2 - Homework

MAP541
Winter 2018/2019

The objectives of the homework

The purpose of this homework is to build an image recognition system, using transfer learning and fine-tuning based on pre tuned convolutional neural networks on ImageNet. (`http://www.image-net.org/`). I started with `https://deeplearningsandbox.com/`. With Python you may use keras pre trained model (see for instance `https://keras.io/applications/`).

Ex. 1 — ImageNet

1. what is ImageNet?
2. how many different kinds of cheese can you find in ImageNet?
3. what is the best classifier on ImageNet and what is its error rate?

Ex. 2 — Build an image recognition system

Build an image recognition system for 1000 everyday object categories (ImageNet ILSVRC) using Keras and TensorFlow[1].

1. import the relevant modules from keras and the pre trained ResNet50[2].

```
from keras.preprocessing import image
from keras.applications.resnet50 import ResNet50, preprocess_input, decode_predictions
```

2. define ResNet50 as you model and check its architecture

```
model = ResNet50(weights='imagenet')
model.summary()
```

3. open an image ("my_image.jpg" in the following example) representing a single object (if possible represented in ImageNet)

```
img = Image.open("my_image.jpg")
```

4. reshape the image to fit the input format of your model

```
target_size = (224, 224)
img = img.resize(target_size)
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
```

5. preprocess the input

```
x = preprocess_input(x)
```

6. get the model predictions

```
preds = model.predict(x)
```

7. display the top 5 recognized objects. Do you find the one of your image?

```
decode_predictions(preds, top=5)[0]
```

[1] `deeplearningsandbox.com`
[2] `https://keras.io/applications/`

Ex. 3 — Your turn

1. based on your previous work, build a classifier with two classes (only two objects) by transfer learning and fine tuning.
   a) choose two classes (cat/dog or muffin/chihuahua or parrot/guacamole or livarot/pont leveque or whatever/whatever else...)
   b) download some (say 10 to 50) images of each class on the web split your images into two sets (training and testing) and setup our data with a training directory and a validation directory as follows:

```
train_dir/
class1/
class2/
val_dir/
class1/
class2/
```

   c) proceed adapting the code from
   `https://keras.rstudio.com/articles/applications.html`
2. Is it better to do transfer learning and fine tuning or both?