EE488 Special Topics in EE
<Deep Learning and AlphaGo>

Sae-Young Chung

Lecture 4

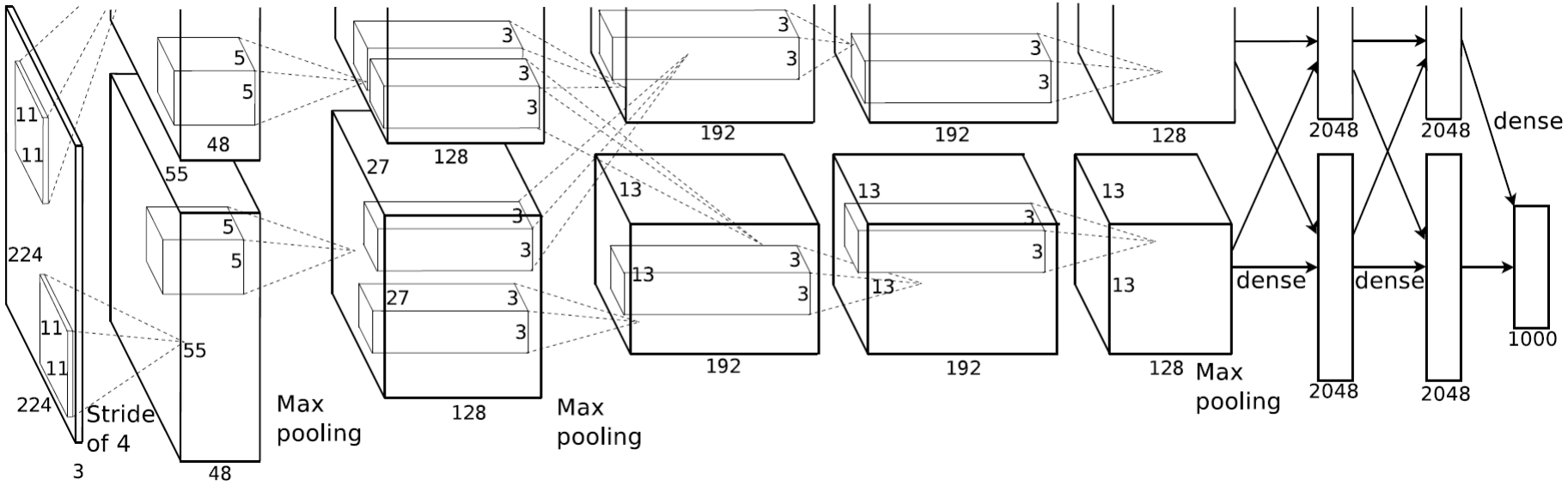
September 11, 2017

Chap. 4 Numerical Computation

- Gradient
- Hessian
- Critical points
- Saddle points
- Gradient descent
- Newton's method
- Constrained optimization

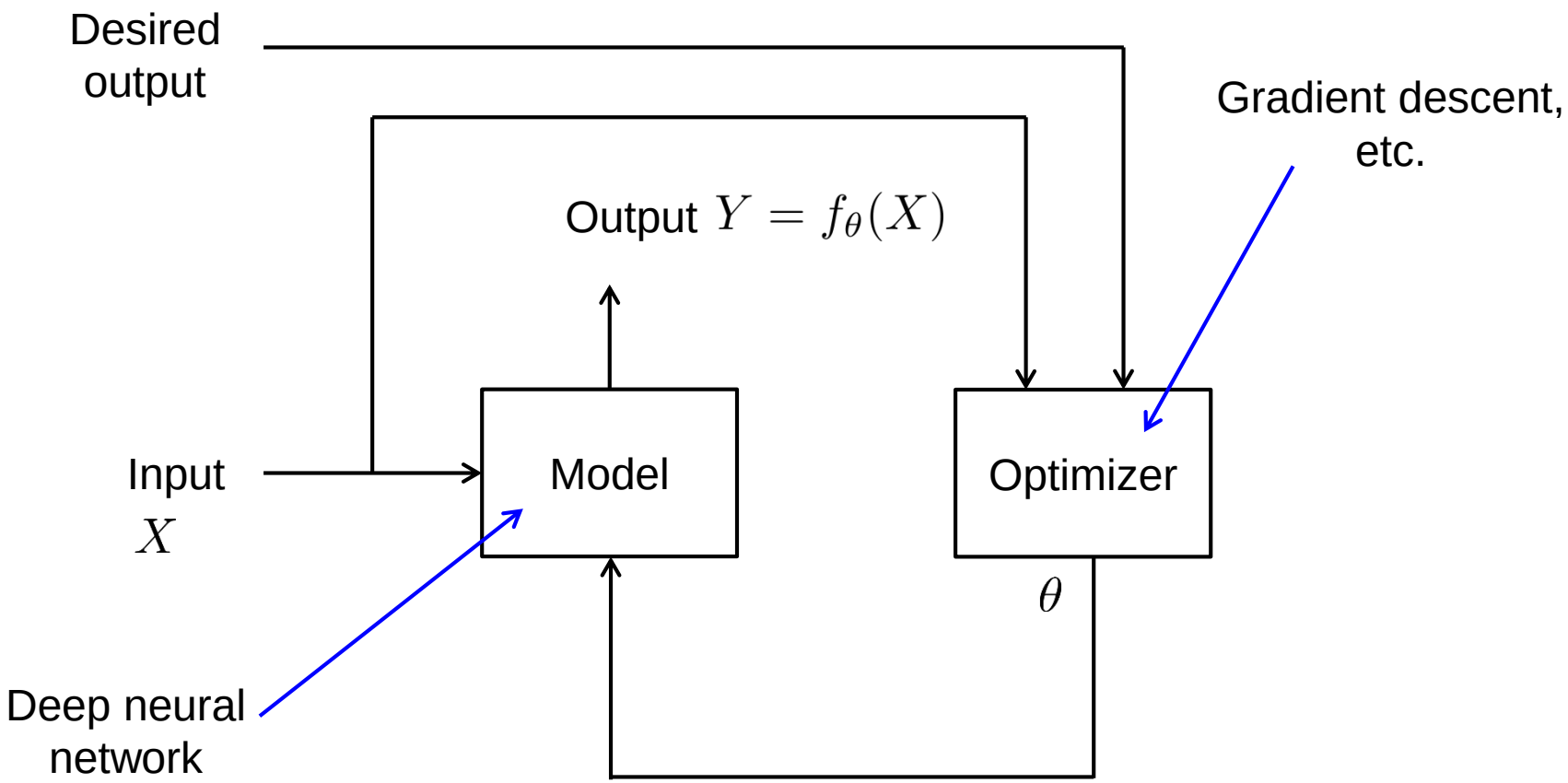
CNN for Image Classification

- AlexNet: 60 million parameters and 650,000 neurons
- Trained using ~ 1 million images
- 1,000 categories



Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, “ImageNet classification with deep convolutional neural networks”, NIPS 2012

(Supervised) Learning System



Optimization

- Optimization problem, where $f : \mathbb{R}^n \to \mathbb{R}$, $\theta \in \mathbb{R}^n$

$$\min_{\theta} f(\theta)$$

- Example of a large-scale optimization problem

$$\min_{\theta} \sum_{k=1}^{N} \|\mathbf{y}_k - f_\theta(\mathbf{x}_k)\|^2$$

  - N: number of training examples (e.g., $1,000,000$)
  - $\mathbf{x}_k$: k-th input image (e.g., $\mathbf{x}_k \in \mathbb{R}^{224\times224\times3}$)
  - $\mathbf{y}_k$: correct label vector for the k-th input image (e.g., $\mathbf{y}_k \in \mathbb{R}^{1,000}$)
  - θ: set of parameters in the neural network (e.g., $\theta \in \mathbb{R}^{60,000,000}$)
  - $f_\theta(\mathbf{x}_k)$: the output of the neural network with θ when the input is $\mathbf{x}_k$

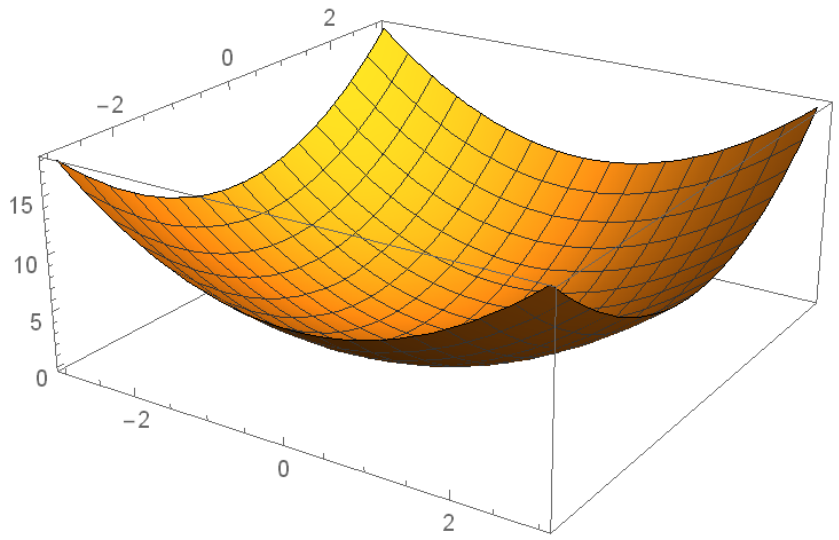
Gradient, Hessian, Taylor Series

- Taylor series approximation of $f : \mathbb{R}^n \to \mathbb{R}$

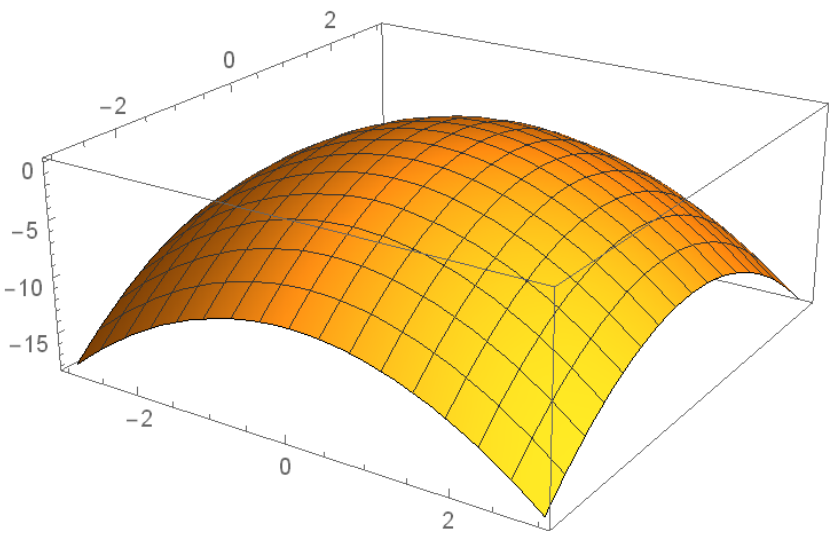
$$f(\mathbf{x}) \sim f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \nabla^2 f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

