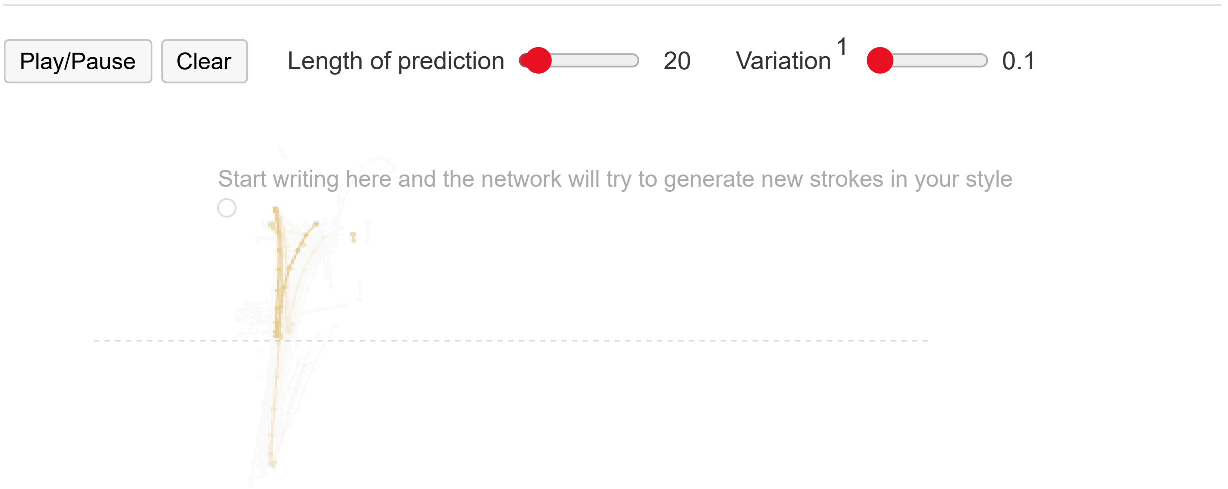
Four Experiments in Handwriting with a Neural Network

Let’s start with generating new strokes based on your handwriting input



Length of prediction 20 Variation[1] 0.1

Start writing here and the network will try to generate new strokes in your style

SHAN CARTER Google Brain
DAVID HA Google Brain
IAN JOHNSON Google Cloud
CHRIS OLAH Google Brain

Dec. 6 2016
Citation: Carter, et al., 2016

Neural networks are an extremely successful approach to machine learning, but it’s tricky to understand why they behave the way they do. This has sparked a lot of interest and effort around trying to understand and visualize them, which we think is so far just scratching the surface of what is possible.

In this article we will try to push forward in this direction by taking a generative model of handwriting[2] and visualizing it in a number of ways. The model is quite simple (so as to run well in the browser) so the generated output mostly produces gibberish letters and words (albeit, gibberish that look like real handwriting), but it is still useful for our purposes of exploring visualization techniques.

In the end we don’t have some ultimate answer or visualization, but we do have some interesting ideas to share. Ultimately we hope they make it easier to divine some meaning from the internals of these model.

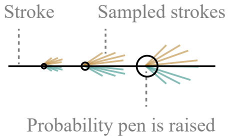
Looking at the Output of the Model

Our first experiment is the most obvious: when we want to see how well someone has learned a task we usually ask them to demonstrate it. So, let’s ask our model to write something for us and see how well it does.

