January 2018

FontGen

Generating fonts using a neural net

Matt Gaikema

```
ClearAll["Global`*"]
SetDirectory[NotebookDirectory[]];
```

I spend a lot of time trying to identify and download fonts, so a program that could do it for me would be wonderful. There are many academic papers on the subject, but none of them have published their code, so I decided to implement it myself and learn more about deep learning along the way.

Data

The data is a whopping 13+ gigabytes and was scraped from the internet by someone else. It can be downloaded here.

```
Import["fonts.hdf5", {"Dimensions"}]
```

```
⟨|/fonts → {56443, 62, 64, 64}|⟩
```

I debugged the neural net using a much smaller dataset, consisting of my personal collection of fonts. The smaller dataset was created by running a script written by the same guy who got the original data.

```
Import["fonts.small.hdf5", {"Dimensions"}]
f = Import["fonts.small.hdf5", {"Datasets", "fonts"}];
(* Show an example character. *)
Image[f[[1]][[1]]]
```

```
⟨|/fonts → {21, 62, 64, 64}|⟩
```

Network

Each letter is 64x64 pixels.

```
(* We use 4 letters as input... *)
4 * 64 ^ 2
(* ... and get 62 letters as output. *)
62 * 64 ^ 2
```

```
16 384
```

```
253 952
```

There are $4*64^2 = 16,384$ input neurons, and $62*64^2 = 253,953$ output neurons. The letters chosen for the inputs are “B”, “A”, “S”, and “Q”.

```
(* http://bit.ly/2oeHyZX *)
Range[0, 25];
letters = AssociationThread[ToUpperCase[#] & /@ Alphabet[], %];
Lookup[letters, {"B", "A", "S", "Q"}]
Image[f[[1]][[# + 1]]] & /@ %
```

```
{1, 0, 18, 16}
```

Architecture

Here is the architecture that I used for the model. Many of the parameters were determined through tedious trial-and-error.

```
In[27]:= (* Created using Keras's `plot_model` function: https://
          keras.io/visualization/ *)
         Import["img/model.png"]
```

After training is complete, running `model.save_weights()` saves the model and its weights to an HDF5 file, which can be loaded to recreate the model (see the documentation).

In[26]:=
```
Import["model.hdf5", {"Dimensions"}]
```

Out[26]=
```
⟨|/batch_normalization_1/batch_normalization_1/beta:0 → {253952},
 /batch_normalization_1/batch_normalization_1/gamma:0 → {253952},
 /batch_normalization_1/batch_normalization_1/moving_mean:0 → {253952},
 /batch_normalization_1/batch_normalization_1/moving_variance:0 → {253952},
 /conv2d_1/conv2d_1/bias:0 → {8}, /conv2d_1/conv2d_1/kernel:0 → {4, 4, 1, 8},
 /conv2d_2/conv2d_2/bias:0 → {8}, /conv2d_2/conv2d_2/kernel:0 → {4, 4, 1, 8},
 /conv2d_3/conv2d_3/bias:0 → {8}, /conv2d_3/conv2d_3/kernel:0 → {4, 4, 1, 8},
 /conv2d_4/conv2d_4/bias:0 → {8}, /conv2d_4/conv2d_4/kernel:0 → {4, 4, 1, 8},
 /conv2d_5/conv2d_5/bias:0 → {10}, /conv2d_5/conv2d_5/kernel:0 → {4, 4, 32, 10},
 /dense_1/dense_1/bias:0 → {253952}, /dense_1/dense_1/kernel:0 → {10, 253952}|⟩
```

Evaluation

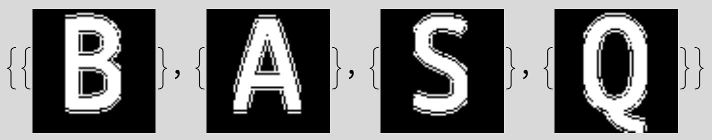
```
(* A function which transforms a list of image arrays to a list of images. *)
displayFont[f_]:=Map[Map[Image,#]&,f];
```

I tested the model on the font Ubuntu Mono, which is available for download here.