- $\nabla_{\mathbf{x}} f(\mathbf{x}_0) = \left(\frac{\partial}{\partial x_1} f, \ldots, \frac{\partial}{\partial x_n} f \right)^T \Big|_{\mathbf{x}_0} \in \mathbb{R}^n$: gradient vector
  - Will drop $\mathbf{x}$ in $\nabla_{\mathbf{x}} f(\mathbf{x}_0)$ if there is no confusion
  - If $\nabla f(\mathbf{x}) = \mathbf{0}$, then $\mathbf{x}$ is called a critical point or a stationary point.
- $\nabla^2 f(\mathbf{x}_0)$: $n \times n$ Hessian matrix evaluated at $\mathbf{x}_0$ whose (i, j)-th element is $\frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}_0)$
  - Will sometimes use $H(\mathbf{x}_0)$ or H instead of $\nabla^2 f(\mathbf{x}_0)$
- If all second-order partial derivatives exist and are continuous, then $\frac{\partial^2}{\partial x_i \partial x_j} f = \frac{\partial^2}{\partial x_j \partial x_i} f$ and Hessian is symmetric.
  - All eigenvalues are real and $H = Q \Lambda Q^T$, where Q is an orthogonal matrix and Λ is a diagonal matrix containing the eigenvalues of H.

Critical Points



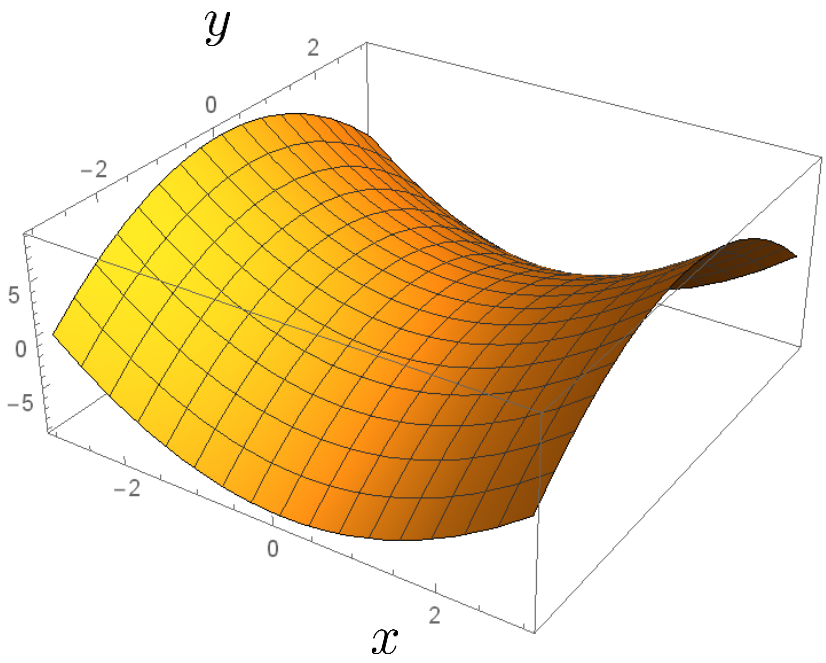
Minimum: all eigenvalues of H are positive at the critical point

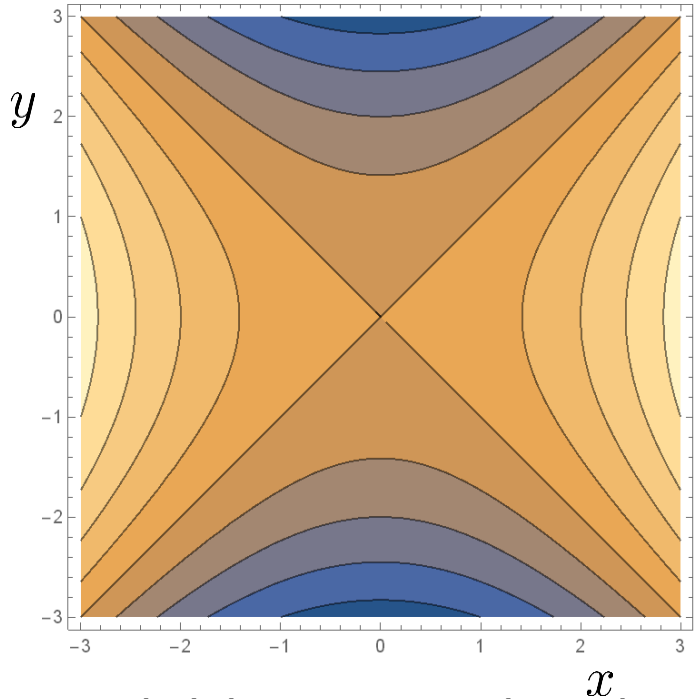


Maximum: all eigenvalues of H are negative at the critical point

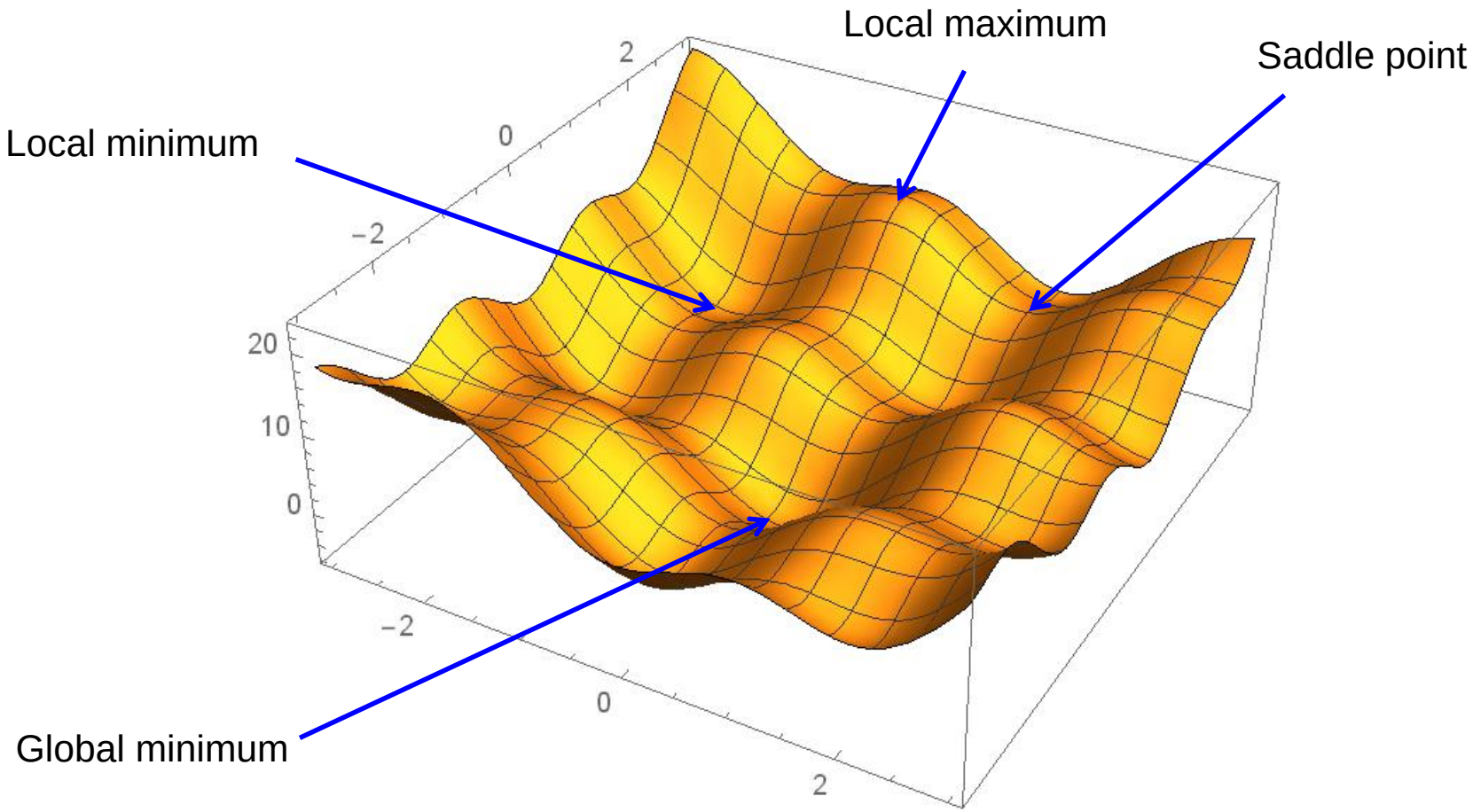
Saddle Point

$$f(x, y) = x^2 - y^2$$





Saddle point: a critical point that is not a local extreme (minimum or maximum). For example, an eigenvalue of H is positive and the other is negative at $(x, y) = (0,0)$ as shown above. Another example: $f(x) = x^3$.


