Applying SIMD in Linear Regression

2024. 02. 14

Presentation by Yeojin Oh, ZhuYongjie, Boseung Kim

yeojinoh@dankook.ac.kr, aeashio1111@dankook.ac.kr, bskim1102@dankook.ac.kr

DANKOOK UNIVERSITY

Dankook University System Software Laboratory

Contents

Applying SIMD in Linear Regression

- Polynomial linear regression

$$\boxed{\hat{y} = \beta_0 x^0 + \beta_1 x^1} + \beta_2 x^2 + \cdots + \beta_m x^m$$

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^0 & \sum_{i=1}^{n} x_i^1 & \sum_{i=1}^{n} x_i^2 & \cdots & \sum_{i=1}^{n} x_i^m \\ \sum_{i=1}^{n} x_i^1 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \cdots & \sum_{i=1}^{n} x_i^{m+1} \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 & \cdots & \sum_{i=1}^{n} x_i^{m+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^m & \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m+2} & \cdots & \sum_{i=1}^{n} x_i^{2m} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdots \\ \beta_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i x_i^0 \\ \sum_{i=0}^{n} y_i x_i^1 \\ \sum_{i=0}^{n} y_i x_i^2 \\ \cdots \\ \sum_{i=0}^{n} y_i x_i^m \end{bmatrix}$$

Applying SIMD in Linear Regression

- Simple linear regression

$$\boxed{y = \hat{\alpha} + \hat{\beta} x}$$

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} y_i x_i \end{bmatrix}$$

$$\widehat{\alpha} = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$\widehat{\beta} = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

Design

- only need to get four parameters and combine them

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$\hat{\beta} = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

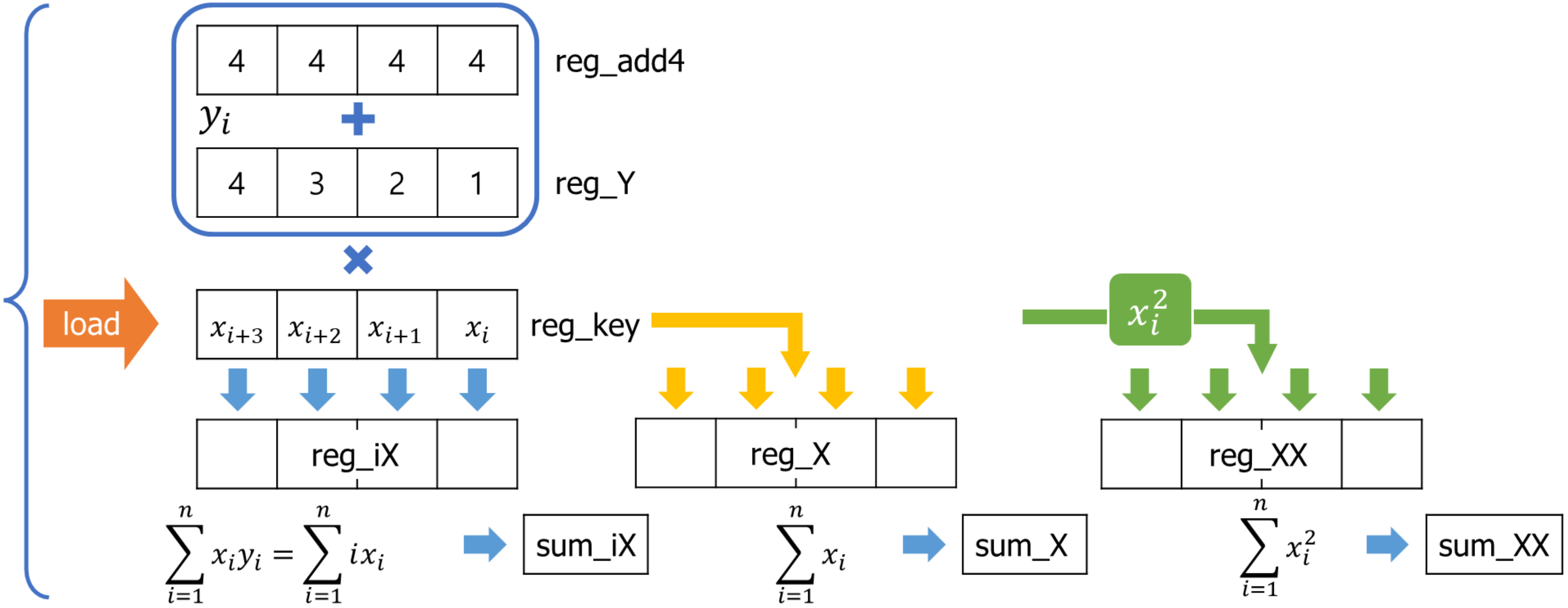
$$\sum_{i=1}^{n} y_i, \sum_{i=1}^{n} x_i^2, \sum_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i y_i$$

