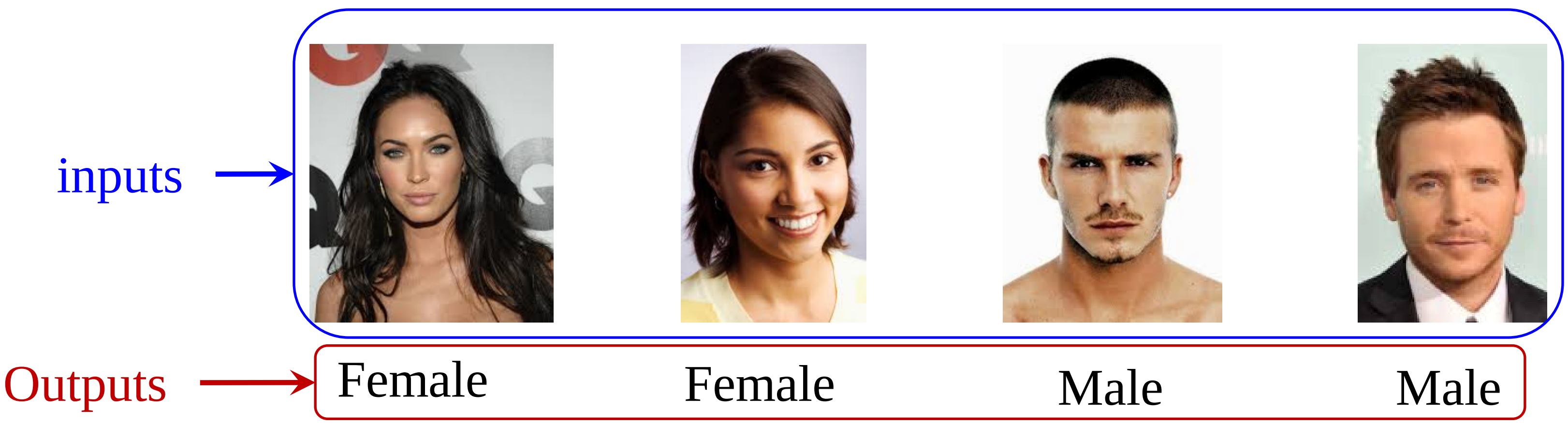
CZ4041/CE4041: Machine Learning

Lecture 1b: Overview of Supervised Learning

Sinno Jialin PAN
School of Computer Science and Engineering, NTU, Singapore
Homepage: https://personal.ntu.edu.sg/sinnopan/

Supervised Learning

- Recall: in supervised learning, the examples presented to computers are pairs of inputs and the corresponding outputs (i.e., labeled training data), the goal is to "learn" a map or a model from inputs to labels



Supervised Learning (cont.)

Labeled training data

In mathematics,

- Given: a set of $\boxed{\{\boldsymbol{x}_i, y_i\}}$ for $i = 1, \dots, N$, where $\boxed{\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]}$, and $\boxed{y_i}$ is a scalar,
- Goal: to learn a mapping $\boxed{f:}\ \boldsymbol{x} \rightarrow y$ by requiring $f(\boldsymbol{x}_i) = y_i$.

Output

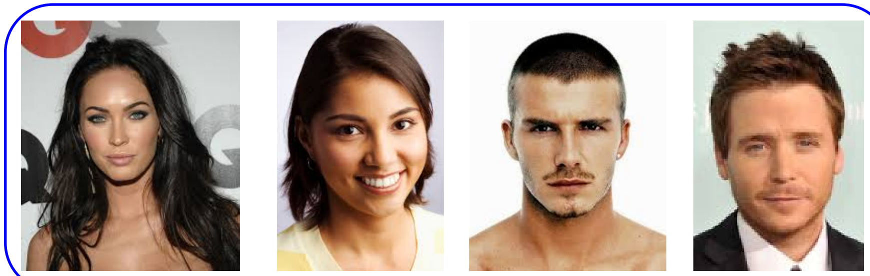
Prediction model

- The learned mapping f is expected to be able to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

A vector of features

Example I

$\boldsymbol{x_1}$ $\boldsymbol{x_2}$ $\boldsymbol{x_3}$ $\boldsymbol{x_4}$



Female Female Male Male

-1 -1 $+1$ $+1$

Image processing & Computer vision tech.

$$y_1 = -1 \quad \boldsymbol{x_1} = [x_{11}, x_{12}, \dots, x_{1m}]$$

$$y_2 = -1 \quad \boldsymbol{x_2} = [x_{21}, x_{22}, \dots, x_{2m}]$$

$$y_3 = 1 \quad \boldsymbol{x_3} = [x_{31}, x_{32}, \dots, x_{3m}]$$

$$y_4 = 1 \quad \boldsymbol{x_4} = [x_{41}, x_{42}, \dots, x_{4m}]$$

Example II

	Compact; easy to operate; very **good** picture quality; looks **sharp**!
	It is also quite **blurry** in very dark settings. I will **never_buy** HP again.

