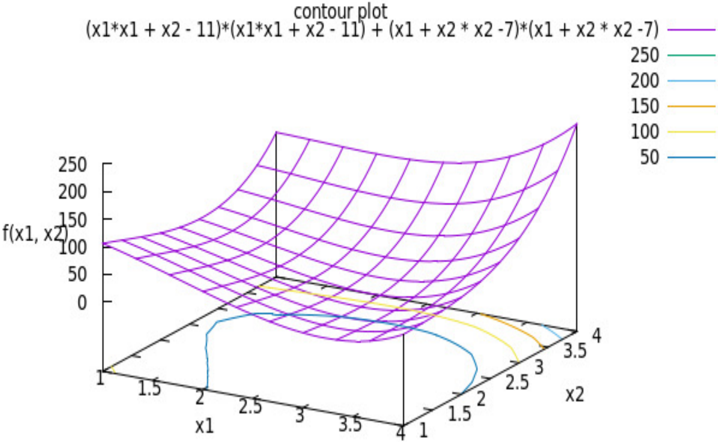
Deep Learning

Vijaya Saradhi

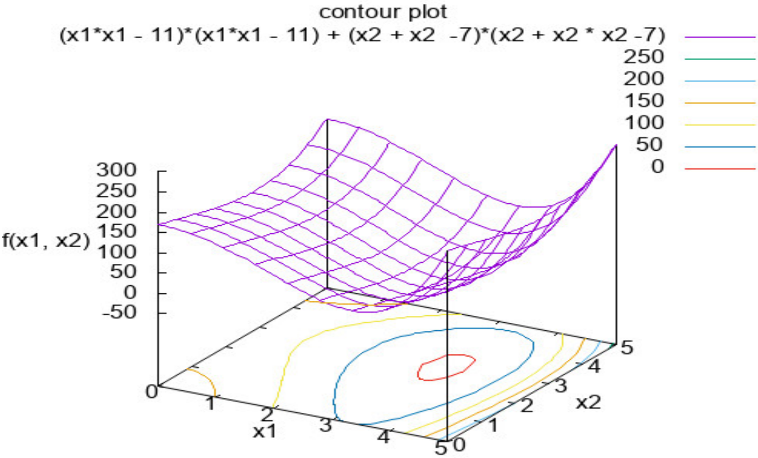
IIT Guwahati

Tue, 15th Sept 2020

Contours


contour plot
(x1*x1 + x2 - 11)*(x1*x1 + x2 - 11) + (x1 + x2 * x2 -7)*(x1 + x2 * x2 -7)
250
200
150
100
50
f(x1, x2)
250
200
150
100
50
0
x1
1
1.5
2
2.5
3
3.5
4
x2
1
1.5
2
2.5
3
3.5
4

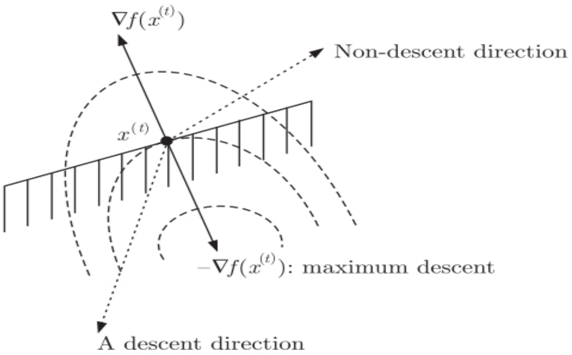
Contours


contour plot
(x1*x1 - 11)*(x1*x1 - 11) + (x2 + x2 -7)*(x2 + x2 * x2 -7)
250
200
150
100
50
0
300
250
200
150
100
50
0
-50
f(x1, x2)
0
1
2
x1
3
4
5
0
1
2
3
4
5
x2

Descent Direction

Definition

A search direction $\mathbf{d}^t$ is a descent direction at point $\mathbf{x}^t$ if the condition $\bigtriangledown f(\mathbf{x}^t).\mathbf{d}^t \leq 0$ is satisfied



