Herramientas Software para Tratamiento de Imágenes
Máster Oficial en Visión Artificial
Hoja de Problemas Evaluable

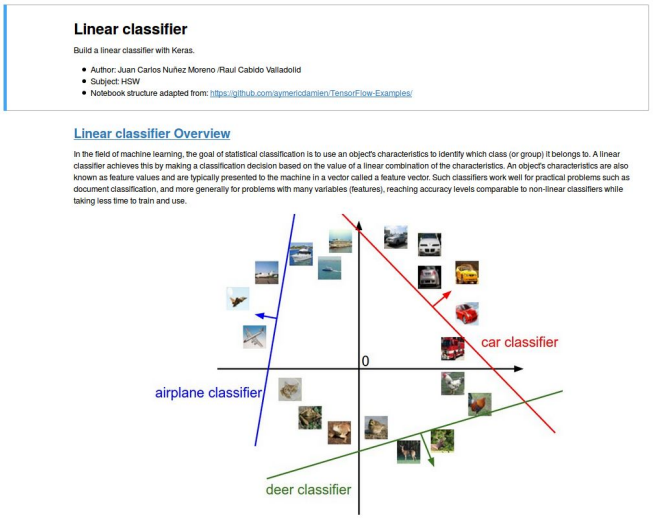
Instrucciones para la entrega

Una vez resueltos los problemas enunciados a continuación, el alumno utilizará el formulario habilitado en el moodle de la asignatura para subir los fuentes en un fichero zip. El nombre del fichero deberá formarse siguiendo el siguiente esquema: “Apellido1_Apellido2_Nombre.zip”.

Fecha límite de entrega: 28 de Noviembre de 2017

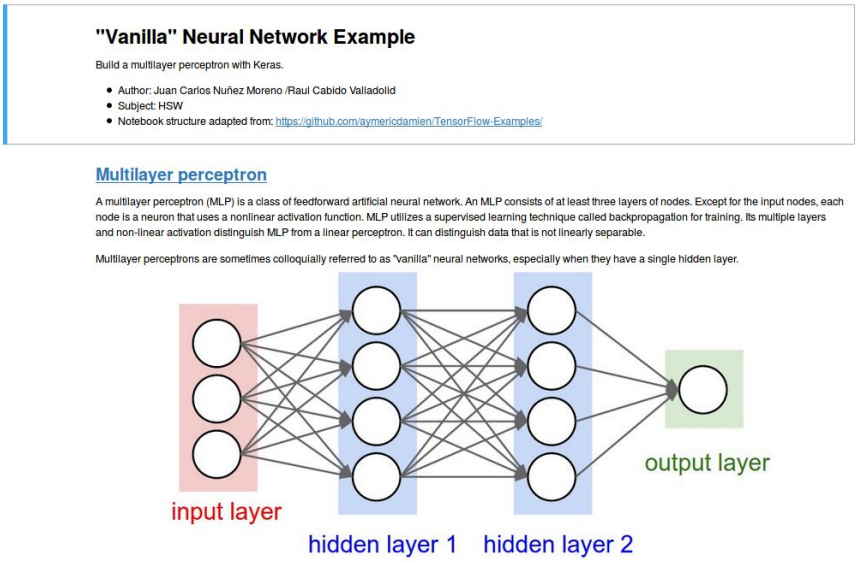
Ejercicio 1

Construir **un clasificador lineal (softmax)** con Keras. Para simplificar la tarea, el profesor pone a disposición de los alumnos un *Jupyter notebook*. La tarea del alumno consistirá en rellenar aquellas partes de código incompletas relacionadas con: la definición del modelo, entrenamiento, evaluación e inferencia. El problema a resolver es el reconocimiento de dígitos manuscritos. Para ello se utilizará el dataset MNIST.



Ejercicio 2

Construir una **red de neuronas** con keras. La arquitectura elegida se corresponde con dos capas ocultas (64 neuronas por capa), más una capa de activación *softmax.* Para simplificar la tarea, el profesor pone a disposición de los alumnos un *Jupyter notebook*. La tarea del alumno consistirá en rellenar aquellas partes de código incompletas relacionadas con: la definición del modelo, entrenamiento, evaluación e inferencia. El problema a resolver es el reconocimiento de dígitos manuscritos. Para ello se utilizará el dataset MNIST.



Ejercicio 3

Construir una **red deconvolución LeNet-5** con keras. La red se constituye de las siguientes capas de convolución:

- CONV2D: 32 filters, (5,5) kernel size
- ACTIVATION: relu
- MAXPOOLING: (2,2) downscale factor
- DROPOUT: 0.25 rate
- CONV2D: 32 filters, (5,5) kernel size
- ACTIVATION: relu
- MAXPOOLING: (2,2) downscale factor
- DROPOUT: 0.25 rate

La última etapa debe corresponderse con una red de neuronas *Fully Connected*, que deberá responder a la siguiente configuración:

- flatten
- capa oculta de 512 neuronas
- ACTIVATION: relu
- softmax

Nuevamente, para simplificar la tarea, el profesor pone a disposición de los alumnos un *Jupyter notebook*. La tarea del alumno consistirá en rellenar aquellas partes de código incompletas relacionadas con: la definición del modelo, entrenamiento, evaluación e

inferencia. El problema a resolver es el reconocimiento de dígitos manuscritos. Para ello se utilizará el dataset MNIST.

