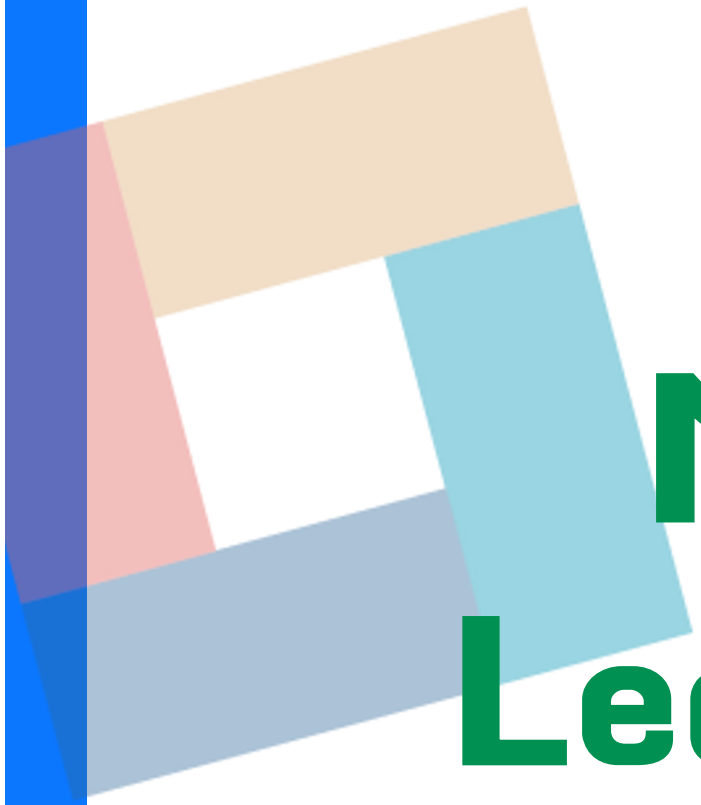


Machine Learning 08

Kihyun Shin
DMSE, HBNU

TheMAD

Back propagation

TheMAD

Derivative example

Cost function $J(w) = w^2$

Let's say $w = 3$ $J(w) = 3^2 = 9$

If we increase w by a tiny amount $\varepsilon = 0.001$, how does $J(w)$ change ?

$w = 3 + 0.001$

$J(w) = w^2 = 9.006001$

If $w \uparrow 0.001$

If $J(w) \uparrow 6 \times 0.001$

ε

$6 \times \varepsilon$

$\frac{\partial}{\partial w} J(w) = 6$

$\varepsilon = 0.002$

$w = 3 + 0.002$

$J(w) = w^2 = 9.012004$

If $w \uparrow 0.002$

If $J(w) \uparrow 6 \times 0.002$

ε

$6 \times \varepsilon$

$\frac{\partial}{\partial w} J(w) = 6$

If epsilon is small enough, there will be no problem

Informal Definition of Derivative

If $w \uparrow \varepsilon$ causes $J(w) \uparrow k \times \varepsilon$ then

$$\frac{\partial}{\partial w} J(w) = k$$

Gradient descent

repeat {

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

}

If derivative → small, then update step → small
If derivative → large, then update step → large

More Derivative Examples

$w = 3$ $J(w) = w^2 = 9$ $w \uparrow 0.001$ $J(w) = J(3.001) = 9.006001$ $\frac{\partial}{\partial w} J(w) = 6$

$J(w) \uparrow 6 \times 0.001$

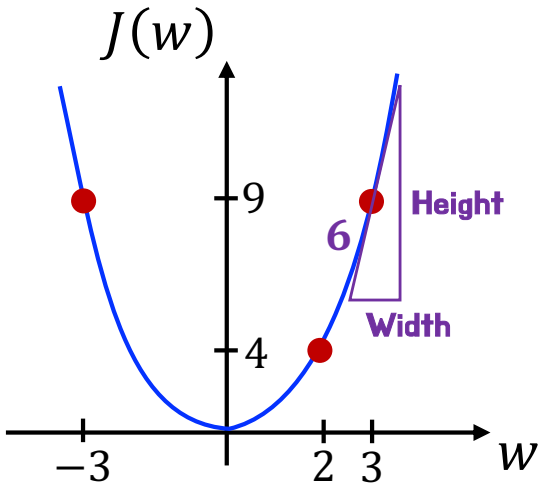
$w = 2$ $J(w) = w^2 = 4$ $w \uparrow 0.001$ $J(w) = J(2.001) = 4.004001$ $\frac{\partial}{\partial w} J(w) = 4$

$J(w) \uparrow 4 \times 0.001$

$w = -3$ $J(w) = w^2 = 9$ $w \uparrow 0.001$ $J(w) = J(-2.999) = 8.994001$ $\frac{\partial}{\partial w} J(w) = -6$

$J(w) \downarrow 6 \times 0.001$

$J(w) \uparrow -6 \times 0.001$



Calculus	w	$\frac{\partial}{\partial w} J(w)$
$\frac{\partial}{\partial w} J(w) = 2w$	3	$2 \times 3 = 6$
	2	$2 \times 2 = 4$
	-3	$2 \times -3 = -6$

TheMAC

Even More Derivative Examples

$w = 2$

$J(w) = w^2 = 4$	$\frac{\partial}{\partial w} J(w) = 2w = 4$	$w \uparrow 0.001$	$J(w) = 4.004001$ $J(w) \uparrow 4 \times 0.001$
$J(w) = w^3 = 8$	$\frac{\partial}{\partial w} J(w) = 3w^2 = 12$	$w \uparrow \varepsilon$	$J(w) = 8.012006$ $J(w) \uparrow 12 \times \varepsilon$
$J(w) = w = 2$	$\frac{\partial}{\partial w} J(w) = 1$	$w \uparrow \varepsilon$	$J(w) = 2.001$ $J(w) \uparrow 1 \times \varepsilon$
$J(w) = \frac{1}{w} = \frac{1}{2}$	$\frac{\partial}{\partial w} J(w) = -\frac{1}{w^2} = -\frac{1}{4}$	$w \uparrow \varepsilon$ $w = \frac{1}{2.001}$	$J(w) = 0.49975$ $= 0.5 - 0.25 \times 0.001$ $J(w) \uparrow -\frac{1}{4} \times \varepsilon$

$$\boldsymbol{\frac{\partial}{\partial w} J(w)} \qquad \boldsymbol{w \uparrow \varepsilon} \qquad \boldsymbol{J(w) \uparrow k \times \varepsilon}$$

A Note on Derivative Notation

If $J(w)$ is a function of one variable (w),

$$\frac{d}{dw}J(w)$$

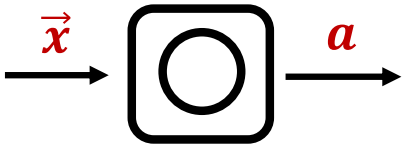
If $J(w_1, w_2, \dots, w_n)$ is a function of more than one variable,

$$\frac{\partial}{\partial w_i}J(w_1, w_2, \dots, w_n) \qquad \frac{\partial J}{\partial w_i} \quad \text{or} \quad \frac{\partial}{\partial w_i}J$$

Called "partial derivative"

