Probability and Estimation

- Probability and Random Variables
- Conditional Probability and Expectation
- Frequentist Viewpoint and the ML Estimate
- Bayesian Viewpoint and the MMSE and MAP Estimates
- The Bias Variance Tradeoff

You need to be an Bell X-1 Test Pilot*

You don't just want to be a Deep Learning commuter…

You want to be a Deep Learning…

Test Pilot!

Probability

- A probability space is a triple $(\Omega, \mathcal{B}, P)$ s.t.
  - Ω be a sample space
  - $\mathcal{B}$ is the set of "well behaved" subsets of Ω
    - Note: If Ω is discrete, then $\mathcal{B} = 2^{\Omega}$, i.e., the power set of Ω
  - $A \in \mathcal{B}$ is an event
  - $P(A)$ is the probability of the event B
- Axioms of probability

$$P(\Omega) = 1$$

$$\forall A \in \mathcal{B}, P(\mathrm{A}) \geq 0$$

$$\text{If } \forall i \neq j \; A_i \cap A_j = \emptyset \text{ , then } P(\bigcup_i A_i) = \sum_i P(A_i)$$

Random Variables

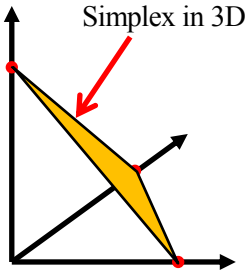
- A random variable is a "well behaved" function
  - $X: \Omega \rightarrow \Re$
  - Can be thought of as the outcome of an experiment
- Probability that $X = 2$ is given by
  - Event $A = \{\omega \in \Omega: X(\omega) = 2\}$
  - Then $P\{X = 2\} = P(A) = P(\{\omega \in \Omega: X(\omega) = 2\})$

Discrete Probability Density

- A discrete random variable, X, with density

$$P\{X = i\} = p_i$$

for $i = 0, \cdots, M-1$, with $p_i \geq 0$ and $1 = \sum_{i=0}^{M-1} p_i$.
Notice that $p \in \mathcal{S}^M \Leftarrow$ is a simplex set.



- Then we have

$$P\{X \in A\} = \sum_{i \in A} p_i$$

Expectation, Mean, and Variance

- Let $P\{X = i\} = p_{\mathrm{i}}$,
  - Expected value of X

$$\mu = E[X] = \sum_{n=0}^{M-1} n\, p_n$$

  - Expected value of $f(X)$

$$E[f(X)] = \sum_{n=0}^{M-1} f(n)\, p_n$$

  - Variance X

$$\sigma^2 = E[(X-\mu)^2] = \sum_{n=0}^{M-1} (n-\mu)^2 p_n$$

Marginal and Conditional Probability

- Consider the two random variables, X and Y

$$P\{X = i, Y = j\} = p(i,j)$$

  - Marginal densities

$$P\{X = i\} = \sum_{j=0}^{M-1} P\{X = i, Y = j\} = \sum_{j=0}^{M-1} p(i,j) = p_x(i)$$

$$P\{Y = j\} = \sum_{i=0}^{M-1} P\{X = i, Y = j\} = \sum_{i=0}^{M-1} p(i,j) = p_y(j)$$

  - Conditional densities

$$p_{x|y}(i|j) = P\{X = i|Y = j\} = \frac{P\{X = i, Y = j\}}{P\{Y = j\}} = \frac{p(i,j)}{p_y(j)} = \frac{p(i,j)}{\sum_{i=0}^{M-1} p(i,j)}$$

$$p_{y|x}(j|i) = P\{Y = j|X = i\} = \frac{P\{X = i, Y = j\}}{P\{X = i\}} = \frac{p(i,j)}{p_x(i)} = \frac{p(i,j)}{\sum_{j=0}^{M-1} p(i,j)}$$

Conditional Expectation

- Consider the two random variables, X and Y

$$P\{X = i, Y = j\} = p(i,j)$$

  - Conditional densities

$$p_{x|y}(i|j) = \frac{p(i,j)}{\sum_{i=0}^{M-1} p(i,j)}$$

$$p_{y|x}(j|i) = \frac{p(i,j)}{\sum_{j=0}^{M-1} p(i,j)}$$

  - Conditional expectation

This is a function of Y

$$E[f(X)|Y] = \sum_{n=0}^{M-1} f(n)\, p_{x|y}(n|Y)$$

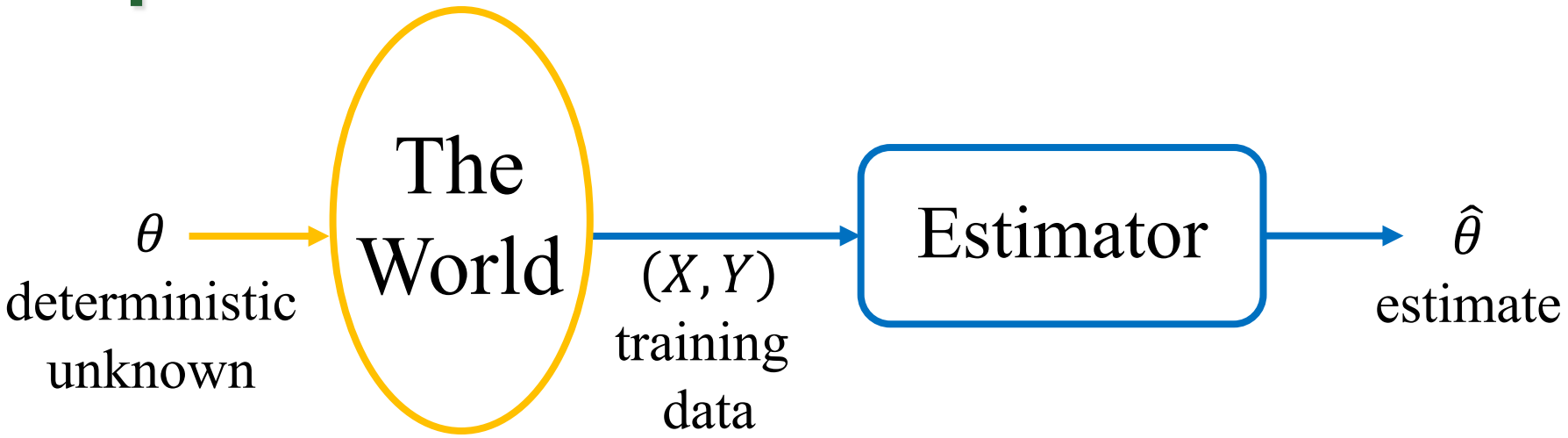
This is a function of X

$$E[f(Y)|X] = \sum_{n=0}^{M-1} f(n)\, p_{y|x}(n|X)$$

Random Variables versus Numbers

- Remember:
  - An function of a random variable is a random variable.
  - The expectation is a number.
  - But the conditional expectation is a random variable.
- Examples:
  - X^2
  - $E[X^2]$
  - $E[X|X]$
  - $E[YX|X]$

