

1. The equations for ϵ_{iu} . Update equations in the Stochastic Gradient Descent algorithm

Ans:-

The equations include the following given derivative steps:-

$$\epsilon_{xi} = 2(r_{xi} - q_i \cdot p_x)$$

$$q_i \leftarrow q_i + \mu_1 (\epsilon_{xi} p_x - 2\lambda_2 q_i)$$

$$p_x \leftarrow p_x + \mu_2 (\epsilon_{xi} q_i - 2\lambda_1 p_x)$$

Handwritten derivation of the Stochastic Gradient Descent equations for matrix factorization. The derivation starts with the definition of matrices Q and P and the loss function E .

Given: $Q = m \times k$, $P = n \times k$, $k = \text{no. of factors}$

Matrix dimensions: Q (users) \times P (movies) \rightarrow R (matrix)

Loss function: $E = \left(\sum_{(i,u) \in \text{training}} (r_{iu} - q_i \cdot p_u)^2 \right) + \lambda \left[\sum_u \|p_u\|_2^2 + \sum_i \|q_i\|_2^2 \right]$

Derivative with respect to P (matrix $n \times n$):

$$\frac{dE}{dP_{(m \times n)}} = 2(r_{iu} - q_i \cdot p_u) \quad \text{--- (1)}$$

Derivative with respect to q_i :

$$\frac{dE}{dq_i} = q_i - \eta \left[(2E \cdot (-p_u^T)) + \lambda 2q_i \right]$$

$$\frac{dE}{dq_i} = q_i + \eta \left[\sum_u p_u^T - 2\lambda q_i \right] \quad \text{--- (2)}$$

In terms of iteration:

$$q_i \leftarrow q_i + \eta * (E_{iu} * p_u - 2 * \lambda * q_i) \quad \text{--- (2)}$$

Derivative with respect to p_u :

$$\frac{dE}{dp_u} = p_u + \eta \left[\sum_i \epsilon_{iu} * q_i - 2 * \lambda * p_u \right]$$

In terms of iteration:

$$p_u \leftarrow p_u + \eta * (E_{iu} * q_i - 2 * \lambda * p_u) \quad \text{--- (3)}$$

2. What was the lowest error you got? What was the value of η ?

1. For learning_rate = 0.01 # η ,
40th iteration error = 80622 #lowest

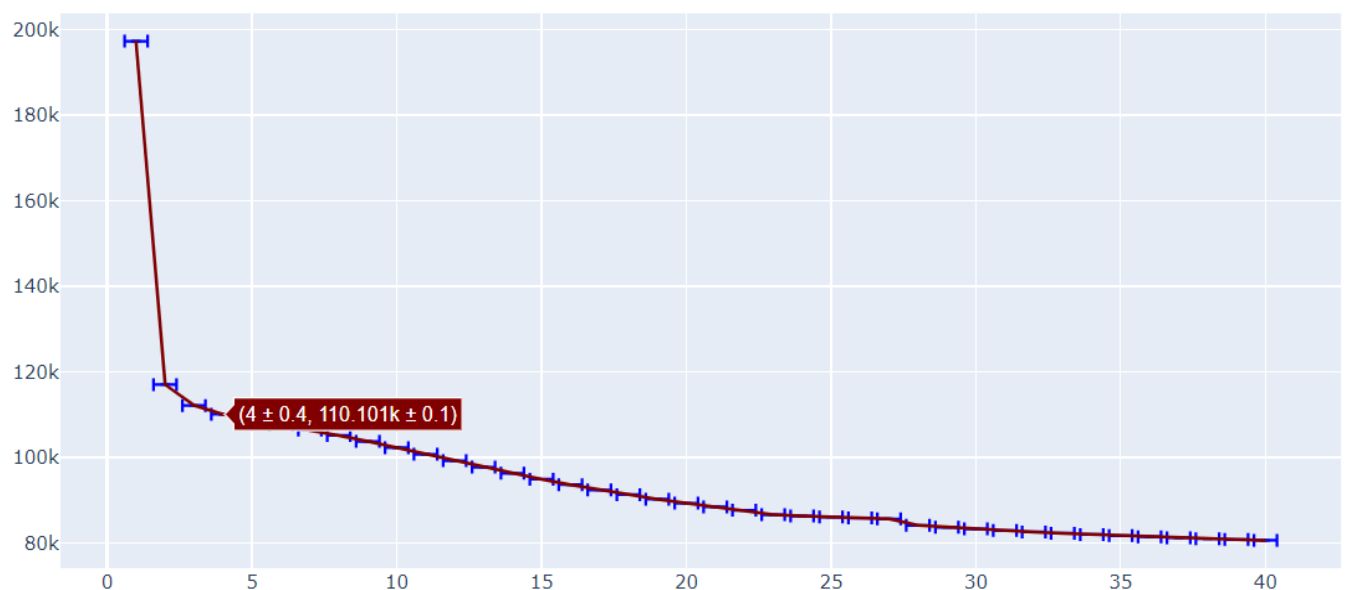
2. For learning_rate = 0.02 # η , 40th error = 84102
3. For learning_rate = 0.03 # η , 40th error = 89872
4. For learning_rate = 0.03 # η , 40th error = 120098
5. For learning_rate = 0.001 # η , 40th error = 88102

3. For the best η , plot of E vs. number of iterations. Make sure your graph has a y-axis so that we can read the value of E .

For learning_rate = 0.01 # η , 40th iteration error = 80622 #lowest

X-axis = number of Iteration and

Y = Errors after execution



Inference:

1. In 1st iteration the error = 197282.6
2. 40th iteration Error = 80622.64
3. With increasing iteration the error value is decreasing
4. **Iteration start time - 08:46:47**
40th iteration completion time:- 16:11:49
Time taken = 07:65:0

As weights are updated after each iteration within individual rows so the error optimization happens. I think tuning the hyper parameter like “learning rate is quite a challenge and time taking”