SWA-Gaussian

Wesley Maddox et al., 2019

Stochastic Weight Averaging (SWA)

Izmailov et al. 2018

Idea: use the information contained in SGD trajectory, which is first proposed by SGD as Approximate Bayesian Inference (Mandt et al. 2017)

- Use a modified learning rate schedule and compute the first moment of SGD — run SGD with a constant learning rate starting from a pre-trained solution, and average the weights of the model it traverses.

$$\theta_{\text{SWA}} = \frac{1}{T}\sum_{i=1}^{T}\theta_i$$

- Optimal constant learning rate under several assumptions:

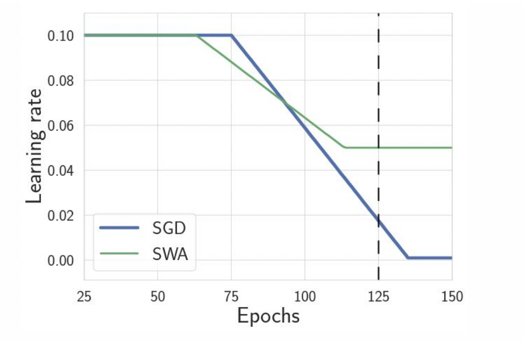
$$\epsilon^* = 2\frac{S}{N}\frac{D}{\text{Tr}(BB^\top)}$$

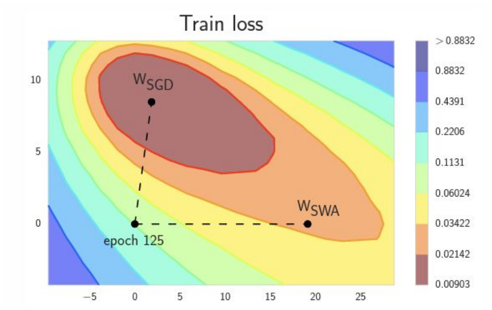
Mandt et al. 2017

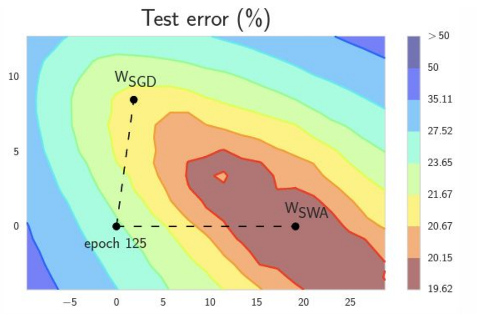
Stochastic Weight Averaging (SWA)

[Izmailov et al. 2018]

- Shown to improve the generalization in deep learning
- A high constant learning rate schedule ensures SGD to explore the set of possible solutions instead of converging to a point estimate







Motivation for SWAG

Construct an approximate posterior distribution over neural network weights, using

- SWA solution as the first moment
- a low rank + diagonal covariance

SWAG can perform well in comparison to popular methods like MC-dropout, SGLD, temperature scaling etc.

The Gaussian distribution fitted to the first two moments of SGD iterates, with a modified learning rate schedule, can capture the local geometry of the posterior well.

Problems on other Bayesian methods

General challenges in Bayesian deep learning

- Millions of parameters
- Posterior over the parameters highly non-convex
- Need mini-batch approaches to get good solutions

MCMC

- HMC - requires full gradients, which is intractable in neural networks
- Stochastic gradient MCMC: SGHMC, SGLD can asymptotically sample from the posterior with infinitely small step size. Since finite learning rate introduces errors, tuning SG-MCMC is difficult

VI

- Difficult to train on neural networks with large architectures
- Insufficient data compression
- Advances in VI for deep learning focus on smaller-scale datasets and architectures

Some non-Bayesian approaches

- *SGD based approximation, using averaged SGD as MCMC sampler*
- *Deep ensemble*
- *Temperature scaling*

SWAG - Diagonal

As one component of the covariance matrix, a simple diagonal matrix is fitted to represent the second moment of each weight, i.e. Var(theta).

