★ Bias Variance Tradeoff

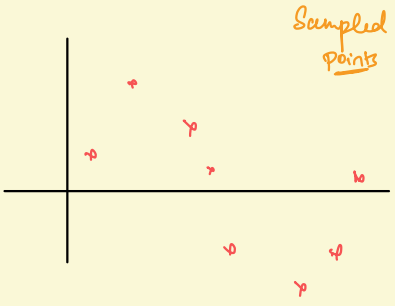
Source: Coursera Andrew Ng
Mitesh M Khapra - Deep Learning NPTEL Week 8
Soundarya Sarathi

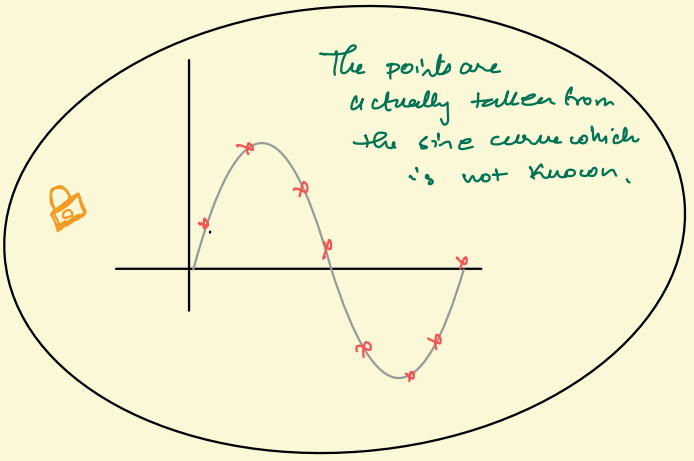
Overview of Bias - Variance:

Consider prob. of fitting a curve through a given set of points.

∃ a true relation b/w X & Y which is unknown.

Consider the points . . . .



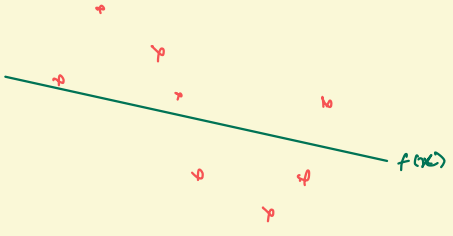


Two models

Linear model
Simple (deg 1)

1) $f(x) = w_1 x + w_0$

How do we get this line?: By loss function

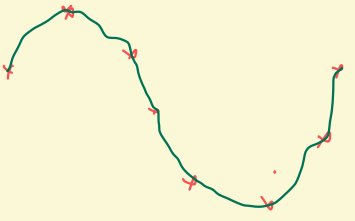


2) On average trying to minimize the distance of f from all points. That's how we get f.

Polynomial model.
Complex (deg 25)

1) $f(x) = \sum_{i=1}^{25} w_i x^i + w_0$

a degree 25 polynomial obtained through gradient descent.

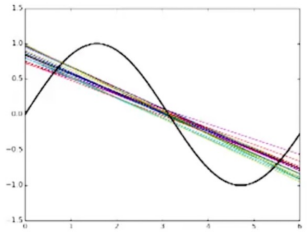


2) It's a complex model.

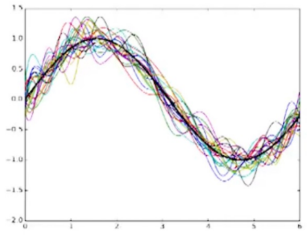
We sample 25 points from training data & train a simple & complex model and repeat the process K times to train multiple models.

We won't get the same curve everytime since the training data is different each time, we get a different curve

(a)



(b)



- Simple models trained on different samples of the data do not differ much from each other
- However they are very far from the true sinusoidal curve (under fitting)
- On the other hand, complex models trained on different samples of the data are very different from each other (high variance)
- But they are closer to the true sinusoidal curve (over fitting)

the different colors correspond to different curves obtained using different training samples

Bias = "has a very strong wrong preconception about the data".

