A Representation-Focused Training Algorithm for Deep Networks

Vincent Roulet, Corinne Jones, Zaid Harchaoui

IEEE Data Science and Learning Workshop 2022

05/22/2022

UNIVERSITY of WASHINGTON

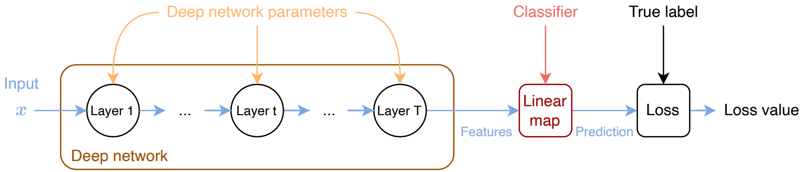
NSF

Representation-Focused Training

Idea

- Training of deep networks consists in:
  - ◇ Learning a representation of the inputs
  - ◇ Classifying the inputs from their representation
- Given a pretrained network, optimizing the classifier is easy

Can we take advantage of separating the training of deep networks into learning a feature representation and classifying the inputs?



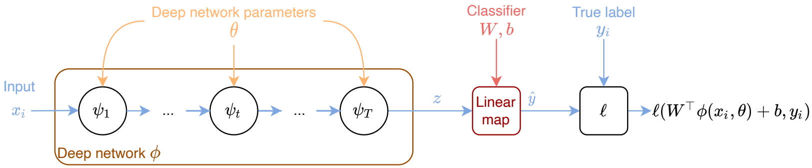
Representation-Focused Training

Overall training objective

Given n data input-output (x_i, y_i) samples, solve

$$\min_{\theta, W, b} \frac{1}{n} \sum_{i=1}^{n} \ell(W^\top \phi(x_i, \theta) + b, y_i) + \Omega(\theta, W) ,$$

with $\Omega(\theta, W)$ some regularization term



Reduced objective

$$f(\theta) := \min_{W, b} \frac{1}{n} \sum_{i=1}^{n} \ell(W^\top \phi(x_i, \theta) + b, y_i) + \Omega(\theta, W).$$

Partially Minimized Objectives

Reduced objectives

Given an objective $h(u, v)$, consider

$$f(u) = \min_v h(u, v)$$

Why?

- May accelerate optimization process

$\rightarrow$ Wiberg algo. for matrix fact. (Wiberg 1976)

$\rightarrow$ Pseudo-likelihood (Besag 1975)

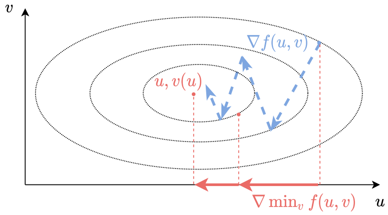
Challenges for deep networks

- Here objective of the form $\sum_i h_i(u, v)$

$\rightarrow$ Amenable to stochastic optimization ✓

- Reduced objective $f(u) = \min_v \sum_i h_i(u, v)$

$\rightarrow$ Breaks finite-sum structure...



Paths taken by gradient descent on original objective vs. reduced objective

Idea: Consider computing reduced objective on mini-batches

Biased Stochastic Gradient Descent on Reduced Objective

Algorithm

1. Compute reduced objective on mini-batch $S \subseteq \{1, \ldots, n\}$
   (given in closed form for, e.g., squared loss, squared penalization)

$$f_S(\theta_k) = \min_{W,b} \frac{1}{m} \sum_{i \in S} \ell(W^\top \phi(x_i, \theta_k) + b, y_i) + \Omega(\theta_k, W),$$

2. Access gradient of $f_S(\theta)$ by auto.-diff. and update

$$\theta_{k+1} = \theta_k - \gamma \nabla f_S(\theta_k)$$

Analysis challenges

- Stochastic estimate $\nabla f_S(\theta)$ of $f(\theta)$ is biased: $\mathbb{E}_S[\nabla f_S(\theta)] \neq \nabla f(\theta)$
- But bias may be controlled by mini-batch size

Convergence Analysis

Setup

- Squared loss $\ell(\hat{y}, y) = (y - \hat{y})^2$, regularization $\Omega(W, \theta) = \lambda \|W\|_F^2 + \Omega(\theta)$
- Bounded, Lip. continuous feature rep. with

$$r = \sup_{\theta, x} \|\phi(x, \theta)\|_2 < +\infty, \quad \ell = \sup_{\theta, x} \|\nabla_\theta \phi(x, \theta)\|_2 < +\infty.$$

Theorem

The mean squared error of the estimate $\nabla f_S(\theta)$ of $\nabla f(\theta)$ is controlled as

$$\mathbb{E}[\|\nabla f_S(\theta) - \nabla f(\theta)\|_2^2] \leq O\left(n^2 q_m \ell^2 r^6 / \lambda^4\right),$$

where $q_m = (n-m)/((n-1)m)$ for mini-batches S of size m.
After K iterations, for a stepsize $\gamma \leq 1/(2L)$ with L the smoothness of the reduced objective f,

$$\min_{k \in \{0, \dots, K-1\}} \mathbb{E}\|\nabla f(\theta_k)\|^2 \leq c\,\frac{f(\theta_0) - f^*}{\gamma K} + O\left(\frac{n^2 q_m l^2 r^6}{\lambda^4}\right),$$

with c a universal constant.

