Bayesian Dark Knowledge

Presenter: Longxu Dou

September 24, 2018

Overview

Abstract

- 'Bayesian Dark Knowledge' is a method unifying Stochastic Gradient Langevin Dynamics with distillation.
- SGLD is a method for learning large-scale Bayesian models like Bayesian Networks. SGLD makes it possible to avoid over-fitting.
- Distillation is a method for training student networks using soft labels created by teacher networks.

Introduction

- Contribution: Approximately learning a Bayesian neural network model while avoiding major storage costs accumulated during training and computational costs during prediction.
- Methods:
  - Training a student model to approximate a Bayesian teacher's predictions. (distillation/model compression)
  - Simultaneous online training of the teacher and student without requiring storage of samples.

Overview

Bayesian Inference

Goal: Estimate the uncertainty of model to avoid over confident, which is useful in reinforcement learning, active learning and classier fusion.

Definition

- First, compute posterior distribution over the parameters:

$$p(\theta|\mathcal{D}_N) \propto p(\theta) \prod_{i=1}^{N} p(d_i|\theta) \propto p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta) \tag{1}$$

  where $\mathcal{D}_N = (x_i, y_i)_{i=1}^N, d_i = (x_i, y_i), x_i \in \mathcal{X}^D$

- Then, compute posterior predictive distribution:

$$p(y|x, \mathcal{D}_N) = \int p(y|x, \theta) p(\theta|\mathcal{D}_N) d\theta \tag{2}$$

Why we need SGLD ?

- In general, we use SGD(MAP estimate) to compute $\hat{\theta}$ and plug-in approximation to get the predictive distribution $p(y|x, \mathcal{D}_N) \approx p(y|x, \hat{\theta})$
- However, the uncertainty of parameters is ignored. We need more accurate approximation of $p(y|x, \mathcal{D}_N)$ or $p(\theta|\mathcal{D}_N)$.
- Stochastic sampling methods such as SGLD incorporate uncertainty into predictive estimates.

SGLD

- SGLD samples θ from the posterior distributions via a Markov Chain:

$$\Delta\theta_t = \frac{\epsilon_t}{2}(\nabla_\theta \log p(\theta_t) + \frac{N}{n}\sum_{i}^{n} \nabla_\theta \log p(y_{ti}|x_{ti}, \theta_t)) + \eta_t, \eta_t \sim N(0, \epsilon_t) \tag{3}$$

- Posterior predictive distributions can be approximated via Monte Carlo approximations as:

$$q(y|x, \mathcal{D}_N) = \frac{1}{S}\sum_{s=1}^{S} p(y|x, \theta^s) \tag{4}$$

  where S is the number of samples.

- In stochastic optimization and stochastic sampling, ϵ_t needs to satisfy: $(1) 0 < \epsilon_{t+1} < \epsilon_t$ $(2) \sum_{t=1}^{\infty} \epsilon_t = \infty$ $(3) \sum_{t=1}^{\infty} \epsilon_t^2 < \infty$

Distillation

- By learning the ensembles networks or the large networks, we can get the good accuracy.
- The above networks are called teacher networks. However, the model size is large.
- After learning the teacher networks, we want to transfer the knowledge in a function into a single smaller model.
- When transferring the knowledge, it is better to use soft targets, which are created by teacher networks, instead of the original labels, i.e., hard targets.

Overview

- Bayesian Dark knowledge is a method of combining SGLD with the concept of distillation.
- SGLD is a useful method for learning Bayesian Deep Networks.
- The problem is that SGLD needs to archive many copies of parameters.
- The motivation is replacing a set of neural networks with a single deep network.

