1 Supervised learning

Method where you train the program by feeding the learning algorithm with a mapping of inputs to correct outputs.

1.1 Regression

Regression is curve fitting: learn a continuous input $\rightarrow$ output mapping from a set of examples.

1.2 Classification

Outputs are discrete variables (category labels). Learn a decision boundary that separates one class from the other. Generally, a confidence is also desired, i.e., how sure are we that the input belongs to the chosen category.

1.3 Training set

The training set is a set of m $(X,\, y)$ pairs, where:

$$\begin{aligned} X \in \mathbb{R}^d &\quad \text{models the input.} \\ y \in \{0, 1\} &\quad \text{models the output.} \end{aligned}$$

1.4 Error function

The error function for a model $f : X \mapsto y$ parameterized by W applied to a dataset $\{(X,\, y)\}$ of size m is:

$$\min_W \sum_{i=1}^{m} \left(f_W(X_i) - y_i\right)^2$$

1.5 Perceptron

Perceptron is the trivial neural network. The model for a parameter $W = (\text{threshold},\, w_1,\, \ldots,\, w_d)$ and inputs of the form $(1,\, x_1,\, \ldots,\, x_d)$ is given by

$$f_W(X) = \text{sign}(W^\top X)$$

If x_i is evidence for approval, then w_i should be high.
If x_i is evidence for denial, then w_i should be low.

1.5.1 Learning algorithm

The learning algorithm of the Perceptron is quite simple. The learning rate $\in (0,\, 1]$ is used to scale each step. the For a training set $S = \{\, (X_1,\, y_1),\;\; (X_1,\, y_1),\;\; \ldots \}$

- Starting with random weights, show each sample in sequence repetitively.
- If the output is correct, do nothing.
- If the produced output is negative, and the correct output is positive, increase the weights.
- If the produced output is positive, and the correct output is negative, decrease the weights.
- The amount to increase/decrease is given by the current sample scaled by the learning rate.

1.6 Error

The error function for a model f in a **training** sample is

$$E_{\text{in}}(f)$$

This function is known and calculable.

The error function for a model f in a **test** sample is

$$E_{\text{out}}(f)$$

This function is **not** known, and only **approachable**.

Given a model f in a set of M models, the bound for the probability of the error deviation surpassing a given ϵ is

$$\mathbb{P}\left(\left|E_{\text{in}}(f) - E_{\text{out}}(f)\right| > \epsilon\right) \leq 2Me^{-2N\epsilon^2}$$

Notably, $E_{\text{in}}(f)$ and $E_{\text{out}}(f)$ deviates as f becomes complex.

2 Reinforcement learning

Method where you train the program by rewarding the learning algorithm positively or negatively according to the produced results. This method is similar to how we teach animals.

3 Unsupervised learning

Given only inputs as training, find a pattern: discover clusters, manifolds, embedding.