Local maximum
Saddle point
Local minimum
Global minimum
2
0
−2
20
10
0
−2
0
2

Necessary Conditions for Optimality

- Optimization problem, where $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

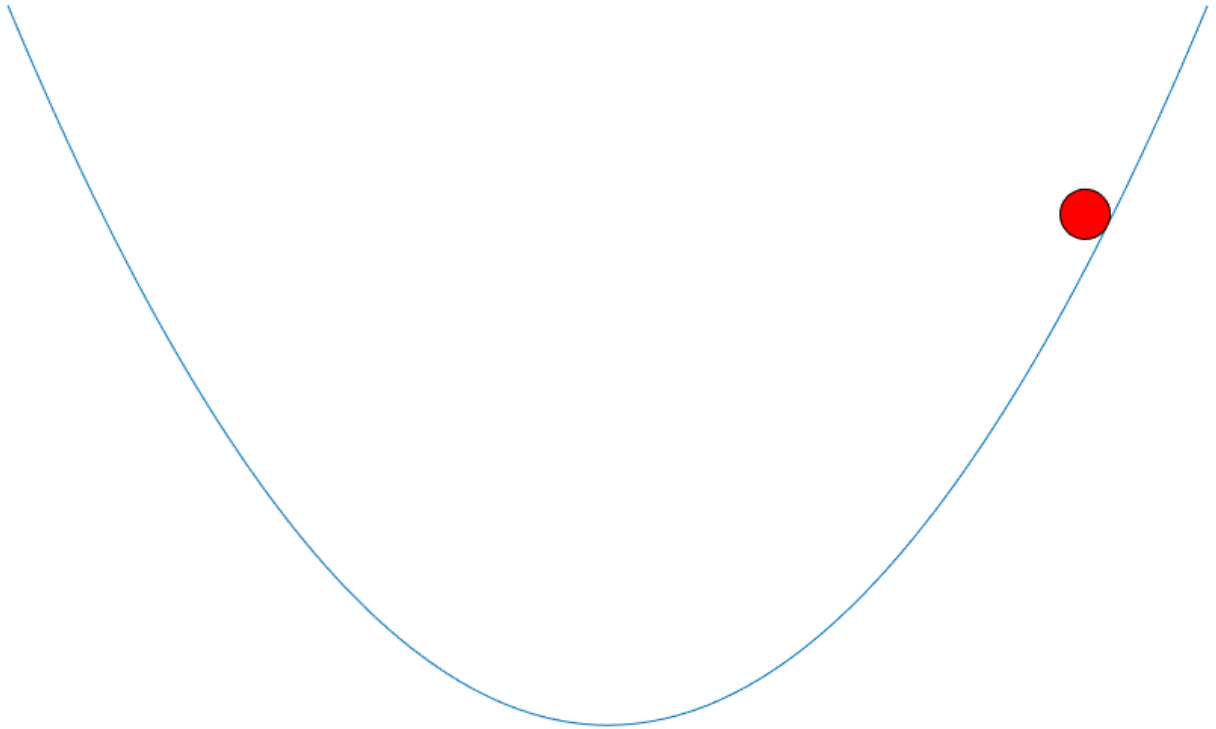
- Assume f is continuously differentiable (differentiable and their derivatives are continuous). If $\mathbf{x}^*$ is a local minimum of (1), then
  - $\nabla f(\mathbf{x}^*) = 0$
- If, furthermore, f is twice continuously differentiable, then
  - $\mathbf{u}^T \nabla^2 f(\mathbf{x}^*)\mathbf{u} \geq 0$ for all $\mathbf{u} \in \mathbb{R}^n$, i.e., $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite

Sufficient Conditions for Optimality

- Assume f is twice continuously differentiable
- $\mathbf{x}^*$ is a local minimum if
  - $\nabla f(\mathbf{x}^*) = 0$ and
  - $\mathbf{u}^T \nabla^2 f(\mathbf{x}^*) \mathbf{u} > 0$ for all $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{u} \neq 0$, i.e., $\nabla^2 f(\mathbf{x}^*)$ is positive definite

Principle of Minimum Energy

~ Second law of thermodynamics



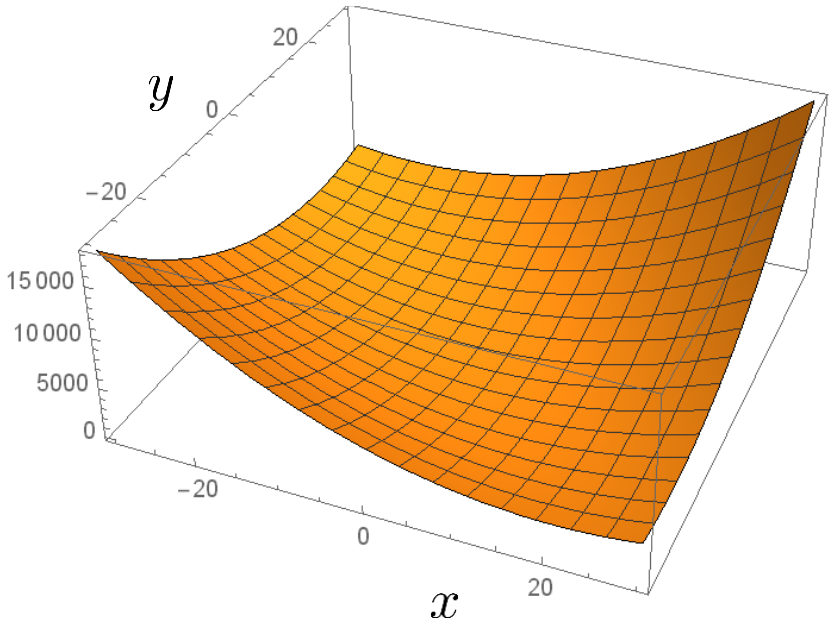
Gradient Descent

- Gradient descent (or steepest descent): a simple algorithm to find a local minimum
- Go downhill such that the gradient is the steepest, i.e.,

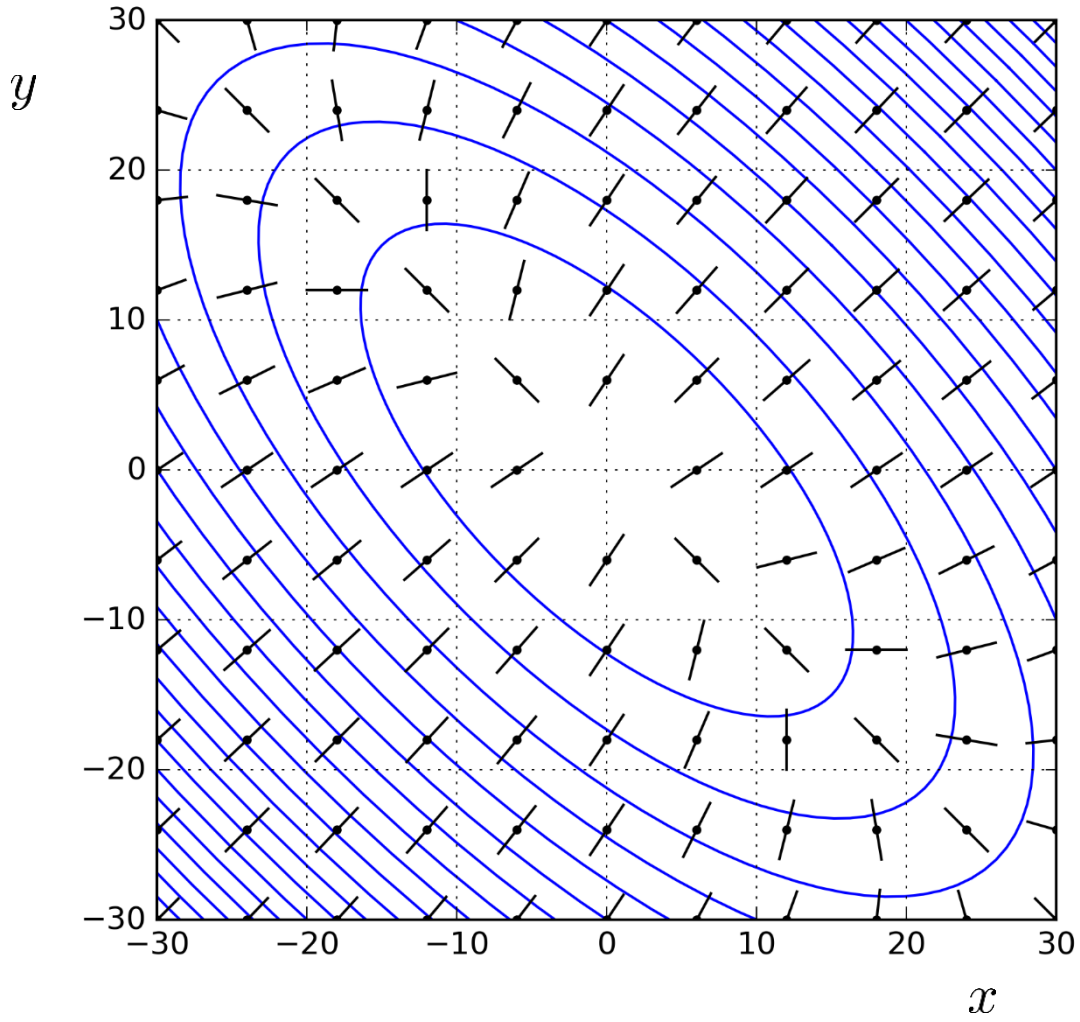
$$\mathbf{x}' = \mathbf{x} - \epsilon \nabla f(\mathbf{x})$$

