CS 189 Introduction to Machine Learning
Spring 2018 DIS8

1 Understanding first order methods

In lecture we have encountered two first order methods so far: The gradient descent method and the stochastic gradient method. Both of them only use first order information (the gradient), hence the name. In this question we want to further our understanding as to why they work. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function that is minimized uniquely at $\mathbf{w}^\star$.

Remember the update for gradient descent on an arbitrary function $f(w)$

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha_k \nabla f(\mathbf{w}^{(k-1)}) \tag{1}$$

for some stepsize $\alpha > 0$ (in general the stepsize might vary in each iteration as seen in the homeworks).

The corresponding stochastic gradient method (with constant stepsize) does not take a step in the direction of $\nabla f(\mathbf{w}^{(k-1)})$ but a random direction $G(\mathbf{w}^{(k-1)})$. G is a random vector-valued function which is in expectation equal to the gradient ∇f, i.e. $\mathbb{E}_G[G(\mathbf{w})] = \nabla f(\mathbf{w})$ for all $\mathbf{w}$ where the expectation is over the stochasticity of the gradient. Therefore, we also have for the random variables $\mathbf{w}^{(k-1)}$ that $\mathbb{E}_{\mathbf{w}^{(k-1)}}\mathbb{E}_G[G(\mathbf{w}^{(k-1)})] = \nabla f(\mathbf{w}^{(k-1)})$, if $\mathbf{w}^{(k-1)}$ is independent of G. Note that strictly speaking we should write G_k instead of G since for each iteration it changes, but omit the index here for simplicity of notation.

The update then reads

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha_k G(\mathbf{w}^{(k-1)}). \tag{2}$$

In this problem, we want to focus on the convergence for (small enough) **constant stepsize** for both methods, so $\alpha_k = \alpha$ for all k.

(a) (optional) **Show that the gradient descent step is actually a descent method for small enough stepsize**, i.e. that $f(\mathbf{w}^{(k)}) \leq f(\mathbf{w}^{(k-1)})$. Hint: Feel free to assume that f is sufficiently differentiable and write down the second order Taylor expansion.

Solution: Let $f(\mathbf{x})$ be three times differentiable, then we get from the Taylor expansion around $\mathbf{w}^{(k-1)}$ that

$$\begin{aligned} f(\mathbf{w}^{(k)}) =& f(\mathbf{w}^{(k-1)}) + \nabla f(\mathbf{w}^{(k-1)})^\top(\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}) \\ &+ (\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)})^\top \mathbf{H}(\mathbf{w}^{(k-1)})(\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}) + O(\|\mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}\|^3), \end{aligned}$$

where $\mathbf{H}(\mathbf{w})$ denotes the Hessian of f at $\mathbf{w}$. Plugging the gradient step $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \lambda\nabla f(\mathbf{w}^{(k-1)})$ into the Taylor expansion yields

$$f(\mathbf{w}^{(k)}) = f(\mathbf{w}^{(k-1)}) - \lambda\|\nabla f(\mathbf{w}^{(k-1)})\|^2 + \lambda^2 \nabla f(\mathbf{w}^{(k-1)})^\top \mathbf{H}(\mathbf{w}^{(k-1)})\nabla f(\mathbf{w}^{(k-1)})$$

$$+ O(\lambda^3 \|\nabla f(\mathbf{w}^{(k-1)}\|^3)$$

By making λ small, we can make sure that in the above, the term with λ dominates the terms with λ^2 and higher order terms. Because $\|\nabla f(\mathbf{w}^{(k-1)})\|^2 > 0$ if $\mathbf{w}^{(k-1)}$ is not already an extreme point of f, going along the gradient direction with step size λ therefore decreases the function value.

For the examples we saw in homeworks and lecture, gradient descent iterates can be shown to get "strictly" closer to the optimum at every step. In fact, when f is strongly convex, gradient descent with correctly chosen constant stepsize converges linearly.

(b) In many practical cases in machine learning applications, the loss f that we want to minimize can be decomposed into a sum of functions, that is $f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w})$. This holds in particular for problems involving an average over the training data, which is for example the case when we want to find the maximum likelihood estimator given i.i.d. data in a generative probabilistic model.