Frequentist View of the World



- Model of world

$$(X, Y) \sim p_\theta(x, y) = P_\theta\{X = x, Y = y\}$$

- Estimate of unknown

$$\hat{\theta} = f(X, Y)$$

  - Unbiased estimate

$$E[\hat{\theta}|\theta] = \theta$$

  - Bias and variance are given by

$$\text{Bias} = E[\hat{\theta}|\theta] - \theta$$

$$\text{Variance} = E\left[\|\hat{\theta} - E[\hat{\theta}|\theta]\|^2|\theta\right]$$

Maximum Likelihood Estimate (MLE)

- Model of world

$$Y \sim p_\theta(x, y)$$

- The MLE estimate is defined as

$$\hat{\theta} = \arg\max_\theta \{p_\theta(X, Y)\}$$

$$= \arg\min_\theta \{-\log p_\theta(X, Y)\}$$

usually can be computed as the solution to

$$0 = \nabla_\theta \log p_\theta(Y)\Big|_{\theta=\hat{\theta}}$$

MLE for Regression

- Example: MLE training for regression

Let $X = (X_0, \cdots, X_{K-1})$ and $Y = (Y_0, \cdots, Y_{K-1})$, where

$$X_k = f_\theta(Y_k) + W_k$$

where $W_k \sim N(0, \sigma^2 I)$ and the components of X and Y are i.i.d. Then we have that

$$\log p_\theta(x, y) = \log p_\theta(x|y) + \log p(y)$$

and

$$p_\theta(x_k|y_k) = \frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left\{-\frac{1}{2}\|x_k - f_\theta(y_k)\|^2\right\}$$

So the MLE is given by

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta\{-\log p_\theta(x, y)\} \\
&= \arg\min_\theta\{-\log p_\theta(x|y) - \log p(y)\} \\
&= \arg\min_\theta\left\{\sum_{k=0}^{K-1} -\log p_\theta(x_k|y_k)\right\} \\
&= \arg\min_\theta\left\{\sum_{k=0}^{K-1} \frac{1}{2}\|x_k - f_\theta(y_k)\|^2 + \frac{p}{2}\log\{2\pi\sigma^2\}\right\} \\
&= \arg\min_\theta\left\{\frac{1}{K}\sum_{k=0}^{K-1} \|x_k - f_\theta(y_k)\|^2\right\} \\
&= \arg\min_\theta L_{MSE}(\theta; x, y)
\end{aligned}$$

MSE Loss ⇒ maximum likelihood regression

MLE for Classification

- Example: MLE training for classification

Let $X = (X_0, \cdots, X_{K-1})$ and $Y = (Y_0, \cdots, Y_{K-1})$, where the X_k are 1-hot encoded. Then

$$P\{X_{k,m} = 1\} = f_\theta(m; Y_k)$$

for $m = 0, \cdots, M-1$. Also, we have that

$$\log p_\theta(x, y) = \log p_\theta(x|y) + \log p(y)$$

and notice that

$$-\log p_\theta(x_k|y_k) = -\sum_{m=0}^{M-1} x_{k,m} \log f_\theta(m; y_k)$$

So the MLE is given by

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta\{-\log p_\theta(x, y)\} \\
&= \arg\min_\theta\{-\log p_\theta(x|y) - \log p(y)\} \\
&= \arg\min_\theta\left\{\sum_{k=0}^{K-1} -\log p_\theta(x_k|y_k)\right\} \\
&= \arg\min_\theta\left\{-\sum_{k=0}^{K-1}\sum_{m=0}^{M-1} x_{k,m} \log f_\theta(m; y_k)\right\} \\
&= \arg\min_\theta\left\{\frac{1}{K}\sum_{k=0}^{K-1} \rho_{CE}\big(x_k, f_\theta(m; y_k)\big)\right\} = \arg\min_\theta\{L_{CE}(\theta; x, y)\}
\end{aligned}$$

Cross Entropy Loss ⇒ maximum likelihood classification

MLE for Multinomial Distribution

- Example: Let $Y = (Y_0, \cdots, Y_{N-1})$ be a sequence of independent and identically distributed (i.i.d.) random variables

$$P_\theta\{Y_n = i\} = \theta_i \quad \text{for } \theta \in \mathcal{S}^M$$

$$p_\theta(y) = P_\theta\{Y = y\} = \prod_{i=0}^{N-1} \theta_i^{N_i}$$

where

$$N_i = \sum_{n=0}^{N-1} \delta(Y_n = i)$$

Then

$$-\frac{1}{N}\log p_\theta(y) = -\sum_{n=0}^{N-1} \frac{N_i}{N} \log \theta_i = CrossEntropy\left(\frac{N_i}{N}, \theta_i\right)$$

