Deep Learning

LM Computer Science, Data Science, Cybersecurity

2nd semester - 6 CFU

Luca Pasa, Nicolò Navarin & Alessandro Sperduti

Learning with gradient (Chp 4)

Numerical concerns for implementations of deep learning algorithms

- Algorithms are often specified in terms of real numbers; real numbers cannot be implemented in a finite computer
- Does the algorithm still work when implemented with a finite number of bits?
- Do small changes in the input to a function cause large changes to an output? (poor conditioning amplifies rounding errors)
- Rounding errors, noise, measurement errors can cause large changes
- more details in Chapter 4

Gradient-based optimization

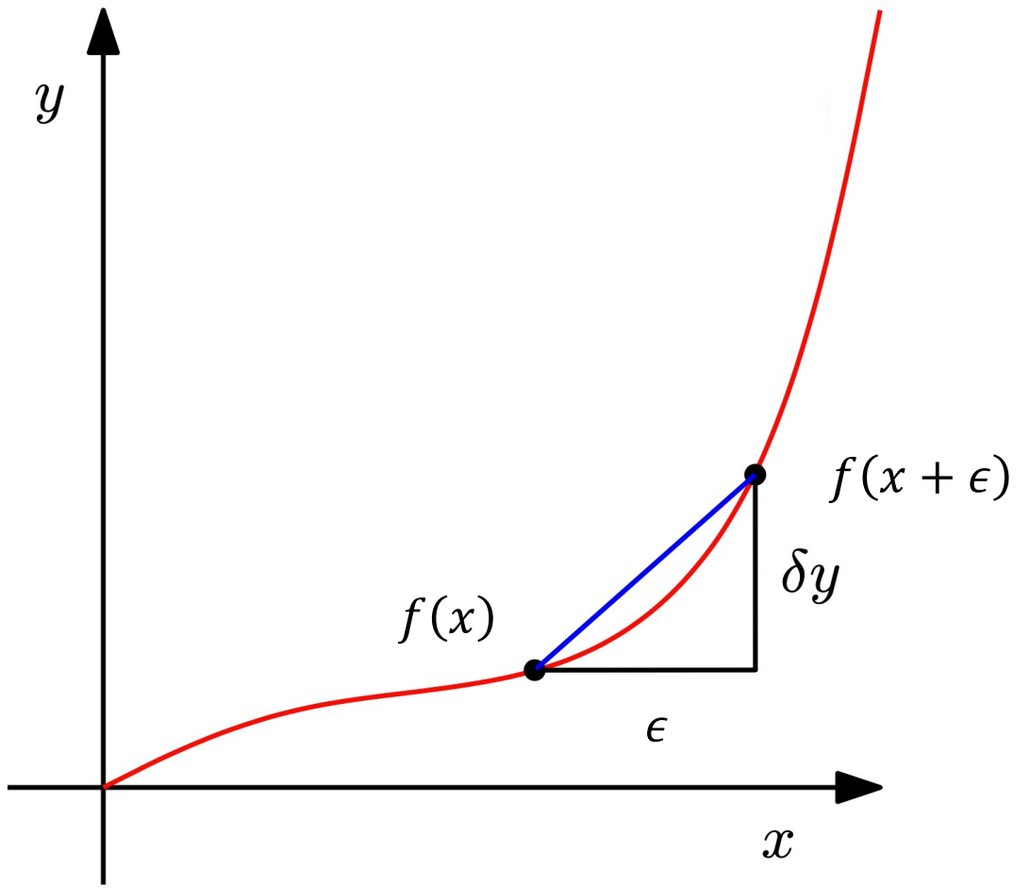
We want to maximize/minimize an **objective function**

- Minimizing **Cost/error/loss** function

Consider a function $y = f(x)$

- The **Difference Quotient** compute the **slope** of the secant line through two points

$$\frac{\delta y}{\epsilon} = \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



Gradient-based optimization

- In the limit for $\epsilon \to 0$, we obtain the tangent of f at x, if f is differentiable
- The tangent is then the derivative of f at x
- The **derivative** $f'(x)$ or $\frac{dy}{dx}$ gives the **slope** of $f(x)$ at point x, or equivalently

$$f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}, \qquad \epsilon > 0$$

$$f(x+\epsilon) \approx f(x) + \epsilon f'(x)$$

- Derivative tells us how to change x to make a small improvement in y
- **Critical/stationary** points: $f'(x) = 0$ No Information!
- Can be **local maxima, minima** or **saddle** points

Univariate calculus

- In this course, we will assume some familiarity with calculus!
- In Moodle you can find a cheat sheet!
  - Also, a link to some resources, useful to refresh your memories!
  - http://www.columbia.edu/itc/sipa/math/ (bottom of the page: calculus resources)
- Let's refresh some basic concepts together..
- Chapter 5 of "Mathematics for Machine Learning"

Derivatives

Definition and Notation

If $y = f(x)$ then the derivative is defined to be $f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$.

If $y = f(x)$ then all of the following are equivalent notations for the derivative.

$$f'(x) = y' = \frac{df}{dx} = \frac{dy}{dx} = \frac{d}{dx}\left(f(x)\right) = Df(x)$$

If $y = f(x)$ all of the following are equivalent notations for derivative evaluated at $x = a$.

$$f'(a) = y'\Big|_{x=a} = \frac{df}{dx}\Big|_{x=a} = \frac{dy}{dx}\Big|_{x=a} = Df(a)$$

Interpretation of the Derivative

If $y = f(x)$ then,

1. $m = f'(a)$ is the slope of the tangent line to $y = f(x)$ at $x = a$ and the equation of the tangent line at $x = a$ is given by $y = f(a) + f'(a)(x-a)$.
2. $f'(a)$ is the instantaneous rate of change of $f(x)$ at $x = a$.
3. If $f(x)$ is the position of an object at time x then $f'(a)$ is the velocity of the object at $x = a$.