Overview

Methods

- Goal: Train a student neural network (SNN) to approximate the Bayesian predictive distribution of the teacher, which is a Monte Carlo ensemble of teacher neural network(TNN).
- SNN Objective function:
  - $p(y|x, \mathcal{D}_N)$: TNN's prediction
  - $\mathcal{S}(y|x, w)$: SNN's prediction
  - w: parameters of the student network

$$
\begin{aligned}
L(w|x) &= \mathrm{KL}(p(y|x, \mathcal{D}_N)||\mathcal{S}(y|x, w)) = -\mathbb{E}_{p(y|x,\mathcal{D}_N)} \log \mathcal{S}(y|x, w) + \text{const} \\
&= -\int \left[\int p(y|x, \theta) p(\theta|D_N) d\theta \right] \log \mathcal{S}(y|x, w) dy \\
&= -\int p(\theta|D_N) \int p(y|x, \theta) \log \mathcal{S}(y|x, w) dy \, d\theta \\
&= -\int p(\theta|D_N) \left[\mathbb{E}_{p(y|x,\theta)} \log \mathcal{S}(y|x, w) \right] d\theta
\end{aligned}
$$

Monte Carlo Approximation

- Integrate out Θ:

$$\hat{L}(w|x) = \frac{1}{|\Theta|} \sum_{\theta^s \in \Theta} \mathbb{E}_{p(y|x,\theta_s)} \log \mathcal{S}(y|x, w) \tag{5}$$

where Θ is a set of samples from $p(\theta|\mathcal{D}_N)$.

- Integrate out x:

$$\begin{aligned} \hat{L}(w) &= \int p(x) L(w|x) dx \approx \frac{1}{|\mathcal{D}'|} \sum_{x' \in \mathcal{D}'} L(w|x') \\ &\approx -\frac{1}{|\Theta|} \frac{1}{|\mathcal{D}'|} \sum_{\theta^s \in \Theta} \sum_{x' \in \mathcal{D}'} \mathbb{E}_{p(y|x',\theta^s)} \log \mathcal{S}(y|x', w) \end{aligned} \tag{6}$$

where $\mathcal{D}'$ is a set of data samples.

Algorithm

Algorithm 1: Distilled SGLD

Input: $\mathcal{D}_N = \{(x_i, y_i)\}_{i=1}^N$, minibatch size M, number of iterations T, teacher learning schedule η_t, student learning schedule ρ_t, teacher prior λ, student prior γ

for $t = 1 : T$ **do**

// Train teacher (SGLD step)

Sample minibatch indices $S \subset [1, N]$ of size M

Sample $z_t \sim \mathcal{N}(0, \eta_t I)$

Update $\theta_{t+1} := \theta_t + \frac{\eta_t}{2}\left(\nabla_\theta \log p(\theta|\lambda) + \frac{N}{M}\sum_{i\in S} \nabla_\theta \log p(y_i|x_i, \theta)\right) + z_t$

// Train student (SGD step)

Sample $\mathcal{D}'$ of size M from student data generator

$w_{t+1} := w_t - \rho_t \left(\frac{1}{M}\sum_{x'\in\mathcal{D}'} \nabla_w \hat{L}(w, \theta_{t+1}|x') + \gamma w_t\right)$

- The method does not require to archive the weights. In the distillation phase, θ is updated online.

Overview

Results on Boston housing

Method	Avg. test log likelihood
PBP (as reported in [HLA15])	-2.574 $\pm$ 0.089
VI (as reported in [HLA15])	-2.903 $\pm$ 0.071
SGD	-2.7639 $\pm$ 0.1527
SGLD	-2.306 $\pm$ 0.1205
SGLD distilled	-2.350 $\pm$ 0.0762

Table 5: Log likelihood per test example on the Boston housing dataset. We report the mean over 20 trials $\pm$ one standard error.

Results on Toy 2d classification problem

Model	Num. params.	KL
SGD	40	0.246
SGLD	40k	0.007
Distilled 2-10-2	40	0.031
Distilled 2-100-2	400	0.014
Distilled 2-10-10-2	140	0.009

Table 2: KL divergence on the 2d classification dataset.