※ $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Design

- How can I obtain four parameters to utilize SIMD?

for (int i=0; i<SIZE, i+=4)



Model training with SIMD

- Initial setting
  - Set each register to its initial value

```
__m256d reg_add4 = _mm256_set_pd(4.0, 4.0, 4.0, 4.0);
__m256d reg_Y    = _mm256_set_pd(4.0, 3.0, 2.0, 1.0);
__m256d reg_X    = _mm256_set_pd(0.0, 0.0, 0.0, 0.0);
__m256d reg_XX   = _mm256_set_pd(0.0, 0.0, 0.0, 0.0);
__m256d reg_iX   = _mm256_set_pd(0.0, 0.0, 0.0, 0.0);
```

- Training (continued…)
  - Running parallel operations using SIMD

```
for(int i=0;i<SIZE;i+=4){
    reg_key = _mm256_loadu_pd((keys+i));

    reg_iX  = _mm256_add_pd(reg_iX, _mm256_mul_pd(reg_Y,reg_key));
    reg_X   = _mm256_add_pd(reg_key,reg_X);
    reg_XX  = _mm256_add_pd(reg_XX, _mm256_mul_pd(reg_key,reg_key));

    reg_Y   = _mm256_add_pd(reg_add4,reg_Y);
}
```

Model training with SIMD

- Training
  - Add the values in the SIMD registers to get the parameters

```
double sum_iX = std::accumulate(arr_iX, arr_iX+4, 0.0);
double sum_X  = std::accumulate(arr_X, arr_X+4, 0.0);
double sum_XX = std::accumulate(arr_XX, arr_XX+4, 0.0);
double sum_Y  = (double)SIZE*(SIZE+1)/2;
double sum_XY = sum_iX;
```

  - Calculate the slope and intercept using the parameters

$$\widehat{\alpha} = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \rightarrow$$

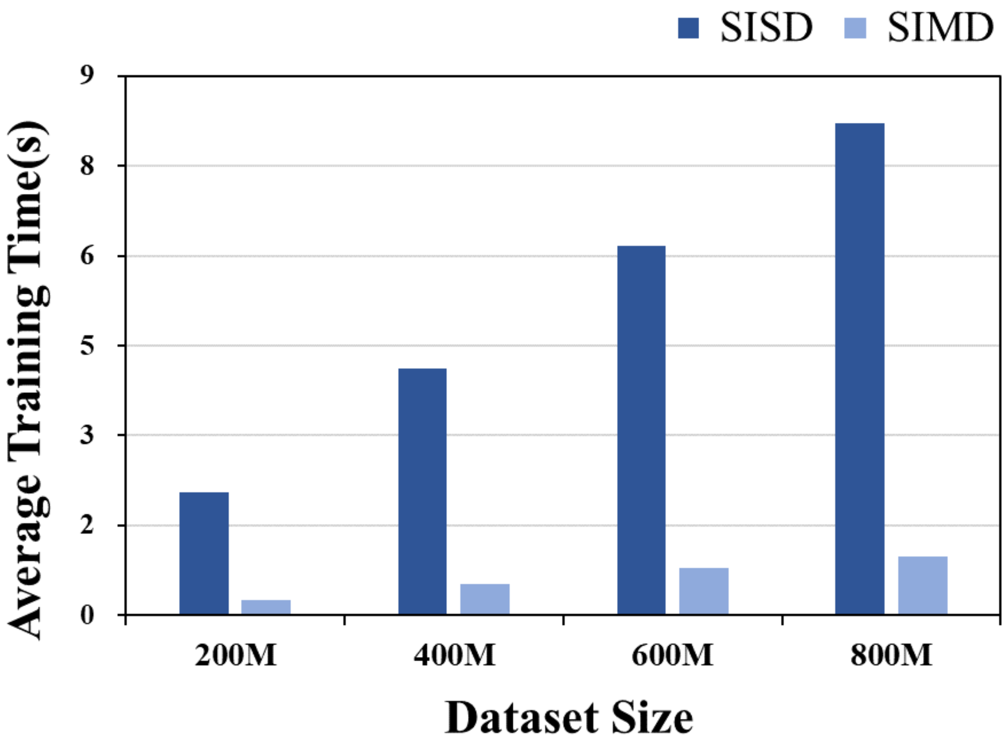
```
double intercept = (sum_Y*sum_XX - sum_X*sum_XY)/(SIZE*sum_XX - sum_X*sum_X);
```

$$\widehat{\beta} = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \rightarrow$$

