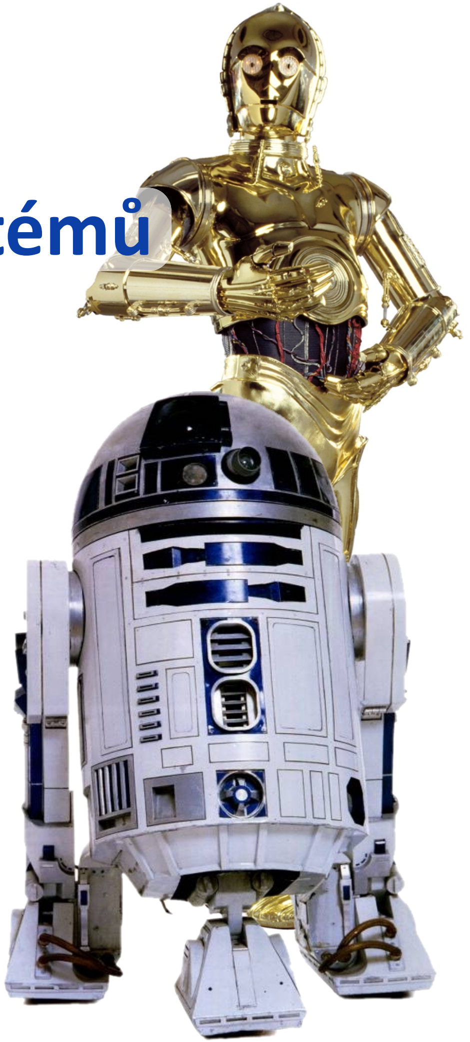
Teorie kognitivních systémů

2 Lineární regrese – I



- Lineární regrese jedné proměnné
- Odvození cenové funkce
- Minimalizace cenové funkce
- Odvození gradientní metody
- Algoritmus gradientního sestupu

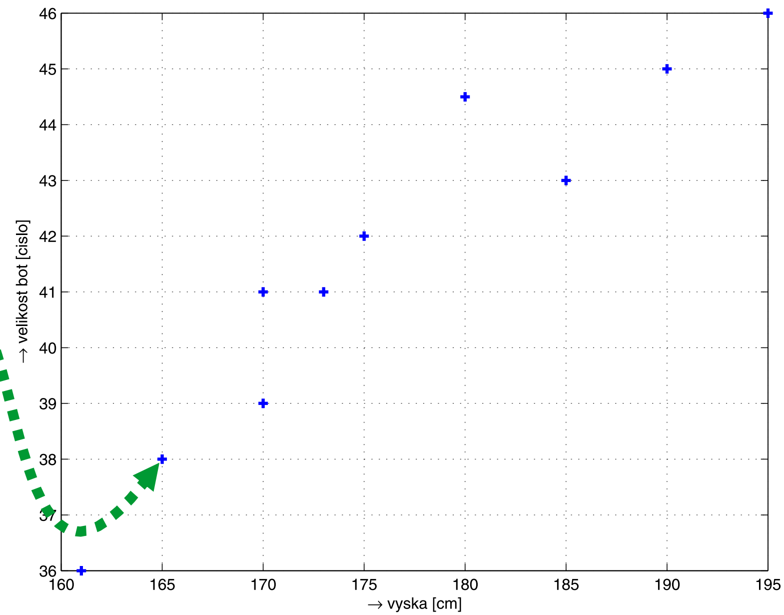
Lineární regrese
Nejjednodušší algoritmus učení s učitelem

- myšlenky vychází z „předpočítačového” období statistiky
- nejjednodušší algoritmus → na praktické úlohy z oblasti UI obvykle nepoužitelný, ale představuje základní stavební prvek a **pochopitelným způsobem demonstruje fundamentální myšlenky techniky regrese**
- jednoduchá, tj. poskytuje snadno interpretovatelný popis toho, jak vstupní data ovlivňují data výstupní
- při predikci často překonává komplikovanější nelineární metody, zejména je-li málo trénovacích dat, jsou-li data znehodnocená šumem nebo velmi řídká
- lineární regresi lze aplikovat na transformace vstupních dat, což značně rozšiřuje možnosti využití → **metody bázových funkcí** (*Basis-Function Methods*), např. RBF, GRBF, ...

