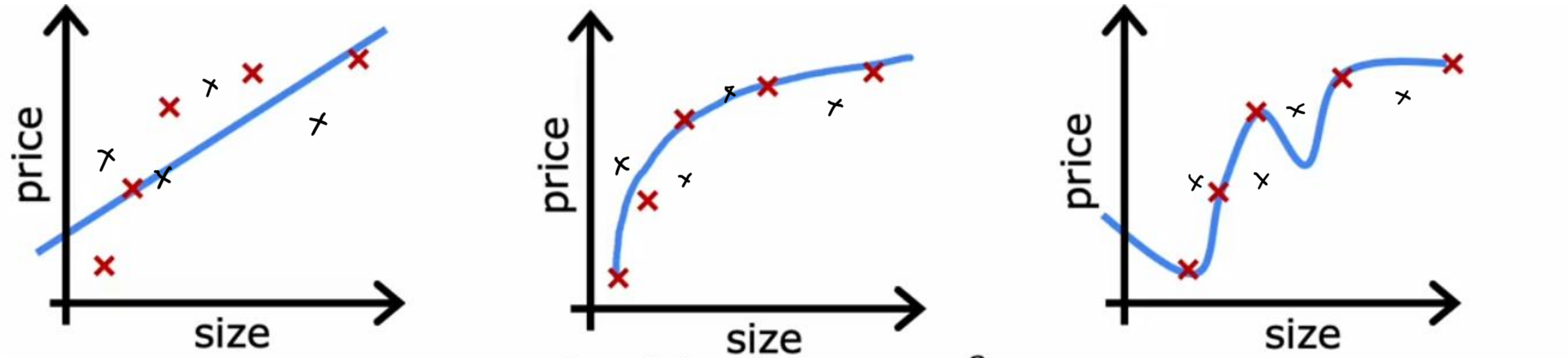
# Machine learning diagnostic

Bias & Variance

# Bias & Variance




$$f_{\vec{w},b}(x) = w_1x + b$$

High bias
(underfit)

$d=1$

$J_{train}$ high

$J_{cv}$ high

$$f_{\vec{w},b}(x) = w_1x + w_2x^2 + b$$

"Just right"