Need to keep track of the 1st and 2nd moments of the SGD trajectory

$$\overline{\theta^2} = \frac{1}{T} \sum_{i=1}^{T} \theta_i^2 \qquad \Sigma_{\text{diag}} = \text{diag}(\overline{\theta^2} - \theta_{\text{SWA}}^2)$$

Gaussian approximation of the posterior:

$$q(\theta) = \mathcal{N}(\theta \mid \theta_{\mathsf{SWA}}, \Sigma_{\mathsf{diag}})$$

Full SWAG

- Diagonal covariance approximation is too restrictive.
- Extend the idea to utilize a more flexible low-rank + diagonal covariance structure

Sample covariance matrix with full rank T: $\Sigma = \frac{1}{T-1} \sum_{i=1}^{T} (\theta_i - \theta_{\text{SWA}})(\theta_i - \theta_{\text{SWA}})^\top$

No access to SWA, approximate it by averaging over the first i samples $\overline{\theta}_i = \frac{1}{i} \sum_{j=1}^{i} \theta_i, \quad D_i = \theta_i - \overline{\theta}_i$

To limit the rank from T to K, only use the last K of Di vectors from the last K epochs of training

$$\widehat{D} = [D_{T-K+1}; D_{T-K+2}; \ldots; D_T] \qquad \Sigma_{\text{low-rank}} = \frac{1}{K-1} \cdot \widehat{D}\widehat{D}^\top$$

Full SWAG

- Combining low-rank approximation and diagonal approximation, the resulting approximate posterior distribution is $\mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}))$

- Space/Memory complexity: need to store K of Di vectors, SWA mean, and theta square mean
- Sampling from SWAG

$$\widetilde{\theta} = \theta_{\text{SWA}} + \frac{1}{\sqrt{2}} \cdot \Sigma_{\text{diag}}^{\frac{1}{2}} z_1 + \frac{1}{\sqrt{2(K-1)}} \widehat{D} z_2, \quad \text{where } z_1 \sim \mathcal{N}(0, I_d),\ z_2 \sim \mathcal{N}(0, I_K)$$

Can be computed in O(Kd)

Algorithm 1 Bayesian Model Averaging with SWAG

θ_0: pretrained weights; η: learning rate; T: number of steps; c: moment update frequency; K: maximum number of columns in deviation matrix; S: number of samples in Bayesian model averaging

Train SWAG

$\overline{\theta} \leftarrow \theta_0, \ \overline{\theta^2} \leftarrow \theta_0^2$ {Initialize moments}

for $i \leftarrow 1, 2, ..., T$ **do**

$\theta_i \leftarrow \theta_{i-1} - \eta \nabla_\theta \mathcal{L}(\theta_{i-1})$ {Perform SGD update}

if $\texttt{MOD}(i, c) = 0$ **then**

$n \leftarrow i/c$ {Number of models}

$\overline{\theta} \leftarrow \dfrac{n\overline{\theta} + \theta_i}{n+1}, \ \overline{\theta^2} \leftarrow \dfrac{n\overline{\theta^2} + \theta_i^2}{n+1}$ {Moments}

if $\texttt{NUM\_COLS}(\widehat{D}) = K$ **then**

$\texttt{REMOVE\_COL}(\widehat{D}[:, 1])$

$\texttt{APPEND\_COL}(\widehat{D}, \theta_i - \overline{\theta})$ {Store deviation}

return $\theta_{\text{SWA}} = \overline{\theta}, \ \Sigma_{\text{diag}} = \overline{\theta^2} - \overline{\theta}^2, \ \widehat{D}$

Test Bayesian Model Averaging

for $i \leftarrow 1, 2, ..., S$ **do**

Draw $\widetilde{\theta}_i \sim \mathcal{N}\left(\theta_{\text{SWA}}, \frac{1}{2}\Sigma_{\text{diag}} + \frac{\widehat{D}\widehat{D}^\top}{2(K-1)}\right)$ (1)

Update batch norm statistics with new sample.