Lineární regrese
Motivační příklad

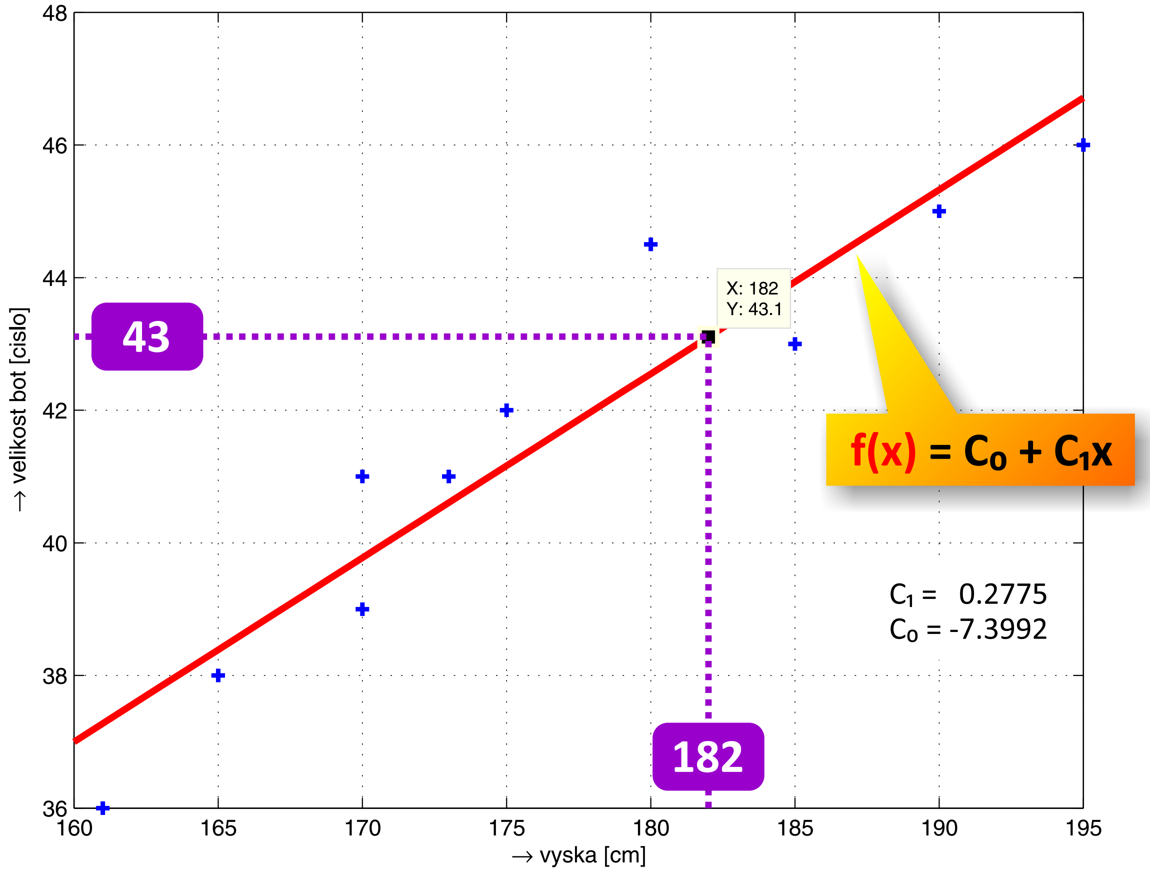
Závislost **čísla bot na výšce osoby**

- **učení s učitelem** (známe „správné" odpovědi v tréno-vací množině)
- **regresní úloha** (odhadovaný vý-stup je spojitý)



➔ potřebujeme data proložit **lineární funkcí** tak, abychom mohli pro libovolnou hodnotu vstupu (tj. výšky osoby na ose x) předpovědět velikost obuvi...

Lineární regrese
Motivační příklad



Lineární regrese
Motivační příklad – trénovací množina

Trénovací množina – pozorované páry (výška osoby, velikost obuvi)

Trénovací vzorek – např. $(x^{(5)}, y^{(5)})$

shoes <10x2 double>

	1	2
1	161	36
2	165	38
3	170	39
4	170	41
5	173	41
6	175	42
7	180	44.5000
8	185	43
9	190	45
10	195	46

$x^{(i)}$ $y^{(i)}$

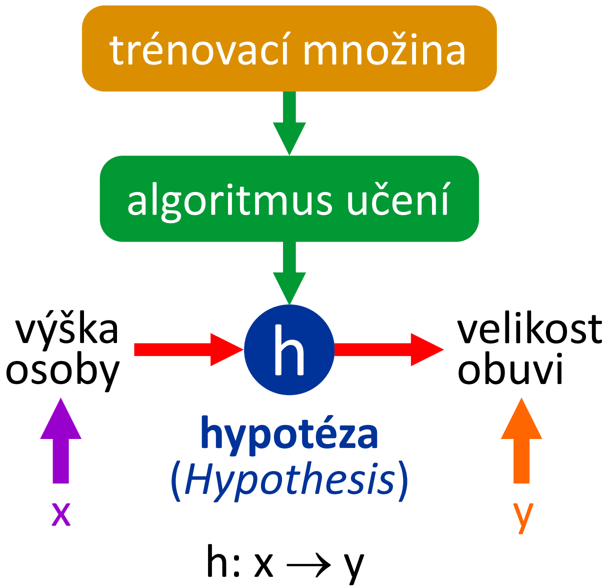
Značení:

m – celkový počet trénovacích vzorků (tj. velikost trénovací množiny čili počet řádek)

x – vstup (proměnná, obecně vektor příznaků)

y – výstup (proměnná, obvykle skalární výsledná predikovaná hodnota)

Lineární regrese
Schéma učení s učitelem



Reprezentace hypotézy h:

$h_\Theta(x) = \Theta_0 + \Theta_1 x$

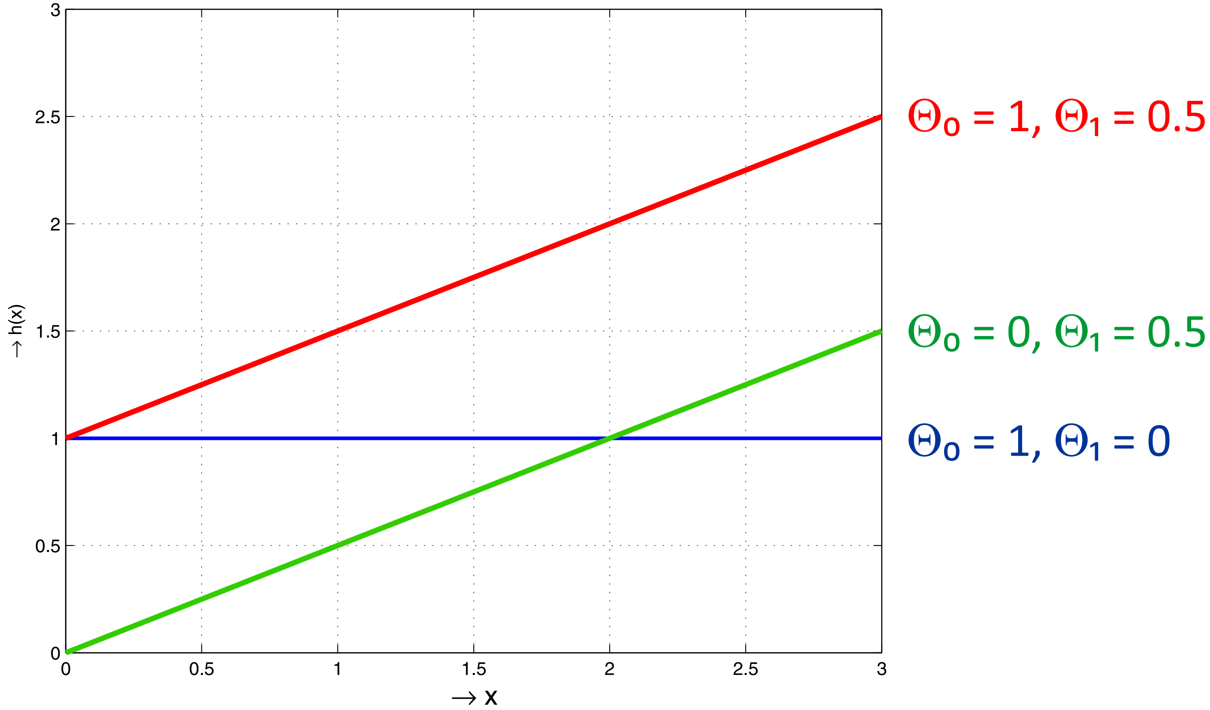
(nebo krátce h(x))

Lineární regrese jedné proměnné (*Univariate Linear Regression*)

Matematický model hypotézy závisí na 2 parametrech Θ_0, Θ_1

Cenová funkce
Jak nastavit parametry hypotézy?

