Convergence guarantees for gradient descent methods in optimization for non-convex function approximation (distributed neural network training)

Hoang Trung Hieu

2021

Previous semester

Challenges in fererated learning

- *Massively distributed*. The number of mobile device owners is massively bigger than average of the number of training samples on each device.
- *Unbalanced*. Some users produce significantly more data than others.
- **Non-IID. The data generated by each user are quite different.**

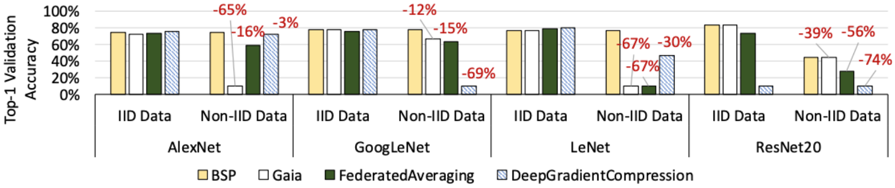

Figure. Top-1 validation accuracy for IMAGE CLASSIFICATION over the CIFAR-10 dataset.

Problem description

Let us consider the $C-$class classification problem.
Let $f : X \to S = \{z | \sum_{i=1}^{C} z_i = 1, z_i \geq 0 \ \forall i \in [C]\}$ denote the prediction function and f_i predicts the probability that the sample belongs to the $i-$th class.
The learning problem is to solve the following optimization problem:

$$\min_w L(w) = \min_w \mathbb{E}_{x,y \sim p} \left[\sum_{i=1}^{C} \mathbb{1}_{y=i} \log f_i(x, w) \right]$$
$$= \min_w \sum_{i=1}^{C} p(y = i) \mathbb{E}_{x|y=i} [\log f_i(x, w)]$$

Weight updates at centralized setting:

$$w_t^{(c)} = w_{t-1}^{(c)} - \eta \sum_{i=1}^{C} p(y=i) \nabla_w \mathbb{E}_{x|y=i}[\log f_i(x, w_{t-1}^{(c)})]$$

Weight updates at the $k-$th client

$$w_t^{(k)} = w_{t-1}^{(k)} - \eta \sum_{i=1}^{C} p^{(k)}(y=i) \nabla_w \mathbb{E}_{x|y=i}[\log f_i(x, w_{t-1}^{(k)})]$$

The $m-$th synchronization (assume synchronization is conducted every T steps)

$$w_{mT}^{(f)} = \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} w_{mT}^{(k)}$$

$n^{(k)}$ denote the amount of data and $p^{(k)}$ denote the data distribution on client k.

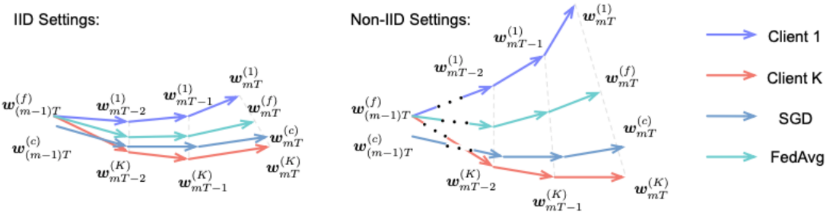

Figure. Ilustration of the weight divergence for federated learning with-IID and non-IID data.

$$||\boldsymbol{w}_{mT}^{(f)} - \boldsymbol{w}_{mT}^{(c)}|| \leq \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} (a^{(k)})^T ||\boldsymbol{w}_{(m-1)T}^{(f)} - \boldsymbol{w}_{(m-1)T}^{(c)}|| + \eta \sum_{k=1}^{K} \frac{n^{(k)}}{\sum_{k=1}^{K} n^{(k)}} \sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)|| \sum_{j=1}^{T-1} (a^{(k)})^j g_{max}(\boldsymbol{w}_{mT-1-k}^{(c)}),$$

where $g_{max}(\boldsymbol{w}) = \max_{i=1}^{C} ||\nabla_{\boldsymbol{w}} \mathbb{E}_{\boldsymbol{x}|y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})||$ *and* $a^{(k)} = 1 + \eta \sum_{i=1}^{C} p^{(k)}(y=i) \lambda_{\boldsymbol{x}|y=i}$.

