Gradient Descent

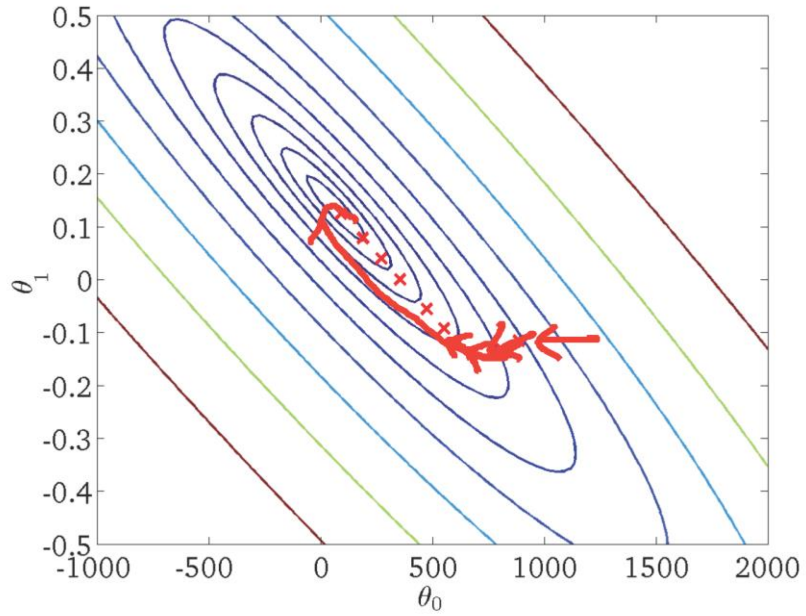
Batch Gradient Descent

Initialize θ

Repeat {

$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) - y_i\right) x_{ij}}_{\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})} \qquad \text{for } j = 0 \ldots d$$

}



Stochastic Gradient Descent

Initialize θ

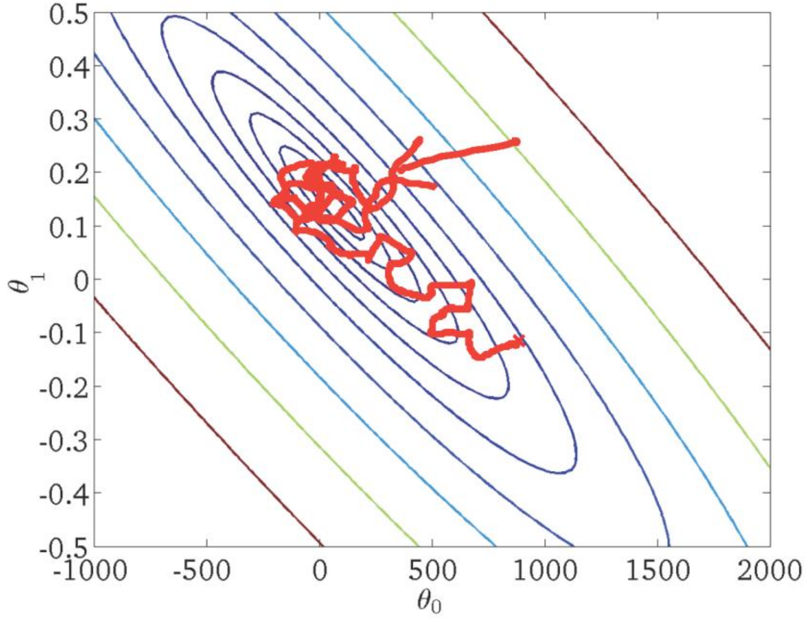
Randomly shuffle dataset

Repeat { (Typically 1 – 10x)

For $i = 1 \ldots n$, *do*

$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) - y_i\right) x_{ij}}_{\frac{\partial}{\partial \theta_j} \mathrm{cost}_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i)} \qquad \text{for } j = 0 \ldots d$$

}



Adaptive alpha is not required for hw3 (Just follow the previous slide for hw3)

Stochastic Gradient Descent

Initialize θ

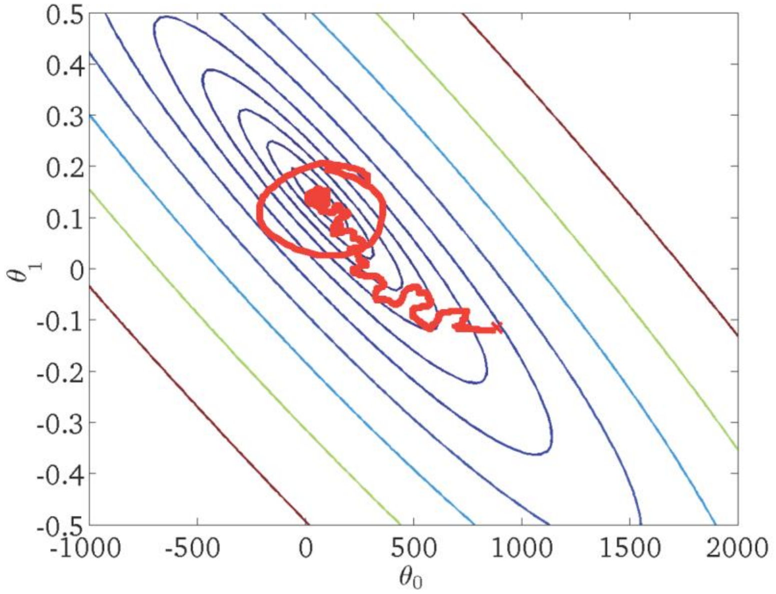
Randomly shuffle dataset

Repeat { (Typically 1 – 10x)

For $i = 1...n$, *do*

$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) - y_i\right) x_{ij}}_{\frac{\partial}{\partial \theta_j} \text{cost}_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i)} \quad \text{for } j = 0...d$$

}



Learning rate α is typically held constant. Can slowly decrease α over time if we want θ to converge. (E.g. $\alpha = \frac{\texttt{const1}}{\texttt{iterationNumber + const2}}$)