Basic Properties and Formulas

If $f(x)$ and $g(x)$ are differentiable functions (the derivative exists), c and n are any real numbers,

1. $(cf)' = cf'(x)$
2. $(f \pm g)' = f'(x) \pm g'(x)$
3. $(fg)' = f'g + fg'$ – **Product Rule**
4. $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$ – **Quotient Rule**
5. $\frac{d}{dx}(c) = 0$
6. $\frac{d}{dx}(x^n) = nx^{n-1}$ – **Power Rule**
7. $\frac{d}{dx}(f(g(x))) = f'(g(x))g'(x)$
   This is the **Chain Rule**

Common Derivatives

$\frac{d}{dx}(x) = 1$

$\frac{d}{dx}(\sin x) = \cos x$

$\frac{d}{dx}(\cos x) = -\sin x$

$\frac{d}{dx}(\tan x) = \sec^2 x$

$\frac{d}{dx}(\sec x) = \sec x \tan x$

$\frac{d}{dx}(\csc x) = -\csc x \cot x$

$\frac{d}{dx}(\cot x) = -\csc^2 x$

$\frac{d}{dx}(\sin^{-1} x) = \frac{1}{\sqrt{1-x^2}}$

$\frac{d}{dx}(\cos^{-1} x) = -\frac{1}{\sqrt{1-x^2}}$

$\frac{d}{dx}(\tan^{-1} x) = \frac{1}{1+x^2}$

$\frac{d}{dx}(a^x) = a^x \ln(a)$

$\frac{d}{dx}(\mathbf{e}^x) = \mathbf{e}^x$

$\frac{d}{dx}(\ln(x)) = \frac{1}{x}, \quad x > 0$

$\frac{d}{dx}(\ln|x|) = \frac{1}{x}, \quad x \neq 0$

$\frac{d}{dx}(\log_a(x)) = \frac{1}{x \ln a}, \quad x > 0$

Chain Rule Variants

The chain rule applied to some specific functions.

1. $\frac{d}{dx}\left(\left[f(x)\right]^n\right) = n\left[f(x)\right]^{n-1} f'(x)$
2. $\frac{d}{dx}\left(\mathbf{e}^{f(x)}\right) = f'(x)\mathbf{e}^{f(x)}$
3. $\frac{d}{dx}\left(\ln\left[f(x)\right]\right) = \frac{f'(x)}{f(x)}$
4. $\frac{d}{dx}\left(\sin\left[f(x)\right]\right) = f'(x)\cos\left[f(x)\right]$
5. $\frac{d}{dx}\left(\cos\left[f(x)\right]\right) = -f'(x)\sin\left[f(x)\right]$
6. $\frac{d}{dx}\left(\tan\left[f(x)\right]\right) = f'(x)\sec^2\left[f(x)\right]$
7. $\frac{d}{dx}\left(\sec\left[f(x)\right]\right) = f'(x)\sec\left[f(x)\right]\tan\left[f(x)\right]$
8. $\frac{d}{dx}\left(\tan^{-1}\left[f(x)\right]\right) = \frac{f'(x)}{1+\left[f(x)\right]^2}$

Higher Order Derivatives

The Second Derivative is denoted as $f''(x) = f^{(2)}(x) = \frac{d^2 f}{dx^2}$ and is defined as $f''(x) = \left(f'(x)\right)'$, *i.e.* the derivative of the first derivative, $f'(x)$.

The n[th] Derivative is denoted as $f^{(n)}(x) = \frac{d^n f}{dx^n}$ and is defined as $f^{(n)}(x) = \left(f^{(n-1)}(x)\right)'$, *i.e.* the derivative of the (n-1)[st] derivative, $f^{(n-1)}(x)$.

Implicit Differentiation

Find y' if $\mathbf{e}^{2x-9y} + x^3y^2 = \sin(y) + 11x$. Remember $y = y(x)$ here, so products/quotients of x and y will use the product/quotient rule and derivatives of y will use the chain rule. The "trick" is to differentiate as normal and every time you differentiate a y you tack on a y' (from the chain rule). After differentiating solve for y'.

$$\begin{aligned}
&\mathbf{e}^{2x-9y}(2-9y') + 3x^2y^2 + 2x^3y\,y' = \cos(y)y' + 11\\
&2\mathbf{e}^{2x-9y} - 9y'\mathbf{e}^{2x-9y} + 3x^2y^2 + 2x^3y\,y' = \cos(y)y' + 11 \quad \Rightarrow \quad y' = \frac{11 - 2\mathbf{e}^{2x-9y} - 3x^2y^2}{2x^3y - 9\mathbf{e}^{2x-9y} - \cos(y)}\\
&\left(2x^3y - 9\mathbf{e}^{2x-9y} - \cos(y)\right)y' = 11 - 2\mathbf{e}^{2x-9y} - 3x^2y^2
\end{aligned}$$

Increasing/Decreasing – Concave Up/Concave Down

Critical Points

$x = c$ is a critical point of $f(x)$ provided either **1.** $f'(c) = 0$ or **2.** $f'(c)$ doesn't exist.