$d=2$

$J_{train}$ Low

$J_{cv}$ Low

$$f_{\vec{w},b}(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

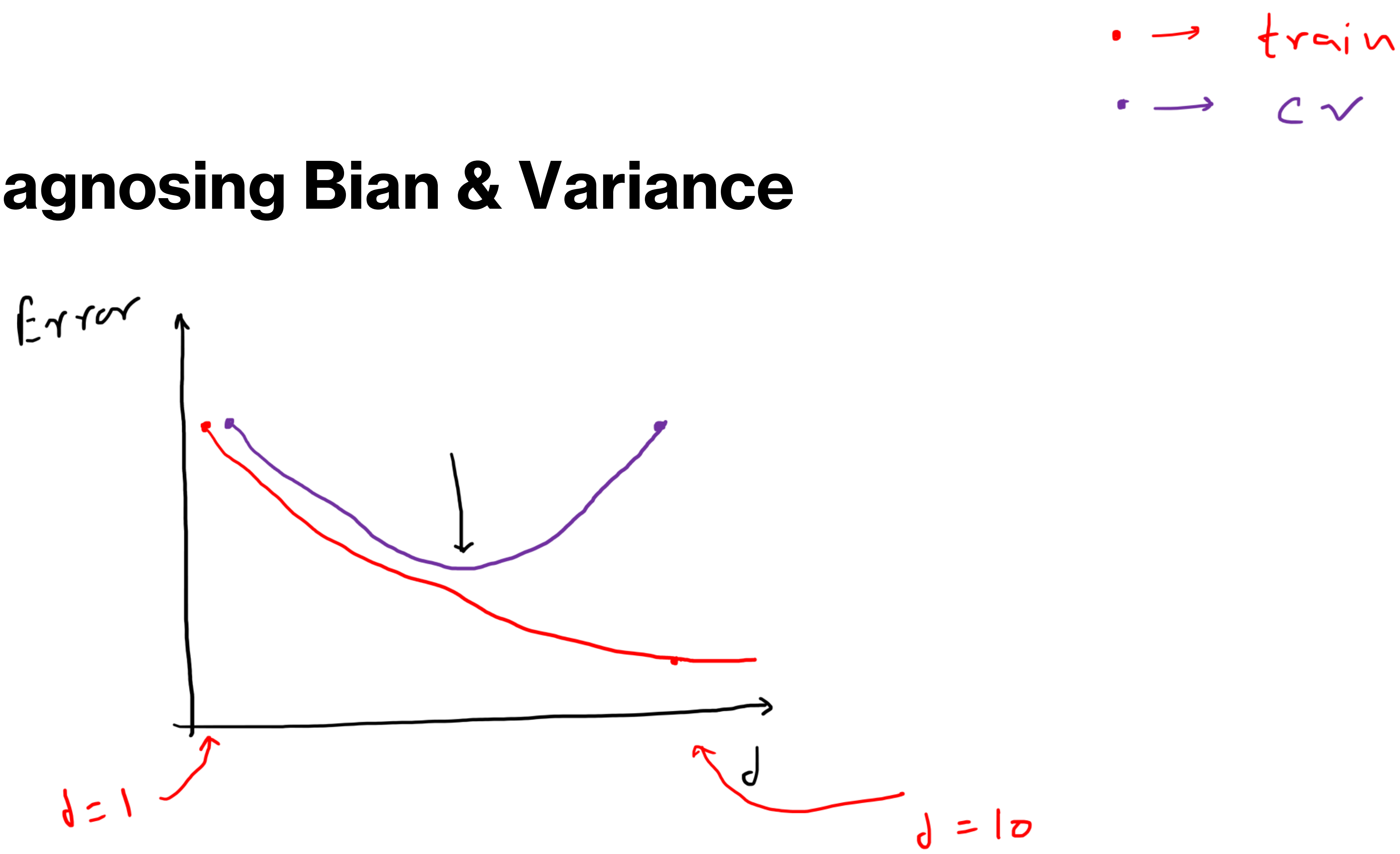
High variance
(overfit)