- ϵ: learning rate

$$f(x, y) = 5(x + y)^2 + (x - y)^2$$

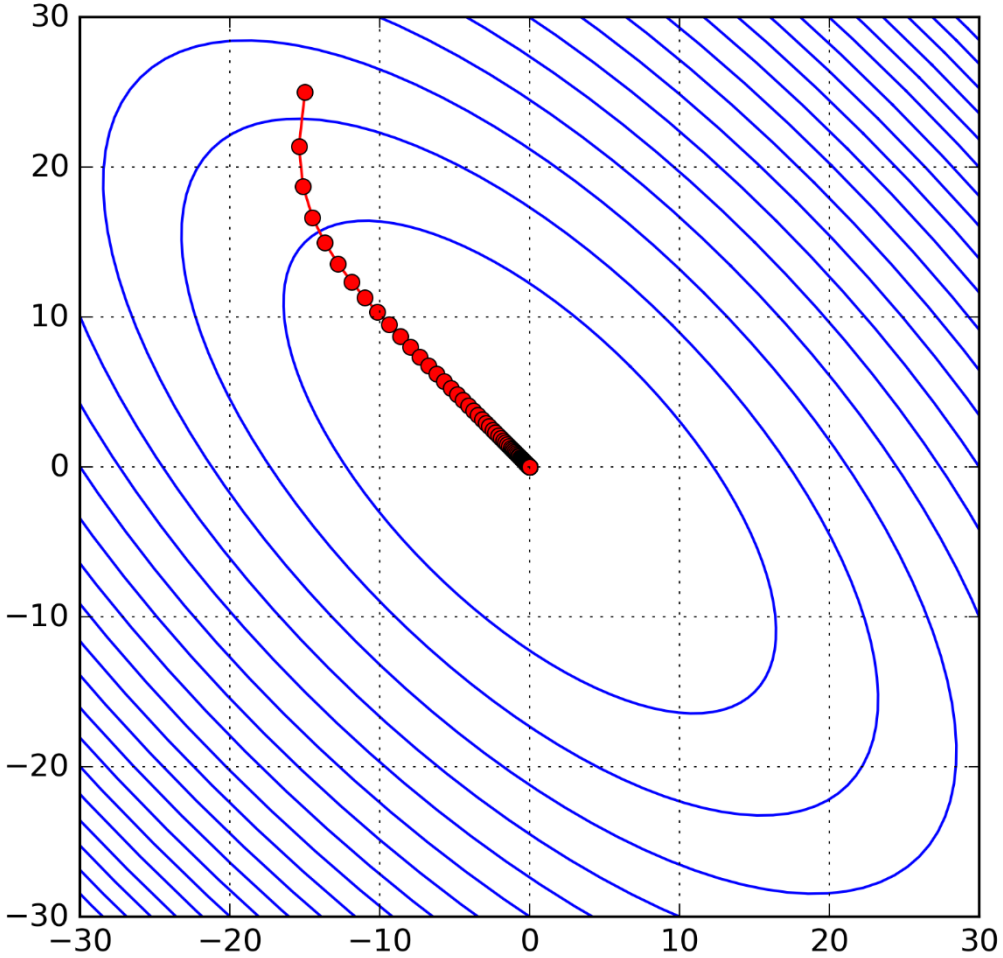


$$\nabla f(x,y) = (10(x+y) + 2(x-y), 10(x+y) - 2(x-y))^T$$

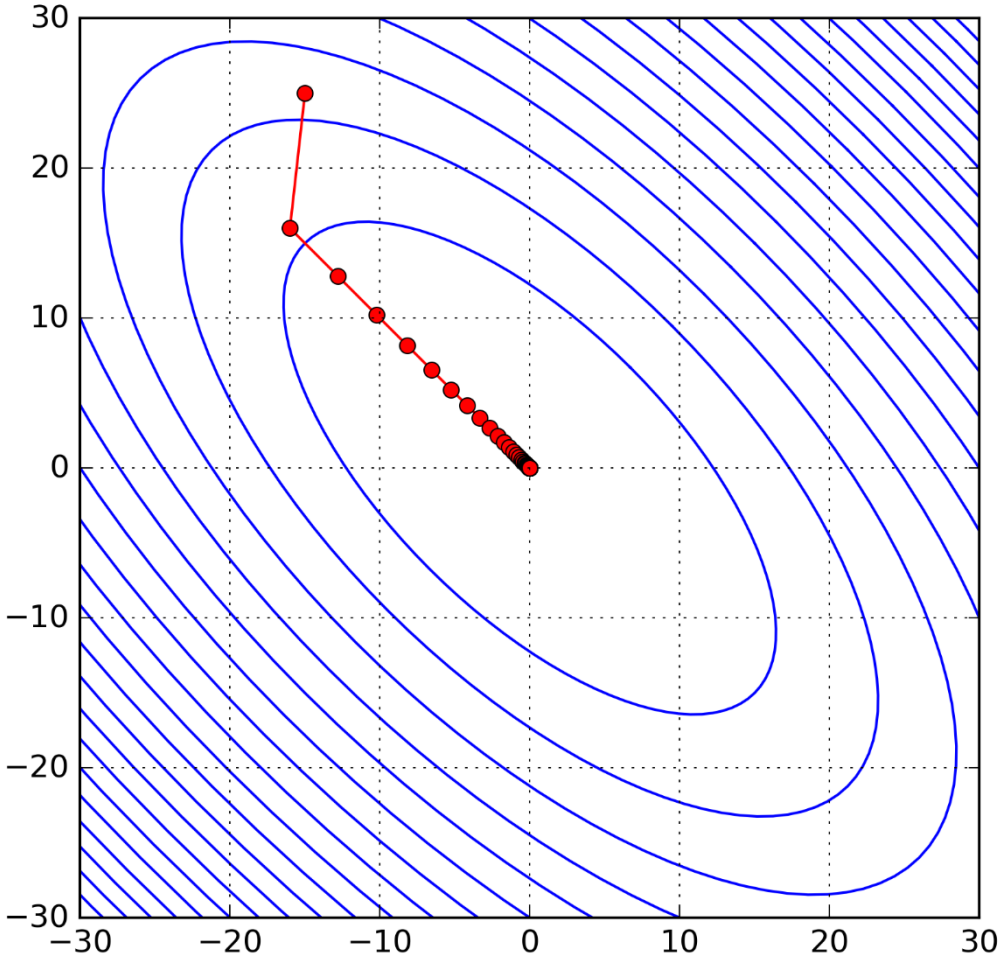


GD with Small Step Size

$\epsilon = 0.02$

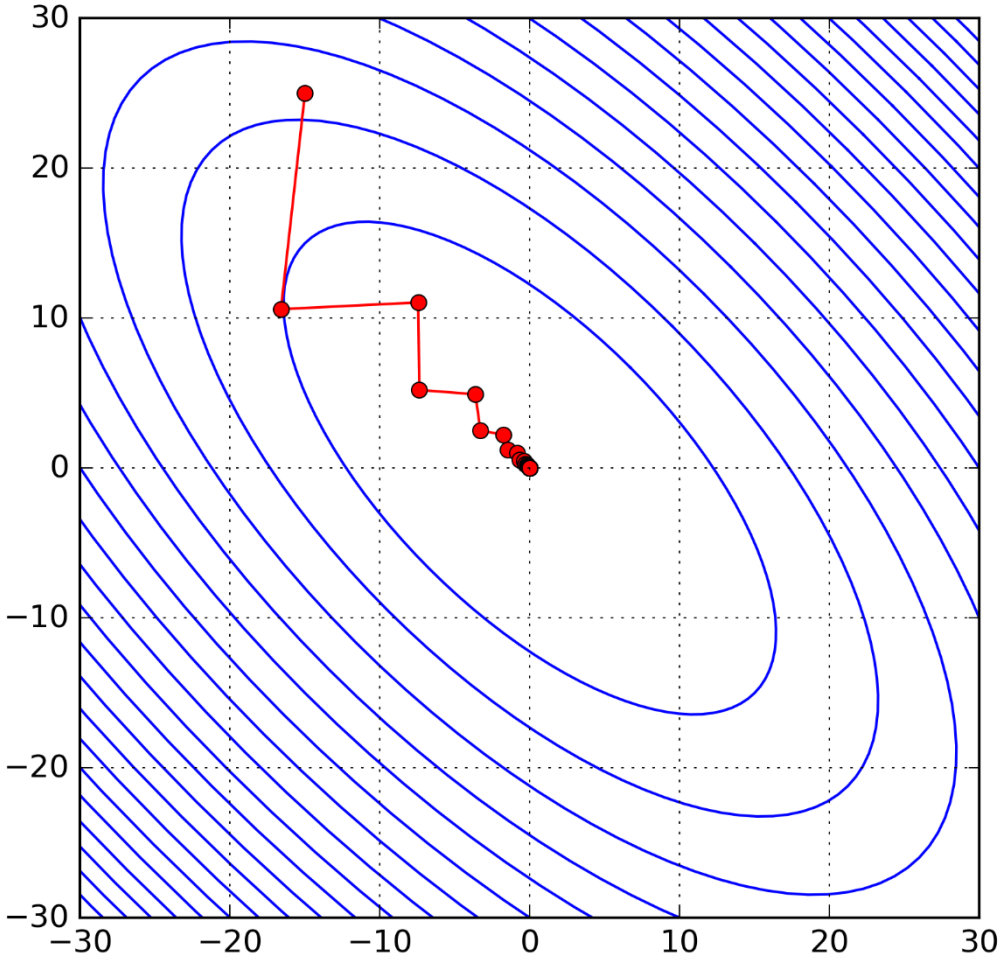


$\epsilon = 0.05$



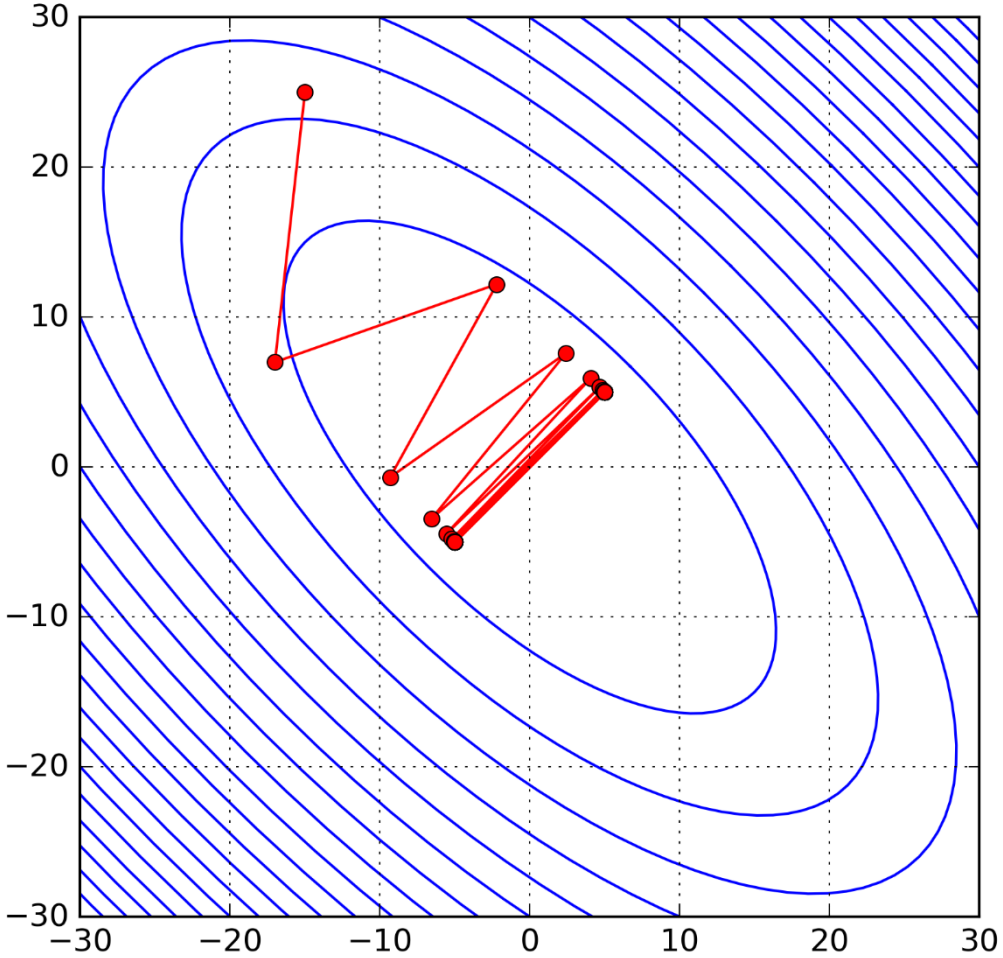
Good Step Size

$\epsilon = 0.08$



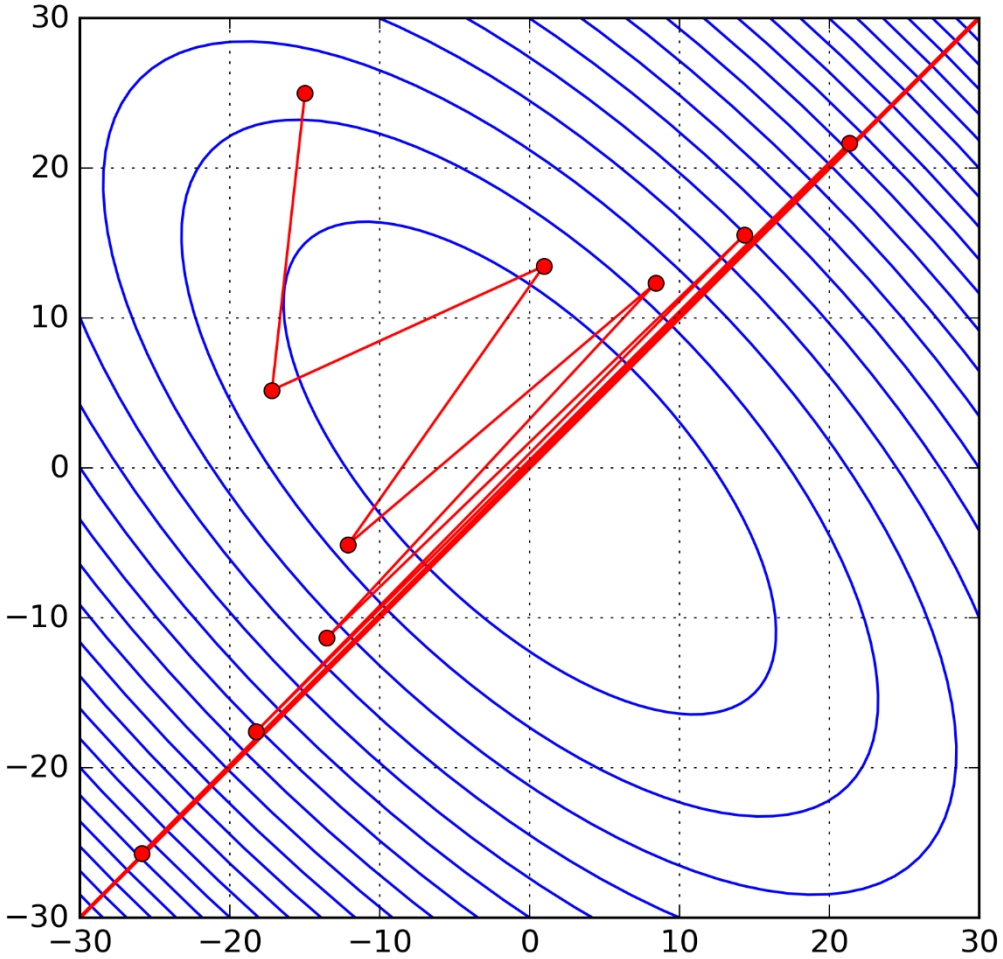
Oscillatory Behavior

$\epsilon = 0.1$



Step Size Too Big

$\epsilon = 0.11$



Optimal Learning Rate

- Taylor series approximation of $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\mathbf{x}) \sim f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T H (\mathbf{x} - \mathbf{x}_0)$$