In this case, we can define a random direction by just drawing an index i uniformly at random from $\{1,\ldots,n\}$ and setting $G(\mathbf{w}) = \nabla f_i(\mathbf{w})$. **Show that this choice of stochastic gradient is unbiased, i.e.** $\mathbb{E}_i[G(\mathbf{w})] = \nabla f(\mathbf{w})$ **where the expectation is over the random draw.**

Solution:

$$\begin{aligned}\mathbb{E}_i[G(\mathbf{w})] = \mathbb{E}_i[\nabla f_i(\mathbf{w})] &= \sum_{j=1}^n P(i=j)\nabla f_j(\mathbf{w})\\ &= \frac{1}{n}\sum_{j=1}^n \nabla f_j(\mathbf{w}) = \nabla f(\mathbf{w})\end{aligned}$$

(c) Let us denote the random index at iteration k by i_k. Given SGD updates $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha_k \nabla f_{i_k}(\mathbf{w}^{(k-1)})$, one key step in any convergence proof (including the one in homework) is to use the so-called law of iterated expectation (or *tower property*)

$$\mathbb{E}[\langle G(\mathbf{w}^{(k)}), \mathbf{w}^{(k)} - \mathbf{w}^\star\rangle] = \mathbb{E}_{i_1,\ldots,i_{k-1}}[\mathbb{E}_{i_k}[\langle G(\mathbf{w}^{(k)}), \mathbf{w}^{(k)} - \mathbf{w}^\star\rangle | i_1,\ldots,i_{k-1}]]. \tag{3}$$

In order to see that this is true, let us consider the following abstract general setting: Given a joint discrete probability distribution P on a finite number of vectors $\{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_n\}$ and scalars $\{k_1,\ldots,k_m\}$. Now let $\boldsymbol{\ell}$, K be r.v. drawn from the distribution. Recall the conditional probability $P(\boldsymbol{\ell}|K=k)$ and for fixed $k \in \{k_1,\ldots,k_m\}$ the definition of $\mathbb{E}(\boldsymbol{\ell}|k) := \sum_{i=1}^n \boldsymbol{\ell}_i P(\boldsymbol{\ell} = \boldsymbol{\ell}_i | K = k)$. When k is now random too, **show that by taking another expectation over** K **we obtain**

$$\mathbb{E}_K[\mathbb{E}_{\boldsymbol{\ell}}(\boldsymbol{\ell}|K)] = \mathbb{E}\boldsymbol{\ell}$$

using Bayes rule. **How can we use this for proving** (3)**?**

Solution: First show the equation

$$\begin{aligned}
\mathbb{E}_k[\mathbb{E}_{\boldsymbol{\ell}}(\boldsymbol{\ell}|K)] &= \mathbb{E}_k \sum_{i=1}^{n} \boldsymbol{\ell}_i P(\boldsymbol{\ell} = \boldsymbol{\ell}_i|K) \\
&= \sum_{j=1}^{m} P(K = k_j) \sum_{i=1}^{n} \boldsymbol{\ell}_i P(\boldsymbol{\ell} = \boldsymbol{\ell}_i|K = k_j) \\
&= \sum_{i=1}^{n} \boldsymbol{\ell}_i \sum_{j=1}^{m} P(K = k_j) P(\boldsymbol{\ell} = \boldsymbol{\ell}_i|K = k_j) \\
&= \sum_{i=1}^{n} \boldsymbol{\ell}_i \sum_{j=1}^{m} P(\boldsymbol{\ell} = \boldsymbol{\ell}_i, K = k_j) \\
&= \sum_{i=1}^{n} \boldsymbol{\ell}_i P(\boldsymbol{\ell} = \boldsymbol{\ell}_i)
\end{aligned}$$

The intuition behind conditioning is that the variable which we condition on is now considered fixed and the randomness is over the remaining variables. Now realize that $\mathbf{w}^{(k)}$ only depends on $i_1, \ldots, i_{k-1}$ (the indices that were chosen in iterations $1, \ldots, k-1$. When we condition on $i_1, \ldots, i_{k-1}$, the only randomness that remains is in i_k. Hence $G(\mathbf{w}^{(k)}) = \nabla f_{i_k}(\mathbf{w}^{(k)})$ conditioned $i_1, \ldots, i_k$ is really a discrete random variable that can take on the possible values (vector values!) $\{\nabla f_1(\mathbf{w}^{(k)}), \ldots, \nabla f_n(\mathbf{w}^{(k)})\}$ with $\mathbf{w}^{(k)}$ fixed! Thus equation (3) follows.