Increasing/Decreasing

1. If $f'(x) > 0$ for all x in an interval I then $f(x)$ is increasing on the interval I.
2. If $f'(x) < 0$ for all x in an interval I then $f(x)$ is decreasing on the interval I.
3. If $f'(x) = 0$ for all x in an interval I then $f(x)$ is constant on the interval I.

Concave Up/Concave Down

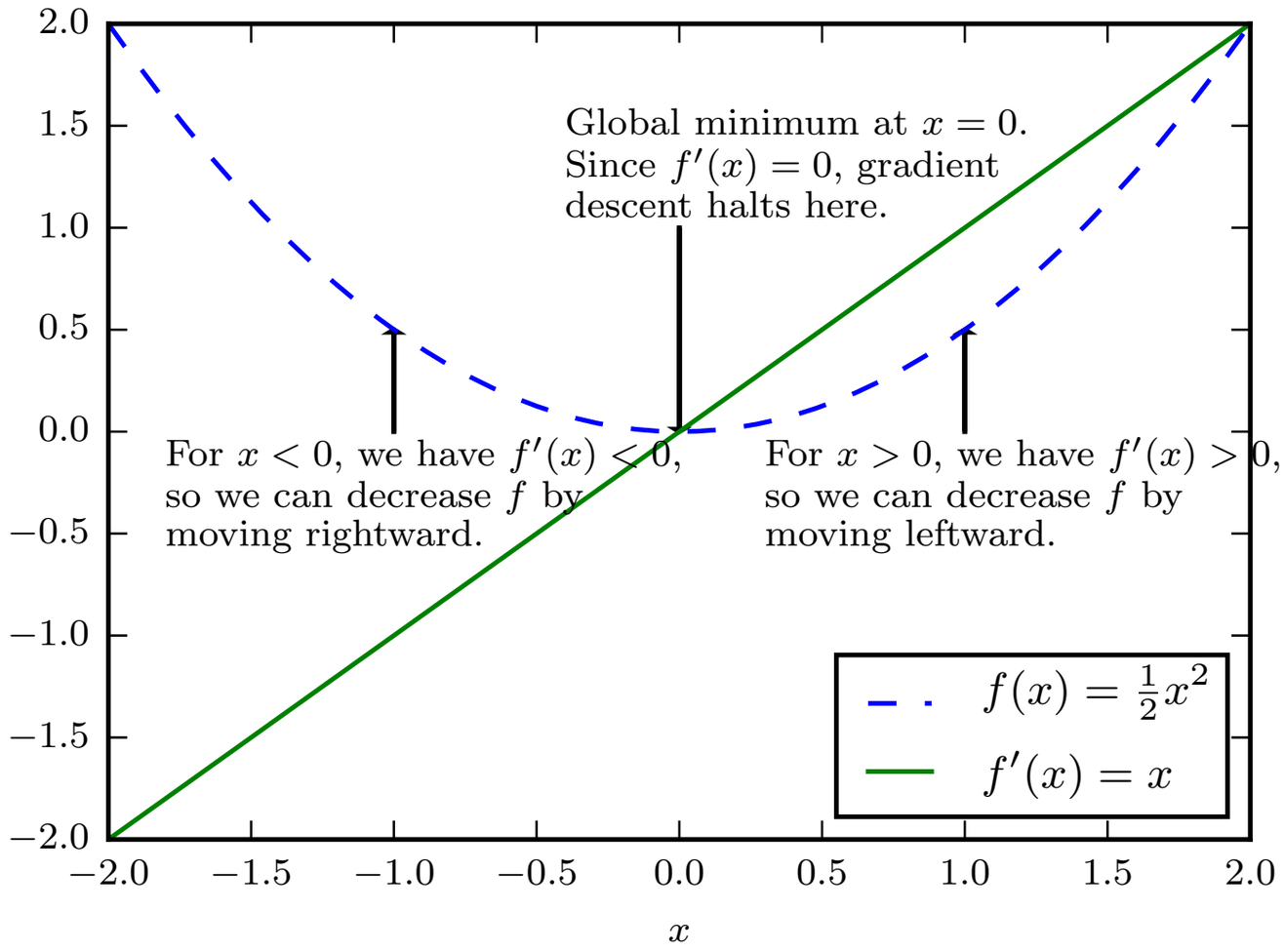
1. If $f''(x) > 0$ for all x in an interval I then $f(x)$ is concave up on the interval I.
2. If $f''(x) < 0$ for all x in an interval I then $f(x)$ is concave down on the interval I.

Inflection Points

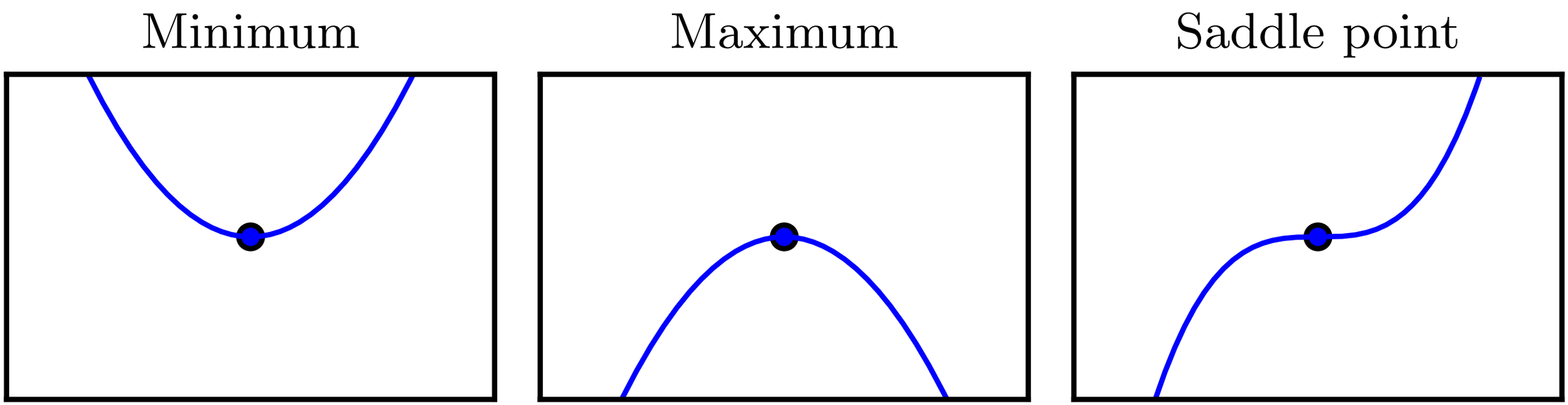
$x = c$ is a inflection point of $f(x)$ if the concavity changes at $x = c$.

