Derivation of gradient descent training rule for a single unit with output O

where $O = w_0 + w_1 x_1 + x_2 x_2 + x_1 x_1^2 + x_2 x_2^2 + \cdots + x_n x_n + x_n x_n^2$ —(*)

Derivation

We first need to note that

=> The training Error function depending on weights and bias is given by

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - O_d)^2 \quad \text{———} (1)$$

Derivation of gradient descent training rule for a single unit with output O

where $O = w_0 + w_1 x_1 + x_2 x_2 + x_1 x_1^2 + x_2 x_2^2 + \cdots + x_n x_n + x_n x_n^2$ —(*)

Derivation

We first need to note that

=> The training Error function depending on weights and bias is given by

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - O_d)^2 \quad \text{———} (1)$$

where t_d => is our training value

O_d => is output function value

=> Note that in our given output function with bias and weight given as in (*), It is represented as some linear unit input and some polynomial function which

Derivation of gradient descent training rule for a single unit with output O
where $O = w_0 + w_1 x_1 + x_2 x_2 + x_1 x_1^2 + x_2 x_2^2 + \dots + x_n x_n + x_n x_n^2$ —(*)

Derivation

We first need to note that
=> The training Error function depending on weights and bias is given by

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - O_d)^2 \quad \text{——} (1)$$

where t_d => is our training value
O_d => is output function value

=> Note that in our given output function with bias and weight given as in (*), It is represented as some linear unit input and some polynomial function which is not completely linear but only linear in weights but non-linear in input vector values x_i

=> Now we need to differentiate the ~~error~~ training error function (1) with respect to each individual weights

unit with output O
where $O = w_0 + w_1 x_1 + x_2 x_2 + x_1 x_1^2 + x_2 x_2^2 + \dots + x_n x_n + x_n x_n^2$ —(*)

Derivation

We first need to note that
=> The training Error function depending on weights and bias is given by

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - O_d)^2 \quad \text{——} (1)$$

where t_d => is our training value
O_d => is output function value

=> Note that in our given output function with bias and weight given as in (*), It is represented as some linear unit input and some polynomial function which is not completely linear but only linear in weights but non-linear in input vector values x_i

=> Now we need to differentiate the ~~error~~ training error function (1) with respect to each individual weights

$$\Rightarrow \frac{\partial E}{\partial w_i} = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \frac{\partial E}{\partial w_3}, \dots \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial w_{n+1}} \right] \quad \text{—} (2)$$

=> Differentiating eqn (1) we have,

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{d \in D} 2 [t_d - O_d] \frac{\partial}{\partial w_i} (t_d - O_d) \quad \text{——} (3)$$

we first need to note that

⇒ The training error function depending on weights and bias is given by

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - O_d)^2 \qquad \text{——— (1)}$$

where

t_d ⇒ is our training value

O_d ⇒ is output function value

⇒ Note that in our given output function with bias and weights given as in (*), It is represented as some linear unit input and some polynomial function which is not completely linear but only linear in weights but non-linear in input vector values, x_i

⇒ Now we need to differentiate the ~~error~~ training error function (1) with respect to each individual weights

$$\Rightarrow \frac{\partial E}{\partial w_i} = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \frac{\partial E}{\partial w_3}, \cdots \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial w_{n+1}} \right] \quad \text{— (2)}$$

⇒ Differentiating eqn (1) we have

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{d \in D} 2[t_d - O_d] \frac{\partial}{\partial w_i} (t_d - O_d) \qquad \text{——— (3)}$$

⇒ Since everything is a vector representation, whether it is weights or inputs, so we can basically represent (3) as

$$\Delta \vec{w} = (t_d - O_d) \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - O_d) \qquad \text{——— (4)}$$

where

t_d ⇒ is our training value

O_d ⇒ is output function value

⇒ Note that in our given output function with bias and weights given as in (*), It is represented as some linear unit input and some polynomial function which is not completely linear but only linear in weights but non-linear in input vector values, x_i

⇒ Now we need to differentiate the ~~error~~ training error function (1) with respect to each individual weights

$$\Rightarrow \frac{\partial E}{\partial w_i} = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \frac{\partial E}{\partial w_3}, \cdots \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial w_{n+1}} \right] \quad \text{— (2)}$$

⇒ Differentiating eqn (1) we have

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{d \in D} 2[t_d - O_d] \frac{\partial}{\partial w_i} (t_d - O_d) \qquad \text{——— (3)}$$

⇒ Since everything is a vector representation, whether it is weights or inputs, so we can basically represent (3) as

$$\Delta \vec{w} = (t_d - O_d) \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - O_d) \qquad \text{——— (4)}$$

⇒ Question is what is O_d is our case according the given output function (*)

⇒ We need to write a general representation of our output function as below

$$O_d = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \Rightarrow O_d(\vec{x}) = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \quad \text{— } (5)$$

where $\vec{w}$ is the weight vector that is general for all weights regardless of whether it is $w_0, w_1, w_2, \ldots$ so on