Bag-of-words representation

		F1	F2	F3	F4	F5	F6	…	
y_1	$+1$	1	1	0	0	1	0	…	$\boldsymbol{x}_1$
y_2	-1	0	0	1	1	0	1	…	$\boldsymbol{x}_2$

Dictionary

F1	F2	F3	F4	F5	F6	…
compact	easy	quite	blurry	good	never_buy	…

Examples III

binary categorical continuous class

Tid	**Home Owner**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Divorced	95K	**Yes**

One-hot encoding

	HO	MS_S	MS_M	MS_D	TaxInc	Cheat	
x_1	1	1	0	0	125K	-1	y_1
x_2	0	0	1	0	100K	-1	y_2
x_3	0	0	0	1	95K	$+1$	y_3

Classification v.s. Regression

- For classification, y is discrete
  - If y is binary, then binary classification
  - If y is <u>nominal</u> not binary, then multi-class classification
- For regression, y is continuous
- Can an example be assigned to more than one categories?

Typical Learning Procedure

Two phases:

- Training phase
  - Given labeled training data $\{\boldsymbol{x}_i, y_i\}$, for $i = 1, \ldots, N$
  - Apply supervised learning algorithms on $\{\boldsymbol{x}_i, y_i\}$ to learn a model f such that $f(\boldsymbol{x}_i) = y_i$
- Testing or prediction phase
  - Given unseen test data $\boldsymbol{x}_i^*$, for $i = 1, \ldots, T$
  - Use the trained model f to make predictions $f(\boldsymbol{x}_i^*)$'s

Training data

		F1	F2	F3	F4	F5	F6	…	
y_1	$+1$	1	1	0	0	1	0	…	$\boldsymbol{x}_1$
y_2	-1	0	0	1	1	0	1	…	$\boldsymbol{x}_2$
		…							
y_N	-1	0	1	0	0	1	1	…	$\boldsymbol{x}_N$

Some classification algorithm

$$f: \boldsymbol{x} \rightarrow y$$

Predicted label

$$\boxed{y^*} = f(\boldsymbol{x}^*)$$

Test data

	F1	F2	F3	F4	F5	F6	…	
?	1	1	0	0	1	0	…	$\boldsymbol{x}^*$

Inductive Learning

Training data

		F1	F2	F3	F4	F5	F6	…	
y_1	$\boxed{+1}$	1	1	0	0	1	0	…	$\boldsymbol{x}_1$
y_2	$\boxed{-1}$	0	0	1	1	0	1	…	$\boldsymbol{x}_2$
					…				
y_N	$\boxed{-1}$	0	1	0	0	1	1	…	$\boldsymbol{x}_N$

