

PRÁCTICA EVALUABLE – 4.1 – Trabajo con un perceptrón


Antonio Gallego Peñalver

Ejercicios sobre el perceptrón:

Descarga el cuaderno de Jupiter practica4.1_perceptron.ipynb y ejecuta las celdas para crear un perceptrón y aplicarlo al dataset iris.data. Después de ejecutarlo, contesta a las siguientes preguntas:

1. ¿Qué representa el atributo `w_`? ¿Y el atributo `w_[0]`?
 - `w_`: es el vector de pesos sinápticos del perceptrón. Se utiliza para ponderar las características de entrada.
 - `w_[0]`: es el peso asociado al bias. Permite desplazar la función de activación, para que la frontera de decisión no se vea forzada a pasar por el origen.
2. El atributo `w_` se inicializa con las sentencias:


```
rgen = np.random.RandomState(semilla)
self.w_ = rgen.normal(loc=0.0, scale=0.01, size=1 + X.shape[1])
```

 ¿Qué significado tienen estas sentencias?
 - `rgen`: Es un generador de números aleatorios que utiliza la semilla que se le pasa. Hace que los números generados sean siempre los mismos cuando se use la misma semilla, lo que permite reproducir los resultados del entrenamiento.
 - `self.w`: Se usa para inicializar el vector de pesos con valores aleatorios extraídos de una distribución normal, con una media de 0.0 y una desviación estándar de 0.01.
3. ¿Qué misión tiene el método `net_input(self, X)`? ¿Qué valores tenemos que enviar en el parámetro `X`?
 - Se usa para calcular la entrada neta al perceptrón. Se envía como valores un array de las características de entrada (longitud de pétalo y sépalo del dataset de Iris).
4. ¿Qué valor retornaría el método `net_input()` si le pasáramos los valores [2.1,4.2]? Crea una celda en el cuaderno y pega el resultado de la ejecución? ¿Qué significa el número entero resultado?
 - Devuelve 0.5712133553549159. Indica la combinación lineal de pesos y entradas
5. ¿Qué misión tiene el método `predict(self,X)`? ¿Qué predice el método si le pasamos los valores [2.1,4.2]?
 - Devuelve la etiqueta de clase para una muestra, basándose en el valor obtenido en el método `net_input()`.
 - Al pasar dichos valores, se calcula la entrada (0.5712133553549159). Al ser mayor que 0, al aplicar `predict()` retorna 0.
6. ¿Qué relación tiene el método `predict` con la función umbral (también llamado escalón unitario) ?
 - El método `predict()` aplica la función umbral a la suma ponderada de las entradas.

PRÁCTICA EVALUABLE – 4.1 – Trabajo con un perceptrón

Antonio Gallego Peñalver

7. Modifica el método fit de la clase Perceptron para mostrar, en las distintas épocas del entrenamiento, dónde se producen errores de clasificación y cómo se modifica el vector de pesos sinápticos para aprender. Al entrenarse, el perceptrón debería emitir cadenas de texto como las siguientes:

```
Valor inicial [ 0.01624345 -0.00611756 -0.00528172]
época=0; xi=[7. 4.7]; y=1; predicción=-1; Errores=1; Pesos=[0.21624345 1.39388244 0.93471828]
época=1; xi=[5.1 1.4]; y=-1; predicción=1; Errores=1; Pesos=[0.01624345 0.37388244 0.65471828]
época=1; xi=[4.9 1.4]; y=-1; predicción=1; Errores=2; Pesos=[-0.18375655 -0.60611756 0.37471828]
época=1; xi=[7. 4.7]; y=1; predicción=-1; Errores=3; Pesos=[0.01624345 0.79388244 1.31471828]
época=2; xi=[5.1 1.4]; y=-1; predicción=1; Errores=1; Pesos=[-0.18375655 -0.22611756 1.03471828]
época=2; xi=[4.9 1.4]; y=-1; predicción=1; Errores=2; Pesos=[-0.38375655 -1.20611756 0.75471828]
época=2; xi=[7. 4.7]; y=1; predicción=-1; Errores=3; Pesos=[-0.18375655 0.19388244 1.69471828]
época=3; xi=[5.1 1.4]; y=-1; predicción=1; Errores=1; Pesos=[-0.38375655 -0.82611756 1.41471828]
época=3; xi=[5.7 3.5]; y=1; predicción=-1; Errores=2; Pesos=[-0.18375655 0.31388244 2.11471828]
época=4; xi=[5.1 1.4]; y=-1; predicción=1; Errores=1; Pesos=[-0.38375655 -0.70611756 1.83471828]
```

