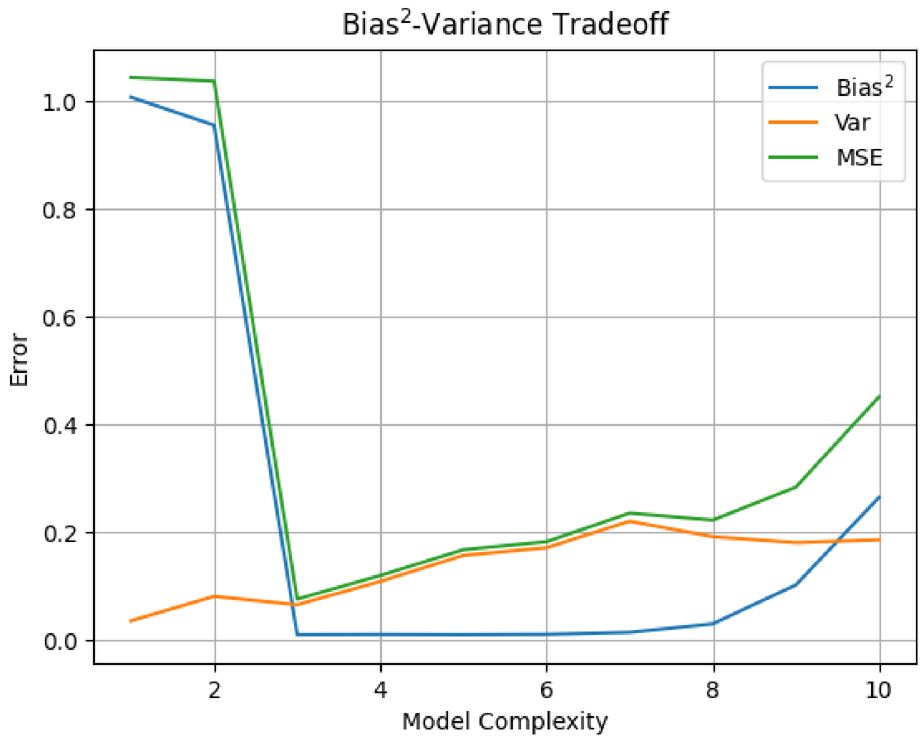
When the model is too complex and has high variance, it may overfit the data, and the total error is high. This is shown on the plot as a low bias^2 and high variance. Overfitting occurs when the model captures noise in the training data and does not generalize well to new data.

In our data, the overfitting seems to happen after degree 8 (approximately), as the model is not able to correctly determine the values, and rather is trying to over-generalize the given test data.

Type of Data:

From the plot of Bias^2-Variance, we can see that as the degree of the polynomial model increases, the bias initially decreases, while the variance initially increases and then starts to increase rapidly. For higher degrees of polynomial model, we see that the bias starts to increase, while the variance increases even more rapidly. Also, the irreducible error almost remains constant, except when the value of variance rises sharply.

This mostly implies that data is relatively simple, and that the optimal degree of polynomial model is somewhere around degree 5, where the bias and variance are both relatively low, and the mean squared error (MSE) is also relatively low.



Other Graphs: