

# HW5.0 (50 points) REMEMBER TO START SMALL THEN SCALE UP

## TEXT CLASSIFICATION:

- Create a labeled dataset by selecting and downloading at least 3 novels from project Gutenberg (the label being the title of the book)
- Generate the following scripts

### 01-clean.py

- Write a script to clean the data and break the novels into chunks of text (each with the relevant label)
- Convert into a dataset similar to the IMDB format (feel free to do it differently if you prefer) <https://www.gutenberg.org>
  - You don't have to use the entire novel (especially when debugging)
- Save the processed data in HW5.0 so you don't have to pre-process every time you train

### 02-train.py

- Load the data from 01-clean.py and further process if needed (vectorize or use embedding layer)
- Train both a 1D CNN and a RNN model to predict the novel title (category) based on the fragments from the text

You can use any type of RNN you wish (SimpleRNN, GRU, LSTM)

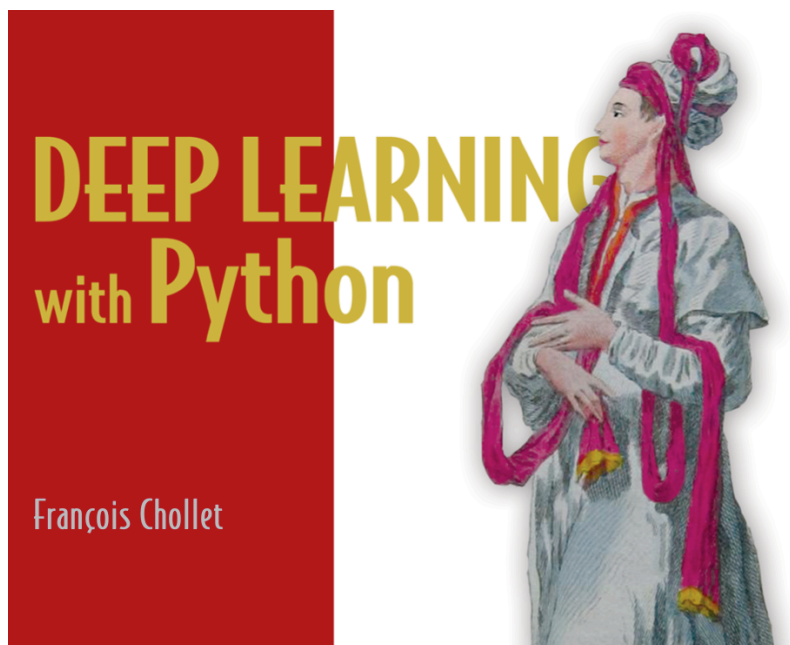
- Try to maximize your accuracy using methods for the lecture and textbook
- Do K=1 validations (i.e just one split of training-test-validation)(fix the seed so you can recreate the validation set)
- Try to follow guidelines from the “universal ML workflow” used throughout the course
- Include some form of regularization (L1, L2, Dropout)
- Track multiple classification metrics including ROC or AUC
- MAKE SURE YOU SAVE TRAINING HISTORY PLOTS TO PNG FILES SO THE TA's DON'T HAVE TO RE-TRAIN
- WRITE A LOG FILE (log.txt) WITH TRAINING INFO AND FINAL METRICS
- Save your final trained model

### 03-evaluate.py

- Write a short second script to load the model and write important metrics (training, test, val) to the screen
- **Optional:** repeat but try to predict something else like genre or date written (would require many more more than 3 books)
- **Extra credit (+10 points):** Repeat the exercise but generate a labeled dataset (text, topic) using the Wikipedia API (see lecture codes) Then train both CNN and RNN models to predict the topic of a given webpage based on it's text (do in separate directory called WIKI)

**IMPORTANT NOTE:** I have scripts that I periodically run to compare every student's files to all other student files (copying code from another student in any way isn't recommended)

# Reading Assignment



## 6 *Deep learning for text and sequences* 178

### 6.1 Working with text data 180

*One-hot encoding of words and characters* 181 ▪ *Using word embeddings* 184 ▪ *Putting it all together: from raw text to word embeddings* 188 ▪ *Wrapping up* 195

### 6.2 Understanding recurrent neural networks 196

*A recurrent layer in Keras* 198 ▪ *Understanding the LSTM and GRU layers* 202 ▪ *A concrete LSTM example in Keras* 204 ▪ *Wrapping up* 206

### 6.3 Advanced use of recurrent neural networks 207

*A temperature-forecasting problem* 207 ▪ *Preparing the data* 210 ▪ *A common-sense, non-machine-learning baseline* 212 ▪ *A basic machine-learning approach* 213 ▪ *A first recurrent baseline* 215 ▪ *Using recurrent dropout to fight overfitting* 216 ▪ *Stacking recurrent layers* 217 ▪ *Using bidirectional RNNs* 219 ▪ *Going even further* 222 ▪ *Wrapping up* 223

### 6.4 Sequence processing with convnets 225

*Understanding 1D convolution for sequence data* 225 ▪ *1D pooling for sequence data* 226 ▪ *Implementing a 1D convnet* 226 ▪ *Combining CNNs and RNNs to process long sequences* 228 ▪ *Wrapping up* 231

### 6.5 Chapter summary 232