Large Scale Data Analysis Using Deep Learning

Numerical Computation

U Kang
Seoul National University

In This Lecture

- Condition number and its impact on optimization
- Gradient descent

Overflow and Underflow

- $softmax(\boldsymbol{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$

- The exponentiation can underflow when the argument is very negative, or overflow when it is very positive

- A solution: softmax($\mathbf{z}$) where $\mathbf{z} = \boldsymbol{x} - \boldsymbol{max_i x_i}$
  - Why?

Condition Number

- Conditioning: how rapidly a function changes with respect to small changes in its inputs
- Function that changes rapidly when their inputs are perturbed slightly can be problematic for scientific computation because rounding errors in the inputs can result in large changes in the output
- Consider $f(x) = A^{-1}x$. When A has an eigendecomposition, its condition number is $\max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|$

Condition Number

- Condition number for the equation Ax=b: the maximum ratio of the relative error in x divided by the relative error in b
- Let e be the error in b, then the condition number is

$$\frac{\frac{||A^{-1}e||}{||A^{-1}b||}}{\frac{||e||}{||b||}} = \frac{||A^{-1}e||}{||e||}\frac{||b||}{||A^{-1}b||}$$ whose maximum value is

$$max_{e,b\neq 0}\frac{||A^{-1}e||}{||e||}\frac{||b||}{||A^{-1}b||}$$

$$= max_{e\neq 0}\frac{||A^{-1}e||}{||e||}max_{b\neq 0}\frac{||b||}{||A^{-1}b||} = |\frac{\lambda_{max}}{\lambda_{min}}|$$