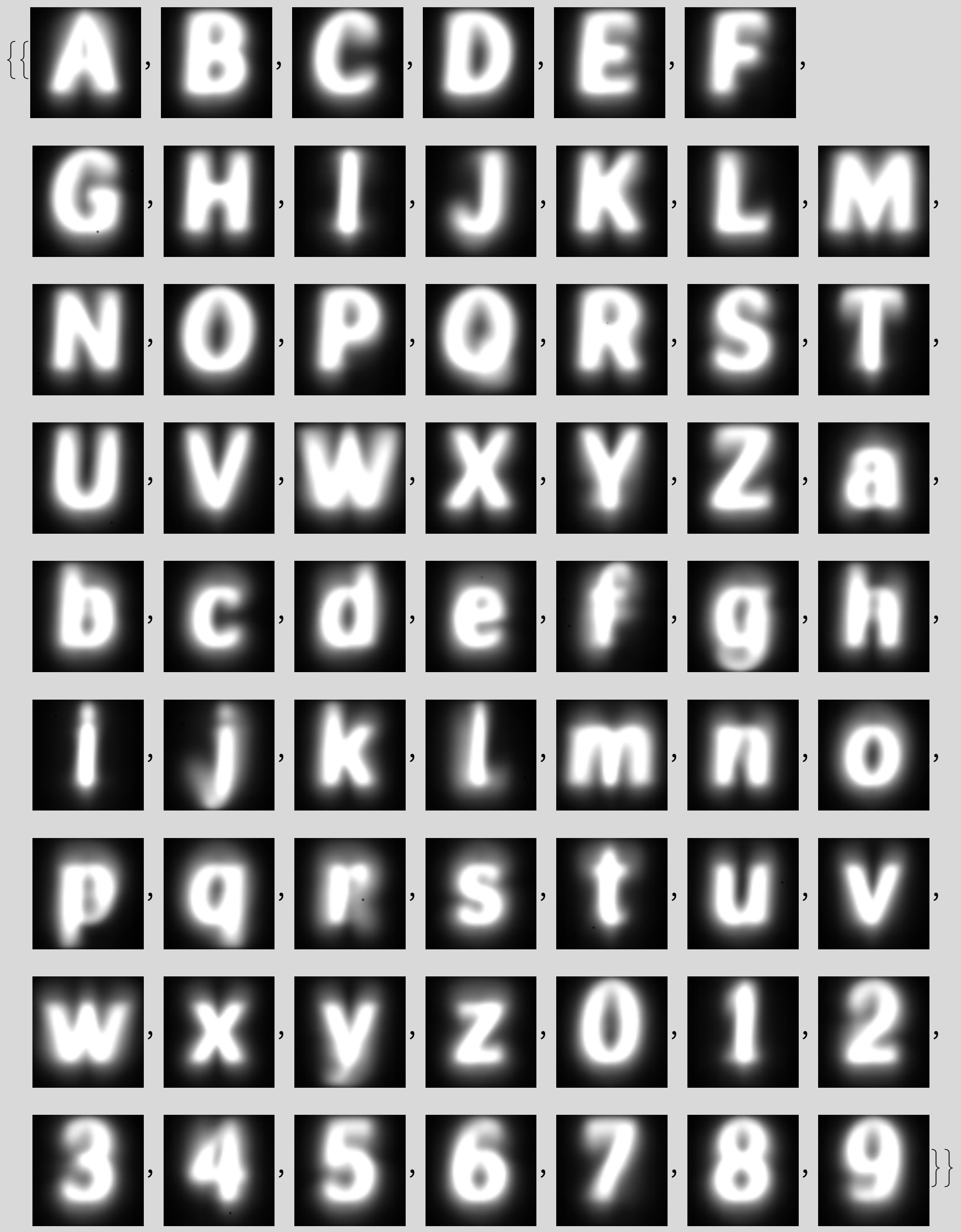
In[58]:=
```
Import["test.hdf5", {"Dimensions"}]
testInput = Import["test.hdf5", {"input"}];
displayFont[testInput]
output = Import["test.hdf5", {"output"}];
displayFont[output]
```

Out[58]=
```
⟨|/input → {4, 1, 64, 64, 1}, /output → {1, 62, 64, 64}|⟩
```

Out[60]=



Out[62]=



References

```
(* There is no better way to insert references on Mac/Linux. See http://
 bit.ly/2lo3Pkc *)
Import["sources.bib"]
```

```
@inproceedings{paper,
    title = {Learning Typographic Style},
    author  = {Shumeet Baluja},
    year  = {2016},
    URL = {http://arxiv.org/abs/1603.04000},
    booktitle = {arXiv}
}

@misc{deepfont,
    url =
  {https://erikbern.com/2016/01/21/analyzing-50k-fonts-using-deep-neural-networks
  .html},
    title = {Analyzing 50k fonts using deep neural networks},
    author = {Erik Bernhardsson},
    year = {2016},
    month = JAN
}

@misc{fonsi,
    author = {Fonsi Bonilla},
    howpublished = {Personal Correspondance},
    year = {2017},
    month = DEC
}
```