Predicted label

$\boxed{y^*}$

Test data

	F1	F2	F3	F4	F5	F6	…	
?	1	1	0	0	1	0	…	$\boldsymbol{x}^*$

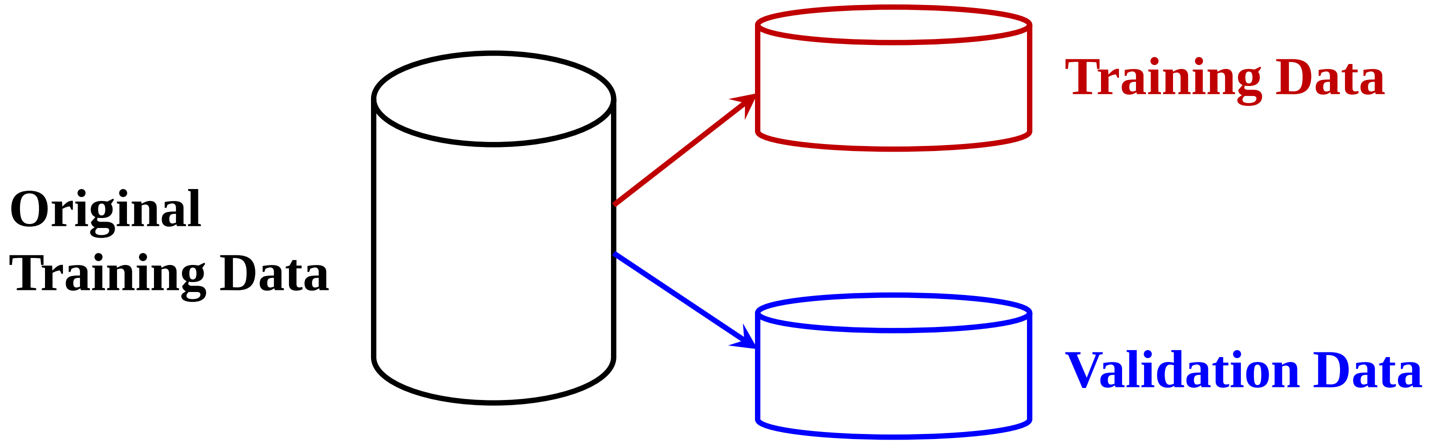
Lazy Learning E.g., K Nearest neighbor classifier

Inductive Learning

- How to learn a predictive model $f(\cdot)$
  - Bayesian Classifiers
  - Decision Trees
  - Artificial Neural Networks
  - Support Vector Machines
  - Regularized Regression Model

Evaluation of Supervised Learning

- Recall: the learned predictive model should be able to make predictions on previously unseen data as accurately as possible
- Solution: the whole training data is divided into “training” and “test” sets, with “training” set used to learn the predictive model and “test” set used to validate the performance of the learned model



All Training data

		F1	F2	F3	F4	F5	F6	…	
y_1	$+1$	1	1	0	0	1	0	…	$\boldsymbol{x}_1$
y_2	-1	0	0	1	1	0	1	…	$\boldsymbol{x}_2$
		…							
y_N	-1	0	1	0	0	1	1	…	$\boldsymbol{x}_N$

Used for training

		F1	F2	F3	F4	F5	F6	…	
y_1	$+1$	1	1	0	0	1	0	…	$\boldsymbol{x}_1$
y_2	-1	0	0	1	1	0	1	…	$\boldsymbol{x}_2$
		…							
y_L	$+1$	1	1	1	0	1	1	…	$\boldsymbol{x}_L$

Used for evaluation

		F1	F2	F3	F4	F5	F6	…	
y_{L+1}	$+1$	1	1	0	0	1	0	…	$\boldsymbol{x}_{L+1}$
		…							
y_N	$+1$	1	1	1	0	1	1	…	$\boldsymbol{x}_N$

Learn

$$f: \boldsymbol{x} \rightarrow y$$

$$\hat{y}_i = f(\boldsymbol{x}_i)$$

for $i = L+1, \ldots, N$

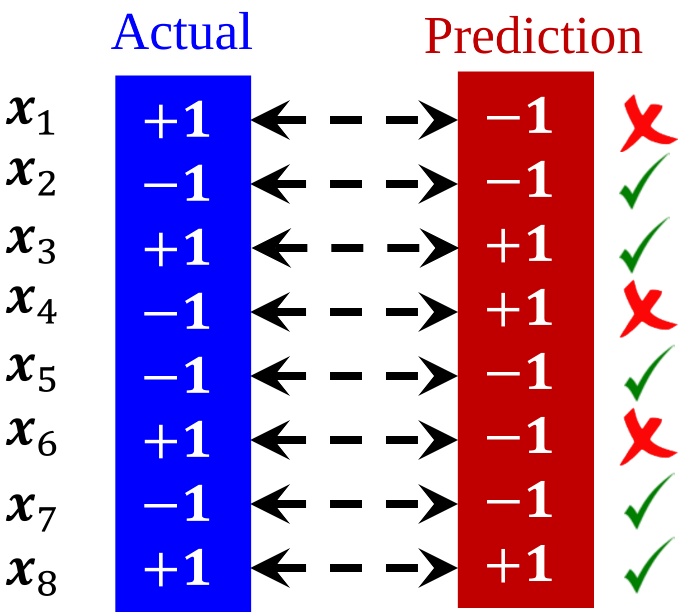
Evaluate

Some evaluation metric

$\hat{y}_{L+1}$		y_{L+1}
…	v.s	…
$\hat{y}_N$		y_N

Common Evaluation Metrics

- Classification: Accuracy, error rate, F-score etc.