(d) In this question we want to see how the noise of the stochastic gradient $\boldsymbol{\epsilon} = G(\mathbf{w}) - \nabla f(\mathbf{w})$ affects the convergence of SGD compared to GD with noiseless gradients.

Here, we assume $\boldsymbol{\epsilon}$ are independent and zero-mean. As you will see in the homework, for "nice" functions, the convergence of SGD for **constant stepsize** obeys

$$\mathbb{E}\|\mathbf{w}^{(k)} - \mathbf{w}^{\star}\|^2 \leq \beta^k \mathbb{E}[\|\mathbf{w}^{(0)} - \mathbf{w}^{\star}\|_2^2] + c\alpha B^2. \tag{4}$$

with $B^2 = \mathbb{E}\|\boldsymbol{\epsilon}\|^2$ and the constants $\beta < 1$ and c depend on the loss function. Recall the gradient descent convergence rate you derived in HW 7, assuming a suitably chosen constant stepsize:

$$\|\mathbf{w}^{(k)} - \mathbf{w}^{\star}\|_2^2 \leq \beta^{2k} \|\mathbf{w}^{(0)} - \mathbf{w}^{\star}\|_2^2 \tag{5}$$

where β depends on the minimum and maximum eigenvalues of the Hessian.

Describe what the repercussions are of using noisy gradients by comparing the two bounds (4) **and** (5)**. Where does SGD converge to for constant stepsize? Does the squared error of the estimator $\|\mathbf{w}^{(k)} - \mathbf{w}^{\star}\|^2$ strictly decrease with each iteration?**

Solution: This shows that compared to gradient descent, the noise of the gradient

- contributes to the final error if the stepsize does not decay. That means that the distance to the optimal solution might decrease but then the iterate wonders around in the neighborhood of radius $c\alpha B^2$. There is no guarantee that it actually converges to some fixed point for $k \to \infty$! Note that if stepsizes decay and it thus converges to optimum (as shown in HW 8) with no final error, bound (4) does not hold anymore and we only have $1/k$ convergence

- only allows convergence guarantees (and strict decrease) of the **expected** error. This means that when you plot the training error of one run of SGD, for some iterations, the distance to the optimum may actually increase from one iteration to the next.

Additional clarifications In lecture we assumed $\mathbb{E}\|G(\mathbf{w}^{(k)})\|^2 \leq M^2$ for all k and achieved $\log k/k$-convergence for strongly convex functions using decaying stepsize. In general we do not have strict boundedness, but the following more general condition holds for all $\mathbf{w} \in \mathbb{R}^d$:

$$\mathbb{E}\|G(\mathbf{w})\|_2^2 \leq M_g^2\|\mathbf{w} - \mathbf{w}^\star\|^2 + B^2. \tag{6}$$

When this general assumption (6) is fulfilled, one can still show $1/k$-convergence to the optimum for decaying stepsize (see homework).

For your interest, here is a table with convergence rates for corresponding step sizes (that need to be suitably chosen for each method) that might be good to have in mind. For gradient descent methods we additionally assume that the function is differentiable and smooth, that is

$$\langle \nabla f(x) - \nabla f(x'), \mathbf{x} - \mathbf{x}' \rangle < M\|\mathbf{x} - \mathbf{x}'\|^2$$

for some constant M, while for the stochastic gradient methods we assume that condition (6) is fulfilled (where the constants M_g and B also depend on the smoothness constant M of f).

