Global Optimality Beyond Two Layers: Training Deep ReLU Networks via Convex Programs

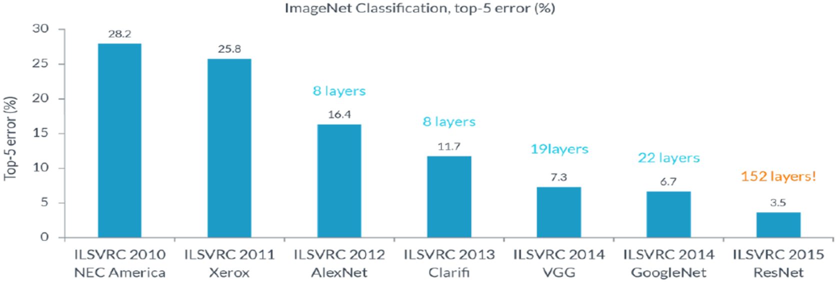
ICML 2021

Tolga Ergen & Mert Pilanci

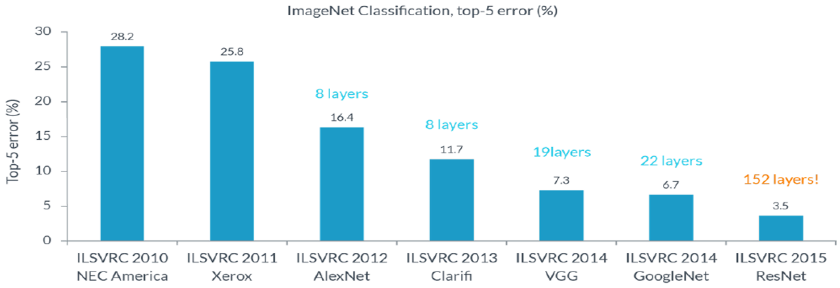
July 19, 2021

Stanford University

Deep Learning Revolution



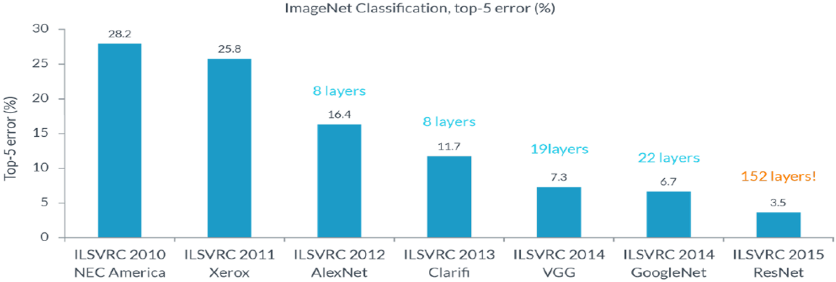
Deep Learning Revolution



Deep learning models:

- often provide the best performance due to their large capacity
    - challenging to train

Deep Learning Revolution

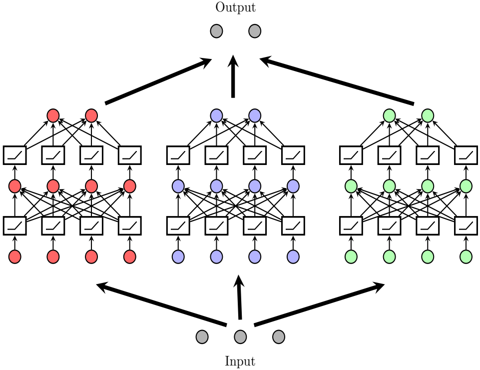


Deep learning models:

- often provide the best performance due to their large capacity
  - challenging to train
- are complex black-box systems based on non-convex optimization
  - hard to interpret what the model is actually learning

Problem Formulation

Model:



Notation:

