LONDON METROPOLITAN UNIVERSITY

Machine Learning CS7052

Lecture 9, Model Evaluation

Dr. Elaheh Homayounvala

Week 9

Outline of today’s lecture

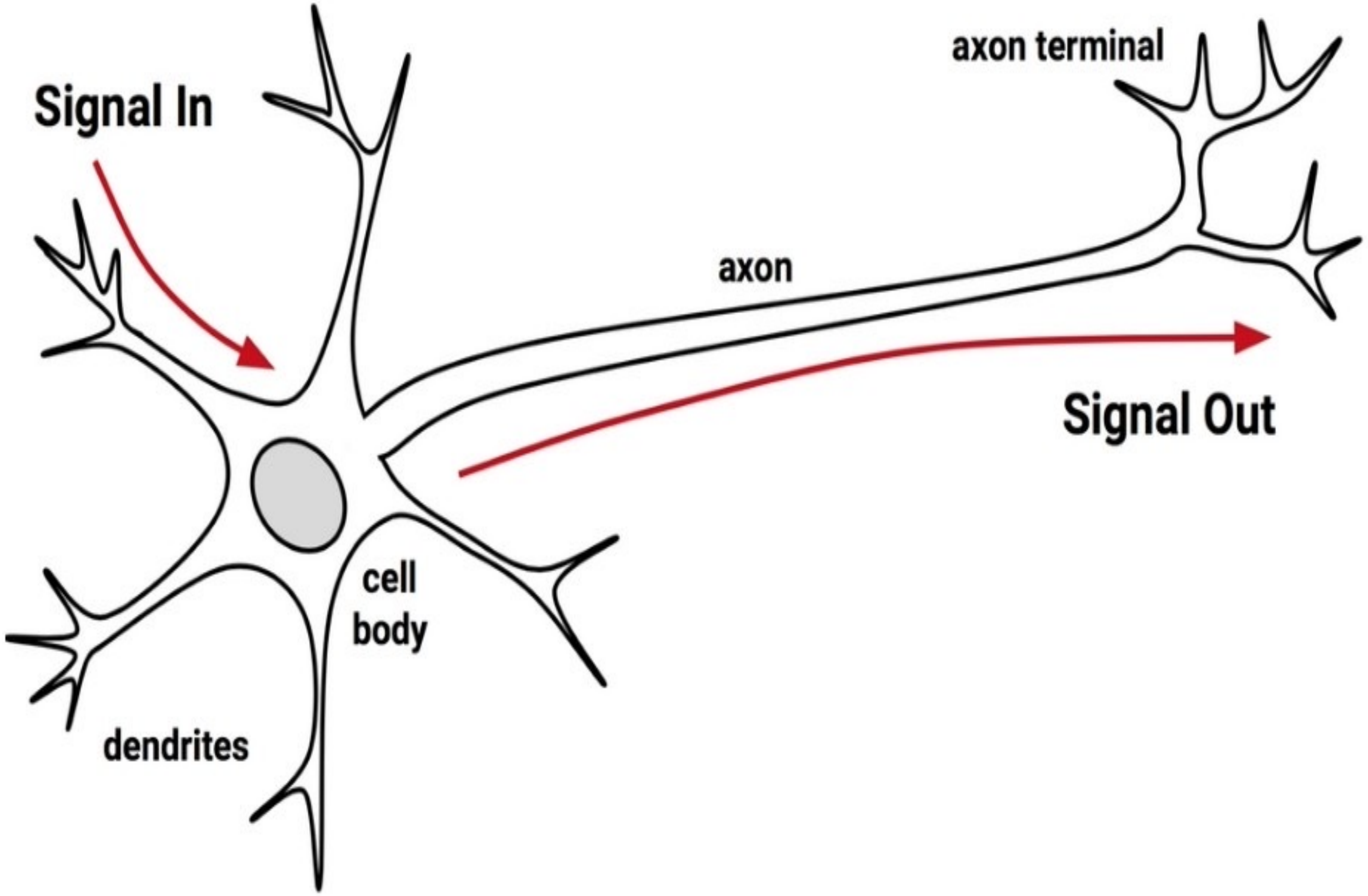
- Summary of Neural networks and deep learning
- Model evaluation and improvements, cross-validation