Descent Direction

Condition

$$
\begin{aligned}
f(\mathbf{x}^{(t+1)}) &< f(\mathbf{x}^t) \\
f(\mathbf{x}^t + \alpha \bigtriangledown f(\mathbf{x}^t).\mathbf{d}^t) &< f(\mathbf{x}^t)
\end{aligned} \tag{1}
$$

That is function value at new point $\mathbf{x}^{(t+1)}$ is less than function value at the current point $\mathbf{x}^{(t)}$

Maximum Descent Direction

Condition

Let $\mathbf{d}^t = (1, 0)^T$ (arbitrary direction)

Let $\mathbf{x}^t = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Let the objective function be: $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$

Condition to be met: $\triangledown f(\mathbf{x}^t).\mathbf{d}^t \leq 0$

$(-46 - 38) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -46$

Maximum Descent Direction

Condition

When $\mathbf{d}^t = -\bigtriangledown f(\mathbf{x}^t)$ maximum decrease in function value is obtained
Let $\mathbf{d}^t = -1 \times (-46, -36)^T$ (direction = negative gradient) Let
$\mathbf{x}^t = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
Example: $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
When $\mathbf{d}^t = -\bigtriangledown f(\mathbf{x}^t) = \begin{pmatrix} 46 \\ 38 \end{pmatrix}$
$(-46 - 38) \begin{pmatrix} 46 \\ 38 \end{pmatrix} = -3560$

Gradient Descent

Find the steepest descent direction

Step 1 Choose: No. of iterations, $\mathbf{x}^{(0)}$, ϵ_1, ϵ_2; set $k = 0$

Step 2 Calculate $\bigtriangledown f(\mathbf{x}^{(k)})$

Step 3 if $\| \bigtriangledown f(\mathbf{x}^{(k)})\| \leq \epsilon_1$ then *terminate*

Step 4 Perform *uni-directional search* to find $\alpha^{(k)}$ using ϵ_2

- such that $f(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k)} - \alpha^{(k)} \bigtriangledown f(\mathbf{x}^{(k)}))$ is minimum
- Terminate when $\bigtriangledown f(\mathbf{x}^{(k+1)}). \bigtriangledown f(\mathbf{x}^{(k)}) \leq \epsilon_2$

Step 5 Increment k = k + 1; Repeat steps 2 to 5

Gradient Descent

Find the length to travel along the steepest descent direction

Step 1 Choose: No. of iterations, $\mathbf{x}^{(0)}$, ϵ_1, ϵ_2; set $k = 0$

Step 2 Calculate $\bigtriangledown f(\mathbf{x}^{(k)})$

Step 3 if $\| \bigtriangledown f(\mathbf{x}^{(k)})\| \leq \epsilon_1$ then *terminate*

Step 4 Perform *uni-directional search* to find $\alpha^{(k)}$ using ϵ_2

- such that $f(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k)} - \alpha^{(k)} \bigtriangledown f(\mathbf{x}^{(k)}))$ is minimum
- Terminate when $\bigtriangledown f(\mathbf{x}^{(k+1)}). \bigtriangledown f(\mathbf{x}^{(k)}) \leq \epsilon_2$

Step 5 Increment k = k + 1; Repeat steps 2 to 5

Gradient Descent

Example

minimize. $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$

Example

Step 1 Let k $= 0$; $\mathbf{x}^0 = (0, 0)^T$; $\epsilon_1 = \epsilon_2 = 0.001$

Gradient Descent

Example

minimize. $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$

Example

Step 1 Let k $= 0$; $\mathbf{x}^0 = (0,0)^T$; $\epsilon_1 = \epsilon_2 = 0.001$

Step 2 $\bigtriangledown f(\mathbf{x}^{(0)}) = (-14, -22)^T$;
$\| \bigtriangledown f(\mathbf{x}^{(0)})\| = ((-14)^2 + (-22)^2) = 680 > \epsilon_1$