Optimization in 1 variable

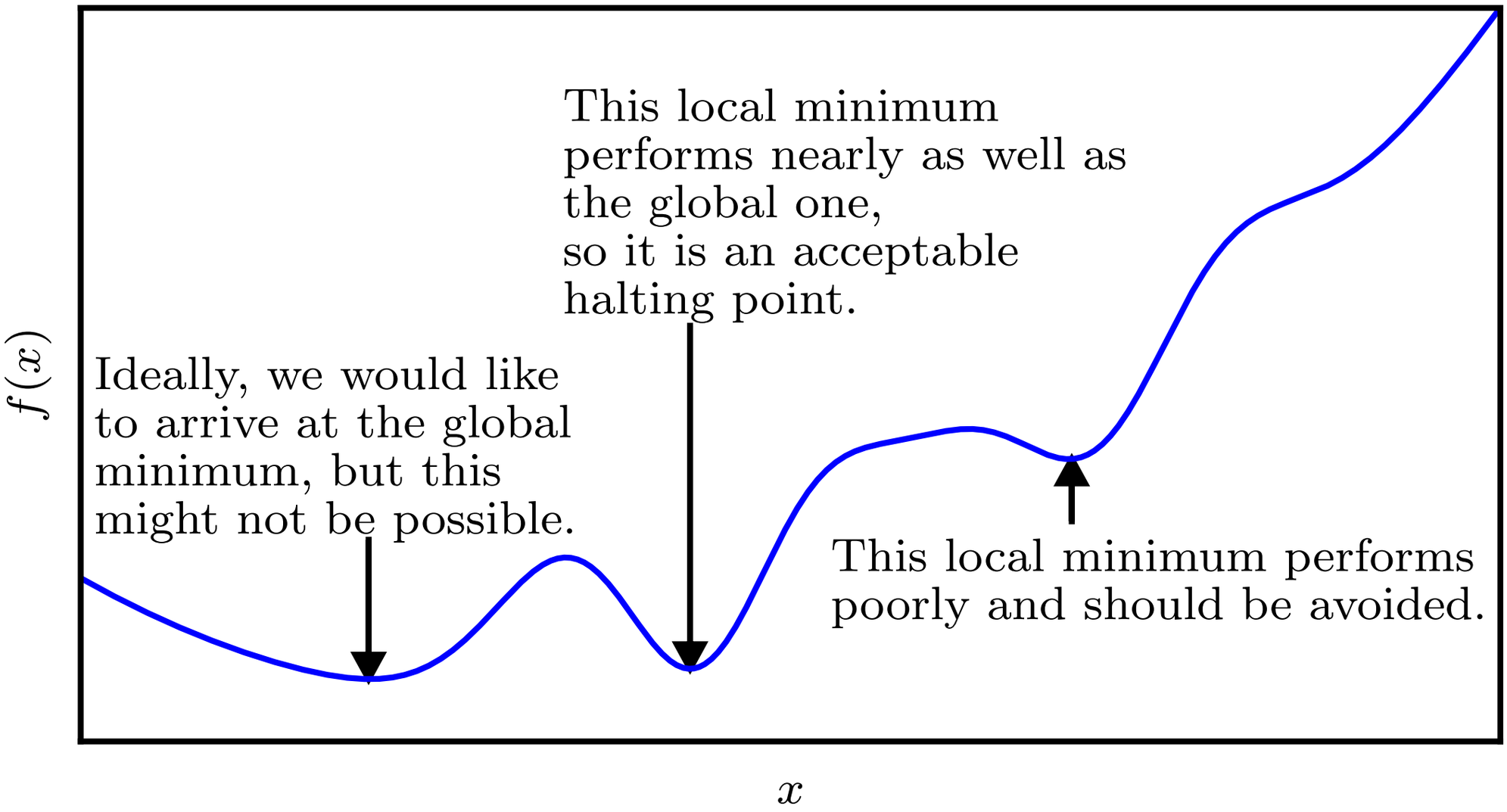
- To minimize a function in 1 variable, we have to move in the direction **opposite** to the derivative



