

# An Improved Stochastic Gradient Method for Training Large-scale Field-aware Factorization Machine

Wei-Sheng Chin

TLC and ML.NET Team, Microsoft, Redmond, WA

## 1 Field-aware Factorization Machine

Assume that we have a feature vector  $\mathbf{x} \in \mathcal{R}^n$ ,  $\mathcal{F}(j)$  denotes the field ID of  $j$ th coordinate in  $\mathbf{x}$ , and  $m$  is the number of all possible fields. The output function of field-aware factorization can be written as

$$\hat{y} = \langle \mathbf{w}, \mathbf{x} \rangle + \sum_{j=1}^n \sum_{j'=j+1}^n \langle \mathbf{v}_{j, \mathcal{F}(j')}, \mathbf{v}_{j', \mathcal{F}(j)} \rangle x_j x_{j'}$$

or equivalently

$$\begin{aligned} \hat{y} = \langle \mathbf{w}, \mathbf{x} \rangle &+ \underbrace{\sum_{f=1}^m \sum_{\substack{j=1 \\ \mathcal{F}(j)=f}}^n \sum_{\substack{j'=1 \\ \mathcal{F}(j')=f \\ j'>j}}^n \langle \mathbf{v}_{j,f}, \mathbf{v}_{j',f} \rangle x_j x_{j'}}_{\text{intra-field interactions in field } f} \\ &+ \underbrace{\sum_{f=1}^m \sum_{f'=f+1}^m \langle \mathbf{q}_{f \rightarrow f'}, \mathbf{q}_{f' \rightarrow f} \rangle}_{\text{inter-field interactions between field } f \text{ and } f'}, \end{aligned} \quad (1)$$

where  $\langle \cdot, \cdot \rangle$  stands for inner product,  $\mathbf{w}$  is linear coefficient,  $\mathbf{v}_{j,f}$  is the  $j$ th feature's latent vector in the  $f$ th field's hidden space, and

$$\mathbf{q}_{f \rightarrow f'} = \sum_{\substack{j=1 \\ \mathcal{F}(j)=f}}^n \mathbf{v}_{j,f'} x_j.$$

Note that  $\hat{y}$  is actually an abbreviation of  $\hat{y}(\mathbf{x})$ , a function which maps the given feature vector to a real number. Also, we summarize the derivatives of (1) with respect to FFM parameters below for later discussion.

- Gradient of  $\hat{y}$  with respect to  $\mathbf{w}$ :

$$\frac{\partial \hat{y}}{\partial \mathbf{w}} = \mathbf{x} \quad (2)$$

- Gradient of  $\hat{y}$  with respect to  $\mathbf{v}_{j,\mathcal{F}(j)}$ :

$$\frac{\partial \hat{y}}{\partial \mathbf{v}_{j,\mathcal{F}(j)}} = (\mathbf{q}_{\mathcal{F}(j) \rightarrow \mathcal{F}(j)} - \mathbf{v}_{j,\mathcal{F}(j)} x_j) x_j \quad (3)$$

- Gradient of  $\hat{y}$  with respect to  $\mathbf{v}_{j,k}$  when  $k \neq \mathcal{F}(j)$ :

$$\frac{\partial \hat{y}}{\partial \mathbf{v}_{j,k}} = \mathbf{q}_{k \rightarrow \mathcal{F}(j)} x_j. \quad (4)$$

As you may have observed in (1), FFM is parameterized by  $\mathbf{w}$  and  $\mathbf{v}_{j,f}$ ,  $j = 1, \dots, n$  and  $f = 1, \dots, m$ . To determine those parameters, we consider an empirical risk minimization problem. If label-feature pairs,  $(y_1, \mathbf{x}_1), \dots, (y_l, \mathbf{x}_l)$ , are available, our objective function is

$$\min_{\mathbf{w}, \mathbf{v}_{1,1}, \dots, \mathbf{v}_{n,m}} \sum_{i=1}^l \left( \sum_{\substack{j=1 \\ x_{ij} \neq 0}}^n \left( \frac{\lambda}{2} w_j^2 + \sum_{f=1}^m \frac{\lambda'}{2} \|\mathbf{v}_{j,f}\|^2 \right) + \xi(\hat{y}_i; y_i) \right), \quad (5)$$

where  $x_{ij}$  is the  $j$ th feature of the  $i$ th example,  $\xi(\hat{y}; y)$  is the considered loss function, and  $\hat{y}_i = \hat{y}(\mathbf{x}_i)$ . Note that for binary classification, a common choice is  $\xi(\hat{y}; y) = \log(1 + e^{-y\hat{y}})$ , and for regression problems, one may consider  $\xi(\hat{y}; y) = (\hat{y} - y)^2$ .