- Assume $\mathbf{x} = \mathbf{x}_0 - \epsilon \mathbf{g}$, where $\mathbf{g} = \nabla f$. Then,

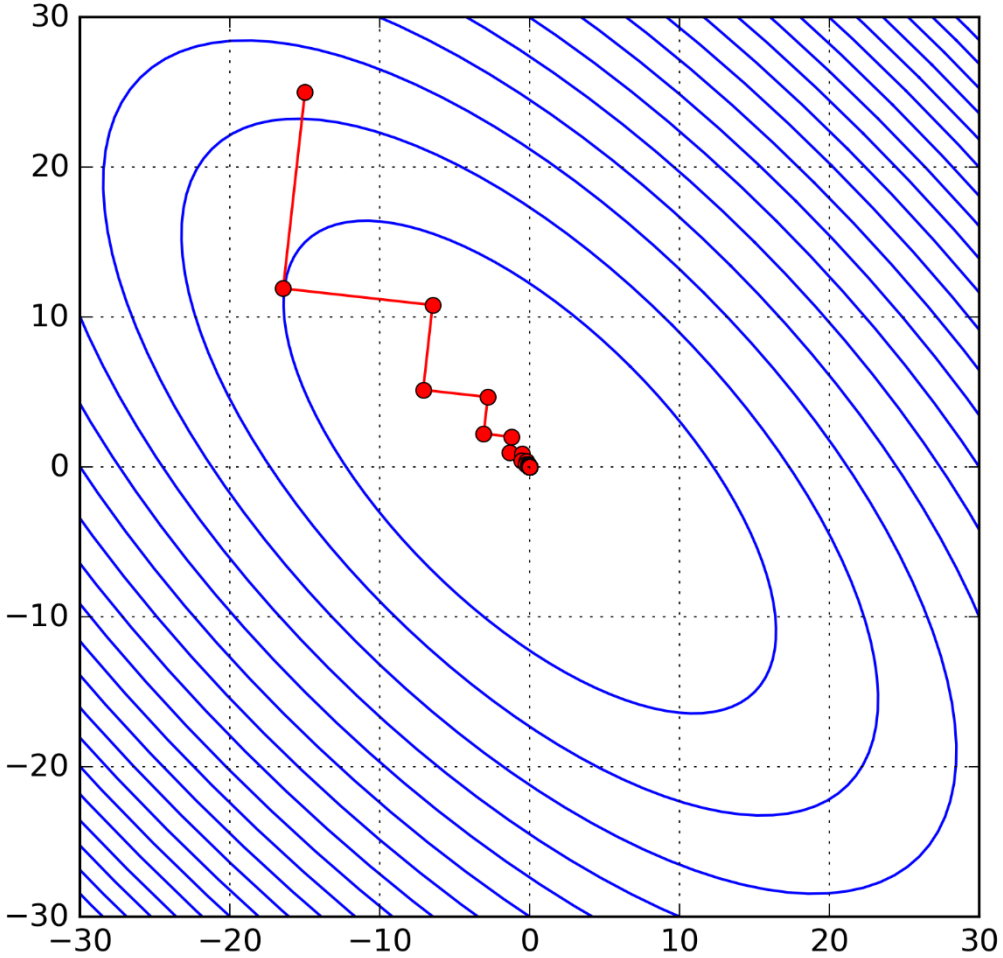
$$\begin{aligned} f(\mathbf{x}_0 - \epsilon \mathbf{g}) &\sim f(\mathbf{x}_0) - \epsilon \mathbf{g}^T \mathbf{g} + \frac{1}{2}\epsilon^2 \mathbf{g}^T H \mathbf{g} \\ &= \frac{1}{2}\mathbf{g}^T H \mathbf{g} \left(\epsilon - \frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T H \mathbf{g}} \right)^2 + c \end{aligned}$$

- Thus, to minimize this Taylor-series approximation, we choose

$$\epsilon^* = \frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T H \mathbf{g}}$$

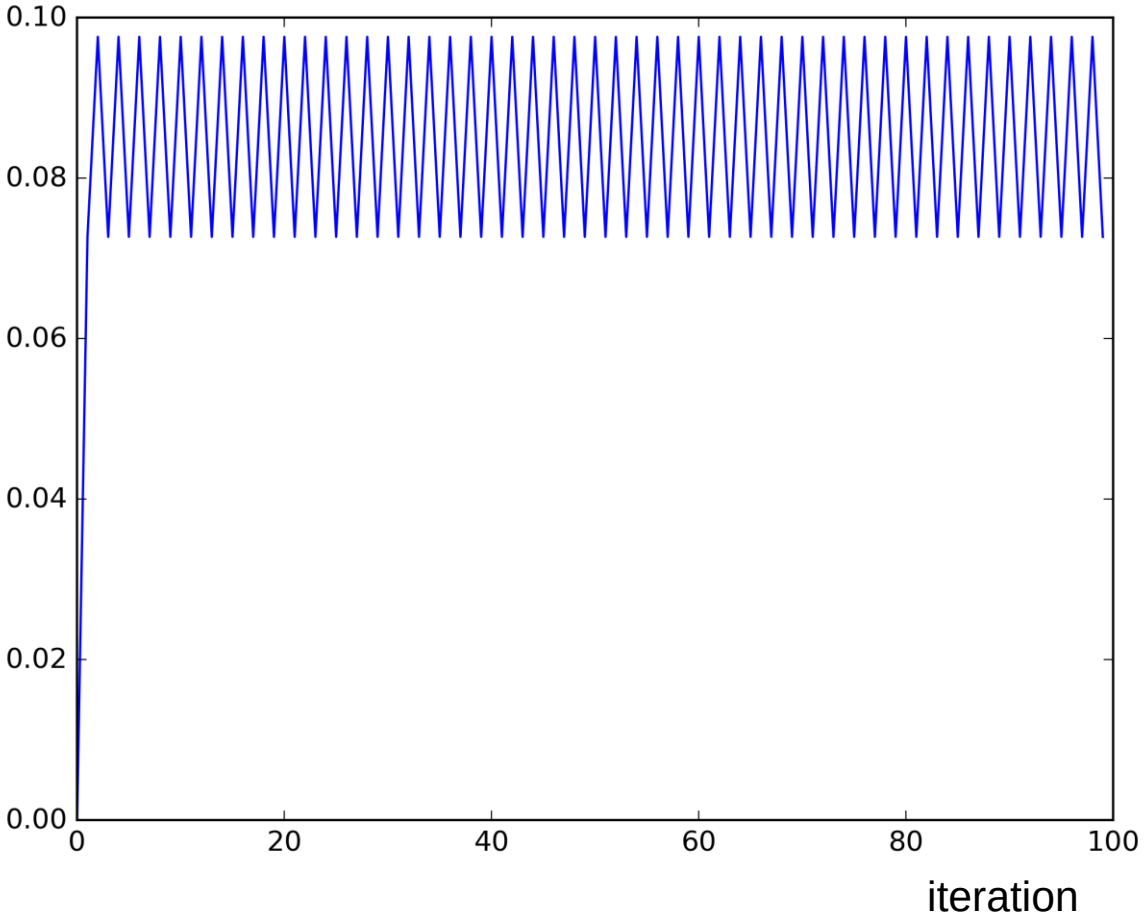
Optimal Learning Rate

$$\epsilon^* = \frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T H \mathbf{g}}$$



