GPU COMPUTING WITH R AND KERAS

Danny Simpson

Graduate Student, Lewis-Sigler Institute for Integrative Genomics

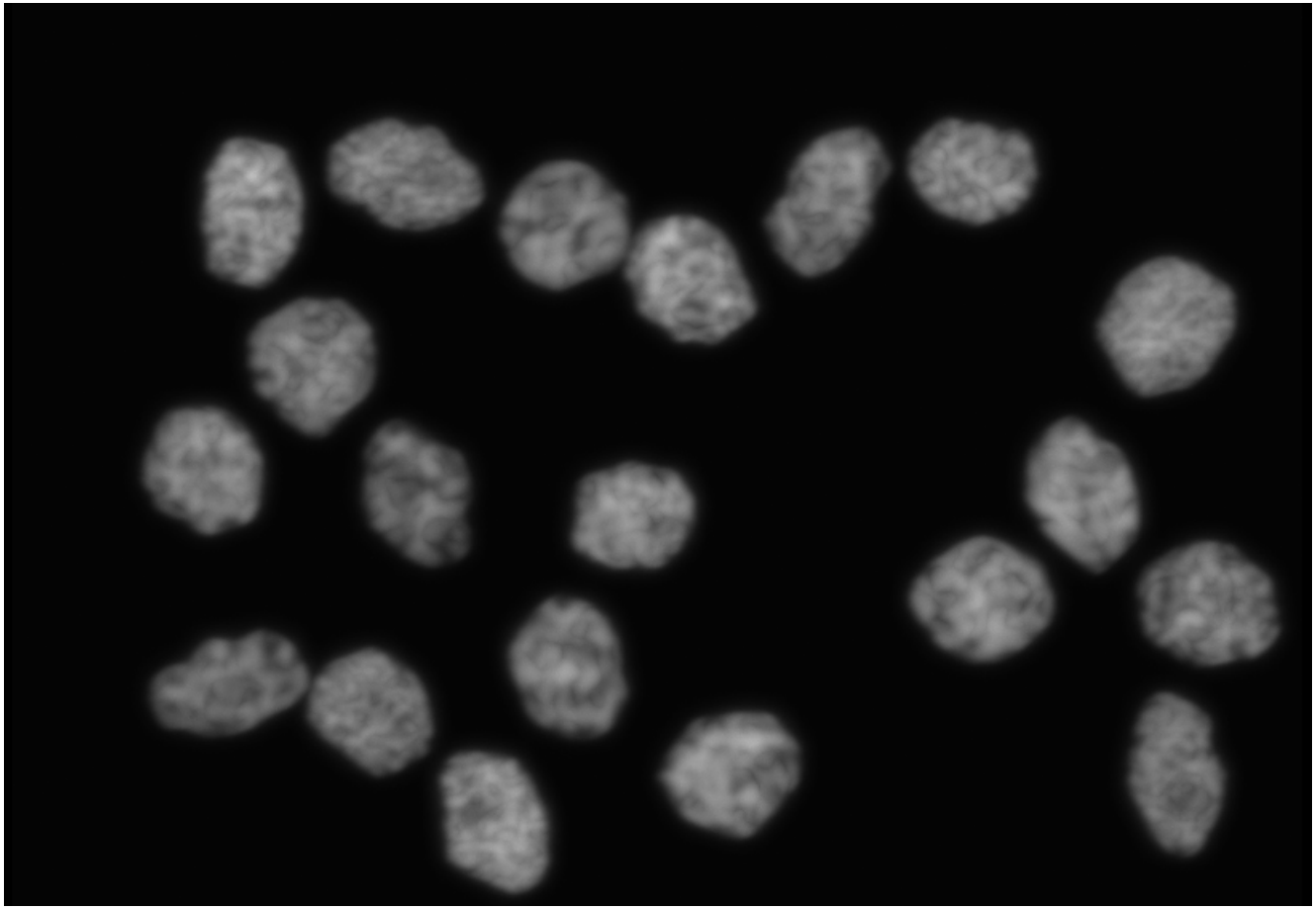
PhD Advisors Britt Adamson and Barbara Engelhardt

Why R?

R

- Has a user-friendly IDE (RStudio)
- Statistical framework
- Easily extensible
- Handles data in nearly any format
- Creates publication-ready graphics quickly

3D HL60 Cell Projection



3D convolutions are computationally expensive

How do I get this code working again?

