Demystifying Batch Normalization in ReLU Networks: Equivalent Convex Optimization Models and Implicit Regularization

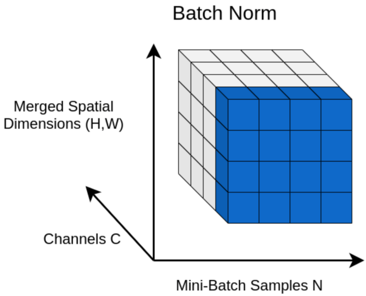
Tolga Ergen*, Arda Sahiner*, Batu Ozturkler, John Pauly, Morteza Mardani, Mert Pilanci

@tolgaergen_ stanford.edu/∼ergen/ ergen@stanford.edu

Why do we need Batch Normalization (BN)?

BN transforms a batch of data to zero mean and standard deviation one, and has two trainable parameters α, γ:

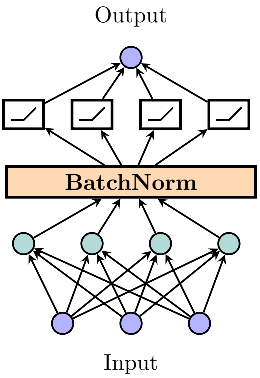
$$\mathrm{BN}_{\gamma,\alpha}(\mathrm{x}) = \frac{(\mathrm{I_d} - \frac{1}{d}\mathbf{1}\mathbf{1}^T)\mathrm{x}}{\|(\mathrm{I_d} - \frac{1}{d}\mathbf{1}\mathbf{1}^T)\mathrm{x}\|_2}\gamma + \alpha$$



BN stabilizes and accelerates training of deep neural networks

Two-layer Neural Networks with ReLU Activation

Model:



Notation:

$X \in \mathbb{R}^{n\times d}$: Data matrix

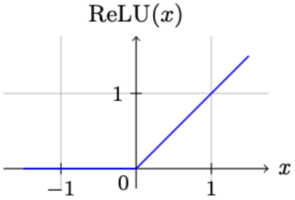
$Y \in \mathbb{R}^{n\times C}$: Label matrix

$\mathcal{L}(\cdot,\cdot)$: Arbitrary convex loss function

$\beta > 0$: Regularization coefficient

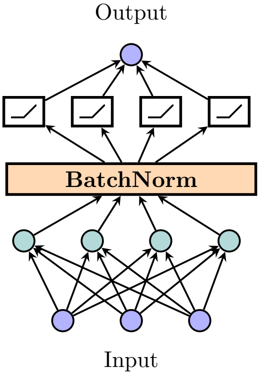
$W^{(1)} \in \mathbb{R}^{d\times m}, W^{(2)} \in \mathbb{R}^{m\times C}$: Layer weights

$\boldsymbol{\theta}$: Represents all trainable parameters



Two-layer Neural Networks with ReLU Activation

Model:



Notation:

$X \in \mathbb{R}^{n \times d}$: Data matrix

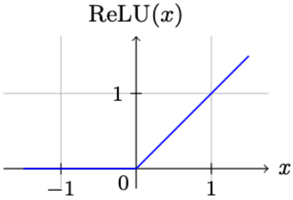
$Y \in \mathbb{R}^{n \times C}$: Label matrix

$\mathcal{L}(\cdot,\cdot)$: Arbitrary convex loss function

$\beta > 0$: Regularization coefficient

$W^{(1)} \in \mathbb{R}^{d \times m}, W^{(2)} \in \mathbb{R}^{m \times C}$: Layer weights

$\boldsymbol{\theta}$: Represents all trainable parameters



Optimization problem:

$$p_{non-convex} := \min_{\boldsymbol{\theta}} \mathcal{L}\left(\phi\left(\mathrm{BN}_{\gamma,\alpha}\left(XW^{(1)}\right)\right)W^{(2)}, Y\right) + \frac{\beta}{2}\|\boldsymbol{\theta}\|_2^2$$

where $\phi(x) = \mathrm{ReLU}(x) = (x)_+$ and $\mathcal{L}(\cdot,\cdot)$ is any convex loss

Two-layer Neural Networks with ReLU Activation

- High-dimensional regime ($n \leq d$)

Theorem

Let $n \leq d$ and $\mathbf{X}$ is full row-rank, then an optimal solution is

$$\left(\mathbf{w}_j^{(1)^*}, \mathbf{w}_j^{(2)^*}\right) = \left(\mathbf{X}^\dagger \mathbf{y}_j, \left(\|\mathbf{y}_j\|_2 - \beta\right)_+ \mathbf{e}_j\right)$$

$$(\gamma_j^*, \alpha_j^*) = \left(\frac{\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{y}_j\|_2}{\|\mathbf{y}_j\|_2}, \frac{\mathbf{1}^T\mathbf{y}_j}{\sqrt{n}\|\mathbf{y}_j\|_2}\right), \ \forall j \in [C]$$

where $\mathbf{e}_j$ is the j^{th} ordinary basis vector.