Then MLE is given by

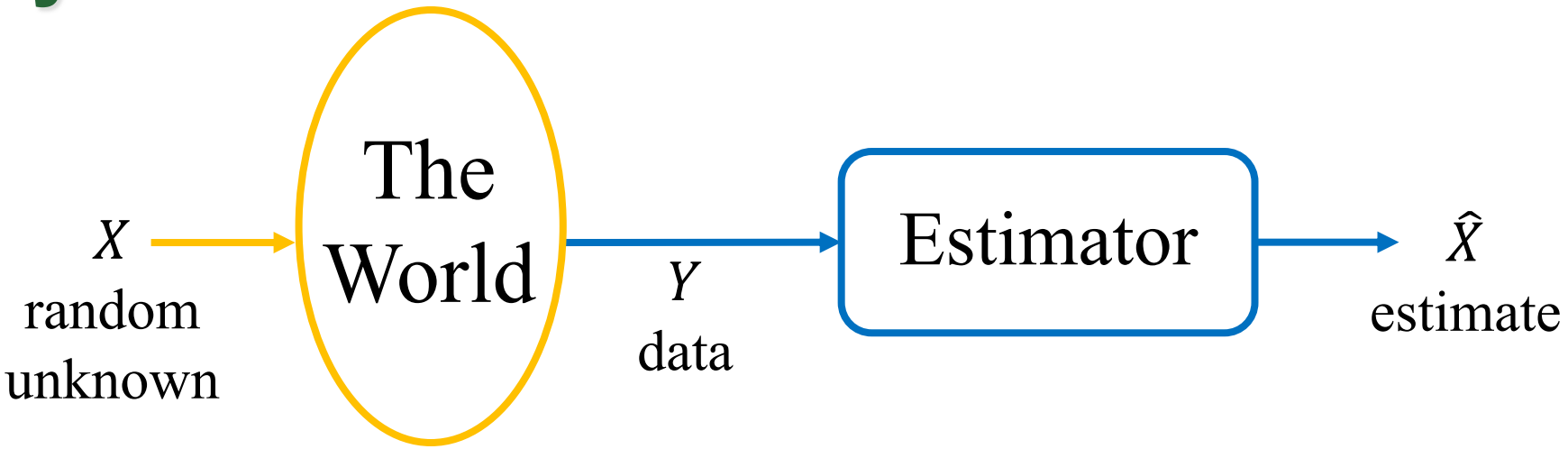
$$\hat{\theta} = \arg\min_\theta \left\{ -\sum_{n=0}^{N-1} \frac{N_i}{N} \log \theta_i \right\} = \frac{N_i}{N}$$

The Big Idea About ML Estimate

- The ML Estimate is…
  - Pure and ideal. It only uses the data.
  - It is (mostly) unbiased.*
  - It is asymptotically efficient, which means it achieves the Cramer Rao bound asymptotically.
  - It's mostly used when there is plenty of data.
- But…
  - It tends to overfit when there is not enough data.

*This isn't really true, but it's mostly true, and the truth is too complicated to explain right now.

Bayesian View of the World



- Model of world

$$Y|X \sim p_{y|x}(y|x) = P\{Y = y|X = x\} \Leftarrow \text{forward model}$$

$$X \sim p_x(x) = P\{X = x\} \Leftarrow \text{prior model}$$

- Estimate of unknown

$$\hat{X} = f(Y)$$

Bayes Law and the Posterior Distribution

- Model of world

$$Y|X \sim p_{y|x}(y|x) = P\{Y = y|X = x\} \Leftarrow \text{forward model}$$

$$X \sim p_x(x) = P\{X = x\} \Leftarrow \text{prior model}$$

- Bayes Law

$$p_{x|y}(x|y) = \frac{p_{y|x}(y|x)\, p_x(x)}{p_y(y)} = \frac{p_{y|x}(y|x)\, p_x(x)}{\sum_{k=0}^{M-1} p_{y|x}(y|k)\, p_x(k)}$$

Posterior Distribution

Common Bayesian Estimators

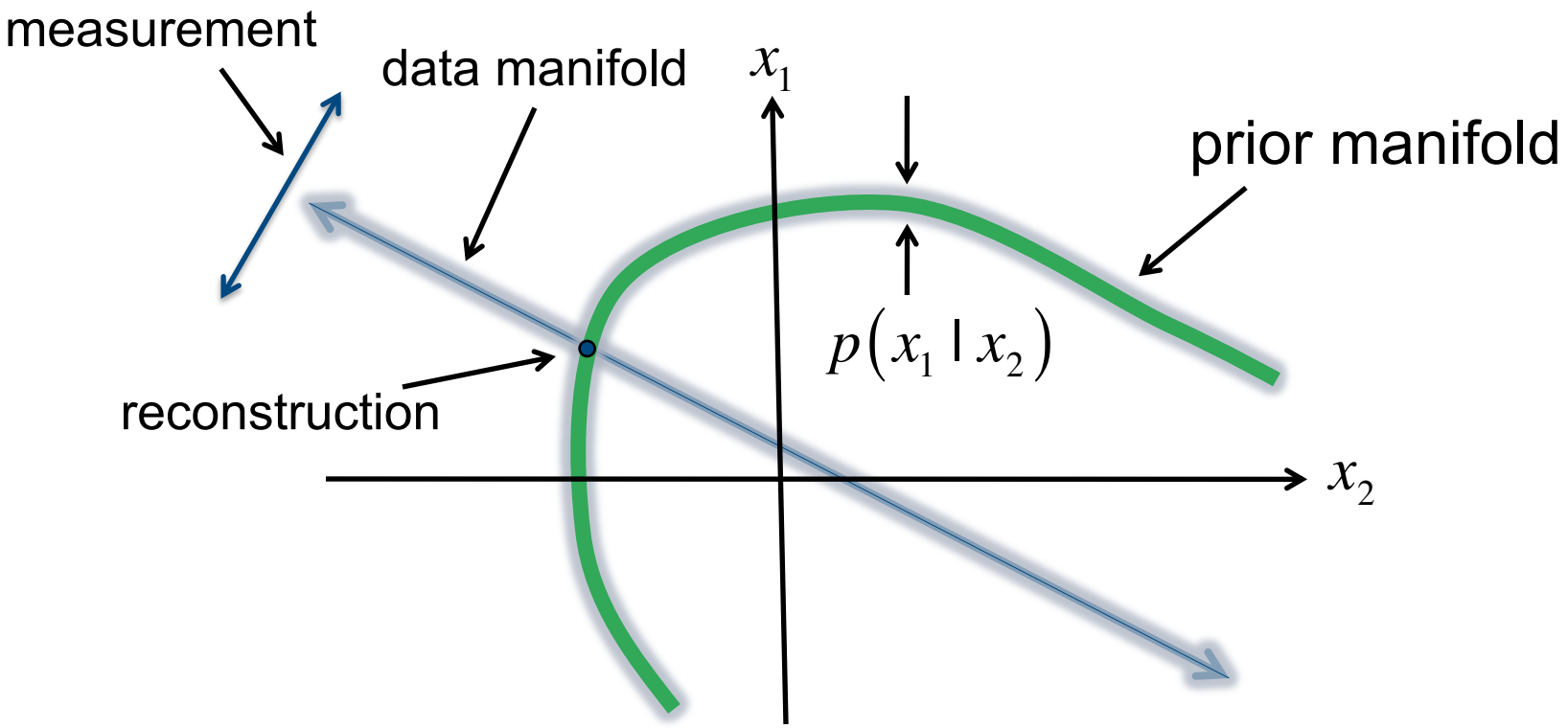
- The Minimum Mean Squared Error (MMSE) estimate

$$\hat{X} = E[X|Y] = \sum_{x=0}^{M-1} x\, p_{x|y}(x|Y)$$

- Maximum A Posteriori (MAP) estimate

$$\begin{aligned}
\hat{x} &= \arg\max_x \{p_{x|y}(x|y)\} \\
&= \arg\min_x \{-\log p_{x|y}(x|y)\} \\
&= \arg\min_x \{-\log p_{y|x}(y|x) - \log p_x(x) + \log p_y(y)\} \\
&= \arg\min_x \{-\log p_{y|x}(y|x) - \log p_x(x)\}
\end{aligned}$$