Play/Pause Variation[1] 0.6

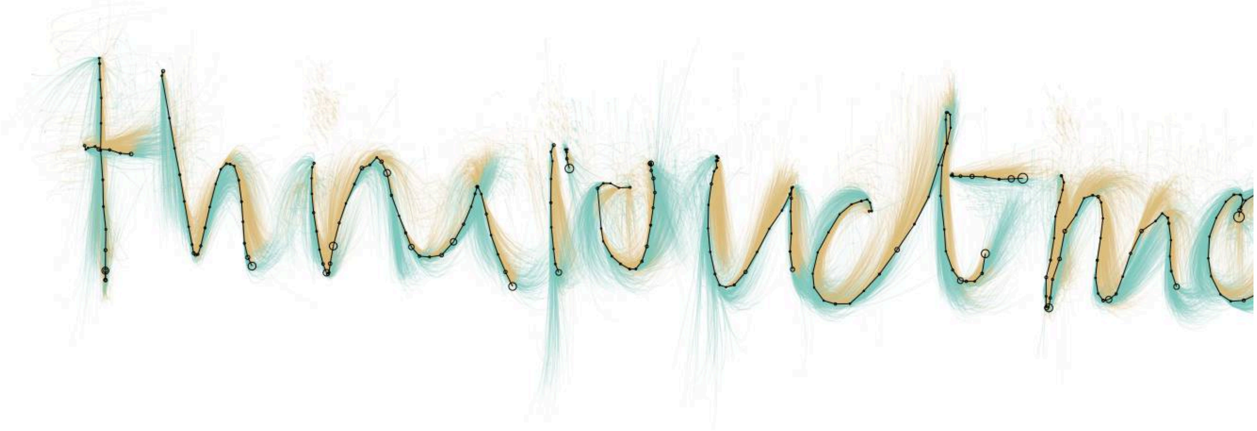
Most of the marks are gibberish, but many of them are surprisingly convincing. Some real (or real-ish) words even start to appear. One surprising thing you’ll notice is that the general style of handwriting is more or less consistent within a sample. This is because the type of architecture used for this model (LSTM) has a mechanism for remembering previous strokes. It is therefore able to remember things like how loopy or jerky the handwriting is, or which letter preceeded the current one. (For more on LSTMs and how they can remember, Chris Olah has a good primer [3].)

Even with this memory, the technique for generating samples from neural networks is probabilistic, meaning each of these samples is one of a much, much larger possibility space. Viewing them one at a time feels unsatisfying — how can we infer much from one or two samples when there are nearly infinite possibilities? If something looks wrong, how do we know if it was because there is something fundamentally wrong with our model or if it was just dumb luck?

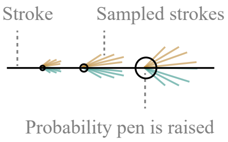


At each iteration our model could have produced many paths. Instead of picking one and throwing the rest away, let's draw, say, 50 of them. Green strokes below are places where the model would have veered rightward from the chosen stroke, orange is where it would have veered leftward.

Generated sample 3 Steps 8 Variation[1] 0.5

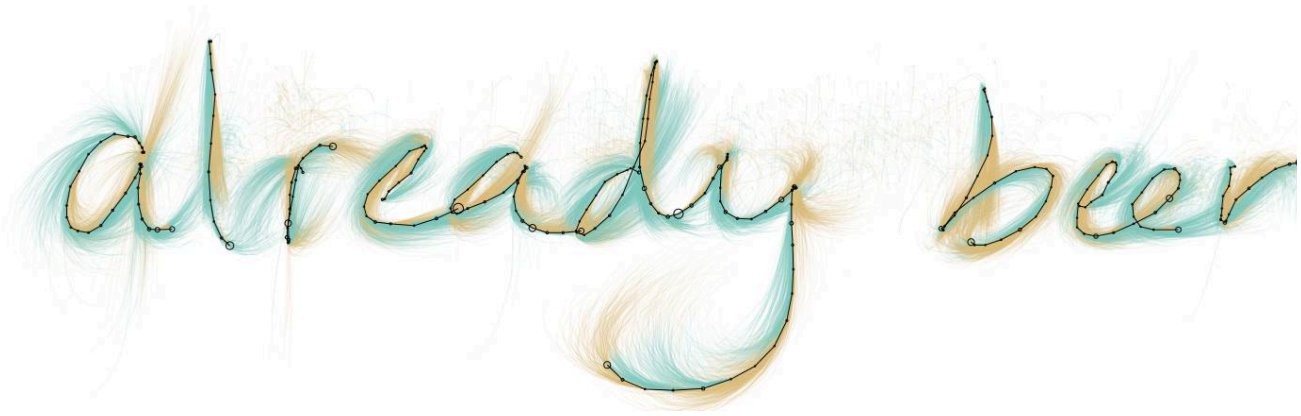


With this technique we are casting a light into a wider possibility space. You can see some areas where there was general consensus as to what to do next. Other areas had more of a "anything can happen next" feeling. Others seem to show a possible fork in the road. We can be drawing an "a" or "g". A few steps later it's clear the model has converged on a cursive "g".



