Derek Lee Deep Learning Fall 2020

Professor Curro Quiz #1

**1. Explain the key differences between Adam and the basic gradient descent algorithm. Prose or mathematics equally acceptable.**

Adam has several advantages over basic gradient descent. The “magnitudes of parameter updates are invariant to rescaling of the gradient, its stepsizes are approximately bounded by the stepsize hyperparameter, it does not require a stationary objective, it works with sparse gradients, and it naturally performs a form of step size annealing.” It updates exponential moving averages of the gradient and the squared gradient. The moving averages estimate the mean and the variance of the gradient. Because the moving averages are initialized to 0, initial timesteps have estimates biased towards 0.

The update rule involves a careful choice of stepsizes. The stepsize has two upper bounds. The first upper bound is , in the case the second term is greater than one. The second upper bound is , in all other cases. The first case only occurs “in the most severe case of sparsity: when a gradient has been zero at all timesteps except the current timestep.” An example of this would be during the initial timesteps, when all moving averages have been initialized to 0. The effective magnitude of steps taken are approximately bounded by . It is possible to deduce the right order of magnitude of so the optima can be reached from the initial state within a set number of iterations. The effective stepsize depends on , with and dependent on the gradients. Thus, rescaling the gradient will not change the effective stepsize, because the scaling factor will be cancelled out.

Adam corrects for the bias caused by initializing the moving average to 0 by introducing bias correction terms. It does this by comparing the expected value of the exponential moving average with the true second moment. The resulting term is used to correct the initialization bias.

**2. Why does momentum really work?**

Standard gradient descent has a significant weakness: speed. Initially, it has a large decrease in the loss. However, the decrease slows down over time. This can be caused by pathological curvature. Regions of a smooth function *f* may be valleys, trenches, and ravines. Iterations can become stuck in a valley or take small steps towards the optimum.

Momentum introduces a ‘short-term memory’ to gradient descent. Its computational cost is almost nonexistent. This additional term is referred to as ‘acceleration’. This helps gradient descent maintain its speed. This can give a “quadratic speedup on many functions”.