Manifold Interpretation of Bayesian Estimators



$$\hat{x} = \arg\min_x \{-\log p_{y|x}(y|x) - \log p_x(x)\}$$

- Notice that prior manifold fills the space but is not a linear manifold
  - But it has thickness
  - Dimension of measurement > dimension of manifold
- Prior information can dramatically simplify the estimation problem.

The Bias and Variance

- Definitions*
  - Frequentist: $\text{bias}_\theta = \hat{\theta} - E\left[\hat{\theta}|\theta\right]$

    $\text{Var}_\theta = E\left[\left\|\hat{\theta} - E\left[\hat{\theta}|\theta\right]\right\|^2|\theta\right]$

    $\text{MSE}_\theta = \text{Var}_\theta + (\text{bias}_\theta)^2$

  - Bayesian*: $\text{bias}_x = x - E\left[\hat{X}|X = x\right]$ (?)

    $\overline{\text{bias}^2} = \text{E}\left[\left(X - E\left[\hat{X}|X\right]\right)^2\right]$

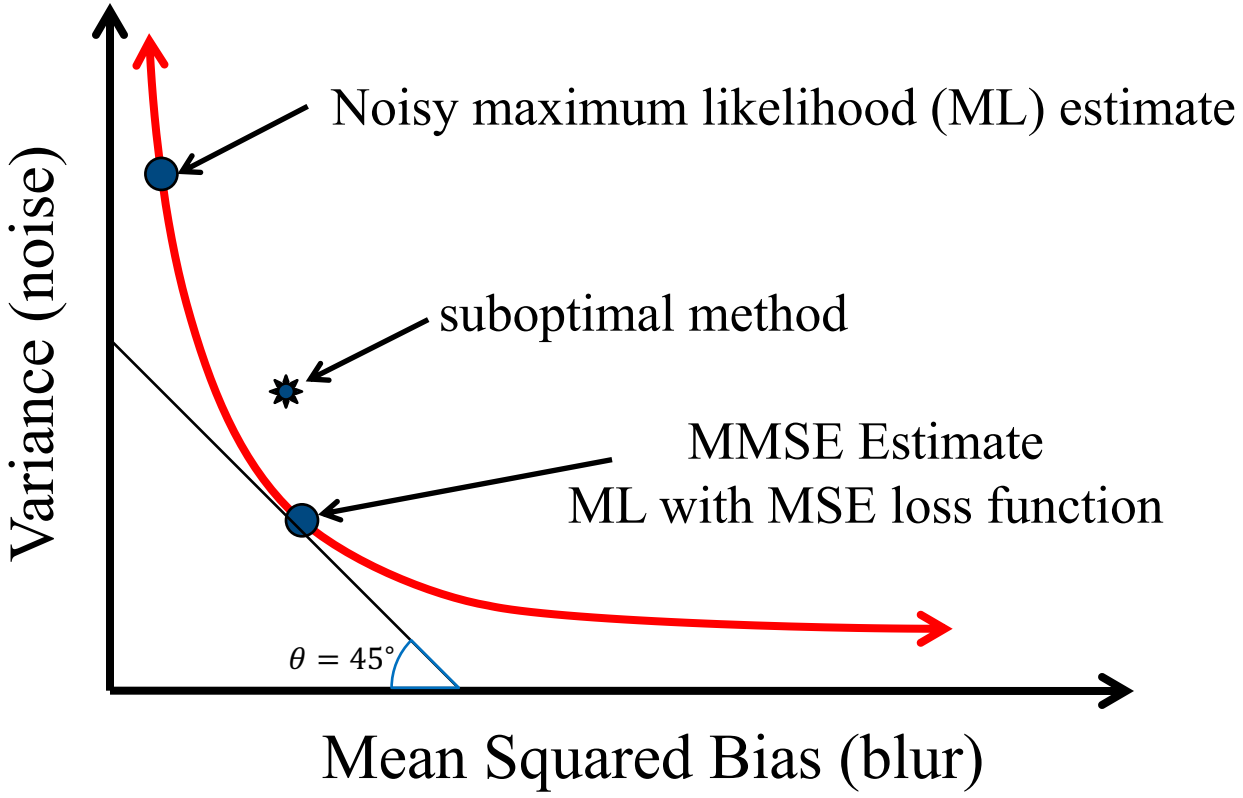
    $\text{Var} = E\left[\left\|\hat{X} - E\left[\hat{X}|X\right]\right\|^2\right]$

    $\text{MSE} = \text{Var} + \overline{\text{bias}^2}$

- Why bias is good
- Why bias is bad

*Bias is really a Frequentist concept.

The Bias Variance Tradeoff



- The tradeoff
  - Bias is usually worse than variance
- Interpretation
  - Variance ⇔ noise
  - Bias ⇔ excessive smoothing, regularization or blur
  - As bias ⇒ 0, then variance ⇒ ∞
  - Not possible to estimate a solution with infinite resolution using a finite amount of data

Bayesian Versus Frequentist Estimation

- Bayesian Estimation
  - Usually used for ML inference
  - Typically high-bias low-variance estimates
  - Most appropriate when prior information is strong
  - Most appropriate when the prior information is strong; or when the amount of data is small and/or the quality of data is poor
- Frequentist Estimation
  - Usually used for parameter estimation
  - Typically low-bias high-variance estimates
  - Most appropriate when prior information is weak; or when the amount of data is large and/or the quality of data is high.

