



Neuronas pythónicas que reconocen caracteres

PythonDay
Rafaela 2012

Agenda

- Objetivo de la charla
- ¿Qué es una red neuronal MLP? ¿Cómo aprende?
- Presentación del ejemplo y esqueleto del proyecto
- Cómo procesar las imágenes para reconocer distintos "objetos" que contenga
- El modelo MLP para reconocer caracteres

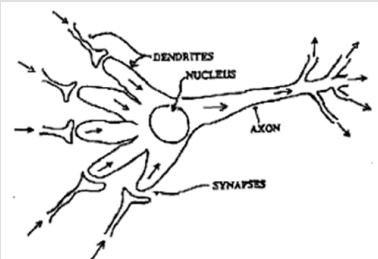
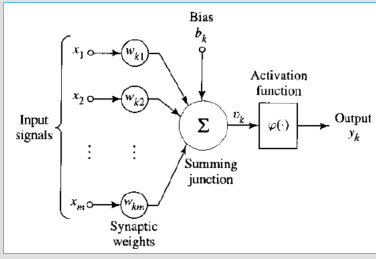
Objetivo

1. Despertar interés sobre redes neuronales
2. Mostrar Scipy -> nos brinda muchas facilidades!!
3. Mostrar como podemos implementar un reconocimiento de caracteres por medio de esta técnica

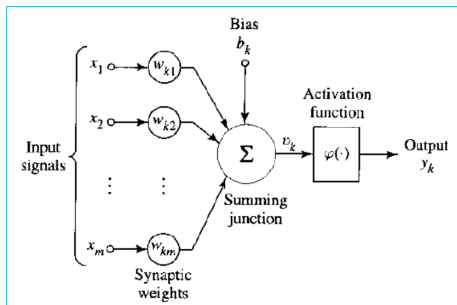
¿Qué es una red neuronal artificial?

Una red neuronal es una gran cantidad de unidades simples de cómputo interconectadas entre si, que tratan de asemejarse al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red desde el ambiente a traves de un proceso de aprendizaje
2. El conocimiento adquirido es almacenado en las conecciones entre neuronas

Neurona biológica	Neurona artificial
	

Modelo matemático de un neurona tipo perceptrón

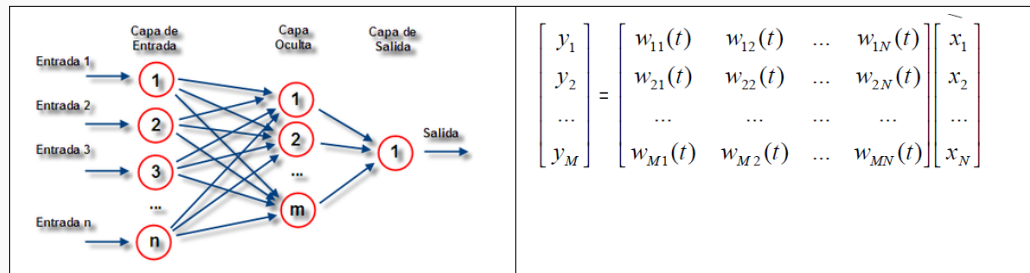


$$u_k = \sum_{i=1}^m w_{kj} x_j$$

$$v_k = u_k + b_k$$

$$y_k = \varphi(v_k)$$

La red neuronal MLP



Entrenamiento para el MLP

- **Entrenamiento supervisado**, tengo patrones de entrenamiento y los salidas deseadas
 - Usamos entrenamiento batch
- **Back Propagation** : descenso por el gradiente del error. El gradiente nos da la dirección de máximo crecimiento del error.

$$W(t+1) = W(t) + \eta \left(- \frac{\partial E(t)}{\partial W(t)} \right)$$

Pasamos al código!!!

Todo bien, pero... ¿cómo usamos esto?

La idea es entrenar una red neuronal para que dada una imagen de un caracter, nos devuelva **el caracter**.

El código es bastante simple :)

```
def convert(image_file):
    img = ip.get_image(image_file)
    lines = []
    for line in ip.get_lines(img):
        words = []
        for word in ip.get_words(img, line):
            chars = []
            for char in ip.get_chars(img, word):
                c = convert_char(img, char)
                chars.append(c)
            words.append(' '.join(chars))
        lines.append(' '.join(words))
    print '\n'.join(lines)
```

Obviamente, el problema es como hacemos get_image, get_line, etc...

Buscando líneas, palabras y caracteres

Los pasos serían:

1. Abrir la imagen y convertir cada pixel a blanco y negro
2. Filtrar un poco la imagen para eliminar ruidos
3. Enumerar los grupos de pixeles que están juntos
4. Buscar los slices que encierran a estos grupos

Para esto, vamos a usar **ndimage**, un paquete de Scipy.

Vamos al código!!

El modelo elegido.

- **Entradas de la red:** ¿Píxeles o características? **Características**
- **Salidas de la red:** 8 binarias. vamos a obtener el ascii del caracter en cuestión
- **Neuronas en la capa oculta:** proporcionales a la cantidad de caracteres distintos a reconocer
- **Funciones de activación:** Tangente hiperbólica en la capa oculta, Linear en la capa de salida

Vamos al código!!

¿Preguntas?

Si me quieren contactar:

- arielrossanigo@gmail.com
- @arielrossanigo

Repo donde podés bajar el código

- https://bitbucket.org/arielrossanigo/neuronaspthonicas_rafaela2012

Referencias bibliográficas

- Haykin, S., Neural Networks: A Comprehensive Foundation, Prentice Hall, 2nd edition (1999)
- R. Rojas, Neural Networks, Springer-Verlag, Berlin, 1996