Gradient Descent

Example

minimize. $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$

Example

Step 1 Let k = 0; $\mathbf{x}^0 = (0, 0)^T$; $\epsilon_1 = \epsilon_2 = 0.001$

Step 2 $\bigtriangledown f(\mathbf{x}^{(0)}) = (-14, -22)^T$;
$\| \bigtriangledown f(\mathbf{x}^{(0)})\| = ((-14)^2 + (-22)^2) = 680 > \epsilon_1$

Step 4 In the direction $- \bigtriangledown f(\mathbf{x}^{(0)})$ perform unidirection search

- Steepest descent direction vector is: $(14, 22)^T$
- Find α^0 such that $f(\mathbf{x}^1) = f(\mathbf{x}^0 - \alpha^0 \bigtriangledown f(\mathbf{x}^{(0)}))$ is minimum
- Let us compute: $\mathbf{x}^1 = \mathbf{x}^0 - \alpha^0 \bigtriangledown f(\mathbf{x}^{(0)})$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} - \alpha^0 \times \begin{pmatrix} -14 \\ -22 \end{pmatrix} = \begin{pmatrix} 14\alpha^0 \\ 22\alpha^0 \end{pmatrix}$$

Gradient Descent

Example

Step 4 To find α^0, minimize the function $f(\mathbf{x}^1)$

- We have computed

$$\mathbf{x}^1 = \begin{pmatrix} 14\alpha^0 \\ 22\alpha^0 \end{pmatrix}$$

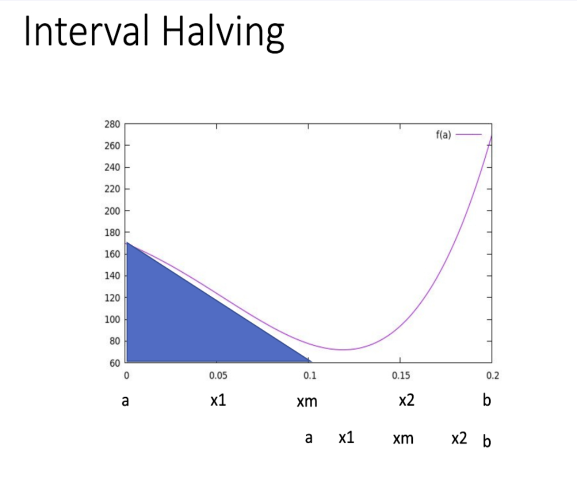
- Therefore

$$f(\mathbf{x}^1) = f\begin{pmatrix} 14\alpha^0 \\ 22\alpha^0 \end{pmatrix}$$

- Substituting in objective function $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ we have:
- $f(\mathbf{x}^1) = ((14\alpha^0)^2 + (22\alpha^0) - 11)^2 + ((14\alpha^0) + (22\alpha^0)^2 - 7)^2$
- Minimize $f(\mathbf{x}^1)$ to find best α^0

Gradient Descent

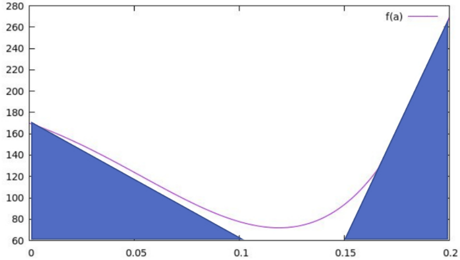
Example



Gradient Descent

Example

Interval Halving



a x1 xm x2 b

a xm b

Gradient Descent

Example

Step 4 Using Interval halving method or any other single variable optimization procedure we obtain $\alpha^0 = 0.127$. Compute $\mathbf{x}^1 = (\mathbf{x}^0 - \alpha^0 \bigtriangledown f(\mathbf{x}^0)) = (14\alpha^0, 22\alpha^0) = (1.788, 2.810)^T$