Regularized Maximum Likelihood

- Regularize ML estimate:

$$\hat{X} = \arg\max_{i}\{-\log p_\theta(x, y) + \beta\, S(\theta)\}$$

where $S(\theta)$ is a "regularizing" function, and β is the regularization weight.

Typical choices are

$$S(\theta) = -\log p(\theta) \quad \Longleftarrow \textit{MAP estimate}$$

$$S(\theta) = \|\theta\|^2 \quad \Longleftarrow \textit{Like a Gaussian Prior; Reduces amplitude of weights}$$

$$S(\theta) = \|\theta\|_1 \quad \Longleftarrow \textit{Like a Laplacian Prior; Encourages weights to go to zero}$$

- Modified Loss function

$$\tilde{L}(\theta) = L(\theta) + \beta\, S(\theta)$$

  - Can be interpreted as MAP estimate with $p(\theta) = \frac{1}{z}\exp\left\{-\frac{\beta}{2}S(\theta)\right\}$
  - Introduces bias into the estimate of θ
  - Reduces overfitting
  - Use regularization if training error >> validation error

Training and Generalization

- Overfitting, Underfitting, and Goldilocks Fitting
- Training, Validation, and Testing Data Sets
- Model Order, Model Capacity, Generalization Loss

Training and Generalization

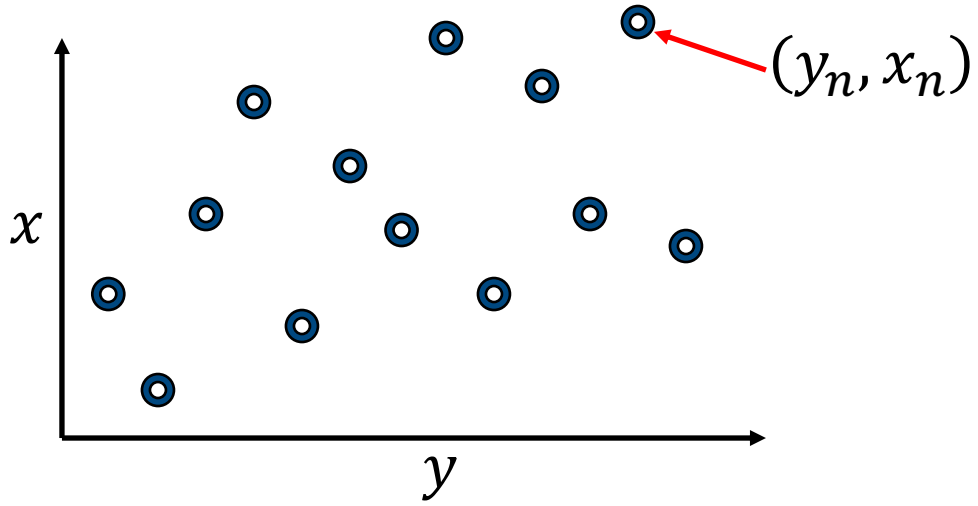
- Goal:
  - Learn the "true relationship" from training data pairs $(x_k, y_k)|_{k=0}^{K-1}$.

$$x = f_\theta(y) + error$$

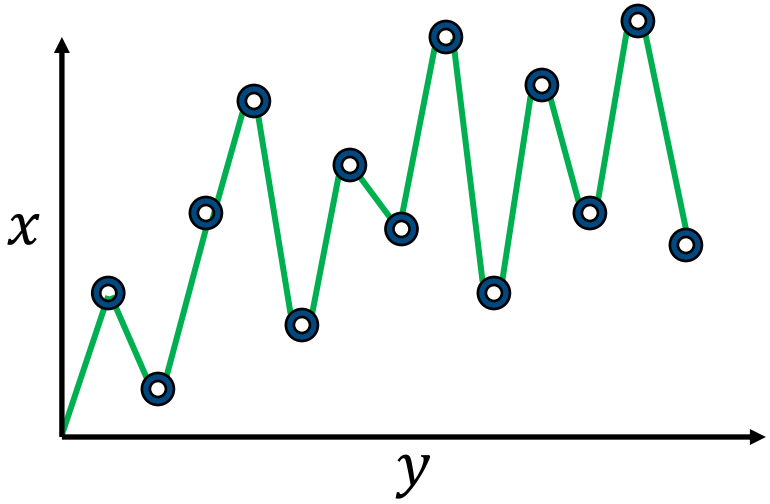
  - What we learn needs to *generalize* beyond the training data.
- Key parameters:
  - P = Model Order = number of parameters = Dimension of $\theta \in \Re^P$
  - $N_x \times K$ = # training points = (Dimension of x)×(# of training pairs)
- Key issues
  - If $P \gg N_x \times K$: Model order is too high and there is a tendency to over fit.
  - If $P << N_x \times K$: Model order is too low, and there is a tendency to under fit