Review last week

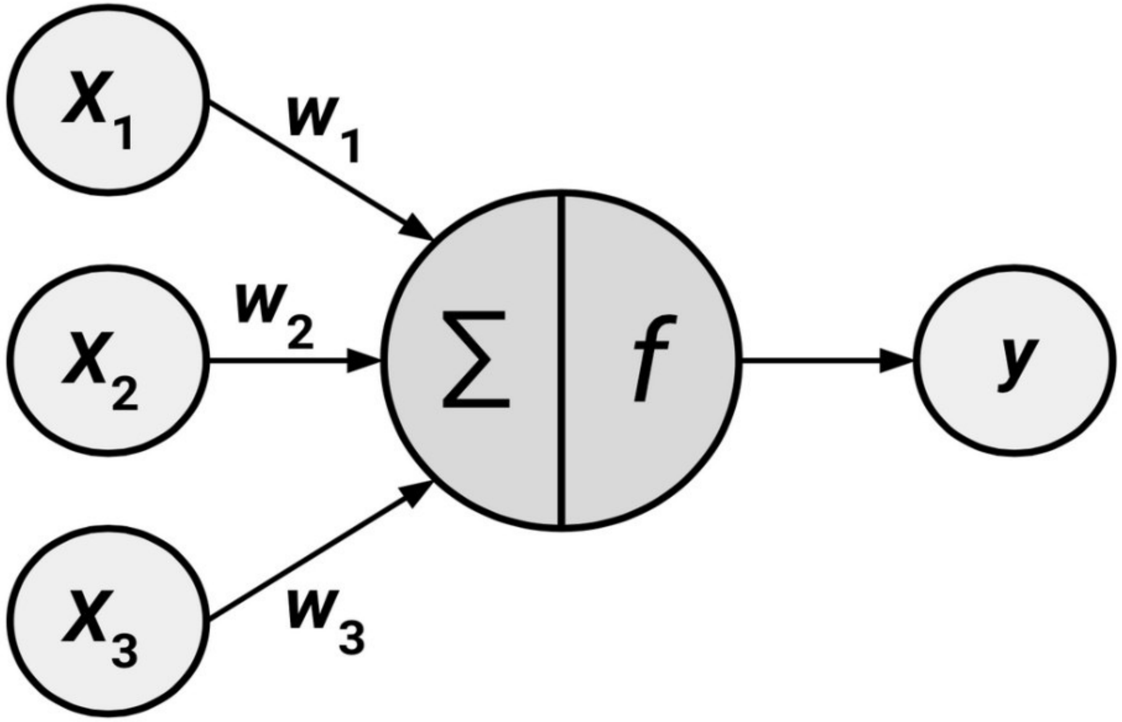
Neural networks

Deep learning

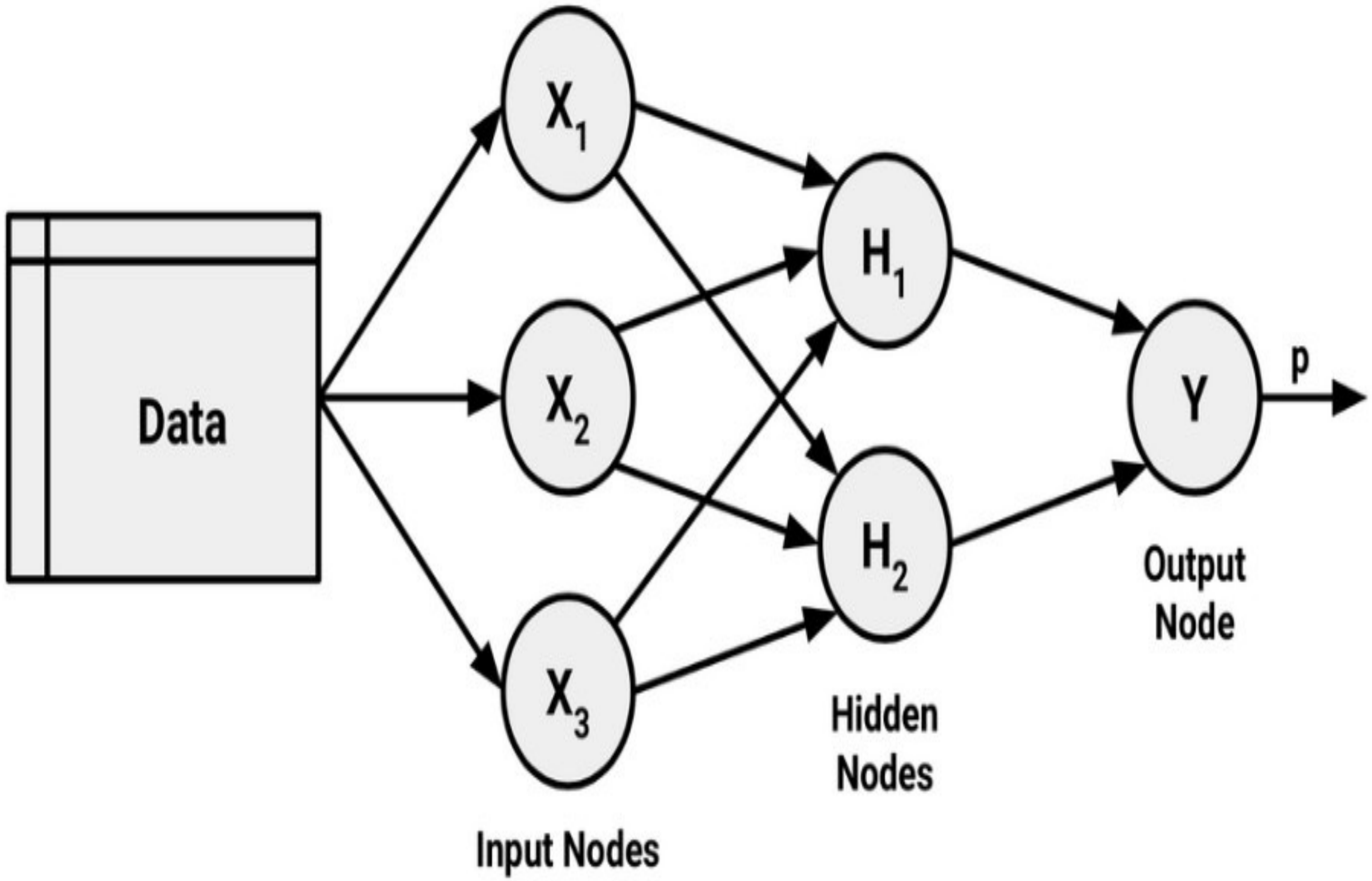
Natural Neural Network



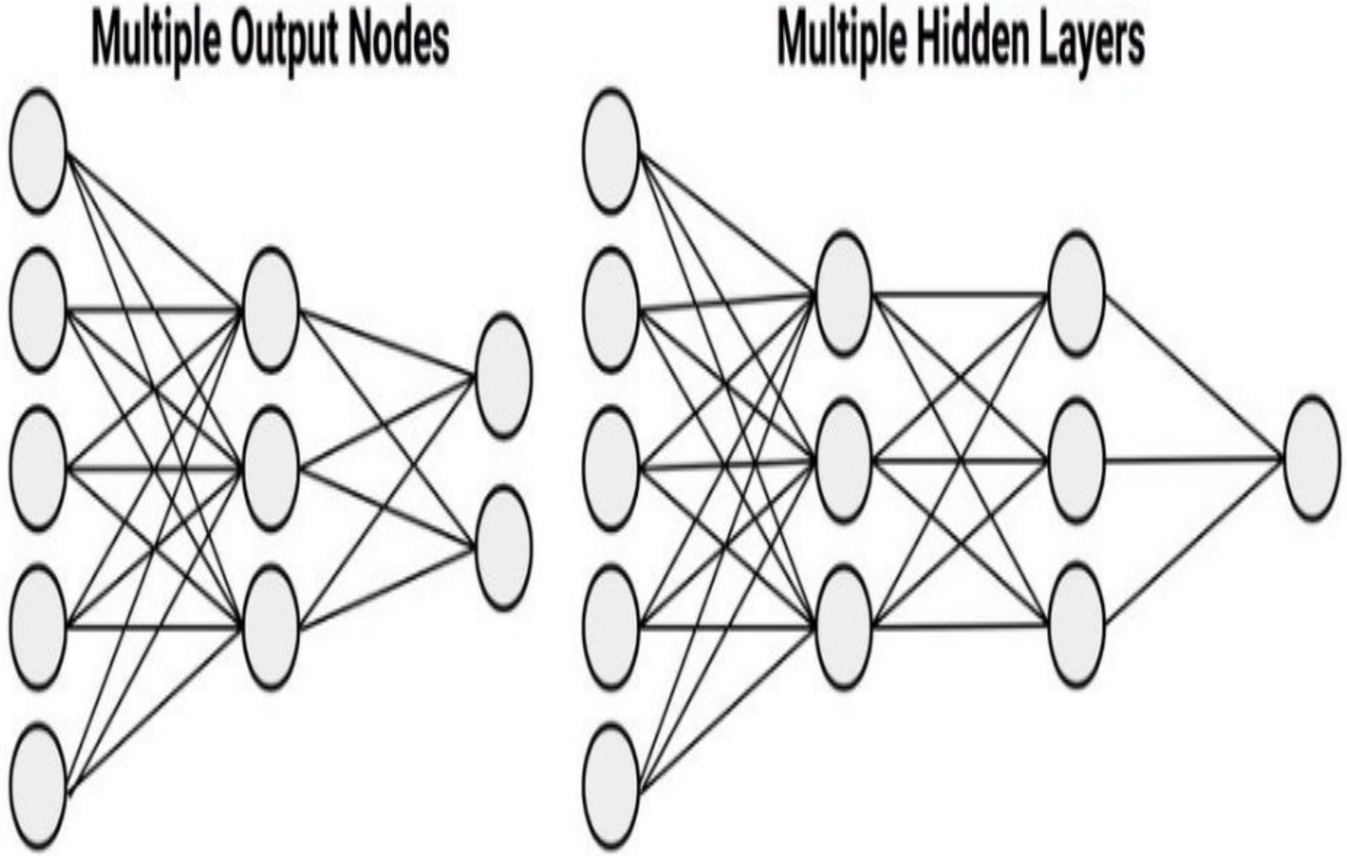
Artificial Neural Network



Artificial Neural Network



Multiple NN



Multi-layer Perceptron

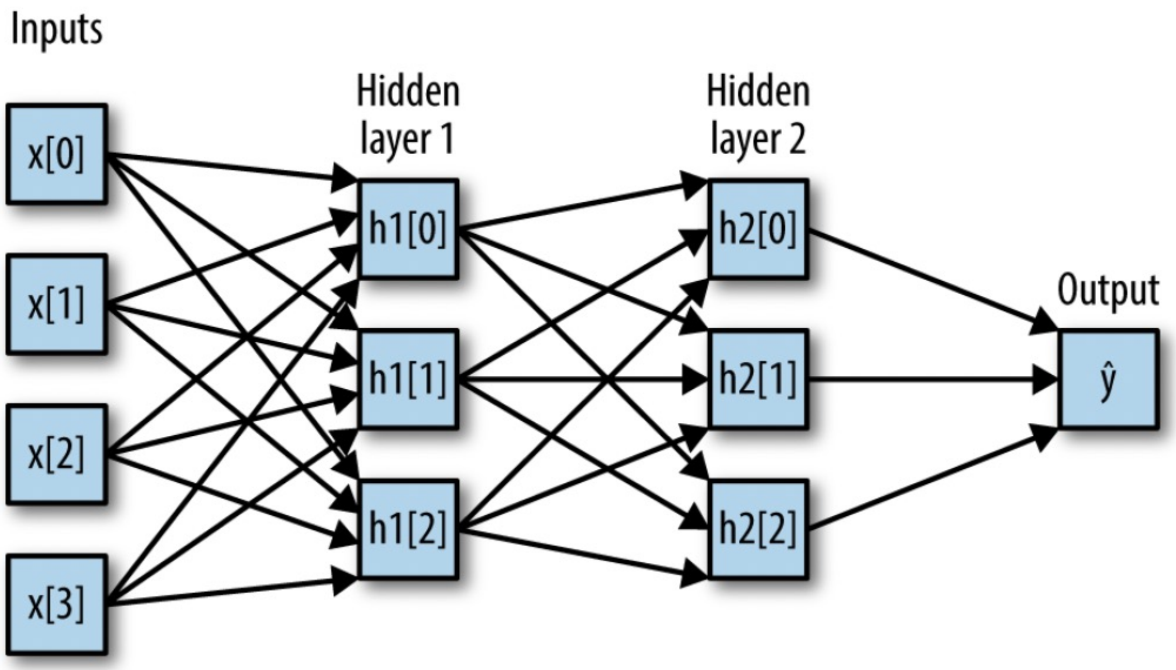


Figure 2-47. A multilayer perceptron with two hidden layers