Notation ∂ will be used in these courses

Small Neural Network Example

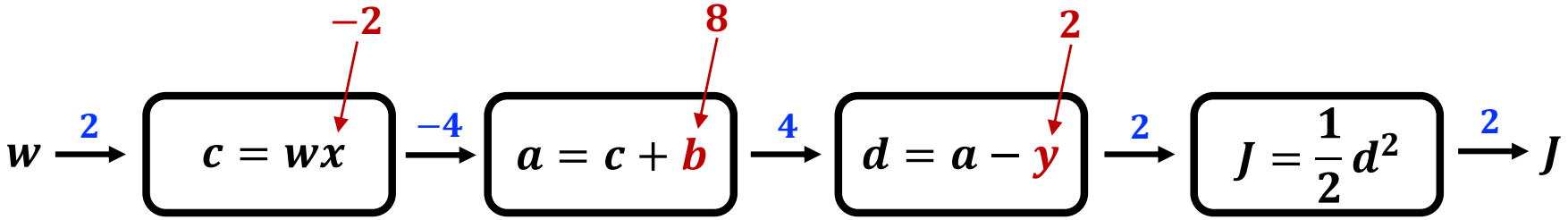


$w = 2$ $\quad x = -2$

$b = 8$ $\quad y = 2$

$a = wx + b$: Linear activation $a = g(z) = z$

$$J(w, b) = \frac{1}{2}(a - y)^2$$



Computation graph

Forward propagation

How do we find the derivatives of J? → $\frac{\partial J}{\partial w}$ or $\frac{\partial J}{\partial b}$

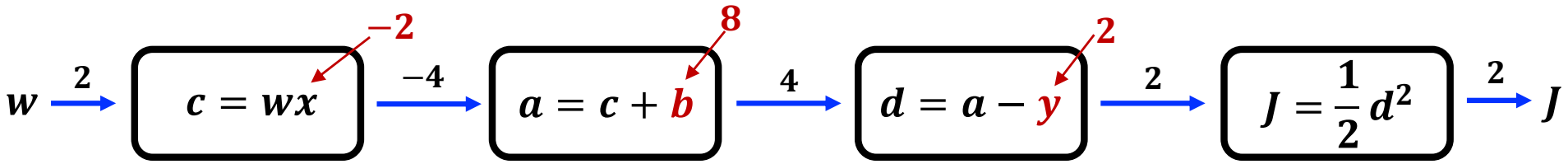
TheMAC

Computing the Derivatives

Forward prop

Back prop

$w = 2 \quad b = 8 \quad x = -2 \quad y = 2 \quad a = wx + b \quad J(w,b) = \frac{1}{2}(a-y)^2$



$\frac{\partial J}{\partial d} = 2$

$d \uparrow \frac{0.001}{\varepsilon} \quad J \uparrow \frac{0.002}{2\varepsilon}$

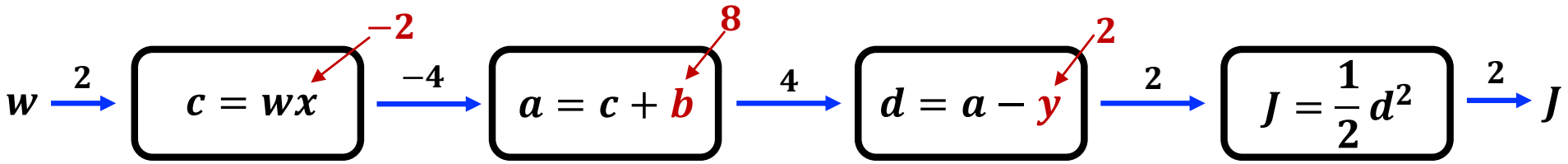
TheMAD

Computing the Derivatives

Forward prop

Back prop

$w = 2 \quad b = 8 \quad x = -2 \quad y = 2 \quad a = wx + b \quad J(w,b) = \frac{1}{2}(a-y)^2$



$\frac{\partial J}{\partial a} = 2$

$\frac{\partial J}{\partial d} = 2$

$a = 4.001 \quad d = 2.001$

$a \uparrow 0.001 \quad d \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$d \uparrow \frac{0.001}{\varepsilon} \quad J \uparrow \frac{0.002}{2\varepsilon}$

$$\frac{\partial J}{\partial a} = \frac{\partial d}{\partial a} \times \frac{\partial J}{\partial d}$$

2 1 2

Chain rule (calculus)

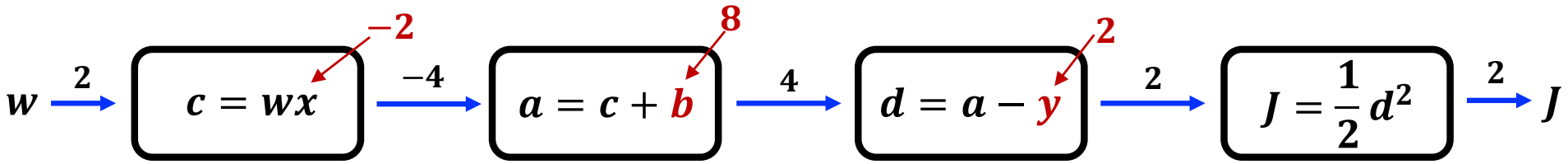
TheMAC

Computing the Derivatives

Forward prop

Back prop

$w = 2 \quad b = 8 \quad x = -2 \quad y = 2 \quad a = wx + b \quad J(w,b) = \frac{1}{2}(a-y)^2$



$\frac{\partial J}{\partial c} = 2 \quad \frac{\partial J}{\partial b} = 2$

$\frac{\partial J}{\partial a} = 2$

$\frac{\partial J}{\partial d} = 2$

$c \uparrow 0.001 \quad a \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$a = 4.001 \quad d = 2.001$

$a \uparrow 0.001 \quad d \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$d \uparrow \underline{0.001}_{\varepsilon} \quad J \uparrow \underline{0.002}_{2\varepsilon}$

$$\underset{2}{\frac{\partial J}{\partial c}} = \underset{1}{\frac{\partial a}{\partial c}} \times \underset{2}{\frac{\partial J}{\partial a}}$$

$b \uparrow 0.001 \quad a \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$$\underset{2}{\frac{\partial J}{\partial b}} = \underset{1}{\frac{\partial a}{\partial b}} \times \underset{2}{\frac{\partial J}{\partial a}}$$

$$\underset{2}{\frac{\partial J}{\partial a}} = \underset{1}{\frac{\partial d}{\partial a}} \times \underset{2}{\frac{\partial J}{\partial d}}$$

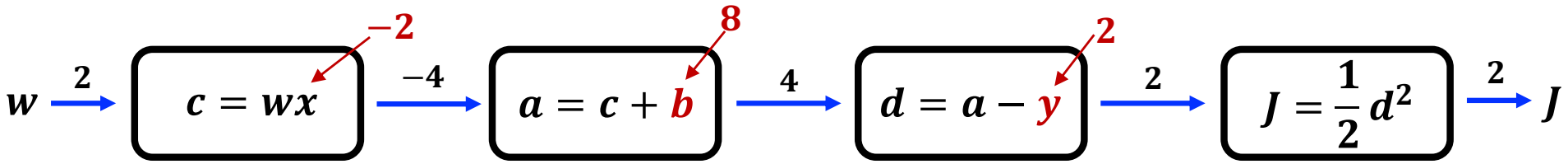
Chain rule (calculus)

Computing the Derivatives

Forward prop

Back prop

$w = 2 \quad b = 8 \quad x = -2 \quad y = 2 \quad a = wx + b \quad J(w,b) = \frac{1}{2}(a-y)^2$