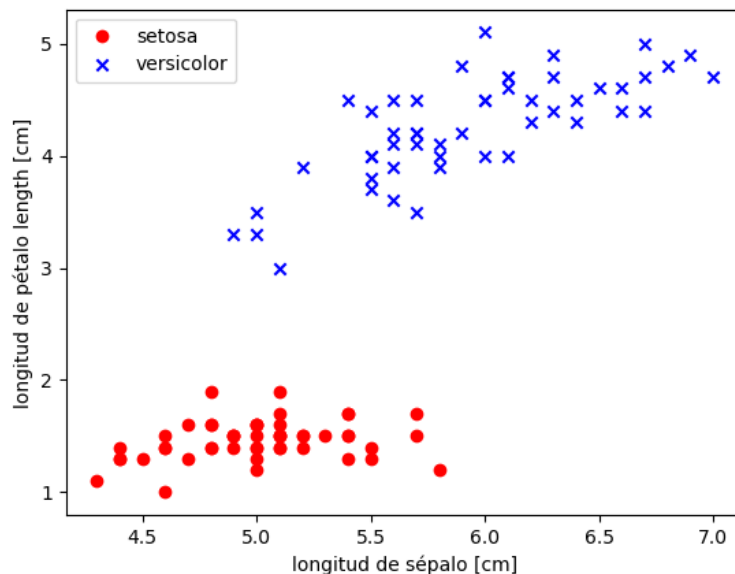
¿A partir de qué época el perceptrón clasifica todas las muestras correctamente?

- A partir de la época 5

8. ¿Qué representa el valor ratio definido en el constructor?

- Controla cuánto se ajustan los pesos sinápticos en cada actualización cuando el perceptrón comete un error. Si el ratio es alto, los pesos se ajustan más rápido. Mientras que si el ratio es bajo, el aprendizaje será mas estable pero tardará en encontrar una solución ópt

Antonio Gallego Peñalver

9. Interpreta geoméricamente el sentido del vector w_- :

Las muestras están claramente separadas por una línea recta.

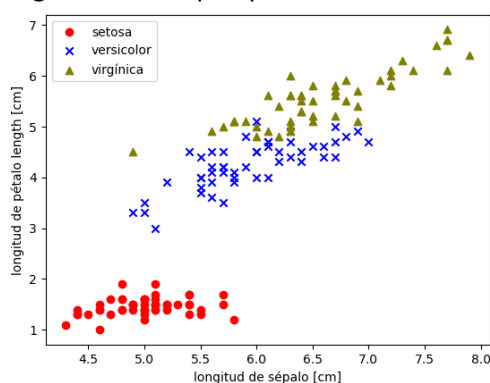
Teniendo en cuenta que la entrada a la red (net_input) está definida como $w_0 + longitud_sépalo * w_1 + longitud_pétalo * w_2$ y que en la figura el eje X está representada la longitud del sépalos y en el eje Y está representada la longitud del pétalo, responde a las siguientes preguntas:

¿Cuándo tendrá “dudas” el perceptrón de si una flor pertenece a una clase u otra?

- Cuando su entrada a la red sea cercana a 0.

Genera valores de longitud de pétalo y longitud de sépalos de manera que la entrada a la red produzca un valor de 0. Muestra estos valores generados en una gráfica junto con las variedades setosa y versicolor, para ver dónde está el límite de la decisión del perceptrón.

10. El ejemplo que encuentras en el cuaderno, sirve para diferenciar las flores ‘Iris-setosa’ de la versicolor, que muy bien separadas linealmente. Muestra en un gráfico matplotlib todas las muestras de setosa, versicolor y virgínica. Tiene que quedarte similar a este:



PRÁCTICA EVALUABLE – 4.1 – Trabajo con un perceptrón

Antonio Gallego Peñalver

¿Son linealmente separables las variedades versicolor y virgínica?

- No, no son linealmente separables.

11. Añade una celda para entrenar otro perceptrón (p2) con las variedades versicolor y virgínica y ejecuta predicciones sobre estas dos instancias de flores:

Pétalo=5, sépalo=5.5

Pétalo=6, sépalo=6.5

Interpreta los resultados

12. ¿Qué superficies de decisión se muestran utilizando las variedades versicolor y virgínica?

LÍMITES DEL PERCEPTRÓN:

Cambia los valores X e y del cuaderno para que, en lugar de cargar los datos de las flores iris, asignes los valores de la función XOR \oplus (en Python es el operador ^).

Por ejemplo:

X0	X1	Y=X0 \oplus x1
1	1	0
0	1	1
1	0	1
0	0	0

Entrena el perceptrón con esos 4 datos, evalúa su convergencia y predice los resultados. ¿Es fiable el perceptrón entrenado para este caso? ¿Por qué?