Results on Toy 2d classification problem

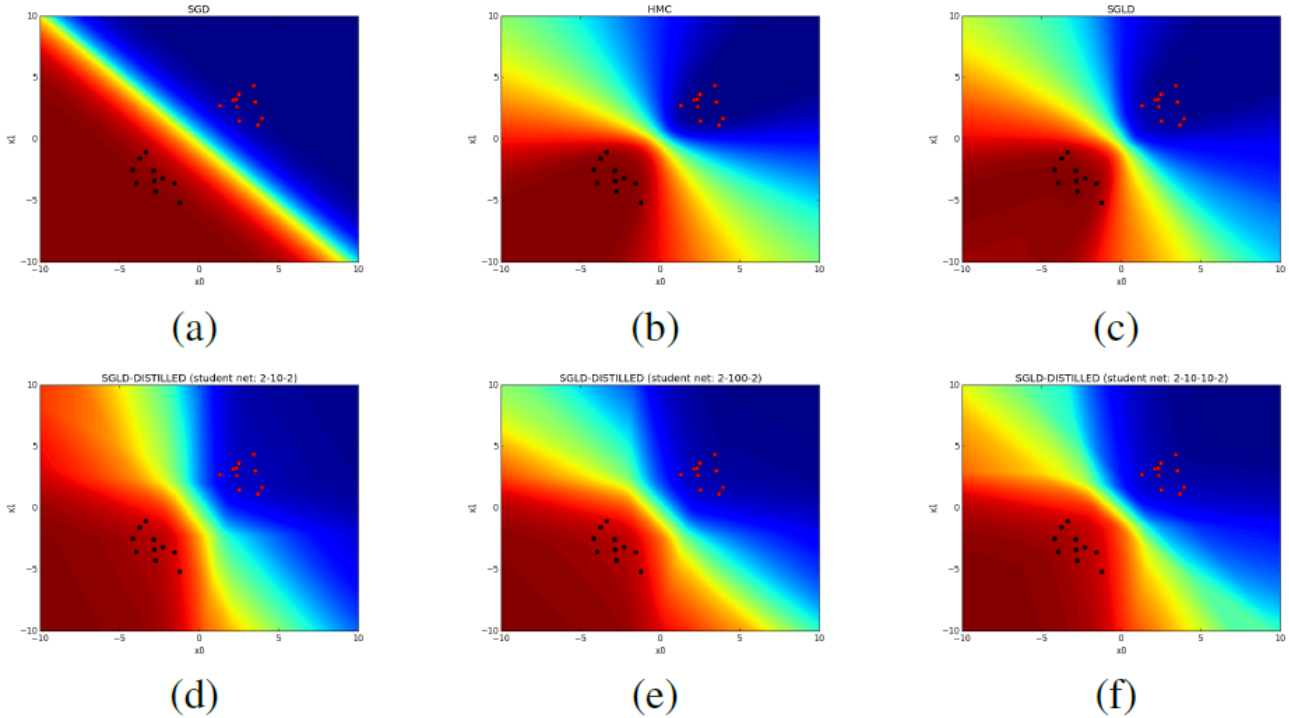

Figure 1: Posterior predictive density for various methods on the toy 2d dataset. (a) SGD (plugin) using the 2-10-2 network. (b) HMC using 20k samples. (c) SGLD using 1k samples. (d-f) Distilled SGLD using a student network with the following architectures: 2-10-2, 2-100-2 and 2-10-10-2.