$d=4$

$J_{train}$ Low

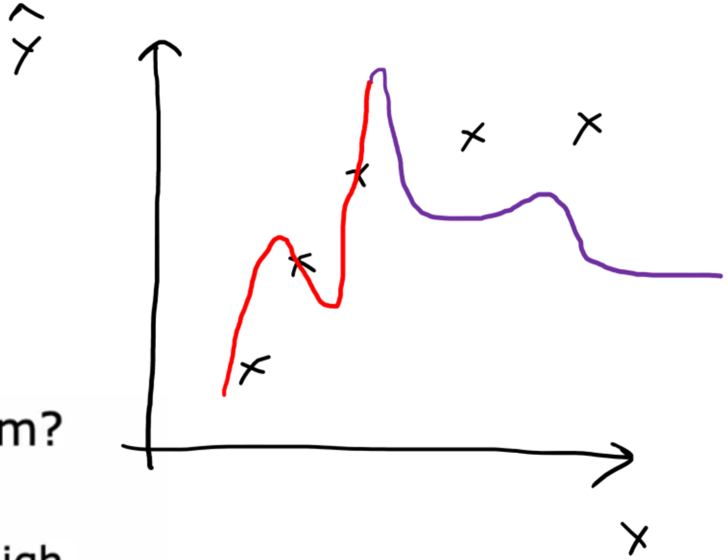
$J_{cv}$ high

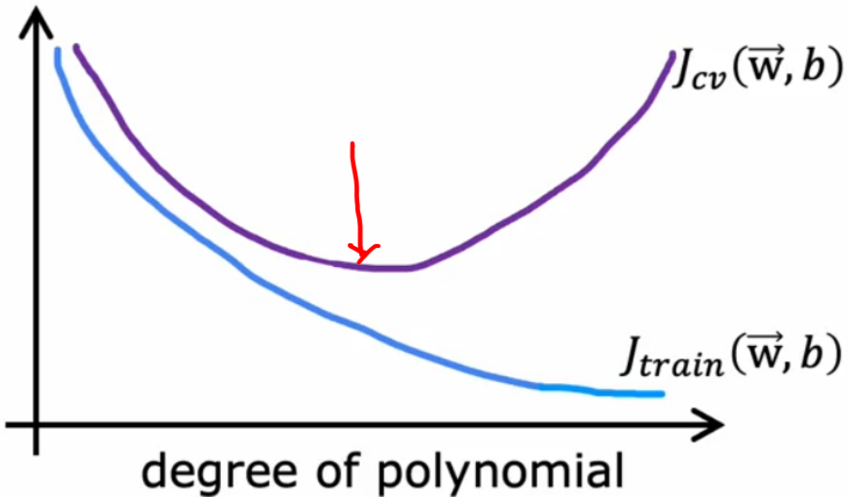
# Diagnosing Bian & Variance



# Diagnosing Bian & Variance

How do you tell if your algorithm has a bias or variance problem?




High bias (underfit)

$J_{train}$ will be high

( $J_{train} \approx J_{cv}$ )

High variance (overfit)

$J_{cv} \gg J_{train}$

( $J_{train}$ may be low)

High bias and high variance

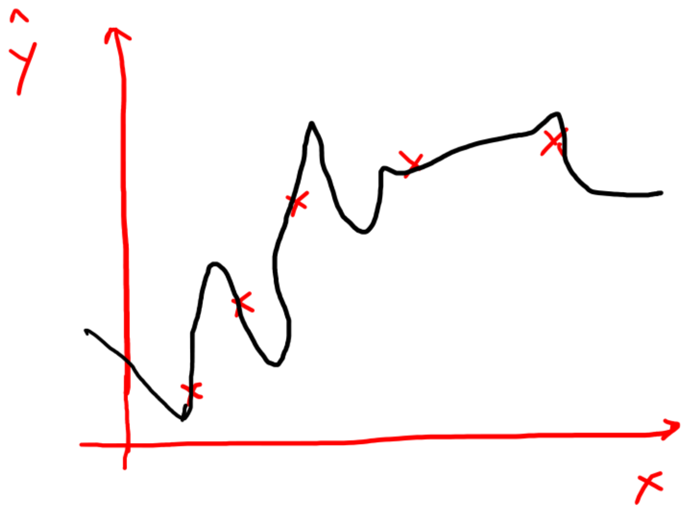
$J_{train}$ will be high

and $J_{cv} \gg J_{train}$

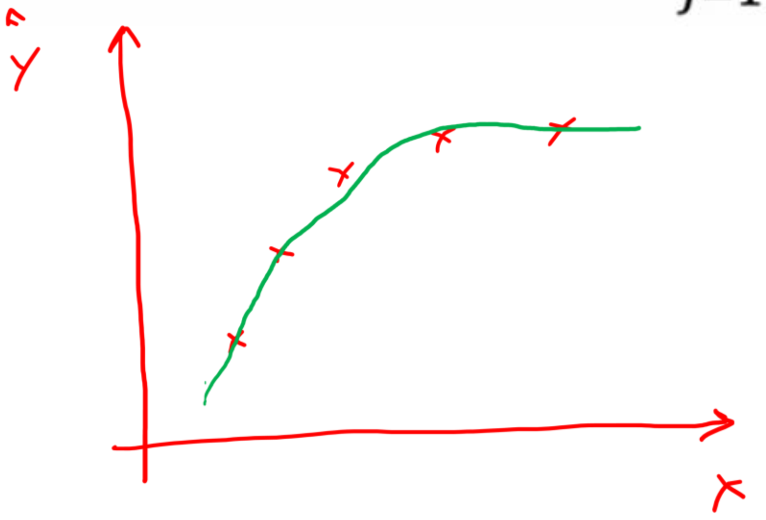
# Regularization and bias/variance

Model: $f_{\vec{\mathrm{w}},b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

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

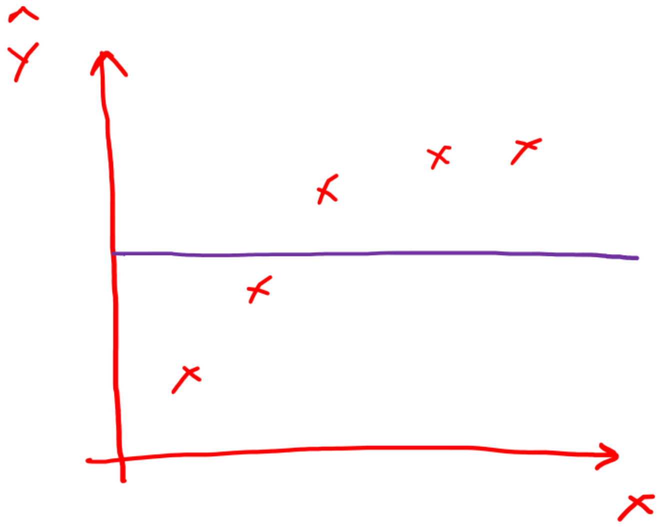



$\lambda \to 0$ High Variance




$\lambda \neq 0$

$\lambda \neq$ very high.




$\lambda \to \infty$

High Bias

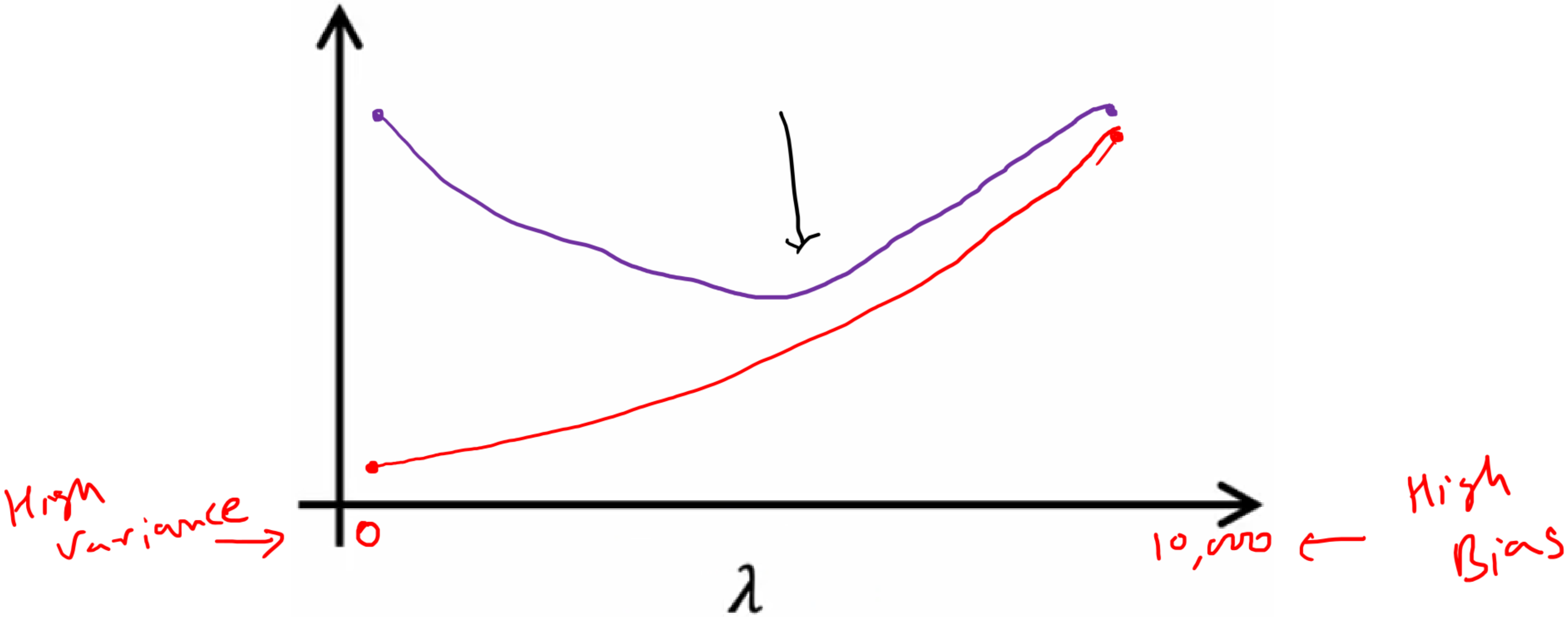
# Regularization parameter (λ) value

Model: $f_{\vec{\mathrm{w}},b}(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$

| | | | |
|---|---|---|---|
| 1. Try $\lambda = 0$ | $\min_{\vec{\mathrm{w}},b} J(\vec{\mathrm{w}},b)$ | $w^{<1>}, b^{<1>}$ | $J_{cv}(w^{<1>}, b^{<1>})$ |
| 2. Try $\lambda = 0.01$ | | $w^{<2>}, b^{<2>}$ | $J_{cv}(w^{<2>}, b^{<2>})$ |
| 3. Try $\lambda = 0.02$ | | | $J_{cv}(w^{<3>}, b^{<3>})$ |
| 4. Try $\lambda = 0.04$ | | | |
| 5. Try $\lambda = 0.08$ | | | $J_{cv}(w^{<5>}, b^{<5>})$ |
| $\vdots$ | | | |
| 12. Try $\lambda \approx 10$ | | $w^{<12>}, b^{<12>}$ | $J_{cv}(w^{<12>}, b^{<12>})$ |

# Bias & Variance as a function of λ

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



# Speech Recognition

$J_{human} = 10.6\%$

$J_{train} = 10.8\%$

$J_{cv} = 14.8\%$

$0.2\%$

$4\%$

High Variance

# Establishing a baseline level of performance

What is the level of error one can reasonably hope to get to?

- Human level performance
- Competing algorithms’ performance
- Guess based on experience

# Bian & Variance examples

- Baseline Performance
- Training error ($J_{train}$)
- Cross Validation error ($J_{cv}$)

10.6%
0.2%
10.8%
4%
14.8%

High Variance

10.6%
4.4%
15.0%
0.5%
15.5%

High Bias

10.6%
4.4%
15.0%
4.7%
19.7%

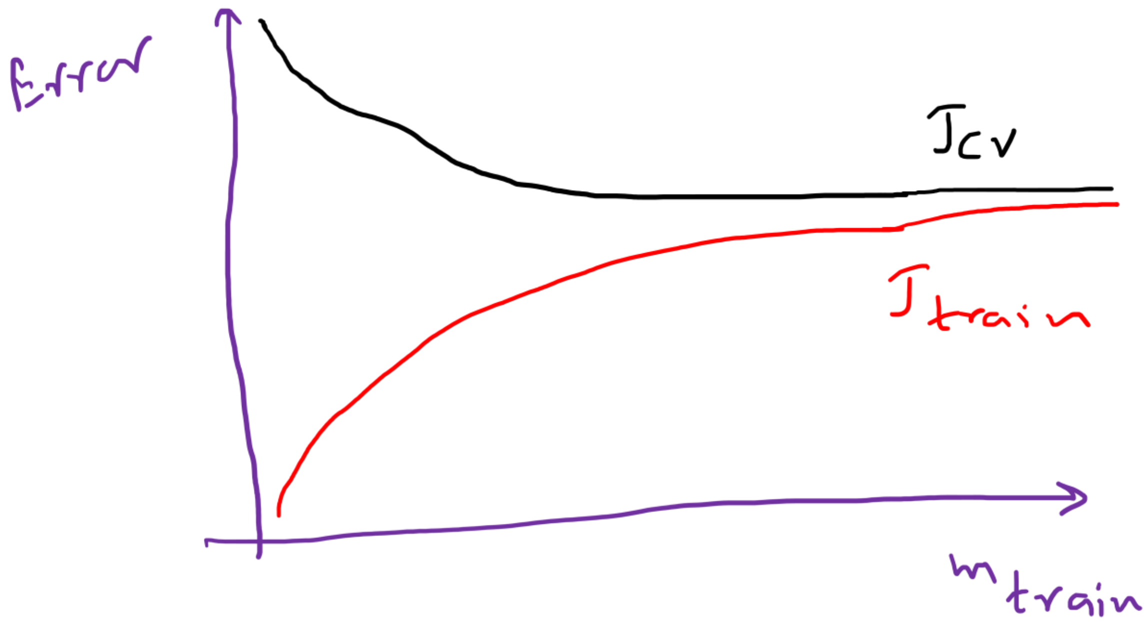
High Bias
High Variance

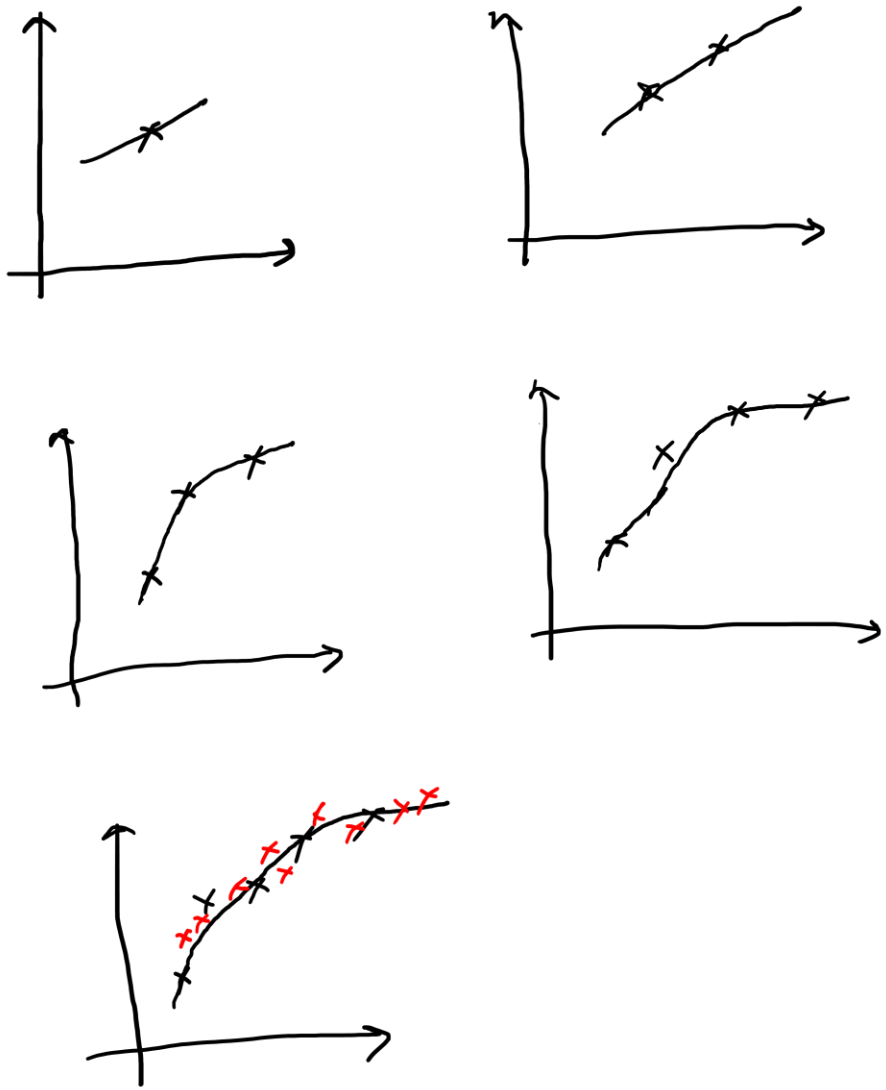
• → $J_{train}$

• → $J_{cv}$

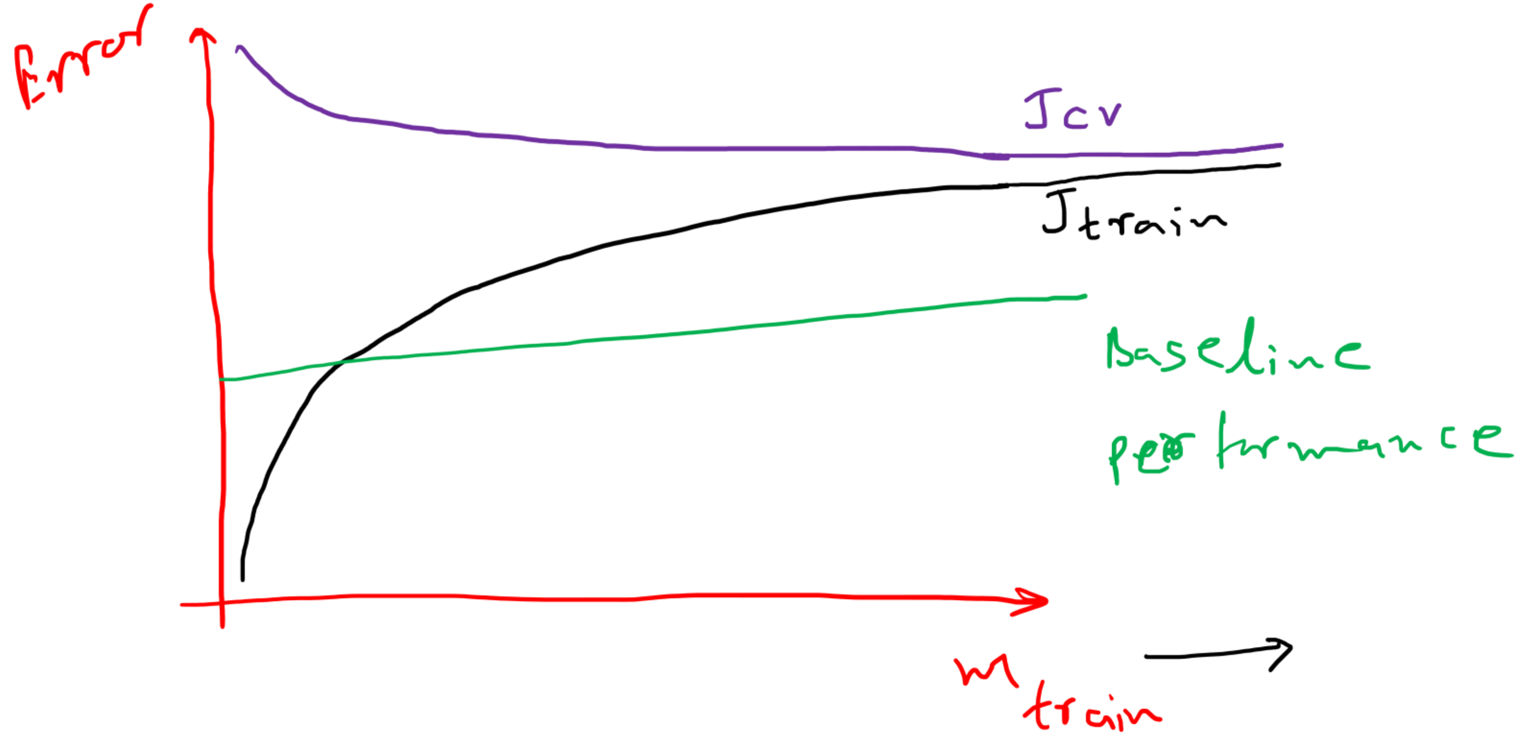
# Learning curves

$$f_{\vec{w},b}(\vec{x}) = w_1 x + w_2 x^2 + b$$

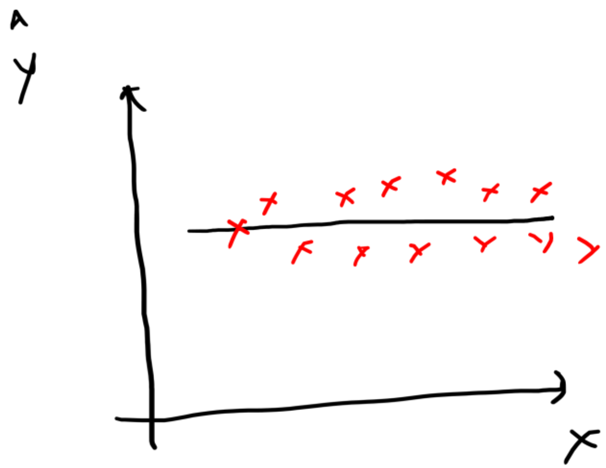


# High Bias

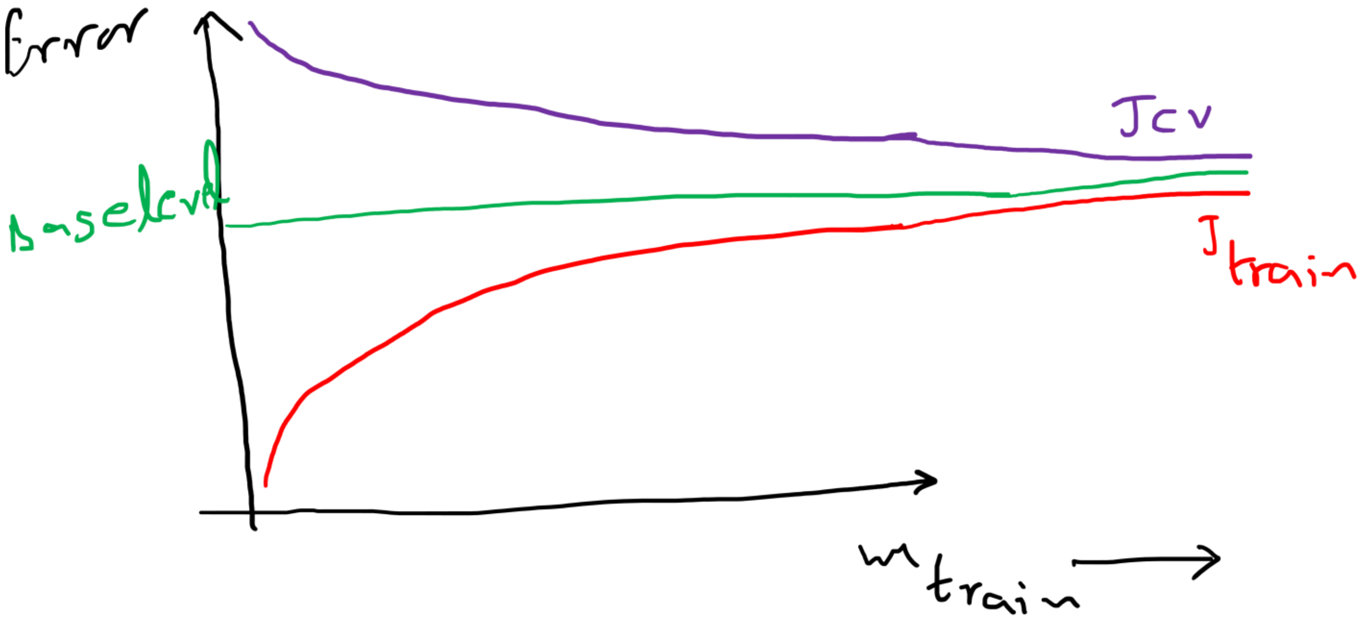


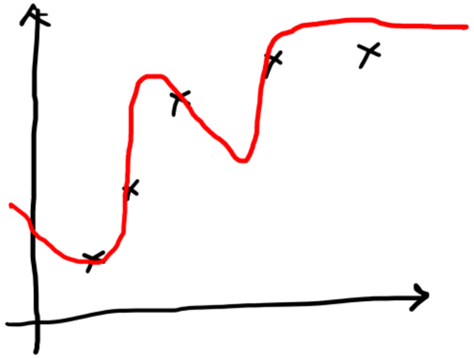

$f = b$

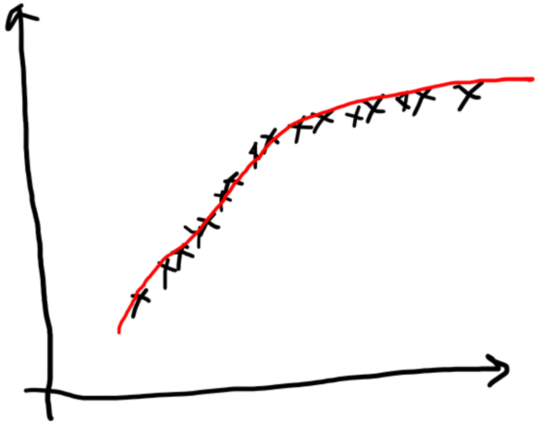


$$f = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

# High Variance





# Debugging a learning algorithm

- Regularized linear regression

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

- Large errors in prediction
- Solutions

1. Get more training examples
2. Try smaller set of features
3. Try getting additional features
4. Try adding polynomial features
5. Try decreasing λ
6. Try increasing λ