184.702 Machine Learning – Exercise 3.2 Deep Learning for Image or Text Classification

Note: please first read the generic introduction document for the 3rd exercise!

Topic 3.2: Deep Learning

N.b.: if you want to work with Deep Learning, but on more advanced topics, also take a look at the task on security (adversarial machine learning: inversion, evasion, poisoning attacks) and on synthetic image generation (using generative adversarial networks) In exercise Topic 3.1.

The main goal of this exercise is to get a feeling and understanding on the importance of representation of complex media content, in this case images or text. You will thus get some datasets that have an image classification target.

(1) In the first step, you shall try to find a good classifier with „traditional" feature extraction methods. Thus, pick

- For Images
  - One simple feature representation (such as a colour histogram, see also the example provided in TUWEL) and
  - A feature extractor based on SIFT (https://en.wikipedia.org/wiki/Scale-invariant_feature_transform) and subsequent visual bag of words (e.g. https://kushalvyas.github.io/BOV.html for python), or a similar powerful approach (such as SURF, HOG, ..).
- For Text
  - One feature extractor based on e.g. Bag Of Words, or n-grams, or similar

You shall evaluate them on a few algorithms (for image, also specifically including an MLP), and parameter settings to see what performance you can achieve, to have a baseline for the subsequent steps.

(2) Then, try to use a deep learning approaches, such as **convolutional neural networks (for images) or recurrent neural networks (for text)**, or other approaches, and see how your performance differs.

Compare not just the overall measures, but perform a detailed comparison and analysis per class (confusion matrix), to identify if the two approaches lead to different types of errors in the different classes, and also try to identify other patterns.

For images, you can base your DL implementation on the tutorial provided by colleagues at the institute, available at https://github.com/tuwien-musicir/DL_Tutorial/blob/master/Car_recognition.ipynb (you can also check the rest of the repository for interesting code; credit to Thomas Lidy (http://www.ifs.tuwien.ac.at/~lidy/)).

Also perform a detailed comparison of runtime, considering both time for training and testing, including also the feature extraction components.

For the **datasets** you shall work with, pick two text/image datasets, from the list of suggestions below. If you have proposals for other datasets, please inform me (mayer@ifs.tuwien.ac.at), and we can see if the dataset is fit.

- For Images (you shall chose only one of the AT&T and Yale Face datasets, but not both of them, as they are both relatively small):
    - The German Traffic Sign Recognition Benchmark (GTSRB), http://benchmark.ini.rub.de/
    - AT&T (Olivetti) Faces: https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, https://scikit-learn.org/0.19/datasets/olivetti_faces.html
    - Yale Face Database: http://vision.ucsd.edu/content/yale-face-database
    - CIFAR-10: https://www.cs.toronto.edu/~kriz/cifar.html
    - Tiny ImageNet: https://tiny-imagenet.herokuapp.com/
    - Labeled Faces in the Wild, http://vis-www.cs.umass.edu/lfw/, using the task for face recognition. See also https://scikit-learn.org/0.19/datasets/#the-labeled-faces-in-the-wild-face-recognition-dataset
- For Text Data:
    - 20 newsgroups: http://qwone.com/~jason/20Newsgroups/
    - Reuters: http://www.daviddlewis.com/resources/testcollections/reuters21578/

Recommendations for CNNs specifically:

- Use architectures of your choice – you can work with something simple like a LeNet, or a bit more advanced architectures (where maybe transfer learning is required for efficiency reasons, see below).

- Use as well data augmentation (you can reuse the code from the tutorial), and compare it to the non-augmented results
- Also consider using transfer learning of pre-trained models