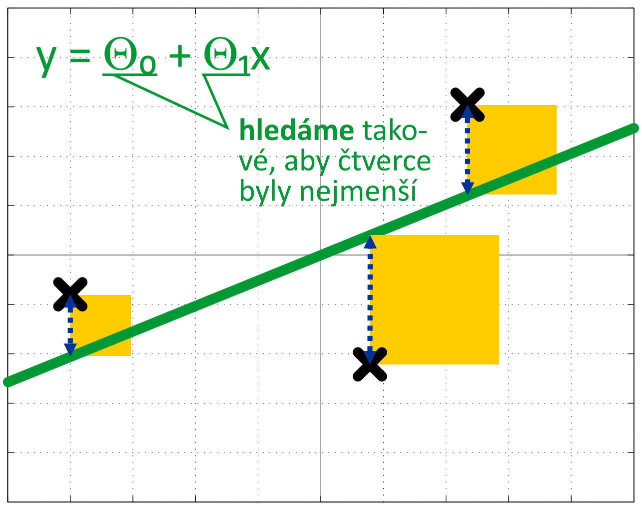
Hypotéza $h_\Theta(x)$ má tvar **lineární funkce** $h_\Theta(x) = \Theta_0 + \Theta_1 x$, tj.:



Cenová funkce
Jak nastavit parametry hypotézy?

Základní myšlenka: stanovit Θ_0, Θ_1 tak, aby hodnoty predikované hypotézou $h_\Theta(x)$ pro $x^{(i)}$ z trénovací množiny byly **co nejblíže** hodnotám $y^{(i)}$ z trénovací množiny:

$$\arg\min_{\Theta_0,\Theta_1} \underbrace{\left(\frac{1}{2m} \sum_{i=1}^{m} \left(h_\Theta(x^{(i)}) - y^{(i)}\right)^2 \right)}_{J(\Theta_0,\,\Theta_1)}$$



$J(\Theta_0, \Theta_1)$

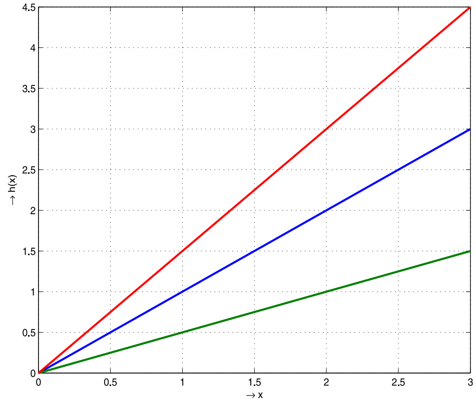
cenová funkce

(*Cost Function*)

→ hledáme takové parametry Θ_0 a Θ_1, pro které nabývá $J(\Theta_0, \Theta_1)$ minima...

Cenová funkce
Odvození – zjednodušená varianta

Pro získání představy o fungování cenové funkce zjednodušme hypotézu na tvar $h_\Theta(x) = \Theta_1 x$ (bude procházet počátkem s. s.):



Cenová funkce bude pak mít tento tvar:

$$J(\Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\Theta(x^{(i)}) - y^{(i)}\right)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left(\Theta_1 x^{(i)} - y^{(i)}\right)^2$$

Tuto cenovou funkci budeme minimalizovat, abychom našli optimální nastavení parametru Θ_1...

Cenová funkce

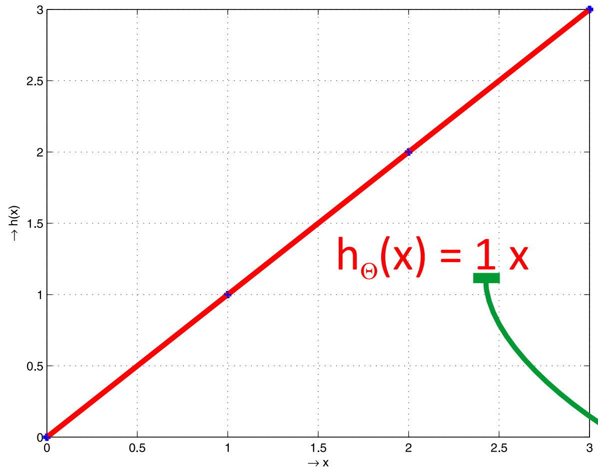
Odvození – zjednodušená varianta

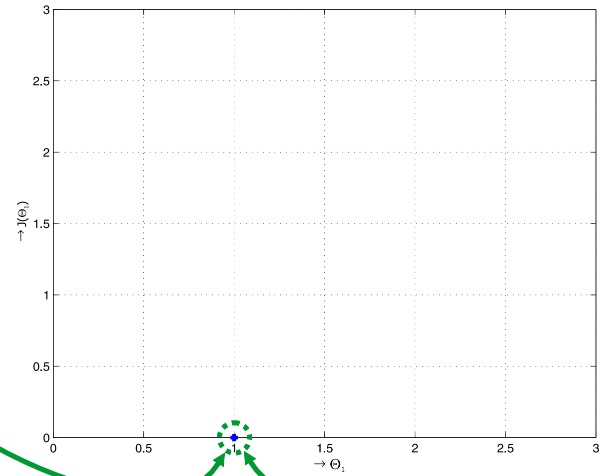
$h_\Theta(x)$

(pro dané Θ_1 je funkcí x)

$J(\Theta_1)$

(funkce parametru Θ_1)



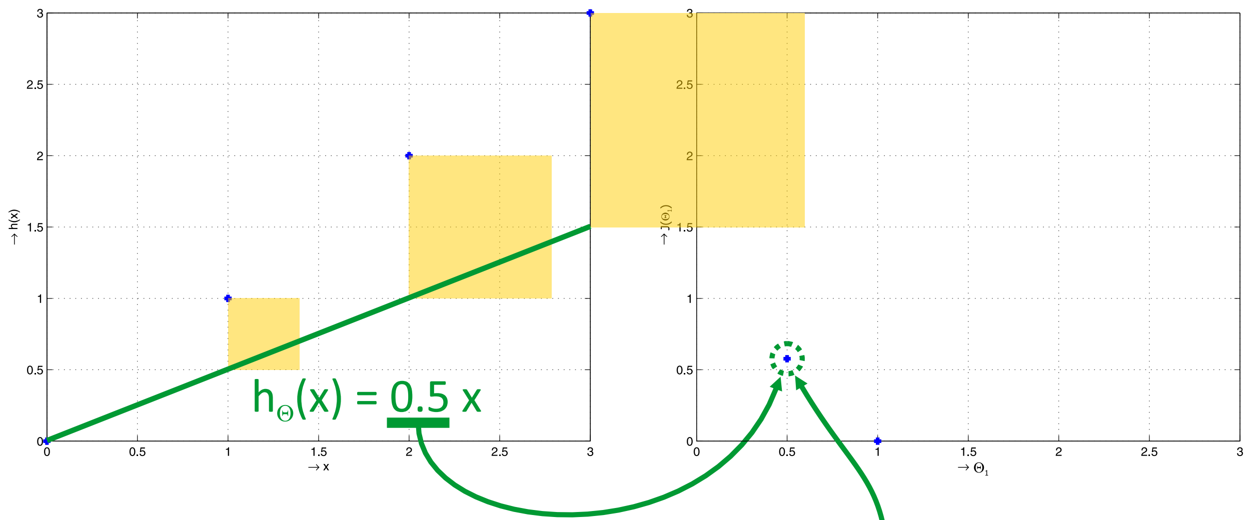


$$J(\Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\Theta_1 x^{(i)} - y^{(i)}\right)^2 = \frac{1}{2 \cdot 3}\left(0^2 + 0^2 + 0^2\right) = 0$$