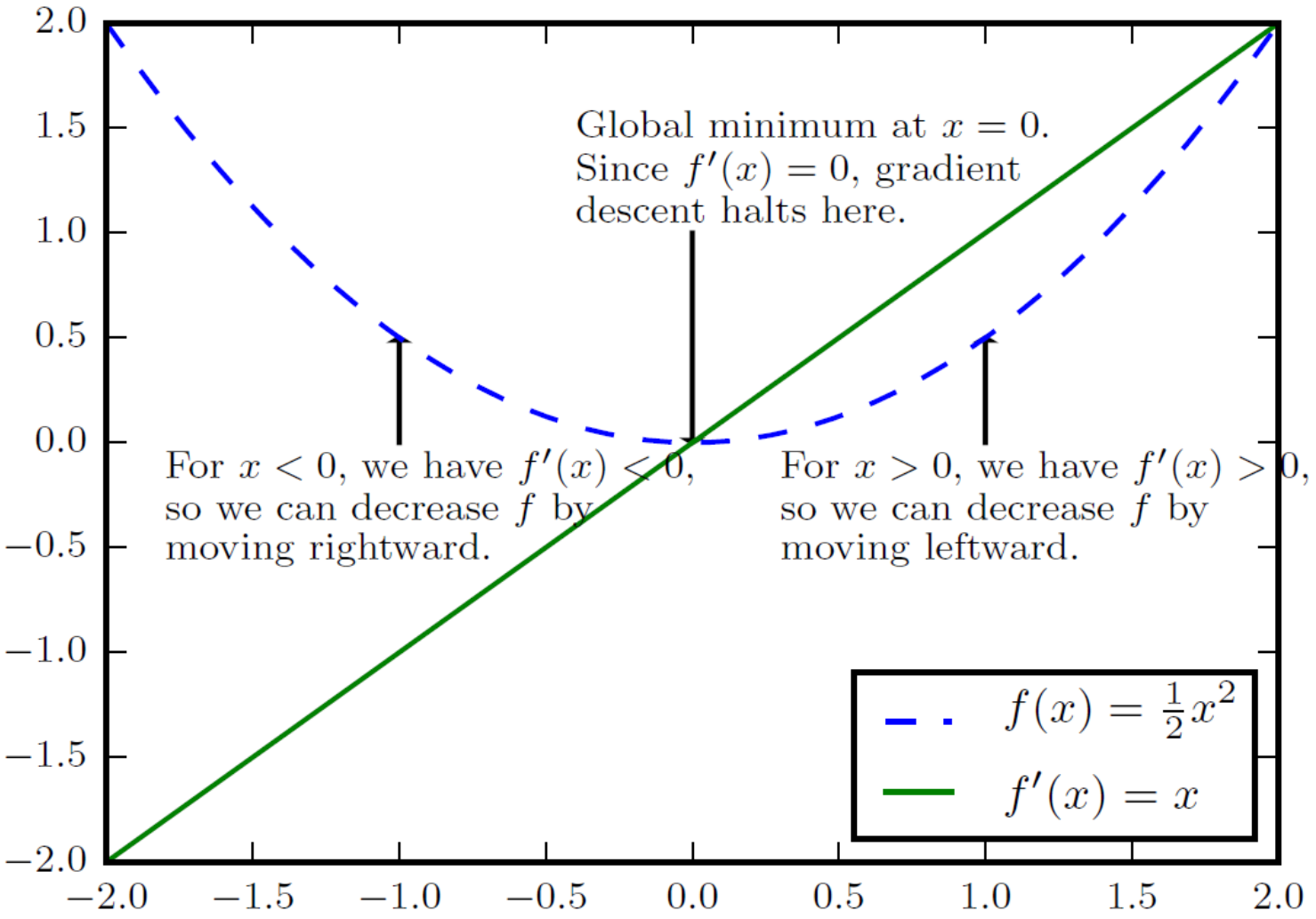
Gradient Based Optimization

- Goal: minimize a function y = f(x)

- Derivative f'(x) (= $\frac{dy}{dx}$)of f(x) gives the slope of f(x) at point x
  - $f(x+\epsilon) \approx f(x) + \epsilon f'(x)$
  - Tells us how to change x in order to make a small improvement in y
  - E.g., $f(x - \epsilon\, sign(f'(x)))$ is less than f(x) for small enough ϵ

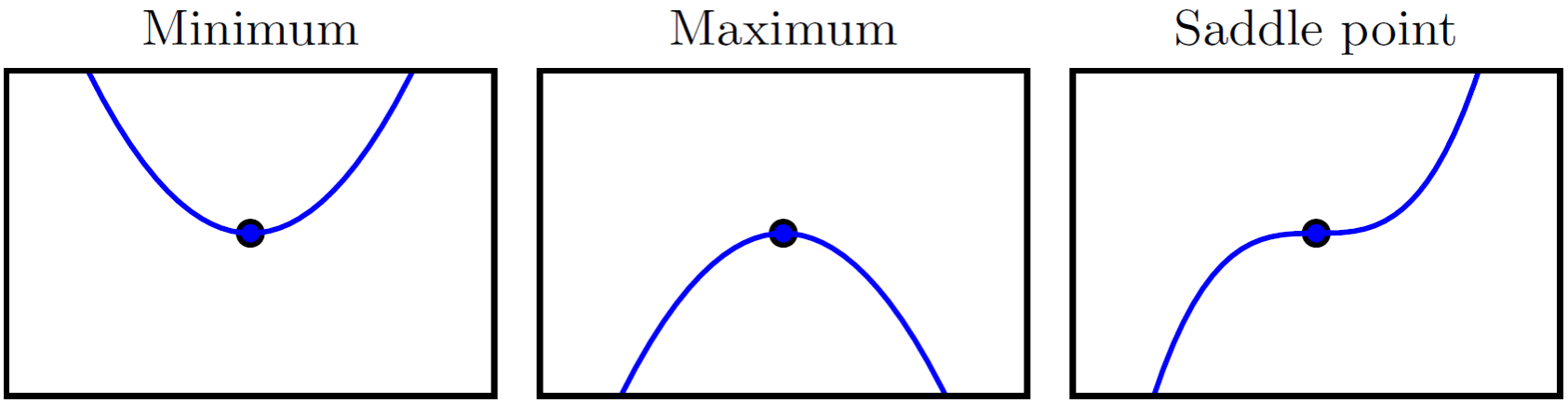
- Gradient descent: reduce f(x) by moving x in small steps with opposite sign of the derivative

