Implement an attention model that takes an image of a PDF math formula, and outputs the characters of the LaTeX source that generates the formula.

Getting Started

For a quick start, download a prebuilt dataset or use these tools to build your own dataset. Alternatively, you can proceed manually with the following steps:

- Download a large number papers from arXiv. There is a collection of 29,000 arXiv papers that you could get started with. It is likely that this set of 29,000 papers may contain several hundred thousand formulas, which is more than enough for getting started. As the bandwidth of arXiv is limited, it is important to be mindful of their constraints and to not write crawlers to download all the papers of arxiv.
- Use a heuristic to find all the LaTeX formulas in the LaTeX source. It can be done by looking for the text that lies between `\begin{equation}` and `\end{equation}`. Here is a list of some of the places where equations can appear in latex files. Additional examples can be found here. It is likely that even a simple heuristic for extracting latex formulas should produce in excess of 100,000 equations; if not, keep refining the heuristic.
- Compile images of all the formulas. To keep track of the correspondence between the latex formulas and their images, it is easiest to place exactly one formula on each page. Then, when processing the latex file, it is easy to keep track of the pages. Be sure to not render formulas so large that they exceed an entire page. Also, be sure to render the formulas in several fonts.
- Train a visual attention sequence-to-sequence model (as in the Show, Attend, and Tell paper, or perhaps a different variant of visual attention) that would take an image of a formula as input, and output the latex source of the formula, one character at a time. A Theano implementation of the Show, Attend, and Tell paper can help you get started. If you wish to implement your model from scratch, TensorFlow can be a good starting point.
- It takes some effort to correctly implement a sequence-to-sequence model with attention. To debug your model, we recommend that you start with a toy synthetic OCR problem, where the inputs are long images that are obtained by concatenating sequences of images of MNIST digits, and the labels should be a sequence of their classifications. While this problem can be solved without an attention model, it is useful as a sanity check, to ensure that the implementation is not badly broken.
- We recommend trying the Adam optimizer.

Notes

A success here would be a very cool result and could be used to build a useful online tool.

While this is a very non-trivial project, we've marked it with a one-star difficulty rating because we know it's solvable using current methods. It is still very challenging to really do it, as it requires getting several ML components together correctly.

Solutions

Results, data set, code, and a write-up are available at http://lstm.seas.harvard.edu/latex/. The model is trained on the above data sets and uses an extension of the Show, Attend and Tell paper combined with a multi-row LSTM encoder. Code is written in Torch (based on the seq2seq-attn system), and the model is optimized using SGD. Additional experiments are run using the model to generate HTML from small webpages.