Algorithm	Assumption	Step Size	Convergence Rate
GD	strongly convex, smooth	constant	$O(\beta^k)$
GD	convex, smooth	constant	$O(1/k)$
SGD	strongly convex, $B = 0$	constant	$O(\beta^k)$
SGD	strongly convex, $B \neq 0$	$O(1/k)$	$O(1/k)$
SGD	only convex, $B \neq 0$	$O(1/\sqrt{k})$	$O(1/\sqrt{k})$

2 Computational complexity of SGD vs. GD

Let us consider a simple least squares problem, where $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{w}, \mathbf{y} \in \mathbb{R}^d$ and we are interested in optimizing the function

$$F(\mathbf{w}) = \frac{1}{2n}\|\mathbf{A}\mathbf{w} - \mathbf{y}\|_2^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(\mathbf{a}_i^\top\mathbf{w} - y_i)^2.$$

(a) Write down the gradient descent update. From HW 7 Problem 3 we know that:

$$d_k \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^{2k} d_0$$

where, $d_k = \|\mathbf{w}^{(k)} - \mathbf{w}^\star\|_2^2$ and $\kappa = \frac{\lambda_{max}(\mathbf{A}^\top\mathbf{A})}{\lambda_{min}(\mathbf{A}^\top\mathbf{A})}$ denotes the condition number of $\mathbf{A}^\top\mathbf{A}$. **Show that $dn\kappa\log(1/\epsilon)$ is the time complexity of computing an ϵ optimal solution.**

Solution: For gradient descent, we have the update

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \frac{\gamma}{n}\mathbf{A}^\top(\mathbf{A}\mathbf{w}^{(k)} - \mathbf{y}).$$

Using the hint, we therefore obtain geometric convergene to the optimum,

$$d_T \leq \left(\frac{\kappa-1}{\kappa+1}\right)^{2T} d_0$$

$$d_T \leq \left(1 - \frac{2}{\kappa+1}\right)^{2T} d_0 \sim e^{-\frac{4T}{\kappa+1}} d_0 \text{ using the approximation } 1 - x \sim e^{-x}$$

$$e^{-\frac{4T}{\kappa+1}} d_0 = \epsilon \Rightarrow \log\epsilon \sim -\frac{T}{\kappa}$$

and the number of iterations is roughly $T \sim \kappa\log(1/\epsilon)$ to converge to within ϵ of optimum. Note that during each iteration, we perform work dn, since $\mathbf{A}\mathbf{w}^{(k)}$ takes dn time to compute, and performing the multiplication $\mathbf{A}^\top(\mathbf{A}\mathbf{w}^{(k)} - \mathbf{y})$ takes dn time as well. So the total time complexity is $dn\kappa\log(1/\epsilon)$.

(b) **Show that the regularization term in ridge regression improves the time complexity of computing an ϵ optimal solution?**

Solution: The update in ridge regression for gradient descent is

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \gamma(\frac{1}{n}\mathbf{A}^\top(\mathbf{A}\mathbf{w}^{(k)} - \mathbf{y}) + \lambda_{ridge}\mathbf{w}^{(k)}).$$

And the condition number is $\kappa = \frac{\lambda_{max}(\mathbf{A}^\top\mathbf{A})+\lambda_{ridge}}{\lambda_{min}(\mathbf{A}^\top\mathbf{A})+\lambda_{ridge}}$. Notice that without ridge regularization, if $\lambda_{min}(\mathbf{A}^\top\mathbf{A})$ is very small the condition number becomes really big. Adding the regularization terms reduces the number of iterations (and hence overall time complexity) since the number of iterations has a linear dependency in terms of the condition number $T \sim \kappa\log(1/\epsilon)$.

(c) Write down the stochastic gradient descent update. In HW 8 you will show that the following bound holds for the SGD iterate $\mathbf{w}^{(k)}$ at time step k updated using a constant stepsize

$$\Delta_k \leq (1 - 2\alpha m)^k\Delta_0 + \alpha\frac{B^2}{m}$$

where, $\Delta_k = \mathbb{E}\left[\|\mathbf{w}^{(k)} - \mathbf{w}^\star\|_2^2\right]$.

Show that $\frac{d}{\epsilon}\log(1/\epsilon)$ is the time complexity of computing an ϵ optimal solution. How does this compare with the complexity of gradient descent in section (a)? Hint: You may choose α as a function of B and ϵ.

