Tutorial 8

Tuesday 11th October, 2016

Problem 1. In class, we saw the detailed derivation of backpropogation update rules when each of the activation units is a sigmoid. You need to derive all the update rules when each activation unit happens to be rectified linear unit (ReLU).

$$\sigma(s) = max(\theta, s)$$

(since we often represent σ).) by $g(.)$, this also means $g(s) = max(\theta, s)$)

Typically, $\theta = 0$. Note that ReLU is differentiable at all points except at $s = \theta$. But by using subgradient $\nabla_s \sigma$ instead of gradient $\nabla \sigma$, we can complete backpropagation as 'subgradient descent'. Note that subgradient is the same as gradient in regions in which the function is differentiable. Thus,

$$\nabla_s \sigma(s) = 1,\ \ s \in (\theta, \infty)\ \ ,\ \nabla_s \sigma(s) = 0 \text{ if } s < \theta \text{ and } \nabla_s \sigma(s) \in [0,1] \text{ if } s = \theta$$

The interval $[0,1]$ is the subdifferential (denoted ∂), which is set of subgradients of σ at θ.

Is there a problem in cascading several layers of ReLU? Recall that we invoked subgradients in justifying the *Iterative Soft Thresholding Algorithm* for LASSO. And that LASSO gave sparsity owing to hard thresholding.

Problem 2. Compute the minimum number of multiplications and additions for a single backpropagation while also estimating the memory required for the minimum number of such multiplications and additions to become possible.

Problem 3. Solve the assignment at `https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/udacity/4_convolutions.ipynb`

Follow the instructions to implement and run each indicated step. Some steps have been implemented for you. This is a self-evaluated assignment. Make sure you are able to solve each problem and answer any posed questions and save the answers/solutions wherever possible.