Critical points



Global vs local minima



- Global minimum vs local minima

Simple example: finding a minimum

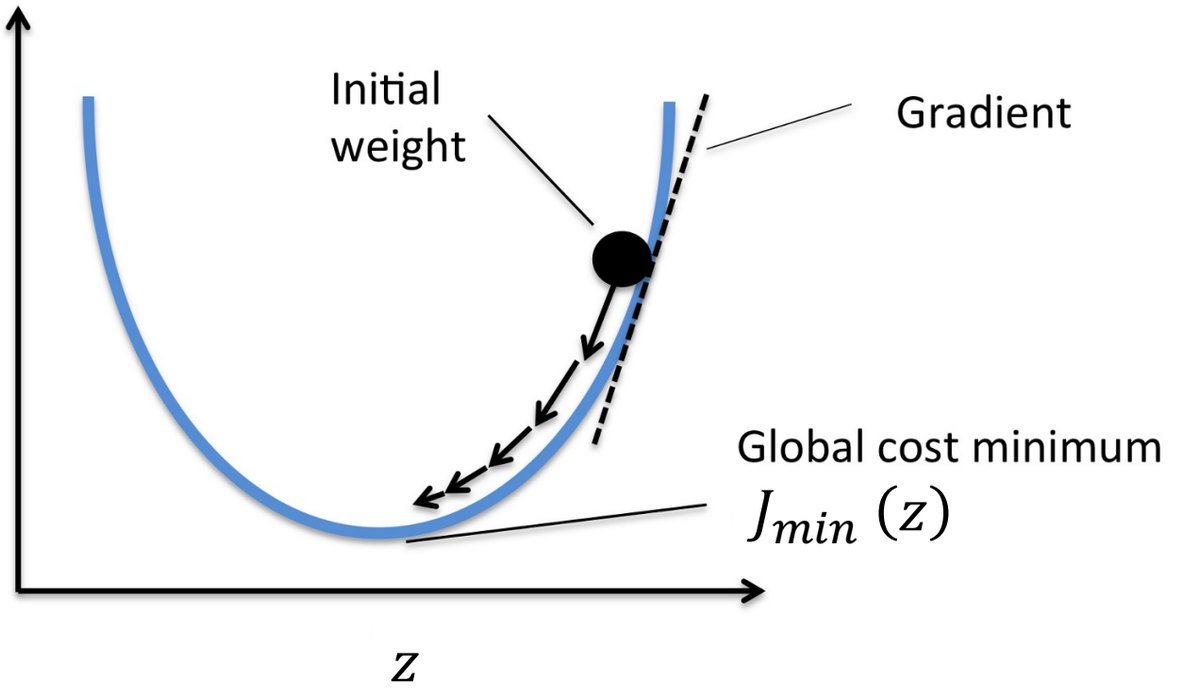
- Let's try to find the minimum with gradient descent of $J(z) = z^2$
- Compute the derivative (gradient) $J'(z) = 2z$ (check the cheat sheet)
- It is the slope of the tangent, i.e. points in the direction of the greatest rate of increase of the function

Idea of gradient descent:

1. Start with a random value for z
2. Update z doing a little step in the opposite direction w.r.t. the derivative (α parameter)

$$z = z - \alpha J'(z)$$

1. Repeat 2 until convergence (i.e. gradient very small)



AL: start from z=2, with α=0.25 find the minimum of the function

Simple example: Gradient descent univariate function

- Let's start (at random) with $z^0 = 2$.
- (gradient) $J'(2) = 2 \cdot 2 = 4$
- Let's update z is the opposite direction w.r.t. the gradient. Let's also define our step size as $\alpha = 0.25$
- $z^1 = z^0 - 0.25 \cdot 4 = 1$
- We now have $J'(1) = 2 \cdot 1 = 2$
- $z^2 = z^1 - 0.25 \cdot 2 = 0.5$
- $J'(0.5) = 2 \cdot 0.5 = 1$
- ....

Multiple inputs

- **Partial derivatives** measure how f changes as only the variable x_i increases at point $\boldsymbol{x}$

$$\frac{\partial}{\partial x_i} f(\boldsymbol{x}) = \lim_{\epsilon \to 0} \frac{f([x_0, \dots, x_{i-1}, x_i + \epsilon, x_{i+1}, \dots]) - f([x_0, x_1, \dots, x_i, \dots])}{\epsilon}$$

- **Gradient** is the generalization of derivative with respect to a vector of input variables

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = [\frac{\partial}{\partial x_1} f(\boldsymbol{x}), \dots, \frac{\partial}{\partial x_m} f(\boldsymbol{x})]$$

  - i.e. It is the vector of the partial derivatives

$$\{\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\}_i = \frac{\partial}{\partial x_i} f(\boldsymbol{x})$$

