

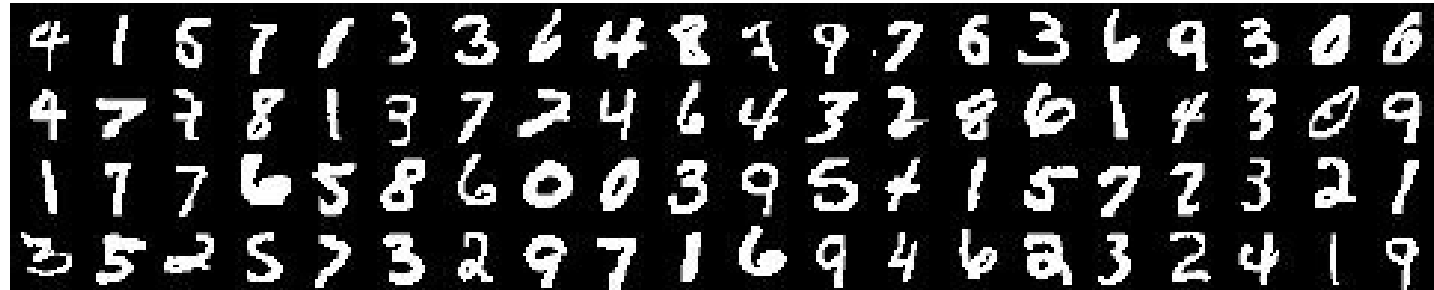


Deep Learning aplicado al análisis de señales e imágenes

P3. REDES NEURONALES EMPLEANDO TF Y MNIST: APRENDIZAJE Y EVALUACIÓN



MNIST



“NIST”: National Institute of Standards and Technology

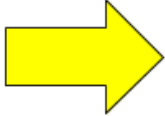
“M” : “modified” (datos preprocesados)

- 60,000 imágenes de training y 10,000 para test
- Imágenes en escala de gris [0,255]
- Cada vector de características es de tamaño 784, correspondiente al nivel de gris de los 28 x 28 pixels de cada imagen.

<http://yann.lecun.com/exdb/mnist/>

Categorical encoding vs one-hot encoding

- La codificación **one-hot encoding** crea un **mapeo binario** de las **etiquetas categóricas** de los datos. Concretamente indica la **presencia** de cada posible valor en los datos originales.



Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Categorical encoding vs one-hot encoding

- Para **transformar etiquetas** que siguen una **codificación categórica a one-hot encoding** haremos uso del objeto **LabelBinarizer** existente en la librería de Machine Learning *scikit learn*.

```
from sklearn import preprocessing

# Definición del grafo
lb = preprocessing.LabelBinarizer()
lb.fit(["azul", "rojo", "amarillo"])

lb.classes_
>>> array(["amarillo", "azul", "rojo"])

lb.transform(["azul", "amarillo"])
>>> array([[0 1 0], [1 0 0]])
```

One-hot encoding MNIST

Entero	Cadena binaria
0	1000000000
1	0100000000
2	0010000000
3	0001000000
4	0000100000
5	0000010000
6	0000001000
7	0000000100
8	0000000010
9	0000000001



Deep Learning aplicado al análisis de señales e imágenes

P3. REDES NEURONALES EMPLEANDO TF Y MNIST: APRENDIZAJE Y EVALUACIÓN

