Regularization

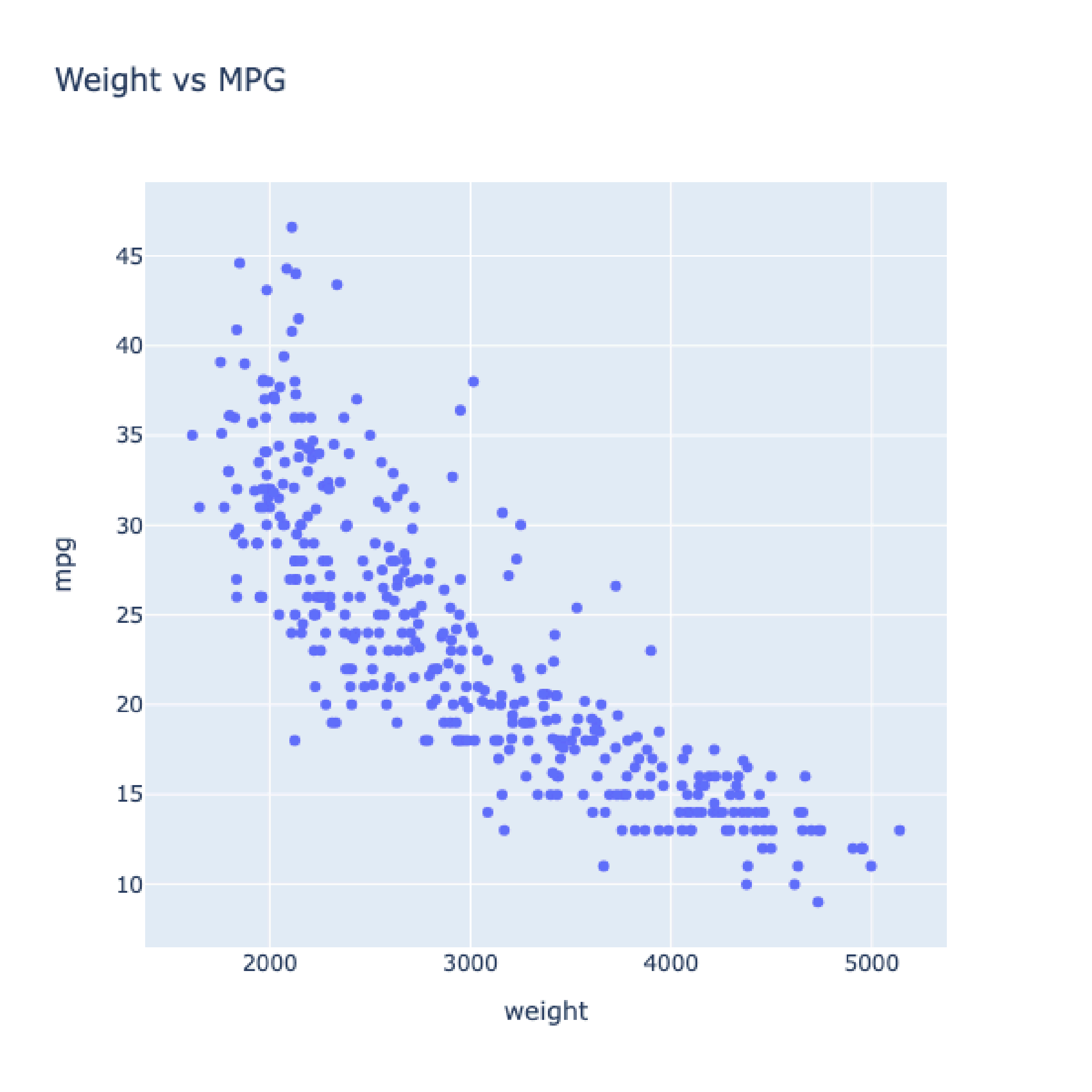
Model complexity: overfitting

Good model:

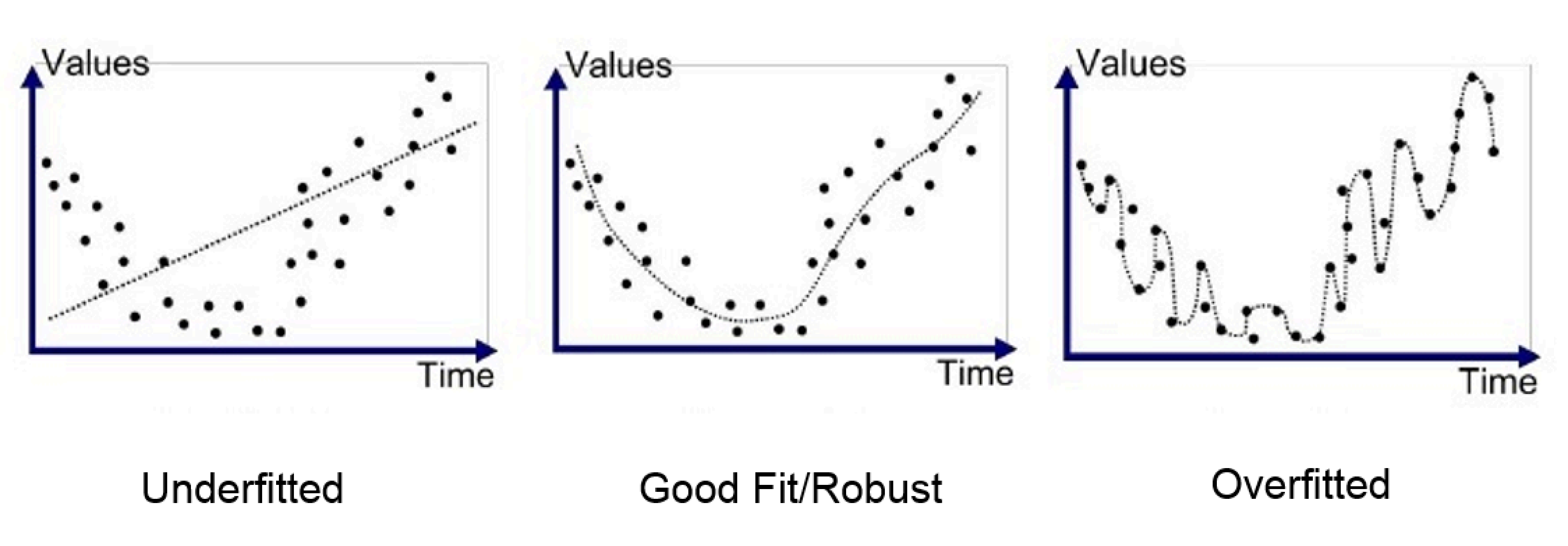
- fits the data well
- generalizes to new data

Overfit model:

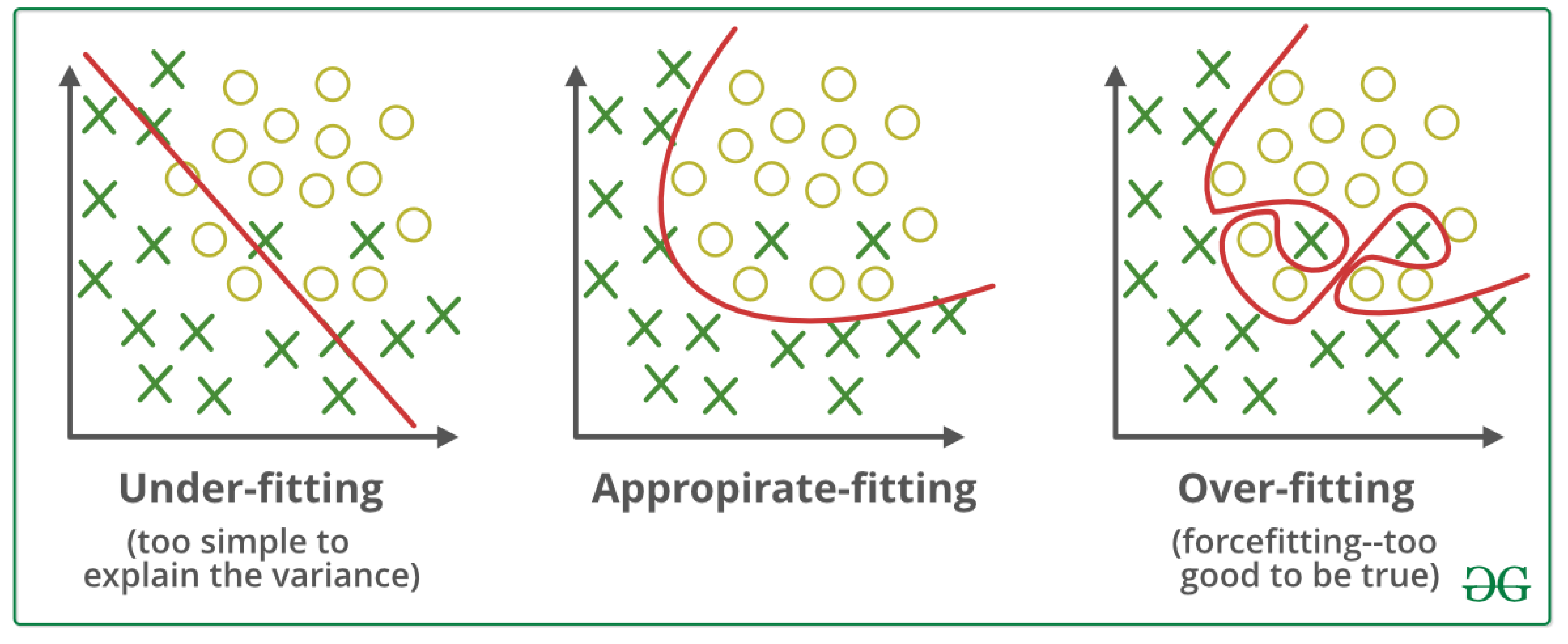
- fits the data **too** well
- fails to generalize to new data