Overfitting

- Training data



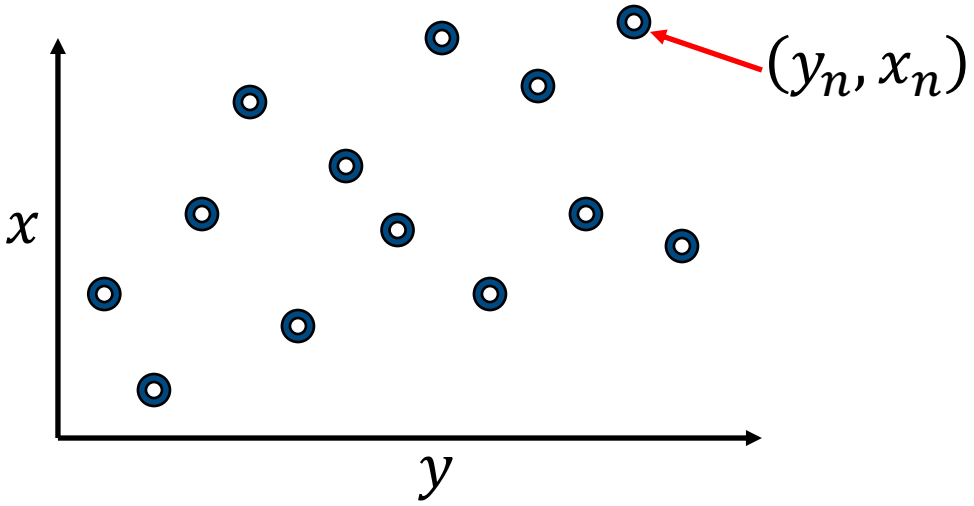
- Overfitting



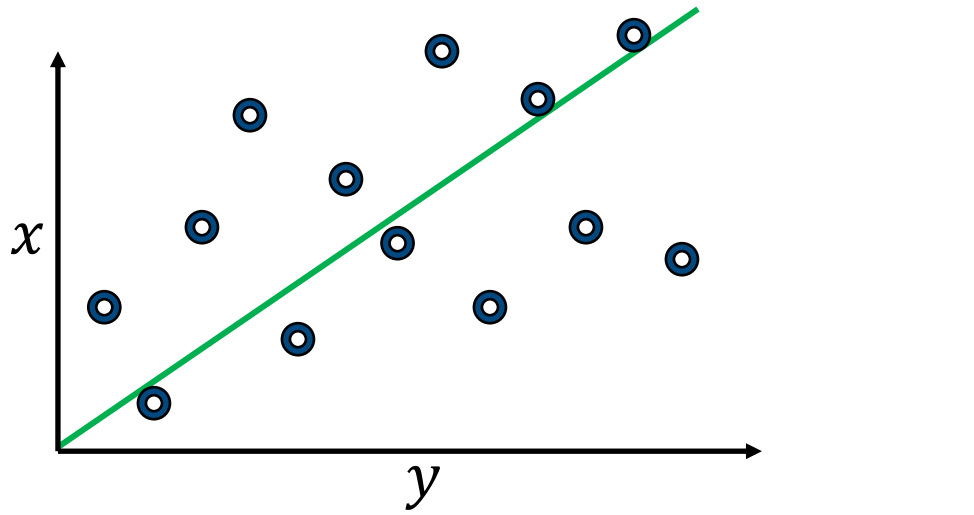
  - Model order too high
  - Doesn't generalize well

Underfitting

- Training data



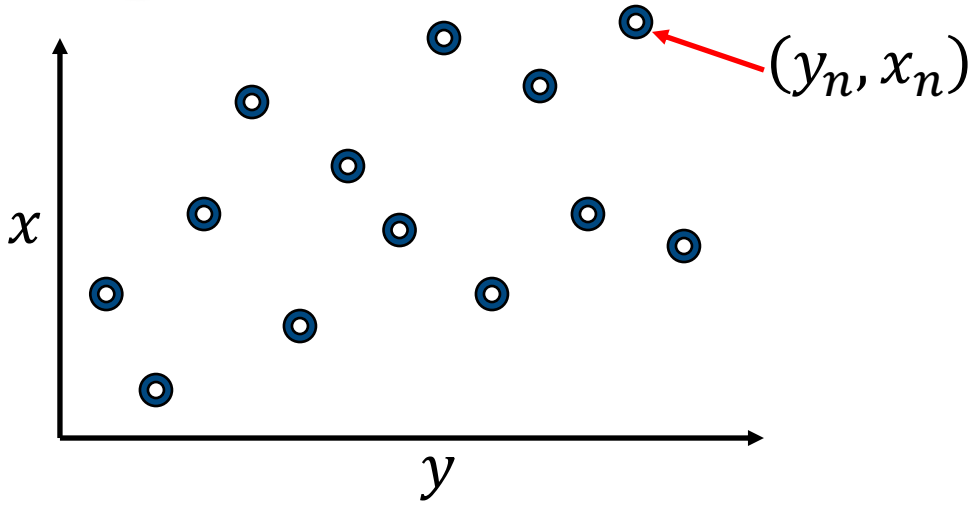
- Underfitting



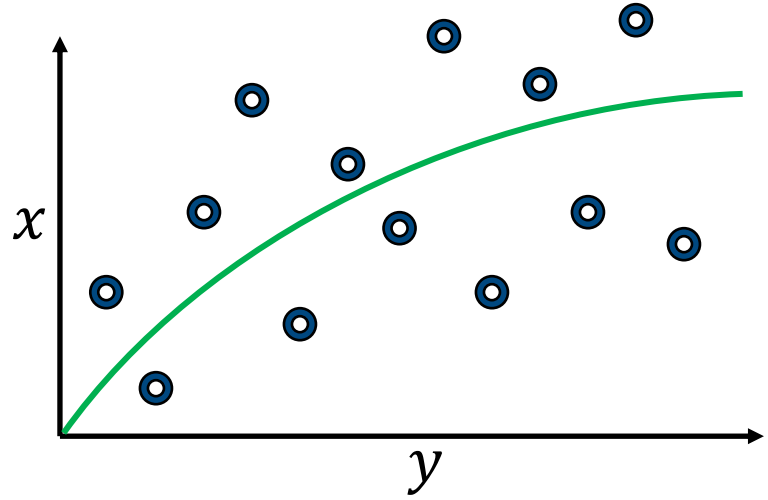
  - Model order too low
  - Doesn't generalize well

Goldilocks Fitting

- Training data



- Best fitting



  - Model order "just right"
  - Best generalization

Partitioning of Labeled Data

- Let (x_k, y_k) for k $\in S = \{0, \cdots, K-1\}$ be the full set data.
  - y_k is the input data.
  - x_k is the label or "ground truth" data.
- Typically, we randomly partition* the data into three subsets:
  - S_T is the training data
  - S_V is the validation data
  - S_E is the testing (evaluation) data

    * Note that "partition" means $S = S_\mathrm{T} \cup S_V \cup S_E$ and $\emptyset = S_T \cap S_V = S_T \cap S_E = S_V \cap S_E$
- For each partition, we define a loss function:

$$L_T(\theta) = \frac{1}{K} \sum_{k \in S_T} \|y_k - f_\theta(x_k)\|^2$$

$$L_V(\theta) = \frac{1}{K} \sum_{k \in S_V} \|y_k - f_\theta(x_k)\|^2$$

$$L_E(\theta) = \frac{1}{K} \sum_{k \in S_E} \|y_k - f_\theta(x_k)\|^2$$

Roles of Data

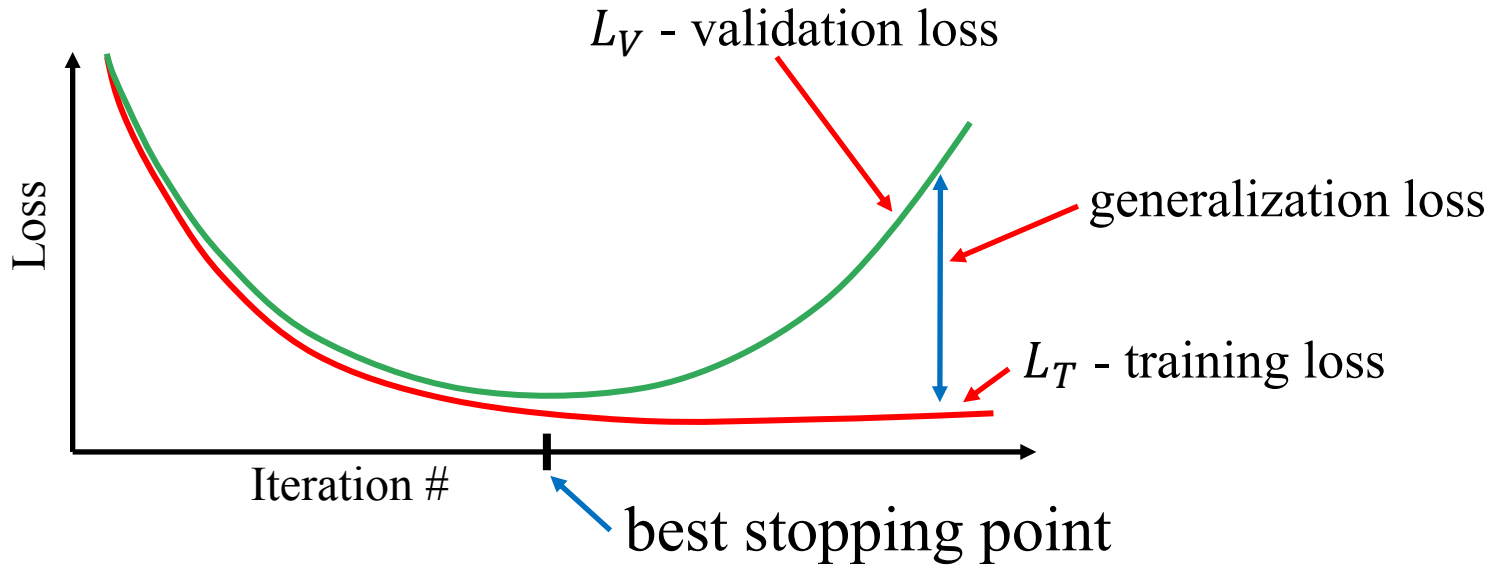
- Training data:
  - Only data used to train model

$$\theta^* = \arg\min_{\theta}\{L_T(\theta)\}$$
$$= \arg\min_{\theta}\left\{\frac{1}{K}\sum_{k\in S_T}\|y_k - f_\theta(x_k)\|^2\right\}$$

- Validation data:
  - Used to compare models of different order.
- Testing data
  - Used for final evaluation of model performance.

Loss Function Convergence

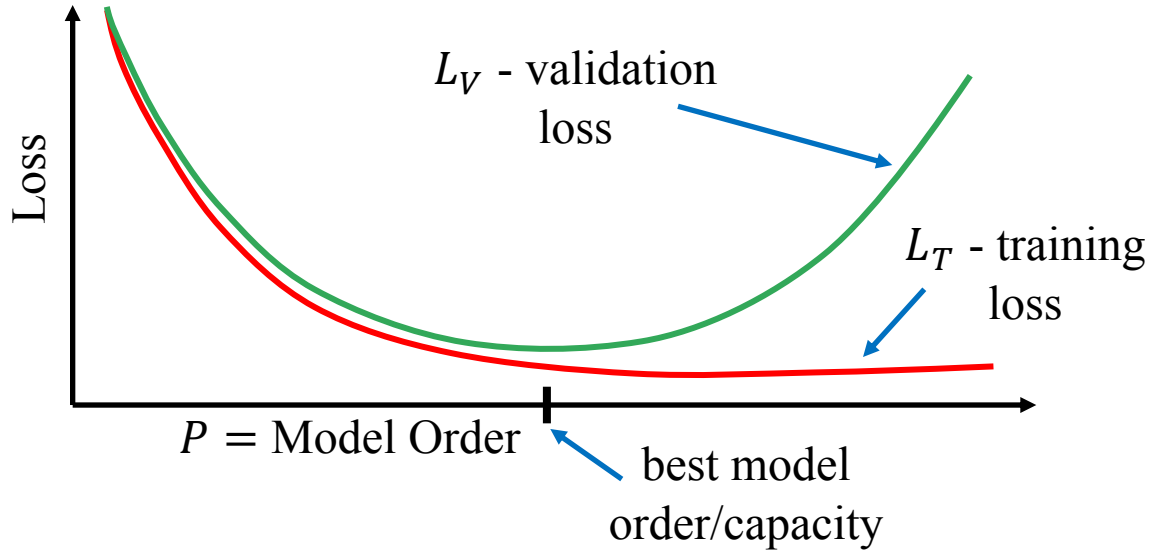
- Loss vs. iterations of gradient-based optimization



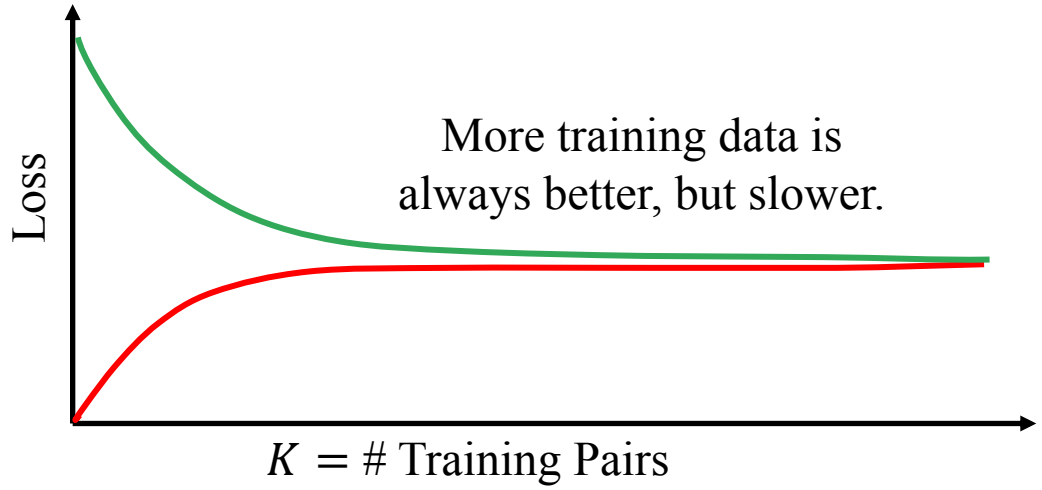
- Notice:
  - As training continues, the model is overfit to the data
  - Best to stop training when L_V is at a minimum
  - Model order is too high, but early termination of training can help fix problem