Step 4 Since the termination condition does not satisfy

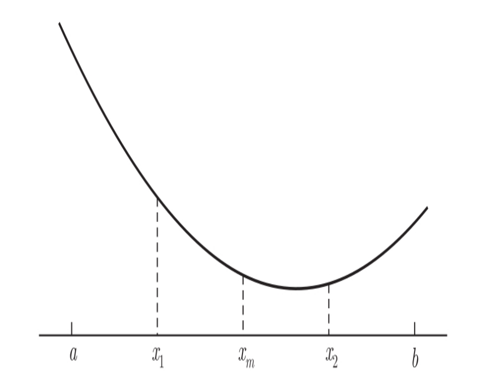
- Terminate when $\bigtriangledown f(\mathbf{x}^{(1)}). \bigtriangledown f(\mathbf{x}^{(0)}) \leq \epsilon_2$
- $\bigtriangledown f(\mathbf{x}^{(1)}) = (30.707, -18.803)^T$
- $\bigtriangledown f(\mathbf{x}^{(0)}) = (-14, -22)^T$
- $$(30.707, -18.803) \begin{pmatrix} -14 \\ -22 \end{pmatrix} \leq \epsilon_2?$$

Step 5 increment k = k + 1; that is k = 1; Repeat the algorithm until termination criteria is met

The optimization obtains $\mathbf{x}^*$ as $(3.008, 1.999)^T$

Single variable optimization

Interval halving method



Single variable optimization

Interval halving method

- Given interval (a, b)
- If $f(x_1) < f(x_m)$ then minimum cannot lie beyond x_m
  That is $f(x_1) < f(x_{m+1}) < \cdots < f(b)$
- The interval will reduce to (a, x_m)
- If $f(x_1) > f(x_m)$ then minimum cannot lie in (a, x_1)

Single variable optimization

Interval halving method - algorithm

Step 1 Given interval (a, b), choose ϵ. Let $x_m = \frac{(a+b)}{2}$; $L = (b - a)$

Step 2 Initialize $x_1 = a + \frac{L}{4}$; $x_2 = b - \frac{L}{4}$; Compute $f(x_1), f(x_2)$

Step 3 If $f(x_1) < f(x_m)$ then $b = x_m; x_m = x_1$; Go to step 5; else go to step 4

Step 4 If $f(x_2) < f(x_m)$ then $a = x_m; x_m = x_2$; Go to step 5; else $a = x_1, b = x_2$; go to step 5;

Step 5 Calculate $L = (b - a)$. If ($|L| < \epsilon$) terminate else go to step 2

Text books to read

Optimization

- Engineering Optimization - Theory and Practice Singiresu S Rao
- Chapter 1 of the above book, sections 6.8 and 6.9
- mec.nit.ac.ir/file_part/master_doc/20149281833165301436305785.pdf
- Optimization for Engineering Design Kalyanmoy Deb
- Section 3.4 of the above book.

Perceptron

Notataion

Data set is of the form $(\mathbf{x}_i, d_i)_{i=1}^{N}$

Where N is the number of data points say N = 1000 emails

Each data point has m attributes.

$$i^{th} \text{ Data point's } m \text{ attributes are: } \mathbf{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{im} \end{pmatrix}$$

d_i is the desired response that is true label of the example

Assumption Data points are linearly separable

Linearly Separable Data

Separate data using lines

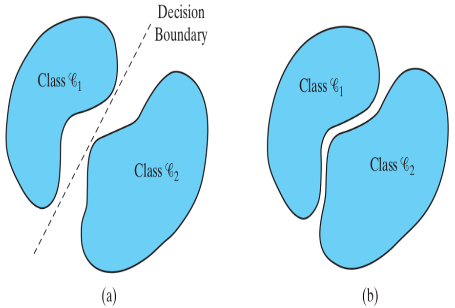


FIGURE 1.4 (a) A pair of linearly separable patterns. (b) A pair of non-linearly separable patterns.