Muller and Guido’s book, page 110

Computing output

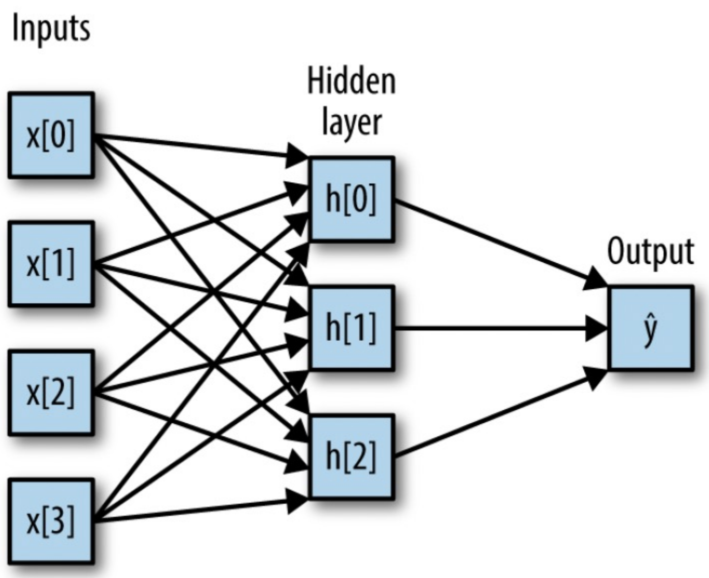
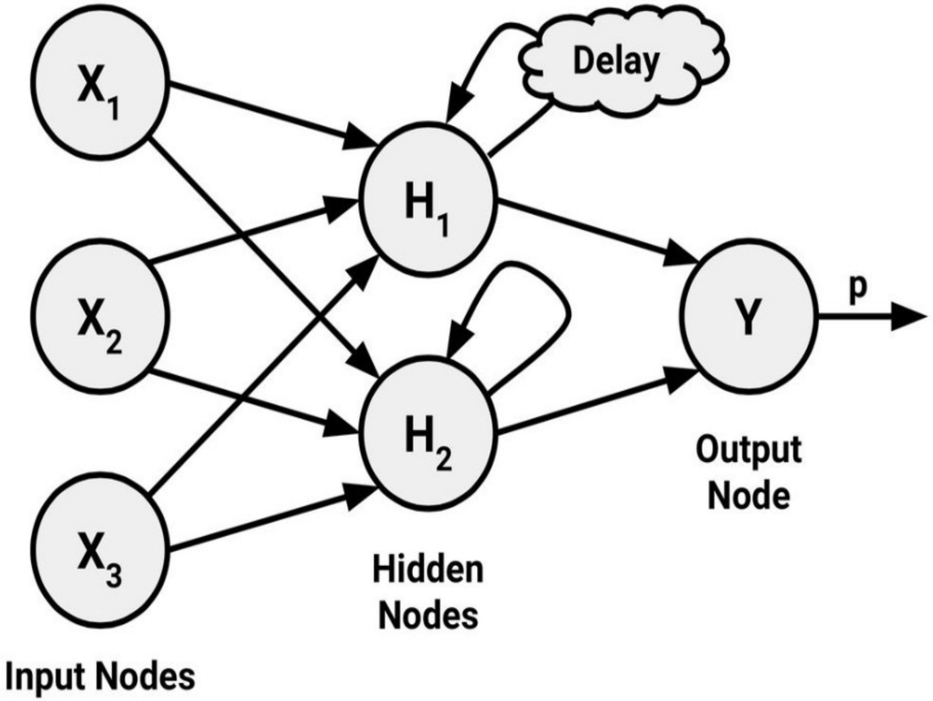


Figure 2-45. Illustration of a multilayer perceptron with a single hidden layer

$$h[0] = \tanh(w[0, 0] * x[0] + w[1, 0] * x[1] + w[2, 0] * x[2] + w[3, 0] * x[3] + b[0])$$
$$h[1] = \tanh(w[0, 1] * x[0] + w[1, 1] * x[1] + w[2, 1] * x[2] + w[3, 1] * x[3] + b[1])$$
$$h[2] = \tanh(w[0, 2] * x[0] + w[1, 2] * x[1] + w[2, 2] * x[2] + w[3, 2] * x[3] + b[2])$$
$$\hat{y} = v[0] * h[0] + v[1] * h[1] + v[2] * h[2] + b$$

Recurrent NN



RNN

- Can handle sequential data
- Considers the current input and also the previously received inputs
- Can memorise previous inputs to its internal memory

- Recurrent neural network works on the principle of saving the output of a layer and feeding this back to the input in order to predict the output of the layer

Applications of RNN

- Image captioning
- ‘A dog catching a ball in the mid-air’

- Time-series problem (stock price prediction)

- Natural language process (text mining and sentiment analysis)