① **Bias**:

let $f(x)$ = true model

$\hat{f}(x)$ = estimate of the model using a set of training points

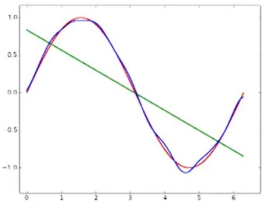
$$\text{Bias}\left(\hat{f}(x)\right) = E\left(\hat{f}(x)\right) - f(x)$$

$E(\hat{f}(x))$ = Expectation over all the approximate functions obtained by training different data samples. (here, expectation of all those diff. colored lines)

i.e. It is the difference b/w true model and $E(\hat{f}(x))$

In (a) this difference is very high.

In (b) this difference is low.



Green Line: Average value of $\hat{f}(x)$ for the simple model
Blue Curve: Average value of $\hat{f}(x)$ for the complex model
Red Curve: True model ($f(x)$)

② **Variance**:

$$\text{Var}\left(\hat{f}(x)\right) = E\left[\left(\hat{f}(x) - E\left(\hat{f}(x)\right)\right)^2\right]$$

describes the spread of the different approx. models.

In (a) All models obtained using different data points are close to each other and not very far from each other. Hence variance is low

In (b) all the models are very different from each other. Spread is very high.

∴ Variance is high

In (a): Simple model: low variance, high bias

In (b): Complex : High variance, low bias.

Is it always the case?

- Train Error vs Test Error

Consider a new point (x, y) not seen during training.

Mean square Error = $E[(y - \hat{f}(x))^2]$ (Averaging square error in predicting y for many such unseen points)

Why is $(y - \hat{f}(x))^2$ is a Rand. variable?

∵ x feeding during test time is going to vary ∴ it's a rand. variable
$\hat{y}$ p x is changing.

y = test data.

Error

$\star$ $E((y - \hat{f}(x))^2) = \text{Bias}^2 + \text{Variance} + \sigma^2$ (irred. error)

Bias = $E(\hat{f}(x)) - f(x)$

Variance = $E((\hat{f}(x) - E(\hat{f}(x)))^2)$

Variance = $\frac{1}{m} \sum_{i=1}^{m} (x_i - \bar{x})^2$

$\text{Var} = E((X - E(X))^2) = E(X^2 + (E(X))^2 - 2XE(X))$

$= E(X^2) + (E(X))^2 - 2(E(X))^2 = E(X^2) - (E(X))^2$

$\text{Error} = E((\hat{f} - f)^2) = E((f - E(\hat{f}) - \hat{f} + E(\hat{f}))^2)$

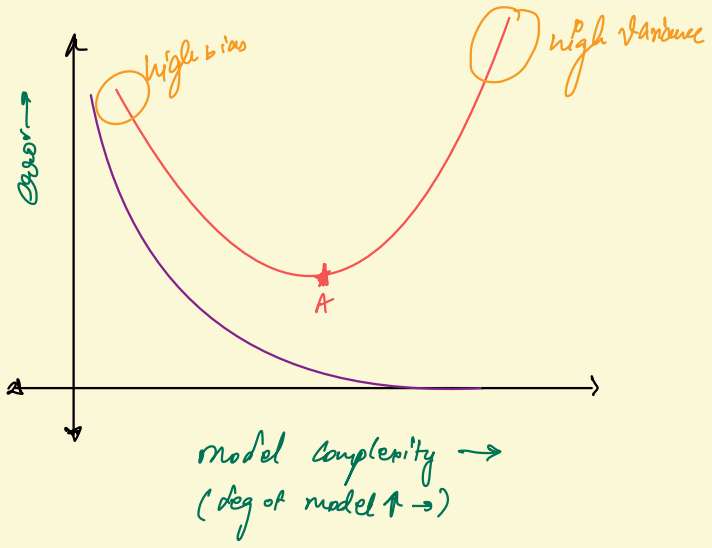
$= E((f - E(\hat{f}))^2) + E((\hat{f} - E(\hat{f}))^2) - 2E((\hat{f} - E(\hat{f}))(f - E(\hat{f})))$ (the last term $= 0$)

(Refer Pg 49 TB Hai Thang)

Error = $\text{Bias}^2 + \text{Variance} + \sigma^2$ (irred. Error)

- If Bias is high, Error ↑ and also if Variance is high, error ↑.
- That's why we don't want a very high bias or a very high variance.
  Hence we want something in b/w. high bias & high variance.

