

Fundamentos de Aprendizaje Automático y Reconocimiento de Patrones

Práctico 02

2024

Agenda

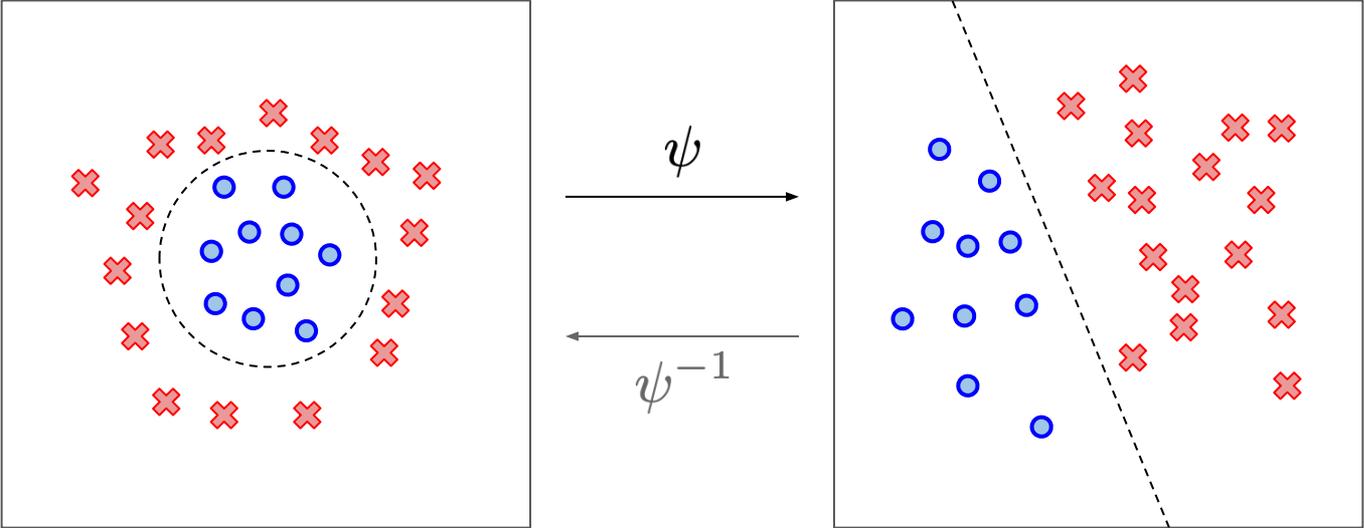
Miércoles 27

- Algoritmo pocket
- Transformaciones no lineales
- Clasificación de imágenes de dígitos

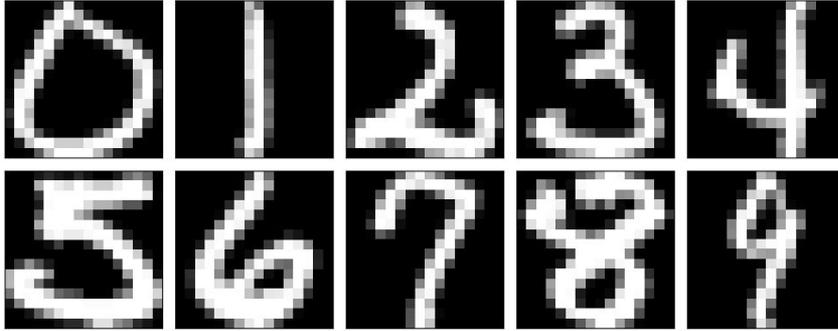
Algoritmo pocket

- Variante del Perceptrón
- Las iteraciones y regla de actualización es la misma, pero...
- Solo se actualiza el vector de pesos si el error que se obtiene es menor que el obtenido hasta el momento.
- Algo más robusto si los datos no son perfectamente linealmente separables.
- No garantiza la convergencia a una solución perfecta en casos de datos no linealmente separables
- Mejora en la estabilidad y desempeño comparado con el Perceptrón estándar.