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

$\mathbf{y} \in \mathbb{R}^{n}$: Label vector

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

$\mathcal{R}(\cdot)$: Regularization function

$\beta > 0$: Regularization coefficient

θ : All parameters

l and k : Layer and sub-network indices

$\mathbf{W}_{lk} \in \mathbb{R}^{m_{l-1} \times m_l}$: Weights

$f_{\theta,k}(\mathbf{X}) := \left((\mathbf{X}\mathbf{W}_{1k})_+ \ldots \mathbf{w}_{(L-1)k} \right)_+ w_{Lk}$

Optimization problem:

$$\min_{\theta} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y} \right) + \beta \sum_{k=1}^{K} \mathcal{R}_k(\theta)$$

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
  - require all local minima to be rank-deficient

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
  - require all local minima to be rank-deficient
  - not valid for common regularization such as weight decay

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
  - require all local minima to be rank-deficient
  - not valid for common regularization such as weight decay
  - # of sub-networks (K) needs to be too large

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
  - require all local minima to be rank-deficient
  - not valid for common regularization such as weight decay
  - $\#$ of sub-networks (K) needs to be too large
- (Zhang et al., 2019) proved strong duality for deep linear networks
  - valid only for hinge loss and linear networks

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
    - require all local minima to be rank-deficient
    - not valid for common regularization such as weight decay
    - # of sub-networks (K) needs to be too large
- (Zhang et al., 2019) proved strong duality for deep linear networks
    - valid only for hinge loss and linear networks
    - require the data matrix to be included in the regularization(thus, not valid for weight decay)

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
  - require all local minima to be rank-deficient
  - not valid for common regularization such as weight decay
  - # of sub-networks (K) needs to be too large
- (Zhang et al., 2019) proved strong duality for deep linear networks
  - valid only for hinge loss and linear networks
  - require the data matrix to be included in the regularization(thus, not valid for weight decay)
  - require assumptions on the regularization parameter β

Prior Work

- (Haeffele and Vidal, 2017) provided conditions to guarantee that each local minimum is a global optimum
  - require all local minima to be rank-deficient
  - not valid for common regularization such as weight decay
  - $\#$ of sub-networks (K) needs to be too large
- (Zhang et al., 2019) proved strong duality for deep linear networks
  - valid only for hinge loss and linear networks
  - require the data matrix to be included in the regularization(thus, not valid for weight decay)
  - require assumptions on the regularization parameter β
- (Pilanci and Ergen, 2020) introduced convex representations for ReLU networks
  - valid only for two-layer networks

Convex Duality for Deep Neural Networks

Lemma

The following problems are equivalent

$$P^* := \min_{\theta \in \Theta} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \frac{\beta}{2}\sum_{k=1}^{K}\sum_{l=L-1}^{L} \|\mathbf{W}_{lk}\|_F^2 = \min_{\theta \in \Theta_p} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \beta \sum_{k=1}^{K} |w_{Lk}|,$$

where $\Theta_p := \{\theta \in \Theta : \|\mathbf{W}_{lk}\|_F \leq 1, \forall l \in [L-2],\ \|\mathbf{w}_{(L-1)k}\|_2 \leq 1, \forall k\}.$

Convex Duality for Deep Neural Networks

Lemma

The following problems are equivalent

$$P^* := \min_{\theta \in \Theta} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \frac{\beta}{2} \sum_{k=1}^{K} \sum_{l=L-1}^{L} \|\mathbf{W}_{lk}\|_F^2 = \min_{\theta \in \Theta_p} \mathcal{L}\left(\sum_{k=1}^{K} f_{\theta,k}(\mathbf{X}), \mathbf{y}\right) + \beta \sum_{k=1}^{K} |w_{Lk}|,$$

where $\Theta_p := \{\theta \in \Theta : \|\mathbf{W}_{lk}\|_F \leq 1, \forall l \in [L-2], \; \|\mathbf{w}_{(L-1)k}\|_2 \leq 1, \forall k\}.$