* 2 Errors: train error
test error



- As model complexity ↑, all the training data fits perfectly hence the training error becomes negligible as model complexity ↑.
- The training error is minimal @ point A. So overfitting will produce a high training error.

b Let there be n training points & m test points.

$$\text{train err} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

$$\text{test err} = \frac{1}{m}\sum_{i=n+1}^{n+m}(y_i - \hat{f}(x_i))^2$$

As model complexity ↑ train err becomes overly optimistic and gives a corong picture of how close $\hat{f}$ is to f.

$$y_i = f(x_i) + \varepsilon_i$$

means, y_i is related to x_i by some true relation but there is also some noise ε in the relation.

Assume $\varepsilon \sim N(0, \sigma^2)$ for simplicity.

We use $\hat{f}$ to approximate f and the parameter using $T \subset D$ s.t

$$y_i = \hat{f}(x_i)$$

As the model complexity ↑, train error becomes overly optimistic & gives a corong pic of how close $\hat{f}$ is to f

We are interested in Knowing $E[(\hat{f}(x_i) - f(x_i))^2]$

$$E[(\hat{y}_i - y_i)^2] = E[\hat{f}(x_i) - f(x_i) - \varepsilon_i)^2] = E[(\hat{f}(x_i) - f(x_i))^2] - 2E[\varepsilon_i(\hat{f}(x_i) - f(x_i))] + E[\varepsilon_i^2]$$

$$\therefore E[(\hat{f}(x_i) - f(x_i))^2] = E[(\hat{y}_i - y_i)^2] - E[\varepsilon_i^2] + 2E[\varepsilon_i(\hat{f}(x_i) - f(x_i))]$$

$$E[(\hat{y}_i - y_i)^2] = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

We can Empirically evaluate RHS by using training or test observations.

Case I · Using test observations

$$E\left(\left(\hat{f}(x_i) - f(x_i)\right)^2\right) = \frac{1}{m}\sum_{i=n+1}^{n+m}\left(y_i - \hat{y}_i\right)^2 - \frac{1}{m}\sum_{i=n+1}^{n+m}\varepsilon_i^2 + 2E\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right]$$

- $E\left(\left(\hat{f}(x_i) - f(x_i)\right)^2\right)$: true error
- $\frac{1}{m}\sum_{i=n+1}^{n+m}\left(y_i - \hat{y}_i\right)^2$: Empirical Estimation of error
- $\frac{1}{m}\sum_{i=n+1}^{n+m}\varepsilon_i^2$: small constant
- $2E\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right]$: covariance $\varepsilon_i^2\left(\hat{f}(x_i) - f(x_i)\right)$

When we have covariance b/w 2 things the two things must be random variables.

$E\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right]$

Parameters of $\hat{f}$ were known from the training data ε_i's are coming from test data that's why ε_i & $\left(\hat{f}(x_i) - f(x_i)\right)$ are independent.

$$\text{Cov}(X, Y) = E\left[(X - \mu_X)(Y - \mu_Y)\right] \quad (\mu_\varepsilon = 0,\ \hat{f} - f)$$
$$= E[XY - \mu_X Y - X\mu_Y + \mu_X\mu_Y]$$
$$= E(XY) - E[X\mu_Y] = E(XY) - \mu_Y E(X)$$
$$= E(XY)$$

$\therefore \varepsilon \perp \left(\hat{f}(x_i) - f(x_i)\right)$ i.e., $E\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right] = E(\varepsilon_i)\,E\left(\hat{f}(x_i) - f(x_i)\right) = 0 \cdot E(\hat{f} - f) = 0$

$\therefore$ True error = empirical test error + small const.

This tells if a model is trained & then I calculate the estimate of the error, on some data which was not used for the training, then that error is actually very close to the true error. It only differs by a constant.

Eg We are trying to estimate the true error (which cannot be explicitly estimated) by using the data. If data used is test data then empirical Estimation of test error is close to the true error.