Gradient Descent



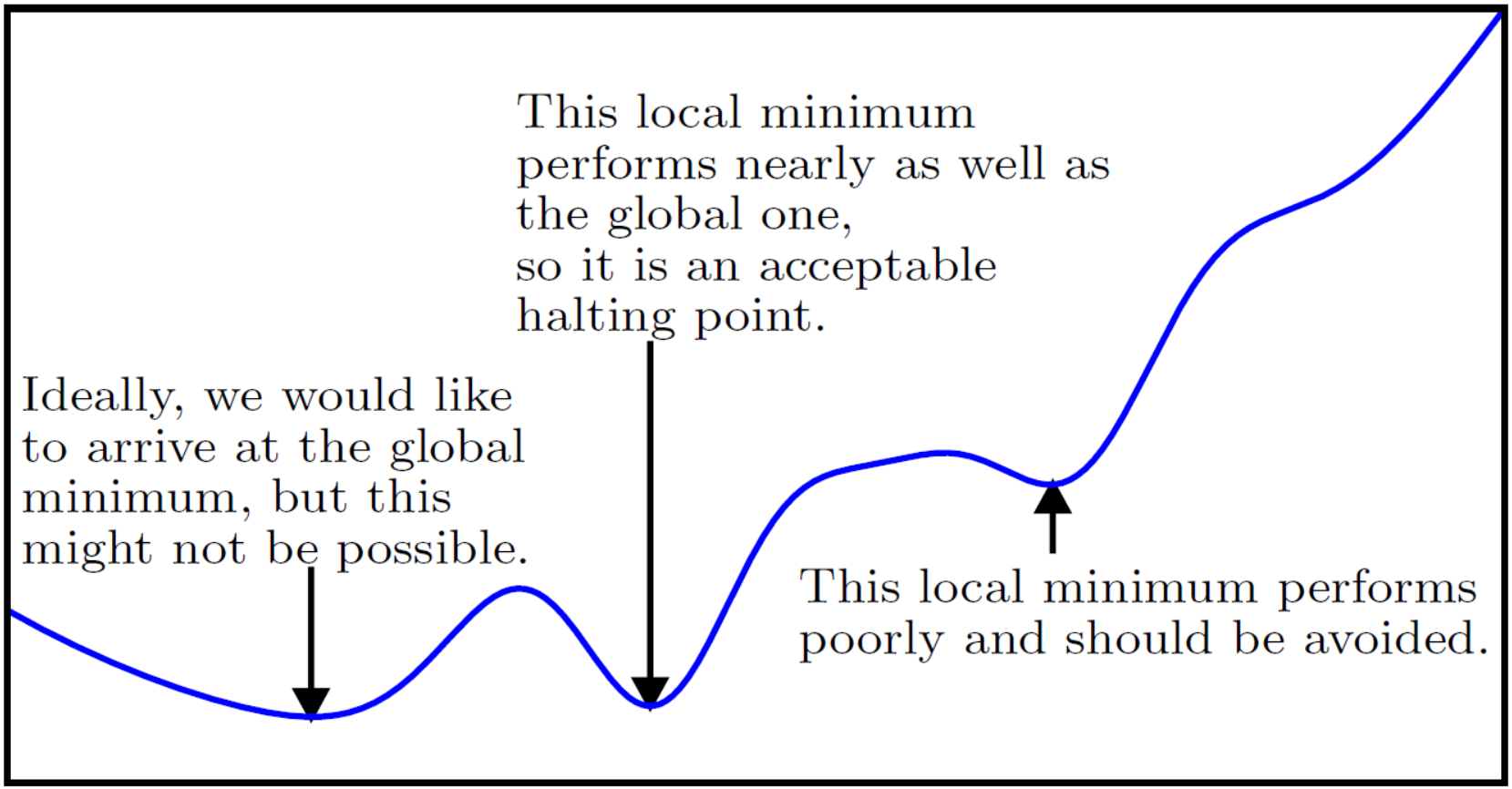
Critical Points

- Critical points or stationary points: f’(x) = 0



Approximate Optimization

- Global minimum vs local minimum



Gradient Descent for Multiple Input

- Functions with multiple inputs: f : $R^n \rightarrow R$
- Partial derivative $\frac{\partial}{\partial x_i} f(\boldsymbol{x})$ measures how f changes as only the variable x_i increases at point $\boldsymbol{x}$.
- Gradient $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ of f is the vector containing all the partial derivatives
- Critical points: every element of the gradient is equal to 0