Dual problem with respect to w_{Lk}**:**

$$P^* \geq D^* := \max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \max_{\theta \in \Theta_p} \left| \mathbf{v}^T \left((\mathbf{X}\mathbf{W}_1)_+ \ldots \mathbf{w}_{(L-1)} \right)_+ \right| \leq \beta,$$

where $\mathcal{L}^*$ is the Fenchel conjugate function

$$\mathcal{L}^*(\mathbf{v}) := \max_{\mathbf{z}} \mathbf{z}^T \mathbf{v} - \mathcal{L}(\mathbf{z}, \mathbf{y})$$

Our contribution: We first prove strong duality, i.e., $P^* = D^*$ and then derive convex formulations

Convex Program for Three-layer Networks

Theorem

The non-convex training problem can be equivalently stated as

$$\min_{\mathbf{w},\mathbf{w}' \in \mathcal{C}} \frac{1}{2} \left\| \tilde{\mathbf{X}} \left(\mathbf{w}' - \mathbf{w} \right) - \mathbf{y} \right\|_2^2 + \beta \left(\|\mathbf{w}\|_{2,1} + \|\mathbf{w}'\|_{2,1} \right)$$

where $\| \cdot \|_{2,1}$ *is* d *dimensional group norm:* $\|\mathbf{w}\|_{2,1} := \sum_{i=1}^{P} \|\mathbf{w}_i\|_2$

$$\tilde{\mathbf{X}} := \begin{bmatrix} \tilde{\mathbf{X}}_s & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{X}}_s \end{bmatrix}, \quad \tilde{\mathbf{X}}_s := \begin{bmatrix} \mathbf{D}_1 \mathbf{X} & \mathbf{D}_2 \mathbf{X} & \dots & \mathbf{D}_P \mathbf{X} \end{bmatrix}.$$

Convex Program for Three-layer Networks

Theorem

The non-convex training problem can be equivalently stated as

$$\min_{\mathbf{w},\mathbf{w}' \in \mathcal{C}} \frac{1}{2} \left\| \tilde{\mathbf{X}} \left(\mathbf{w}' - \mathbf{w} \right) - \mathbf{y} \right\|_2^2 + \beta \left(\|\mathbf{w}\|_{2,1} + \|\mathbf{w}'\|_{2,1} \right)$$

where $\| \cdot \|_{2,1}$ *is* d *dimensional group norm:* $\|\mathbf{w}\|_{2,1} := \sum_{i=1}^{P} \|\mathbf{w}_i\|_2$

$$\tilde{\mathbf{X}} := \begin{bmatrix} \tilde{\mathbf{X}}_s & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{X}}_s \end{bmatrix}, \quad \tilde{\mathbf{X}}_s := \begin{bmatrix} \mathbf{D}_1 \mathbf{X} & \mathbf{D}_2 \mathbf{X} & \dots & \mathbf{D}_P \mathbf{X} \end{bmatrix}.$$

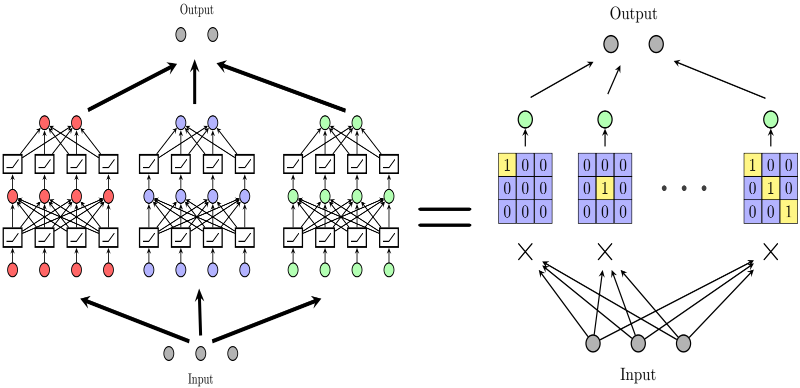
Diagonal matrices (D):

$$(\mathbf{X}\mathbf{w})_+ = \mathbf{D}\mathbf{X}\mathbf{w} \iff \begin{matrix} \mathbf{D}\mathbf{X}\mathbf{w} \geq 0 \\ (\mathbf{I}_n - \mathbf{D})\mathbf{X}\mathbf{w} \leq 0 \end{matrix} \iff (2\mathbf{D} - \mathbf{I}_n)\mathbf{X}\mathbf{w} \geq 0,$$

where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix of zeros and ones, i.e., $\mathbf{D}_{ii} \in \{0, 1\}$

