Assignment 3 - Final Assignment

October 12, 2019

Final Assignment

Introduction (remove before submission)

The final assignment ties in with the final assignment for the course Data Engineering. In this assignment, we will take the data 'preprocessed' in the data pipeline (starting from reading in the files to obtaining clean (X,y) data frames), and use deep learning models to model the data and make predictions.

This is a 'freeform' assignment in that you will be given some guidelines, but otherwise you will have a lot of freedom to choose the exact outcome of the machine learning step, as long as it makes sense given the datasets that you chose.

As in the data engineering assigment, you will need to do this for at least two datasets and their corresponding prediction task:

- Flower dataset: Predict the flower name given the image.
- Reuters dataset: Predict the topic of a news article given the words that occur in them
- IMDB movie review dataset: Predict whether a review is positive or negative
- Twitter dumps: Choose a tweet collection that has labels and do the corresponding predictions. Choose carefully, some collections cannot be used.

Students that do not follow the Data Engineering course can instead use the Street View House Numbers Dataset from assignment 1 and/or one of the text-related datasets included in Keras (e.g. news wires, movie reviews,. . .)

As before, you should be smart in how you use the data to develop your model (also *carefully* read the guidelines below). In addition, given that some datasets are 'easier' than others, you are allowed to simplify the harder problems by taking a subsample of the data for training, or even selecting a subset of all possible classes (e.g. select 20 out of the 102 classes of flowers). However, you must motivate such decisions in the report, e.g. by showing running times on small samples to project how long it would take to train on much more data, or showing how much performance drops (and training time increases) as you add more classes.

Team details

Add the names of all team members and a short description of each member's contributions.
TEAM MEMBER 1 = . . .
TEAM MEMBER 2 = . . .
TEAM MEMBER 3 = . . .
TEAM MEMBER 4 = . . .
TEAM MEMBER 5 = . . .

Instructions and advice (remove before submission)

- Answer the questions in this notebook, including the code, results, and discussions.
- Add precise explanations when interpreting the results of your experiments. Use markdown cells for this.
- Submit both this notebook and a PDF through Canvas. To create the PDF, see File > Export Notebook as PDFs.
- Submit only one notebook and one PDF per team. In Jupyter Lab, you can copy-paste cells (with results) into a single notebook.
- Keep the PDF below 20 pages. Remove these instructions and the general advise below in the final PDF.
- Avoid all(!) unnecessary outputs. Only output the answers to the questions. Add ';' behind lines that > generate output to suppress unnecessary output.
- All tasks can be completed with Keras. You are allowed (but not required) to use other tools as well.
- Training models can take time. Make sure to start computations well in advance of the deadline.
- Be efficient. Don't try every model on all the data at once. Test your code on a small part of the data (even just 1%) until you are sure that there are no more bugs or other issues.
- Observe how long it takes to train different models on a small part of your data to learn whether the models are fast enough for what you want to do.
- You are allowed to run code elsewhere (e.g. in Python scripts) and store the results on file. If so, copy the code in this notebook and load/visualize/discuss the results here as well.
- Questions 3 and 4 depend on question 2, but otherwise there are no strong dependencies. Use this fact when planning work in your team.
- On Google Colab you can run the notebook in the cloud.

1. Simple networks (20 points)

- Evaluate a dense network or CNN of 1-3 hidden layers. Explain how you design the network: how many nodes did you use in each layer and why, how did you choose the output layer, which loss function did you choose, etc... Take the dataset size into account, e.g. deeper networks are hard to train on small datasets.
- Evaluate the performance in an adequate way.
- Apply different types of regularization to your dataset (e.g. L1/L2 regularization, dropout,...). Evaluate whether regularization helps and discuss your findings.
- Try to further improve performance by manually tuning other hyperparameters, such as the choice of optimizer, learning rate,...

2. Embeddings (30 points)

- Choose an adequate embedding for your data. For instance, for textual data you can use word embeddings, while for image data you can use a pretrained network. Explain how you chose the embeddings and how you implemented them.
- Add 1-2 dense layers to the embedded representations and train the resulting model on your data.
- Evaluate the embeddings: Do they yield better models compared to question 1?

3. Tuning (20 points)

- Try to further improve the models from question 2 by adding regularization and by tuning hyperparameters (e.g. with random search).
- If applicable for your dataset, use data augmentation and evaluate whether it yields better models.
- Explain your reasoning and discuss the results. Did you improve much?

4. Transfer learning (30 points)

- Use the embeddings created in question 2, but without the final layer, and export the activations of the last layer to a new dataframe. This should give you a new representation of your original data.
- On this new dataframe, choose and evaluate several of the 'classical' machine learning models. E.g. for classification you can use logistic regression, SVMs, random forests, and gradient boosting. Explain your choice.
- Perform adequate tuning and evaluate whether the resulting models are better that the ones you obtained before.

5. Bonus (5 points)

- If you feel that you went 'above and beyond' the assignment, provide a motivation to gain up to 5 bonus points.
- For instance, maybe you chose a very hard dataset, tried much more complex models, did very extensive tuning, or you tried an additional technique?
- If not, simply ignore this question.

Note: Not all of the above steps make sense on every dataset. If you cannot apply a certain step, motivate this clearly in the report.

Good luck!