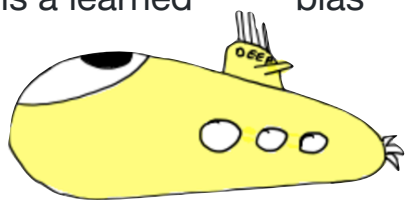
Cerebras Deep Learning Challenge

Zoe is a deep learning engineer for DeepWater Systems, and she uses a deep neural network to classify oil spills on the ocean floor. The oil spills look a lot like MNIST digits.

As the submersible does not have the computing power to run a framework like TensorFlow or Torch, help her implement a fully connected MLP in Python using only NumPy. Two hidden layers with a ReLU and SoftMax categorical cross-entropy loss is all she needs to get started. Train it on the MNIST dataset (available online at *http://yann.lecun.com/exdb/mnist/* in binary or at *https://pjreddie.com/projects/mnist-in-csv/* in easy to parse CSV) using stochastic gradient descent. Pick reasonable parameters for the width of the layers, batch size, learning rate and so on, but don't worry about state of the art numbers!

Submit your code as a private GitHub repository by sharing with GitHub user "ursk". Include a README.md that explains your design choices and the results you got. The code should be "production quality" with documentation and unit tests. It should include a script that can be run to reproduce the results. There should be no other dependencies than standard NumPy.

Extra credit:
Zoe wants to use a network with batch normalization (*arxiv.org/abs/1502.03167*), but unfortunately the embedded system also does not have enough computing power for that. She decides that *mean-only* batch normalization is good enough. Implement mean-only batch normalization, with scaling `x_hat = x - mu`, (where `mu` is the mean of the current mini-batch) and output `y = x_hat + beta` (where `beta` is a learned bias parameter) for the forward pass. Compare the performance of the two models.



Completing the project should take no more than one day. Good luck!