Optimal Learning Rate

$$\epsilon^* = \frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T H \mathbf{g}}$$



Second-order Optimization

- Taylor series approximation of $f : \mathbb{R}^n \to \mathbb{R}$

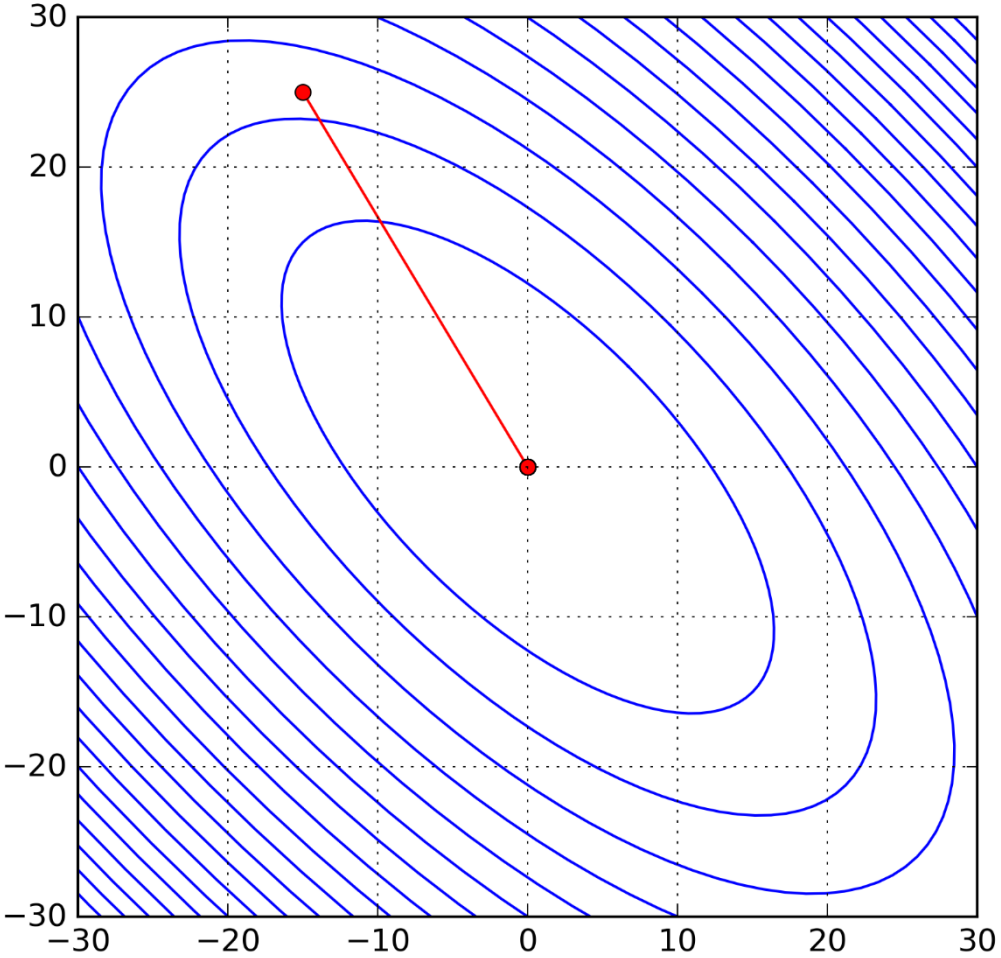
$$f(\mathbf{x}) \sim f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T H (\mathbf{x} - \mathbf{x}_0)$$
$$= \frac{1}{2}(\mathbf{x} - \mathbf{x}_0 + H^{-1}\nabla f(\mathbf{x}_0))^T H (\mathbf{x} - \mathbf{x}_0 + H^{-1}\nabla f(\mathbf{x}_0)) + c$$

- Optimal update rule based on the approximation (Newton's method - second-order optimization algorithm)

$$\mathbf{x}^* = \mathbf{x}_0 - H^{-1}\nabla f(\mathbf{x}_0)$$

- Gradient descent: first-order optimization algorithm

Newton's Method



Ill-conditioned Hessian

- Condition number of $A \in \mathbb{R}^{m \times n}$

$$\frac{\sigma_{\max}}{\sigma_{\min}}$$

- For Hessian, equal to

$$\frac{|\lambda|_{\max}}{|\lambda|_{\min}}$$

- Ill-conditioned matrix has a high condition number. Problematic when inverting a matrix, e.g., when you solve $A\mathbf{x} = \mathbf{b}$.
- Can also be problematic for gradient descent if the Hessian is ill-conditioned.

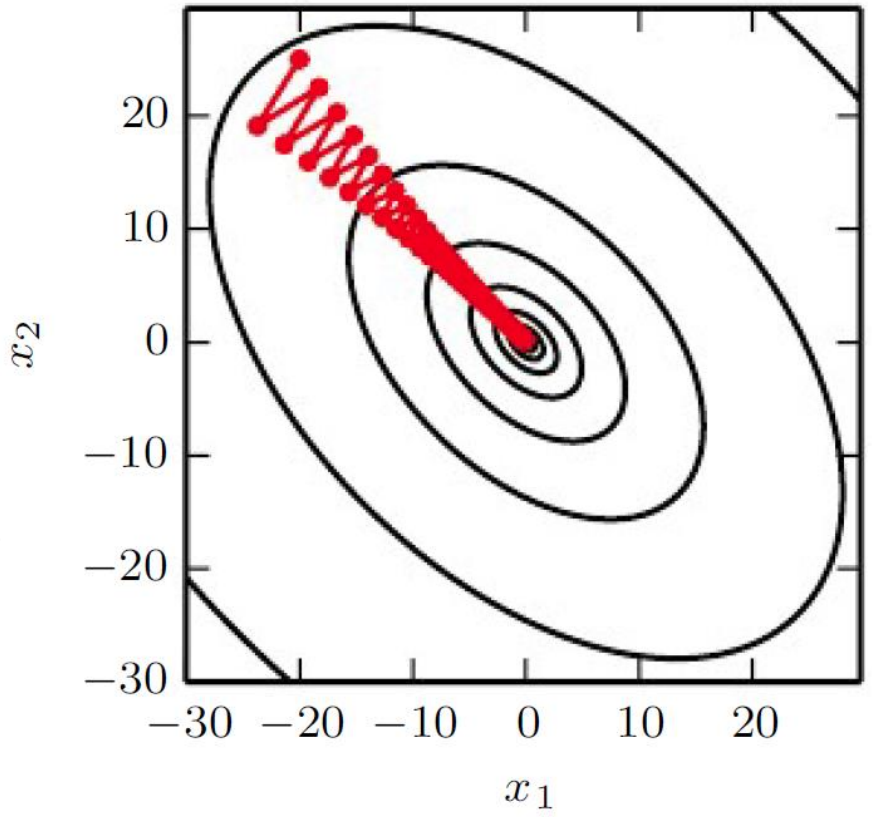
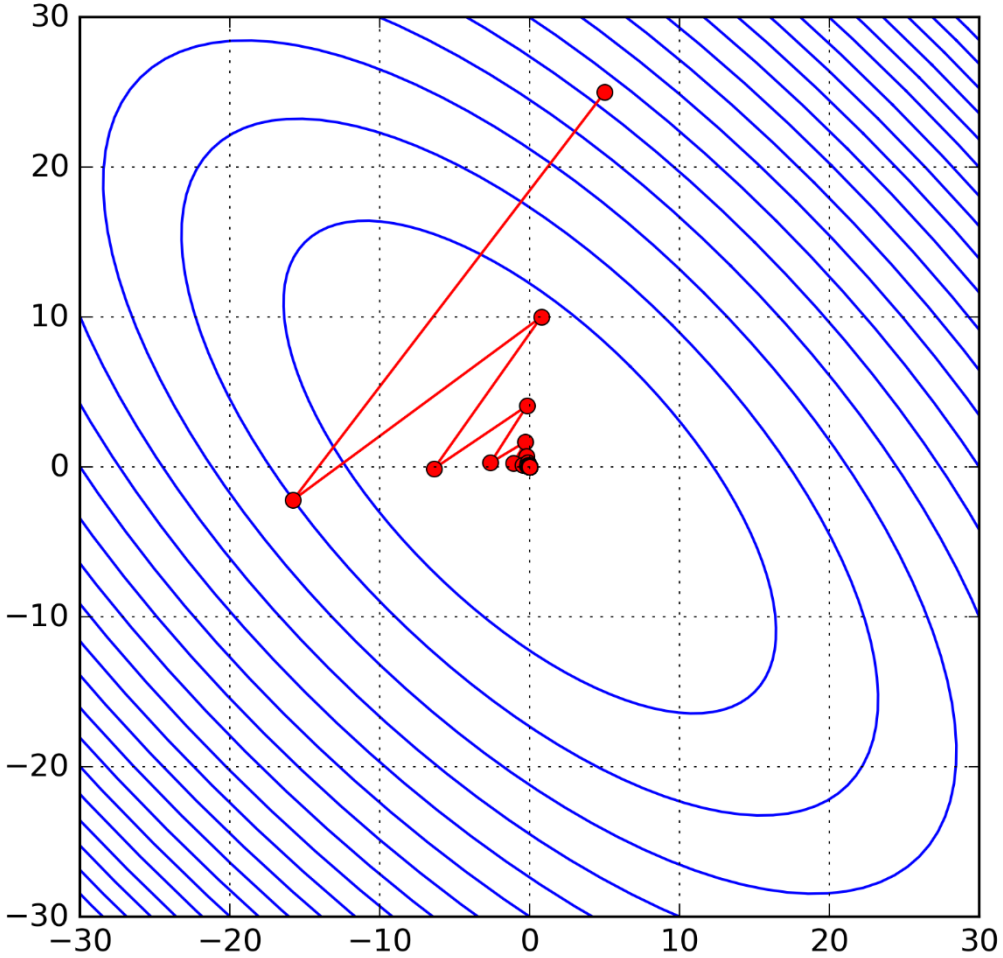


