Foundations of Machine Learning AI2000 and AI5000

FoML-17
Perceptron

Dr. Konda Reddy Mopuri
Department of AI, IIT Hyderabad
July-Nov 2025

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

So far in FoML

- Intro to ML and Probability refresher
- MLE, MAP, and fully Bayesian treatment
- Supervised learning
    a. Linear Regression with basis functions (regularization, model selection)
    b. Bias-Variance Decomposition (Bayesian Regression)
    c. Decision Theory - three broad classification strategies
        - Probabilistic Generative Models - Continuous & discrete data
        - (Linear) Discriminant Functions - least squares solution

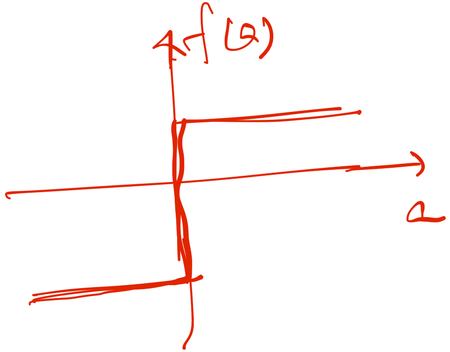
The Perceptron

The Perceptron Algorithm

- Input: $x \in \mathbb{R}^D$
- Targets (2 classes): $t \in \{C_1, C_2\} \rightarrow \{-1, +1\}$
- Prediction: $y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x}))$

Activation function f(a)

$$f(a) = \begin{cases} 1 & a \geq 0 \\ -1 & a < 0 \end{cases}$$



The Perceptron Algorithm

- Class decisions:
  - Assign x to C_1 if: $w^T\phi(\underline{x}) \geq 0$
  - Assign x to C_{-1} if: $w^T\phi(\underline{x}) < 0$

$t_n = \cancel{+1} \quad w^T\phi(\underline{x}_n) > 0$

$t_n = -1 \quad w^T\phi(\underline{x}_n) < 0$

- Criterion for correct classification:

$t_n \cdot w^T\phi(\underline{x}_n) > 0$

The Perceptron Algorithm

- The loss (perceptron criterion):

$$E_P(\mathbf{w}) = \sum_{n \in M} -t_n w^T \phi(x_n)$$

Misclassified samples

It is easy to see that the error function just counts the number of misclassified samples.

$\Rightarrow$ error minimization means reducing the no. of misclassified samples

Perceptron learning: SGD

$$E_P(\mathbf{w}) = \sum_{n\in\mathcal{M}} \mathbf{w}^T\phi(\mathbf{x}_n)t_n$$
$$= \sum_{n\in\mathcal{M}} E_n(\mathbf{w})$$

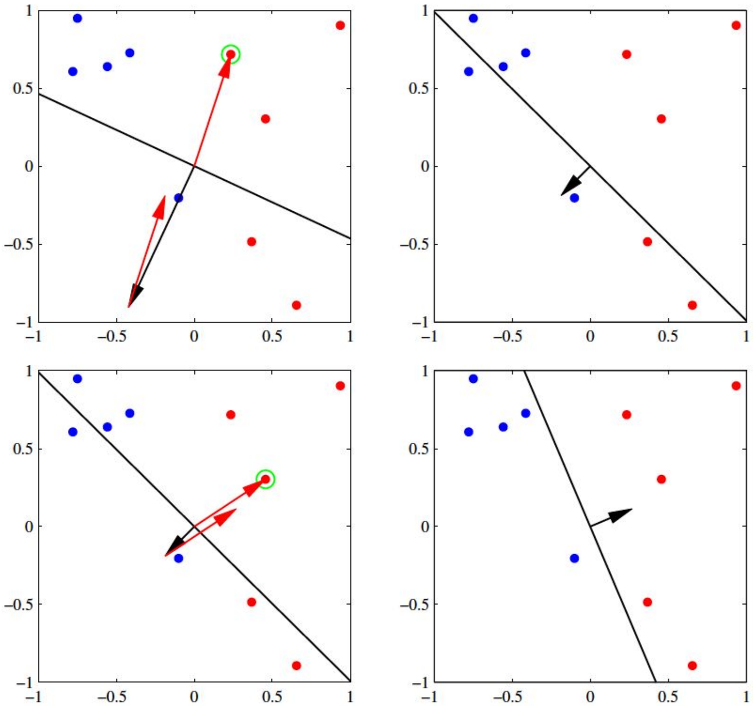
SGD: for each misclassified sample x_n:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta\left[t_n\,\phi(x_n)\right]$$

$$= \underbrace{-t_1\, w^T\phi(x_1)}_{E_1} + t_3\, \underbrace{w^T\phi(x_3)}_{E_3} + t_{33}\, \underbrace{w^T\phi(x_{33})}_{E_{33}} + \cdots$$

Ideal loss function to run SGD, because only one term needs to be derived per misclassified sample

Perceptron learning: SGD



If data is linearly separable, perceptron converges

Perceptron - Issues

- Works only for 2 classes
- More than one solutions
    - Initialization and the order in which the data is presented
- Will not converge if the dataset is not linearly separable
- Need to define basis functions
    - This is the case for all the methods that we discussed so far