- Machine translation

Model Evaluation

Cross-validation

Grid search

Confusion matrix

Model evaluation and improvement

- Model evaluation
- Improvement by tuning parameters

Model evaluation so far

To evaluate our supervised models, so far we have:

- split our dataset into a training set and a test set using the train_test_split function,
- built a model on the training set by calling the fit method,
- and evaluated it on the test set using the score method (which for classification computes the fraction of correctly classified samples)

Cross-validation

- Cross-validation is a statistical method of evaluating generalization performance that is more stable and thorough than using a split into a training and a test set

- The data is instead split repeatedly, and multiple models are trained

k-fold cross-validation

- where k is a user-specified number
- usually 5 or 10.

5-fold cross validation

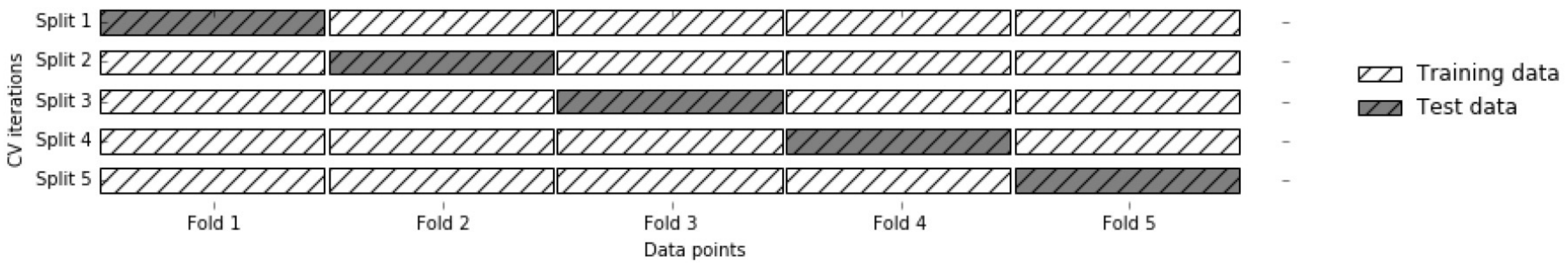


Figure 5-1. Data splitting in five-fold cross-validation

The first model is trained using the first fold as the test set,
The remaining folds (2–5) are used as the training set

Benefits of cross-validation

- Single split is random
- Imagine we are lucky, hard one to predict are in the training set

1. when using cross-validation, each example will be in the test set exactly once
2. We'll know how sensitive our model is to the selection of the training dataset
3. We use our data more effectively, more data for training means more accurate model

