Machine Learning for Engineering and Science Applications

Gradient Descent-2

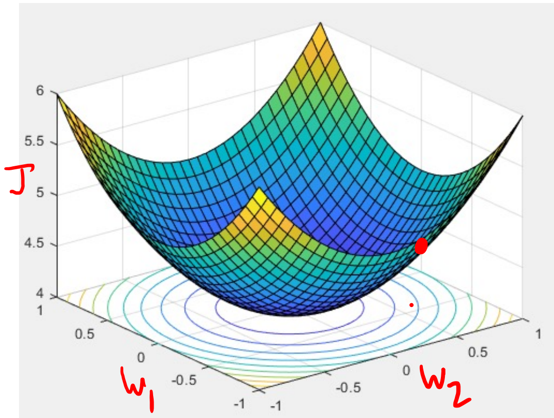
Proof of Steepest Descent

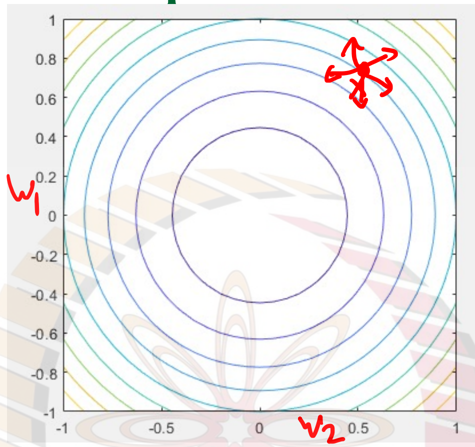
Numerical Gradient calculation

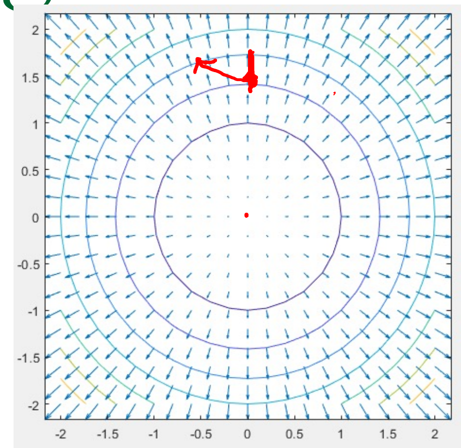
Stopping criteria

Gradient and steepest change

$\vec{W}$, $\vec{W} + \Delta\vec{W}$
J, $J + \Delta J$
Vectors $= \vec{\nabla J}$







Claim : The direction of maximum rate of change for a function $J(w)$ is given by $\nabla_w J$

Proof: Recall that rate of change in a given direction **v** is given by $\frac{\partial J}{\partial \mathbf{v}}$

That is, rate of change along **v** is $\frac{\partial J}{\partial v} = \vec{\nabla J} \cdot \hat{\mathbf{v}}$

$= |G||v|\cos\theta$

θ is $\angle$ between $\vec{G}$ & $\vec{v}$

$\theta = 0 \Rightarrow \vec{v}$ & $\vec{G}$ are $\parallel$

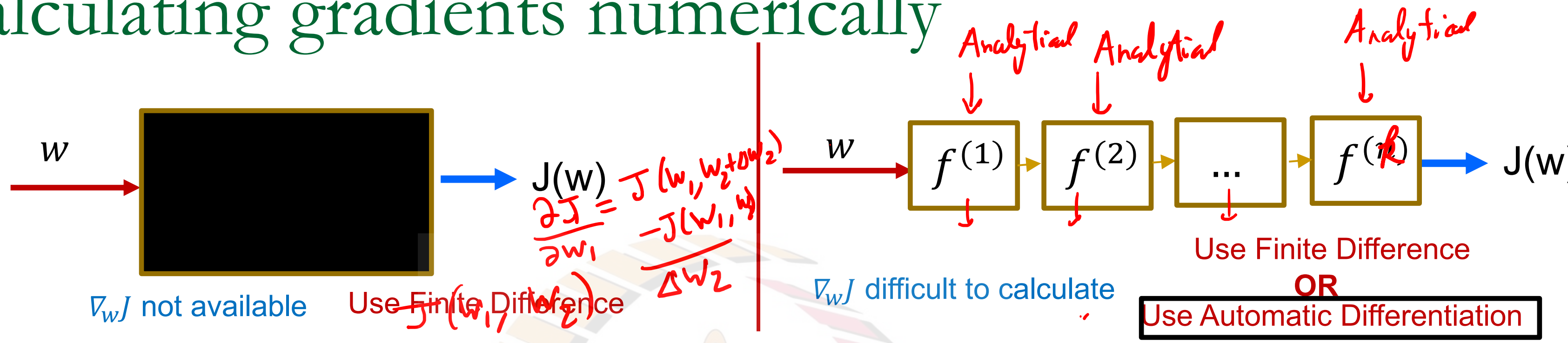
,where $G = \nabla J$ and θ is the angle between the gradient and v

This is a maximum/minimum when $\theta = 0$ and π respectively.

