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

Wei-Sheng Chin

TLC Team, Microsoft, Redmond, WA

1 Field-aware Factorization Machine

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

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

or equivalently

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

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

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

Note that $\hat{y}$ is actually an abbreviation of $\hat{y}(\boldsymbol{x})$, 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 $\boldsymbol{w}$:

$$\frac{\partial \hat{y}}{\partial \boldsymbol{w}} = \boldsymbol{x} \tag{2}$$

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

$$\frac{\partial \hat{y}}{\partial \boldsymbol{v}_{j,\mathcal{F}(j)}} = (\boldsymbol{q}_{\mathcal{F}(j)\to\mathcal{F}(j)} - \boldsymbol{v}_{j,\mathcal{F}(j)} x_j) x_j \tag{3}$$

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

$$\frac{\partial \hat{y}}{\partial \boldsymbol{v}_{j,k}} = \boldsymbol{q}_{k\to\mathcal{F}(j)} x_j. \tag{4}$$

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

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

where x_{ij} is the the jth feature of the ith example, $\xi(\hat{y}; y)$ is the considered loss function, and $\hat{y}_i = \hat{y}(\boldsymbol{x}_i)$. 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 ith example's gradient can be computed via

$$g_{w_j} = \begin{cases} \lambda w_j + \xi_i' x_{ij} & \text{if } x_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and

$$\boldsymbol{g}_{\boldsymbol{v}_{j,k}} = \begin{cases} \lambda' \boldsymbol{v}_{j,\mathcal{F}(j)} + \xi_i' (\boldsymbol{q}_{\mathcal{F}(j)\to\mathcal{F}(j)} - \boldsymbol{v}_{j,\mathcal{F}(j)} x_{ij}) x_{ij} & \text{if } k = \mathcal{F}(j) \text{ and } x_{ij} \neq 0 \\ \lambda' \boldsymbol{v}_{j,k} + \xi_i' \boldsymbol{q}_{k\to\mathcal{F}(j)} x_{ij} & \text{if } k \neq \mathcal{F}(j) \text{ and } x_{ij} \neq 0 \\ \boldsymbol{0} & \text{otherwise.} \end{cases} \tag{7}$$

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 $\boldsymbol{q}_{f \to f'}$, $\forall f, f' \in \{1, \ldots, m\}$, obtained in Algorithm 1 can be reused in this step. Our full algorithm is summarized in Algorithm 3.

Algorithm 1 Evaluation of (1).

Given model parameters $\boldsymbol{w}, \boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{n,m}$.
Apply zero initialization to $\hat{y}, \boldsymbol{q}_{1,1}, \ldots, \boldsymbol{q}_{m,m}$.
for $j = 1, \ldots, n$ **do**
  if $x_j = 0$ **then**
    continue
  end if
  $\hat{y} \leftarrow \hat{y} + w_j x_j$ ▷ linear term
  for $f' = 1, \ldots, m$ **do**
    $\boldsymbol{q}_{\mathcal{F}(j) \to f'} \leftarrow \boldsymbol{q}_{\mathcal{F}(j) \to f'} + \boldsymbol{v}_{\mathcal{F}(j), f'} x_j$
  end for
end for
for $f = 1, \ldots, m$ **do**
  $\hat{y} \leftarrow \hat{y} + \frac{1}{2} \left\langle \boldsymbol{q}_{f \to f}, \boldsymbol{q}_{f \to f} \right\rangle$ ▷ intra-field interaction
  for $f' = f + 1, \ldots, m$ **do**
    $\hat{y} \leftarrow \hat{y} + \left\langle \boldsymbol{q}_{f \to f'}, \boldsymbol{q}_{f' \to f} \right\rangle$ ▷ inter-field interaction
  end for
end for
for $j = 1, \ldots, n$ **do**
  if $x_j = 0$ **then**
    continue
  end if
  $\hat{y} \leftarrow \hat{y} - \frac{1}{2} \left\langle \boldsymbol{v}_{\mathcal{F}(j), \mathcal{F}(j)}, \boldsymbol{v}_{\mathcal{F}(j), \mathcal{F}(j)} \right\rangle x_j^2$ ▷ remove self-interaction
end for

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

Given model parameters $\boldsymbol{w}, \boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{n,m}$, their learning rates $G_1, \ldots, G_j, H_{1,1}, \ldots, H_{n,m}$, and $\boldsymbol{q}_{1,1}, \ldots, \boldsymbol{q}_{m,m}$ obtained via Algorithm 2.
Apply zero initialization to $\hat{y}, \boldsymbol{q}_{1,1}, \ldots, \boldsymbol{q}_{m,m}$.
Compute ξ_i'.
for $j = 1, \ldots, n$ **do**
    if $x_j = 0$ **then**
        continue
    end if
    Compute g_{w_j} via (6).
    $G_j \leftarrow G_j + g_{w_j}^2$ ▷ accumulate squared gradient
    $w_j \leftarrow w_j - \eta G_j^{-\frac{1}{2}} g_{w_j}$ ▷ ADAGRAD step
end for
for $j = 1, \ldots, n$ **do**
    if $x_j = 0$ **then**
        continue
    end if
    for $f' = 1, \ldots, m$ **do**
        Compute $\boldsymbol{g}_{\boldsymbol{v}_{j,f'}}$ via (7).
        $H_{j,f'} \leftarrow H_{j,f'} + \mathrm{diag}(\boldsymbol{g}_{\boldsymbol{v}_{j,f'}})^2$ ▷ accumulate squared gradient
        $\boldsymbol{v}_{j,f'} \leftarrow \boldsymbol{v}_{j,f'} - \eta H_{j,f'}^{-\frac{1}{2}} \boldsymbol{g}_{\boldsymbol{v}_{j,f'}}$ ▷ ADAGRAD step
    end for
end for

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

Initialize $\boldsymbol{w}, \boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{n,m}$ with random variables and specify leraning rate scale η.
Assign one to all learning rates, $G_1, \ldots, G_j, H_{1,1}, \ldots, H_{n,m}$.
for $t = 1, \ldots, T$ **do**
    Sample $(y, \boldsymbol{x})$.
    Perform Algorithm 1.
    Perform Algorithm 2.
end for