Convex optimization of FedAvg on Non-IID Data (X.Li, K.Huang, W.Yang, S.Wang and Z.Zhang, 2020)

Assumptions

- $L-$ **Lipschitz gradient**:$\|\nabla f_i(u) - \nabla f_i(v)\| \leq L \|u - v\|$
- $\mu-$ **strongly convex:** $f_i(u) \geq f_i(v) + (u - v)^T \nabla f_i(v) + \frac{\mu}{2} \|u - v\|^2$
- **Bounded variance:**

$$\mathbb{E}_{\xi_k \sim D_i} \left[\|\nabla F(w, \xi_k) - \nabla f_k(w)\|^2 \right] \leq \sigma^2, \ \forall k, w$$

- **Bounded gradient:**

$$\mathbb{E}_{\xi_k \sim D_i} \left[\|\nabla F(w, \xi_k)\|^2 \right] \leq G^2, \ \forall k, w$$

Main aim : Give an analysis whether it is possible to give convergence guarantees (non-convex) in case the data over each devices is non-iid.

Non-convex optimization on gradient-based methods

Ideal 1: Splitting the loss function into convex terms and non-convex terms.

$$\min_{x \in C} f(x) = \min_{x \in C} \{g(x) - h(x)\}$$

Assumptions

- *f is bounded below C*
- *h is continuous and convex*
- *g is continuous differentiable and M_g smooth*

Algorithm 1: Subgradient-type method.

Initialization. Choose $x_0 \in \int(C)$, level set of x_0 is in C.

for *each round* $k = 1, 2, \cdots, T$ **do**

Update: $x_{k+1} = x_k - \alpha(\nabla g(x_k) - u_k)$ where u_k is chosen randomly from $\partial h(x_k)$ and learning rate $\alpha \in [0, 1/M_g]$.

end

Theorem

$f(x_k)$ is strictly decreasing and converges. The limit point of x_k is also a critical point of f. Moreover, for all k,

$$Avg\left(\|\nabla f(x_k)\|_2^2\right) \leq \frac{2(f(x_0) - f(x*))}{(k+1)}$$

Non-convex optimization on gradient-based methods

Ideal 2: Using local smoothness and maximal nondegeneracy

$$f(\theta) = \mathbb{E}[F(\theta, X)]$$

Assumptions

- *$\mathcal{M}$ is locally smooth (there exists an open set $U \subseteq \mathbb{R}^d$such that $M \cap U$ is a non-empty $\Psi-$dimensional C^1-submanifold of $\mathbb{R}^d$).*
- *f is locally three times continous differentiable on the local set of $\mathcal{M}$. And the Hessian matrix of f is maximally non degenerate. (rank Hess $(f)(\theta) = d - \Psi = codim(M \cap U)$)*

Algorithm. Initialization: The initial data was sample from the bounded open set $A \subseteq \mathbb{R}^d$ that contains at least one element in local set of $\mathcal{M}$. Denote by $\theta_0^{k,M,r} : \Omega \to \mathbb{R}^d$ indicates k-th sample in the sampling set of size K, mini-batch size M, and parameter $r > 0$ involving to learning parameter. The initial data is uniformly distributed on A. i.i.d and independent from $X_{k,m,n}$.

Weights update: We compute independent solutions to SGD in the way

$$\theta_n^{k,M,r} = \theta_{n-1}^{k,M,r} - \frac{r}{n^\rho M}\left[\sum_{i=1}^{m} \nabla_\theta F(\theta_{n-1}^{k,M,r}, X_{k,n,m})\right]$$

Mini-batch approximation: $F^{K,\mathfrak{M},n} : \mathbb{R}^d \times \Omega \to \mathbb{R}$ is approximated as

$$F^{K,\mathfrak{M},n}(\theta, w) = \frac{1}{\mathfrak{M}} \sum_{i=1}^{\mathfrak{M}} F\left(\theta, X_{1,n+1,m}(w)\right)$$

After that, we identify the value that minimizes $F^{K,\mathfrak{M},n}$ in the sense that we compute a random variable $\theta_n^{K,M,\mathfrak{M},r} : \Omega \to \mathbb{R}^d$ which satisfies that

$$\sum_{m=1}^{\mathfrak{M}} F\left(\theta_n^{K,M,\mathfrak{M},r}, X_{1,n+1,m}\right) = \min_{k\in[K]} \left[\sum_{m=1}^{\mathfrak{M}} F\left(\theta_n^{k,M,r}, X_{1,n+1,m}\right)\right]$$