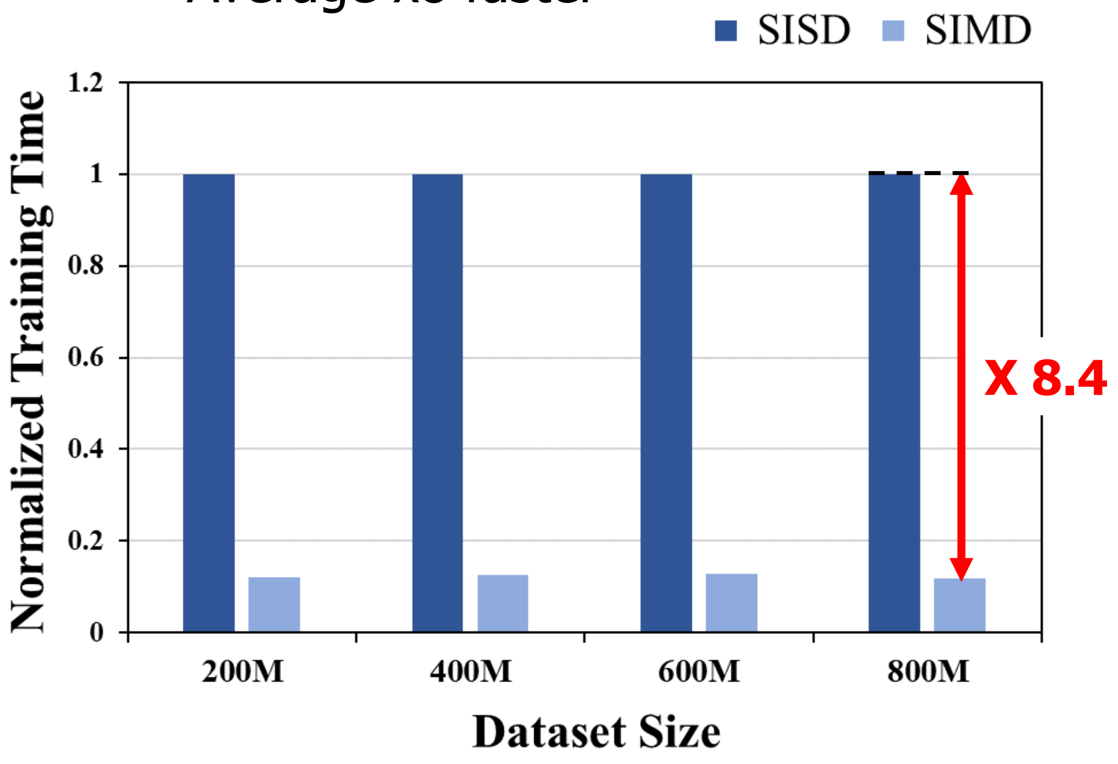
```
double slope     = (SIZE*sum_XY - sum_X*sum_Y)/(SIZE*sum_XX - sum_X*sum_X);
```

Model training with SIMD

- OLS Training Time - SIMD vs SISD
  - Average Training Time



  - Normalized Training Time
    - Average x8 faster



Future work

- Implement in real RMI code
- Apply SIMD in lookup
  - Parallel Prediction with SIMD
  - Parallel Correction with SIMD
    - Find a binary, exponential, or other search method that can apply SIMD
- Add sampling to linear regression with SIMD

Next week

- Implement in real RMI code
- Original algorithm with SISD

  vs new algorithm with SISD

  vs new algorithm with SIMD
- Possibilities for lookup of B+-tree with SIMD
- Find correction search method of RMI for SIMD

Thank you

2024. 02. 14

Presentation by Yeojin Oh, ZhuYongjie, Boseung Kim

yeojinoh@dankook.ac.kr, aeashio1111@dankook.ac.kr, bskim1102@dankook.ac.kr