Solution: We have the update equation

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha \mathbf{a}_{i_k}(\mathbf{a}_{i_k}^\top \mathbf{w}^{(k)} - y_{i_k}),$$

where i_k is chosen uniformly at random from the set $\{1, 2, 3, \ldots, n\}$.

Given that

$$\Delta_T \leq (1 - 2\alpha m)^T \Delta_0 + \alpha \frac{B^2}{m}.$$

Now if we want the LHS to be less than ϵ, it suffices to set each of the above terms to be less than $\epsilon/2$. In particular, we have the relations

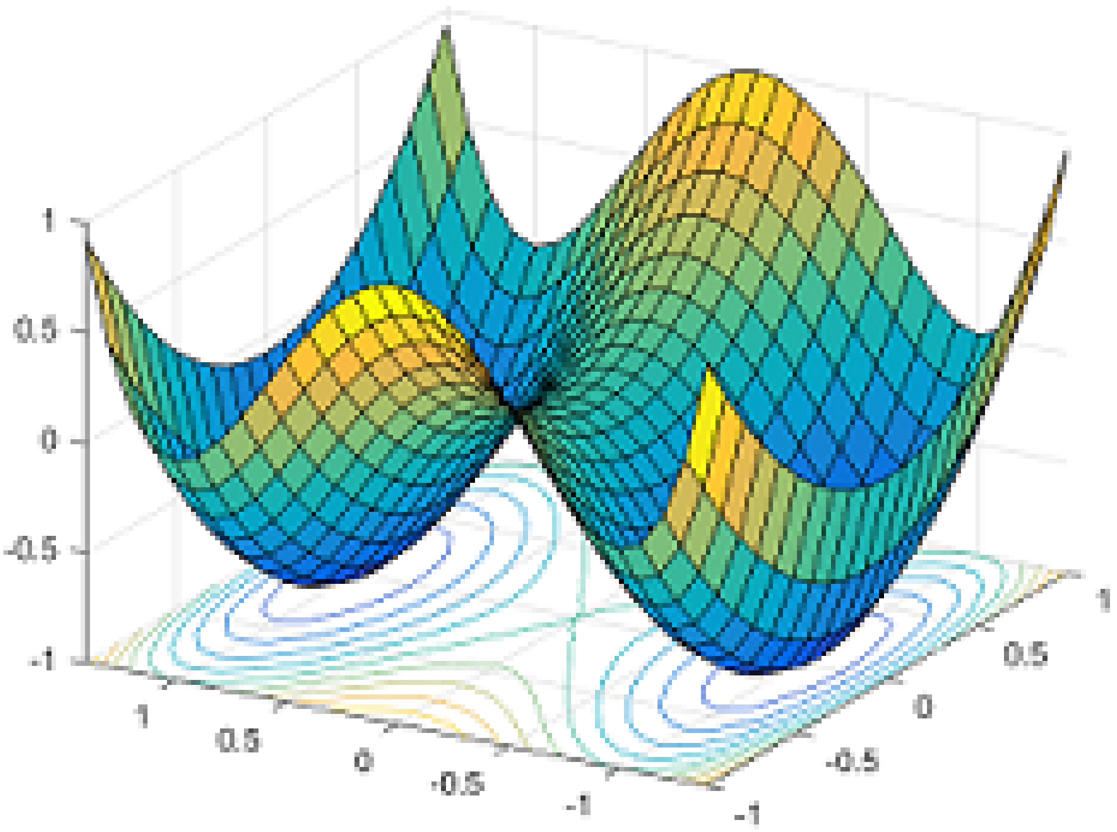
$$\alpha \frac{B^2}{m} = \epsilon/2 \text{ and}$$
$$(1 - 2\alpha m)^T \Delta_0 = \epsilon/2.$$