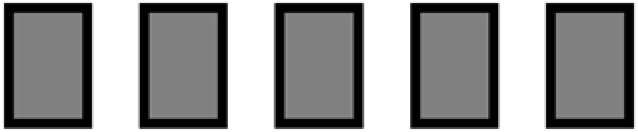


$$\text{Accuracy} = 5/8$$
$$\text{Error rate} = 3/8$$

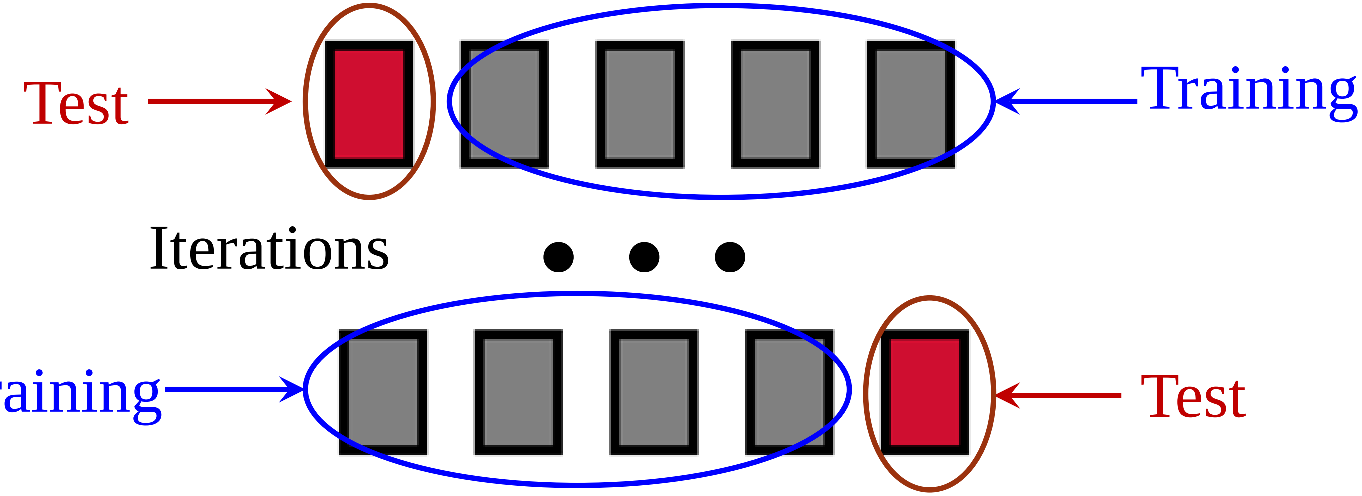
- Regression: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)

