$$
\begin{array}{c}
 \\ n=25
\end{array}
\begin{array}{c}
\begin{array}{ccccc} x_0 & x_1 & x_2 & x_3 \end{array} \\
\left[\begin{array}{cccc}
1 & & & \\
1 & & & \\
1 & & & \\
1 & & A & \\
1 & & & \\
1 & & & \\
1 & & &
\end{array}\right]
\end{array}
\begin{array}{c}
y \\
\left[\begin{array}{c} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{array}\right]
\end{array}
=
\left[\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array}\right]
$$

$n=4$

$(25 \times 4) \quad (4 \times 1) \quad (25 \times 1)$

$$\theta = \underbrace{\underbrace{(A^T A)}_{B}{}^{-1} A^T}_{C} . Y$$

$\theta_0 = 0.14210$	$\theta_0 = 100$
$\theta_1 = 8.64061$	$\theta_1 = 7$
$\theta_2 = -2.00702$	$\theta_2 = -2$
$\theta_3 = 5.00001$	$\theta_3 = 5$
$J = 2215.47$	$J = 0$

solução encontrada. gráficos muitíssimo próximos

Gradient Descent X Normal Equation

- Needs to choose α
- Needs many iterations
- Works well even when $\underline{n}$ is large (even thousands of features)

$n = 10^4$ or greater

$O(K \cdot n^2)$

- No need for α
- Doesn't need to iterate
- Need to compute $(X^T X)^{-1}_{n \times n} \rightarrow O(n^3)$
- Slow if $\underline{n}$ is very large

$n = 100$ OK

$n = 1000$ OK

$n = 10000$ hmm... maybe

Normal equation usually doesn't work with more sofisticated learning algorithms

in this case, we apply the gradient descent

$(X^T X)$ não ~~é~~ será inversível se ...

1) houver colunas redundantes (proporcionais)
   Eg: $x_1 = size(ft^2)$, $x_2 = size(m^2)$
2) too many features $m \leq n$
   ↳ delete some features or use REGULARIZATION