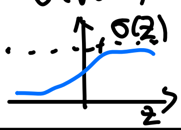
Review:

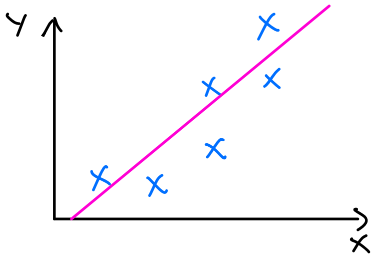
linear $\underbrace{\text{supervised learning}}$

learning from data (X, Y)

X: input Y: correct answer

	linear regression	logistic regression	softmax regression
Task	regression $Y \in \mathbb{R}$	binary classification $Y \in \{-1, 1\}$	multi-class classification $Y \in \{1, 2, \dots, C\}$
Parameters	$w \in \mathbb{R}^d$ $x \in \mathbb{R}^d$	$w \in \mathbb{R}^d$	$w^{(1)}, \dots, w^{(C)} \in \mathbb{R}^d$ $C \cdot d$ params
Probabilistic story	$Y \sim N(w^T x, \sigma^2)$ mean: $w^T x$	$P(Y=1 \mid X=x) = \sigma(w^T x)$ 	$P(Y=j) = \dfrac{\exp(w^{(j)T}x)}{\sum_{k=1}^{C} \exp(w^{(k)T}x)}$ normalizes to prob. dist.
How to get loss functions	MLE, maximize prob. of data = negative log likelihood $\prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; w)$ w.r.t. $w \Leftrightarrow$ minimize NLL		
How to minimize losses	gradient descent or normal eqns	gradient descent (1st-order) or Newton-Raphson method (2nd-order)	

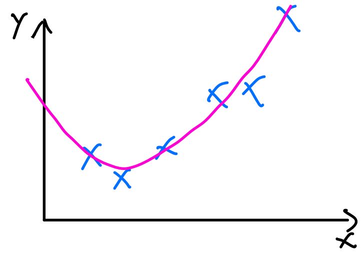
Overfitting



features = $[1, x]$

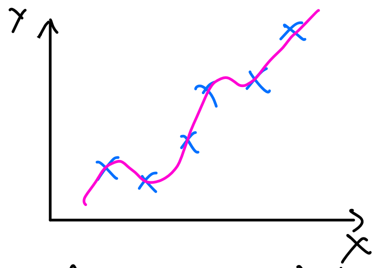
too simple

"underfitting"



features = $[1, x, x^2]$

☆ best fit



features = $[1, x, x^2, x^3, x^4]$

sensitive to fluctuations in training

"overfitting"

Data Splits

Always have 3 disjoint datasets

Training

Use it to choose parameters

Validation

Choose hyperparameter

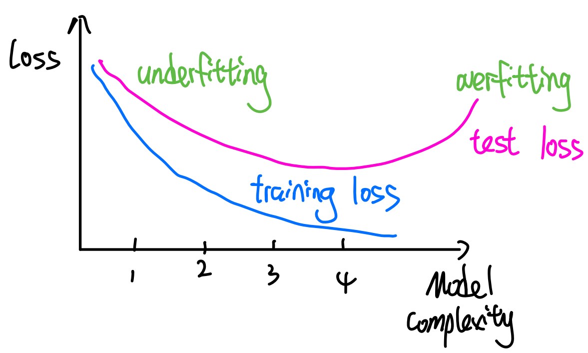
any setting of learning algorithm

- learning rate
- features
- when to stop training

→ parameters chosen by the learning algorithm

Test

Evaluate how well our model performs



Rule: choose hyperparameters to minimise validation loss,
only evaluate on test set at very end.

degree	development loss (validation)	test loss
1	100	100
2	51	50
3	50	50
4	49	50
5	75	75

① train 5 models
② evaluate each on dev
③ pick the best based on ②
④ evaluate that model on test set

Regularization (to reduce overfitting Prefer "simpler" functions)

L_2 Regularization

$\sum_{j=1}^{d} w_j^2 = \|w\|^2$

encourage square of L_2 norm to be small

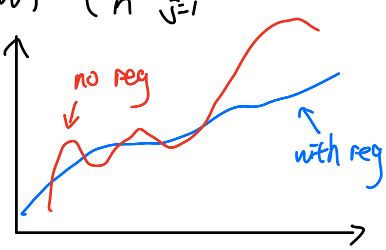
e.g. linear regression

$$L(w) = \left(\frac{1}{n}\sum_{j=1}^{n}(w^T x^{(j)} - y^{(j)})^2\right) + \lambda\|w\|^2$$

some constant hyperparameter

$\lambda = 0 \Rightarrow$ no reg

λ large $\Rightarrow$ strong reg



How does this change the gradient

gradient of $\lambda\|w\|^2 = 2\lambda\|w\|$

doing GD, subtract $\eta \cdot 2\lambda w$

learning rate

L_1 Regularization

add $\lambda\|w\|$ to objective $\qquad \|w\| = \sum_{j=1}^{d}|w_j|$

gradient of L_1: $\frac{\partial}{\partial w_j}\lambda\|w\| = \lambda\,\mathrm{sgn}(w_j)$

full gradient is $\lambda\begin{bmatrix}\mathrm{sgn}(w_1)\\ \vdots \\ \mathrm{sgn}(w_d)\end{bmatrix}$ vs $2\lambda w$ ← if $w \approx 0$, take very small step

constant size step towards 0

L_1 reg has a sparsifying effect

(leads to $\underbrace{\text{sparse}}_{\text{many entries} = 0}$ w)

L_2 reg avoids very big entries of w