Fig. 4.6

Ill-conditioned Hessian

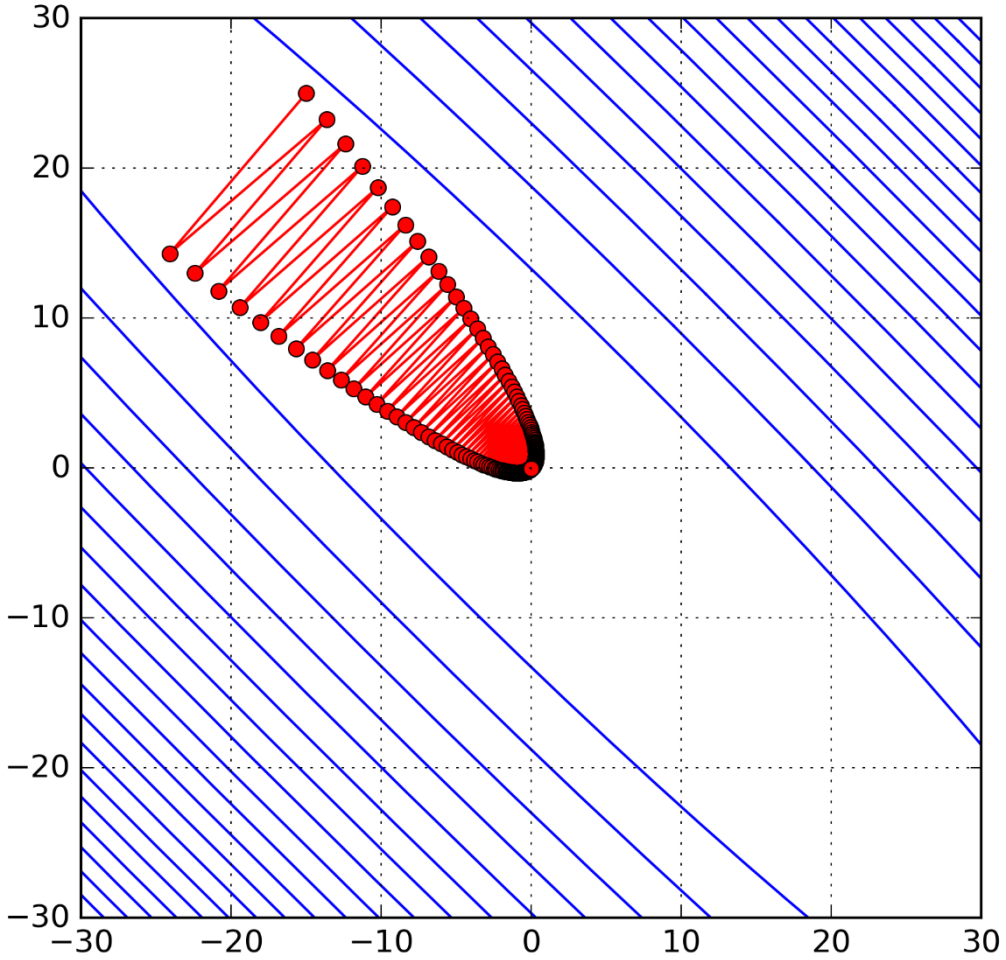
$\epsilon = 0.08$



Ill-conditioned Hessian

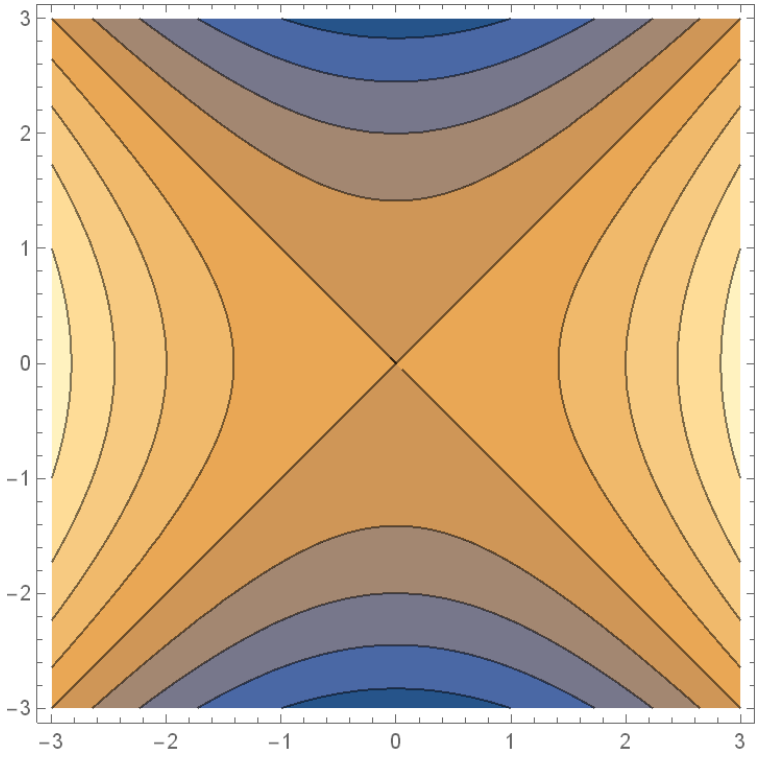
$$f(x, y) = 50(x + y)^2 + (x - y)^2$$

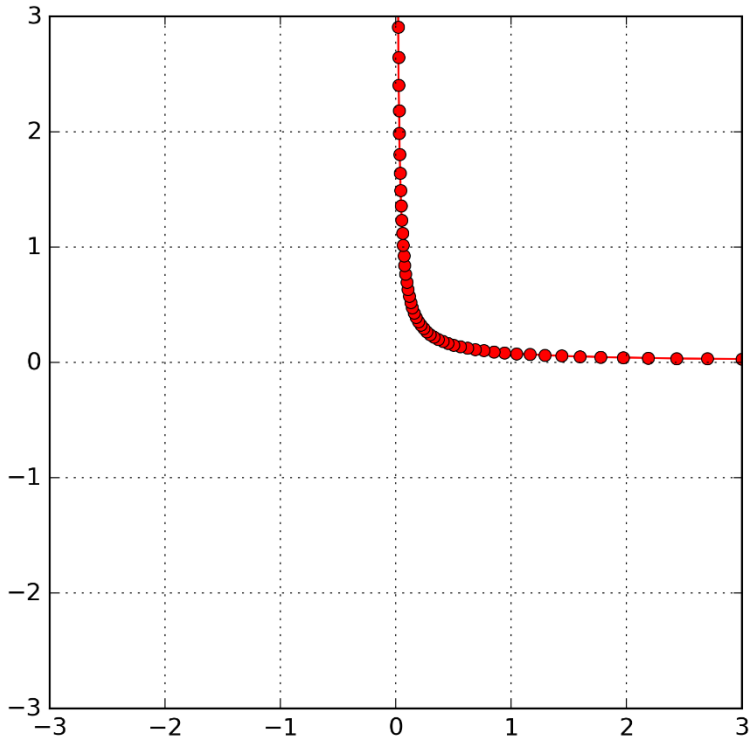
$$\epsilon = 0.0099$$

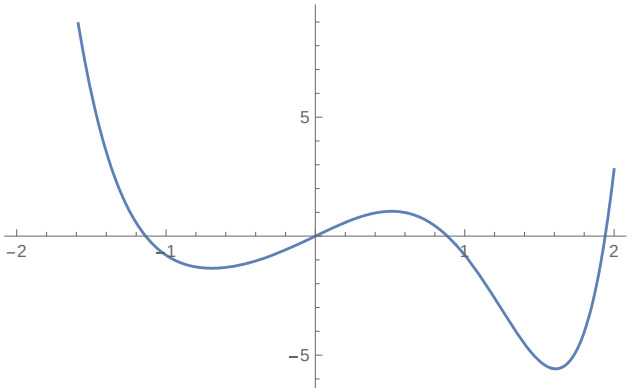


GD and Saddle Point

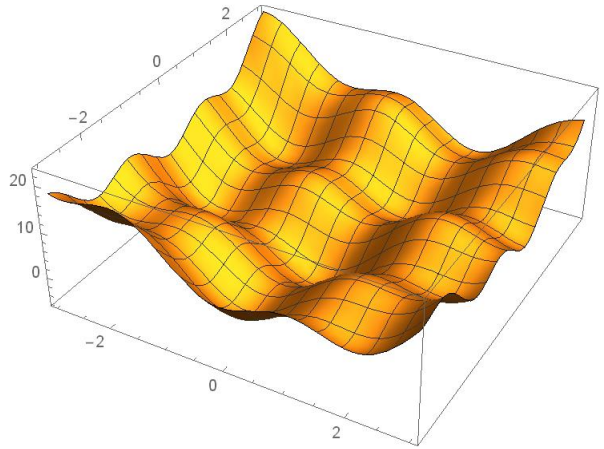
- Gradient descent can be very slow near a saddle point







- Low-dimensional optimization
  - Main problems: local minima



- High-dimensional optimization
  - Main problems: saddle points
  - Conceptually speaking, there can be many more saddle points than local minima since all eigenvalues of H are positive at a local minimum while they have mixed signs at a saddle point.