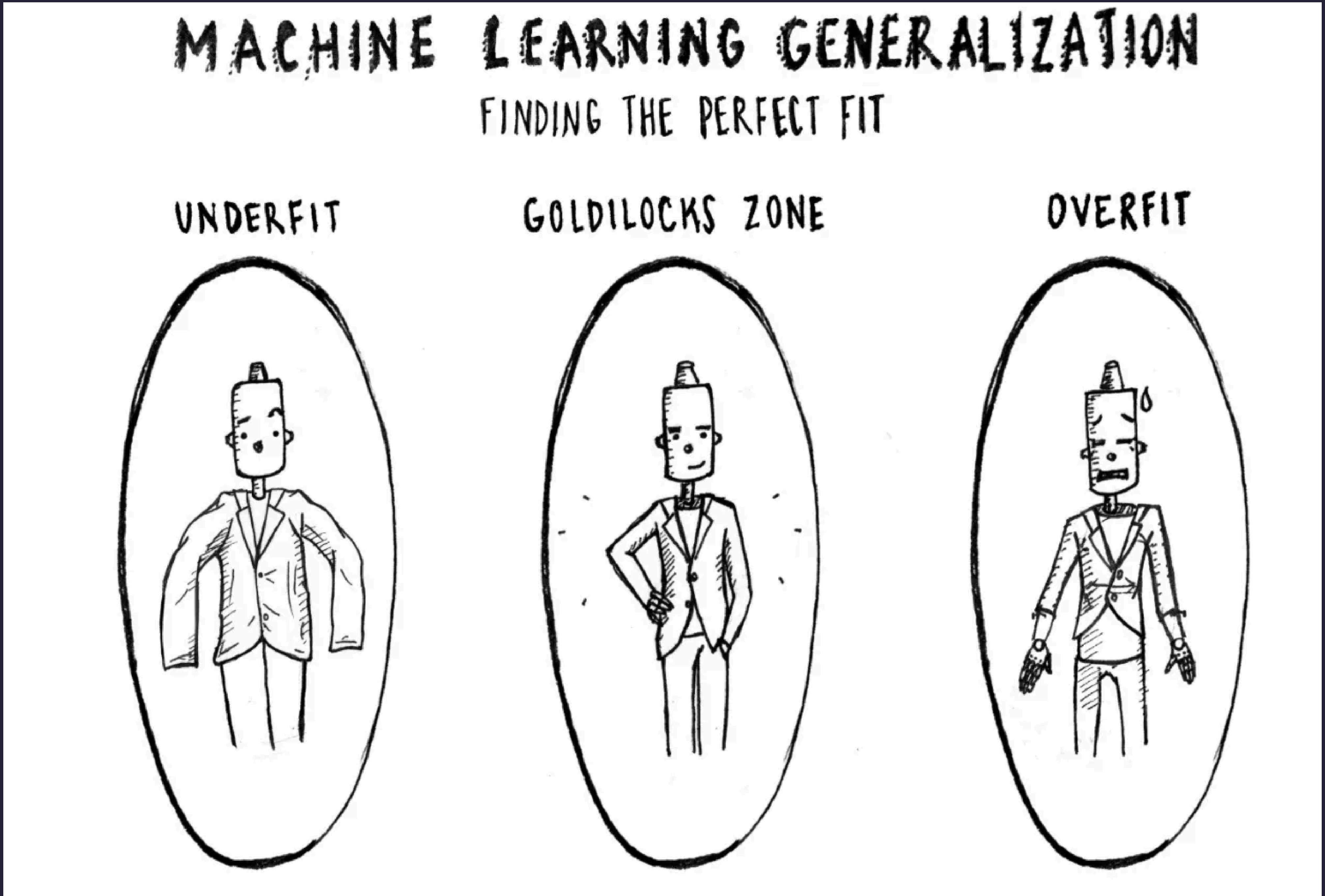


Regression example



Classification example




MACHINE LEARNING GENERALIZATION
FINDING THE PERFECT FIT
UNDERFIT
GOLDILOCKS ZONE
OVERFIT

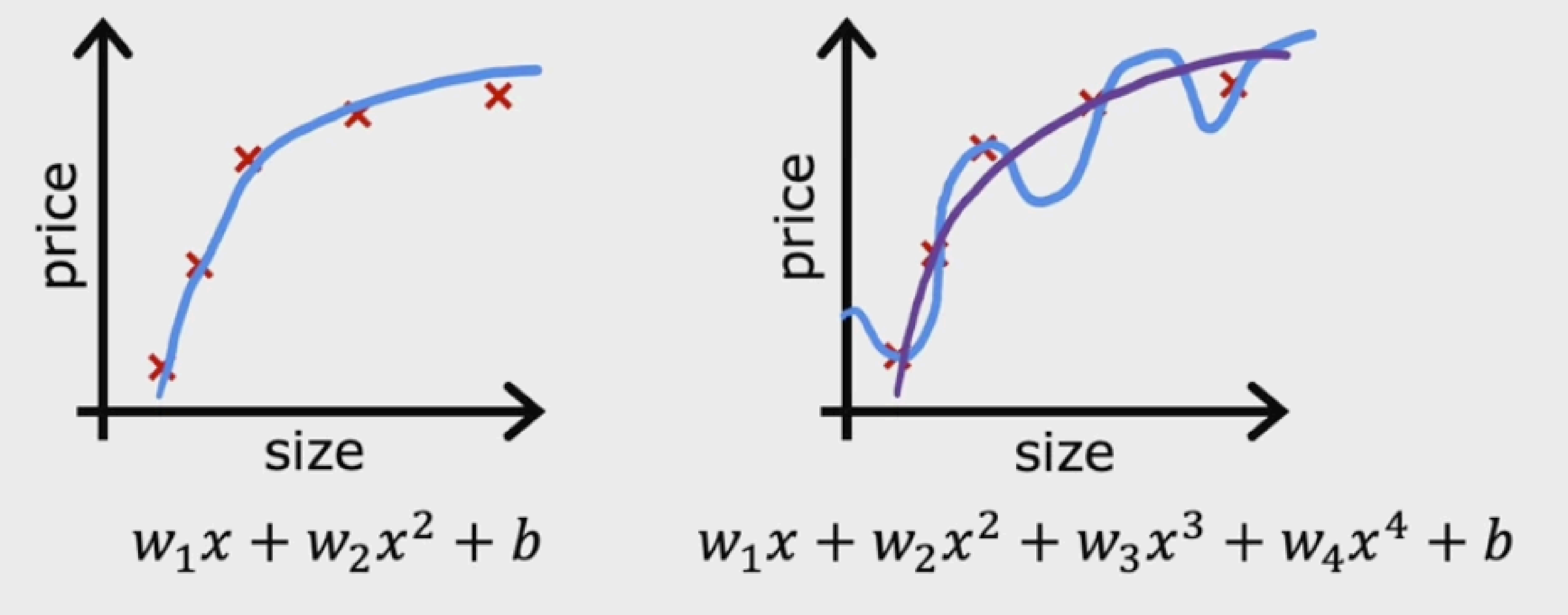
How to prevent overfitting?

Collect more data

Feature selection: restrict the model complexity by choosing fewer features

> Regularization: restrict the model complexity by penalizing large weights

How to make the model simpler?



