Artificial Intelligence
TP 5-6
Convolutional Neural Networks

Tutor : francesca.galassi@inria.fr

The goal of this practical is to initiate you to the use of Convolutional Neural Networks (CNNs) to complete basic image classification tasks [1].

1. The MNIST Handwritten Digit Classification Dataset [2]

The Modified National Institute of Standards and Technology (MNIST) dataset comprises 60000 grayscale images of size 28 x 28 pixels. Each image is an handwritten digit between 0 and 9. The task is to classify a given image into one of the 10 classes representing integer values from 0 to 9.

Download the Python notebook file `AI_TP5-6_mnist_conv.ipynb` from the Moodle and open it into your notebook. Complete the cells where indicated i.e. *TO DO*, and answer the following :

1. Describe the MNIST dataset.
2. model.summary() provides a description of the CNN architecture :
   (a) Compute the number of trainable parameters for each layer
   (b) How does the max-pooling operation influence the size of the feature map ?
   (c) What is the role of the dropout layer ?
3. Describe the learning curves and observe the accuracy of the model on the test set.
4. Remove the dropout layers : how does this affect the CNN ?
5. Modify the hyper-parameters and the CNN, and observe the effects on the accuracy. Document your experimental observations.

2. The Kaggle Dogs and Cats Classification Dataset

Kaggle has hosted the popular Dogs and Cats Classification challenge since 2013. The original dataset contains about 25000 training photos of dogs and cats [3]. We have selected

[1] This practical assumes that you are using standalone Keras running on top of TensorFlow with Python 3

[2] `http://yann.lecun.com/exdb/mnist/`

[3] `https://www.kaggle.com/c/dogs-vs-cats/data`

a much smaller subset to train your model on. Download the subset from : `https://www.dropbox.com/sh/2vqbw8x3l1e8zeu/AACigf0jXJMS3_Qjej3tm0kXa?dl=0`

2.1 As in the previous exercise, complete the cells where indicated in the Python file. Answer the following :

1. Describe the dataset.
2. Describe your CNN for image classification.
3. Describe the learning curves and observe the accuracy of the model on the train and validation set.
4. Modify the hyper-parameters and the CNN, and observe the effects on the accuracy. Document your observations.

2.2 In this exercise, you will explore dropout regularization, data augmentation on the fly, and transfer learning.

1. Introduce/remove dropout layers in your CNN. Train your model and review the learning curves. Document your observations.
2. Modify the code to perform data augmentation. Specify the augmentations as arguments to the ImageDataGenerator.
   Refer to the documentation : `https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator`.
   Train your model and review the learning curves. Document your observations.
3. The $transfer\_model()$ function in the notebook file allows you to use a pre-trained VGG16 model for your classification task. Refer to the documentation : `https://keras.io/api/applications/vgg/#vgg16-function`.
   Answer the following :
   (a) Describe the VGG16 architecture and the $transfer\_model()$ function.
   (b) Modify your code to use the $transfer\_model()$ function on your dataset. Note that the VGG16 model expects 224 x 224 input images.
   (c) Describe the learning curves. Document your observations.