Name of Experiment............ Date............

Experiment No............ Experiment Result............



Name: Yashwanth. M

Regno: 18GAEC9077

Class: VII Sem CSE

## Assignment II

1.

a) Write gradient descent Algorithm to train a linear unit along with the derivation of gradient descent rule.

The direction of steepest can be found by computing the derivative of E with respect to each component of the $\vec{w}$. The vector derivative is called the gradient of E with respect to $\vec{w}$, written as

$$\nabla E[\vec{w}] = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots \frac{\partial E}{\partial w_n}\right]$$

The gradient specifies the direction of steepest increase of E, the training rule for gradient descent is,

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

where

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

→ $\eta$ is positive learning rate.

This training rule can also be written in its component form

$$\omega_i \leftarrow \omega_i + \Delta\omega_i$$

where,

$$\Delta\omega_i = -\eta \frac{\partial E}{\partial \omega_i} \longrightarrow ①$$

Calulate the gradient at each step. The vector of $\frac{\partial E}{\partial w_i}$ derivatives that form the gradient can be obtained by differentiating $E$ from $E(\vec{w}) \equiv \frac{1}{2}\sum_{d \in D}(t_d - O_d)^2$

$$\frac{\partial E}{\partial \omega_i} = \frac{\partial}{\partial \omega_i} \frac{1}{2} \sum_d (t_d - O_d)^2$$

$$= \frac{1}{2} \sum_d \frac{\partial}{\partial w_i} (t_d - O_d)^2$$

$$= \frac{1}{2} \sum_d 2(t_d - O_d) \frac{\partial}{\partial w_i} (t_d - O_d)$$

$$= \sum_d (t_d - O_d) \frac{\partial}{\partial w_i} (t_d - \vec{w} \cdot \vec{x}_d)$$

$$\frac{\partial E}{\partial w_i} = \sum_d (t_d - O_d)(-x_{i,d}) \longrightarrow ②$$

Substituting ② in ① yields the weight update rule for gradient descent

$$\Delta\omega_i = \eta \sum_{d \in D} (t_d - O_d) x_{id}$$

## Algorithm

Each training example is a pair of the form $(\vec{x}, t)$, where $\vec{x}$ is the vector of input values, and $t$ is the target output value, $\eta$ is the learning rate

* Initialize each $w_i$ to some small random value

* Until the termination condition is met, Do

→ Initialize each $\Delta w_i$ to zero
→ For each $(\vec{x}, t)$ in training examples, Do
→ Input the instance $\vec{x}$ to the unit and compute the output $o$
→ For each linear unit weight $w_i$, Do
$\Delta w_i \leftarrow \Delta w_i + \eta(t-o)x_i$
→ → → For each linear unit weight $w_i$. Do
$w_i \leftarrow w_i + \Delta w_i$

b) Write derivation of Backpropagation rule considering unit $j$ as output unit and unit $j$ as hidden unit.

For each training example $d$ every weight $w_{ji}$ is updated by adding to it $\Delta w_{ji}$

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} \longrightarrow (1)$$

where, $E_d$ is the error on training example $d$,

$$E_d(\vec{w}) \equiv \frac{1}{2} \sum_{k \in output} (t_k - O_k)^2$$

$t_k$ is target value of unit $k$
$O_k$ is output value of unit $k$

$x_{ji}$ = the $i^{th}$ input to unit $j$
$w_{ji}$ = the weight associated with the $i^{th}$ input to unit $j$
$net_j = \Sigma_i w_{ji} x_{ji}$
$O_j$ = output
$t_j$ = target
$\sigma$ = sigmoid $f^n$
Downstream $(j)$ = set of units whose immediat inputs include the output of unit $j$

derive an expression for $\frac{\partial E_d}{\partial w_{ji}}$

Using chain rule,

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

$$= \frac{\partial E_d}{\partial net_j} x_{ji} \longrightarrow (2)$$

Derive a convenient expression for $\frac{\partial E_d}{\partial net_j}$

Name of Experiment ............ Date ............ Page No. 3

Experiment No. ............ Experiment Result ............

Case 1:- Training rule for O/p unit weights.