In addition to visualizing samples generated by the model, this technique can also be applied to human-generated samples. Below we can see what the model would have done at each point if it had taken over for the person.

Validation sample 2 Steps 8 Variation[1] 0.5



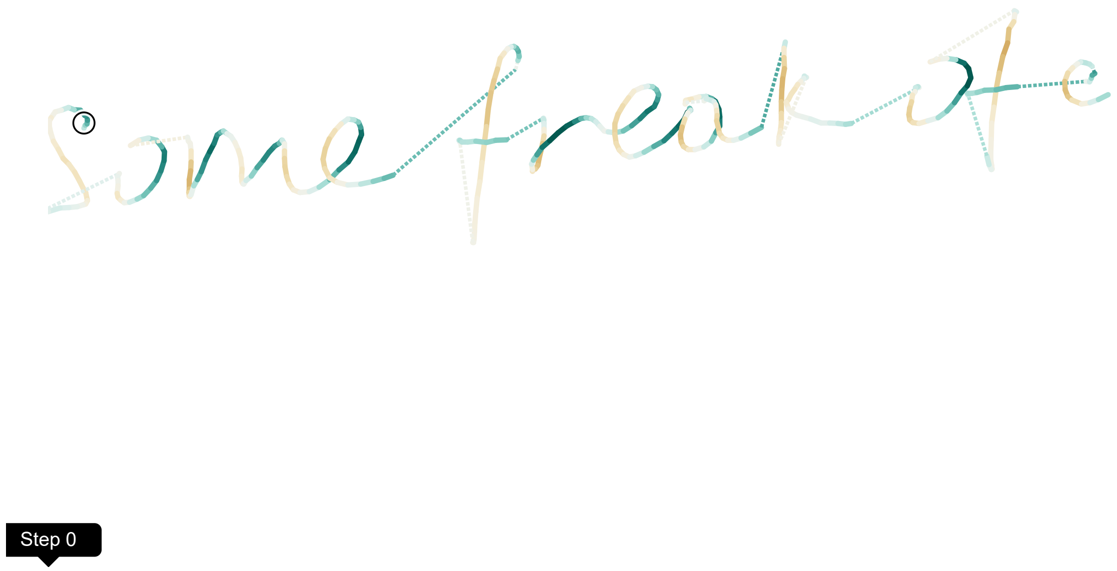
It is obvious from these experiments that this model has learned quite a lot about human handwriting. Which sort of raises the question, can we extract that knowledge in any meaningful way, rather than just blindly using it to mimic handwriting?