Transformación no lineal



Clasificación de dígitos



- USPS dataset, dígitos escritos a mano.
- 7921 de entrenamiento y 2007 de test.
- Imágenes $I(m,n)$ de 8 bits de 16x16 píxeles en [0,1]:
 - matriz de 16 filas y 16 columnas
 - 256 posibles valores en cada pixel
 - los valores van entre 0 y 255

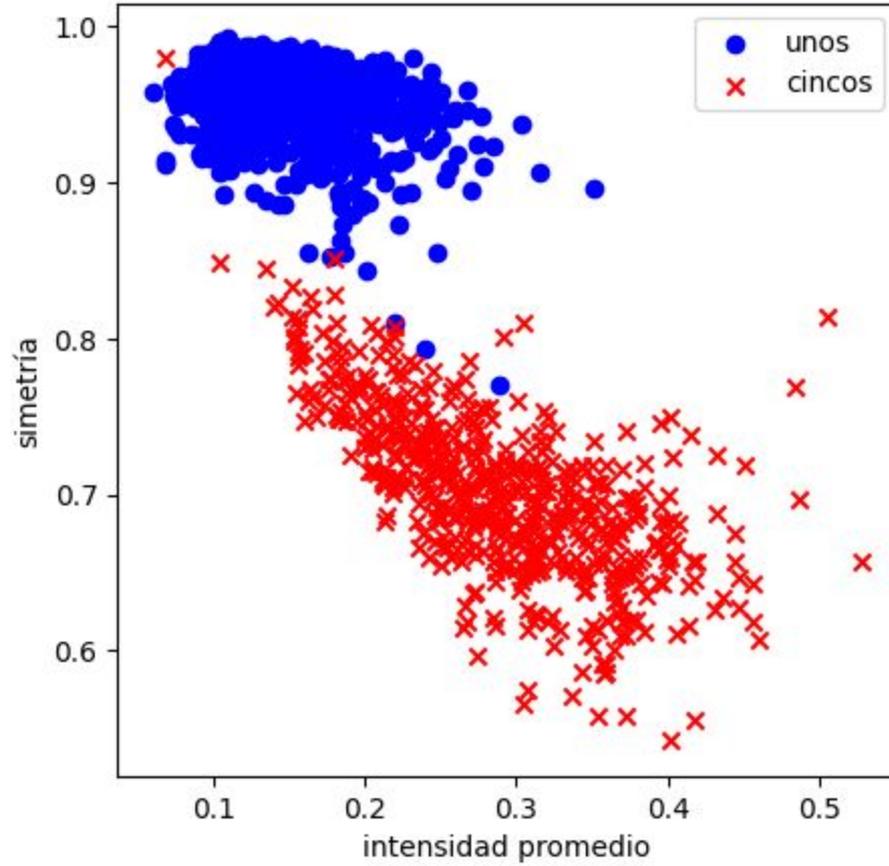
$$\text{intensidad media} = \frac{1}{256} \sum_{m=0}^{15} \sum_{n=0}^{15} I(m,n)$$

$$\begin{aligned} \text{simetría} &= 1 - \frac{1}{256} \sum_{m=0}^{15} \sum_{n=0}^{15} \frac{|I(m,n) - I(m,15-n)| + |I(m,n) - I(15-m,n)|}{2} \\ &= \frac{\text{simetría horizontal} + \text{simetría vertical}}{2} \end{aligned}$$

DESCRIPTORES O
CARACTERÍSTICAS

```
1 def caracterizar_digitos(digitos):
2     """
3     Genera las características que se utilizarán para clasificar los dígitos
4     Entrada:
5     digitos: arreglo de (N,256) que contiene N dígitos
6     Salida:
7     features: arreglo de (N,2) con las características calculadas. En principio se
8     sugiere calcular la intensidad promedio y una medida de simetría,
9     pero podría calcularse alguna otra medida.
10    """
```

