Parallel Implementation of Machine Learning Algorithms

ME766 Course Project

Team Members:

Saurabh Parekh (170100016)

Nabajyoti Majumdar (170100081)

Sai Vamseedhar Bojja (170070048)

1. Introduction

Machine learning can simply be defined as using data instead of logic to perform tasks by a machine. We use data to train the machine, as in, tell it what it has to do and then test the trained model on different tasks to see whether the training has been successful or not. A machine is said to be learning from past Experiences(input data) with respect to some class of Tasks, if its Performance in a given Task improves with the Experience.

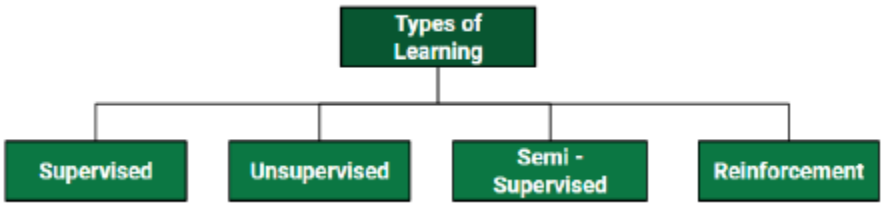

Fig. 1. Types of Machine Learning

Algorithm: Gradient Descent

Input: $Y, \Theta, \mathbf{X}, \alpha$, tolerance, max iterations

Output: $\mathbf{\Theta}$

for $i = 0; i < max\ iterations;\ i++$ **do**
  current cost $= Cost(Y, \mathbf{X}, \mathbf{\Theta})$
  if *current cost* $<$ *tolerance* **then**
    break
  else
    gradient $= Gradient(Y, \mathbf{X}, \mathbf{\Theta})$
    $\theta_j \leftarrow \theta_j - \alpha \cdot$ gradient

Fig. 2. Gradient descent algorithm pseudo code

Supervised learning is when the model is getting trained on a labelled dataset. Labelled dataset is one which has both input and output parameters. In this type of learning both training and validation datasets are labelled. Types of Supervised Learning:

- Classification : It is a Supervised Learning task where output is having defined labels(discrete value).Example: Gmail classifies mails in more than one classes like social, promotions, updates, forum.
- Regression : It is a Supervised Learning task where output is having continuous value.

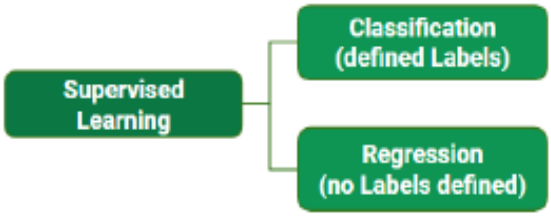

Fig. 3. Types of Supervised Learning

Regression is broadly classified into two types:

- Linear regression
- Logistic regression

A. Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent vari- ables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on the kind of relationship between dependent and independent variables they are considering and the number of independent variables being used. It finds a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. Hypothesis function for Linear Regression : y = m*x + c

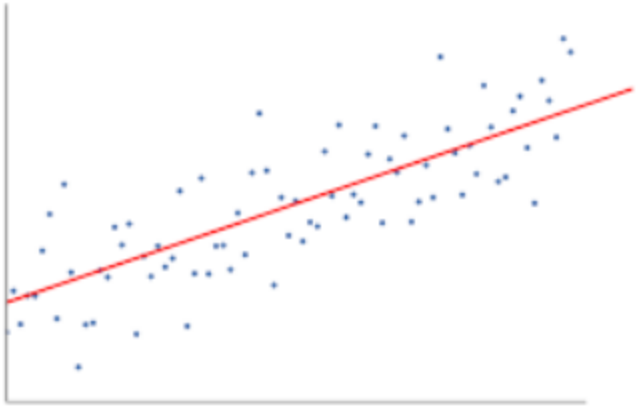

Fig. 4. Linear regression function

To update m and c values in order to reduce cost function (minimizing RMSE value) and achieve the best fit line, the model uses gradient descent. The idea is to start with random m and c values and then iteratively updating the values, reaching minimum cost.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$$

Computation of gradient descent becomes as follows,

Repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)x_{ij}$$, where x_{ij} is the j^{th} feature of i^{th} observation

(for j=0 to j=n) → (Update all θ_j s simultaneously before moving to next iteration.)