Two-layer Neural Networks with ReLU Activation

- High-dimensional regime ($n \leq d$)

Theorem

Let $n \leq d$ and $\mathbf{X}$ is full row-rank, then an optimal solution is

$$\left(\mathbf{w}_j^{(1)^*}, \mathbf{w}_j^{(2)^*}\right) = \left(\mathbf{X}^\dagger \mathbf{y}_j, \left(\|\mathbf{y}_j\|_2 - \beta\right)_+ \mathbf{e}_j\right)$$
$$(\gamma_j{}^*, \alpha_j{}^*) = \left(\frac{\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{y}_j\|_2}{\|\mathbf{y}_j\|_2}, \frac{\mathbf{1}^T\mathbf{y}_j}{\sqrt{n}\|\mathbf{y}_j\|_2}\right), \ \forall j \in [C]$$

where $\mathbf{e}_j$ is the j^{th} ordinary basis vector.

- Generic case (arbitrary n, d):

Theorem

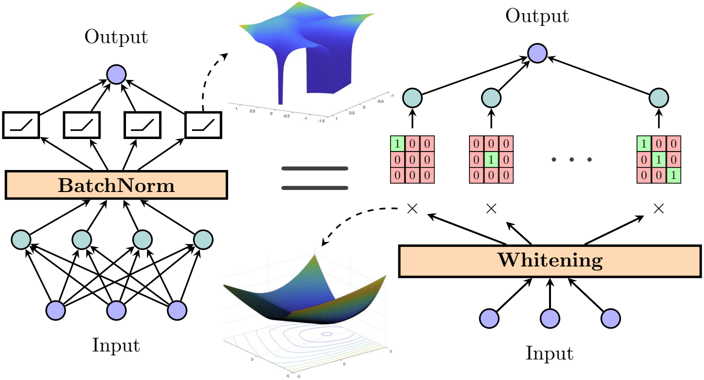
Given the SVD of $\mathbf{X}$ as $\mathbf{X} := \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, the non-convex training problem is equivalent to

$$p_{convex} := \min_{\mathbf{S}_i} \mathcal{L}\left(\sum_{i=1}^{P} \mathbf{D}_i \mathbf{U}\mathbf{S}_i, \mathbf{Y}\right) + \beta \sum_{i=1}^{P} \|\mathbf{S}_i\|_{c_i,*}$$

where $\mathbf{D}_1, \ldots, \mathbf{D}_P$ are fixed diagonal matrices, and $\|\cdot\|_{c_i,*}$ is a constrained version of the standard nuclear norm.

ReLU+BN $\equiv$ Convex+Low-rank+Whitening

Non-convex | Convex



$$X \in \mathbb{R}^{n\times d} \xRightarrow[\text{ReLU+BN}]{} \tilde{X} = \begin{bmatrix} D_1 U & \dots & D_P U \end{bmatrix} \in \mathbb{R}^{n\times dP}$$

ReLU+BN $\equiv$ Low-rank convex model applied to whitened data $\tilde{X}$

Deep ReLU Networks

Model: $f_{\theta,L}(\mathsf{X}) := \mathsf{A}^{(L-1)}\mathsf{W}^{(L)}, \quad \text{where } \mathsf{A}^{(l)} := \left(\mathrm{BN}_{\gamma,\alpha}\left(\mathsf{A}^{(l-1)}\mathsf{W}^{(l)}\right)\right)_+$

