Bregman Lifted Network

Xiaoyu Wang [1] Martin Benning [2]

[1]Department of Applied Mathematics and Theoretical Physics, University of Cambridge [2]School of Mathematical Sciences, Queen Mary University of London

Contribution of this work

- In this work we introduce a novel lifted approach for training deep neural networks based on a formulation with Bregman loss terms.
- The advantage of this new formulation is that its partial derivatives with respect to the network's individual weight- and bias-terms do not require the computation of derivates of the individual activation functions.
- Our proposed Bregman lifted approach avoids limitations of the classical lifted training approach and further improves learning performances.

Lifted Neural Network

A feed-forward neural network model of L layers can be written in the nested form:

$$f(x) = W_L^\top(\sigma_{L-1}(W_{L-1}^\top \dots \sigma_l(W_l^\top \dots \sigma_1(W_1^\top x))))$$

Learning the parameters of such a neural network model can be formulated as an empirical risk minimisation problem, where the hidden layers are defined recursively via the non-linear constraints:

$$\min_{\{W\}_{l=1}^L,\{X\}_{l=1}^{L-1}} \sum_i^s \ell(y^i, W_L^\top x_{L-1}^i) + \sum_{l=1}^L \text{s.t. } x_l^i = \sigma_l(W_l^\top x_{l-1}^i) \text{ for } l = 1, 2, \dots, L-1\,.$$

- ℓ is the data function on the last layer output chosen a-priori .

Learning via stochastic (sub)gradient descent in combination with back-propagation often suffers from the following limitations:

- vanishing or exploding gradient problems during training slow down convergence
- back-propagation does not easily allow parallelization over layers

Lifted approach on the other hand replaces the non-linear constraints with penalties in the learning objective:

$$\min_{\{W\}_{l=1}^L,\{X\}_{l=1}^{L-1}} \sum_{i=1}^s \ell(y^i, W_L^\top x_{L-1}) + \sum_{l=1}^L \frac{\gamma}{2}\|x_l^i - W_l^\top x_{l-1}^i\|^2 \quad (1)$$
$$\text{s.t. } x_l^i \in C \text{ for } l = 1, 2, \dots, L-1\,.$$

where γ controls the non-linear constraint relaxation and C denotes convex constraint sets for the activated variables x_l^i, e.g. $x_l^i \in \{v \in \mathbb{R}^n | v \geq 0\}$ if applies ReLU activation.

Proximal activation function

The proximal map $\sigma : \mathbb{R}^n \to \text{dom}(\Psi) \subset \mathbb{R}^n$ of a proper, lower semi-continuous and convex function $\Psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is defined as

$$\sigma(z) := \arg\min_{u \in \mathbb{R}^n} \left\{ \frac{1}{2}\|u - z\|^2 + \Psi(u) \right\}.$$

Example

The rectifier or ramp function can be interpreted as the proximal map of the characteristic function over the non-negative orthant:

$$\Psi(u) := \begin{cases} 0 & u \in [0, \infty)^n \\ \infty & \text{otherwise} \end{cases} \quad \Longrightarrow \quad \sigma(z)_j = \max(0, z_j)\,,\ \forall j \in \{1, \dots, n\}$$

Bregman Lifted Network

Alternatively, our proposed Bregman lifted approach considers minimising the learning objective $E(W; X)$, which is defined as:

$$E(W; X) = \sum_{i=1}^s \left[\ell(y^i, W_L^\top x_{L-1}^i) + \sum_{l=1}^L L_\Psi(x_l^i, W_l^\top x_{l-1}^i) \right] \quad (2)$$

where we relax the non-linear constraints with Bregman loss term L_Ψ.

Based on the definition of the generalised Bregman distance and the assumption $y \in \text{dom}(\Psi)$, L_Ψ is defined as:

$$L_\Psi(y, z) := \frac{1}{2}\|y - \sigma(z)\|^2 + D_\Psi^{z-\sigma(z)}(y, \sigma(z))\,, \quad (3)$$

for the (valid) subgradient $z - \sigma(z) \in \partial\Psi(\sigma(z))$ and D_Ψ denotes the generalised Bregman distance with respect to Ψ.

References

[1] Armin Askari, Geoffrey Negiar, Rajiv Sambharya, and Laurent El Ghaoui. Lifted neural networks, 2018.

[2] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200--217, 1967.

[3] Jean Jacques Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes Rendus de l'Académie des Sciences de Paris*, A255(22), November 1962.

[4] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[5] Christopher Zach and Virginia Estellers. Contrastive learning for lifted networks, 2019.

Classical Lifted Network vs Bregman Lifted Network

Consider the case of a single training sample pair (x^i, y^i), the optimality system for weight parameters W_l in the classical lifted approach is $W_l^\top x_{l-1} = x_l$ for non-vanishing activated inputs.

- This suggests that the non-linear activation layers behave linearly.
- The classical lifted model learns by construction, linear transformations such that the post-activations are pushed towards the direction of staying in the constraint sets.

The optimality system for our proposed Bregman lifted approach on the other hand writes:

$$\nabla_{W_l} E(W; X) = 0 \iff (\sigma_l(W_l^\top x_{l-1}) - x_l)x_{l-1}^\top = 0$$
$$\iff \forall j, j' : x_{l,j} = 0 \vee \sigma_l(W_l^\top x_{l-1})_{j'} = x_{l,j'}.$$

- This shows that our Bregman lifted formulation recovers the true action of non-linear activations of the neural network.
- Our approach updates weight parameters with a larger degree of freedom and hence further improves the classical lifted approach.

Application and Numerical results

- We consider a four-layer network: 784-64-64-64-10 and use ReLU activation function across all hidden layers.
- Tables below show train and test classification accuracy on the MNIST and Fashion-MNIST dataset respectively against classical lifted network training approach.

Model	Train	Test
Back-prop	99.8%	97.7%
Standard Lifted Network	85.2%	86.3%
Bregman Lifted Network	99.3%	96.7%

Table 1:Classification accuracy on the MNIST dataset.

Model	Train	Test
Back-prop	95.6%	88.0%
Standard Lifted Network	81.4%	80.0%
Bregman Lifted Network	93.5%	85.7%

Table 2:Classification accuracy on Fashion-MNIST dataset.

We also evaluate the percentage of linear activations by computing the percentage of nodes in the hidden layer that perform linearly.

Model	Layer 1	Layer 2	Layer 3
Back-prop	38.0%	39.6%	38.4%
Standard Lifted Network	99.9%	99.9%	99.9%
Bregman Lifted Network	50.6%	12.5%	25.4%

Table 3:Linear activation percentage trained on Fashion-MNIST dataset