Regularization

Small weights (≈ 0) to make the model simpler

$$f(x) = 28x + 385x^2 - 39x^3 + 174x^4 + 100$$

$$\Downarrow$$

$$f(x) = 13x - 0.23x^2 - 0.0000012x^3 + 0.0002x^4 + 4$$

To make w_3, w_4 small (≈ 0), modify the cost function:

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} L(a^{(i)}, y^{(i)}) + \underbrace{(1000w_3^2 + 1000w_4^2)}_{\text{penalty for large weights}}$$

Modified cost function with regularization

Minimize both `loss` and `complexity`

$$J(\vec{w}, b) = \underbrace{\frac{1}{m} \sum_{i=1}^{m} L(a^{(i)}, y^{(i)})}_{\text{loss}} + \lambda \underbrace{\sum_{j=1}^{n} w_j^2}_{\text{complexity}}$$

- j: index of the feature ($j = 1, 2, \ldots, n$)
- w_j: weight of the feature j
- **loss**: how well the model fits the data (same as before)
- **complexity**: how complex the model is
- λ **(lambda): regularization parameter**

Regularization parameter

$$J(\vec{w}, b) = \underbrace{\frac{1}{m}\sum_{i=1}^{m} L(a^{(i)}, y^{(i)})}_{\text{loss}} + \lambda \underbrace{\sum_{j=1}^{n} w_j^2}_{\text{complexity}}$$

large λ

- **Complexity** dominates
- Weights close to zero

small λ:

- **Complexity** close to zero $\Rightarrow$ Non-regularized model
- Weights close to non-regularized values

Regularized linear regression

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

Partial derivative of the cost function with respect to w_j

$$\frac{\partial J(\vec{w}, b)}{\partial w_j} = \frac{2}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)}) x_j^{(i)} + 2\lambda w_j$$

Partial derivative of the cost function with respect to b

$$\frac{\partial J(\vec{w}, b)}{\partial b} = \frac{2}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)})$$

Regularization: "Shrinking" the weights

Gradient descent update rule for w_j

$$
\begin{aligned}
w_j &= w_j - \alpha \frac{\partial J(\vec{w}, b)}{\partial w_j} \\
&= w_j - \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)}) x_j^{(i)} + 2\lambda w_j \right) \\
&= w_j - 2\alpha\lambda w_j - \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)}) x_j^{(i)} \right) \\
&= \underbrace{w_j(1 - 2\alpha\lambda)}_{\text{shrink factor}} \underbrace{- \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)}) x_j^{(i)} \right)}_{\text{usual update}}
\end{aligned}
$$

Regularized logistic regression

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(a^{(i)}) + (1 - y^{(i)})\log(1 - a^{(i)})] + \lambda\sum_{j=1}^{n} w_j^2$$

Partial derivative of the cost function with respect to w_j

$$\frac{\partial J(\vec{w}, b)}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})x_j^{(i)} + 2\lambda w_j$$

Partial derivative of the cost function with respect to b

$$\frac{\partial J(\vec{w}, b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})$$

Regularized softmax regression

$$J(\vec{w}, \vec{b}) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \log(a_k^{(i)}) + \lambda \sum_{j=1}^{n} \sum_{k=1}^{K} w_{j,k}^2$$

Partial derivative of the cost function with respect to $w_{j,k}$

$$\frac{\partial J(\vec{w}, \vec{b})}{\partial w_{j,k}} = \frac{1}{m} \sum_{i=1}^{m} (a_k^{(i)} - y_k^{(i)}) x_j^{(i)} + 2\lambda w_{j,k}$$

Partial derivative of the cost function with respect to b_k

$$\frac{\partial J(\vec{w}, \vec{b})}{\partial b_k} = \frac{1}{m} \sum_{i=1}^{m} (a_k^{(i)} - y_k^{(i)})$$

Ridge and Lasso regression

Ridge regression:

- **L2** regularization term: w_j^2
- shrink weights (close but not equal to zero)

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} L(y^{(i)}, a^{(i)}) + \lambda \sum_{j=1}^{n} w_j^2$$

Lasso regression:

- **L1** regularization term: $|w_j|$
- sparse weights (set some weights to zero)

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L(y^{(i)}, a^{(i)}) + \lambda \sum_{j=1}^{n} |w_j|$$

💻 Regularization