Name - Md Fanhan Istmam
180041120

Name : Md Fanhan Istmam
ID : 180041120
Semesten : Sixth
Date : 07-Feb-22
Course Code : CSE 4621
Course Name : Machine Learning
Exam : Mid Semesten

Ans. to Q.no. 3(a)

$$\frac{dL}{dz^{[1]}} = \frac{dL}{da^{[2]}} \times \frac{da^{[2]}}{dz^{[2]}} \times \frac{dz^{[2]}}{da^{[1]}} \times \frac{da^{[1]}}{dz^{[1]}} \quad [\text{Chain-Rule}]$$

Now, $L = -\hat{y} \log \hat{y} - (1-y) \log (1-\hat{y}) \quad [\hat{y} = a^{[2]}]$

$$\frac{dL}{da^{[2]}} = \frac{d}{da^{[2]}} \left[-y \log a^{[2]} - (1-y) \log (1-a^{[2]}) \right]$$

$$= -\frac{y}{a^{[2]}} + \frac{1-y}{1-a^{[2]}} \quad \text{---------- (i)}$$

$$\frac{da^{[2]}}{dz^{[2]}} = \frac{d}{dz^{[2]}} (\sigma(z^{[2]}))$$

$$= \sigma(z^{[2]}) \{1 - \sigma(z^{[2]})\}$$

$$= a^{[2]} (1 - a^{[2]}) \quad \text{---------- (ii)}$$

$$\frac{dz^{[2]}}{da^{[1]}} = \frac{d}{da^{[1]}} (w^{[2]} a^{[1]} + b^{[2]})$$

$$= w^{[2]} \quad \text{---------- (iii)}$$

$$\frac{da^{[1]}}{dz^{[1]}} = \frac{d}{dz^{[1]}} \sigma(z^{[1]})$$

$$= a^{[1]} (1 - a^{[1]}) \quad \text{---------- (iv)}$$

180041120
Ashna

Multiplying (i), (ii), (iii) and (iv),

$$\frac{dL}{dz^{[1]}} = \left(-\frac{y}{a^{[2]}} + \frac{1-y}{1-a^{[2]}}\right)\left\{a^{[2]}(1-a^{[2]})\right\} \cdot w^{[2]} \cdot a^{[1]}(1-a^{[1]})$$

$$= (-y + a^{[2]}) \cdot w^{[2]} \cdot a^{[1]}(1-a^{[1]})$$

$$= w^{[2]} \cdot a^{[1]}(1-a^{[1]})(a^{[2]} - y) \quad \text{(Ans.)}$$

This can be rewritten as

$$dz^{[1]} = dz^{[2]} \cdot \sigma'(a^{[1]}) \cdot w^{[2]}$$

The update equation for $w^{[1]}$ is

$$\frac{dL}{dw^{[1]}} = \frac{dL}{dz^{[1]}} \times \frac{dz^{[1]}}{dw^{[1]}}$$

Now, $\frac{dz^{[1]}}{dw^{[1]}} = \frac{d}{dw^{[1]}}\left(w^{[1]}x + b^{[1]}\right)$

$$= x$$

$$\therefore \frac{dL}{dw^{[1]}} = w^{[2]} \cdot a^{[1]}(1-a^{[1]})(a^{[2]} - y) \cdot x \quad \text{(Ans.)}$$

$$= dz^{[1]} \cdot x$$

Update equation.

$$w^{[1]} := w^{[1]} - \alpha\, dw^{[1]}$$

or, $w^{[1]} := w^{[1]} - \alpha(dz^{[1]} \cdot x)$

or, $w^{[1]} := w^{[1]} - \alpha\left\{w^{[2]} a^{[1]} x (1-a^{[1]})(a^{[2]} - y)\right\}$

(Ans.)

180041120

Ans to Q.no. 3(b)

Given, $x = (1, 1) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$; $y = 0$.