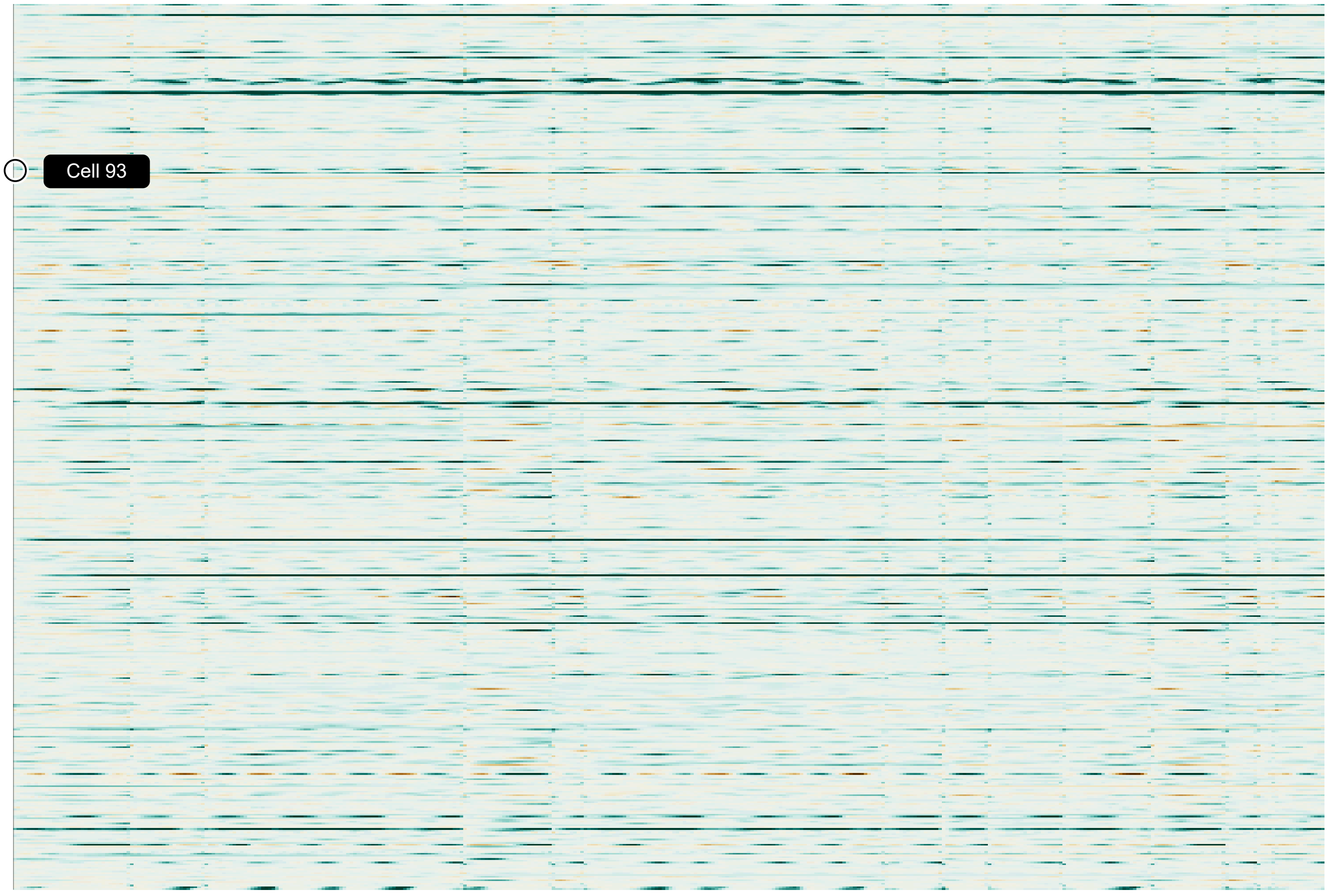
Examining the Internals of the Model

Our model has 500 cells which act as a sort of memory which it will use as part of its input when deciding what to generate. If we can see what those cells are doing as the model progresses we may be able to gain some intuitive understanding about what the model is doing.

We begin by showing the activation of the cells over time. Each column in the heatmap below represents one line segment of the handwriting. Each row represents one cell of the model and is colored by its activations on that part of the stroke. By inspecting the diagram you may be able to see some patterns in the way certain cells activate for certain types of strokes. You can change which cell is used to color the strokes by clicking on the diagram.