$p(y^*|\text{Data}) \mathrel{+}= \frac{1}{S} p(y^*|\widetilde{\theta}_i)$

return $p(y^*|\text{Data})$

Bayesian model averaging for inference

Given x*, y* as test inputs and outputs, marginalized over theta posterior, the unconditional predictive distribution is

$$p(y_*|\mathcal{D}, x_*) = \int p(y_*|\theta, x_*)p(\theta|\mathcal{D})d\theta$$

Using Monte Carlo in practice:

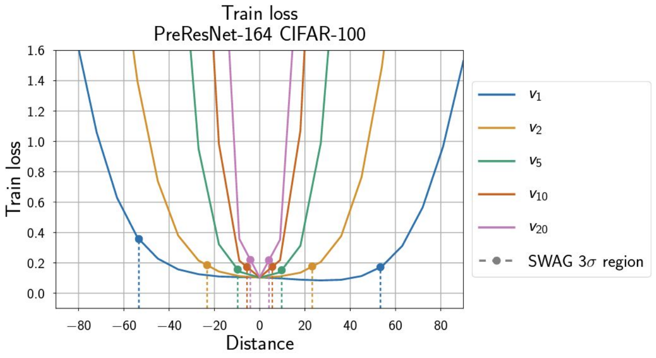
$$p(y_*|\mathcal{D}, x_*) \approx \tfrac{1}{T}\sum_{t=1}^{T} p(y_*|\theta_t, x_*)\,, \quad \theta_t \sim p(\theta|\mathcal{D})$$

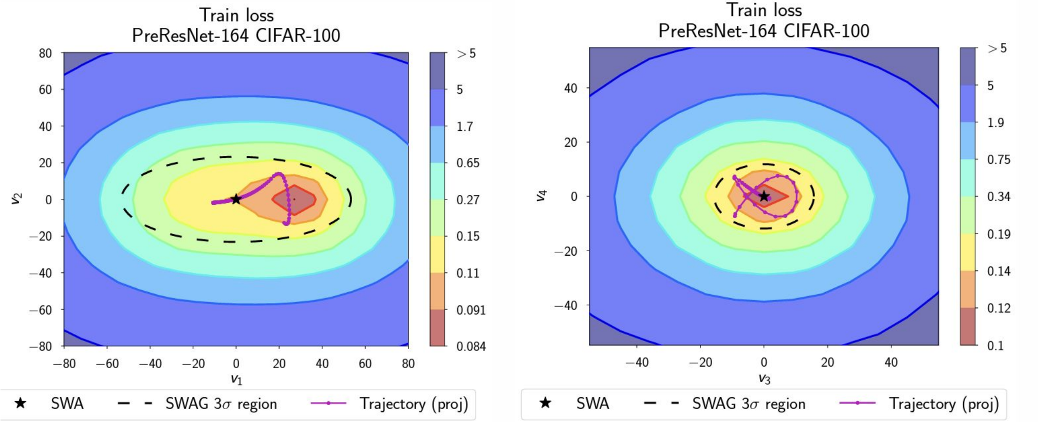
General BMA:

$$\begin{aligned} p(\mathcal{D}_{\text{test}} \mid \mathcal{D}_{\text{train}}) &= \mathbb{E}_{p(\theta|\mathcal{D}_{\text{train}})}\left[p(\mathcal{D}_{\text{test}} \mid \theta)\right] \\ &\approx \frac{1}{N}\sum_{i=1}^{N} p(\mathcal{D}_{\text{test}} \mid \widehat{\theta}_i), \quad \widehat{\theta}_i \sim q(\theta) \end{aligned}$$

Loss landscape

1D/2D loss geometry along eigenvectors of the low-rank covariance matrix





$$\phi(t) = \mathcal{L}(\theta_{\text{SWA}} + t \cdot \frac{v_i}{\|v_i\|})$$

$$\psi(t_1, t_2) = \mathcal{L}(\theta_{\text{SWA}} + t_1 \cdot \frac{v_i}{\|v_i\|} + t_2 \cdot \frac{v_j}{\|v_j\|})$$

Thanks

Model