(i) $a_1^{[1]} = \sigma(W_1^{[1]} x + b_1^{[1]})$

$= \sigma(0.8 - 0.5 \times 1 + 0.4 \times 1)$

$= \sigma(0.7)$

$= 0.332$

$a_2^{[1]} = \sigma(W_2^{[1]} x + b_2^{[1]})$

$= \sigma(0.9 \times 1 + 1.0 \times 1 - 0.1)$

$= \sigma(1.8)$

$= 0.858$

$\hat{y} = a^{[2]}$ ~~$= \sigma(W_0^{[2]} a^{[0]} + b^{[2]})$~~

$= \sigma(W^{[2]} a^{[2]} + b^{[2]})$

$= \sigma(0.3 + 0.332 \times (-1.2) + 1.1 \times 0.858)$

$= \sigma(0.8454)$

$= 0.7$

$\therefore \hat{y} = 0.7$.

180041120

(ii) $dw^{[2]} = \frac{dL}{da^{[2]}} \times \frac{da^{[2]}}{dz^{[2]}} \times \frac{dz^{[2]}}{dw^{[2]}}$

$= (a^{[2]} - y) \times a^{[1]}$

$db^{[2]} = a^{[2]} - y$

$w^{[2]} = w^{[2]} - \alpha\, dw^{[2]}$

$b^{[2]} = b^{[2]} - \alpha\, db^{[2]}$ [assume, $\alpha = 1$]

$w^{[2]} = \begin{bmatrix} -1.2 \\ 1.1 \end{bmatrix} - (0.7 - 0) \times \begin{bmatrix} 0.332 \\ 0.858 \end{bmatrix}$

$= \begin{bmatrix} -1.2 \\ 1.1 \end{bmatrix} - \begin{bmatrix} 0.2324 \\ 0.6 \end{bmatrix}$

$= \begin{bmatrix} -1.432 \\ 0.5 \end{bmatrix}$ (Ans)

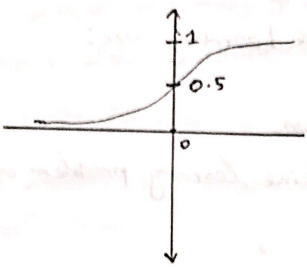
(iii) $b^{[2]} = b^{[2]} - \alpha \times db^{[2]}$

$= 0.3 - 1 \times (0.7 - 0)$

$= 0.4$ (Ans.)

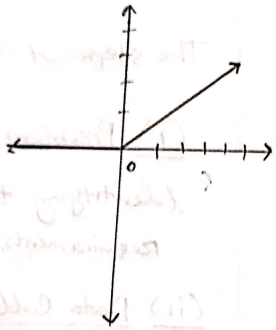
180041120
Ishmam

Ans to Q.no. 3(e)



Sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}} \in (0,1)$$



ReLu

$$ReLu(z) = \max(0, z)$$

Benefits of ReLu over sigmoid:

(i) Sigmoid punishes negative values by taking it close to zero. It serves the purpose while a ReLu assigns exactly 0 to negative costs, which speeds up training.

(ii) ReLu is simpler and faster. Easier to train as it is less computationally expensive.

(iii) ReLu is better for Image Processing using CNN as the negative cost of sigmoid makes it harder to train such models. They can generalize easier using ReLu.

(iv) Sigmoid is more prone to vanishing and exploding gradient problem.

180041120
Ishma

Ans to Qno. 1(a)

The steps of Machine Learning are:

(i) Problem Definition

Identifying the machine learning problem with its requirements.

(ii) Data Collection

Gathering the data required for the machine learning task through third-party sources, survey, databases and so on.

(iii) Data pre-processing

Cleaning the data, scaling and normalizing it to make it suitable for training. Data splitting is performed.

(iv) Model Selection

Selecting an appropriate model, and learning algorithm for the task. This includes – linear regression, neural network, support vector machine etc.

(v) Training the Model

Using whole data or portion of it, the model is trained by minimizing the cost through iterations.

(vi) Hyperparameter Tuning

The learning rate, regularization parameter, network architecture are changed and fine-tuned to improve the results in cross-validation dataset.