Results on Toy 1d regression problem

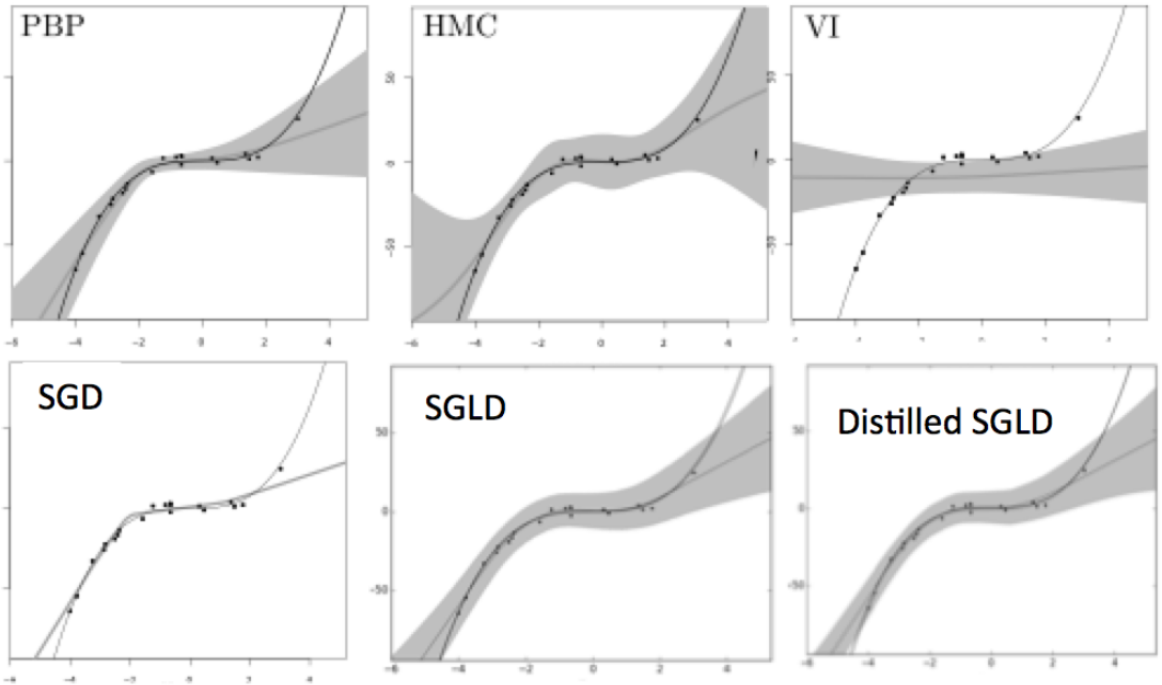

Figure 2: Predictive distribution for different methods on a toy 1d regression problem. (a) PBP of [HLA15]. (b) HMC. (c) VI method of [Gra11]. (d) SGD. (e) SGLD. (f) Distilled SGLD. Error bars denote 3 standard deviations. (Figures a-d kindly provided by the authors of [HLA15]. We replace their term "BP" (backprop) with "SGD" to avoid confusion.)

Overview

Comments VS. Rebuttal

Slow mixing rate induced by high auto-correlation

- High step size for SGLD and annealing the step size quickly for SGD.
- Generating a pool of samples using SGLD ahead of time.
- Getting samples from multiple parallel SGLD chains

Discussion about the training set $\mathcal{D}'$

For image data, we can use perturbed versions of train data, unlabeled images or adversarial examples.

Distillation phase

- Make approximate Bayesian inference at this scale trustworthy.
- Simple spherical Gaussian priors are equivalent to $L2$ regularization in Bayesian neural networks only if doing MAP estimation.

Future work

- Show the utility of this model in an end-to-end task, where predictive uncertainty is useful (such as with contextual bandits or active learning).
- Keeping a running minibatch of parameters uniformly sampled from the posterior, which can be done online using reservoir sampling.
- Exploring more intelligent data generation methods for training the student.
- Reducing the prevalence of confident false predictions on adversarial generated examples.

References

Bayesian Learning via Stochastic Gradient Langevin Dynamics

Max Welling, Yee Whye Teh (2011)

Distilling the Knowledge in a Neural Network

Geoffrey Hinton, Oriol Vinyals, Jeff Dean (2015)

Bayesian Dark Knowledge

Anoop Korattikara, Vivek Rathod, Kevin Murphy, Max Welling (2015)

Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks

Chunyuan Li, Changyou Chen, David Carlson and Lawrence Carin (2015)