

Gradient descent algorithms

Supervisor: Ben Graham

A key component of many optimization algorithms is some form of gradient descent. For example, to train an artificial neural network (ANN) to perform a particular task, the network parameters must be fine tuned. For example, ANNs can be trained to do optical character recognition with accuracy exceeding 99% on some datasets, such as MNIST:



There are a large number of variations of the key principle: calculate a collection of derivatives and then move in that direction. The techniques of dropout and drop-Connect for turning ANNs into ensemble learners require a drastic re-interpretation of how gradient descent works.

The goal of this project is to explore the family of gradient descent algorithms, and to see how they interact with dropout.

Experience with programming in R or Python required.

1. Improving neural networks by preventing co-adaptation of feature detectors
Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov <http://arxiv.org/abs/1207.0580>
2. Regularization of Neural Networks using DropConnect, Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus <http://cs.nyu.edu/~wanli/dropc/>
3. No More Pesky Learning Rates Tom Schaul, Sixin Zhang, Yann LeCun
<http://arxiv.org/abs/1206.1106>