0.1 Learning representations of data

0.1.1 AI/MI/ML/DL

Definition:

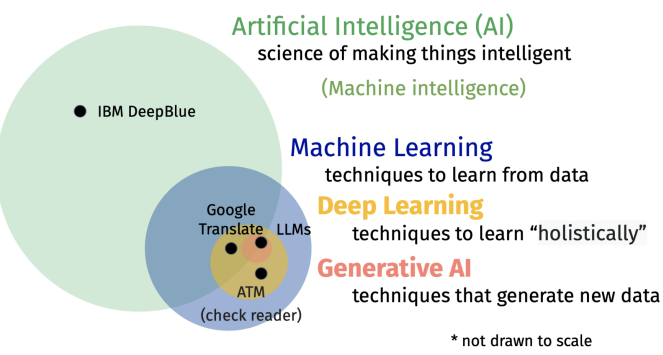

Figure 1:

0.1.2 Learning algorithms

Definition: "A **computer program M** is said to **learn** from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at **tasks in T**, as measured by **P**, improves with **experience E**"

- **experience E** ∼ Data
- **performance measure P** ∼ Loss function, evaluation metric
- **tasks T** ∼ "Prediction problem"
- **computer program M** ∼ Model
- **learn** ∼ Optimize

0.1.3 Linear models

Example: **Linear Regression**

$$W \cdot x = y \tag{1}$$

- **E** (x and y)
- **P** mean squared error
- **T** Predict y from x
- **M** Linear model (W)
- **learn?** Analytical solution or gradient descent

Example: **Generalized Linear models in equations**

$$\text{Link}(W \cdot x) = y \tag{2}$$

- x: Input features
- W: Linear transformation
- y: Output / target
- Link(x): Warping function

1. If x has dim 50 and W projects to dimension 100, what is the shape of W?
   - W is a 100×50 matrix
2. If W is learnable, how many parameters does W have?
   - $100 \times 50 = 5000$ parameters

Notes: **How does a generalized linear model make a prediction?** By either mapping to a line or separating data by a line (hyperplane)

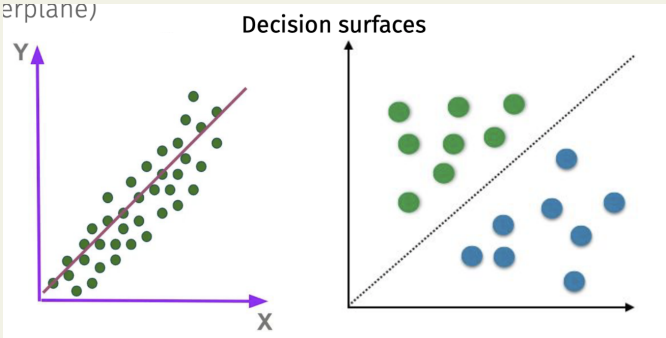

Figure 2:

Notes: **What can we do when the data cannot be separated by a line?** Resort to different decision surfaces.

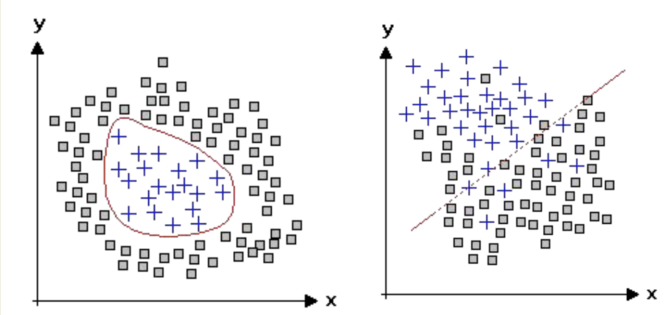

Figure 3:

0.1.4 Representations

Notes:

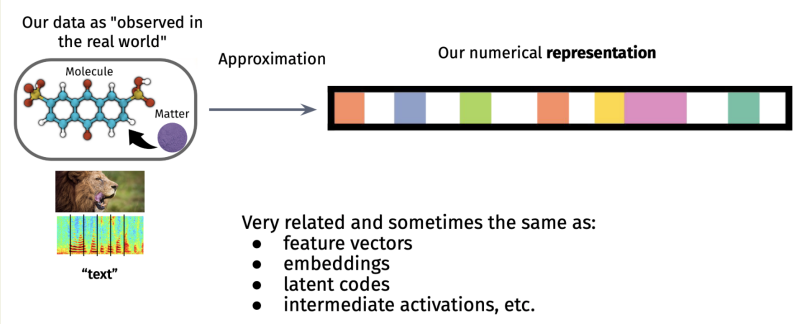

Figure 4:

0.2 Neural networks

Definition: Learnable (optimizable) transformations of data.

$$x \overset{\text{Model}}{\mapsto} y \tag{3}$$

0.2.1 2-layer MLP

Definition:

$$\text{Link}(W_2 \cdot \text{relu}(W_1 \cdot x)) = y \tag{4}$$

- **E** (x and y)
- **P** ? mean squared error
- **T** ? Predict y from x
- **M** ? Neural net (W1, W2)
- **learn?** gradient descent

0.2.2 Geometric intuition

0.2.3 Encoder-Decoder view