On Dropout and Nuclear Norm Regularization

Poorya Mianjy and Raman Arora

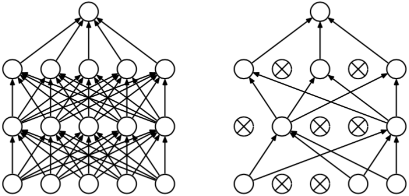
Johns Hopkins University

June 10, 2019

Motivation

- Algorithmic approaches endow deep learning systems with certain inductive biases that help generalization.
- In this paper we study dropout, one of the most popular algorithmic heuristics for training deep neural nets.

Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov



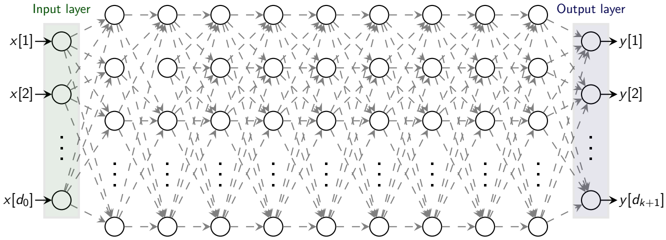
(a) Standard Neural Net (b) After applying dropout.

Problem Setup

- Deep linear networks with k hidden layers

$$f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{x}, \ \ \mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$$

where $\mathrm{w} = \{\mathrm{W}_i\}_{i=1}^{k+1}$ is the set of weight matrices.

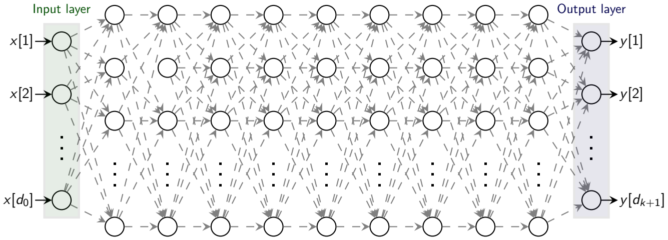


Problem Setup

- Deep linear networks with k hidden layers

$$f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{x}, \;\; \mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$$

  where $\mathrm{w} = \{\mathrm{W}_i\}_{i=1}^{k+1}$ is the set of weight matrices.
- $\mathrm{x} \in \mathbb{R}^{d_0}, \; \mathrm{y} \in \mathbb{R}^{d_{k+1}}, \; (\mathrm{x}, \mathrm{y}) \sim \mathcal{D}$. Assume $\mathbb{E}[\mathrm{x}\mathrm{x}^\top] = \mathrm{I}$.



Problem Setup

- Deep linear networks with k hidden layers

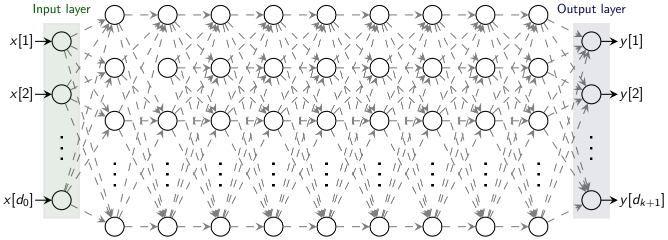
$$f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{x}, \;\; \mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$$

  where $\mathrm{w} = \{\mathrm{W}_i\}_{i=1}^{k+1}$ is the set of weight matrices.

- $\mathrm{x} \in \mathbb{R}^{d_0}, \;\; \mathrm{y} \in \mathbb{R}^{d_{k+1}}, \;\; (\mathrm{x}, \mathrm{y}) \sim \mathcal{D}$. Assume $\mathbb{E}[\mathrm{x}\mathrm{x}^\top] = \mathrm{I}$.
- Learning problem: minimize the *population risk*

$$L(\mathrm{w}) := \mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[\|\mathrm{y} - f_{\mathrm{w}}(\mathrm{x})\|^2]$$

  based on iid samples from the distribution.

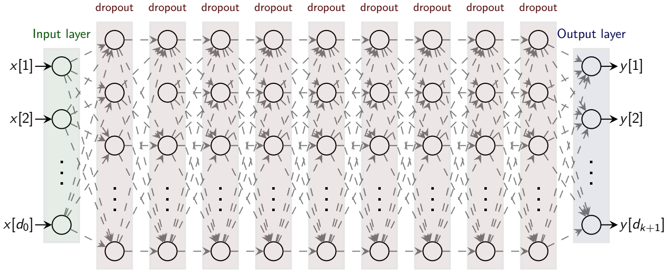


Problem Setup

- Network perturbed by dropping hidden nodes at random, computing

$$\bar{f}_{\mathrm{w}}(\mathrm{x}) = \mathrm{W}_{k+1}\mathrm{B}_k\mathrm{W}_k \cdots \mathrm{B}_1\mathrm{W}_1\mathrm{x},$$

where $\mathrm{B}_i(j,j) = 0$ with probability $1-\theta$, and $\frac{1}{\theta}$ with probability θ.