$\frac{\partial J}{\partial w} = -4$

$\frac{\partial J}{\partial c} = 2 \quad \frac{\partial J}{\partial b} = 2$

$\frac{\partial J}{\partial a} = 2$

$\frac{\partial J}{\partial d} = 2$

$w = 2.001 \quad c = -4.002$

$w \uparrow 0.001 \quad c \downarrow 0.002$

$c \uparrow -0.002$

$\rightarrow J \uparrow -0.004$

$c \uparrow 0.001 \quad a \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$b \uparrow 0.001 \quad a \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$a = 4.001 \quad d = 2.001$

$a \uparrow 0.001 \quad d \uparrow 0.001$

$\rightarrow J \uparrow 0.002$

$d \uparrow \frac{0.001}{\varepsilon} \quad J \uparrow \frac{0.002}{2\varepsilon}$

$$\frac{\partial J}{\partial w} = \frac{\partial c}{\partial w} \times \frac{\partial J}{\partial c}$$
$$-4 \quad -2 \quad 2$$

$$\frac{\partial J}{\partial c} = \frac{\partial a}{\partial c} \times \frac{\partial J}{\partial a}$$
$$2 \quad 1 \quad 2$$

$$\frac{\partial J}{\partial b} = \frac{\partial a}{\partial b} \times \frac{\partial J}{\partial a}$$
$$2 \quad 1 \quad 2$$

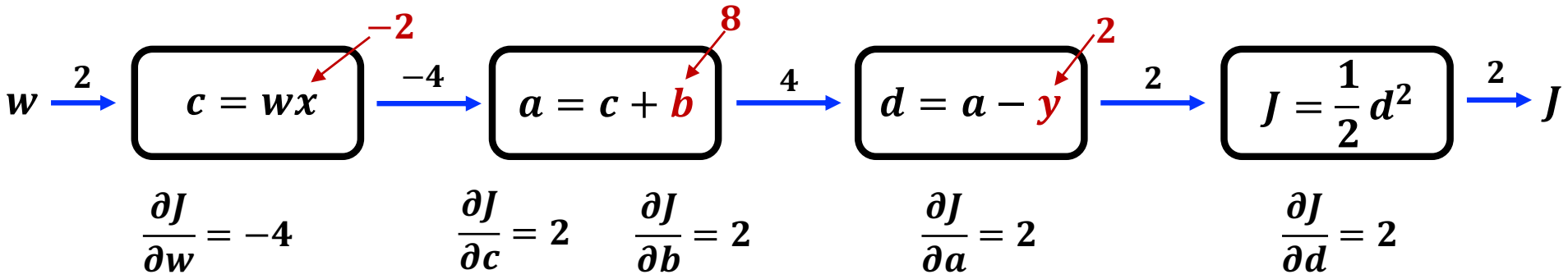
$$\frac{\partial J}{\partial a} = \frac{\partial d}{\partial a} \times \frac{\partial J}{\partial d}$$
$$2 \quad 1 \quad 2$$

Chain rule (calculus)

TheMAC

Computing the Derivatives

$w = 2 \quad b = 8 \quad x = -2 \quad y = 2 \quad a = wx + b \quad J(w,b) = \frac{1}{2}(a-y)^2$



$$J = \frac{1}{2}\big((wx+b)-y\big)^2 = \frac{1}{2}\big((2\times(-2)+8)-2\big)^2 = 2$$

$w \uparrow 0.001$

$$\rightarrow \frac{1}{2}\big((2.001\times(-2)+8)-2\big)^2 = 1.996002 \approx 2 - 0.004$$

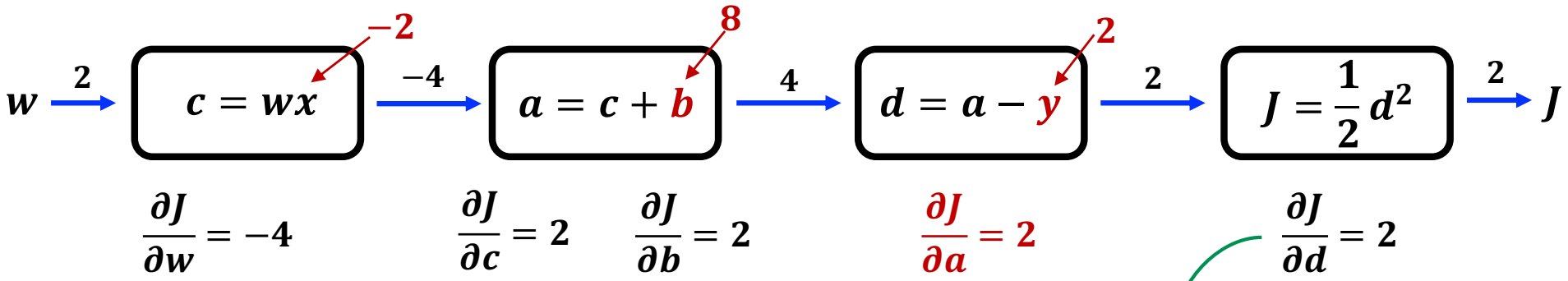
$J \downarrow \quad 4\times 0.001$

$J \uparrow -4\times 0.001$

$$\frac{\partial J}{\partial w} = -4$$

Backprop is an Efficient Way

$w = 2 \quad b = 8 \quad x = -2 \quad y = 2 \quad a = wx + b \quad J(w,b) = \frac{1}{2}(a-y)^2$

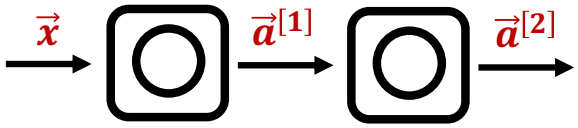


Compute $\frac{\partial J}{\partial a}$ once and use it to compute both $\frac{\partial J}{\partial w}$ and $\frac{\partial J}{\partial b}$

If N nodes and P parameters, compute derivatives in roughly N+P steps rather than N×P steps.

N	P	N+P	N×P
10^4	10^5	1.1×10^5	10^9

Neural Network Example



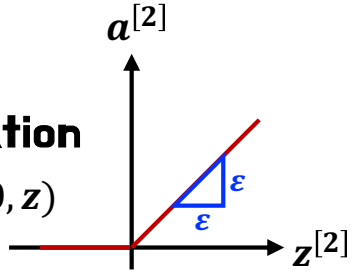
$x = 1$ $\quad y = 5$

$w^{[1]} = 2$ $\quad b^{[1]} = 0$

$w^{[2]} = 3$ $\quad b^{[2]} = 1$

ReLU activation

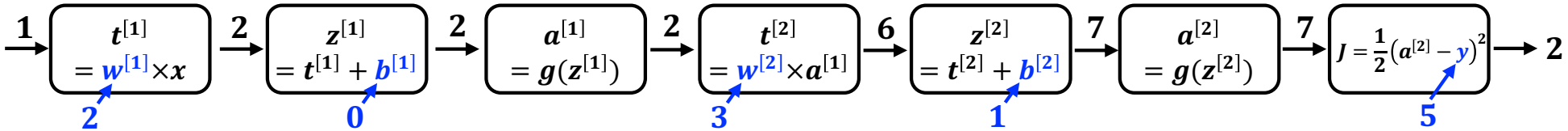
$g(z) = max(0, z)$



$$a^{[1]} = g(z^{[1]}) = g(w^{[1]}x + b^{[1]}) = g(2\times 1 + 0) = g(2) = 2$$

$$a^{[2]} = g(z^{[2]}) = g(w^{[2]}a^{[1]} + b^{[2]}) = g(3\times 2 + 1) = g(7) = 7$$

$$J(w,b) = \frac{1}{2}(a^{[2]} - y)^2 = \frac{1}{2}(7-5)^2 = 2$$



$\frac{\partial J}{\partial w^{[1]}} = 6$ $\quad w^{[1]} \uparrow 0.001$ $\quad J \uparrow 0.006$