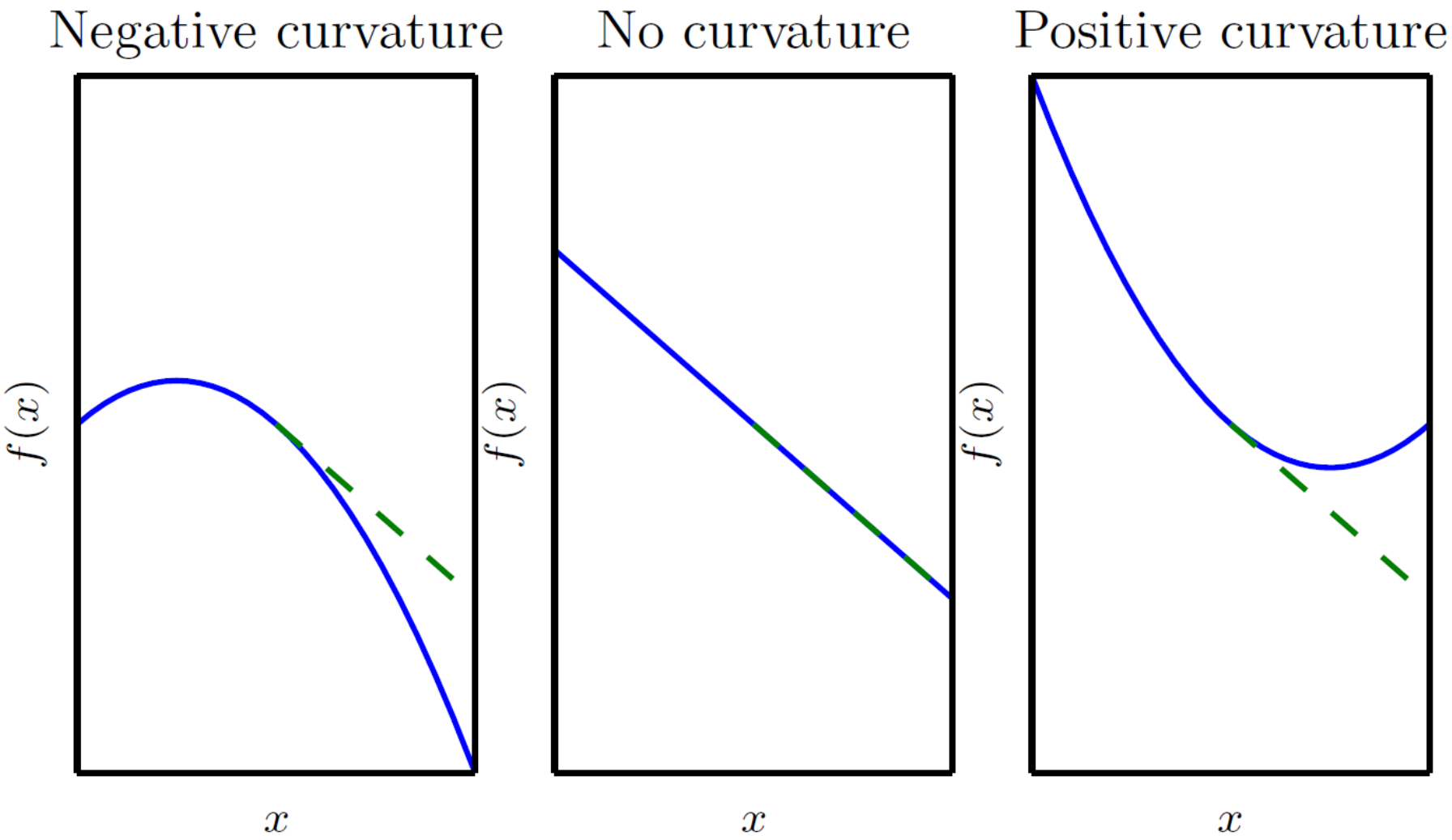
Gradient Descent for Multiple Input

- Directional derivative in direction u (a unit vector): the slope of the function f in direction u
  - Derivative of the function $f(\boldsymbol{x} + \alpha \boldsymbol{u})$ with respect to α, evaluated at $\alpha = 0$: $\frac{\partial}{\partial \alpha} f(\boldsymbol{x} + \alpha \boldsymbol{u}) = u^T \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$
- Find the direction in which f decreases the fastest
  - $min_{u,u^T u=1} u^T \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = min_{u,u^T u=1} ||u||_2 ||\nabla_{\boldsymbol{x}} f(\boldsymbol{x})||_2 cos\theta$
  - This is minimized when u points in the opposite direction as the gradient: $\boldsymbol{x}' = \boldsymbol{x} - \epsilon \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$
  - This method is called the *steepest descent* or *gradient descent*
  - ϵ (called step size) can be set to a small constant, to a value that minimizes f(x), or to a value that results in the smallest f(x) among several values (line search)