## 2 Stochastic Gradient Methods for Solving (5)

We consider an advanced stochastic gradient method, ADAGRAD, to solve (5). Let

$$\xi'_i = \frac{\partial \xi(\hat{y}_i; y_i)}{\partial \hat{y}_i}.$$

With (2), (3), (4), and chain rule, the  $i$ th example's gradient can be computed via

$$g_{w_j} = \begin{cases} \lambda w_j + \xi'_i x_{ij} & \text{if } x_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

and

$$\mathbf{g}_{\mathbf{v}_{j,k}} = \begin{cases} \lambda' \mathbf{v}_{j,\mathcal{F}(j)} + \xi'_i (\mathbf{q}_{\mathcal{F}(j) \rightarrow \mathcal{F}(j)} - \mathbf{v}_{j,\mathcal{F}(j)} x_{ij}) x_{ij} & \text{if } k = \mathcal{F}(j) \text{ and } x_{ij} \neq 0 \\ \lambda' \mathbf{v}_{j,k} + \xi'_i \mathbf{q}_{k \rightarrow \mathcal{F}(j)} x_{ij} & \text{if } k \neq \mathcal{F}(j) \text{ and } x_{ij} \neq 0 \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (7)$$

For the sake of simplicity, we drop the example index  $i$  if the context is clear. One our training iteration can be decomposed into two consecutive steps, Algorithm 1 and Algorithm 2. In the first step, we compute the output value via (1). The second step calculates the stochastic gradient and then update the model. Notice that some intermediate variables,  $\hat{y}$  and  $\mathbf{q}_{f \rightarrow f'}$ ,  $\forall f, f' \in \{1, \dots, m\}$ , obtained in Algorithm 1 can be reused in this step. Our full algorithm is summarized in Algorithm 3.

---

**Algorithm 1** Evaluation of (1).

---

```

1: Given model parameters  $\mathbf{w}, \mathbf{v}_{1,1}, \dots, \mathbf{v}_{n,m}$ .
2: Apply zero initialization to  $\hat{y}, \mathbf{q}_{1,1}, \dots, \mathbf{q}_{m,m}$ .
3: for  $j = 1, \dots, n$  do
4:   if  $x_j = 0$  then
5:     continue
6:   end if
7:    $\hat{y} \leftarrow \hat{y} + w_j x_j$  ▷ linear term
8:   for  $f' = 1, \dots, m$  do
9:      $\mathbf{q}_{\mathcal{F}(j) \rightarrow f'} \leftarrow \mathbf{q}_{\mathcal{F}(j) \rightarrow f'} + \mathbf{v}_{\mathcal{F}(j), f'} x_j$ 
10:  end for
11: end for
12: for  $f = 1, \dots, m$  do
13:    $\hat{y} \leftarrow \hat{y} + \frac{1}{2} \langle \mathbf{q}_{f \rightarrow f}, \mathbf{q}_{f \rightarrow f} \rangle$  ▷ intra-field interaction
14:   for  $f' = f + 1, \dots, m$  do
15:      $\hat{y} \leftarrow \hat{y} + \langle \mathbf{q}_{f \rightarrow f'}, \mathbf{q}_{f' \rightarrow f} \rangle$  ▷ inter-field interaction
16:   end for
17: end for
18: for  $j = 1, \dots, n$  do
19:   if  $x_j = 0$  then
20:     continue
21:   end if
22:    $\hat{y} \leftarrow \hat{y} - \frac{1}{2} \langle \mathbf{v}_{\mathcal{F}(j), \mathcal{F}(j)}, \mathbf{v}_{\mathcal{F}(j), \mathcal{F}(j)} \rangle x_j^2$  ▷ remove self-interaction
23: end for

```

---

---

**Algorithm 2** Update of parameters via stochastic gradient method. We use  $\text{diag}(\cdot)$  to denote the diagonal matrix formed by the input vector.

---

```

1: Given model parameters  $\mathbf{w}, \mathbf{v}_{1,1}, \dots, \mathbf{v}_{n,m}$ , their learning rates
    $G_1, \dots, G_j, H_{1,1}, \dots, H_{n,m}$ , and  $\mathbf{q}_{1,1}, \dots, \mathbf{q}_{m,m}$  obtained via Algorithm
   2.
2: Apply zero initialization to  $\hat{y}, \mathbf{q}_{1,1}, \dots, \mathbf{q}_{m,m}$ .
3: Compute  $\xi'_i$ .
4: for  $j = 1, \dots, n$  do
5:   if  $x_j = 0$  then
6:     continue
7:   end if
8:   Compute  $g_{w_j}$  via (6).
9:    $G_j \leftarrow G_j + g_{w_j}^2$  ▷ accumulate squared gradient
10:   $w_j \leftarrow w_j - \eta G_j^{-\frac{1}{2}} g_{w_j}$  ▷ ADAGRAD step
11: end for
12: for  $j = 1, \dots, n$  do
13:   if  $x_j = 0$  then
14:     continue
15:   end if
16:   for  $f' = 1, \dots, m$  do
17:     Compute  $\mathbf{g}_{\mathbf{v}_{j,f'}}$  via (7).
18:      $H_{j,f'} \leftarrow H_{j,f'} + \text{diag}(\mathbf{g}_{\mathbf{v}_{j,f'}})^2$  ▷ accumulate squared gradient
19:      $\mathbf{v}_{j,f'} \leftarrow \mathbf{v}_{j,f'} - \eta H_{j,f'}^{-\frac{1}{2}} \mathbf{g}_{\mathbf{v}_{j,f'}}$  ▷ ADAGRAD step
20:   end for
21: end for

```

---



---

**Algorithm 3** A  $T$ -iteration procedure for learning field-aware factorization machine.

---

```

1: Initialize  $\mathbf{w}, \mathbf{v}_{1,1}, \dots, \mathbf{v}_{n,m}$  with random variables and specify learning
   rate scale  $\eta$ .
2: Assign one to all learning rates,  $G_1, \dots, G_j, H_{1,1}, \dots, H_{n,m}$ .
3: for  $t = 1, \dots, T$  do
4:   Sample  $(y, \mathbf{x})$ .
5:   Perform Algorithm 1.
6:   Perform Algorithm 2.
7: end for

```

---