Perceptron

Prelimininaries

- Let $\mathcal{C}_1, \mathcal{C}_2$ are linearly separable
- $\sum_{i=1}^{m} w_i x_i$ is written as:

$$\left(w_1 w_2 \cdots w_m\right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}$$

$$\mathbf{w}^T \mathbf{x}$$

Perceptron

Prelimininaries

- Let the following hold for these classes:

$$\begin{aligned} \mathbf{w}^T\mathbf{x} > 0 \quad \forall\, \mathbf{x} \in \mathcal{C}_1 \\ \mathbf{w}^T\mathbf{x} \leq 0 \quad \forall\, \mathbf{x} \in \mathcal{C}_2 \end{aligned}$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$

Perceptron

Prelimininaries

- Let the following hold for these classes:

$$\begin{aligned}\underbrace{\mathbf{w}^T\mathbf{x}}_{AdderFunction} &> 0 \quad \forall\, \mathbf{x} \in \mathcal{C}_1 \\ \underbrace{\mathbf{w}^T\mathbf{x}}_{AdderFunction} &\leq 0 \quad \forall\, \mathbf{x} \in \mathcal{C}_2\end{aligned}$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$

Perceptron

Prelimininaries

- Let the following hold for these classes:

$$\begin{aligned} \mathbf{w}^T\mathbf{x} &\underbrace{> 0}_{Threshold function} \quad \forall\, \mathbf{x} \in \mathcal{C}_1 \\ \mathbf{w}^T\mathbf{x} &\underbrace{\leq 0}_{Threshold function} \quad \forall\, \mathbf{x} \in \mathcal{C}_2 \end{aligned}$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$

Perceptron

Update Rule 01

- Input $\mathbf{x}$ cannot be changed by the learning method
- Learning method should only change $\mathbf{w}$
- Initialize $\mathbf{w}$.
- If the following is not violated then there is no change in $\mathbf{w}$

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \quad \mathbf{x}(n) \in \mathcal{C}_1 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \leq 0 \quad \mathbf{x}(n) \in \mathcal{C}_2 \end{aligned}$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$

Perceptron

Update Rule 02

- If the following is violated then there is no change in $\mathbf{w}$

$$\begin{array}{lll}\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)\mathbf{x}(n) & \text{if } \mathbf{w}^T(n)\mathbf{x}(n) > 0 & \mathbf{x}(n) \in \mathcal{C}_2 \\ \mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{x}(n) & \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \leq 0 & \mathbf{x}(n) \in \mathcal{C}_1\end{array}$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$

Perceptron

Update Rule 02 - Close look

- Correct rule: $\mathbf{w}^T(n)\mathbf{x}(n) > 0$ for every $\mathbf{x}(n) \in \mathcal{C}_1$
- Violation is due to $\mathbf{w}^T(n)\mathbf{x}(n) > 0$ holds but $\mathbf{x}(n) \in \mathcal{C}_2$
- When $\mathbf{x}(n) \in \mathcal{C}_2$ we should have the quantity $\mathbf{w}^T(n)\mathbf{x}(n) \leq 0$
- In order to reduce the quantity $\mathbf{w}^T(n)\mathbf{x}(n)$, the only choice is to reduce from $\mathbf{w}(n)$ some quantity equal to:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)\mathbf{x}(n) \quad \text{if} \mathbf{w}^T(n)\mathbf{x}(n) > 0 \quad \mathbf{x}(n) \in \mathcal{C}_2$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$

Perceptron

Update Rule 02 - Close look

- Correct rule: $\mathbf{w}^T(n)\mathbf{x}(n) \leq 0$ for every $\mathbf{x}(n) \in \mathcal{C}_2$
- Violation is due to $\mathbf{w}^T(n)\mathbf{x}(n) \leq 0$ holds but $\mathbf{x}(n) \in \mathcal{C}_1$
- When $\mathbf{x}(n) \in \mathcal{C}_1$ we should have the quantity $\mathbf{w}^T(n)\mathbf{x}(n) > 0$
- In order to increase the quantity $\mathbf{w}^T(n)\mathbf{x}(n)$, the only choice is to increase from $\mathbf{w}$ some quantity equal to:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{x}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \leq 0 \quad \mathbf{x}(n) \in \mathcal{C}_1$$

- The case that if $\mathbf{w}^T\mathbf{x} = 0$ then $\mathbf{x} \in \mathcal{C}_2$