Laboratório 5: Perceptron

O objetivo deste trabalho é estudar uma implementação de diferentes algoritmos do Perceptron e avaliar a sua eficiência como classificador. Um template de script acompanhado com funções de visualização é disponibilizado.

1 Definições e notações

Consideramos o conjunto de dados, $D = (x^n, y^n)_{n=1,\cdots,N}$ onde $x^n \in \mathbb{R}^I$ e $y^n \in \{-1, 1\}$. Para cada x, definimos a sua extensão $\widetilde{x} \in \mathbb{R}^{I+1}$ tal como $\widetilde{x}^T = (1, x^T)$. Um perceptron é caracterizado pela função de classificação

$$x \in \mathbb{R}^I \to \widehat{y}(x; \widetilde{w}) = \text{sign}(\widetilde{w}^T \widetilde{x})$$

onde $\widetilde{w} \in \mathbb{R}^{I+1}$ são os parâmetros do classificador. Um evento (x^m, y^m) é bem classificado com os parâmetros $\widetilde{w}$ se $\widehat{y}(x^m; \widetilde{w})y^m = \widehat{y}^m y^m = 1$, seja ainda $\widetilde{w}^T \widetilde{x}^m y^m > 0$.

Para avaliar a qualidade do classificador $\widehat{y}(x; \widetilde{w})$ com a base de dados D, introduzimos a função custo

$$E(\widetilde{w}; D) = \frac{1}{N} \sum_{n=0}^{N} \frac{1}{2} |y^n - \widehat{y}(x^m; \widetilde{w})| = \frac{1}{N} \sum_{n=0}^{N} \max(0, -y^n \widehat{y}^m)$$

A situação ideal consiste em determinar um conjunto de $\widetilde{w}^\star$ tal como $E(\widetilde{w}^\star; D) = 0$. Infelizmente, uma tal situação é possível apenas se o conjunto D é linearmente sepáravel. Caso contrario, procuramos

$$\widetilde{w}^\star = \arg \min_{\widetilde{w} \in \mathbb{R}^{I+1}} E(\widetilde{w}; D)$$

Vamos determinar uma aproximação do vetor $\widetilde{w}^\star$ usando o algoritmo do perceptron e as suas variantes.

2 Implementação das funções

A construção do script é baseada em 3 funções

- `predictor(x,ew)` que calcula o predictor $\widehat{y}(x; \widetilde{w})$ e retorna o sinal;
- `cost(X,Y,N,ew)` que calcula o custo $E(\widetilde{w}; D)$ onde `X` e `Y` agrupam respetivamente os dados (x^n) e (y^n), $n = 1, \cdots, N$;
- `update(x,y,eta,ew)` que retorna $w(t+1)$ calculado com $w(t)$, x e y. O parâmetro η representa a taxa de aprendizagem.

Implementar as 3 funções e experimentar com a base de dados correspondente ao AND, o OR e o XOR.

3 Época e método estocástico

Vamos realizar a aprendizagem de duas maneiras diferentes.

1) Implementar uma função `run_epoch(X,Y,N,eta,MAX_EPOCH,ew,err)` que vai percorrer toda a base de dados até `MAX_EPOCH` para determinar $\widetilde{w}^{\star}$. Uma época consiste em realizar o procedimento de `update` usando os (x^n, y^n),$n = 1, \cdots, N$. Acabamos o ciclo de aprendizagem se o custo $E(\widetilde{w}; D)$ é nulo ou se percorremos o número máximo de época. A função retorna o último $\widetilde{w}$ calculado.

2) Implementar uma função `run_stocastic(X,Y,N,eta,MAX_ITER,ew,err)` que vais escolher até `MAX_ITER` elementos (x^m, y^m) da base de dados aleatoriamente. O ciclo de aprendizagem acaba quando o custo $E(\widetilde{w}; D)$ é nulo o se percorremos o numéro máximo de iterações. A função retorna o último $\widetilde{w}$ calculado.

3) Experimentar as duas técnicas usando as bases de dados simples fornecidas no template. Ajustar os valores de η que permitem uma aprendizagem rápida. O que se passa quando a base de dados não é linearmente separável?

4 *In-samples* e *Out-samples* erros

O segundo ficheiro template permite carregar duas bases de dados:

- `line1500.txt` para a classificação de linhas verticais e horizontais;
- `digits.txt` para classificar os números dígitos. Funções de visualizações são também disponibilizadas
  1) Implementar de novo (copy-paste) as funções necessárias para a classificação dos dados.

2) Dividir a base de dados numa base para o *training* (80%) e uma base para a *validation.*

3) Experimentar diferentes opções de aprendizagens entre épocas e estocásticas. Determinar os valores η que permitem de reduzir os erros.

4) Um estratégia usual consite em treinar 1000 vezes (por exemplo) com um certo valor de η e depois treinar mais 1000 vezes com um valor mais baixo. Esta estratégia estende-se a repetir 5 ou 6 este procedimento reduz o valor de η.

5) No final, usando o algoritmo estocástico, definir uma função de η em função do número de iterações para melhorar a convergência.