Hence to estimate the error, we should always use a validation set (independent of the training set).

Case II Using training observations.

$$E\left[\left(\hat{f}(x_i) - f(x_i)\right)^2\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 - \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2 + 2E\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right]$$

- $\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2$: Empirical estimation of train error
- $\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2$: small constant
- $2E\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right]$: covariance $\left[\varepsilon_i\left(\hat{f}(x_i) - f(x_i)\right)\right] \neq 0$

$\varepsilon \not\perp \hat{f}(x)$ because ε used for estimating parameters of $\hat{f}(x)$

$$E\left[\varepsilon_i \left[f(x_i) - \hat{f}(x_i)\right]\right] \neq E[\varepsilon_i] E\left[\hat{f}(x_i) - f(x_i)\right] \neq 0.$$

- Empirical train error is small compared to the true error. & doesn't give a true picture of the error. In other words, its overly optimistic.

Stein's Lemma

$$\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \left[\hat{f}(x_i) - f(x_i)\right] = \frac{\sigma^2}{n}\sum_{i=1}^{n} \frac{\partial \hat{f}(x_i)}{\partial y_i}$$

The derivative is high if $\hat{f}$ change y/s a bit, the change in $\hat{f}(x_i)$ is going to be large.

↓

This happens for complex models (because Bias is high recall!)

Hence,

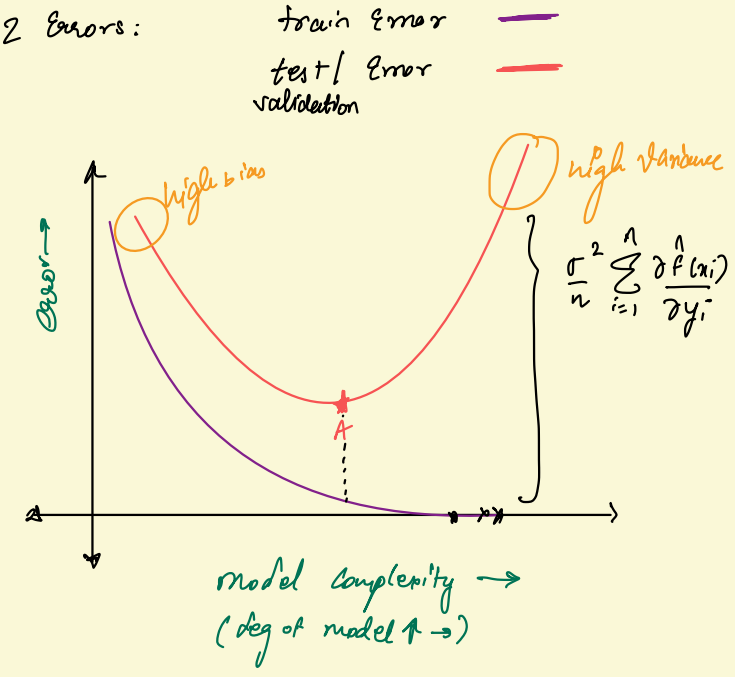
true error = Empirical train error + small const + Ω (model complexity)

$$\min_{\text{wrt } \theta} \mathcal{L}_{\text{train}}(\theta) + \Omega(\theta) = \mathcal{L}(\theta)$$

where $\Omega(\theta)$ would be high for complex models & small for simple models.

$\Omega(\theta)$ acts as an approximate of $\frac{\sigma^2}{n}\sum_{i=1}^{n} \frac{\partial \hat{f}(x_i)}{\partial y_i}$.

- 2 Errors:



we are aiming for the point A.
We should ensure using $\Omega(\theta)$ that the gap is also minimised.

$$\min_{\theta} \mathcal{L}(\theta) + \Omega(\theta)$$

Hence model would generalize better on the test data.

* L^2 Regularization:

$$\tilde{L}(w) = L(w) + \frac{\alpha}{2}\|w\|^2$$

↳ $\Omega(\theta)$, relating to model complexity

by minimizing $\|w\|$, we are reducing the complexity of the model by ensuring that the weights don't take any possible value.