}

Fig. 5. Linear regression objective function

B. Logistic Regression

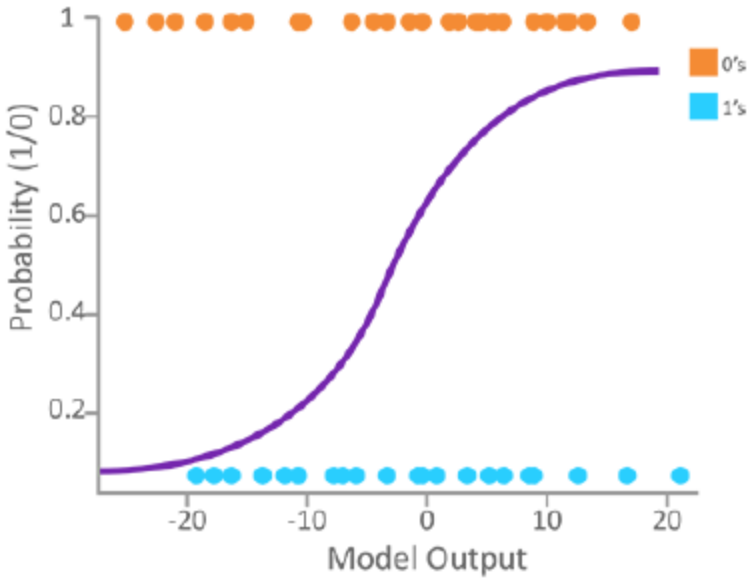

Fig. 6. Logistic Regression function

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for given set features(or inputs), x. Hypothesis function for Logistic Regression :
y = 1 / (1 + exp(−x))

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x), y)$$

$$\text{Cost}(h_\theta(\mathbf{x}), y) = -y log(h_\theta(\mathbf{x})) - (1-y) log(1-h_\theta(\mathbf{x}))$$

Gradient descent for logistic regression:

while not converged {

$$\theta_j^{\text{new}} = \theta_j^{\text{old}} - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \text{ for } j = 0, 1, \ldots, n$$

}

Fig. 7. Logistic regression objective function

2. Results

Number of threads(n)	Time taken for Linear Regression (seconds)		Time taken for Logistic Regression (seconds)	
	OpenMP	MPI	OpenMP	MPI
2	1.221	1.447	1.957	2.6713
4	0.6783	0.835	1.223	1.6554
6	0.655	0.984	1.492	1.657
8	0.899	0.878	1.328	1.789

Here, N = number of data points = 1,00,000
Time taken is averaged over 5 runs

Number of data points (N)	Time taken for Linear Regression (seconds)			Time taken for Logistic Regression (seconds)		
	Serial	OpenMP	MPI	Serial	OpenMP	MPI
100	0.0179	0.0112	0.0438	0.028	0.015	0.018
1000	0.088	0.0496	0.058	0.1656	0.092	0.102
10000	0.2608	0.1534	0.237	0.4675	0.25	0.304
50000	1.115	0.6198	0.894	1.701	0.972	1.128
100000	2.173	1.221	1.447	3.718	1.957	2.6713
500000	12.29	6.4728	8.3548	21.48	11.02	15.477

Here, n = number of threads = 2 for OpenMP and MPI codes
Time taken is average over 10 runs

3. Conclusions

- OpenMP is slightly faster than MPI as the average times taken by OpenMP are less than MPI.
- Linear Regression takes less time than logistic regression because logistic regression additionally requires computing the logistic function values.

- Also linear regression converges closer to the actual parameter values than logistic regression because the data is linearly distributed.
- Average time taken increases as we increase the number of data points since it requires more time to compute the error.
- Number of iterations required to converge increases slightly as the number of data points increases.

4. References

- https://www.geeksforgeeks.org/ml-types-learning-supervised-learning
- https://machinelearning-blog.com/2018/01/24/linear-regression/
- https://www.vertica.com/wp-content/uploads/2017/08/logistic-regression-1.png
- https://miro.medium.com/max/2752/1*oJKalifbWzwuo3fRjWJjTg.png
- https://www.crayondata.com/wp-content/uploads/2017/09/image17.png
- https://i.stack.imgur.com/zgdnk.png