Getting your environment ready

- Set up a conda environment using Python 3.6 and install Tensorflow and Keras
  - tensorflow-gpu
- Also install Keras package in R
- These are the only prerequisites to install

Using the environment

- Add use_condaenv(ENVNAME) to R scripts
- Request a GPU node
- Load the following modules (version matters!)
  - Cudatoolkit/10.0
  - Cudnn/cuda-10.0/7.5.0
- Activate the conda environment in job script

Code and tutorial are available at

github.com/simpsondl/mnist_r_tutorial

and

https://oncomputingwell.princeton.edu/2019/06/installing-and-using-tensorflow-with-r/

Defining and compiling the model

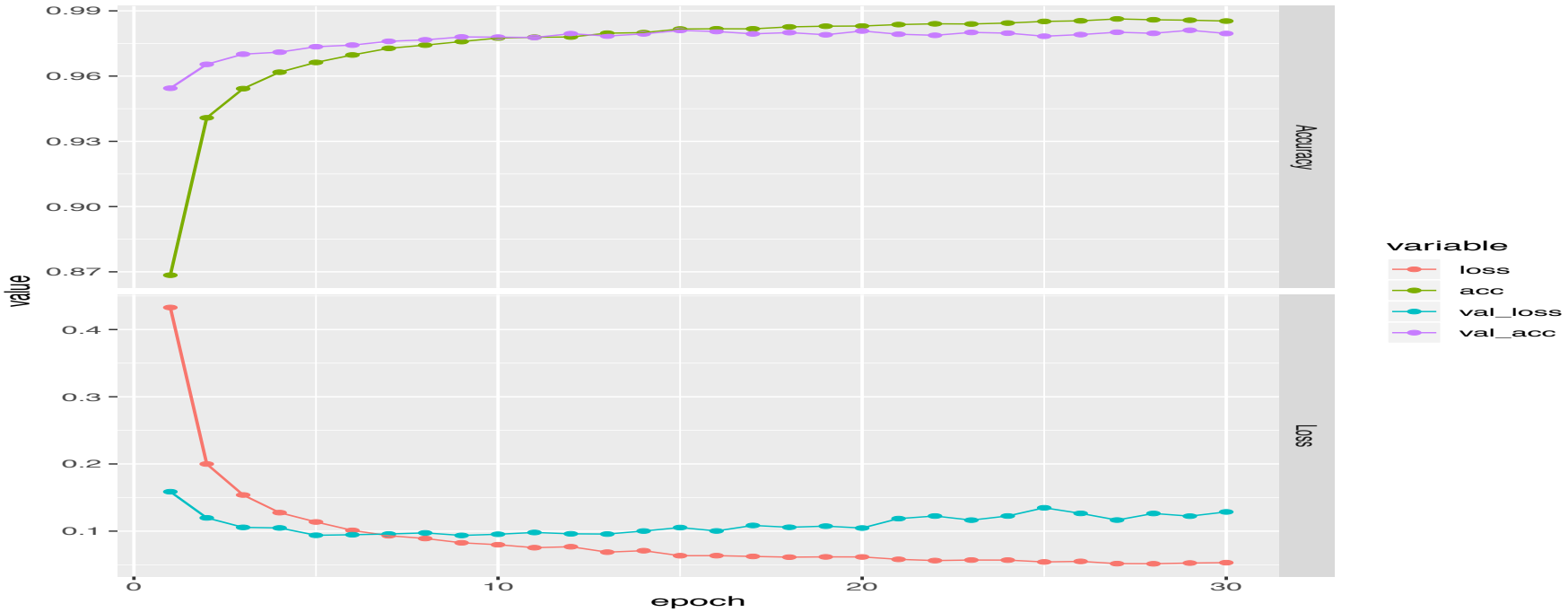
```
# Define the model
model <- keras_model_sequential()

model %>%
  layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 10, activation = 'softmax')
```

