确定性优化方法，第二章已有，不再赘述，除了第二章内容，确定性优化还有准牛顿方法，使用**矩阵近似表达黑森矩阵的逆**

随机梯度下降SGD

一次只使用单个示例的优化算法有时称为随机方法，有时称为**在线方法**。

- E.g., consider the cost function of linear regression as

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- Now, we ignore the superscript i, then for each x we have ,

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x)-y)^2 \\
&= 2\cdot\frac{1}{2}(h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x)-y) \\
&= (h_\theta(x)-y)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \qquad \boldsymbol{x}\in\mathbb{R}^n \\
&= (h_\theta(x)-y)\,x_j
\end{aligned}$$

SGD算法：

Loop {

for i=1 to m, {

$$\theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right)x_j^{(i)} \qquad (\text{for every } j).$$

}

}

SGD

SGD可能永远不会“收敛”到最小值，并且参数θ将保持在J（θ）的最小值附近振荡；
但在实践中，大多数接近最小值的值都是真正最小值的合理好的近似值。因此，当**训练集很大**时，SGD通常比批量梯度下降更可取。

不仅仅使用一个数据，使用少量数据（多于一个，少于全部）也称为随机梯度下降

Stochastic Gradient Descent

- Most algorithms (called *minibatch* or *minibatch stochastic* methods) fall somewhere in between, using more than one but less than all of the training examples.
- **Note**: Confusing again..... It is now common to simply call them *stochastic* methods.

Algorithm 8.1 Stochastic gradient descent (SGD) update at training iteration k

Require: Learning rate ϵ_k.
Require: Initial parameter $\boldsymbol{\theta}$
while stopping criterion not met **do**
  Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
  Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow +\frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
  Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon\hat{\boldsymbol{g}}$
end while

批梯度下降BGD

批量梯度下降必须扫描整个训练集，然后再进行一步——如果m很大，这是一项成本高昂的操作。

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every } j\text{)}.$$

加速SGD的方法

Polyak's Classical Momentum，经典动量

经典动量（CM）积累了过去梯度的指数衰减移动平均值，并继续朝着它们的方向移动。

Letting η be the learning rate.

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{v}_{t+1}$$

$$\boldsymbol{v}_{t+1} = \mu\boldsymbol{v}_t - \eta\nabla_{\boldsymbol{w}_t} f$$

μ为0时就是梯度下降

速度矢量vt：一个存储器，用于累积在前t步中选择的减速方向。

v的影响由动**量系数μ∈[0,1]**控制。μ通常略小于1。当μ=0时：这只是梯度下降。

经典动量存在的问题：可能动量积累过大，导致在极值点梯度仍然不为0

Nesterov's Accelerated Gradient

- The update equations of NAG are:

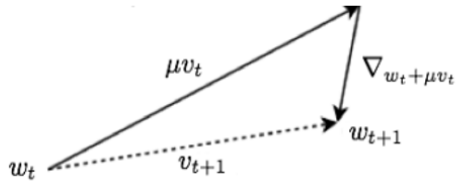
$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{v}_{t+1}$$

$$\boldsymbol{v}_{t+1} = \mu\boldsymbol{v}_t - \eta\nabla_{\boldsymbol{w}_t+\mu\boldsymbol{v}_t} f$$

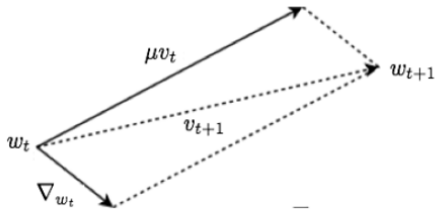
CM

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{v}_{t+1}$$

$$\boldsymbol{v}_{t+1} = \mu\boldsymbol{v}_t - \eta\nabla_{\boldsymbol{w}_t} f$$



NAG



CM

Illustration of the comparison between CM and NAG.

首先在上一个累积梯度的方向上做一个大的跳跃。然后测量**你结束的梯度**并进行校正。

CM→在w_t的当前迭代处检查梯度

忠实信任当前迭代；

NAG→在wt+μvt处检测梯度。

对当前迭代不太信任，并按照速度向量建议的方向向前看。

微小的差异使NAG能够更快、更稳定地适应。

自适应的学习率

研究人员早就意识到，**学习率是最难设置的超参数**，因为它对模型性能有重大影响。

- Adagrad
- RMSProp
- Adam

Adagrad

设置全局学习率之后，每次通过，**全局学习率逐参数的除以历史梯度平方和的平方根**，使得每个参数的学习率不同

在参数空间更为平缓的方向，会取得更大的进步（因为平缓，所以历史梯度平方和较小，对应学习下降的幅度较小），并且能够使得陡峭的方向变得平缓，从而加快训练速度。***平缓加速，陡峭减速***

弱点：由于每个增加的项都是正的，累积的总和不断增长，这反过来又导致**学习率下降，最终变得无限小**。需要手动设置学习率

squared values

$$g_{t,i} = \nabla_{\theta_t} J(\theta_{t,i})$$

The SGD update

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i}$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

$G_t \in \mathbb{R}^{d \times d}$ is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradients w.r.t. θ_i up to time step t.

Adadelta

Adadata是Adagrad的延伸，旨在降低其激进、单调递减的学习率。Adadelta不是累积所有过去的平方梯度，而是将**累积的过去梯度的窗口限制为某个固定大小w。**

RMSprop

RMSprop is an unpublished, adaptive learning rate method proposed by Geoff Hinton in his Coursera Class. RMSprop and Adadelta have both been developed independently around the same time to resolve Adagrad's radically diminishing learning rates.

衰减速率：可调超参

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

采用指数衰减平均，以丢弃遥远过去的历史。

Adam

Adam存储了过去平方梯度v_t（方差）的指数衰减平均值，如Adadeta和RMSprop。

Adam还保持了过去梯度的指数衰减平均值m_t（均值），类似于动量