Convolutional Neural Network (CNN) Example

Build a LeNet-5 convolutional neural network with Keras.

- Author: Juan Carlos Nuñez Moreno /Raul Cabido Valladolid
- Subject: HSW
- Notebook structure adapted from: https://github.com/aymericdamien/TensorFlow-Examples/

CNN Overview

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

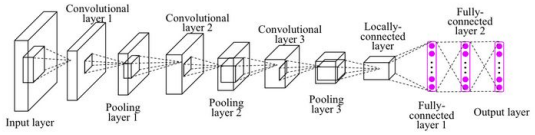
Convolutional networks were inspired by biological processes in which the connectivity pattern between neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

They have applications in image and video recognition, recommender systems and natural language processing.

Lenet-5

LeNet-5, a pioneering 7-level convolutional network by LeCun et al. that classifies digits, was applied by several banks to recognise hand-written numbers on checks digitized in 32x32 pixel images. The ability to process higher resolution images requires larger and more convolutional layers, so this technique is constrained by the availability of computing resources.



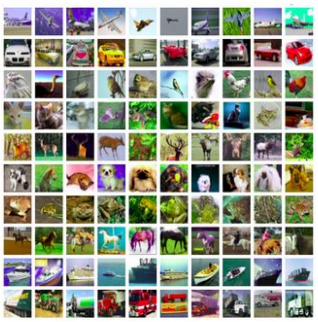
Ejercicio 4

Aplicar la **red deconvolución LeNet-5** al problema de clasificación de imágenes. Para ello haremos uso del dataset CIFAR-10. El profesor pone a disposición de los alumnos un *Jupyter notebook*. La tarea del alumno consistirá en rellenar aquellas partes de código incompletas.

CIFAR-10 Dataset Overview

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.



More info: https://www.cs.toronto.edu/~kriz/cifar.html

In [1]:
```
import keras
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D
```

```
Using TensorFlow backend.
```

Ejercicio 5

Montar un Jupyter notebook similar a los facilitados por el profesor en ejercicios anteriores. Dicho notebook deberá mostrar los pasos necesarios para llevar a cabo la etapa de inferencia/predicción utilizando una de las siguientes redes de clasificación: *AlexNet*, *VGG*, ResNet o *Inception*. Todos lo modelos previamente mencionados se encuentran pre-entrenados y disponibles en keras. Para ello, se pide:

- Generar un pequeño dataset de test con imágenes descargadas de internet (10-20 imágenes)
- Cargar la red seleccionada y hacer predicción sobre el dataset completo
- Por cada imagen mostrar sus tres categorías/clases más probables junto con su probabilidad

Ejercicio 6

Montar un Jupyter notebook que permita comparar la sintaxis de los frameworks TensorFlow (Low level API) y Keras. Tomar como base el Jupyter notebook del Ejercicio 3 y añadir celdas intermedias con el código equivalente para TensorFlow. Nuevamente se quieren cubrir las etapas de definición del modelo, entrenamiento, evaluación e inferencia.

Ejercicio 7

Desarrollar un script en python para realizar detección de objetos utilizando el API de TensorFlow. El script deberá responder a los siguientes opciones de parámetros:

```
object_detection.py --image=image00.jpg
            --model=./ssd_mobilenet_v1_coco_11_06_2017.tar.gz
            --detection=imageimage00_detection.jpg

object_detection.py --folder=./images4test
           --model=./ssd_mobilenet_v1_coco_11_06_2017.tar.gz
           --detections=./image4test_detections
```

Para desarrollar el script, el profesor facilita a los alumnos un Jupyter notebook donde se detallan los pasos a seguir para hacer uso del API de TensorFlow para detección de objetos. El alumno deberá:

- Generar un pequeño dataset de test con imágenes descargadas de internet (10-20 imágenes)
- Clonar el repositorio de modelos de TensorFlow
- Ejecutar notebook de detección de objetos : object_detection_tutorial.ipynb
- Desarrollar un script `object_detection.py` basado en el Jupyter notebook anterior
- Probar el script sobre el conjunto de imágenes descargadas

Nota: antes de ejecutar el notebook para detección de objetos es necesario compilar la dependencia `protobuf`. Para ello, se deben seguir las instrucciones descritas aquí.

Ejercicio 8

Montar un Jupyter notebook similar a los facilitados por el profesor en ejercicios anteriores. Dicho notebook deberá mostrar los pasos necesarios para llevar a cabo la etapa de inferencia/predicción utilizando alguno de los modelos disponibles en Caffe Model Zoo.Se pide:

- Seleccionar uno de los modelos disponibles en Caffe Model Zoo. La temática es libre. El alumno podrá elegir el modelo a probar. Existen múltiples opciones: modelos para reconocimiento facial, detección de gradiente, *landmarks* faciales, clasificación de proteínas, clasificación de modelos de coche, reconocimiento de escenas, etc.
    - Si no existiera ninguna preferencia, los profesores sugieren probar la red de segmentación semántica FCN. Existen múltiples modelos pre-entrenados de esta red. Se recomienda el uso del modelo FCN-8s PASCAL
- Cargar la red y hacer predicción sobre un conjunto de imágenes
- Por cada imagen mostrar información relativa a su predicción (dependerá del tipo de modelo seleccionado)