Again using the approximation $\exp\{-x\} \sim 1 - x$ we get that $\log \frac{1}{1-\alpha m} \sim \alpha m$. Doing some algebra, we are led to the choices $\alpha = \frac{\epsilon m}{2B^2}$, and

$$\begin{aligned} (1 - 2\alpha m)^T &= \frac{\epsilon}{2\Delta_0} \\ e^{-2T\alpha m} &= \frac{\epsilon}{2\Delta_0} \\ T &= \frac{1}{2\alpha m} \log\left(\frac{2\Delta_0}{\epsilon}\right) \\ T &= \frac{B^2}{\epsilon m^2} \log\left(\frac{2\Delta_0}{\epsilon}\right) \end{aligned}$$

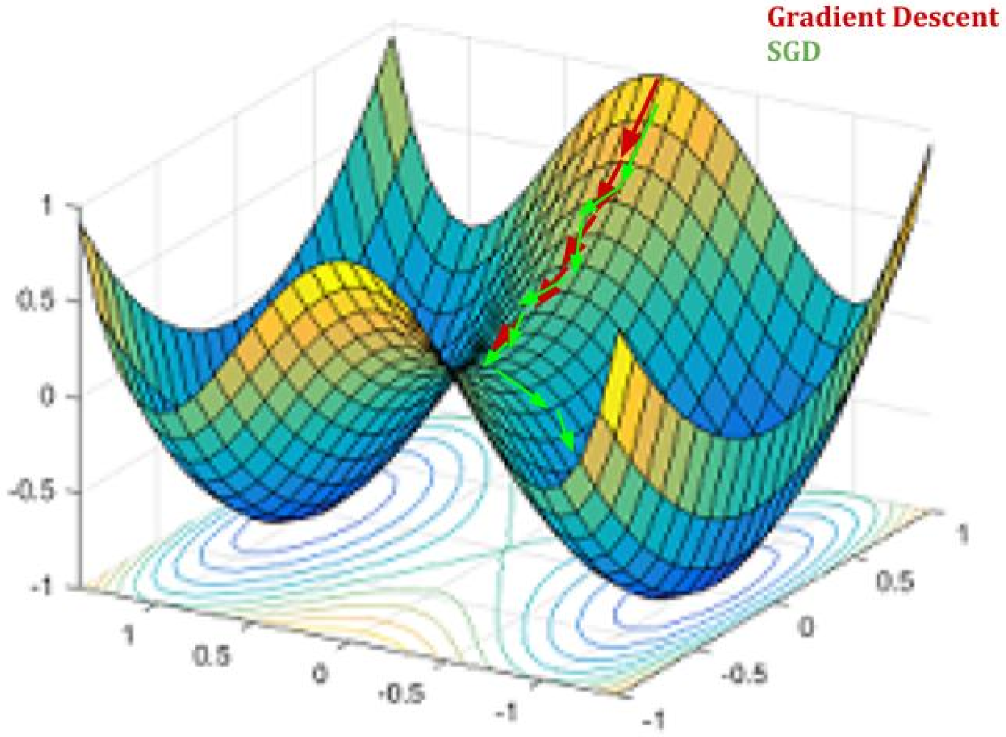
In effect, we converge in $T \sim \frac{1}{\epsilon} \log(1/\epsilon)$ iterations, and each iteration takes $O(d)$ time (why?). Comparing SGD and GD, the quantities are $dn \log(1/\epsilon) \sim \frac{d}{\epsilon} \log(1/\epsilon)$. In other words, SGD provides gains in convergence when $n \sim 1/\epsilon$, i.e., when we have sufficiently many samples.

(d) Comment on the ability of GD and SGD to escape saddle points and draw it qualitatively on the figure below.



Solution:

Note that near saddle points you can find the presence of ”plateaus” (in other words, the saddle is not steep) - in these cases, taking too small of steps would indeed result in premature convergence before having escaped the saddle region. For gradient descent if the gradient direction is pointing in the same direction as the saddle point, then you risk getting stuck there. If you perform SGD instead, you take all the steps one after the other, but if you’re still moving in a single direction, you can reach the saddle point and find that the gradient on all sides is fairly flat and the step size is too small to go over this flat part. If the fluctuations occur only along one dimension, with the steps getting smaller and smaller, it would converge at the saddle point, as well. SGD however can sometimes break out of simple saddle points, if the fluctuations are along other directions, and if the step size is large enough for it to go over the flatness.


Gradient Descent
SGD
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1