Critical points: all elements of the gradient are 0

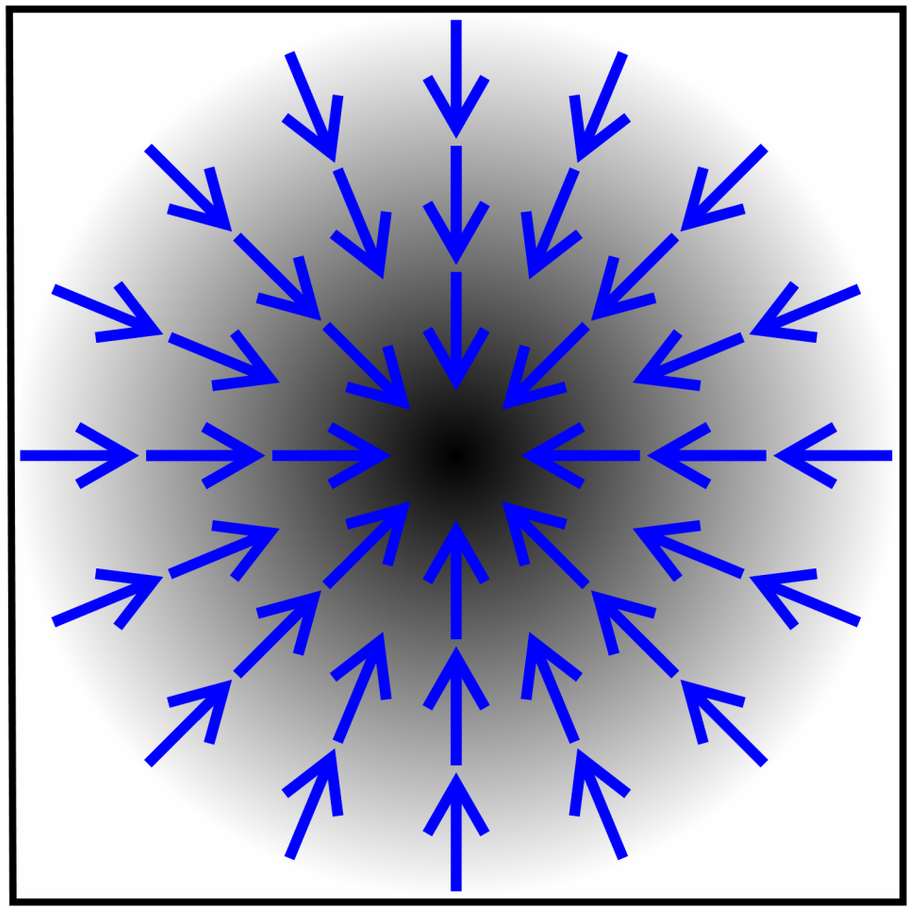
Multivariate calculus

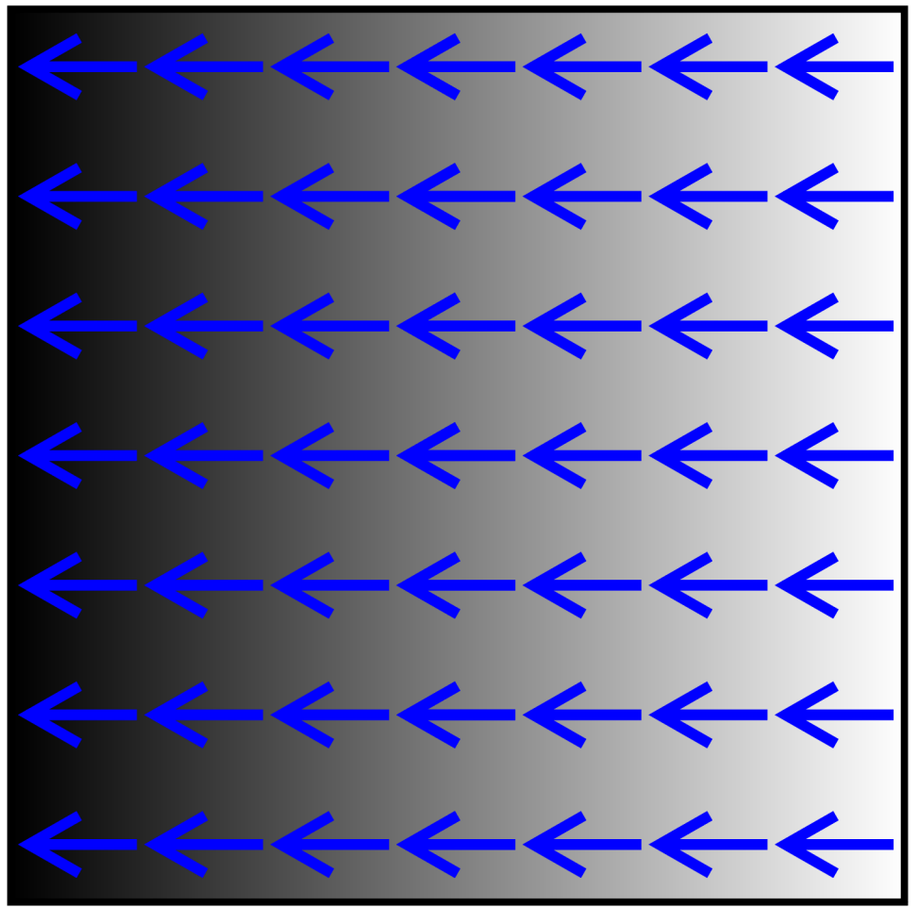
- The rules of partial differentiation follow exactly the same logic as univariate differentiation.
- When computing partial derivatives w.r.t. a variable, consider other variables as constants

- Chapter 5 of “Mathematics for Machine Learning”
- Derivation property for multivariate calcolus
- Example

Gradient

- The gradient points directly uphill, while the **negative** gradient points **downhill**





Gradient descent

Idea: We can decrease f moving in the direction of negative gradient

- Iterative algorithm, proposes a new point

$$\boldsymbol{x}' = \boldsymbol{x} - \alpha \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$$

  - $\alpha > 0$ is the **learning rate** (small)

- The algorithm converges when the gradient is zero (or very small)

- In some cases, we may be able to analytically solve $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = 0$. Not the case for neural networks

Jacobian

- If input and output are both vectors, the **Jacobian** matrix is the matrix of all the partial derivatives

$f: \mathbb{R}^m \to \mathbb{R}^n, \boldsymbol{J} \in \mathbb{R}^{n \times m}$

$$\boldsymbol{J} = \left[\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x_m}\right] =$$

