## Week 6 | Advice for applying ML

- Your dataset:
  - 60% training set
  - 20% cross validation set (CV)
  - 20% test set

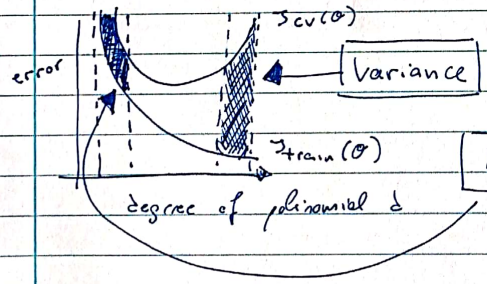
- Train / validation / test error

$$J_\alpha = \frac{1}{2m_\alpha} \sum_{i=1}^{m} \left( h_\theta(x_\alpha^{(i)}) - y_\alpha^{(i)} \right)^2 \; ; \; \alpha \in \{\text{train, cv, test}\}$$

- Diagnosing bias / variance

We need to find a compromise between them

If your algorithm is performing badly, is it a bias or variance problem?




Variance: $J_{train} \ll$, $J_{cv} \gg J_{train}$

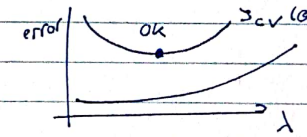
bias: $J_{train} \gg$, $J_{cv} \approx J_{train}$

Using $\lambda$: $h_\theta(x) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{m} \theta_j^2$

If $\lambda \gg \Rightarrow h_\theta(x) \approx \theta_0 \Rightarrow$ underfit

If $\lambda \gg \Rightarrow h_\theta(x) = [\dots] \Rightarrow$ overfit

How choosing the right $\lambda$? Iterating little by little through $\lambda$s and minimizing $J(\theta)$ for each one. Find the minimum



High variance
$J_{cv} >> J_{train}$

- Get more training examples
- Try smaller set of features
- Try increasing $\lambda$

High bias
$J_{cv} \approx J_{train}$
$J_{cv}, J_{train} >>$

- Try additional features
- Try adding polynomial features
- Try decreasing $\lambda$

# Precission / Recall

- Precission = positive predictive value

  : Fraction of relevant instances among retrieved instances

- Recall = sensitivity

  Fraction of the total amount of relevant instances that were actually retrieved

Ex A ML algorithm identifies 8 dogs in a picture where there are 12 dogs and some cats. Of the 8 identified dogs, 5 are dogs (true positives) and 3 cats (false positives).

Precission = 5/8 — from all dogs identified, just 5 were actually dogs

Recall = 5/12 — From all the dogs, just 5 were actually identified

- F1 Score $\equiv \frac{P \cdot R}{P + R}$ — It gives greater weights to lower values of P and R

- Classification accuracy

$$\text{Accuracy} = \frac{\text{n° of correct predictions}}{\text{total of predictions}} = \frac{TP + FN}{\text{total of samples}}$$

If 98% of ~~subtitles~~ samples are type A, our model can easily get 98% training accuracy, so it's important to have a well balanced system

- Confusion Matrix

| n = 165 | Pred NO | Pred YES |
|---|---|---|
| Actual NO | 50 | 10 |
| Actual YES | 5 | 100 |

TP → pred YES output YES
TN → pred NO output NO
FP → pred YES output NO
FN → pred NO output YES