Validation example 1 ⌄ *colored by activations of* cell 93 ⌄

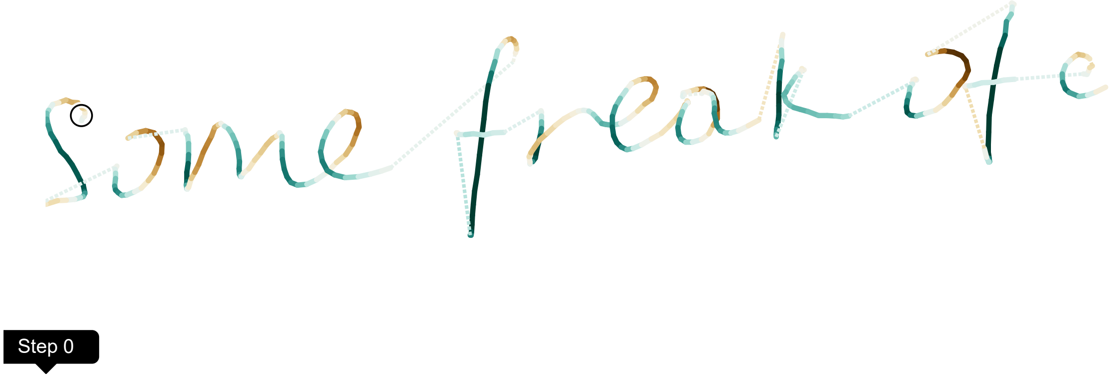


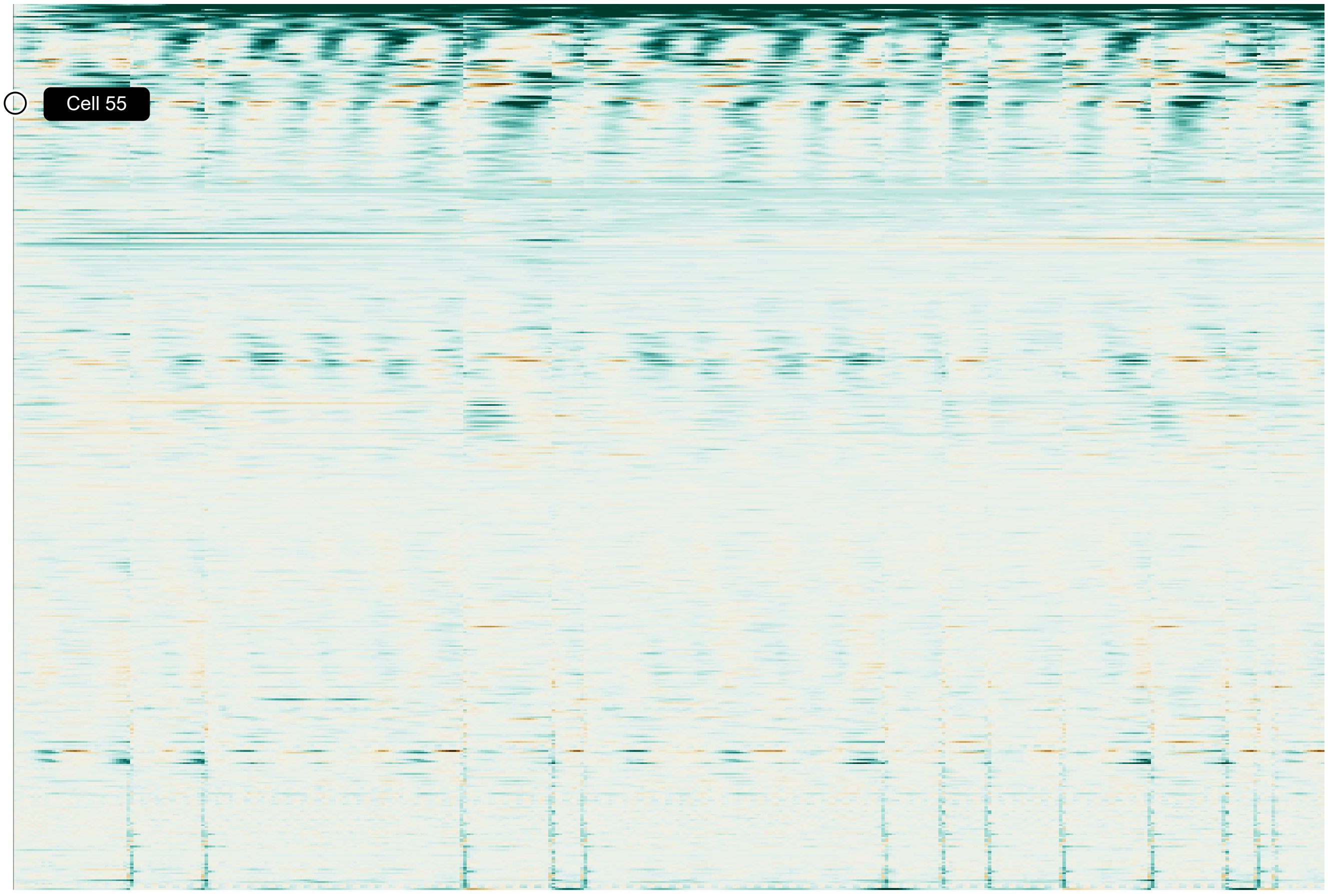


You may have been able to pick out one or two cells with interesting patterns, but we have reason to believe that cells work in tandem, such that an individual cell's activity may not be as interesting as a group of cells.

Is there a way to order the cells to make this structure clearer? We've found that applying one-dimensional t-SNE [4] to the activations of the cells over time can organize them, bringing cells with similar behavior close together. This makes the diagram easier to read. We use a few small tricks to achieve the best results.[3]

Validation example 1 *colored by activations of* cell 55





In this version we can find a few clear behaviors. For example, cells 11-70 at the top seem sensitive to slightly different directions and curvatures of the pen path — see in particular cells 25 and 55. On the other hand, at the bottom, cells below 427 seem focused on pen lifts; for example, cell 494 seems to predict whether the pen is about to be lifted. In the middle, we see a variety of cells, including some tracking absolute position. Cell 136 and its neighbors seem concerned with horizontal position, while cells around 236 seem concerned with vertical position. Cell 242 appears to track position within a word.

