Samuel Valentim-Gervais   November 10th, 2019

Deliverable 3

Final model results

I finally decided to go with deep learning models for both age and gender. I redid some of the preprocessing for this: instead of the bag-of-words representation for the blog posts, I created a list of words, sorted by frequency, and converted each blog post into a list of indexes into this list, so information like word order and frequency within each blog post would be preserved. As for the model itself, I first added a layer that converted word indices into a vector in a 20-dimensional vector (this layer learns embeddings too, it's pretty cool!), followed by 2 1-dimensional convolutional layers. This allows the model to consider words appearing together, and not just the presence of words individually. Then, I added a 200 neuron hidden layer and an output neuron (sigmoid for gender, relu for age). The models perform pretty well, for what they're worth; here are my preliminary results for gender classification:

Naïve bayes test accuracy (gender): 60.11%
Random forest test accuracy (gender): 58.96%

And here's how the deep learning model performs:

Test accuracy (gender): 64.11%

Since the dataset is a 50-50 split gender-wise, other classification metrics (precision, recall, etc.) aren't really relevant here. The increase may not seem like much, but it's about 1.5 times as many percentage points above random as the classical learning models. Plus, gender classification based purely on text is a hard task, even for humans!

What's really exciting, is the model's performance for predicting age:

Test mean absolute error (age): 5.08

The model is, on average, off by just 5 years. Five. That's insanely accurate. I didn't have any preliminary results for age, so this is a first try, too. FIVE.

Final demo proposal

My plan is to have an app with a text box that people can write in, with the model's predictions given on the side in real time, updated every time they type a word (every whitespace or punctuation character). I'm afraid it might underperform because people who are just trying it out might just type un-blogposty things that don't lend themselves to this sort of prediction. (For example: "hello! I wonder if this thing can predict who i am. Test 123 test123 no its not working :O this thing is trash") Some simple javascript thing could do the trick if the interface needs to be in a webapp, I haven't really done much javascript, but all I need is a frontend, it shouldn't be that hard. (Famous last words, I know.) If not, I can code the

UI in java as a standalone program (I know a good java GUI editor, and the less UI work for me, the better).