$\frac{\partial}{\partial x_1} f(\boldsymbol{x})_1$	$\cdots$	$\frac{\partial}{\partial x_m} f(\boldsymbol{x})_1$
$\cdots$		
$\frac{\partial}{\partial x_1} f(\boldsymbol{x})_n$		$\frac{\partial}{\partial x_m} f(\boldsymbol{x})_n$

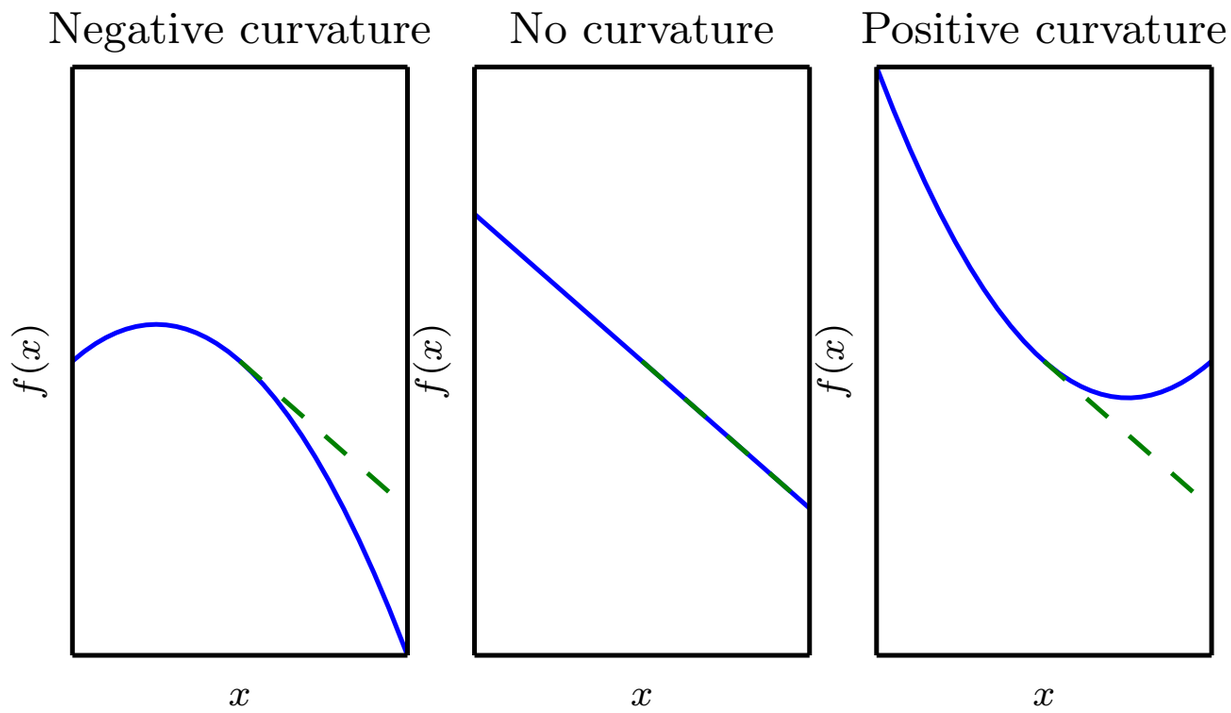
$\leftarrow \nabla_{\boldsymbol{x}} f(\boldsymbol{x})_1$

Second derivatives

First-order approximation

$$f(x+\epsilon) \approx f(x) + \epsilon f'(x)$$

- Second derivatives $\frac{d^2}{dx^2} f$ or $f''(x)$ measures the **curvature** of a function
- Provide information about the curvature of the function
- Consider quadratic functions, step size ϵ



$f''(x) < 0$

$f(x+\epsilon) < f(x) + \epsilon f'(x)$

$f''(x) > 0$

$f(x+\epsilon) > f(x) + \epsilon f'(x)$

$f''(x) = 0$

$f(x+\epsilon) = f(x) + \epsilon f'(x)$

Hessian matrix

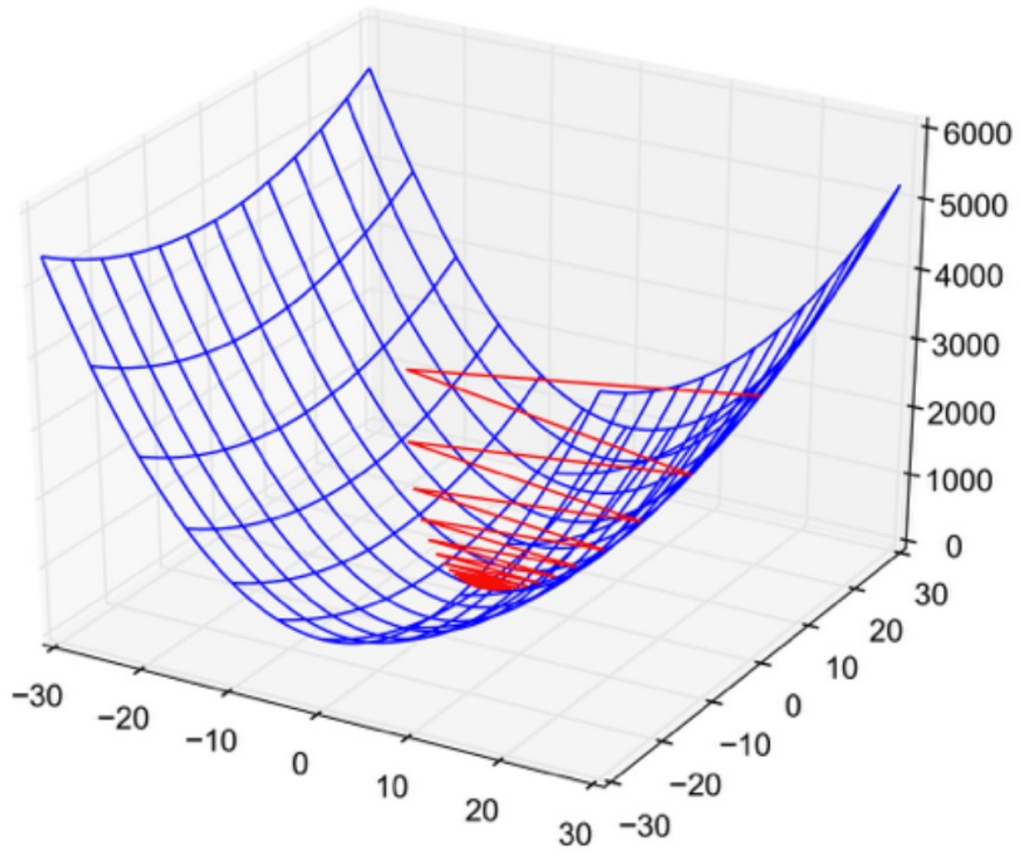
- The matrix of second derivatives (the Jacobian of the gradient)