$\frac{\partial J}{\partial t^{[1]}} = 6$ $\quad t^{[1]} \uparrow 0.001$ $\quad J \uparrow 0.006$

$\frac{\partial J}{\partial b^{[1]}} = 6$ $\quad b^{[1]} \uparrow 0.001$ $\quad J \uparrow 0.006$

$\frac{\partial J}{\partial z^{[1]}} = 6$ $\quad z^{[1]} \uparrow 0.001$ $\quad J \uparrow 0.006$

$\frac{\partial J}{\partial w^{[2]}} = 4$ $\quad w^{[2]} \uparrow 0.001$ $\quad J \uparrow 0.004$

$\frac{\partial J}{\partial a^{[1]}} = 6$ $\quad a^{[1]} \uparrow 0.001$ $\quad J \uparrow 0.006$

$\frac{\partial J}{\partial b^{[2]}} = 2$ $\quad b^{[2]} \uparrow 0.001$ $\quad J \uparrow 0.002$

$\frac{\partial J}{\partial t^{[2]}} = 2$ $\quad t^{[2]} \uparrow 0.001$ $\quad J \uparrow 0.002$

$\frac{\partial J}{\partial z^{[2]}} = 2$

$\frac{\partial J}{\partial a^{[2]}} = 2$

$\frac{\partial J}{\partial z^{[2]}} = \frac{\partial a^{[2]}}{\partial z^{[2]}} \times \frac{\partial J}{\partial a^{[2]}}$

2 $\quad$ 1 $\quad$ 2

$z^{[2]} \uparrow 0.001$ $\quad J \uparrow 0.002$

$a^{[2]} \uparrow 0.001$ $\quad J \uparrow 0.002$

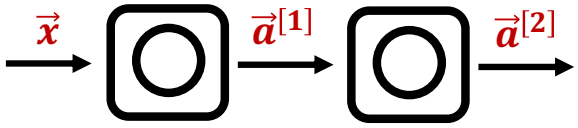
$$\frac{\partial J}{\partial w^{[1]}} = 6$$

Backpropagation

If $w^{[1]} \uparrow \varepsilon$, then $J \uparrow 6\times\varepsilon$. Let's verify this!

TheMAC

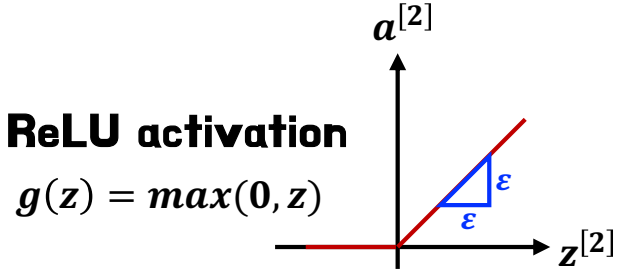
Neural Network Example



$x = 1$ $\quad y = 5$

$w^{[1]} = 2$ $\quad b^{[1]} = 0$

$w^{[2]} = 3$ $\quad b^{[2]} = 1$



$$a^{[1]} = g(z^{[1]}) = g(w^{[1]}x + b^{[1]}) = g(2\times 1 + 0) = g(2) = 2$$

$$a^{[2]} = g(z^{[2]}) = g(w^{[2]}a^{[1]} + b^{[2]}) = g(3\times 2 + 1) = g(7) = 7$$

$$J(w,b) = \frac{1}{2}(a^{[2]} - y)^2 = \frac{1}{2}(7-5)^2 = 2$$

$$a^{[1]} = g(z^{[1]}) = g(w^{[1]}x + b^{[1]}) = g(2.001\times 1 + 0) = g(2.001) = 2.001$$

$$a^{[2]} = g(z^{[2]}) = g(w^{[2]}a^{[1]} + b^{[2]}) = g(3\times 2.001 + 1) = g(7.003) = 7.003$$

$$J(w,b) = \frac{1}{2}(a^{[2]} - y)^2 = \frac{1}{2}(7.003-5)^2 = 2.006005$$

$w^{[1]} \uparrow 0.001$ $\quad J \uparrow 0.006$ $\quad \dfrac{\partial J}{\partial w^{[1]}} = 6$

Evaluating model

Debugging a Learning Algorithm

You've implemented regularized linear regression on housing prices

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

But it makes unacceptably large errors in predictions.
What do you try next?

- **Get more training examples**
- **Try smaller sets of features**
- **Try getting additional features**
- **Try adding polynomial features ($x_1^2, x_2^2, x_1x_2, etc$)**
- **Try decreasing λ**
- **Try increasing λ**

Machine Learning Diagnostic

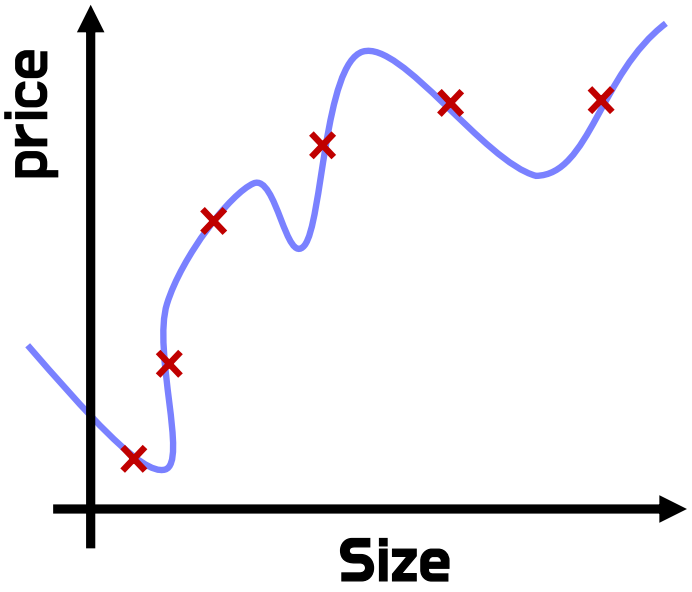
Diagnostic:

A test that you run to gain insight into what is/isn't working With a learning algorithm, to gain guidance into improving its performance

Diagnostics can take time to implement

But It will be useful to improve your model properly

Evaluating Your Model



Model fits the training data well,

but will fail to generalize to new

examples not in the training set

→ Need the way to evaluate model

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

- x_1 = size in feet2
- x_2 = no. of bedrooms
- x_3 = no. of floors
- x_4 = age of home in years

Evaluating Your Model

Dataset:

	size	price
	2200	390
	1700	325
	2450	365
	1400	220
	3100	536
70 %	1930	300
	1530	320
	1430	188
30 %	1375	210
	1485	235

Training set

m_{train} = no. training examples

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$(x^{(m_{train})}, y^{(m_{train})})$$

