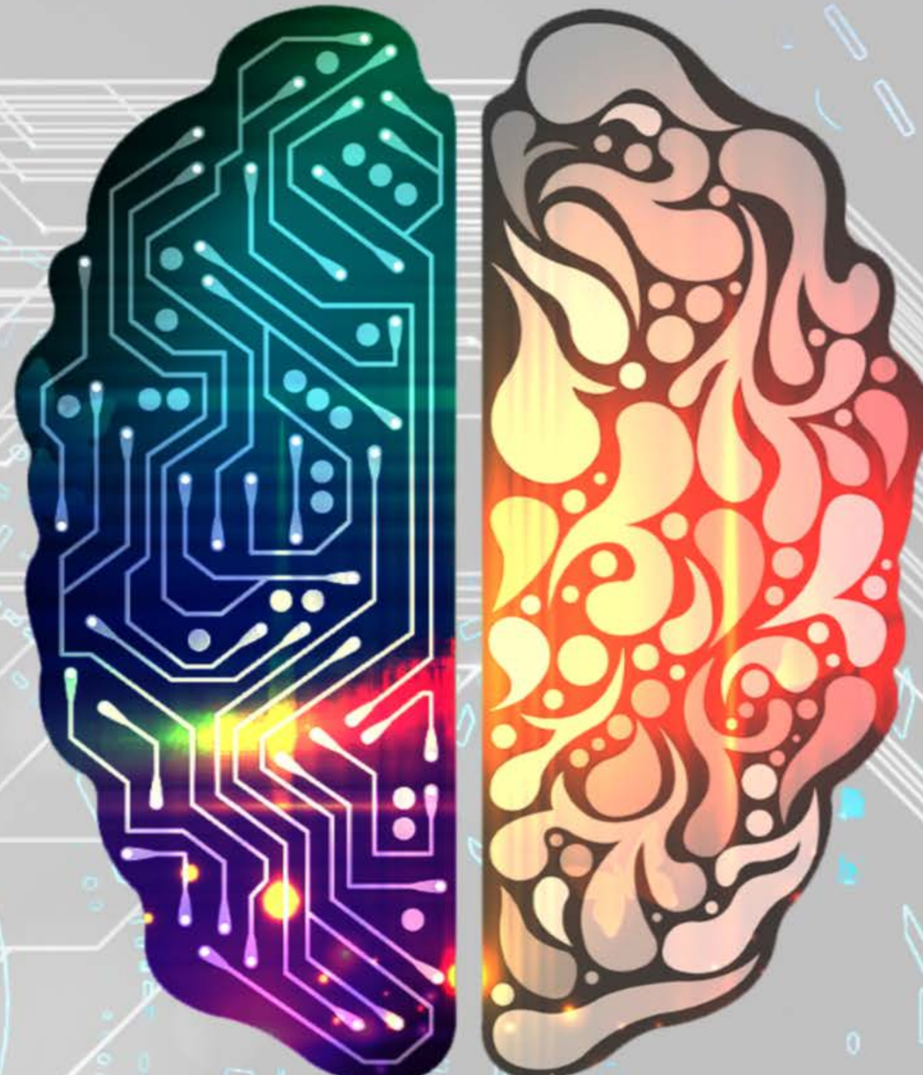
جامعة قطر
QATAR UNIVERSITY

CMPS 460 – Spring 2022

MACHINE LEARNING

Tamer Elsayed



1.d

Formalizing the Learning Problem

Chapter 1

ML as Function Approximation

With DT?

Problem setting

- Set of possible instances X
- Set of labels Y
- Unknown target function $f^*: X \rightarrow Y$
- Set of function hypotheses $\mathcal{F} = \{f \mid f : X \rightarrow Y\}$

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$ of unknown target function f^*

Output

- Hypothesis $f \in \mathcal{F}$ that **best approximates** the target function f^*

To learn f, we need to know:

1. ***How good/bad the predictions are***

2. ***How to model the data***

Loss Function

- A measure of error: how bad a system's prediction is.

- $l(y, f(x))$ where y is the truth and $f(x)$ is the system's prediction

$$\text{e.g., } l(y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases}$$

- Decided based on goals of learning
  - e.g., for regression?

Where does the data come from?

- Data generating distribution
  - A probability distribution D over (x, y) pairs
    - Some pairs are more probable than others.

- We don't know what D is!
  - We only get a random sample from it: ***our training data***

Expected loss

- f should make good predictions
  - as measured by loss l
  - on **future** examples that are also drawn from D
- Formally

  ε, the expected loss of f over D with respect to l should be small

$$\varepsilon \triangleq \mathbb{E}_{(x,y)\sim D}\{l(y,f(x))\} = \sum_{(x,y)} D(x,y)\,l(y,f(x))$$

Can we compute this?

Training error

- We can't compute expected loss because we don't know what D is!
- We only have a sample of D
  - training examples $\{(x^{(1)},y^{(1)}),\ldots(x^{(N)},y^{(N)})\}$
- All we can compute is the training error

$$\varepsilon \triangleq \sum_{n=1}^{N} \frac{1}{N} l(y^{(n)}, f(x^{(n)}))$$

Is that sufficient?

Training error is not sufficient!

- Goal is **NOT** to build a model that gets 0% error on the training data.
  - this would be easy!

- A tree can classify training data perfectly, yet classify new examples incorrectly.

Why?

Formalizing Induction

- **Given**
  - a loss function l
  - a sample from some unknown data distribution D
- **Our task** is to compute a function f that has low ***expected*** error over D with respect to l.

$$\mathbb{E}_{(x,y)\sim D}\{l(y,f(x))\} = \sum_{(x,y)} D(x,y)\, l(y,f(x))$$

We care about generalization to new (unseen) examples