Cenová funkce
Odvození – zjednodušená varianta

$h_\Theta(x)$
(pro dané Θ_1 je funkcí x)

$J(\Theta_1)$
(funkce parametru Θ_1)

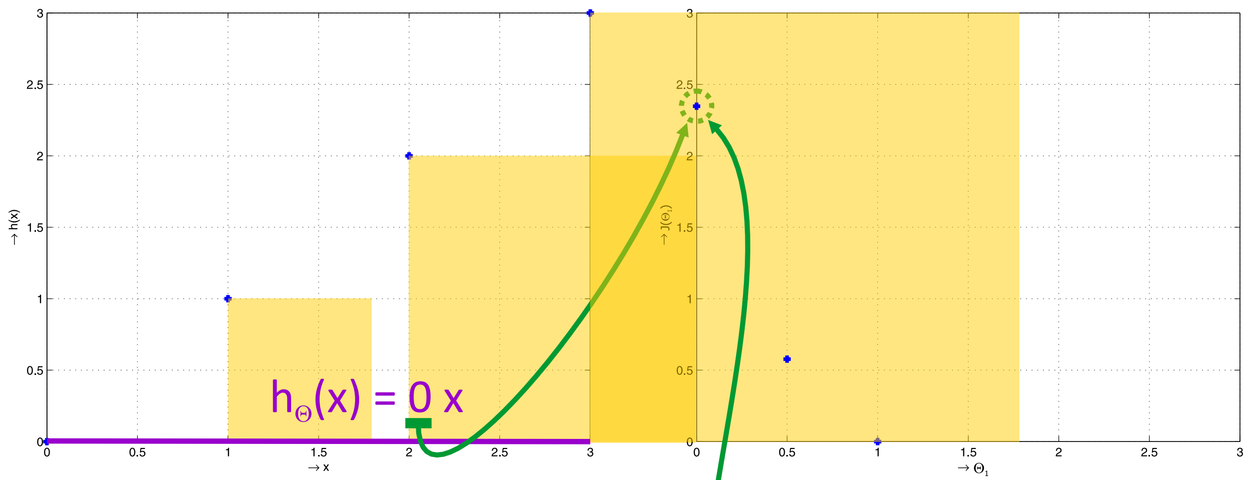


$$J(\Theta_1) = \frac{1}{2 \cdot 3}\left(\left(\frac{1}{2}\right)^2 + 1^2 + \left(\frac{3}{2}\right)^2\right) = 0.58\overline{3}$$

Cenová funkce
Odvození – zjednodušená varianta

$h_\Theta(x)$
(pro dané Θ_1 je funkcí x)

$J(\Theta_1)$
(funkce parametru Θ_1)

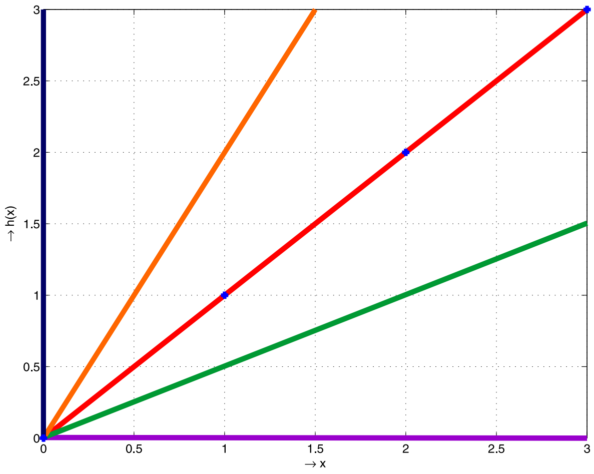


$$J(\Theta_1) = \frac{1}{2 \cdot 3}\left(1^2 + 2^2 + 3^2\right) = 2.\overline{3}$$

Cenová funkce
Odvození – zjednodušená varianta

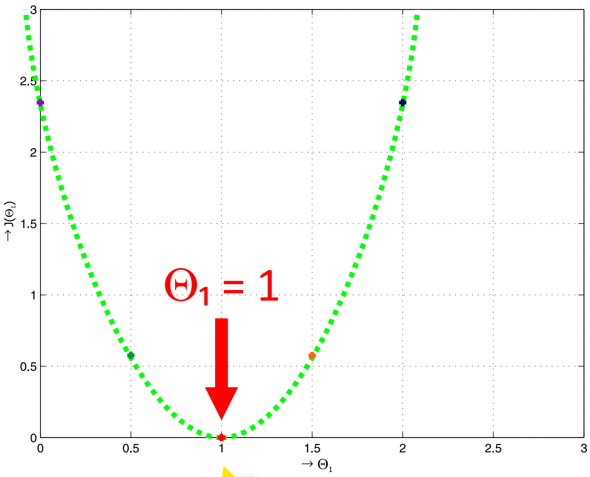
$h_\Theta(x)$

(pro dané Θ_1 je funkcí x)



$J(\Theta_1)$

(funkce parametru Θ_1)



$$\arg\min_{\Theta_1} J(\Theta_1) = 1$$

Cenová funkce
Odvození – varianta pro příklad s obuví

- **hypotéza** má tvar $h_\Theta(x) = \Theta_0 + \Theta_1 x$

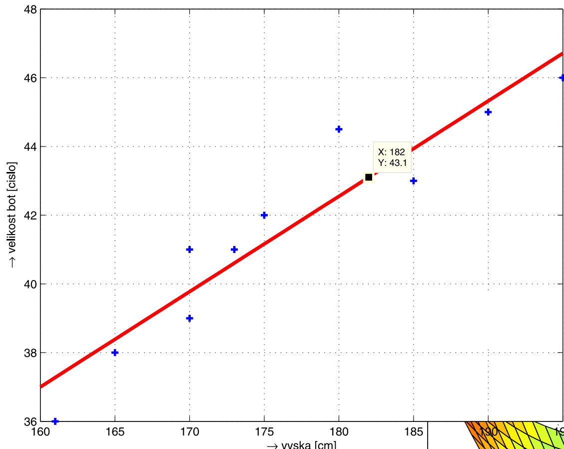
- **parametry** jsou tedy Θ_0, Θ_1

- **cenová funkce** je $J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\Theta(x^{(i)}) - y^{(i)}\right)^2$