Test set

m_{test} = no. test examples

$$(x^{(1)}, y^{(1)})$$
$$\vdots$$
$$(x^{(m_{test})}, y^{(m_{test})})$$

Train/Test Procedure for Linear Regression

Fit parameters by minimizing cost function $J(\vec{w}, b)$

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

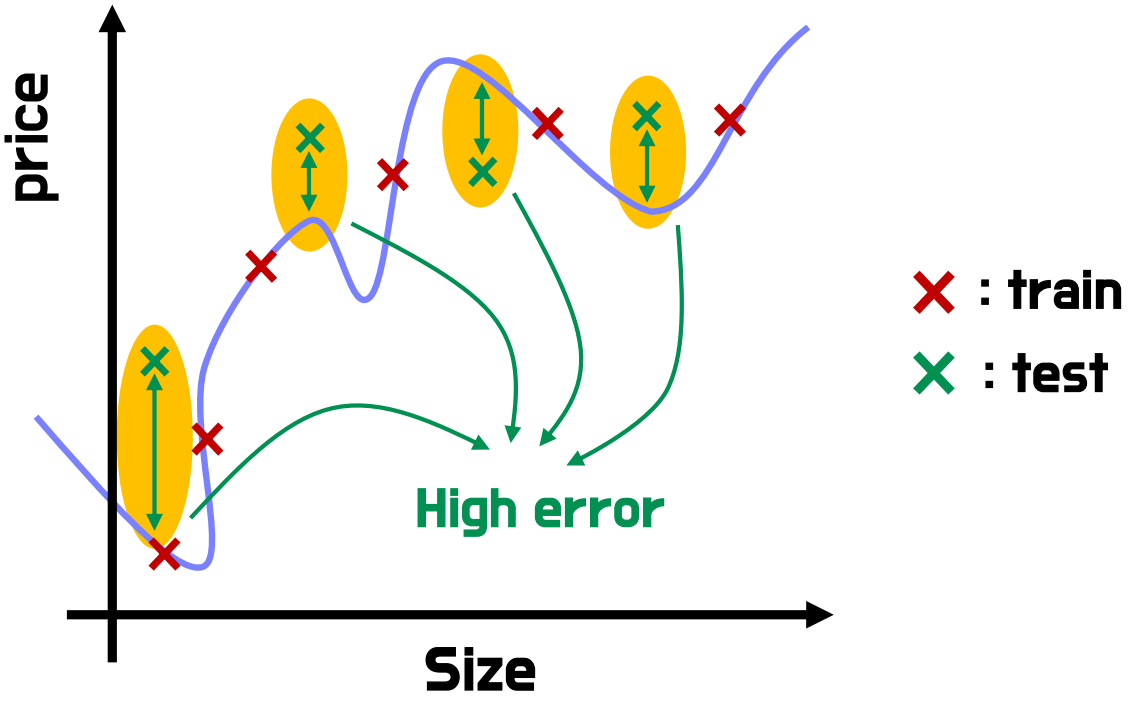
Compute test error:

$$J_{test}(\vec{w}, b) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(f_{\vec{w},b}\left(\vec{x}_{test}^{(i)}\right) - y_{test}^{(i)}\right)^2$$

Compute training error:

$$J_{train}(\vec{w}, b) = \frac{1}{2m_{train}}\sum_{i=1}^{m_{train}}\left(f_{\vec{w},b}\left(\vec{x}_{train}^{(i)}\right) - y_{train}^{(i)}\right)^2$$

Train/Test Procedure for Linear Regression



$J_{train}(\vec{w}, b)$ will be low $J_{test}(\vec{w}, b)$ will be high

Train/Test Procedure for Classification

Fit parameters by minimizing $J(\vec{w}, b)$ to find $\vec{w}, b$

For example,

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

Compute test error:

$$J_{test}(\vec{w}, b) = \frac{-1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log\left(f_{\vec{w},b}\left(\vec{x}_{test}{}^{(i)}\right)\right) + (1 - y_{test}^{(i)}) \log\left(1 - f_{\vec{w},b}\left(\vec{x}_{test}{}^{(i)}\right)\right)$$

Compute training error:

$$J_{train}(\vec{w}, b) = \frac{-1}{m_{train}} \sum_{i=1}^{m_{train}} y_{train}^{(i)} \log\left(f_{\vec{w},b}\left(\vec{x}_{train}{}^{(i)}\right)\right) + (1 - y_{train}^{(i)}) \log\left(1 - f_{\vec{w},b}\left(\vec{x}_{train}{}^{(i)}\right)\right)$$

Train/Test Procedure for Classification

Fraction of the test set and the fraction of the train set that the algorithm has misclassified.

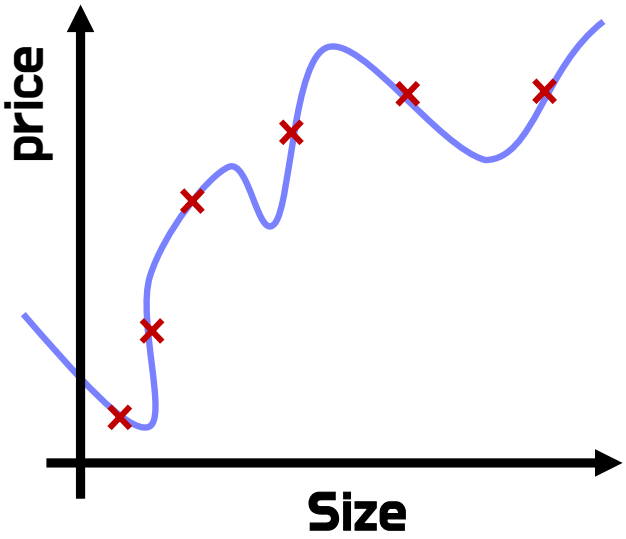
$$\hat{y} = \begin{cases} 1 \; if \; f_{\vec{w},b}(\vec{x}^{(i)}) \geq 0.5 \\ 0 \; if \; f_{\vec{w},b}(\vec{x}^{(i)}) < 0.5 \end{cases}$$

Count $\hat{y} \neq y$

$J_{test}(\vec{w}, b)$ **is the fraction of the test set that has been misclassified.**

$J_{train}(\vec{w}, b)$ **is the fraction of the train set that has been misclassified.**

Model Selection



Once parameters $\vec{w}, b$ are fit to the training set, the training error $J_{train}(\vec{w}, b)$ is likely lower than the actual generalization error.

$J_{test}(\vec{w}, b)$ is better estimate of how well the model will generalize to new data compared to $J_{train}(\vec{w}, b)$.

$$x = x_1$$

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

Model Selection

$d = 1$ **1.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + b$ $\longrightarrow$ $w^{<1>}, \mathrm{b}^{<1>}$ $J_{test}(w^{<1>}, \mathrm{b}^{<1>})$

$d = 2$ **2.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + w_2 x^2 + b$ $\longrightarrow$ $w^{<2>}, \mathrm{b}^{<2>}$ $J_{test}(w^{<2>}, \mathrm{b}^{<2>})$

$d = 3$ **3.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + w_2 x^2 + w_3 x^3 + b$ $\longrightarrow$ $w^{<3>}, \mathrm{b}^{<3>}$ $J_{test}(w^{<3>}, \mathrm{b}^{<3>})$

$\vdots$

$d = 10$ **10.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + w_2 x^2 + \cdots + w_{10} x^{10} + b$ $\longrightarrow$ $w^{<10>}, \mathrm{b}^{<10>}$ $J_{test}(w^{<10>}, \mathrm{b}^{<10>})$

Choose $w_1 x + \cdots + w_5 x^5 + b$ $d = 5$ $J_{test}(w^{<5>}, \mathrm{b}^{<5>})$

How well does the model perform ?

