Assignment 12
Ensemble Learning
EECE-6822 Machine Learning

Cris Ababei
Electrical and Computer Engr., Marquette University

1. Objective

The objectives of this activity include: (1) run several code examples that study and illustrate ensemble learning; and (2) develop VotingClassifiers and Model Averaging Ensembles for a dataset with images of cats and dogs.

2. Prerequisite Readings

Murphy
- Ch.18: Trees, Forests, Bagging, and Boosting (18.2 Ensemble learning)

Geron
- Ch.7: Ensemble Learning and Random Forests

Raschka
- Ch.7: Combining Different Models for Ensemble Learning

Supplemental:

Hastie ESLII
- Ch.16: Ensemble learning

3. Code Examples

Example 1: Ensemble methods in SciKit-Learn ← (section "8. Voting Classifier")

This is the example code from **Ch.7** from Aurelian Geron's book.
[*B3-Geron] Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 2022.
Open in your google colab and go through the code. You should read first the chapter itself from the book, before going through the code.

https://github.com/ageron/handson-ml3/blob/main/07_ensemble_learning_and_random_forests.ipynb

The chapter discusses the most popular Ensemble methods, including bagging, boosting, and stacking..

Example 2: Ensemble learning in Python

This code example is from **Ch.7** from Sebastian Rashka's book:
[*B3-Raschka] Sebastian Raschka, Yuxi Liu, and Vahid Mirjalili, Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, Packt Publishing, 2022.
The source code (both Python code an Jupyter Notebook) is located at the GitHub repository. You should read first the chapter itself from the book, before going through the code.
https://github.com/rasbt/machine-learning-book/tree/main/ch07

To execute the code, I found the easiest usually to run directly the Python code (same code as in the notebook) in Anaconda Spyder. However, because the source code was developed with earlier versions of various libraries, you may need to fix little things here and there to make it work.

This chapter explores different methods for constructing a set of classifiers that can often have a better predictive performance than any of its individual members:

- Make predictions based on majority voting
- Use bagging to reduce overfitting by drawing random combinations of the training dataset with repetition
- Apply boosting to build powerful models from weak learners that learn from their mistakes

Example 3: Trees and Forests

These are examples suggested by K. Murphy on his github repository for the textbook:

https://github.com/probml/pyprobml/tree/master/notebooks/book1/18

There are several interesting code examples that you may want to explore, including:

https://colab.research.google.com/github/probml/pyprobml/blob/master/notebooks/book1/18/feature_importance_trees_tutorial.ipynb

https://github.com/probml/pyprobml/blob/master/notebooks/book1/18/rf_feature_importance_mnist.ipynb

Example 4: mlm tutorials

Here, we look at several tutorials from **machinelearningmastery (mlm)**. Read the following tutorials and run the code where applicable:

--Essence of Bootstrap Aggregation Ensembles

https://machinelearningmastery.com/essence-of-bootstrap-aggregation-ensembles/

--Essence of Boosting Ensembles for Machine Learning

https://machinelearningmastery.com/essence-of-boosting-ensembles-for-machine-learning/

--Ensemble Machine Learning With Python (7-Day Mini-Course)

https://machinelearningmastery.com/ensemble-machine-learning-with-python-7-day-mini-course/

--Develop a Bagging Ensemble with Different Data Transformations

https://machinelearningmastery.com/bagging-ensemble-with-different-data-transformations/

--How to Develop a Bagging Ensemble with Python

https://machinelearningmastery.com/bagging-ensemble-with-python/

--How to Develop a Gradient Boosting Machine Ensemble in Python

https://machinelearningmastery.com/gradient-boosting-machine-ensemble-in-python/

--How to Develop a Weighted Average Ensemble for Deep Learning Neural Networks ←

https://machinelearningmastery.com/weighted-average-ensemble-for-deep-learning-neural-networks/

--How to Develop an Ensemble of Deep Learning Models in Keras

https://machinelearningmastery.com/model-averaging-ensemble-for-deep-learning-neural-networks/

4. Assignment

Part 1:
First, run and study to understand the code from section "8. Voting Classifier" of Geron's book:
https://github.com/ageron/handson-ml3/blob/main/07_ensemble_learning_and_random_forests.ipynb
It trains several classifiers on the MNIST dataset: Random Forest classifier, Extra-Trees classifier, SVM, and MLP Classifier. Then, it combines the classifiers into an ensemble that outperforms them all on the validation set, using a soft or hard voting classifier. It uses:

from sklearn.ensemble import VotingClassifier

You only need to focus on the first part of this section, that shows the use of **VotingClassifier**; no need to go to the portion that "Let's remove the SVM to see if performance improves."

In **Part 1 of this assignment**, you must develop a similar VotingClassifier but for a different dataset that contains cats and dogs images. You can see how to download and import the dataset "cats_and_dogs_filtered" in the beginning of this example:
https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/transfer_learning.ipynb
However, in your implementation, instead of working with images 160x160, you should **rescale them to IMAGE_SIZE = (64, 64)** in order to reduce the computational runtime; hopefully, that will be enough rescaling.
You must find the classification accuracies of the four individual classifiers as well as of the VotingClassifier; report them into a table.

Part 2:
Here, you must conduct an investigation into the use of Model Averaging Ensemble (with n_members = 8) and compare the results to those obtained in Part 1.

First, read and do the code example from this tutorial:
https://machinelearningmastery.com/weighted-average-ensemble-for-deep-learning-neural-networks/ (Note that the Python code of this tutorial is included within the .zip file for this lecture)
Focus on the code from "6. Model Averaging Ensemble" because you need to reuse it for your cats and dogs dataset (of course, with images scaled as in Part 1).
Feel free to change the number of epochs to reduce computational runtime to reasonable times.
You will need to change the code a bit to define a model = Sequential() with the correct parameters.
Use a number of 16 units/neurons instead of 25 on the first Dense layer.
The images will be input into the MLP model by flattening them into 64x64 values.

You must create and include in your report the plot "*Line Plot Showing Single Model Accuracy (blue dots) and Accuracy of Ensembles of Increasing Size (orange line)*" from the section "4. Model Averaging Ensemble". However, please add axis labels to make it clearer. Compare the accuracy to that obtained in Part 1.

5. Deliverables

You must write (typed) a report and upload it as a PDF file on D2L. The report should be named "**hw12_report_LastName.pdf**". You should also create a .zip archive with all your code and implementations of all parts of the assignment. Upload also this archive .zip file with the name "**hw12_implementation_code_LastName.zip**" to D2L. Hence, your D2L should contain two items: the report and the .zip file. **Do not include the report inside the .zip and upload only the .zip. They should be two separate items!**

The report should include the following sections and subsections. Make sure section titles are in bold font and pages are numbered.

1) **Title + course info + your name**
2) **Summary.** Describe in one paragraph what the objective of the assignment is.
3) **Description of Experiments and Discussion.** Describe the experiments you did. All tables and figures should be numbered and should have captions. All plots in all figures should have axes labels and titles. Present a meaningful discussion with the interpretation of the results you obtained. Explain if you expected the results or not; discuss the intuition behind it.
4) **Conclusion.** Present your conclusions; highlight what are your main takeaways that you learned from this assignment. Describe what issues you encountered and how you solved them.
5) **References.** Include all references that you used, as a numbered list. Cite them in the report itself; do not just list them! If your report has References that are not numbered and cited in the report, points will be deducted!