Beyond the Gradient: Jacobian and Hessian

- Jacobian : a matrix containing partial derivatives of a function whose input and output are both vectors
  - For a function f: $R^m \rightarrow R^n$, Jacobian $J \in R^{n \times m}$ is defined such that $J_{i,j} = \frac{\partial}{\partial x_j} f(\boldsymbol{x})_i$
- Second derivative: derivative of a derivative
  - For a function f: $R^m \rightarrow R$, the derivative with respect to x_i of the derivative of f with respect to x_j is denote by $\frac{\partial^2}{\partial x_i \partial x_j} f$
  - In a single dimension, second derivative is denote by f''(x)
  - Tells us whether a gradient step will cause as much of an improvement as we would expect based on the gradient alone

Curvature


Negative curvature
No curvature
Positive curvature
$f(x)$
x
$f(x)$
x
$f(x)$
x

Hessian

- Second derivatives of a function f: $R^n \rightarrow R$ with multiple input dimensions
- Hessian matrix $H(f)(\boldsymbol{x})$ is defined such that $H(f)(\boldsymbol{x})_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} f(\boldsymbol{x})$
- The differential operators are commutative: $\frac{\partial^2}{\partial x_i \partial x_j} f(\boldsymbol{x}) = \frac{\partial^2}{\partial x_j \partial x_i} f(\boldsymbol{x})$. Thus, H is symmetric.
- The second derivative in a specific direction represented by a unit vector d is given by $d^T H d$.
  - (pf) Note that the first derivative is given by $d^T \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$. The second derivative is given by $d^T \nabla_{\boldsymbol{x}} \left(d^T \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \right) = d^T (H^T d) = d^T H d$

Aside: Chain Rule of Calculus

- Let x be a real number, and f and g be functions from R to R. Suppose y = g(x), and z = f(g(x)) = f(y). Then the chain rule states that $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$
- Suppose $x \in R^m, y \in R^n, g: R^m \to R^n, f: R^n \to R$. If $\boldsymbol{y} = g(\boldsymbol{x})$ and $z = f(\boldsymbol{y})$, then $\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i}$. In vector notation: $\nabla_{\boldsymbol{x}} z = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^T \nabla_{\boldsymbol{y}} z$, where $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ is the n x m Jacobian matrix of g
  - E.g., suppose $z = d^T \boldsymbol{y}$, and $\mathbf{y} = \nabla_{\boldsymbol{x}} g(\boldsymbol{x})$. Then $\nabla_{\boldsymbol{x}} z = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^T \nabla_{\boldsymbol{y}} z = H^T d$

Hessian

- If d is an eigenvector of H, the directional second derivative $d^T H d$ is the corresponding eigenvalue. For other direction, the directional second derivative is a weighted average of all of the eigenvalues
- The second derivative tells us how well we can expect a gradient descent step to perform
  - $f(x) \approx f\left(x^{(0)}\right) + \left(x - x^{(0)}\right)^T g + \frac{1}{2}\left(x - x^{(0)}\right)^T H\left(x - x^{(0)}\right)$ where g is the gradient and H is the Hessian at $x^{(0)}$.
  - Using a learning rate of ϵ, then the new point x will be given by $x^{(0)} - \epsilon g$.
  - Then, $f\left(x^{(0)} - \epsilon g\right) \approx f\left(x^{(0)}\right) - \epsilon g^T g + \frac{1}{2}\epsilon^2 g^T H g$