- **Report test set error** $J_{test}(w^{<5>}, \mathrm{b}^{<5>})$ **?**
- **The problem:** $J_{test}(w^{<5>}, \mathrm{b}^{<5>})$ **is too optimistic.**
- **Since extra parameter** d **(degree of polynomial) was chosen**
- **That is, test set is spoiled**
- **Need the different way !**

Training/Cross Validation/Test

Dataset:

	size	price
60 %	2200	390
	1700	325
	2450	365
	1400	220
	3100	536
	1930	300
20 %	1530	320
	1430	188
20 %	1375	210
	1485	235

Training set

m_{train} = **no. training examples**

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$(x^{(m_{train})}, y^{(m_{train})})$$

Cross validation

m_{CV} = **no. cross validation examples**

$$(x^{(1)}, y^{(1)})$$
$$\vdots$$
$$(x^{(m_{CV})}, y^{(m_{CV})})$$

Test set

m_{test} = **no. test examples**

$$(x^{(1)}, y^{(1)})$$
$$\vdots$$
$$(x^{(m_{test})}, y^{(m_{test})})$$

Training/Cross Validation/Test

Training error:

$$J_{train}(\vec{w}, b) = \frac{1}{2m_{train}} \sum_{i=1}^{m_{train}} \left(f_{\vec{w},b}\left(\vec{x}_{train}{}^{(i)} \right) - y_{train}^{(i)} \right)^2$$

Cross validation error:

$$J_{CV}(\vec{w}, b) = \frac{1}{2m_{CV}} \sum_{i=1}^{m_{CV}} \left(f_{\vec{w},b}\left(\vec{x}_{CV}{}^{(i)} \right) - y_{CV}^{(i)} \right)^2$$

Test error:

$$J_{test}(\vec{w}, b) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(f_{\vec{w},b}\left(\vec{x}_{test}{}^{(i)} \right) - y_{test}^{(i)} \right)^2$$

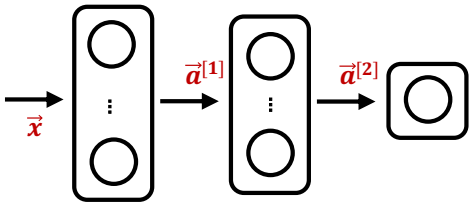
Model Selection

$d = 1$	**1.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + b$	$w^{<1>}, \mathrm{b}^{<1>}$	$J_{CV}(w^{<1>}, \mathrm{b}^{<1>})$
$d = 2$	**2.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + w_2 x^2 + b$	$w^{<2>}, \mathrm{b}^{<2>}$	$J_{CV}(w^{<2>}, \mathrm{b}^{<2>})$
$d = 3$	**3.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + w_2 x^2 + w_3 x^3 + b$	$w^{<3>}, \mathrm{b}^{<3>}$	$J_{CV}(w^{<3>}, \mathrm{b}^{<3>})$
	⋮		
$d = 10$	**10.** $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + w_2 x^2 + \cdots + w_{10} x^{10} + b$	$w^{<10>}, \mathrm{b}^{<10>}$	$J_{CV}(w^{<10>}, \mathrm{b}^{<10>})$

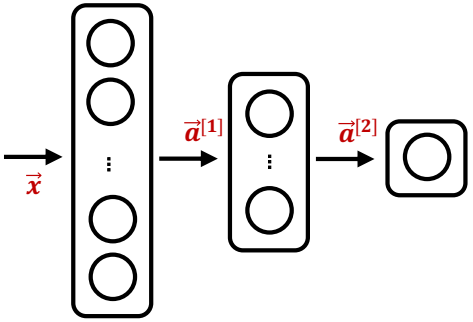
Pick $f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x + \cdots + w_4 x^4 + b$ $J_{CV}(w^{<4>}, \mathrm{b}^{<4>})$

Estimate generalization error using test set $J_{test}(w^{<4>}, \mathrm{b}^{<4>})$

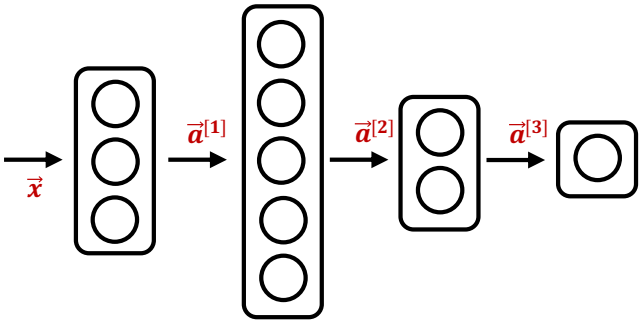
Model Selection

1. 

$w^{<1>}, b^{<1>}$ $J_{CV}(w^{<1>}, b^{<1>})$

2. 

$w^{<2>}, b^{<2>}$ $J_{CV}(w^{<2>}, b^{<2>})$

3. 

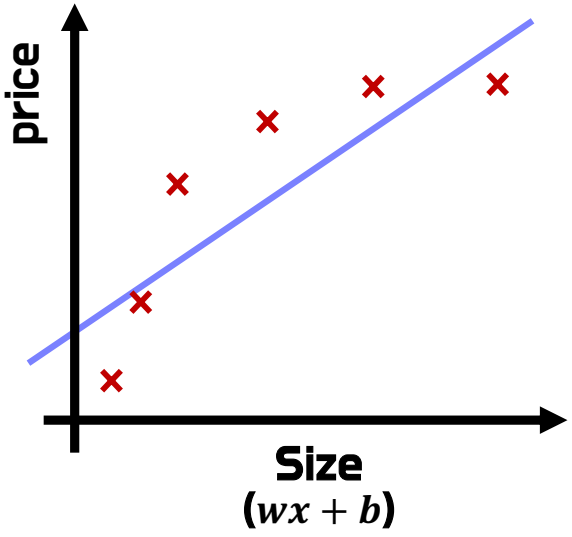
$w^{<3>}, b^{<3>}$ $J_{CV}(w^{<3>}, b^{<3>})$

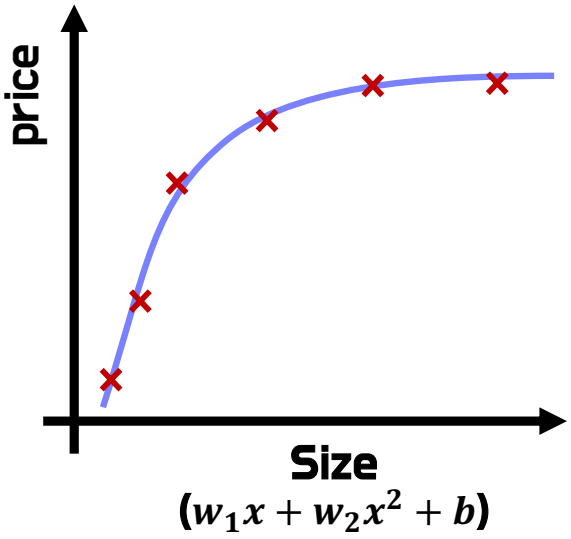
Pick $w^{<2>}, b^{<2>}$ $J_{CV}(w^{<2>}, b^{<2>})$

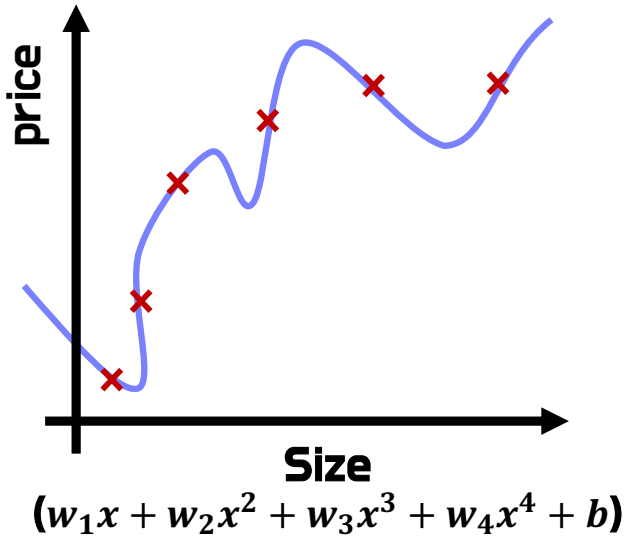
Estimate generalization error using the test set $J_{test}(w^{<2>}, b^{<2>})$