Clasificación de dígitos



Cuestionario práctico

passwd: 8734

Evaluación interactiva

Evaluación interactiva

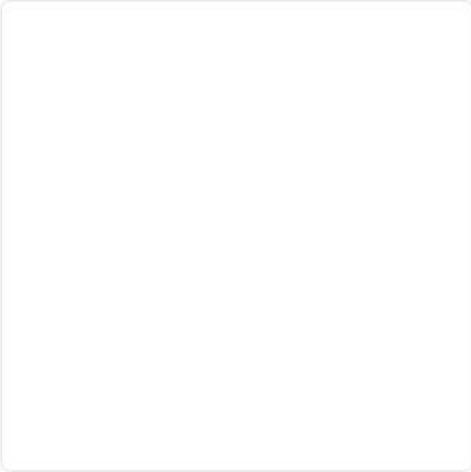
A continuación se plantea realizar una evaluación interactiva del funcionamiento del clasificador de dígitos. Se reconocerán los dígitos dibujados sobre el *sketchpad* de la biblioteca *gradio*.

La función `preprocesar()` provista en la siguiente celda, realiza la conversión de las imágenes generadas con el *sketchpad* al formato de los datos de entrenamiento. A diferencia de los datos de entrenamiento, los dígitos en el *sketchpad* de *gradio* son negros sobre fondo blanco, tomando el blanco el valor 0 y el negro el 255.

```
1 def preprocesar(img):  
2     # Se reescala la imagen al tamaño de los dígitos de entrenamiento  
3     img = zoom(img, 16 / img.shape[1])  
4     # Se normaliza de [0, 255] -> [0, 1]  
5     # img = img / 255  
6     img = img / np.max(img[:])  
7     # Se invierten los valores (0,1) -> (1,0)  
8     # img = 1 - img  
9     img = img.reshape(1, 16 * 16).astype(float)  
10    return img
```

Evaluación interactiva

img



Layer 1

Clear Submit

output 0

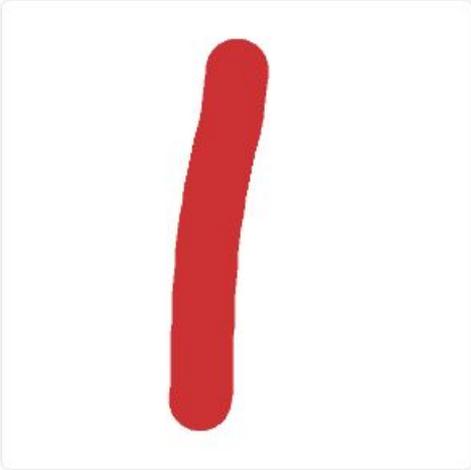
{..}

El garabato se parece a:

Flag

Evaluación interactiva

img



Layer 1

Clear Submit

output 0

```
1 | {
2 |   "intensidad": "0.09359",
3 |   "simetria": "0.90776",
4 |   "score": "0.90897"
5 | }
```

El garabato se parece a:

1

Flag

Evaluación interactiva



img

Layer 1

Clear Submit

The image shows a hand-drawn red number '5' on a white background. The drawing is contained within a canvas interface with a toolbar at the top right (undo, redo, delete) and a layer control at the bottom (Layer 1, zoom, pan, eraser). Below the canvas are 'Clear' and 'Submit' buttons.

```
output 0
```

```
1 | {
2 |   "intensidad": "0.28407",
3 |   "simetria": "0.75421",
4 |   "score": "-1.57673"
5 | }
```

El garabato se parece a:

5

Flag

The right panel displays the output of a classification model. It shows a JSON object with three fields: 'intensidad', 'simetria', and 'score'. Below the JSON is a text prompt 'El garabato se parece a:' followed by the predicted digit '5'. At the bottom is a 'Flag' button.