An Adaptive Empirical Bayesian Method for Sparse Deep Learning

Wei Deng, Xiao Zhang, Faming Liang, Guang Lin
Purdue University, West Lafayette, IN, USA.

Algorithm

Algorithm 1 SGLD-SA with SSGL priors

Initialize: $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{\rho}^{(1)}$, $\boldsymbol{\kappa}^{(1)}$, $\sigma^{(1)}$ and $\delta^{(1)}$ from scratch, set target sparse rates $\mathbb{D}$, $\mho$ and $\mathbb{S}$
for $k \leftarrow 1 : k_{\max}$ **do**
  Sampling
  $\boldsymbol{\beta}^{(k+1)} \leftarrow \boldsymbol{\beta}^{(k)} + \epsilon^{(k)} \nabla_{\boldsymbol{\beta}} Q(\cdot|\mathcal{B}^{(k)}) + \mathcal{N}(0, 2\epsilon^{(k)}/\tau)$
  Stochastic Approximation for Latent Variables
  SA: $\boldsymbol{\rho}^{(k+1)} \leftarrow (1-\omega^{(k+1)})\boldsymbol{\rho}^{(k)} + \omega^{(k+1)}\tilde{\boldsymbol{\rho}}^{(k+1)}$
  SA: $\boldsymbol{\kappa}^{(k+1)} \leftarrow (1-\omega^{(k+1)})\boldsymbol{\kappa}^{(k)} + \omega^{(k+1)}\tilde{\boldsymbol{\kappa}}^{(k+1)}$
  SA: $\sigma^{(k+1)} \leftarrow (1-\omega^{(k+1)})\sigma^{(k)} + \omega^{(k+1)}\tilde{\sigma}^{(k+1)}$
  SA: $\delta^{(k+1)} \leftarrow (1-\omega^{(k+1)})\delta^{(k)} + \omega^{(k+1)}\tilde{\delta}^{(k+1)}$
  if Pruning then
    Prune the bottom-s% lowest magnitude weights
    Increase the sparse rate $s \leftarrow \mathbb{S}(1-\mathbb{D}^{k/\mho})$
  end if
end for

Probability driven by the L_2 penalty

$$\tilde{\rho}_{lj}^{(k+1)} = \mathrm{P}(\gamma_{lj}=1|\boldsymbol{\beta}_l^{(k)}, \delta_l^{(k)}) = \frac{a_{lj}}{a_{lj}+b_{lj}},$$

where $a_{lj} = \pi(\beta_{lj}^{(k)}|\gamma_{lj}=1)\delta_l^{(k)}$ and $b_{lj} = \pi(\beta_{lj}^{(k)}|\gamma_{lj}=0)(1-\delta_l^{(k)})$.

Unnormalized adaptive L_1 and L_2 penalties

$$\tilde{\kappa}_{lj0} = \mathop{\mathrm{E}}_{\gamma_l|\cdot,\mathcal{B}}\left[\frac{1}{v_0(1-\gamma_{lj})}\right] = \frac{1-\rho_{lj}}{v_0};$$

$$\tilde{\kappa}_{lj1} = \mathop{\mathrm{E}}_{\gamma_l|\cdot,\mathcal{B}}\left[\frac{1}{v_1\gamma_{lj}}\right] = \frac{\rho_{lj}}{v_1}.$$

Data-driven adaptive standard deviation

$$\tilde{\sigma}^{(k+1)} = \begin{cases} \dfrac{R_b + \sqrt{R_b^2 + 4R_aR_c}}{2R_a} & \text{(Reg)}, \\ \dfrac{C_b + \sqrt{C_b^2 + 4C_aC_c}}{2C_a} & \text{(Cla)}, \end{cases}$$

where $R_a = N + \sum_{l\in\mathcal{X}} p_l + \nu$, $C_a = \sum_{l\in\mathcal{X}} p_l + \nu + 2$, $R_b = C_b = \sum_{l\in\mathcal{X}} ||\boldsymbol{\mathcal{V}}_{0l}\boldsymbol{\beta}_l^{(k+1)}||_1$, $R_c = I + J + \nu\lambda$, $C_c = J + \nu\lambda$, $I = \frac{N}{n}\sum_{i\in\mathcal{S}}\left(y_i - \psi(\boldsymbol{x}_i;\boldsymbol{\beta}^{(k+1)})\right)^2$, $J = \sum_{l\in\mathcal{X}} ||\boldsymbol{\mathcal{V}}_{1l}^{1/2}\boldsymbol{\beta}_l^{(k+1)}||^2$.

Layer-wise adaptive sparsity

$$\tilde{\delta}_l^{(k+1)} = argmax_{\delta_l\in\mathbb{R}} Q_2(\delta_l|\boldsymbol{\beta}_l^{(k)}, \delta_l^{(k)}) = \frac{\sum_{j=1}^{p_l}\rho_{lj} + a - 1}{a + b + p_l - 2}.$$

Abstract

We propose a novel adaptive empirical Bayesian method for sparse deep learning, where the sparsity is ensured via a class of self-adaptive spike-and-slab priors. The proposed method works by alternatively sampling from an adaptive hierarchical posterior distribution using stochastic gradient Markov Chain Monte Carlo and smoothly optimizing the hyperparameters using stochastic approximation. We further prove the convergence of the proposed method to the asymptotically correct distribution. Empirical applications of the proposed method lead to the state-of-the-art performance on MNIST and Fashion MNIST with shallow convolutional neural networks and the state-of-the-art compression performance on CIFAR10 with Residual Networks. The proposed method also improves resistance to adversarial attacks.