$\vec{X}_d$ is the input vector at each instance

Substituting (5) into (4) we have

$$= \sum_{d \in D} [t_d - O_d] \frac{\partial}{\partial w_i} \left[t_d - (\vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2)\right] \quad \text{— } (6)$$

Now, differentiating eqn (6) we have

$$\sum_{d \in D} [t_d - O_d][0 - \vec{X}_{i,d} - \vec{X}_{i,d}^2]$$

in x_i we have

⇒ Question is what is O_d is our case according the given output function (*)

⇒ We need to write a general representation of our output function as below

$$O_d = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \Rightarrow O_d(\vec{x}) = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \quad \text{— } (5)$$

where $\vec{w}$ is the weight vector that is general for all weights regardless of whether it is $w_0, w_1, w_2, \ldots$ so on

$\vec{X}_d$ is the input vector at each instance

⇒ Substituting (5) into (4) we have

$$= \sum_{d \in D} [t_d - O_d] \frac{\partial}{\partial w_i} \left[t_d - (\vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2)\right] \quad \text{— } (6)$$

Now, differentiating eqn (6) we have

$$\Rightarrow \sum_{d \in D} [t_d - O_d][0 - \vec{X}_{i,d} - \vec{X}_{i,d}^2]$$

Factoring out negative in x_i we have

$$\Rightarrow \sum [t_d - O_d] - (\vec{X}_{i,d} + \vec{X}_{i,d}^2)$$

=> Question is what is O_d is our case according the given output function (*)

=> We need to write a general representation of our output function as below

$$O_d = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \Rightarrow O_d(\vec{x}) = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \quad \text{— (5)}$$

where $\vec{w}$ is the weight vector that is general for all weights regardless of whether it is $w_0, w_1, w_2, \ldots$ so on

$\vec{X}_d$ is the input vector at each instance

=> Substituting (5) into (4) we have

$$= \sum_{d \in D} [t_d - O_d] \frac{\partial}{\partial w_i} \left[t_d - (\vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2)\right] \quad \text{—— (6)}$$

=> Now, differentiating eqn (6) we have

$$\Rightarrow \sum_{d \in D} [t_d - O_d][0 - \vec{X}_{i,d} - \vec{X}_{i,d}^2]$$

=> Factoring out negative in x_i we have

$$\Rightarrow \sum_{d \in D} [t_d - O_d] - (\vec{X}_{i,d} + \vec{X}_{i,d}^2)$$

Error function $\Rightarrow -\sum_{d \in D} [t_d - O_d][\vec{X}_{i,d} + \vec{X}_{i,d}^2] = \nabla E(\vec{w})$

output function (*)

=> We need to write a general representation of our output function as below

$$O_d = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \Rightarrow O_d(\vec{x}) = \vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2 \quad \text{— (5)}$$

where $\vec{w}$ is the weight vector that is general for all weights regardless of whether it is $w_0, w_1, w_2, \ldots$ so on

$\vec{X}_d$ is the input vector at each instance

=> Substituting (5) into (4) we have

$$= \sum_{d \in D} [t_d - O_d] \frac{\partial}{\partial w_i} \left[t_d - (\vec{w}\vec{X}_d + \vec{w}\vec{X}_d^2)\right] \quad \text{—— (6)}$$

=> Now, differentiating eqn (6) we have

$$\Rightarrow \sum_{d \in D} [t_d - O_d][0 - \vec{X}_{i,d} - \vec{X}_{i,d}^2]$$

=> Factoring out negative in x_i we have

$$\Rightarrow \sum_{d \in D} [t_d - O_d] - (\vec{X}_{i,d} + \vec{X}_{i,d}^2)$$

Error function $\Rightarrow -\sum_{d \in D} [t_d - O_d][\vec{X}_{i,d} + \vec{X}_{i,d}^2] = \nabla E(\vec{w})$

- let's have a learning rate given by η

⇒ $\Delta \vec{w} = -\eta \nabla E(\vec{w})$

⇒ This can be represented in other notation as

$$\Delta w_i = \eta \frac{\partial E}{\partial w_i}$$

⇒ Thus the training rule for gradient descent is

$$\vec{w}_i \leftarrow \vec{w}_i + \Delta w_i$$

⇒ Therefore the modified weight equation is given by

$$w_i \leftarrow w_i - \eta \sum_{d \in D} \left[t_d - O_d\right]\left[\vec{X}_{i,d} + \vec{X}_{i,d}^{2}\right]$$

⇒ $\Delta \vec{w} = -\eta \nabla E(\vec{w})$

⇒ This can be represented in other notation as

$$\Delta w_i = \eta \frac{\partial E}{\partial w_i}$$

⇒ Thus the training rule for gradient descent is

$$\vec{w}_i \leftarrow \vec{w}_i + \Delta w_i$$

⇒ Therefore the modified weight equation is given by

$$w_i \leftarrow w_i - \eta \sum_{d \in D} \left[t_d - O_d\right]\left[\vec{X}_{i,d} + \vec{X}_{i,d}^{2}\right]$$