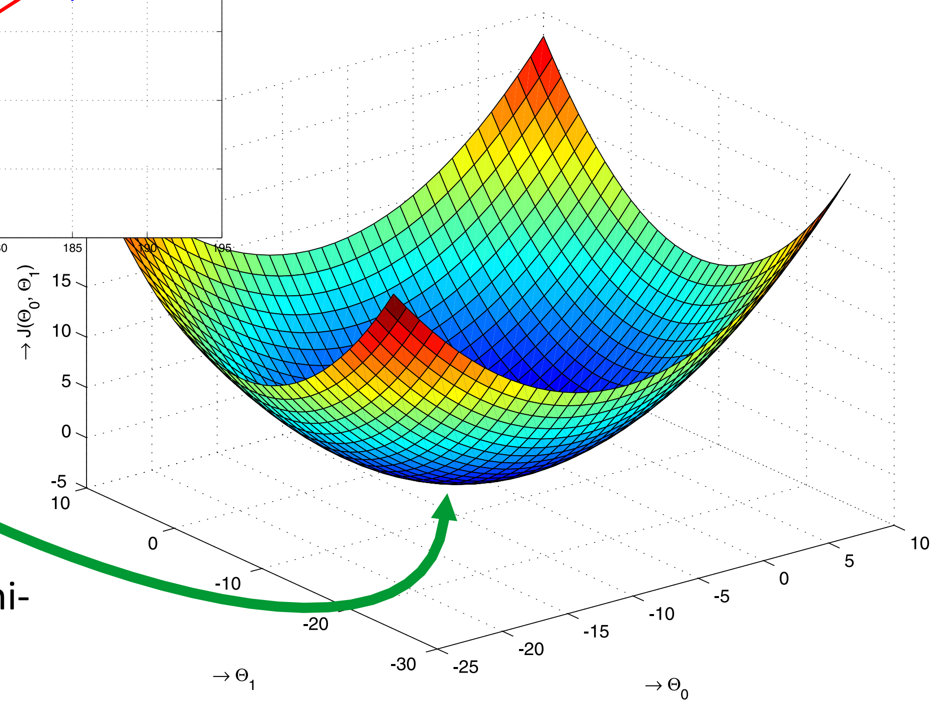
- cílem **optimalizace** je nalézt takové argumenty Θ_0, Θ_1 cenové funkce $J(\Theta_0, \Theta_1)$, aby pro ně funkce nabývala minima, tj. minimalizace funkce $J(\Theta_0, \Theta_1)$, neboli:

$$\arg\min_{\Theta_0,\Theta_1}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(h_\Theta(x^{(i)}) - y^{(i)}\right)^2\right)$$

Cenová funkce
Odvození – varianta pro příklad s obuví



cenová funkce $J(\Theta_0, \Theta_1)$



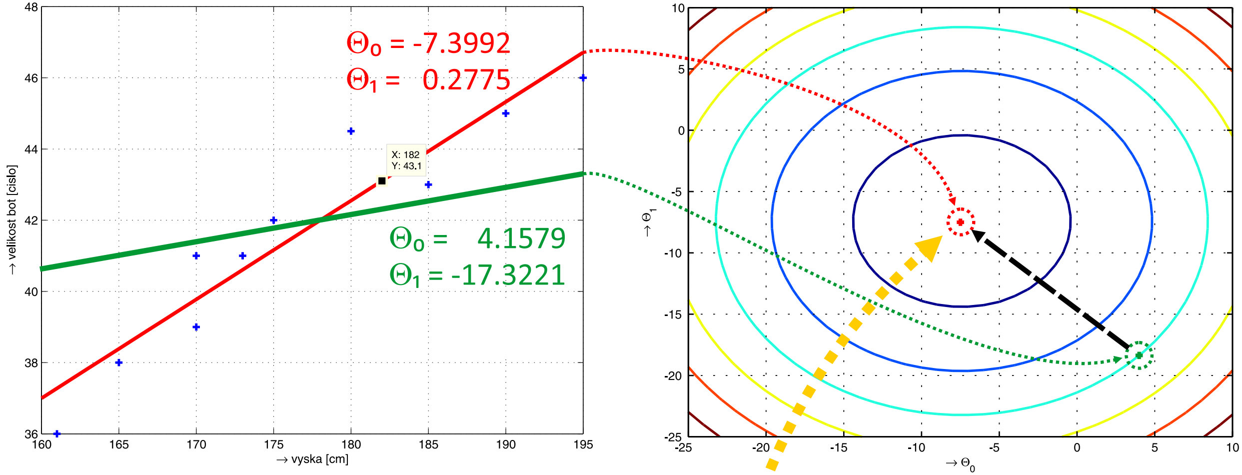
$\Theta_0 = -7.3992$
$\Theta_1 = \ \ 0.2775$

cenová funkce nabývá minima na souřadnicích (-7.3992, 0.2775)

Cenová funkce
Odvození – varianta pro příklad s obuví

$h_\Theta(x)$
(pro dané Θ_0, Θ_1 je funkcí x)

$J(\Theta_0, \Theta_1)$
(funkce parametrů Θ_0, Θ_1)



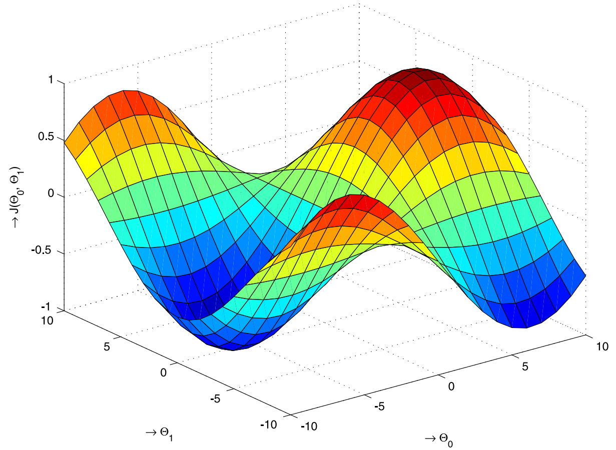
Jak nalézt (numericky) minimum cenové funkce $J(\Theta_0, \Theta_1)$?

Pomocí techniky **gradientního sestupu** (*Gradient Descent*)...