Constrained Optimization

- Constrained optimization

$$\min_{\mathbf{x}:\ \mathbf{x}\in\mathbb{S}} f(\mathbf{x}) \tag{2}$$

- $\mathbb{S}$: feasible set (or constraint set), $\mathbf{x}$ is called a feasible point if $\mathbf{x} \in \mathbb{S}$.
- We will assume $\mathbb{S} \neq \varnothing$ and

$$\mathbb{S} = \{\mathbf{x} | g_i(\mathbf{x}) = 0, 1 \leq i \leq m, h_j(\mathbf{x}) \leq 0, 1 \leq j \leq r\}$$

- We will also assume f, g, h are continuously differentiable (differentiable and their derivatives are continuous).
- Generalized Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_i \lambda_i g_i(\mathbf{x}) + \sum_j \mu_j h_j(\mathbf{x})$$

- Observe that

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda}} \max_{\boldsymbol{\mu}:\ \boldsymbol{\mu}\geq 0} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x}\in\mathbb{S}} f(\mathbf{x})$$

Karush-Kuhn-Tucker Condition

- KKT conditions: necessary conditions for optimality
- If $\mathbf{x}^*$ is a local minimum of (2) and is regular, then there exist unique $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ such that

  1. $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0$
  2. $\mathbf{x}^* \in \mathbb{S}$, $\boldsymbol{\mu}^* \geq 0$
  3. $\mu_i^* h_i(\mathbf{x}^*) = 0$ for all $1 \leq i \leq r$ (complementary slackness)

- A feasible vector $\mathbf{x}$ is regular if $\nabla g_i(\mathbf{x})$, $\forall i$, and $\nabla h_j(\mathbf{x})$, $\forall j$ such that $h_j(\mathbf{x}) = 0$, are linearly independent.
- If $\mathbf{x}^*$ is not regular, Lagrange multipliers may not exist.
- If, in addition, f, g, h are twice continuously differentiable, then

$$\mathbf{u}^T \nabla^2_{\mathbf{x},\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{u} \geq 0$$

  for all $\mathbf{u} \in \mathbb{R}^n$ such that $\nabla g_i(\mathbf{x}^*)^T \mathbf{u} = 0$, $1 \leq i \leq m$, and $\nabla h_j(\mathbf{x}^*)^T \mathbf{u} = 0$, $1 \leq j \leq r$ for which $h_j(\mathbf{x}^*) = 0$.

Existence of Lagrange Multiplier

- What happens if $\mathbf{x}^*$ is not regular?
- Assume $f(x) = x$, $g(x) = x^2$, $x \in \mathbb{R}$, then $x^* = 0$, but we have

$$\nabla_x L(x^*, \lambda^*) = 1 + 2\lambda^* x^* = 1$$

  and there is no λ^* satisfying $\nabla_x L(x^*, \lambda^*) = 0$.

- Assume $f(x_1, x_2) = x_1 + x_2$, $g_1(x_1, x_2) = x_2 - 2x_1 + 1$, $g_2(x_1, x_2) = x_1^2 - x_2$, then $x_1^* = x_2^* = 1$. But we have

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) &= (1 - 2\lambda_1^* + 2\lambda_2^* x_1^*, 1 + \lambda_1^* - \lambda_2^*)^T \\ &= (1 - 2\lambda_1^* + 2\lambda_2^*, 1 + \lambda_1^* - \lambda_2^*)^T\end{aligned}$$

  and there is no $\boldsymbol{\lambda}^*$ satisfying $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*) = (0, 0)^T$.

Visualization: Lagrange Multiplier

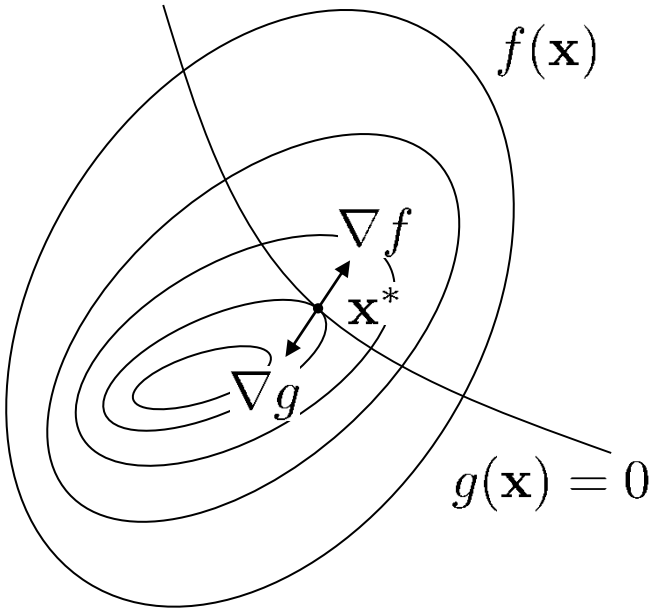
- Constrained optimization with an equality constraint

$$\min_{\mathbf{x}:\ g(\mathbf{x})=0} f(\mathbf{x})$$

- Necessary condition for optimality

$$\nabla f(\mathbf{x}^*) + \lambda \nabla g(\mathbf{x}^*) = 0$$

Same or opposite direction



Visualization: KKT Conditions

- Constrained optimization with an inequality constraint

$$\min_{\mathbf{x}:\ g(\mathbf{x})\leq 0} f(\mathbf{x})$$

- If the constraint is active $(g(\mathbf{x}^*) = 0)$

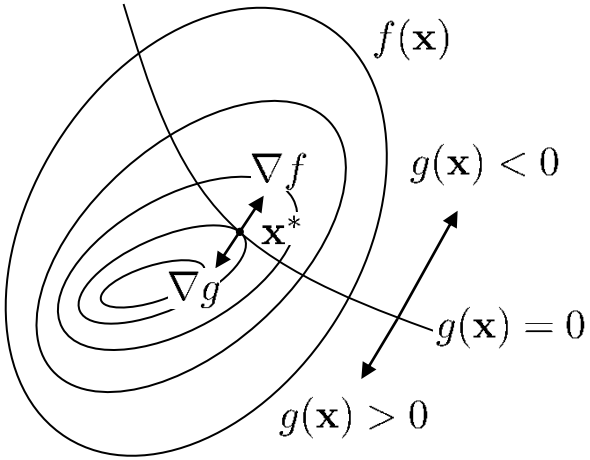
$$\nabla f(\mathbf{x}^*) + \lambda \nabla g(\mathbf{x}^*) = 0$$

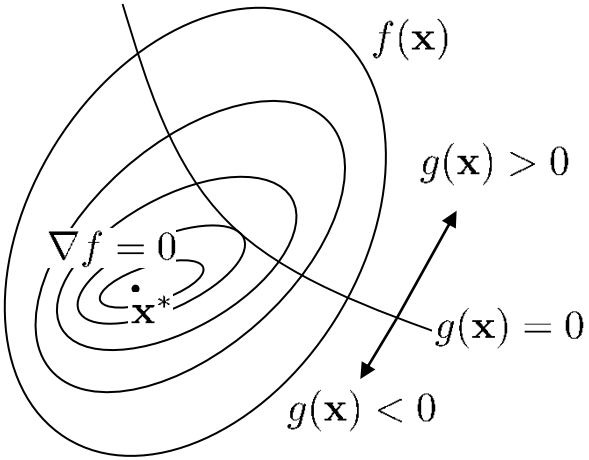
- If inactive $(g(\mathbf{x}^*) < 0)$

$$\nabla f(\mathbf{x}^*) = 0$$

- Combining the two, we get the KKT conditions

$$\nabla f(\mathbf{x}^*) + \lambda \nabla g(\mathbf{x}^*) = 0$$
$$\lambda \geq 0 \text{ if } g(\mathbf{x}^*) = 0$$
$$\lambda = 0 \text{ if } g(\mathbf{x}^*) < 0$$





Barrier Method

- Barrier method for inequality constraints

$$\min_{\mathbf{x}:\ \mathbf{x}\in\mathbb{S}} f(\mathbf{x})$$

  where $\mathbb{S} = \{\mathbf{x} | h_j(\mathbf{x}) \leq 0, 1 \leq j \leq r\}$

- Add a barrier function to the cost, e.g.,

$$\min_{\mathbf{x}} f(\mathbf{x}) + \epsilon B(\mathbf{x})$$

  where $B(\mathbf{x}) = -\sum_{j=1}^{r} \ln[-h_j(\mathbf{x})]$ or $B(\mathbf{x}) = -\sum_{j=1}^{r} \frac{1}{h_j(\mathbf{x})}$ and $\epsilon > 0$

  - As $\epsilon \to 0$, a solution to the above unconstrained problem converges to a global minimum of the original problem
  - Iteratively reduce ϵ, say by a factor of 2, each time using the solution from the previous iteration as the initial point

Penalty Method

- Penalty method for inequality constraints

$$\min_{\mathbf{x}:\ \mathbf{x}\in\mathbb{S}} f(\mathbf{x})$$

  where $\mathbb{S} = \{\mathbf{x} | h_j(\mathbf{x}) \leq 0, 1 \leq j \leq r\}$

- Add a penalty function to the cost, e.g.,

$$\min_{\mathbf{x}} f(\mathbf{x}) + cP(\mathbf{x})$$

  where $P(\mathbf{x}) = \sum_{j=1}^{r} \max\{0, h_j(\mathbf{x})\}^2$ and $c > 0$

  - As $c \to \infty$, a solution to the above unconstrained problem converges to a global minimum of the original problem
  - Iteratively increase c, say by a factor of 2, each time using the solution from the previous iteration as the initial point

- $P(\mathbf{x}) = \sum_{i=1}^{m} g_i^2(\mathbf{x})$ can be used for equality constraints $g_i(\mathbf{x}) = 0$, $i = 1, \ldots, m$

Installing Python

- Lightweight version (but, you have to install each package manually)
  - Download python 2.7.13 from https://www.python.org/ and install it.
  - Open a command window (Windows) or terminal (Mac OS X or linux).
  - You may need to perform the following for upgrading pip:
    - `python –m pip install –-upgrade pip`
  - Install numpy and matplotlib as follows:
    - `pip install numpy`
    - `pip install matplotlib`
- Anaconda (easier to use, but requires 200~300MB to install)
  - Download Adaconda for Python 2.7 from https://www.continuum.io/downloads and install it.
  - It already includes many packages such as numpy and matplotlib.
- Then, run the following
  - `python gradient_descent.py`
  - gradient_descent.py can be downloaded from KLMS.
- Tutorials and manuals for numpy and matplotlib
  - Numpy: http://www.numpy.org/
  - Matplotlib: http://matplotlib.org

Reading Assignment

- Chapter 5 of DL book