DeLINEATE: A deep learning toolbox for neuroscientists

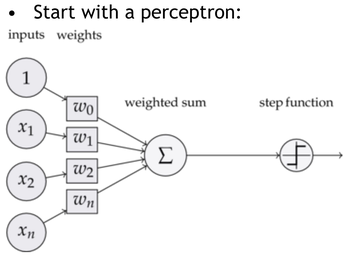
Karl M. Kuntzelman, Jacob M. Williams, Ashok Samal, Prahalada K. Rao, & Matthew R. Johnson

Why?

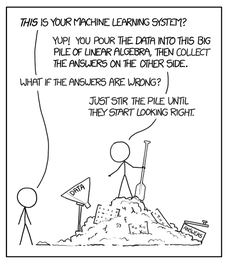
- Many modern approaches to neuroimaging data analysis (i.e., Multivariate Pattern Analysis) rely on machine learning-based classifiers
- Classical approaches (e.g., Support Vector Machines or Logistic Regression) are inherently limited in the kinds of classification they can perform, and become computationally costly with large amounts of data
- Modern neural network classifiers are more flexible and computationally tractable, at the cost of more researcher degrees of freedom without any established best practices
- To facilitate the exploration of deep learning in neuroscience, we need tools and structure for researchers to better engage with and manage additional complexity

What is deep learning?

- Linear algebra, but more fundable
- Start with a perceptron:



https://tex.stackexchange.com/questions/104334/tikz-diagram-of-a-perceptron



https://xkcd.com/1838/

- Wire a whole bunch of them together, sprinkle in some other data transformations to taste
- Feed some feature values into the first layer, propagate results to the end
- Compare the last layer to your class labels and use backpropagation to yell at the model
- Repeat until you have solved consciousness and/or created SkyNet

Have questions?

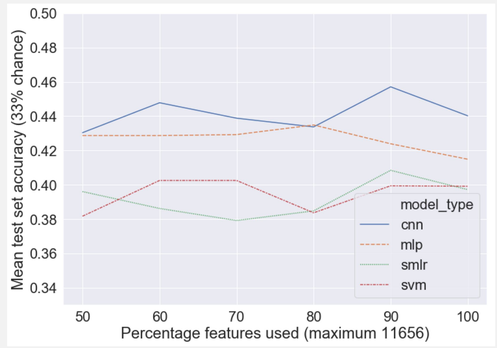
- Drop us a line, we'll be happy to discuss your project or help you get set up.
- Email:

  delineate.toolbox@unl.edu
  kkuntzelman2@unl.edu
  matthew.r.johnson@unl.edu
- Project website:

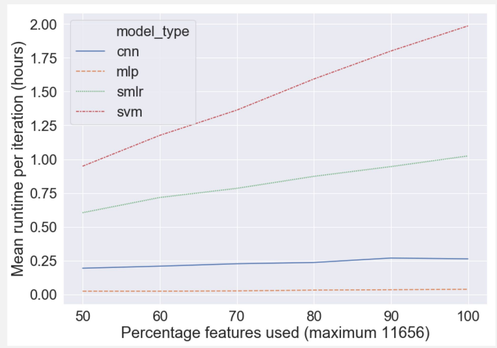
  http://delineate.it
- Project git repository:

  https://bitbucket.org/delineate/delineate

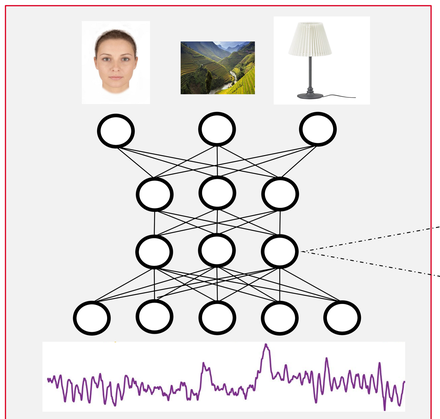
Features



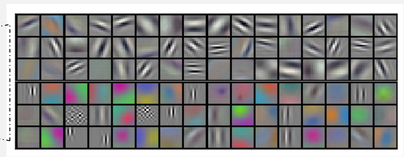
Classify simple stuff as accurately as classical MVPA!



…but way faster!

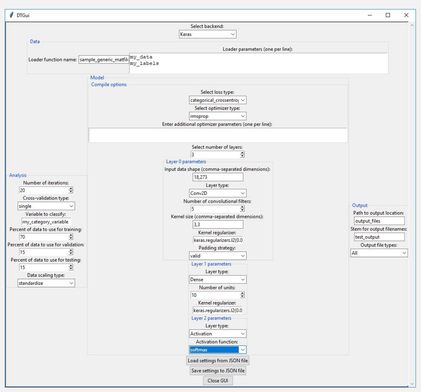


True multi-class models!



Krizhevsky et al., 2012

Rediscover feature detectors!



Define your analysis in an exciting new GUI!

```
"model": {"layers": [
            {"layer_type": "Dense",
             "units": 32,
             "input_shape": [11656],
             "kernel_regularizer": "keras.regularizers.l2(0.01)"
            },
            {"layer_type": "LeakyReLU",
             "alpha": 0.1
            },
            {"layer_type": "Dropout",
             "rate": 0.4
            },
            {"layer_type": "BatchNormalization"
            },
            {"layer_type": "Dense",
             "units": 32,
             "kernel_regularizer": "keras.regularizers.l2(0.01)"
```

…or directly in JSON for more options!

What do you need to get started?

Hardware

- Workstation capable of running CUDA (Formally any modern OS, Linux is probably easiest to set up)
- A decent GPU (nVIDIA with CUDA capability)

Software

- Python
- GPU drivers for parallel computing
- Theano or Tensorflow (These do the actual work but are Not Fun to interact with)
- Keras (Makes things friendlier if you know how to code)
- DeLINEATE (Friendly for all! See http://delineate.it or https://bitbucket.org/delineate/delineate/overview)

Files

- Data
- A loader function to get your data into the right shape (you can model this off of included examples)
- JSON model specification (ditto, or build it with the new GUI!)

Ongoing Development

- We take requests!
- Support for non-sequential models
- Expanded options and sanity checking in the GUI
- Better default parameter suggestions
- More documentation
- Include prefab published models

References & Acknowledgements

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Paper current draft/preprint available at:

Supported by:
NSF/EPSCoRgrant #1632849 and NIH P20 GM130461,
awarded to MRJ and colleagues
and
nVIDIA GPU grant