Gradientní sestup (*Gradient Descent*)
Nástin algoritmu

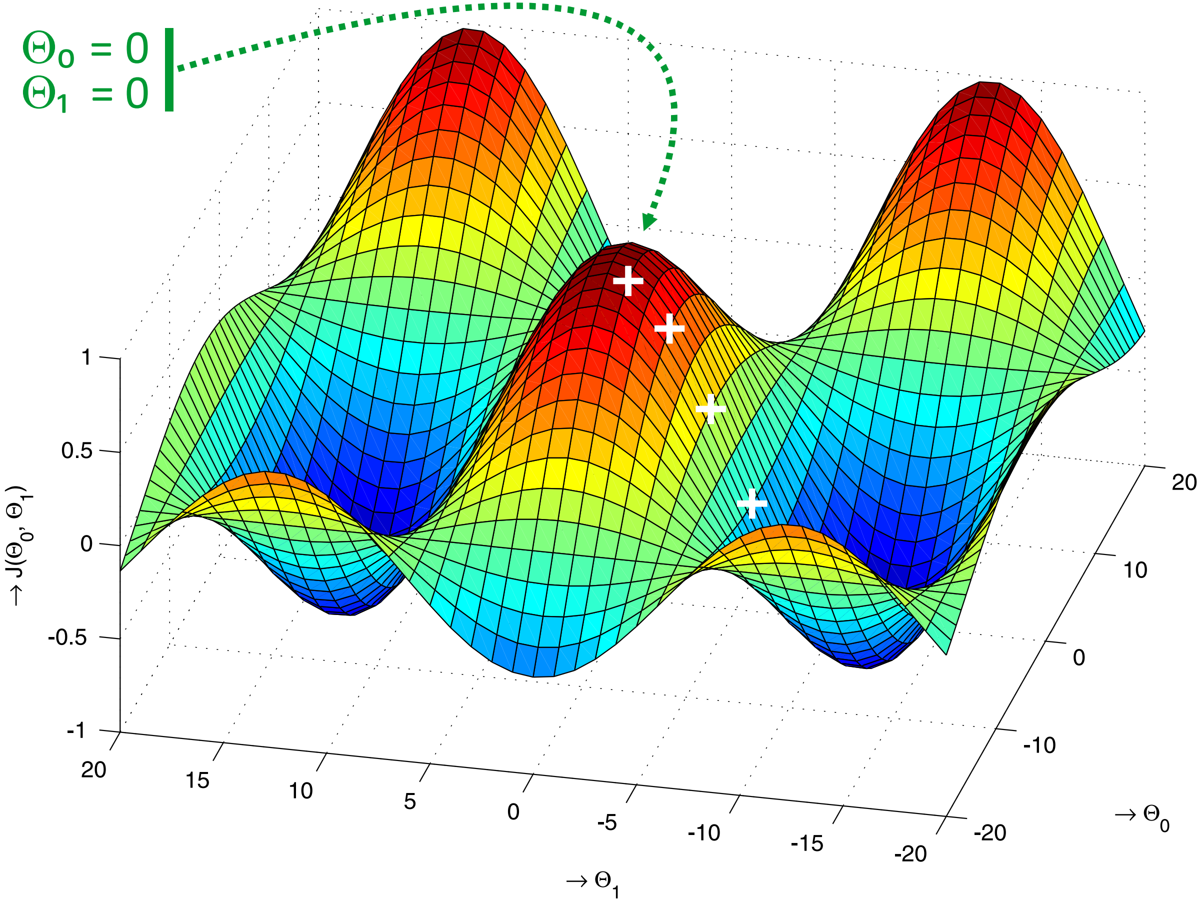
Je **dána** cenová funkce $J(\Theta_0, \Theta_1)$ nebo obecně $J(\Theta_0, \ldots, \Theta_n)$.
Chceme najít hodnoty Θ_i, pro které **nabývá minima**.

Postup:
- začneme s nějakými hodnotami Θ_i, např. 0 nebo náhodnými
- měníme hodnoty Θ_i tak, aby se snižovala funkční hodnota cenové funkce $J(\Theta_0, \ldots, \Theta_n)$, dokud nenalezneme její **minimum** (snad)



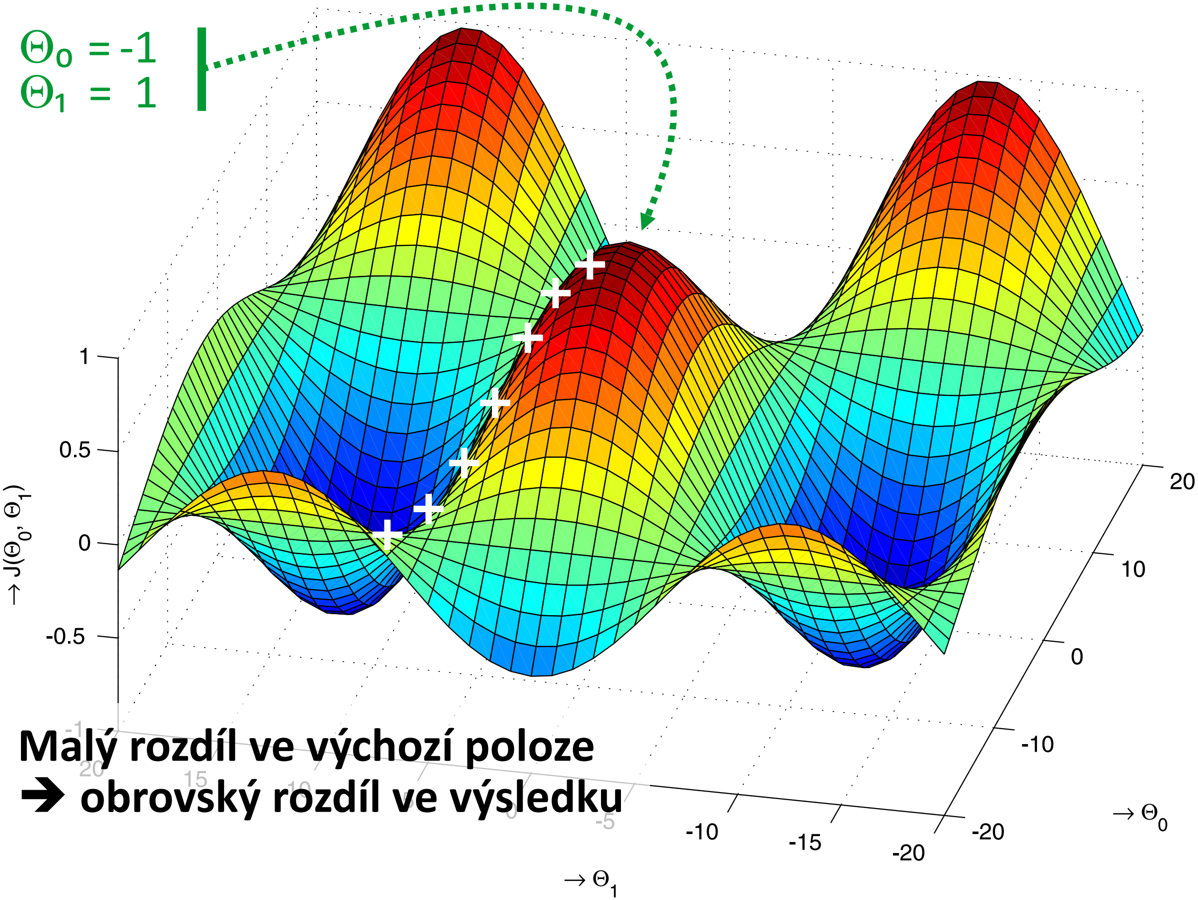
Gradientní sestup (*Gradient Descent*)
Problém algoritmu – lokalita extrému