The original value

improvement due to slope of f

correction to account for the curvature of f

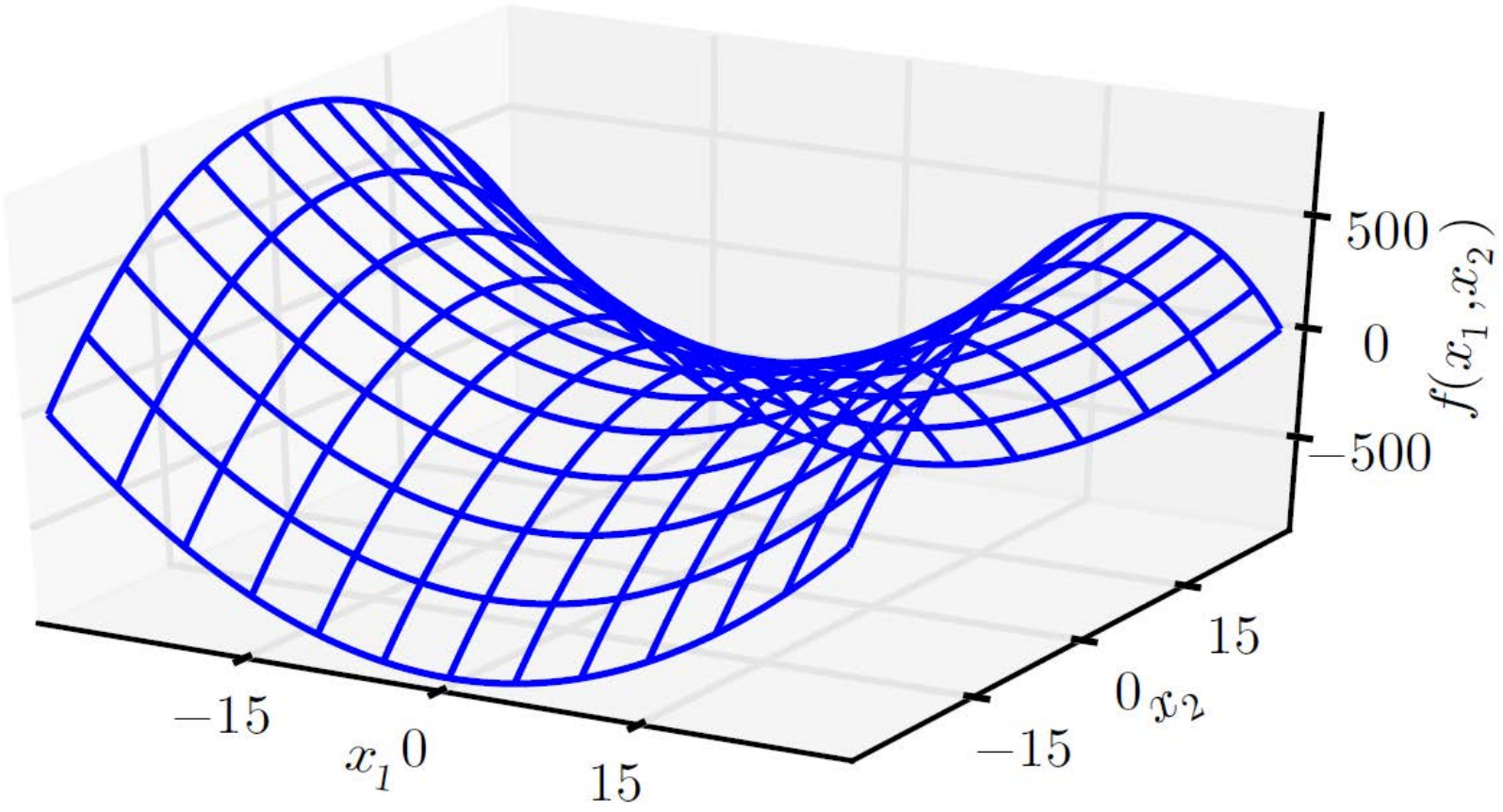
Hessian

- The term $\frac{1}{2}\epsilon^2 g^T H g$
  - If $g^T H g$ is too large, the gradient descent step can actually move uphill
  - If $g^T H g$ is 0 or negative, increasing ϵ would decrease f
  - When $g^T H g$ is positive, the optimal ϵ is given by $\frac{g^T g}{g^T H g}$
  - If f is approximated well by a quadratic function, the eigenvalues of the Hessian determine the scale of the learning rate

Hessian

- Second derivative determines whether a critical point is a local maximum, a local minimum, or a saddle point
  - f'(x) = 0 and f''(x) > 0: x is a local minimum
  - f'(x) = 0 and f''(x) < 0: x is a local maximum
  - f'(x) = 0 and f''(x) = 0: x may be a saddle point or a part of a flat region
- In multiple dimension: need to examine all of the second derivatives
  - At a critical point where $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \mathbf{0}$, we can examine eigenvalues of the Hessian to determine whether the critical point is a local maximum, local minimum, or a saddle point
  - If H is positive definite, then x is a local minimum
  - If H is negative definite, then x is a local maximum
  - If H has both positive and negative eigenvalues, then x is a saddle point