That is, the maximum increase is along the gradient and maximum decrease is opposite to the gradient.

Proved

Calculating gradients numerically



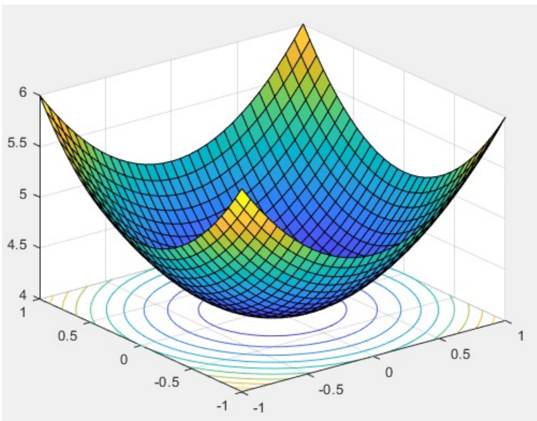
- In most cases, we do not have explicit expressions for the gradient.
- This can usually happen because of two reasons
  - There is no analytical expression for J as it is available only as a black box
  - The expression for J is available as a composition of functions and is too complicated.
- In both cases, the simplest solution is to use the Finite Difference Method

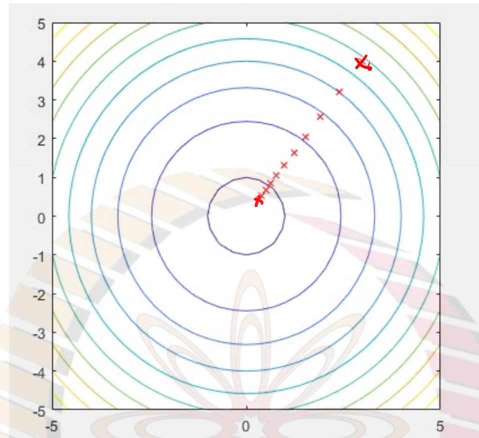
$$\frac{\partial J}{\partial w_j} \approx \frac{J(w_1, w_2, \dots w_j + \delta w_j, \dots w_n) - J(w_1, w_2, \dots w_j, \dots w_n)}{\delta w_j}$$

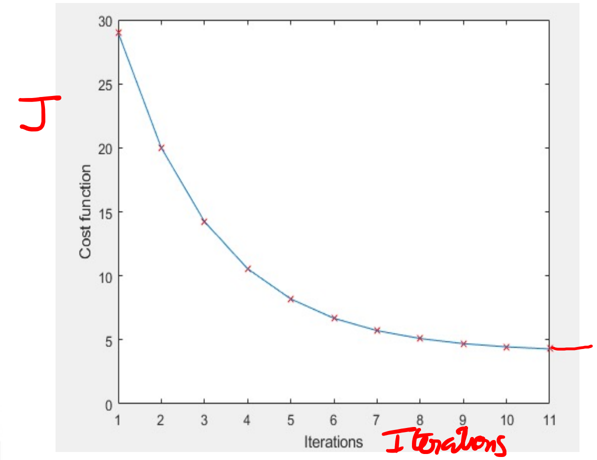
- This can, however, become very expensive if the number of features is large
- In case J is analytical but result of a chain $J = f\left(g\left(h(\dots)\right)\right)$ one can use Automatic Differentiation
  - This is a cheaper, numerical way to calculate the derivative via chain rule
  - For Neural Networks the equivalent method is called Backpropagation

Stopping criteria

$\vec{w}^{(3)} = [1.012 \quad 1.011]$
$\vec{w}^{(4)} = [1.009 \quad 1.010]$







- Ideally, we should stop when $\vec{\nabla_w J} = \vec{\mathbf{0}}$. This almost never happens in practice as
  - The number of iterations required could be infinite
  - Because of finite precision
- In practice, we decide on some precision ϵ (say, $\epsilon \approx 10^{-5}$).
- Multiple options for stopping criteria. Stop when
  - $\|\vec{w}^{k+1} - \vec{w}^{k}\| \leq \epsilon$ ①
  - $\|\nabla_w J(w^k)\| \leq \epsilon$ ②
  - $|(J(w^{k+1}) - J(w^{k+1})| \leq \epsilon$

Summary of the Gradient Descent procedure

1. Decide on α, ϵ and stopping criterion

2. Make an initial guess for $w = w^0$

3. Calculate $w^{k+1} = w^k - \alpha \nabla_w J$
   1. Calculate gradient numerically, if required

4. Calculate stopping criterion
   1. If satisfied, stop
   2. If not satisfied, go to Step 3