$\Theta_0 = 0$
$\Theta_1 = 0$



Gradientní sestup (*Gradient Descent*)
Problém algoritmu – lokalita extrému

$\Theta_0 = -1$
$\Theta_1 = 1$



Malý rozdíl ve výchozí poloze
➔ obrovský rozdíl ve výsledku

Algoritmus gradientního sestupu
v pseudokódu

```
while not converged() do {
    for i = 1 .. n do
```

$$\Theta_i \leftarrow \Theta_i - \alpha \frac{\partial J(\Theta_0, \ldots, \Theta_n)}{\partial \Theta_i}$$

```
}
```

úprava hodnot Θ_i musí proběhnout „současně” (výpočtem ze stejných hodnot parametrů funkce J)

$$\begin{aligned} \texttt{temp[0]} &= \Theta_0 - \alpha \tfrac{\partial J(\Theta_0,\ldots,\Theta_n)}{\partial \Theta_0} \\ &\vdots \\ \texttt{temp[n]} &= \Theta_n - \alpha \tfrac{\partial J(\Theta_0,\ldots,\Theta_n)}{\partial \Theta_n} \end{aligned}$$

$$\begin{aligned} \Theta_0 &= \texttt{temp[0]} \\ &\vdots \\ \Theta_n &= \texttt{temp[n]} \end{aligned}$$

$$\begin{aligned} \texttt{temp[0]} &= \Theta_0 - \alpha \tfrac{\partial J(\Theta_0,\ldots,\Theta_n)}{\partial \Theta_0} \\ \Theta_0 &= \texttt{temp[0]} \\ &\vdots \\ \texttt{temp[n]} &= \Theta_n - \alpha \tfrac{\partial J(\Theta_0,\ldots,\Theta_n)}{\partial \Theta_n} \\ \Theta_n &= \texttt{temp[n]} \end{aligned}$$

Algoritmus gradientního sestupu
Matematický rozbor

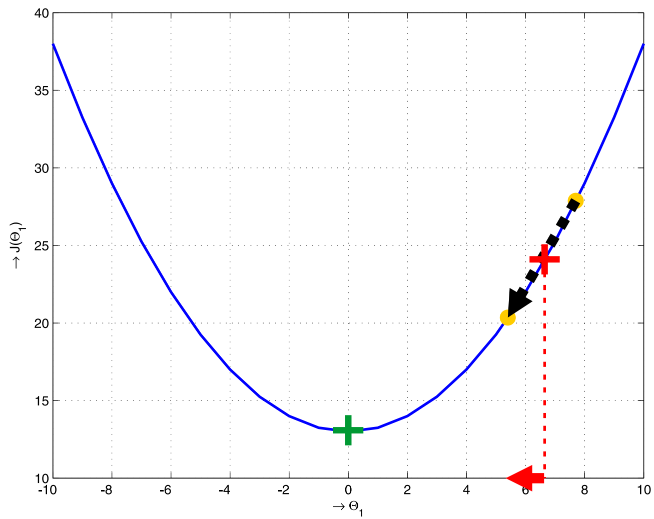
Úprava hodnot parametrů Θ_i:

$$\Theta_i \leftarrow \Theta_i - \alpha \frac{\partial J(\Theta_0, \ldots, \Theta_n)}{\partial \Theta_i}$$

míra učení
(*Learning Rate*)

parciální derivace
(*Partial Derivative*)

Geometrická interpretace derivace funkce v bodě:

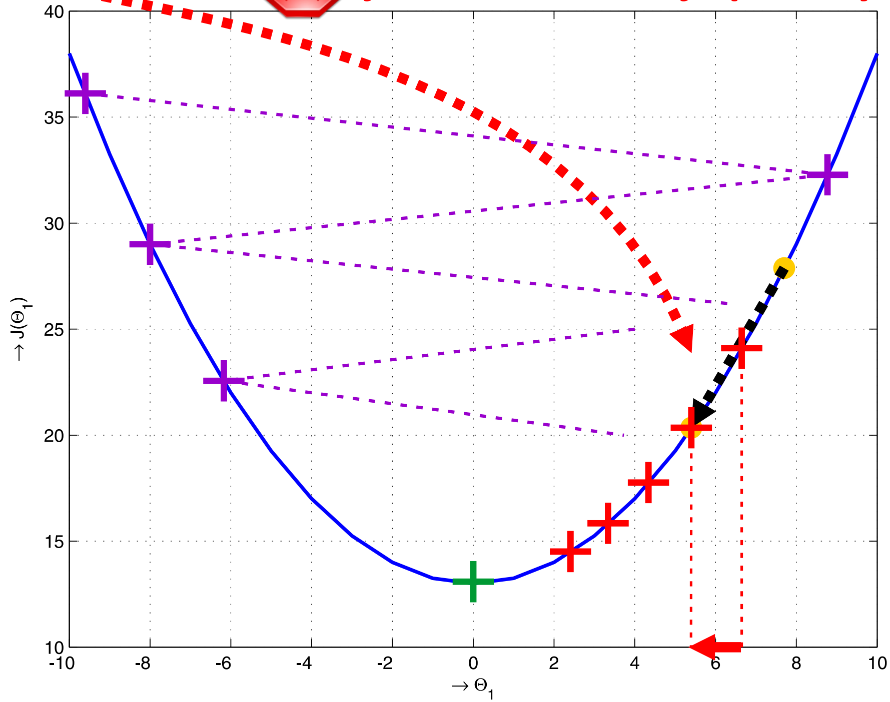


Algoritmus gradientního sestupu
Matematický rozbor

$$\Theta_i \leftarrow \Theta_i - \alpha \frac{\partial J(\Theta_0, \ldots, \Theta_n)}{\partial \Theta_i}$$