* Regularized Linear Regression:

$$L(w) = \arg\min_{w} \left(\sum_{i=1}^{m} (w^T x_i^{\circ} - y_i^{\circ})^2 + \lambda \|w\|_{L^2}^2 \right)$$

Closed form solution.

$$\frac{\partial}{\partial w}(w^T w) = 2w$$

$$\frac{\partial}{\partial x}(x^T y) = y$$

$$\sum_{i=1}^{m} (w^T x_i^{\circ} - y_i^{\circ})^2 = \|Xw - y\|^2 \quad , \quad \|w\|^2 = w^T w$$

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_m^T \end{bmatrix}_{m \times n}$$

m samples
n features.

$$L(w) = \arg\min_{w} \|Xw - y\|^2 + \lambda \|w\|^2$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \frac{\partial}{\partial w}\left((Xw - y)^T (Xw - y) \right) + \lambda \frac{\partial}{\partial w}(w^T w) = 0$$

$(Xw^T X^T - y^T)(Xw - y)$

$$\Rightarrow \frac{\partial}{\partial w}\left(-y^T Xw + y^T y + w^T X^T X w - w^T X^T y \right) + 2w\lambda = 0$$

$$\Rightarrow -y^T X + \frac{\partial}{\partial w}(X w X w) - X^T y + 2w\lambda = 0 \qquad (A-B)^T = A^T - B^T$$

$$\Rightarrow -y^T X + \frac{\partial}{\partial w}(X X^T w^T w) - X^T y + 2w\lambda = 0$$

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

$$\Rightarrow 2(X^T X + \lambda I) w = 2X^T y$$

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

which is the closed form solution.

* Regularized linear regression.

$$\min_{w,b} J(w,b) = \min_{w,b} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2 \right)$$

m samples
n features

$$\frac{\partial}{\partial w_j} J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j$$

$$\frac{\partial}{\partial b} J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)$$

Gradient Descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

} simultaneous update

For implementation:

$$\begin{bmatrix} dw_1 \\ \vdots \\ dw_n \end{bmatrix}^T = \frac{1}{m} \left[(f_{x_1} - y_1), \ldots, (f_{x_m} - y_m) \right]_{s_1 \ldots s_m} \begin{bmatrix} \text{feat 1} & \text{feat 2} & \ldots & \text{feat n} \\ x_1^1 & x_2^1 & \ldots & x_n^1 \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^m & x_2^m & \ldots & x_n^m \end{bmatrix}$$

(s: samples)

$$\begin{bmatrix} dw_1 \\ \vdots \\ dw_n \end{bmatrix} = \begin{matrix} \text{fe1} \\ \vdots \\ \text{fen} \end{matrix} \begin{bmatrix} x_1^1 & x_1^2 & \ldots & x_1^m \\ x_2^1 & x_2^2 & \ldots & x_2^m \\ \vdots & \vdots & & \vdots \\ x_n^1 & x_n^2 & \ldots & x_n^m \end{bmatrix} \begin{matrix} s_1 \\ s_2 \\ \ldots \\ s_m \end{matrix} \begin{bmatrix} f(x_1) - y_1 \\ f(x_2) - y_2 \\ \vdots \\ f(x_m) - y_m \end{bmatrix} + \frac{\lambda}{m} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$+ \frac{\lambda}{m} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}^T$$

$$X^T \cdot (f - y) + \frac{\lambda}{m} W$$

$$w_j = w_j - \frac{\alpha \lambda}{m} w_j - \frac{\alpha}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

$\left(1 - \frac{\alpha \lambda}{m}\right)$ = is a quantity smaller than 1 effectively reducing the size of w_j

* Regularized Logistic Regression

$$J(w,b) = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log\left(f_{w,b}(x^{(i)}) \right) + \left(1 - y^{(i)}\right) \log\left(1 - f_{w,b}(x^{(i)})\right) + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2$$

Gradient descent is same as the linear regression

Only $f_{w,b}(x)$ differs.

$$\frac{\partial}{\partial w_j} J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j$$

↳ only f differs as $\frac{1}{1 + e^{-z}}$

$$\frac{\partial}{\partial b} J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)$$