Cross Validation

- k-fold cross-validation: partition data into k subsets of the same size



- Hold aside one group for testing and use the rest to build model

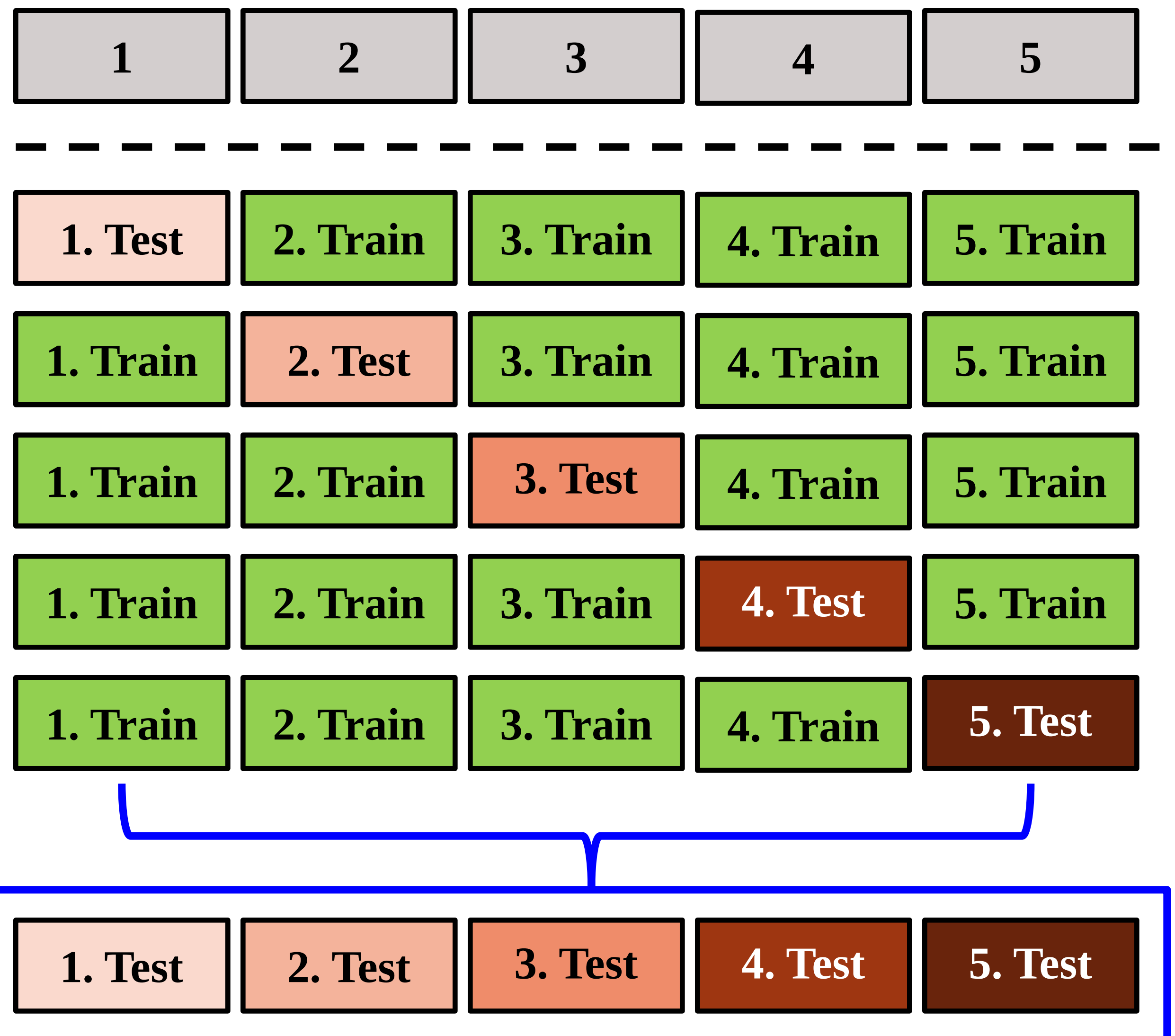


5-fold Cross-Validation

Partition into 5 subsets

Hold aside 1 group for testing and use the rest 4 for training

Evaluate performance



Notes on Cross-Validation

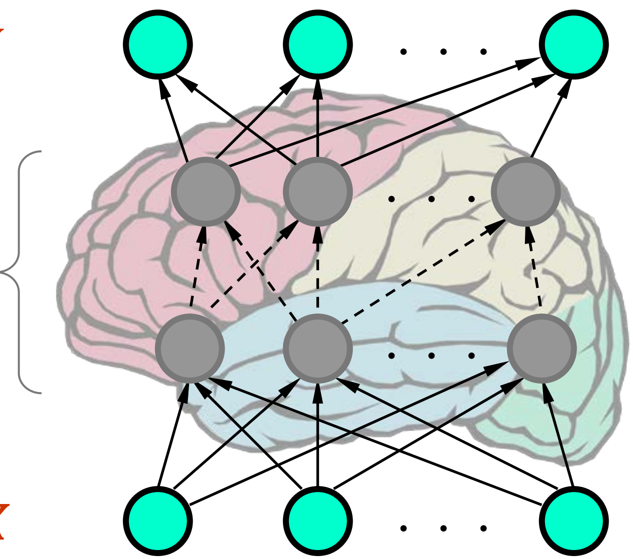
- Using Neural Networks as an example

Parameters of a neural network, which are to be **learned** from training data

Output Layer *Y*

Hidden Layer(s)

Input Layer ***X***



$$f(x;\boxed{\Theta})$$

Hyper-parameters of a specific neural network: how many hidden layers, how many hidden nodes in a specific hidden layer, etc.

$$f(x;\Theta,\boxed{H})$$

Note: hyper-parameters need to be manually determined, not learned from training data!

For a specific setting of hyper-parameters H_1, e.g., 10 convolutional layers followed by 2 fully connected layers, etc.

