

Universidad Simón Bolívar
Departamento de Cómputo Científico

Tarea II: Perceptrón

Georvic Tur - 12-11402

30/05/2017

1. Para los datos en cuatro dimensiones disponible en la página web del curso busque un clasificador lineal con perceptrones. Indique la red utilizada, características, tipo de aprendizaje, constantes del algoritmo utilizadas, número de iteraciones empleadas, etc.

Solución:

A continuación se muestra la salida de un programa implementado en Python. Dicho programa usa las librerías *Pandas*, *NumPy* y *Scikit-learn*. Para resolver el problema del enunciado, se implementó un perceptrón multiclase basado en aprendizaje supervisado competitivo cuya regla de aprendizaje es la actualización simple con constante de aprendizaje. La matriz de pesos se inicializa de manera aleatoria con valores entre cero y uno. No se usó normalización de los datos iniciales pues se detectó que esto no mejoraba el desempeño del modelo.

La razón por la cual es supervisado es que se usan los valores reales de las clases para el entrenamiento [1]. Por su parte, la razón por la cual es competitivo es que sólo se actualiza el vector de pesos cuyo producto interno con una instancia sea el mayor para dicha instancia [2].

Para evaluar este modelo, se usó *3-fold cross-validation* y se mostró la exactitud promedio al final de las iteraciones. Adicionalmente, se compara el rendimiento de este modelo con el rendimiento del perceptrón implementado por la librería *Scikit-learn*.

TASA DE APRENDIZAJE: 0.5

NÚMERO DE ITERACIONES: 5

CROSS-VALIDATION fold: 0

Vectores de pesos por cada clase

	X1	X2	X3	X4	BIAS	CLASE
0	-0.669749	1.803158	0.306217	-0.216690	0.5	1
1	0.179322	0.308381	0.667268	-1.147142	-1.0	2
2	0.547716	-0.343539	0.248083	0.826460	0.0	3
3	1.805880	0.255773	-0.000367	1.836275	0.5	4

CORRECTOS: 4

INCORRECTOS: 130

SCORE DE MI PERCEPTRON: 0.029850746268656716

```

PERCEPTRON DE SKLEARN
SCORE DE SKLEARN: 0.373134328358209

CROSS-VALIDATION fold: 1
Vectores de pesos por cada clase
      X1      X2      X3      X4  BIAS  CLASE
0  0.657338  0.391800  1.241209  1.203816   1.5    1
1  0.152310 -0.207302 -0.034343  0.205680   1.5    2
2  1.322330  1.804836  1.306645  1.484614  -1.5    3
3 -0.884719 -0.348804 -0.705287 -1.229000  -1.5    4
CORRECTOS: 130
INCORRECTOS: 3
SCORE DE MI PERCEPTRON: 0.9774436090225563
PERCEPTRON DE SKLEARN
SCORE DE SKLEARN: 0.5864661654135338

CROSS-VALIDATION fold: 2
Vectores de pesos por cada clase
      X1      X2      X3      X4  BIAS  CLASE
0  1.135518  0.664156  1.221425  0.552969   3.0    1
1 -0.084453 -0.238174 -0.017304 -0.228541   3.0    2
2  2.378076  3.254983  2.170754  3.077546  -3.5    3
3 -1.457619 -1.963226 -1.494303 -1.289621  -2.5    4
CORRECTOS: 133
INCORRECTOS: 0
SCORE DE MI PERCEPTRON: 1.0
PERCEPTRON DE SKLEARN
SCORE DE SKLEARN: 0.6240601503759399

EXACTITUD PROMEDIO DE MI PERCEPTRON: 0.6690981184304045
EXACTITUD PROMEDIO DE SKLEARN: 0.5278868813825609

```

2. Para los mismos datos de la pregunta anterior busque un clasificador lineal mediante un aprendizaje por reforzamiento.

Solución:

Para resolver el problema del enunciado, se ha implementado una red basada en la arquitectura propuesta en detalle por Sanger [5] y mencionada por [4]. La misma se basa en el algoritmo de Oja aplicado en repetidas ocasiones de manera secuencial para calcular componentes principales múltiples [4]. Los datos iniciales fueron normalizados.

El tipo de aprendizaje que usa es no supervisado, puesto que no se tiene acceso a los valores de las clases durante el entrenamiento del modelo [6]. Adicionalmente, este tipo de

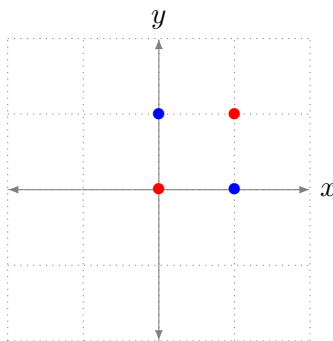
aprendizaje es por reforzamiento, pues cada entrada produce un reforzamiento de los pesos de la red [7].