Bias and variance

Bias/Variance







High bias

(Underfit)

J_{train} **is high**

J_{CV} **is high**

$d = 1$

Just right

J_{train} **is low**

J_{CV} **is low**

$d = 2$

High variance

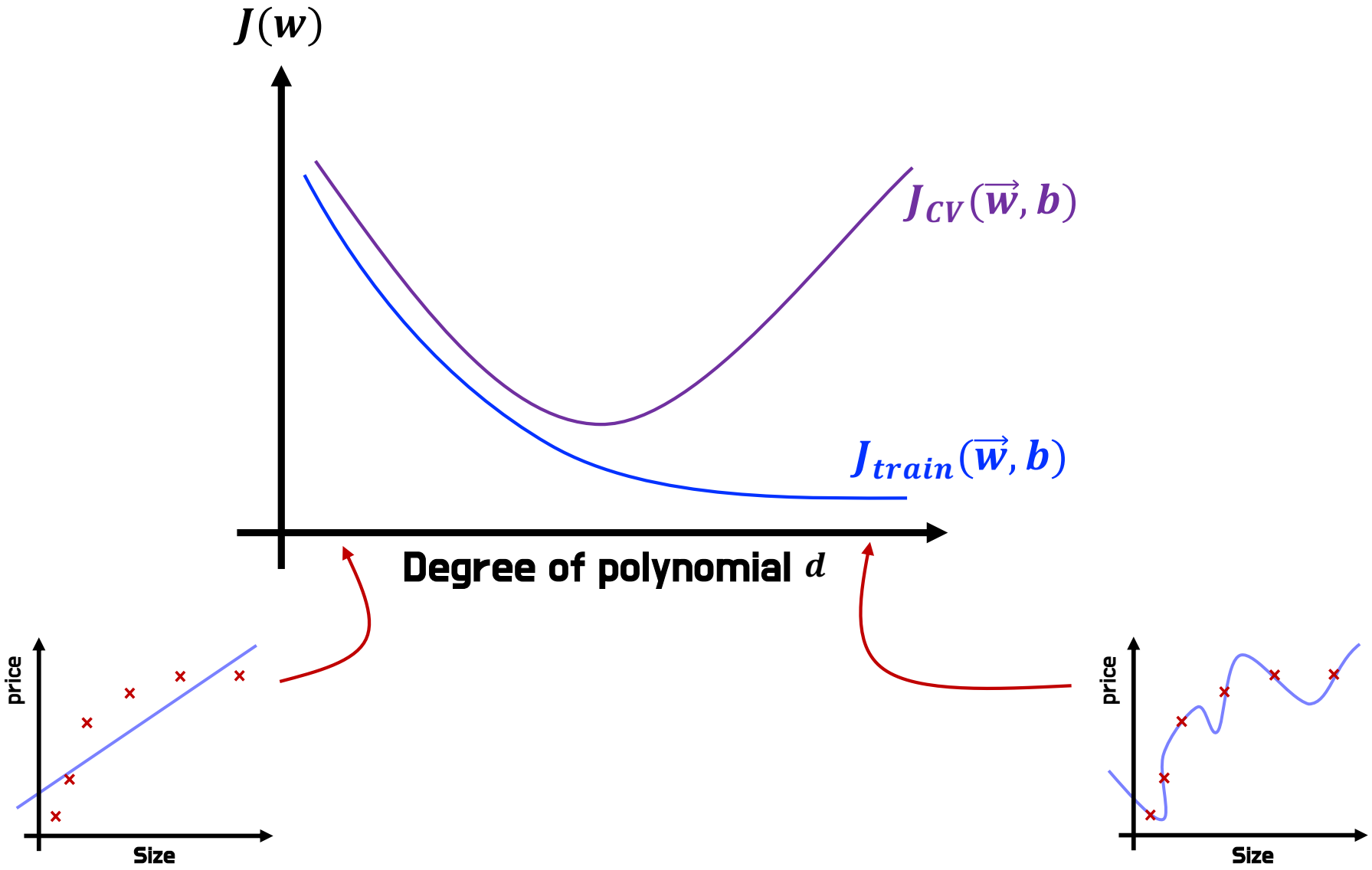
(Overfit)

J_{train} **is low**

J_{CV} **is high**

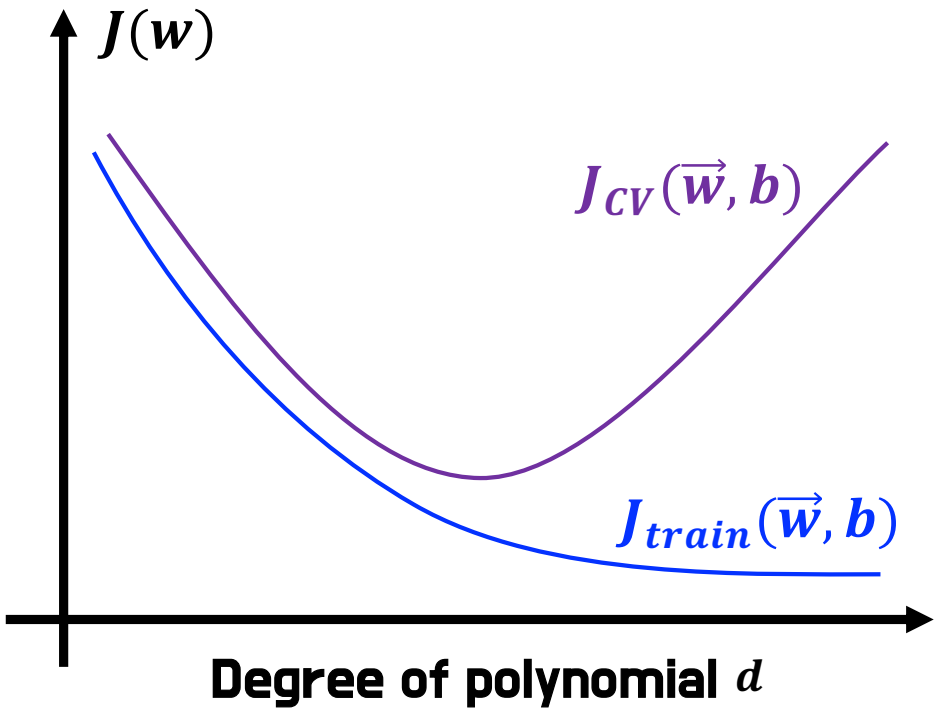
$d = 4$

Understanding Bias and Variance



Diagnosing Bias and Variance

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



High bias (underfit)

J_{train} **is high**

$(J_{train} \sim J_{CV})$

High variance (overfit)