180041120
Ishwar

(vii) Evaluating the Model

The model accuracy is measured on Test set. This step is done concurrently with hyperparameter tuning.

(viii) Model deployment

The model is deployed and can be used for practical applications.

Major types of learning —

(i) Supervised — Training is performed with labeled dataset. Ex - linear, logistic regression, neural networks.

(ii) Unsupervised — Trained using unlabeled dataset.
Ex - Principal Component Analysis (PCA)
Anomality Detection.

(iii) Semi-supervised — Dataset contains both labeled and unlabeled data. Trained using labeled data, fine-tuned / refined using unlabeled data.

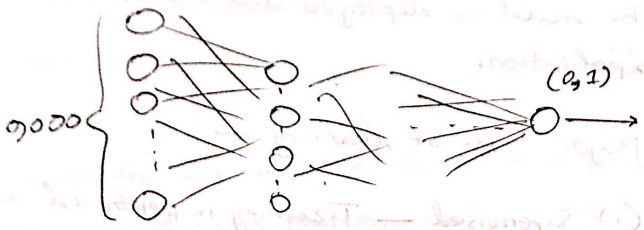
(iv) Reinforcement Learning — Reward / punishment based learning. If model does wrong it gets punished and tries to do something new.

Ex - AlphaZero (deep N.N. to play chess).

180041120
Ishan

Ans to Qno 1(b)

As the number of input features is huge, 10,000 ~~9000~~, I believe using a deep neural network will be the best choice.



Deep N.N.

This model is chosen because 10,000 ~~9000~~ inputs is a huge feature set and at least 9,000 will be used. Deep Neural networks work better on datasets with large input dimensions. A 10-12 layered neural network with a sufficient number of neurons in each layer should give good accuracy.

Regularization –

I will use Dropout Regularization, as there are 10,000 genes, some genes like 9000 will be used. So, rest 1000 can be removed using dropout regularization.

1800411120

Ashmin

Ans to Q no. 1(c)

Log-loss cost

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)}) \log (1-\hat{y}^{(i)})\right]$$

~~$\nabla_\theta \mathcal{L} = \frac{1}{m}$~~

The vectorized form is

$$\mathcal{L} = \frac{-1}{m}\left(y \log \hat{y} + (1-y)\log(1-\hat{y})\right)$$

$$= -\frac{1}{m}\left[y \log \sigma(\theta x) + (1-y)\log(1-\sigma(\theta x))\right]$$

~~$\nabla_\theta \mathcal{L} = -\frac{1}{m}\frac{\partial}{\partial \theta}$~~

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial \theta_z} \times \frac{\partial z}{\partial \theta} \quad \left[\because \sigma'(z) = \sigma(z)(1-\sigma(z))\right]$$

$$= \left(-\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right) \times \left\{\hat{y}(1-\hat{y})\right\} \times \ldots \times$$

$$= (\hat{y} - y) \cdot x$$

$$\nabla_\theta^2 \mathcal{L} = \frac{\partial^2 \mathcal{L}}{\partial \theta^2} = \left(\frac{\partial}{\partial \theta}\hat{y}\right) \cdot x$$

$$= \hat{y}(1-\hat{y}) \cdot x^T x$$

This term is always positive as $\hat{y} \geq 0$ for $\hat{y} = \sigma(\theta x)$

$\therefore \nabla_\theta^2 \mathcal{L} \geq 0$

This is the second derivative test for convexity.

$\therefore$ Log loss function is convex (proved.)

1S0041126

Ans.to Q.no.2(a)

Suppose, we have $x_1 \in [0, 10000]$

$x_2 \in [0, 5]$.

When using linear or logistic regression, the model will be more dependent on x_1 as it has higher values. For similar weights model will prefer x_1 and will try to optimize it first. This prevents the model from generalizing and makes it harder to train.