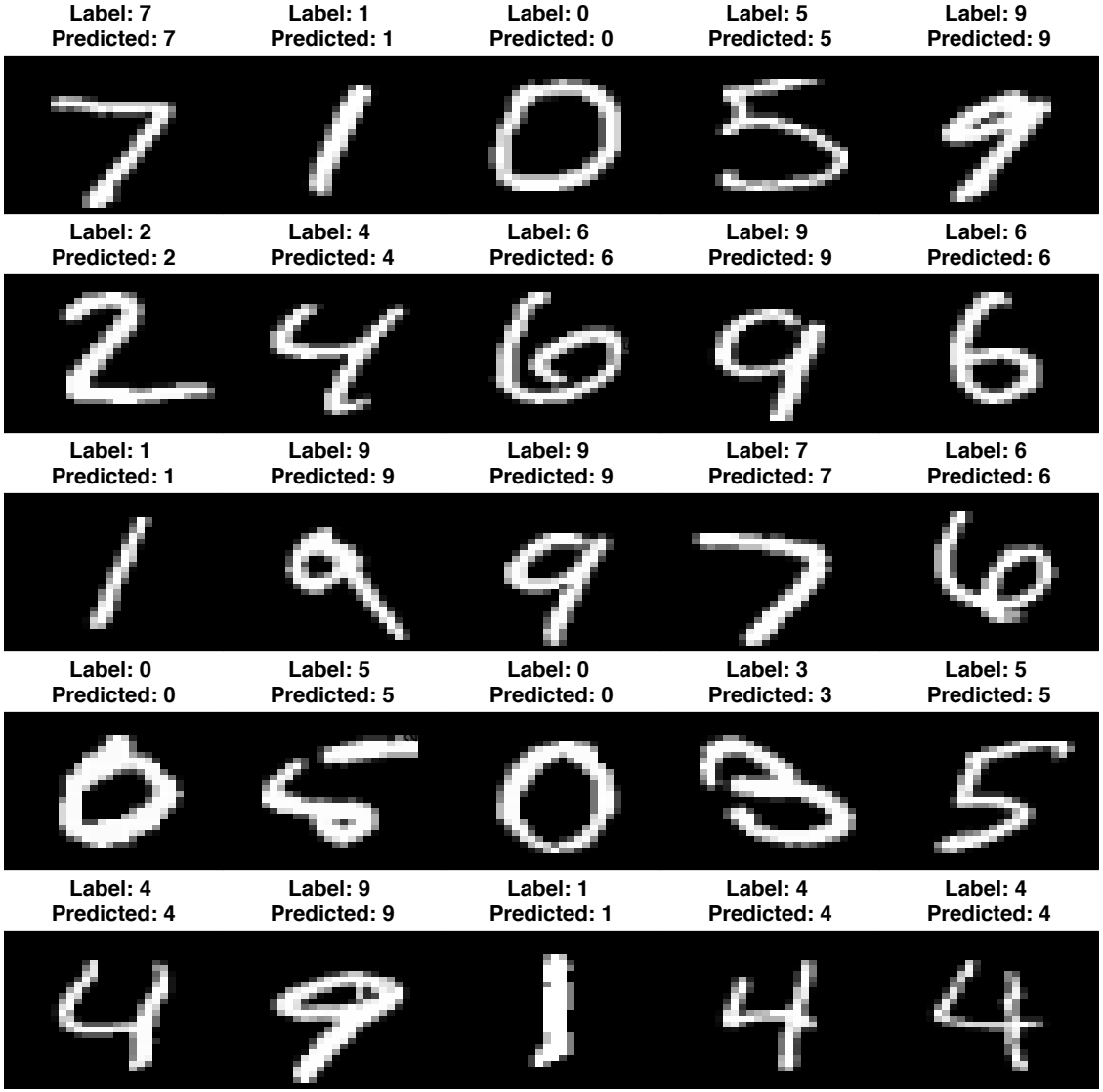
```
# Compile model
model %>%
  compile(loss = 'categorical_crossentropy',
          optimizer = optimizer_rmsprop(),
          metrics = c('accuracy'))
```

Training and evaluating

```
# Train model
history <- model %>%
            fit(x_train_pp, y_train_pp,
                epochs = 30, batch_size = 128,
                validation_split = 0.2)
```



Example Images and Labels



Acknowledgments

- Research Computing Support
- Adamson and Engelhardt Labs
  - Ann Cirincione
- Tensorflow and PyTorch User Group
- Lewis-Sigler Institute for Integrative Genomics