Empirical Bayesian via Stochastic Approximation

Our interest is to sample from $\pi(\boldsymbol{\beta}, \boldsymbol{\theta}_*)$, where $\boldsymbol{\theta}_*$ is obtained when the distribution of $\boldsymbol{\beta}$ converges to the invariant distribution $\pi(\boldsymbol{\beta}, \boldsymbol{\theta}_*)$. The stochastic approximation algorithm can be applied:

(1) Sample $\boldsymbol{\beta}^{(k+1)}$ from a SGLD transition kernel $\Pi_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\beta})$, which yields the distribution $\pi(\boldsymbol{\beta}, \boldsymbol{\theta}^{(k)})$,
(2) Update $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \omega^{(k+1)}H(\boldsymbol{\theta}^{(k)}, \boldsymbol{\beta}^{(k+1)}) = \boldsymbol{\theta}^{(k)} + \omega^{(k+1)}(h(\boldsymbol{\theta}^{(k)}) + \Omega^{(k)})$.

The stochastic approximation differs from the Robbins–Monro algorithm in that sampling $\boldsymbol{\beta}$ from a transition kernel instead of a distribution introduces a Markov state-dependent noise $\Omega^{(k)}$. **By decomposing the Markov state-dependent noise Ω into martingale difference sequences and perturbations**, where the latter can be controlled by the regularity of the solution of Poisson's equation, we can guarantee the consistency.

Theorem (L_2 convergence)

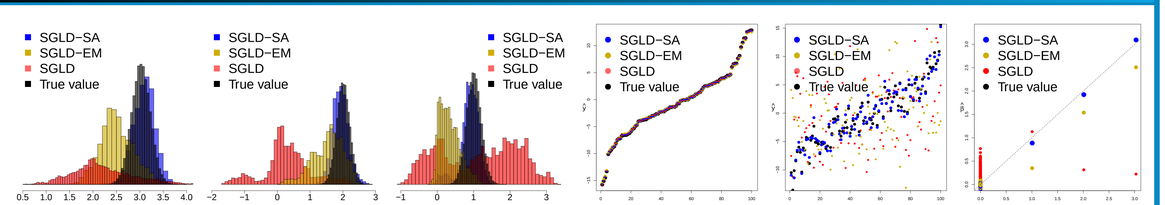
For any $\alpha \in (0, 1]$, under assumptions in the appendix, the algorithm satisfies: there exists a constant λ and an optimum $\boldsymbol{\theta}^*$ such that $\mathrm{E}\left[\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2\right] \le \lambda k^{-\alpha}$.

SGLD with adaptive latent variables forms a sequence of inhomogenous Markov chains and the weak convergence to the target posterior is equivalent to proving the weak convergence of SGLD with biased gradients.

Corollary (Weak convergence)

The random vector $\boldsymbol{\beta}^{(k)}$ from the adaptive transition kernel $\Pi_{\boldsymbol{\theta}^{(k-1)}}$ converges weakly to the invariant distribution $e^{\tau L(\boldsymbol{\beta},\boldsymbol{\theta}^*)}$ as $\epsilon \to 0$ and $k \to \infty$.

Large-p-small-n Linear Regression



Experiments

SGHMC-SA outperforms all the baselines. Nevertheless, without smooth adaptive update, SGHMC-EM often performs worse than SGHMC. While with simulated annealing, we observe further improved performance in most of the cases.

Regression on UCI datasets

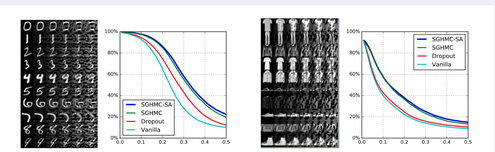
Dataset Hyperparameters	Boston 1/0.1	Yacht 1/0.1	Energy 0.1/0.1	Wine 0.5/0.01	Concrete 0.5/0.07
SGHMC	2.783±0.109	0.886±0.046	1.983±0.092	0.731±0.015	6.319±0.179
A-SGHMC	2.848±0.126	0.808±0.048	1.419±0.067	0.671±0.019	5.978±0.166
SGHMC-EM	2.813±0.140	0.823±0.053	2.077±0.108	0.729±0.018	6.275±0.169
A-SGHMC-EM	2.767±0.154	0.815±0.052	1.435±0.069	0.627±0.008	5.762±0.156
SGHMC-SA	**2.779±0.133**	**0.789±0.050**	**1.948±0.081**	**0.654±0.010**	**6.029±0.131**
A-SGHMC-SA	**2.692±0.120**	**0.782±0.052**	**1.388±0.052**	**0.620±0.008**	**5.687±0.142**

We compared the proposed algorithm with other popular ones on MNIST and Fashion MNIST.

Auto-tuning Hyperparameters on (F)MNIST

Dataset	MNIST	DA-MNIST	FMNIST	DA-FMNIST
Vanilla	99.31	99.54	92.73	93.14
Dropout	99.38	99.56	92.81	93.35
SGHMC	99.47	99.63	92.88	94.29
SGHMC-SA	**99.59**	**99.75**	**93.01**	**94.38**

Defenses against Adversarial Attacks



(a) $\zeta = ...$ (b) MNIST (c) $\zeta = ...$ (d) FMNIST

Figure 2: Adversarial test accuracies based on adversarial images of different levels

Residual Network Compression

This is the first adaptive sampling algorithm used in DNN compression problems, which achieves the state of the art **91.68% in terms of sparse rates on CIFAR10 using only 27K parameters (90% sparsity) with Resnet20.**

Methods	30%	50%	70%	90%
A-SGHMC	94.07	94.16	93.16	90.59
A-SGHMC-EM	94.18	94.19	93.41	91.12
SGHMC-SA	94.13	94.11	93.52	91.45
A-SGHMC-SA	**94.23**	**94.27**	**93.74**	**91.68**