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial net_j} \longrightarrow (3)$$

First term in (3),
$$\frac{\partial E}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$$

The derivative of $\frac{\partial}{\partial o_j}(t_k - o_k)^2$ is 0 for k except $k=j$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} (t_j - o_j)^2$$

$$= \frac{1}{2} 2(t_j - o_j) \frac{\partial (t_j - o_j)}{\partial o_j}$$

$$= -(t_j - o_j) \longrightarrow (4)$$

Second term in (3), Since $o_j = \sigma(net_j)$
$$\frac{\partial o_j}{\partial net_j} = \sigma(net_j)(1 - \sigma(net_j))$$

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial \sigma(net_j)}{\partial net_j}$$

$$= o_j (1 - o_j) \longrightarrow (5)$$

(4) & (5) in (3)
$$\frac{\partial E_d}{\partial net_j} = -(t_j - o_j)\, o_j (1 - o_j)$$

and combining with ① and ②

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = \eta (t_j - o_j)\, o_j (1-o_j)(x_{ji}) \rightarrow ⑦$$

Case 2:- Training rule for Hidden Unit Weights.

- → In this case, $j$ is hidden unit
- → the derivation of training rule $w_{ji}$ must take into account the indirect ways in which $w_{ji}$ can influence the network outputs.
- → we will find it useful to refer to the set of all units immediately downstream of unit $j$
- → $net_j$ can influence the network outputs only through the units in Downstream($j$)

$$\frac{\partial E_d}{\partial net_j} = \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j}$$

$$= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial net_j}$$

$$= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j}$$

$$= \sum_{k \in Downstream(j)} -\delta_k\, w_{kj} \frac{\partial o_j}{\partial net_j}$$

$$= \sum_{k \in Downstream(j)} -\delta_k\, w_{kj}\, o_j (1-o_j) \rightarrow ⑧$$

$$\delta_j = o_j(1-o_j) \sum_{k \in Downstream(j)} \delta_k w_{kj}$$

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

2

a) Derive derivation to show maximum likelyhood hypothesis has least squared error hypotheses.

Consider the problem of learning a continuous valued target function such as neural network learning, linear regression, and polynomial curve fitting

Using the definition of $h_{ML}$ we have

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ p(D|h)$$

Assuming training examples are mutually independent given h, we can write P(D|h) as the product of the various $p(d_i|h)$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} p(d_i|h)$$

Given the noise $e_i$ obeys a normal distribution with 0 mean and unknown variance $\sigma^2$, each $d_i$ must also obey a Normal distribution around the true target value $f(x_i)$.

Because we are writing the expression for $P(D|h)$, we assume $h$ is the correct description of $f$.

Hence $\mu = f(x_i) = h(x_i)$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - \mu)^2}$$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(x_i))^2}$$

Maximise the less complicated logarithm, which is justified because of the monotonicity of function p

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

The first term in this expression is a constant independent of $h$, and can therefore be discarded

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

Maximizing this negative quantity is equivalent to minimize +ve – '' –

$$h_{ML} = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

discard constant

$$h_{ML} = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Thus, above equation shows that the maximum likelihood hypothesis $H_{ML}$ is the one that minimises the sum of the squared errors between the observed training rules $d_i$ and hypotheses predictions $h(x_i)$

b. Classify the test data and {Red, SUV, Domestic} using Naive Bayes classifier for the data show in Table

| Color | Type | Origin | Stolen |
|---|---|---|---|
| Red | Sports | Domestic | Yes |
| Red | Sports | Domestic | No |
| Red | Sports | Domestic | Yes |
| Yellow | Sports | Domestic | No. |
| Yellow | Sports | Imported | Yes |
| Yellow | SUV | Imported | No |
| Yellow | SUV | Imported | Yes |
| Yellow | SUV | Domestic | No |
| Red | SUV | Imported | No |
| Red | Sports | Imported | Yes |

$$V_{nb} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod P(a_i|v_j) \rightarrow (1)$$

$$P(a_i|v_j) = \frac{n_c + mp}{n+m} \rightarrow (2)$$

$n$ = no. of training examples for which $v = v_j$
$n_c$ = no. of examples for $v = v_j$ & $a = a_i$
$p$ = a priori estimate for $P(a_i|v_j)$
$m$ = the equivalent sample size

We want to classify { Red, Domestic, SUV }

We need to calculate

P(Red | Yes) , P(SUV | Yes ), P( Domestic | Yes)

P(Red | No) , P(SUV | No) , and P(Domestic | No)

and multiply them by P(Yes ) and P(No) respectively.