Extension to Non-Squared Losses

Ultimate Layer Reversal (ULR) step

Given current parameters θ_k, W_k, b_k, step-size γ, mini-batch S

1. Compute predictions $\hat{y}_i = \phi(x_i, \theta_k)^T W_k + b_k$ for $i \in S$
2. Compute quadratic approx. $q_{\ell_i}(\cdot; \hat{y}_i)$ of $\ell_i = \ell(\cdot, y_i)$ around $\hat{y}_i$ for $i \in S$
3. Compute reduced objective S based on quad. approx.

$$f_S(\theta) = \min_{W,b} \frac{1}{m} \sum_{i \in S} q_{\ell_i}(W^\top \phi(x_i, \theta) + b; \hat{y}_i) + \Omega(\theta, W)$$

4. Update parameters $\theta_{k+1} = \theta_k - \gamma \nabla f_S(\theta_k)$ with $\nabla f_S(\theta_k)$ given by auto-diff
5. Compute corresponding classifiers from the quadratic approx., i.e.,

$$W_{k+1}, b_{k+1} = \arg\min_{W,b} \frac{1}{m} \sum_{i \in S} q_{\ell_i}(W^\top \phi(x_i, \theta_{k+1}) + b; \hat{y}_i) + \Omega(\theta_{k+1}, W)$$

Representation-Focused Training

Task

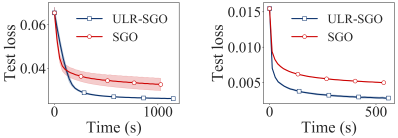
Image classification
with Convolutional Kernel Networks

Algorithms

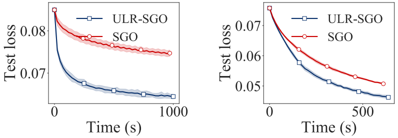
- SGD on original obj.
- SGD on reduced obj. (ULR-SGO)

Results

$\rightarrow$ Optimizing reduced objective with biased gradient estimates can lead to faster optim.



LeNet-5 CKN on MNIST
with 8 filters/layer & 128 filters/layer



All-CNN-C CKN on CIFAR-10
with 8 filters/layer & 128 filters/layer

Squared loss

Representation-Focused Training

Task

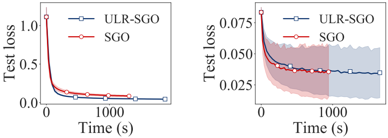
Image classification
with Convolutional Kernel Networks

Algorithms

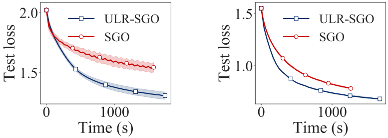
- SGD on original obj.
- SGD on reduced obj. (ULR-SGO)

Results

→ Optimizing reduced objective with biased gradient estimates can lead to faster optim.



LeNet-5 CKN on MNIST
with 8 filters/layer & 128 filters/layer



All-CNN-C CKN on CIFAR-10
with 8 filters/layer & 128 filters/layer

Logistic loss

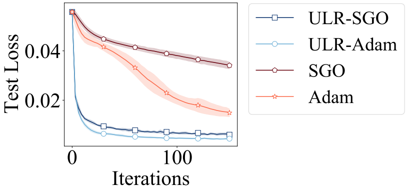
Representation-Focused Training

Task
Image classification
with Convolutional Kernel Networks

Algorithms
- SGD on original obj.
- SGD on reduced obj. (ULR-SGO)

Results
→ Optimizing reduced objective
with biased gradient estimates can
lead to faster optim.

Plug-in Oracle
- Can use $\nabla f_S(\theta)$ in any algo.
such as Adam



Squared Loss

Thank you
for your attention!

Biased Stochastic Gradient Descent on Reduced Objective

- Denote $\Phi(X,\theta) = (\phi(x_1,\theta),\ldots,\phi(x_n,\theta))^\top \in \mathbb{R}^{n\times d}$, objective is

$$\min_{\theta,W,b} \frac{1}{2n}\|\Phi(X,\theta)W + 1_n\, b^\top - Y\|_F^2 + \frac{\lambda}{2}\|W\|_F^2 + \frac{\mu}{2}\|\theta\|_2^2.$$

- Reduced objective is $f_S(\theta) = h_S(Z) + \mu\|\theta\|_2^2/2$, for $Z = (z_1,\ldots,z_n)^\top = \Phi(X,\theta)$,

$$\begin{aligned}
h_S(Z) &= \frac{1}{2m}\|Z_S W_S - Y_S\|_F^2 + \frac{\lambda}{2}\|W_S\|_F^2,\\
W_S &= (\lambda\,\mathrm{I} + \Sigma_S)^{-1} C_S,\\
\Sigma_S &= \mathrm{Cov}_S(z,z), \quad C_S = \mathrm{Cov}_S(z,y),\\
Z_S^\top &= (\delta_{iS}(z_i - E_S[z]))_{i=1}^n,\ \ Y_S^\top = (\delta_{iS}(y_i - E_S[y]))_{i=1}^n,
\end{aligned}$$

- We then have that

$$\nabla h_S(Z) = \frac{1}{m}(Z_S W_S - Y_S)\, W_S^\top,$$

and for $j \in \{1,\ldots,p\}$, denoting $g_{j,i} = \partial\phi(x_i,\theta)/\partial\theta_j$,

$$\frac{\partial f_S(\theta)}{\partial\theta_j} = \frac{1}{m}\sum_{i\in S}(W_S^\top z_{S,i} - y_{S,i})^\top W_S^\top g_{j,i} + \mu\theta_j,$$

where $z_{S,i} = z_i - E_S[z]$, $y_{S,i} = y_i - E_S[y]$.

References

Besag, J. (1975), 'Statistical analysis of non-lattice data', *Journal of the Royal Statistical Society: Series D (The Statistician)* **24**(3), 179–195.

Wiberg, T. (1976), Computation of principal components when data are missing, *in* 'Proc. Second Symp. Computational Statistics', pp. 229–236.