$f(x;\Theta_1', H_1)$	**1. Test**	**2. Train**	**3. Train**	**4. Train**	**5. Train**
$f(x;\Theta_2', H_1)$	**1. Train**	**2. Test**	**3. Train**	**4. Train**	**5. Train**
$f(x;\Theta_3', H_1)$	**1. Train**	**2. Train**	**3. Test**	**4. Train**	**5. Train**
$f(x;\Theta_4', H_1)$	**1. Train**	**2. Train**	**3. Train**	**4. Test**	**5. Train**
$f(x;\Theta_5', H_1)$	**1. Train**	**2. Train**	**3. Train**	**4. Train**	**5. Test**

Evaluation score s_1 for the hyper-parameters setting H_1

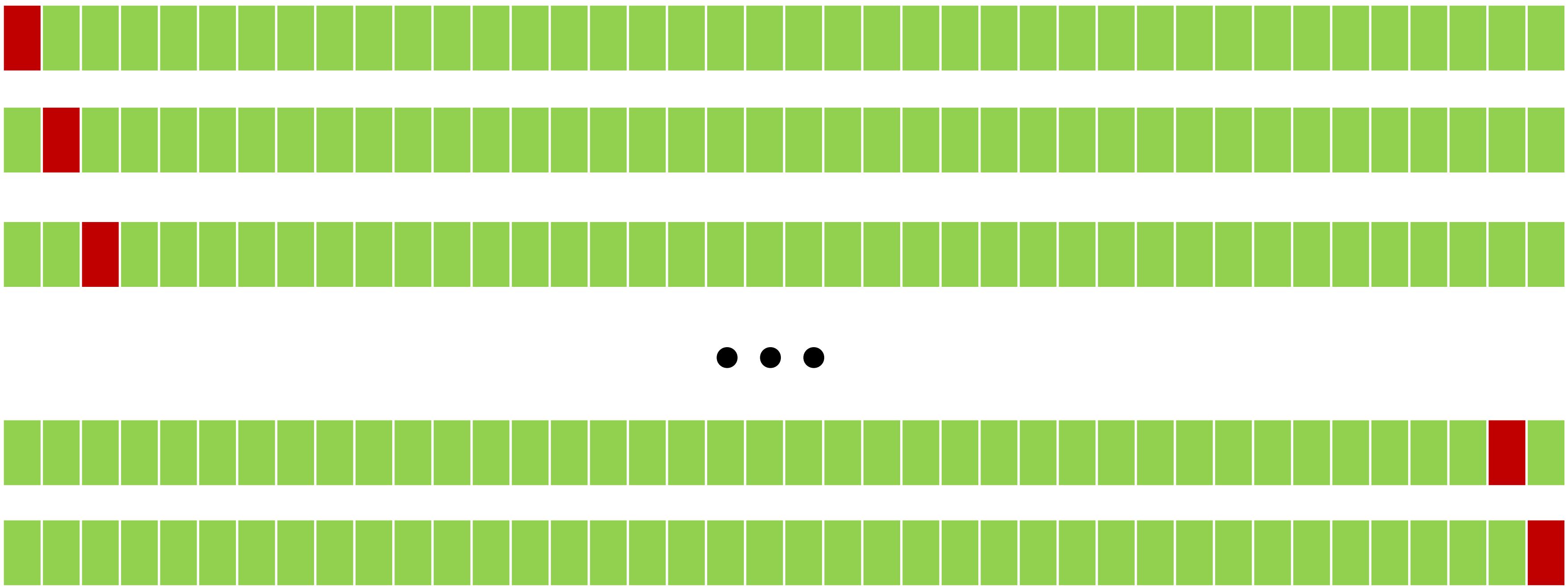
$f(X_1;\Theta_1', H_1)$	$f(X_2;\Theta_2', H_1)$	$f(X_3;\Theta_3', H_1)$	$f(X_4;\Theta_4', H_1)$	$f(X_5;\Theta_5', H_1)$
1. Test	**2. Test**	**3. Test**	**4. Test**	**5. Test**

Based on evaluation scores of other hyper-parameters settings, H_2, H_3, …, to choose the best one

Once the best hyper-parameters setting H_* is determined, all training data is used to train the neural network $f(x;\Theta_*, H_*)$ to deploy for use

Leave-One-Out

- k-fold cross-validation special case:
  - leave-one-out, $k = n$, n is the number of data instances for training



Next Week …

- We will start by introducing Bayesian Classifiers

Thank you!