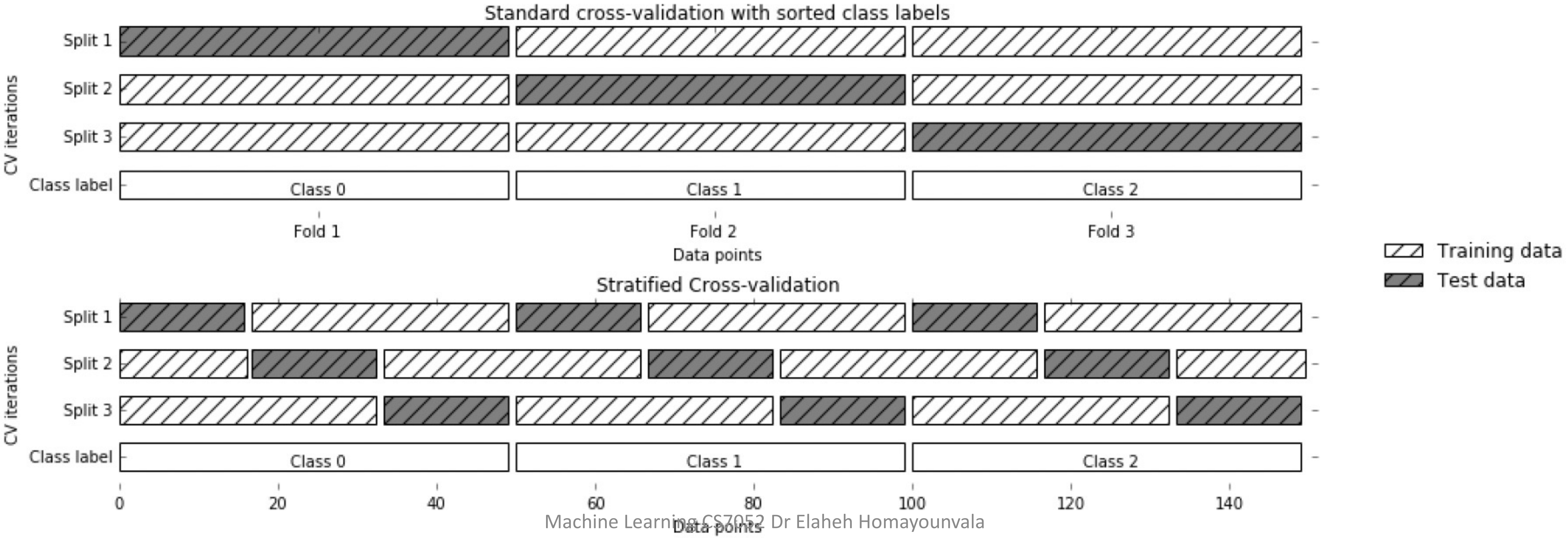
Disadvantage

- The main disadvantage of cross-validation is increased computational cost.

- Cross-validation can only tell us how well a model generalises

Stratified cross-validation

- If 90% of your samples belong to class A and 10% of your samples belong to class B, then stratified cross-validation ensures that in each fold, 90% of samples



Tuning parameters

- Cross-validation only tells how well a model generalises

Next step:

- improve the model's generalisation performance by tuning its parameters

- Finding the values of the parameters of a model is a tricky task

LONDON METROPOLITAN UNIVERSITY

Grid search

- Trying all possible combinations of the parameters of interest.
- A simple grid search just as for loops over the two parameters

	C = 0.001	C = 0.01	...	C = 10
gamma=0.001	SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	...	SVC(C=10, gamma=0.001)
gamma=0.01	SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	...	SVC(C=10, gamma=0.01)
...	...	...	...	...
gamma=100	SVC(C=0.001, gamma=100)	SVC(C=0.01, gamma=100)	...	SVC(C=10, gamma=100)

The Danger of Overfitting

- Always remember overfitting
- We can not simply choose the best accuracy
- This accuracy won't necessarily carry over to new data.
- we used the test data to adjust the parameters, we can no longer use it to assess how good the model is.
- we need an independent dataset to evaluate, one that was not used to create the model.

Train set, Validation set and test set

- Split the data again, so we have three sets:
  - the training set to build the model,
  - the validation (or development) set to select the parameters of the model,
  - test set to evaluate the performance of the selected parameters

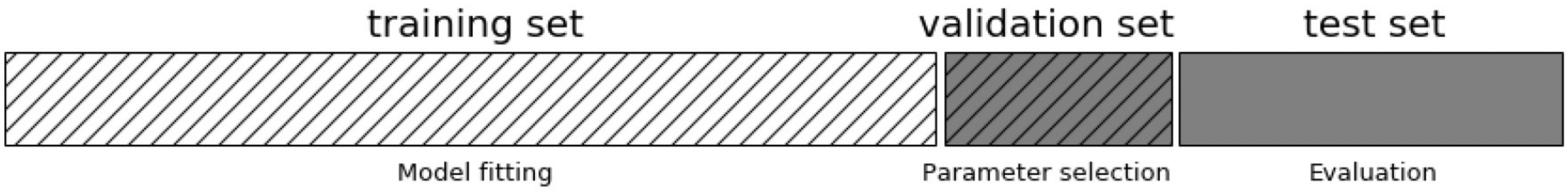


Figure 5-5. A threefold split of data into training set, validation set, and test set

After using validation set

- After selecting the best parameters using the validation set,
- We can rebuild a model using the parameter settings we found,
- but now training on both the training data and the validation data.
- Now use test set to evaluate the performance of new tuned model

Confusion matrices

- One of the most comprehensive ways to represent the result of evaluating binary classification is using confusion matrices

Confusion matrix

TRUE POSITIVE: correctly classified samples belonging to the positive class
TRUE NEGATIVE correctly classified samples belonging to the negative class