Loss vs. Model Order vs. # Training Pairs

- Loss vs. model order

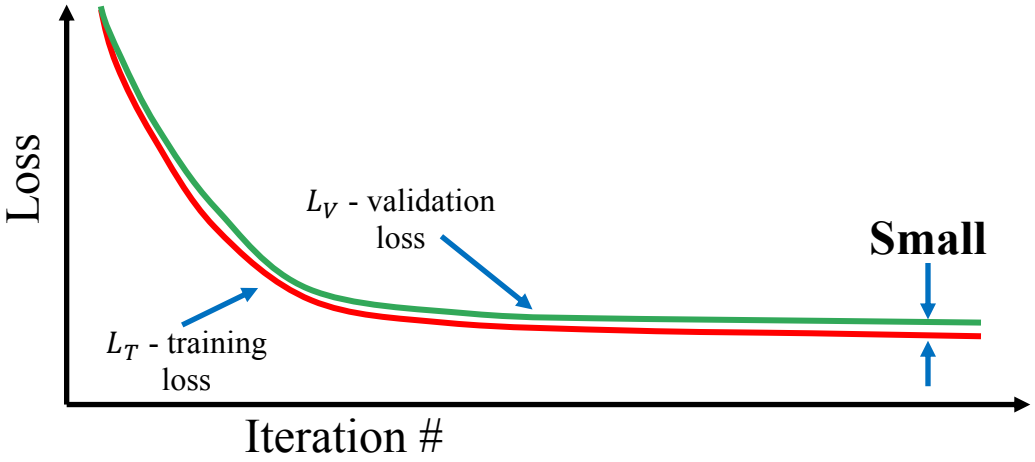


- Loss vs. # of training pairs

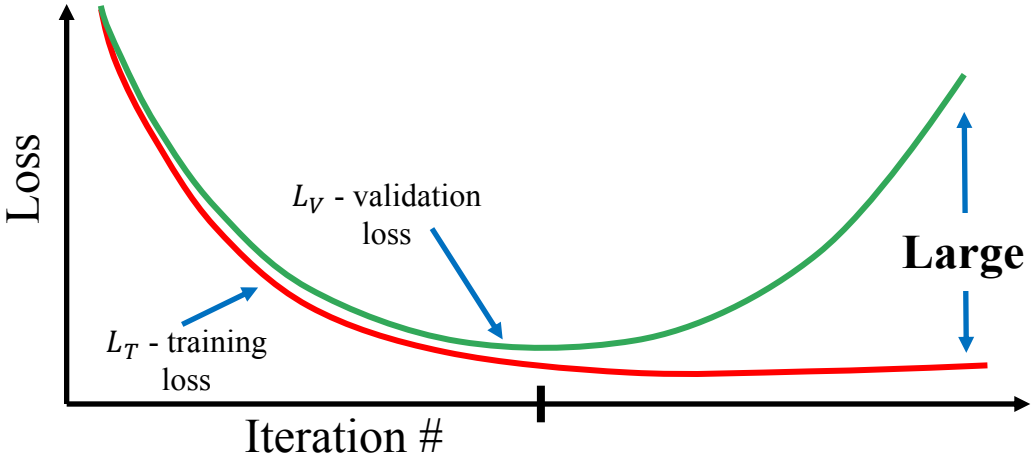


What are L_T and L_V telling you?

- Model order/model capacity may be too low…

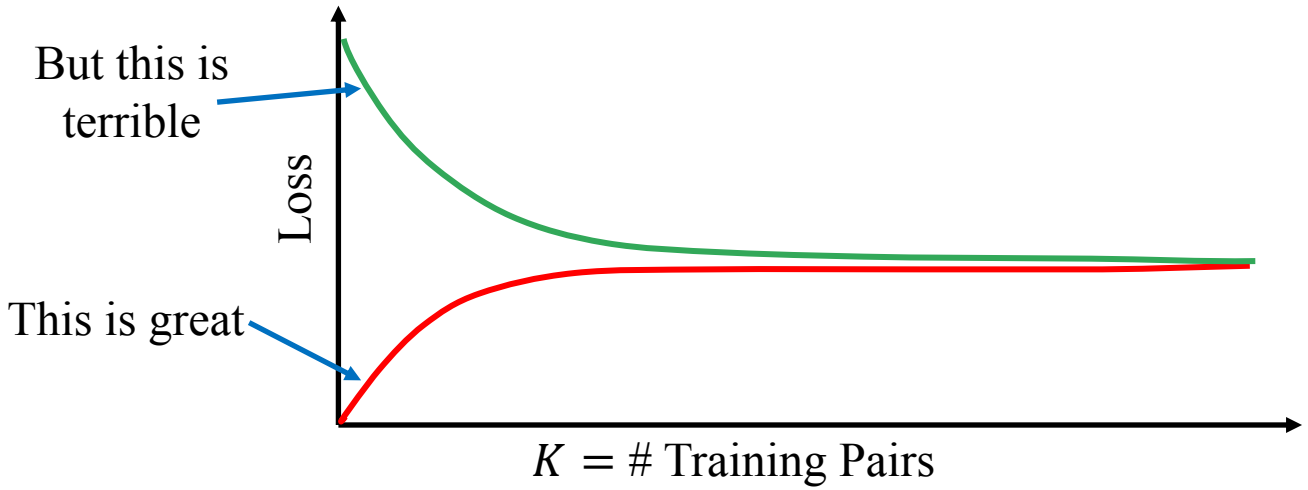


- Model order/model capacity may be too high…



Never Test on Training Data!

- Never report training loss, L_T, as your ML system accuracy!



  - This is like doing a homework problem after you have seen the solution.
  - The network has "memorized" the answers.
- Don't even report validation loss, L_V, as your ML system accuracy.
  - This is also biased by the fact that your tuned model order parameters.
- Only report testing loss, L_E, as your ML system accuracy.
  - This data is sequestered to ensure it is an unbiased estimate of loss.