Sparse Variational Dropout: a New Method for DNN Sparsification

The regularizer favors large dropout rates $\alpha_{ij}$

$$D_{\text{KL}}(q(\tilde{w}_{ij})|w_{ij}, \alpha_{ij}) \propto p(\tilde{w}_{ij})$$

infinitely large dropout rate ($\alpha_{ij} \rightarrow \infty$) means:

- Equivalent binary dropout rate $p = \frac{1}{\alpha_{ij}} \rightarrow 1$, so $\tilde{w}_{ij} = 0$ during training.
- Data-term controls the accuracy and prohibits to set all $\alpha$’s to infinity.

Variance Reduction 1: Additive Noise Parameterization

Optimize the loss $\mathcal{L}$ w.r.t. $\tilde{w}_{ij}$ and $\sigma_{ij}^2 = \alpha_{ij} w_{ij}^2$

$$\frac{\partial \mathcal{L}}{\partial \tilde{w}_{ij}} = \frac{\partial \mathcal{L}}{\partial \sigma_{ij}^2} = \frac{\partial \tilde{w}_{ij}}{\partial \sigma_{ij}^2}$$

Before: $\tilde{w}_{ij} = w_{ij}(1 + \sqrt{\sigma_{ij}^2})$

After: $\tilde{w}_{ij} = w_{ij} + \sigma_{ij}^2$

Variance Reduction 2: Sample activations instead of weights (LRJ [1, 2])

Before: $B = AW$

After: $B \sim q(B)$

Experiments: MNIST LeNet-5

Comparison of different sparsity-inducing techniques on the LeNet-5 architecture. Our method provides the highest level of sparsity with a similar accuracy.

Experiments: VGG-like on CIFAR-10 and CIFAR-100

(a) Results on the CIFAR-10 dataset.

(b) Results on the CIFAR-100 dataset.

Key results

Sparse Variational Dropout: a New Method for DNN Sparsification

- Compression of DNNs through sparsification of weight matrices
- CIFAR-10 VGG: up to 70x compression
- MNIST LeNet-5: up to 280x compression
- No accuracy loss!

Sparse Variational Dropout Training

- Dropout training optimizes the cross-entropy loss under stochastic setting:

$$-E_{w\sim \mathcal{N}(0,1)} \log p(Y|X, W) \rightarrow \min_{W} \mathcal{L}(W) \rightarrow \min_{W}$$

Stochastic Variational Inference

$$E_{q(w_{ij})} \log p(Y|X, W) + D_{KL}(q(\tilde{w}_{ij})|w_{ij}, \alpha_{ij}) \rightarrow \min_{w_{ij}}$$

With the true posterior distribution over weights $W$ approximated by $q(\tilde{w}_{ij})$.

$$\tilde{w}_{ij} \sim \mathcal{N}(w_{ij}, \alpha_{ij})$$

Just a slightly different loss function; implementation is basically the same.

Sparse Variational Dropout

- Gaussian Dropout posterior distribution

$$\tilde{W} = W \circ \varepsilon \quad \text{with} \quad \tilde{w}_{ij} \sim \mathcal{N}(w_{ij}, \alpha_{ij} w_{ij}^2) \quad = \quad q(\tilde{w}_{ij}|w_{ij}, \alpha_{ij})$$

- Sparsity-inducing log-uniform prior [2] favors large dropout rates

$$p(w_{ij}) \propto \frac{1}{|w_{ij}|}$$

- Now we can optimize w.r.t both weights $w_{ij}$ and dropout rates $\alpha_{ij}$

Follow-up papers

- Bayesian Sparsification of Recurrent Neural Networks

- Structured Bayesian Pruning via Log-N Multiplicative Noise

Links and references

3. Song Han et al. Deep Compression: Compressing DNNs with Pruning, Trained Quantization and Huffman Coding, 2016
6. Chiyuan Zhang et al. Understanding deep learning requires rethinking generalization, 2017