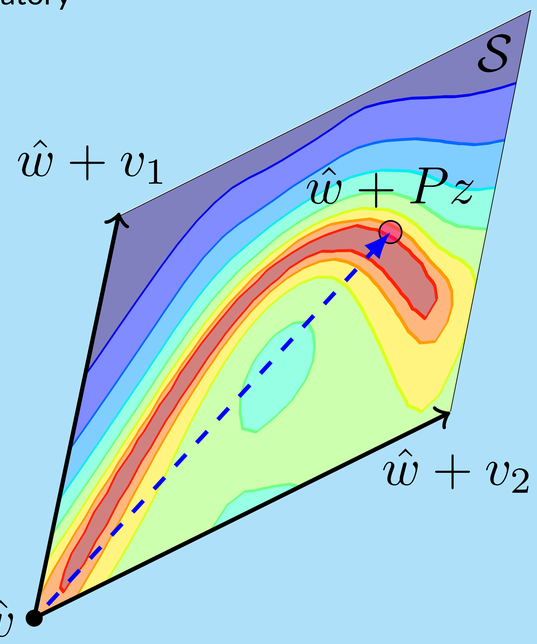
Subspace Inference for Bayesian Deep Learning

Pavel Izmailov*[1], Wesley J. Maddox*[1], Polina Kirichenko*[1]
Timur Garipov*[2], Dmitry Vetrov[3,4], Andrew Gordon Wilson[1]
[1]Cornell University, [2]Samsung AI Center in Moscow, [3]Higher School of Economics, [4]Samsung-HSE Laboratory



Challenge: Standard Bayesian inference procedure struggle with the high dimensional parameter spaces in modern deep learning.

Approach: Use information from the SGD trajectory to perform inference in a low dimensional subspace.

Even when using exceptionally **low dimensional subspaces**, Bayesian inference is possible on large **neural networks** with **minimal computational overhead.**

Take a picture to download the code.

Inference within a Subspace

Assume a set of $K+1$ vectors $\{v_1, v_2, \ldots, v_K, \hat{w}\}$ in the full weight space, $\mathbb{R}^p$; define subspace:

$$\mathcal{S} = \{w | w = \hat{w} + z_1 v_1 + \ldots z_K v_K\} = \{w | w = \hat{w} + Pz\},$$

with $\hat{w} \in \mathbb{R}^p$, $P = (v_1^T, \ldots, v_K^T) \in \mathbb{R}^{p \times K}$, and $z = (z_1, \ldots, z_K)^T \in \mathbb{R}^K$.
New likelihood is a function of z:

$$p(\mathcal{D}|z) = p_{\mathcal{M}}(\mathcal{D}|w = \hat{w} + Pz).$$

Bayesian model averaging:

$$p(\mathcal{D}^*|\mathcal{D}) = \frac{1}{M}\sum_i p_{\mathcal{M}}(\mathcal{D}^*|\tilde{w} = \hat{w} + P\tilde{z}_i), \quad \tilde{z}_i \sim q(z|\mathcal{D}),$$

where $q(z|\mathcal{D})$ is approximate posterior over z, represented by MCMC samples or a deterministic (variational) approach.

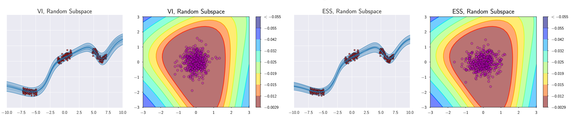
Posterior Tempering. #parameters << #data points in the subspace model, hence posterior over z is extremely concentrated. Instead, we utilize the *tempered* posterior:

$$p_T(z|\mathcal{D}) \propto \underbrace{p(\mathcal{D}|z)^{1/T}}_{\text{likelihood}} \underbrace{p(z)}_{\text{prior}}$$

Subspace Construction

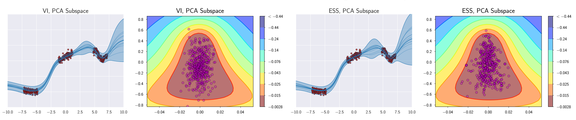
Intuitively we want the subspace $\mathcal{S}$ to (1) contain a diverse (producing meaningfully different predictions on test data) set of models and (2) be cheap to construct.

Random Subspaces



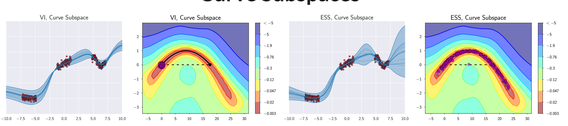
- Directions $v_1, \ldots, v_K \sim \mathcal{N}(0, I_p)$
- Use a pre-trained solution as shift $\hat{w}$

PCA of the SGD Trajectory



- Run SGD with high constant learning rate from a pre-trained solution and collect snapshots w_i of weights
- Use SWA solution as shift $\hat{w} = \frac{1}{T}\sum w_i$
- $\{v_1, v_2, \ldots, v_k\}$ – first K PCA components of vectors $w_i - \hat{w}$

Curve Subspaces



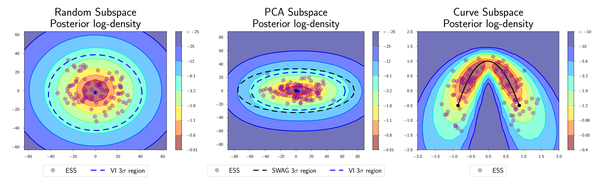
- Garipov et al. 2018 proposed a method to find two-dimensional subspaces containing path of low loss between weights of two independently trained neural networks

Experiments

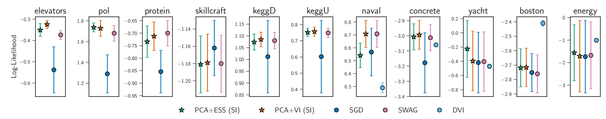
- Curve Subspace > PCA Subspace > Random Subspace
- PCA subspace generally has the best run-time accuracy trade-off.
- Despite its simplicity, Subspace Inference in the PCA subspace is competitive with many popular alternatives: SWAG, MC-Dropout and Temperature Scaling on image classification and UCI regression data.

Negative log-likelihood and Accuracy for PreResNet-164 for 10-dimensional random, 10-dimensional PCA, and 2-dimensional curve subspaces.

	Random	PCA	Curve
NLL	0.6858 ± 0.0052	0.6652 ± 0.004	0.6464 ± 0.01
Accuracy (%)	80.17 ± 0.03	80.54 ± 0.13	81.28 ± 0.26



Posterior log-density surfaces, ESS samples, and VI approximate posterior distribution in **(a)** random, **(b)** PCA and **(c)** curve subspaces for PreResNet-164 on CIFAR-100.



Test log-likelihoods for proposed methods on six UCI regression datasets.

Coverage of 95% prediction interval for proposed methods on UCI regression datasets.

References

Garipov, Timur, et al. "Loss surfaces, mode connectivity, and fast ensembling of DNNs." NeurIPS. 2018.
Izmailov, Pavel, et al. "Averaging weights leads to wider optima and better generalization." UAI. 2018.
Maddox, Wesley, et al. "A simple baseline for Bayesian uncertainty in deep learning." arXiv:1902.02476. 2019).

Cornell University