Training Complexity

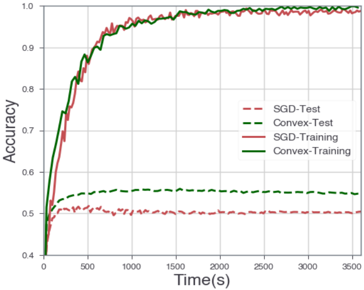
Architecture with three sub-networks ($K = 3$) and ReLU layers ($L = 3$):



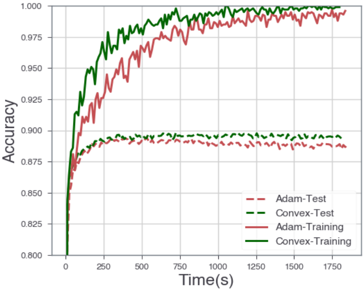
Non-convex **Convex**

Convex program can be globally optimized by standard interior-point solvers with complexity $\mathcal{O}(poly(n, d))$

Numerical Results



(a) CIFAR-10



(b) Fashion-MNIST

Figure 1: Test accuracy of a three-layer architecture trained using the non-convex formulation and the convex program

Takeaways and Open Problems

- Three-layer ReLU networks can be trained via convex optimization
  - don't need hyperparameter search, e.g., learning rate and initialization

Takeaways and Open Problems

- Three-layer ReLU networks can be trained via convex optimization
  - don't need hyperparameter search, e.g., learning rate and initialization
  - don't need heuristics such as dropout

Takeaways and Open Problems

- Three-layer ReLU networks can be trained via convex optimization
  - don't need hyperparameter search, e.g., learning rate and initialization
  - don't need heuristics such as dropout
- Convex problem has polynomial-time complexity with respect to the number of samples n and the feature dimension d

Takeaways and Open Problems

- Three-layer ReLU networks can be trained via convex optimization
  - don't need hyperparameter search, e.g., learning rate and initialization
  - don't need heuristics such as dropout
- Convex problem has polynomial-time complexity with respect to the number of samples n and the feature dimension d
- **Limitations:**
  - convex representation is restricted to three layers (two ReLU layers)

Takeaways and Open Problems

- Three-layer ReLU networks can be trained via convex optimization
  - don't need hyperparameter search, e.g., learning rate and initialization
  - don't need heuristics such as dropout
- Convex problem has polynomial-time complexity with respect to the number of samples n and the feature dimension d
- **Limitations:**
  - convex representation is restricted to three layers (two ReLU layers)
  - we put unit ℓ_2-norm constraints on the first $L-2$ layer weights (weight decay may not be the right way to regularize?)

Takeaways and Open Problems

- Three-layer ReLU networks can be trained via convex optimization
  - don't need hyperparameter search, e.g., learning rate and initialization
  - don't need heuristics such as dropout
- Convex problem has polynomial-time complexity with respect to the number of samples n and the feature dimension d
- **Limitations:**
  - convex representation is restricted to three layers (two ReLU layers)
  - we put unit ℓ_2-norm constraints on the first $L-2$ layer weights (weight decay may not be the right way to regularize?)
  - when the data matrix is full rank, our approach has exponential-time complexity, which is unavoidable unless $\mathrm{P} = \mathrm{NP}$

References

Haeffele, B. D. and Vidal, R. (2017). Global optimality in neural network training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7331–7339.

Pilanci, M. and Ergen, T. (2020). Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7695–7705. PMLR.

References ii

Zhang, H., Shao, J., and Salakhutdinov, R. (2019). Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1099–1109.