Theorem

After running the above algorithm with $p \in (2/3; 1)$. *There exist* $\tau, c > 0$, $\kappa \in (0,1)$ *such that for every* $n, k, M, \mathfrak{M}, r \in (0, \tau)$, $\epsilon \in (0,1)$ *we get*

$$\mathbb{P}\left(\textit{ Distance betweenf}(\theta_n^{k,M,\mathfrak{M},r}) \textit{ and minima bigger than } \epsilon \right)$$

$$= \mathbb{P}\left(f(\theta_n^{k,M,\mathfrak{M},r}) - \inf_{\theta} f(\theta) \geq \epsilon \right) \leq \frac{cK}{\epsilon^2 \mathfrak{M}} + \left[\kappa + c\left(\frac{1}{\epsilon^2 n^{\rho}} + \frac{n^{1-p}}{\sqrt{M}} \right) \right]^K$$

Ideal 3: Using extrapolation

Problem

$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \mathbb{E}\left[f(x, \xi)\right]$$

Assumptions

- *$f(x)$ is $L-$smooth.*
- *There exist Δ such that $f(x) - f(x_*) \leq \Delta$ for all $x \in \mathbb{R}^d$ where x_* be the global minimum of $f(x)$.*
- *$f(x, \xi)$ is differentiable.*
- *(Bounded variance)* $\mathbb{E}\left[\|\nabla F(x, \xi) - \nabla f(x)\|^2\right] \leq G^2, \ \forall x, \xi$

Algorithm 2: Stagewise SGDE

Algorithm `StagewiseSGDE()`

Initialization. $x^0 = x_0$

for $s = 1, \cdots, S$ **do**

$f_s(x) = f(x) + \frac{1}{2\gamma} \left\| x - x^{s-1} \right\|^2$

$x^s = \mathtt{SGDE}(x^{s-1}, f_s, \eta_s, T_s)$

end

return x_τ where τ is chosen from $1, \cdots, S$ with probability $\mathbb{P}(\tau = i) = \frac{w_i}{\sum_{j=1}^{s} w_s}$

Procedure $\mathtt{SGDE}(x_0, f, \eta, T)$

Initialization. $z_0 = x_0; g_0 = \nabla f(x_0, \xi_0)$

for *each round* $t = 1, 2, \cdots, T$ **do**

$x_t = z_{t-1} - \eta g_{t-1}$

$g_t = \nabla f(x_t, \xi_t)$

$z_t = z_{t-1} - \eta g_t$

end

return $\hat{x}_t = \frac{1}{T} \sum_{t=1}^{T} x_t$

Theorem

After running Stagewise SGDE with $\gamma = 1/4L; w_s = s^\alpha$ $(\alpha > 1)$ *and choosing the learning parameter and the number of iteration at* $s-$*stage as follow:* $\eta_s = \frac{c\gamma}{3s} \leq \frac{1}{2L}\frac{\gamma}{3}$ *and* $T_s = \frac{36s}{c}$, *we got the estimation:*

$$\mathbb{E}\left[\|\nabla f(x_\tau)\|^2\right] \leq \frac{20\Delta(\alpha+1)}{\gamma(S+1)} + \frac{480G^2c(\alpha+1)}{S+1} - \frac{60\sum_{s=1}^{S+1} w_s D_{T_s}}{\gamma\sum_{s=1}^{S+1} w_s}$$

where $D_{T_s} = \dfrac{1}{16T_s\eta_s}\sum_{t=1}^{T_s}\|x_t - x_{t-1}\|^2$. *If we expect* $\mathbb{E}\left[\|\nabla f(x_\tau)\|^2\right] \leq \epsilon^2$, *the number of stage should be* $O(1/\epsilon^2)$.

Idea

Traditional loss function:

$$L(w) = \frac{1}{K} \sum_{i=1}^{K} f_i(w)$$

New loss function:

$$L(w) = \frac{1}{K} \sum_{i=1}^{K} \min_{\theta_i \in \mathbb{R}^n} \{ f_i(\theta_i) + \frac{\lambda}{2} \|\theta_i - w\|^2 \}$$

Algorithm 3: Ongoing algorithms

Server executes: Initialization w_0 ;

for *each round* $t = 1, 2, \cdots$ **do**

  for *each client* $k \in S_t$ **do**

    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$;

  end

  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$;

end

ClientUpdate:

for $i = 1, \cdots, N$ **do**

  $w_{i,0}^t = w_t$

  for $r = 1, \cdots, R$ **do**

    $\hat{\theta}(w_{i,r}^t) = prox_{f_i/\lambda}(w_{i,r}^t)$

    $w_{i,r+1}^t \leftarrow w_{i,r} - \eta\lambda(w_{i,t}^t - \hat{\theta}(w_{i,r}^t))$

  end

end

THANK YOU FOR YOUR ATTENTION!