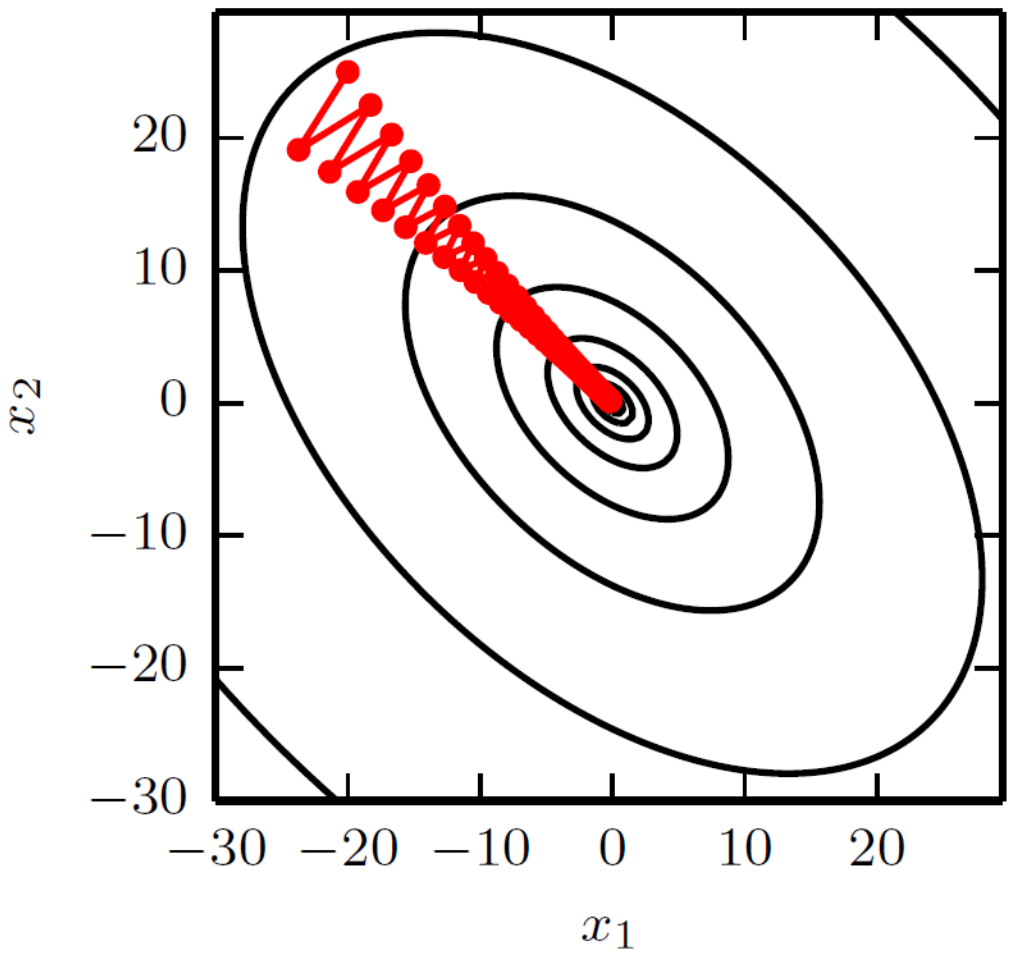
Saddle Points



Gradient Descent and Poor Conditioning

- The condition number of the Hessian measures how much the second derivative vary
- When the Hessian has a poor condition number, gradient descent performs poorly
  - In one direction, the derivative increases rapidly, while in another direction, it increases slowly
  - Gradient descent is unaware of this change in the derivative so it does not know that it needs to explore preferentially in the direction where the derivative remains negative for longer
  - This makes it difficult to choose a good step size: the step size must be small enough to avoid overshooting, but this means the step size is too small to make significant progress in other direction with less curvature

Gradient Descent and Poor Conditioning



Gradient Descent and Poor Conditioning

- The problem of the Hessian with a poor condition number can be solved by using the second-order method
- First-order method: use only the gradient to find the optimum
  - Gradient Descent
- Second-order method: use both the gradient and the Hessian
  - Newton's method

What you need to know

- Gradient descent
  - Useful method for optimization
  - Multiple inputs: Jacobian and Hessian
  - Second derivative gives useful information for optimization
  - For a function with poorly conditioned Hessian, gradient descent performs badly

Questions?