$J_{CV} \gg J_{train}$

$(J_{train}$ **may be low)**

High bias and high variance

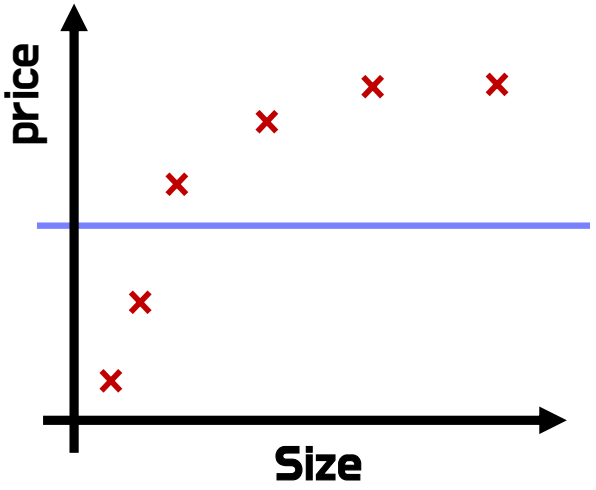
J_{train} **will be high**

$J_{CV} \gg J_{train}$

Linear Regression with Regularization

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

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

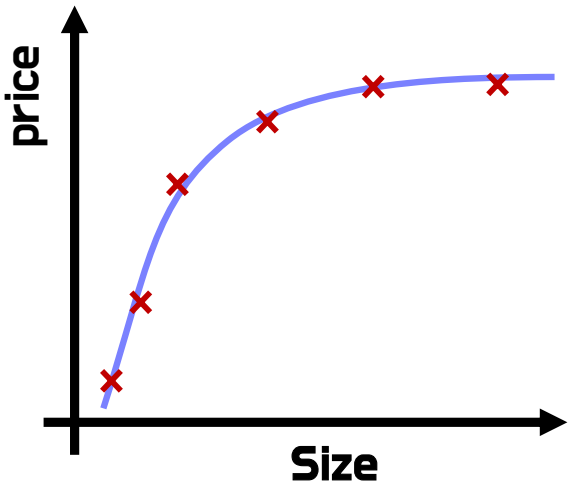


Large λ

High bias (underfit)

J_{train} is high

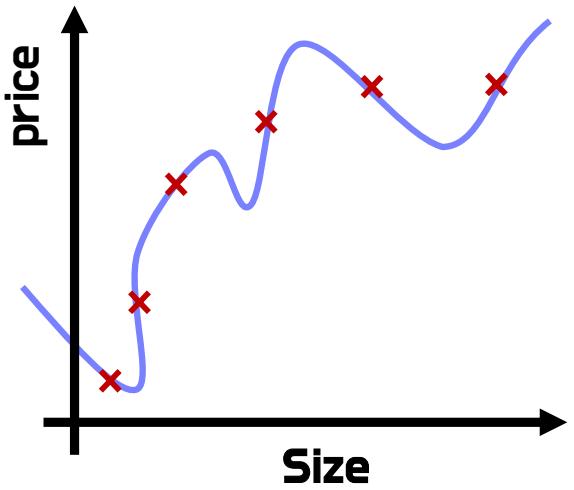
J_{CV} is high



Intermediate λ

J_{train} is low

J_{CV} is low



Small λ

High variance (overfit)

J_{train} is low

J_{CV} is high

Choosing the Regularization Parameter

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

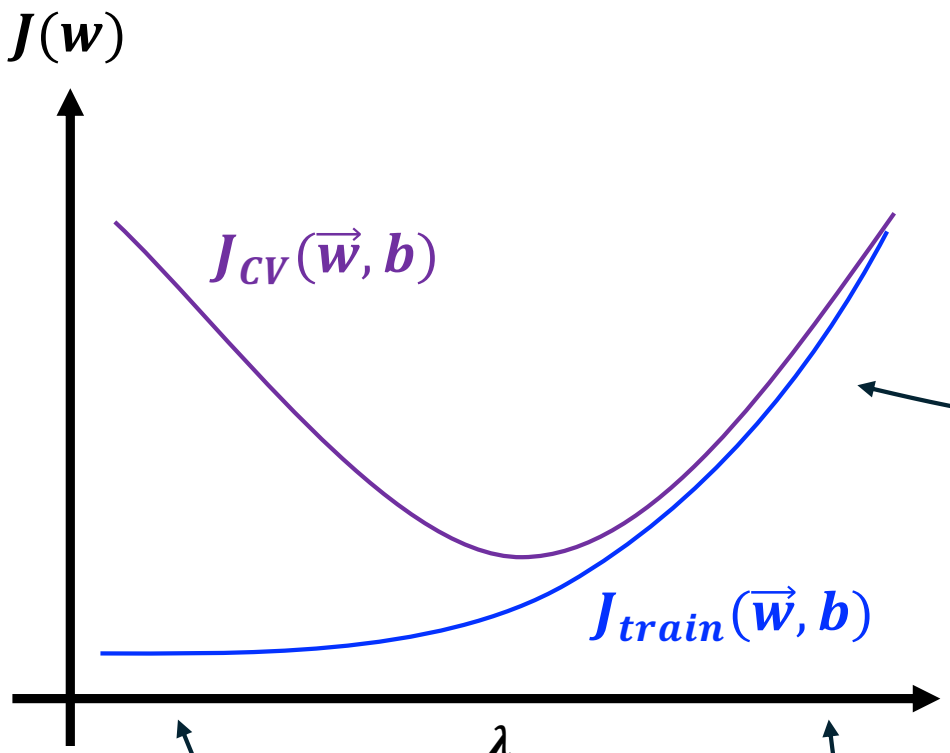
1. Try $\lambda = 0$	$\min_{\vec{w}, b} J(\vec{w}, b)$	$w^{<1>}, b^{<1>}$	$J_{CV}(w^{<1>}, b^{<1>})$
2.Try $\lambda = 0.01$		$w^{<2>}, b^{<2>}$	$J_{CV}(w^{<2>}, b^{<2>})$
3.Try $\lambda = 0.02$		$w^{<3>}, b^{<3>}$	$J_{CV}(w^{<3>}, b^{<3>})$
4.Try $\lambda = 0.04$		$w^{<4>}, b^{<4>}$	$J_{CV}(w^{<4>},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

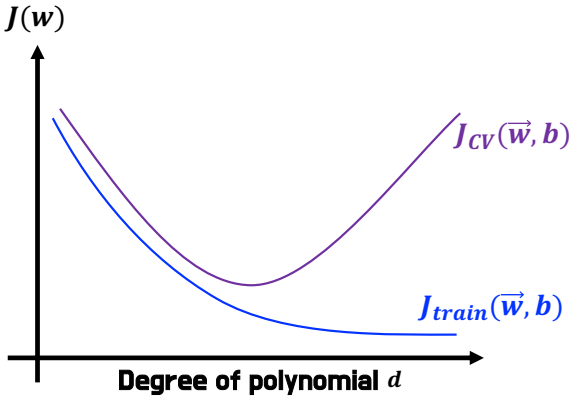
Pick $w^{<5>}, b^{<5>}$ $J_{CV}(w^{<5>}, b^{<5>})$

Report test error: $J_{test}(w^{<5>}, b^{<5>})$

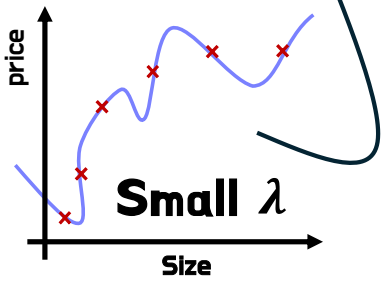
Bias and Variance as Lambda

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

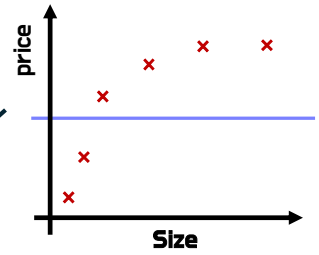




Mirror image



Small λ



Large λ