$$f: \mathbb{R}^n \to \mathbb{R} \quad , H \in \mathbb{R}^{n \times n}, H_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f$$

$$H = \begin{array}{|c|c|c|} \hline \frac{\partial^2}{\partial x_1^2} f(\boldsymbol{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} f(x) \\ \hline \cdots & & \\ \hline \frac{\partial^2}{\partial x_n x_1} f(\boldsymbol{x}) & & \frac{\partial^2}{\partial x_n^2} f(\boldsymbol{x}) \\ \hline \end{array}$$

Condition number

- The ratio of the maximum and minimum nonzero eigenvalues of the Hessian matrix
  - The speed of convergence of gradient descent is dependent on the condition number
- Condition number give us information about the curvature in the different dimensions
  - poorly conditioned problems are long, thin valleys
    - very curved in one direction
    - very flat in the other
- Slow (need to compute Hessian) so not widely used for deep learning



Example: Linear Regression with GD

Let's consider a Perceptron WITHOUT hard-threshold:

$$f(\boldsymbol{x}; \boldsymbol{w}, b) = \boldsymbol{w}^T \boldsymbol{x} + b$$

And let's define Mean squared error as loss function

$$J(\boldsymbol{w}; b) = \frac{1}{2N_{Tr}} \sum_{(\boldsymbol{x}^{(i)}, t^{(i)}) \in Tr} \left(t^{(i)} - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}; b\right) \right)^2$$

Where N_{Tr} is the cardinality of the training set Tr.
The error function measures the mean squared error of the output with respect to the target value.

We would like to find values for $\boldsymbol{w}$ and b such that $J(\boldsymbol{w}; b)$ is MINIMIZED.

Example: Linear Regression with GD

- Let us compute the gradient w.r.t. $\boldsymbol{w}$:
  - For ease of notation, let us extend $\boldsymbol{x}$ with an entry fixed to 1 and let us incorporate b in $\boldsymbol{w}$

$$f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}$$

$$\mathrm{J}(\mathbf{w}) = \frac{1}{2N_{Tr}} \sum_{(\boldsymbol{x}^{(i)}, t^{(i)}) \in \mathrm{Tr}} \left(t^{(i)} - (\boldsymbol{w}^T \boldsymbol{x})\right)^2$$

- Example: gradient descent for linear regression

Matrix notation: Gradient of Linear Regression

$$\nabla_x f(\boldsymbol{x}) = [\frac{\partial}{\partial x_1} f(\boldsymbol{x}), \dots, \frac{\partial}{\partial x_m} f(\boldsymbol{x})]$$

- Let us compute the gradient w.r.t. $\boldsymbol{w}$:
  - For ease of notation, let us extend $\boldsymbol{x}$ with an entry fixed to 1 and let us incorporate b in $\boldsymbol{w}$

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \nabla_{\boldsymbol{w}} \frac{1}{2N_{Tr}} \sum_{(\boldsymbol{x}^{(i)}, t^{(i)}) \in \mathrm{Tr}} \left(t^{(i)} - (\boldsymbol{w}^T \boldsymbol{x}^{(\boldsymbol{i})})\right)^2$$

$$= \frac{1}{2N_{Tr}} \sum_{(\boldsymbol{x}^{(i)}, t^{(i)}) \in \mathrm{Tr}} \nabla_{\boldsymbol{w}} \left(t^{(i)} - (\boldsymbol{w}^T \boldsymbol{x}^{(\boldsymbol{i})})\right)^2 =$$

$$= -\frac{1}{2N_{Tr}} \sum_{(\boldsymbol{x}^{(i)}, t^{(i)}) \in \mathrm{Tr}} 2\left(t^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(\boldsymbol{i})}\right) \quad \boldsymbol{x}^{(\boldsymbol{i})} =$$

$$= -\frac{1}{N_{Tr}} \sum_{(\boldsymbol{x}^{(i)}, t^{(i)}) \in \mathrm{Tr}} \left(t^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(\boldsymbol{i})}\right) \quad \boldsymbol{x}^{(\boldsymbol{i})}$$

- Gradient descent update rule: $\boldsymbol{w}' = \boldsymbol{w} - \boldsymbol{\epsilon}(\nabla_{\boldsymbol{w}} J(\boldsymbol{w}))^T$