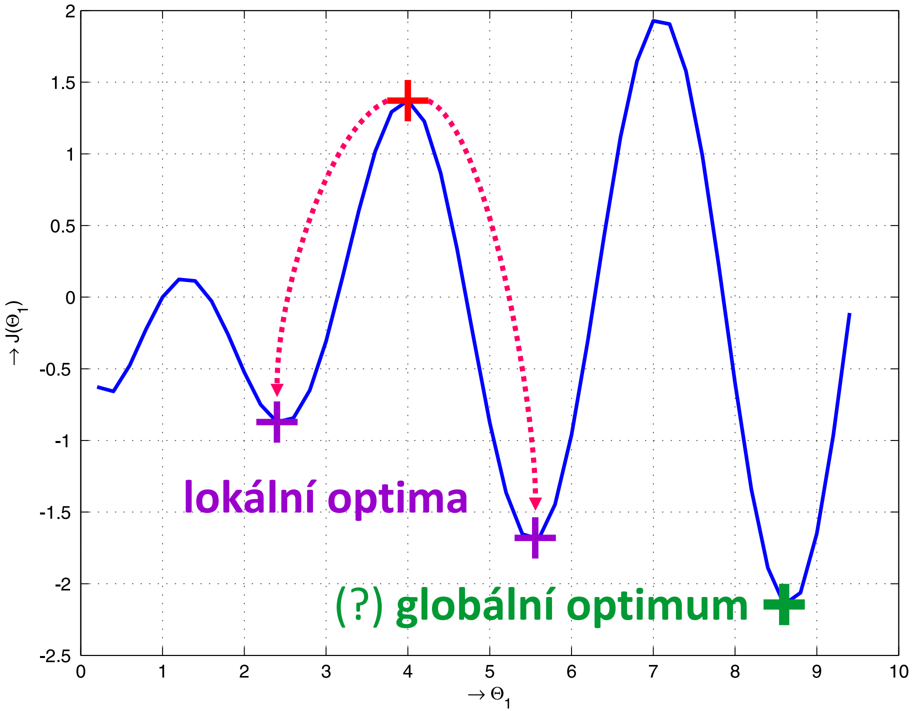
je-li α malé, GS je pomalý

je-li α velké, GS „přestře-luje" správnou pozici extrému – může diver-govat...



Algoritmus gradientního sestupu
Matematický rozbor – lokalita extrému

$$\Theta_i \leftarrow \Theta_i - \alpha \frac{\partial J(\Theta_0, \ldots, \Theta_n)}{\partial \Theta_i}$$



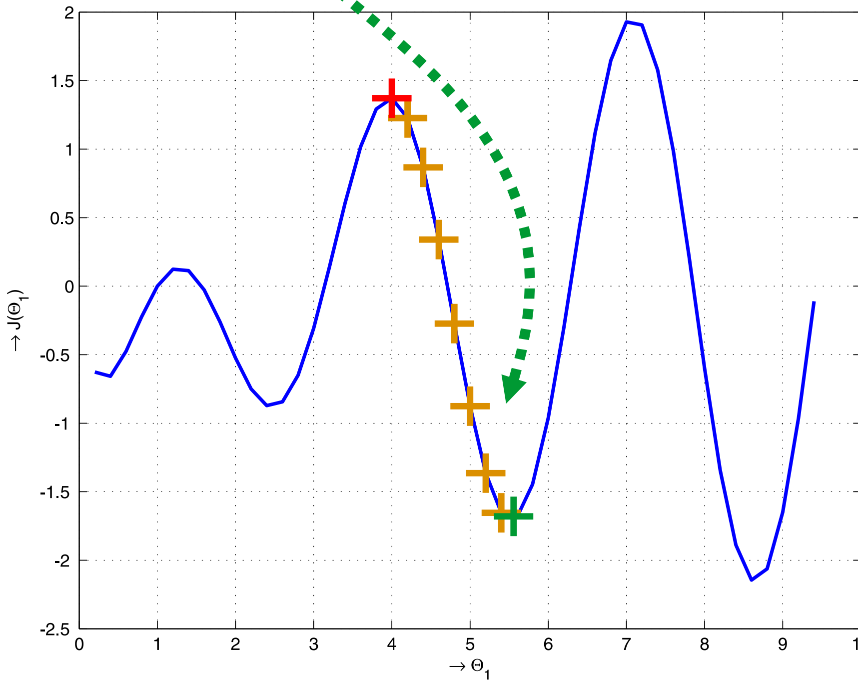
Algoritmus gradientního sestupu
Matematický rozbor – konvergence

$$\Theta_i \leftarrow \Theta_i - \alpha \frac{\partial J(\Theta_0, \ldots, \Theta_n)}{\partial \Theta_i}$$

vlastnost derivace

GS konverguje k lokálnímu minimu i tehdy, je-li α = C...

Jak se blížíme k lokálnímu minimu, krok GS se zmenšuje, tj. není nutné α adaptivně upravovat...



Gradientní sestup
pro potřeby lineární regrese

$$\Theta_i \leftarrow \Theta_i - \alpha \frac{\partial J(\Theta_0, \ldots, \Theta_n)}{\partial \Theta_i}$$

je třeba vyjádřit a spočítat tuto derivaci...

derivace složené funkce!

$$\frac{\partial J(\Theta_0, \Theta_1)}{\partial \Theta_{0\ldots 1}} = \frac{\partial}{\partial \Theta_{0\ldots 1}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\Theta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

$$= \frac{\partial}{\partial \Theta_{0\ldots 1}} \frac{1}{2m} \sum_{i=1}^{m} \left(\Theta_0 + \Theta_1 x^{(i)} - y^{(i)} \right)^2$$

pro Θ_0: $$\frac{\partial J(\Theta_0, \Theta_1)}{\partial \Theta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\Theta\left(x^{(i)}\right) - y^{(i)} \right)$$

pro Θ_1 : $$\frac{\partial J(\Theta_0, \Theta_1)}{\partial \Theta_1} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\Theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x^{(i)}$$

Gradientní sestup
pro potřeby lineární regrese – algoritmus

while not converged() **do** {

$$\mathtt{temp}[0] \;\leftarrow\; \Theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \Big(h_\Theta\big(x^{(i)}\big) - y^{(i)} \Big)$$

$$\mathtt{temp}[1] \;\leftarrow\; \Theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \Big(h_\Theta\big(x^{(i)}\big) - y^{(i)} \Big) \cdot x^{(i)}$$

$$\Theta_0 \;\leftarrow\; \mathtt{temp}[0]$$

$$\Theta_1 \;\leftarrow\; \mathtt{temp}[1]$$

}

Dávkový gradientní sestup (*Batch Gradient Descent*) – v každém kroku sestupu se výpočet provádí se všemi dostupnými trénovacími vzorky...