Theorem

Assume the network is overparameterized s.t. $\mathrm{Range}(\mathsf{A}^{(L-2)}) = \mathbb{R}^n$*, then optimal solution in closed-form is as follows*

$$\begin{aligned}\left(\mathbf{w}_j^{(L-1)^*}, \mathbf{w}_j^{(L)^*}\right) &= \left(\mathsf{A}^{(L-2)^\dagger}\mathbf{y}_j, \left(\|\mathbf{y}_j\|_2 - \beta\right)_+ \mathbf{e}_j\right) \\ \left(\gamma_j^{(L-1)^*}, \alpha_j^{(L-1)^*}\right) &= \left(\frac{\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{y}_j\|_2}{\|\mathbf{y}_j\|_2}, \frac{\mathbf{1}^T\mathbf{y}_j}{\sqrt{n}\|\mathbf{y}_j\|_2}\right)\end{aligned}, \quad \forall j \in [C]$$

where $\mathbf{e}_j$ *is the* j^{th} *ordinary basis vector.*

Deep ReLU Networks

Model: $f_{\theta,L}(\mathsf{X}) := \mathsf{A}^{(L-1)}\mathsf{W}^{(L)}$, where $\mathsf{A}^{(l)} := \left(\mathrm{BN}_{\gamma,\alpha}\left(\mathsf{A}^{(l-1)}\mathsf{W}^{(l)}\right)\right)_+$

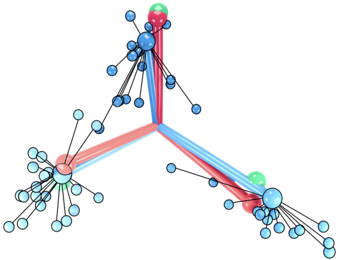
Theorem

Assume the network is overparameterized s.t. $\mathrm{Range}(\mathsf{A}^{(L-2)}) = \mathbb{R}^n$*, then optimal solution in closed-form is as follows*

$$\begin{aligned}\left(\mathbf{w}_j^{(L-1)*}, \mathbf{w}_j^{(L)*}\right) &= \left(\mathsf{A}^{(L-2)\dagger}\mathbf{y}_j, \left(\|\mathbf{y}_j\|_2 - \beta\right)_+ \mathbf{e}_j\right) \\ \left(\gamma_j^{(L-1)*}, \alpha_j^{(L-1)*}\right) &= \left(\frac{\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{y}_j\|_2}{\|\mathbf{y}_j\|_2}, \frac{\mathbf{1}^T\mathbf{y}_j}{\sqrt{n}\|\mathbf{y}_j\|_2}\right)\end{aligned}, \quad \forall j \in [C]$$

where $\mathbf{e}_j$ *is the* j^{th} *ordinary basis vector.*

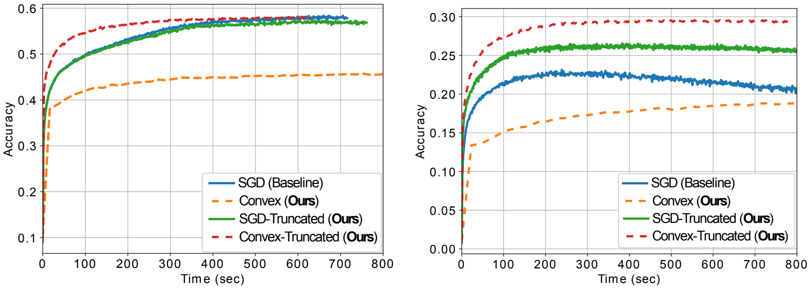
This also explains Neural Collapse in (Papyan et al., 2020)



Truncation: Given the SVD $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, the truncated data matrix is defined as

$$\hat{\mathbf{X}} := \mathbf{U}\mathcal{T}(\boldsymbol{\Sigma})\mathbf{V}^\top,$$

with $\mathcal{T}_k(\boldsymbol{\Sigma})_{ii} := \sigma_i \mathbb{1}\{i \geq k\}$, which takes the k top singular values.



(a) CIFAR-10 (b) CIFAR-100

Figure 1: Image classification on CIFAR datasets

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension
  - convex optimization theory and solvers can be applied

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension
  - convex optimization theory and solvers can be applied
  - don’t need heuristics or hyperparameter search, e.g., learning rate and initialization

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- BN networks seek low-rank structure via nuclear norm regularization

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- BN networks seek low-rank structure via nuclear norm regularization
- BN induces **whitening** effect on the data

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- BN networks seek low-rank structure via nuclear norm regularization
- BN induces **whitening** effect on the data

Future research directions:

- convex programs for deeper networks (more than 2 layers)

Takeaways and Open Problems

- ReLU network training with BN is a convex optimization problem in high dimension
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- BN networks seek low-rank structure via nuclear norm regularization
- BN induces **whitening** effect on the data

Future research directions:

- convex programs for deeper networks (more than 2 layers)
- extensions to various architectures such as GAN, RNN, ResNets ...