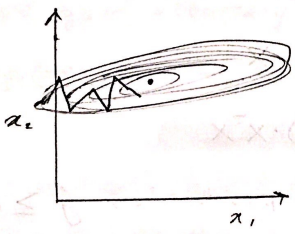
Instead if $x_1 \in [0,1]$

$x_2 \in [0,1]$

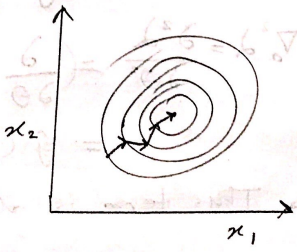
each feature will be given similar priority.

Using feature scaling, $x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$

we can keep the features in a small range [0,1].



Unscaled Dataset Features.



Scaled Features.

Scaled features make it easier to reach the minimum,

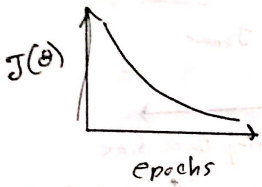
180041120

Ans to Q no 2(b)

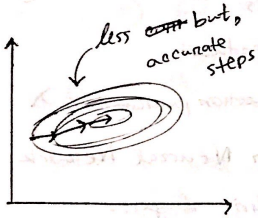
Batch

(i) Whole training data is used in a single epoch.

(ii) Cost decreases in each step.



(iii) Each step takes longer to train but accurate.

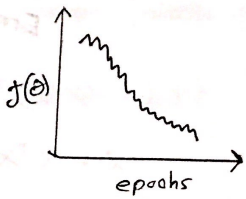


(iv) Usually takes longer time to train.

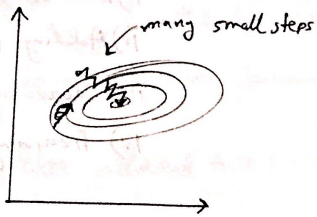
Stochastic

(i) A single training data sample is used.

(ii) Cost might increase, but in multiple steps overall cost decreases.



(iii) Each step takes shorten time to train but can be inaccurate



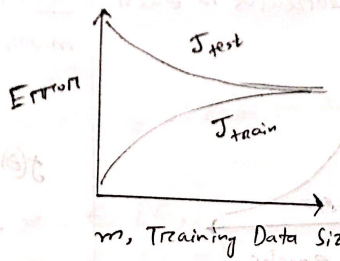
(iv) Usually faster to train.

180041120

Ans. to Q. no. 2(c)

If the training and test error — both are high, then the model is suffering from high bias problem.

No, ~~the~~ increasing the ~~training~~ training data size won't help.



As, we can see increasing data size won't help reduce the error.

Solution →

i) Increasing features
ii) Adding polynomial features
iii) Decreasing regularization parameter, λ
iv) Training a deeper Neural Network
   - → Adding layers
   - → Increases no. of neurons / layer.

18004112O

Ans. to Q no. 2(d)

Given, $J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{n} |\theta_j|$

~~$\nabla_\theta J = \frac{\partial J}{\partial \theta} = \frac{1}{2} \cdot 2$~~

The vector form of the given equation is

$$J(\theta) = \frac{1}{2} (X\theta - Y)^2 + \lambda \Sigma |\theta_j|$$

$$\nabla_\theta J = \frac{\partial J}{\partial \theta} = \frac{1}{2} \times 2 (X\theta - Y) X + \lambda$$

$$\Rightarrow 0 = X^T (X\theta - Y) + \lambda \quad [\because \nabla_\theta J = 0]$$

$$\Rightarrow 0 = X^T X\theta - X^T Y + \lambda$$

$$\Rightarrow X^T Y - \lambda = X^T X \theta$$

$$\Rightarrow (X^T X)^{-1} (X^T Y - \lambda) = \theta$$

$$\Rightarrow \hat{\theta} = (X^T X)^{-1} (X^T Y - \begin{bmatrix} 0 & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{bmatrix} \lambda)$$ (Ans.)

Here, the $\begin{bmatrix} 0 & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{bmatrix}$ term is given because regularization is NOT added to the bias.

We shall avoid normal equation if the number of training data is high in a dataset or the number of features is too high. Because, normal ~~data~~ equation is computationally more expensive and has complexity $O(n^3)$.