Para evaluar los resultados se usaron las matrices retornadas por el algoritmo para reconstruir los datos iniciales. A partir de esto fue posible definir el error cuadrático de la diferencia de dichas matrices. Dicho error cuadrático se colapsó a lo largo de filas y columnas tomando la raíz cuadrada en cada caso, lo cual dio como resultado: 30.86.

3. El perceptrón fue utilizado en clase sobre la función lógica O. Demuestre que las clases de la función lógica del O-exclusivo no son linealmente separables. (Ayuda: Reducción al absurdo) Puede funcionar el algoritmo del perceptrón para separar estos datos?

Solución:

La función que se desea aprender tiene la forma:



Las dos clases posibles en este problema son P y N: positivos y negativos con $(0, 1), (1, 0) \in P$ y $(0, 0), (1, 1) \in N$

Supongamos que este conjunto de cuatro puntos es separable linealmente. Esto significa que existe un vector w que define una recta en el plano y dicha recta separa a los puntos de acuerdo a sus clases. Como dicha recta no necesariamente pasa por el origen, hemos de aumentar el vector w con un sesgo b .

A partir de esto, obtenemos las siguientes ecuaciones.

$$\begin{array}{ll} w_0 * 0 + w_1 * 1 + b > 0 & ((0,1) \text{ está en } P) \\ w_0 * 1 + w_1 * 0 + b > 0 & ((1,0) \text{ está en } P) \\ w_0 * 0 + w_1 * 0 + b \leq 0 & ((0,0) \text{ está en } N) \\ w_0 * 1 + w_1 * 1 + b \leq 0 & ((1,1) \text{ está en } N) \end{array}$$

Simplificando estas ecuaciones, tenemos que:

$$w_1 + b > 0 \quad (1)$$

$$w_0 + b > 0 \quad (2)$$

$$b \leq 0 \quad (3)$$

$$w_0 + w_1 + b \leq 0 \quad (4)$$

Multiplicando a la inecuación 3 por (-1) y sumando (-b) tanto a la inecuación 1 como a la 2, tenemos que:

$$w_1 > 0 \quad (1)$$

$$w_0 > 0 \quad (2)$$

$$b \leq 0 \quad (3)$$

$$w_0 + w_1 + b \leq 0 \quad (4)$$

Aplicando la inecuación 3 en la 4, tenemos que:

$$w_1 > 0 \quad (1)$$

$$w_0 > 0 \quad (2)$$

$$b \leq 0 \quad (3)$$

$$w_0 + w_1 \leq 0 \quad (4)$$

Sin embargo, las inecuaciones 1 y 2 contradicen a la inecuación 4. Como lo único que se asumió fue que existía una recta que separa estos puntos y se ha llegado a una contradicción, esto no puede ser cierto.

Referencias

- [1] Simon Haykin, *Neural networks and learning machines, 3rd ed.* Upper Saddle River: Pearson Education, 2009. p, 34.
- [2] Raúl Rojas, *Neural Networks: a Systematic Introduction, 1st ed.* Berlin: Springer, 1996, pp. 104-105 [ONLINE]. Disponible en: <https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>. [Visito: 30/05/2017]
- [3] Simon Haykin, *Neural networks and learning machines, 3rd ed.* Upper Saddle River: Pearson Education, 2009. p, 369.

- [4] Raúl Rojas, *Neural Networks: a Systematic Introduction, 1st ed.* Berlin: Springer, 1996, pp. 114-119 [ONLINE]. Disponible en: <https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>. [Visto: 30/05/2017]
- [5] T. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network", *Neural Networks*, vol. 2, no. 6, pp. 459-473, 1989. [ONLINE]. Disponible en: <https://pdfs.semanticscholar.org/709b/4bfc5198336ba5d70da987889a157f695c1e.pdf>. [Visto: 30/05/2017]
- [6] Simon Haykin, *Neural networks and learning machines, 3rd ed.* Upper Saddle River: Pearson Education, 2009. p, 37.
- [7] Raúl Rojas, *Neural Networks: a Systematic Introduction, 1st ed.* Berlin: Springer, 1996, pp. 101 [ONLINE]. Disponible en: <https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>. [Visto: 30/05/2017]