~~Yes:~~

~~Red :~~

~~$n = .5$~~

~~$n_c = 3$~~

~~$p = 0.5$~~

~~$m = 3$~~

~~SUV:~~

~~$n = 5$~~

~~$n_c = 1.$~~

Stolen

| | | P(Yes) | P(No) |
|---|---|---|---|
| color | Red | 3/5 | 2/5 |
| | Yellow | 2/5 | 3/5 |

| | | | |
|---|---|---|---|
| Type | Sports | 4/5 | 2/5 |
| | SUV | 1/4 | 3/4 |

Stolen

| Origin | | P(Yes) | P(No) |
|---|---|---|---|
| | Domestic | 2/5 | 3/5 |
| | Imported | 3/5 | 2/5 |

$P(Yes \mid X) = P(Red \mid Yes) * P(SUV \mid Yes) * P(Domestic \mid Yes) * P(Yes)$

$= \frac{3}{5} * \frac{1}{4} * \frac{2}{5} \times \frac{1}{2}$

$= 0.03$

$P(No \mid X) = P(Red \mid No) * P(SUV \mid No) * P(Domestic \mid No) * P(No)$

$= \frac{2}{5} * \frac{3}{4} * \frac{3}{5} * \frac{1}{5}$

$= 0.036$

Since $0.036 > 0.03$, it is classified as 'No'.

3) a) Define

i) Sample Error

ii) True Error

iii) Confidence intervals for discrete valued hypothesis.

Sample Error:

The Sample Error of a hypothesis with respect to some sample S of instances drawn from X is the fraction of S that it misclassifies

Definition: The Sample Error ($error_S(h)$) of hypothesis h with respect to target function f and data sample S is

$$error_S(h) \equiv \frac{1}{n} \sum_{x \in S} \delta(f(x), h(x))$$

where n is the no. of examples in S, and the quantity $\delta(f(x), h(x))$ is 1 if $f(x) \neq h(x)$ and 0 otherwise

True Error

The true error ($error_D(h)$) of hypothesis h with respect to target function f and distribution D, is the probability that h will misclassify an instance drawn at random according to D

$$error_D(h) \equiv \Pr_{x \in D} [f(x) \neq h(x)]$$

Confidence Intervals for Discrete - Valued Hypothesis

Suppose we wish to estimate the true error for some discrete valued hypothesis h, based on its observed sample error over a sample S, where

→ The sample S contains n examples drawn independent of one another, and independent of h, according to the probability distribution D

→ $n \geq 30$

→ Hypothesis h commits r errors over these n examples (i.e $error_S(h) = r/n$)

Under these conditions, statistical theory allows to make the following assertions.

1. Given no other info, the most probable value of $error_D(h)$ is $error_S(h)$.

2. With approx 95% probability, the true $error_D(h)$ lies in the interval

$$error_S(h) \pm 1.96 \sqrt{\frac{error_S(h)\,(1-error_S(h))}{n}}$$

b) Last year, five randomly selected students took a math aptitude test before they began their statistics course. The Statistics Department has three questions.

i) What linear regression equation best predicts statistics performance, based their aptitude scores?

ii) if a student made on 80 on the aptitude test, what grade would we expect her to make in statistics ?

iii) How well does the regression equation fit the data ?

| Student | $x_i$ | $y_i$ |
|---|---|---|
| 1 | 95 | 85 |
| 2 | 85 | 95 |
| 3 | 80 | 70 |
| 4 | 70 | 65 |
| 5 | 60 | 70 |

i) $$\hat{y} = b_0 + b_1 x_i$$

$$b_0 = \frac{(\Sigma y)(\Sigma x^2) - \Sigma(x)\Sigma xy}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b_1 = \frac{n(\Sigma xy) - \Sigma x \Sigma y}{n(\Sigma x^2) - (\Sigma x)^2}$$

| $x$ | $y$ | $x^2$ | $xy$ |
|---|---|---|---|
| 95 | 85 | 9025 | 8075 |
| 85 | 95 | 7225 | 8075 |
| 80 | 70 | 6400 | 5600 |
| 70 | 65 | 4900 | 4550 |
| 60 | 70 | 3600 | 4200 |
| 390 | 385 | 31150 | 30500 |

Name of Experiment.......................................... Date..............................................................

Experiment No.......................................... Experiment Result.......................................... Page No. 8

$$b_0 = \frac{385 \times 31150}{3650} = 26.780$$

$$b_1 = \frac{2350}{3656} = 0.64383$$

$$\therefore \hat{y} = 26.780 + x \times 0.64383$$

ii) $x = 80$

$$\hat{y} = 26.780 + 80 \times 0.64383$$

$$\hat{y} = 78.28$$

iii)

$$E = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

$$E = 83.78$$