Prediction

Ground Truth

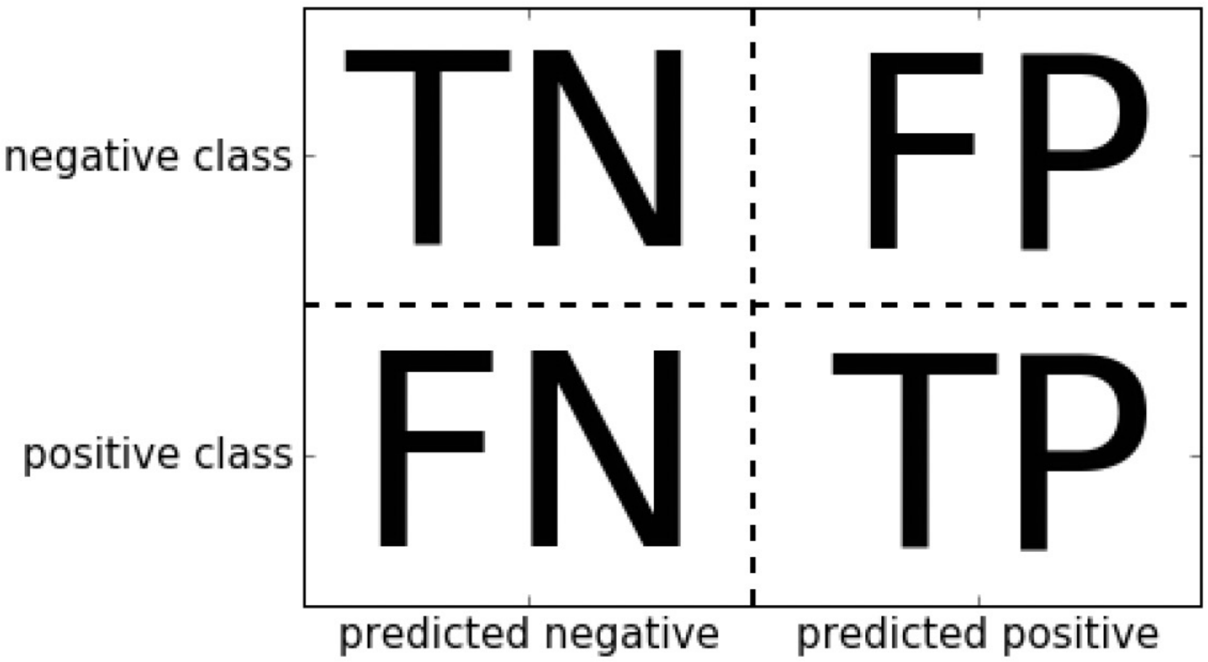


Figure 5-11. Confusion matrix for binary classification

Accuracy and confusion matrix

- Accuracy is the number of correct predictions (TP and TN) divided by the number of all samples:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN} + \text{FP} + \text{FN}}$$

Measures related to confusion matrix

There are several other ways to summarize the confusion matrix, with the most common ones being:

- Precision
- Recall

Precision

- Precision measures how many of the samples predicted as positive are actually positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

- Precision is used as a performance metric when the goal is to limit the number of false positives.

Recall

- Recall measures how many of the positive samples are captured by the positive predictions:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

- Recall is used as performance metric when we need to identify all positive samples

f-score or f-measure or f1-score

- One way to summarize precision and recall is the f-score or f-measure, which is with the harmonic mean of precision and recall:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Multi-class classification

- Confusion matrix:

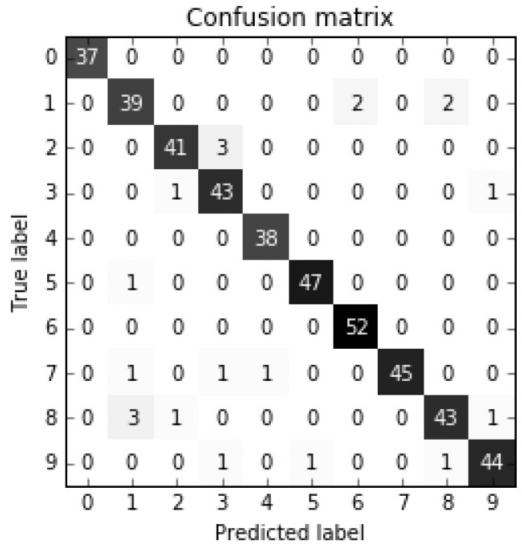


Figure 5-18. Confusion matrix for the 10-digit classification task

Compare Two cat classifiers

- Calculate precision and recall

		Prediction	
		Non-cat	cat
Ground truth	Non-cat	2	1
	cat	2	3

Compare Two cat classifiers

- Calculate precision and recall

		Prediction	
		Non-cat	cat
Ground truth	Non-cat	200	10
	cat	2	3

Another example on confusion matrix

- Draw confusion matrix for this classification and calculate recall, precision and f-score