Another way to explore the activations is to give it a sample and interactively see the activations. Below you can write and see the activations of all the cells in real time.

Clear

Conclusion

The black box reputation of machine learning models is well deserved, but we believe part of that reputation has been born from the programming context into which they have been locked. The experience of having an easily inspectable model available in the same programming context as the interactive visualization environment (here, javascript) proved to be very productive for prototyping and exploring new ideas for this post.

As we are able to move them more and more into the same programming context that user interface work is done, we believe we will see richer modes of human-ai interactions flourish. This could have a marked impact on debugging and building models, for sure, but also in how the models are used. Machine learning research typically seeks to mimic and substitute humans, and increasingly it’s able to. What seems less explored is using machine learning to augment humans. This sort of complicated human-machine interaction is best explored when the full capabilities of the model are available in the user interface context.

Acknowledgments

We are grateful for the comments of Fernanda Viegas, Martin Wattenberg, and Daniel Smilkov.

This work was made possible by the support of the Google Brain team.

Author Contributions

Shan Carter wrote the article and created the interactive experiments in the first section. David Ha created the handwriting model and ported it to Javascript. Ian Johnson created the final diagrams exploring activations. Chris Olah provided guidance and core ideas for the diagrams and edited the article.

Footnotes

1. The model has a parameter which determines how widely it samples from the underlying distribution. It is labeled here as variation but is more commonly referred to as temperature. Temperature is most commonly discussed in Boltzmann distributions, but can be generalized to all probability distributions. In this more general form, changing the temperature by a factor of *T* corresponds to raising all probabilities to the power of *1/T* and normalizing.
2. The model used in this articles is a version of the model described in Section 4.2 of Generating Sequences With Recurrent Neural Network by Alex Graves [1]. It is a small LSTM, with 500 hidden units, trained to perform the unconditional handwriting generation task. For a detailed description of the model and training procedure, please refer to this blog post [2] in addition to the Graves paper. After training the LSTM, we quantized the weights using 8-bit integers, and exported the weights into a small JSON-Base64 formatted file.
3. We use two different tricks to make the t-SNE organization of neurons work better. First, if a cell state is mostly negative, we canonicalize it by flipping its sign. (From the LSTM’s perspective, this is completely equivalent, provided we flip the sign of some weights. The sign of a cell state is arbitrary.) Secondly, we tried using different metrics on points in our dataset for t-SNE, encouraging it to put points we think of as similar close together. We got good results basing our metric on blurred data (so that cells offset forward or backwards by one are still close together) and a blurred Sobolev metric to encourage sharp changes to line up. We achieved slightly better results, presented in the article, yet by creating a metric based on percentiles of cell activation in a neighborhood. See `bin/sort` in the repository for details.

References

1. **Generating sequences with recurrent neural networks** [PDF]
   Graves, A., 2013. arXiv preprint arXiv:1308.0850.
2. **Handwriting Generation Demo in TensorFlow** [link]
   Ha, D., 2016.
3. **Understanding LSTM Networks** [link]
   Olah, C., 2015.
4. **Visualizing data using t-SNE** [PDF]
   Maaten, L.v.d. and Hinton, G., 2008. Journal of Machine Learning Research, Vol 9(Nov), pp. 2579—2605.

Updates and Corrections

View all changes to this article since it was first published. If you see a mistake or want to suggest a change, please create an issue on GitHub.

Citations and Reuse

For attribution in academic contexts, please cite this work as

```
Carter, et al., "Experiments in Handwriting with a Neural Network", Distill, 2016. http://doi.org/10.23915/distill.0000
```

BibTeX citation

```
@article{carter2016experiments,
  author = {Carter, Shan and Ha, David and Johnson, Ian and Olah, Chris},
  title = {Experiments in Handwriting with a Neural Network},
  journal = {Distill},
  year = {2016},
  url = {http://distill.pub/2016/handwriting},
  doi = {10.23915/distill.00004}
}
```