Problem Setup

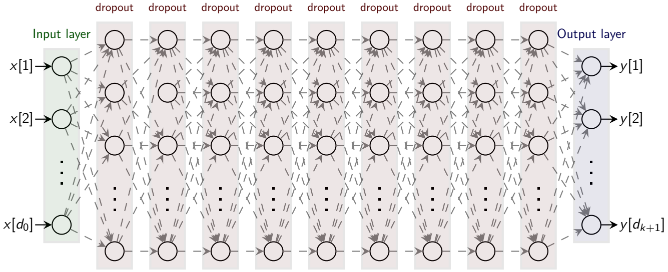
- Network perturbed by dropping hidden nodes at random, computing

$$\bar{f}_{\mathrm{w}}(\mathrm{x}) = \mathrm{W}_{k+1}\mathrm{B}_k\mathrm{W}_k \cdots \mathrm{B}_1\mathrm{W}_1\mathrm{x},$$

  where $\mathrm{B}_i(j,j) = 0$ with probability $1-\theta$, and $\frac{1}{\theta}$ with probability θ.

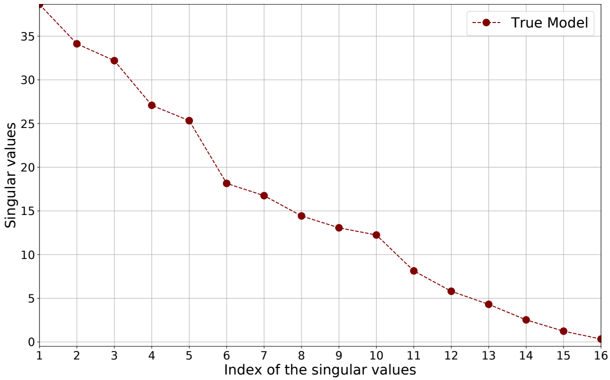
- dropout boils down to SGD on the *dropout objective*

$$L_\theta(\mathrm{w}) := \mathbb{E}_{\{\mathrm{B}_i\},(\mathrm{x},\mathrm{y})} \|\mathrm{y} - \bar{f}_{\mathrm{w}}(\mathrm{x})\|^2$$



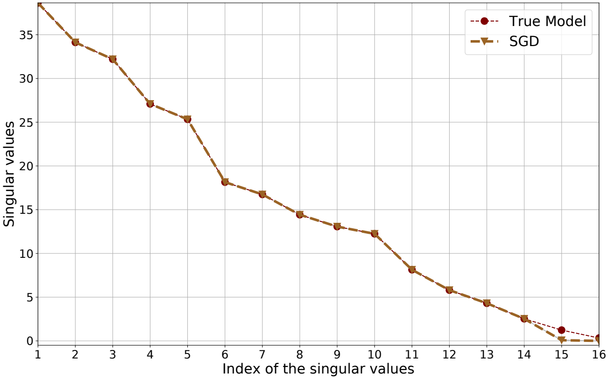
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



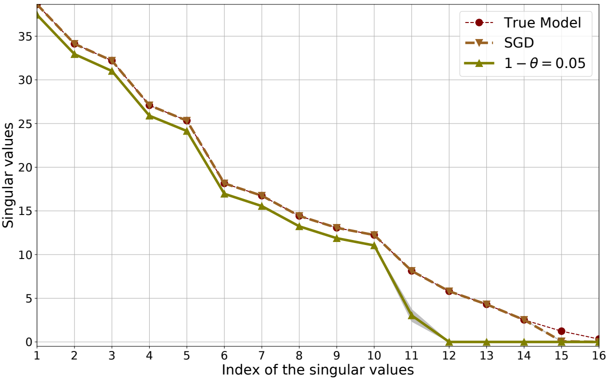
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



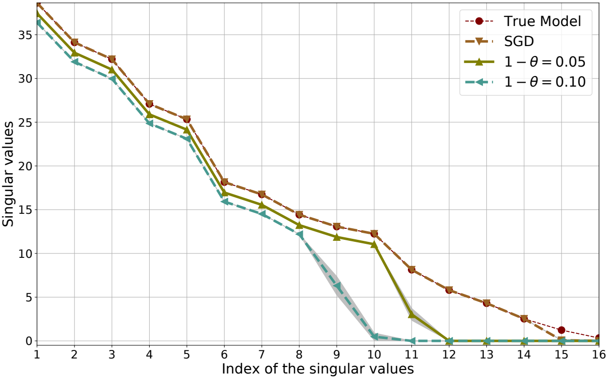
Empirical Observation

- 3-layer network with width/input/output dimensionality = 20.



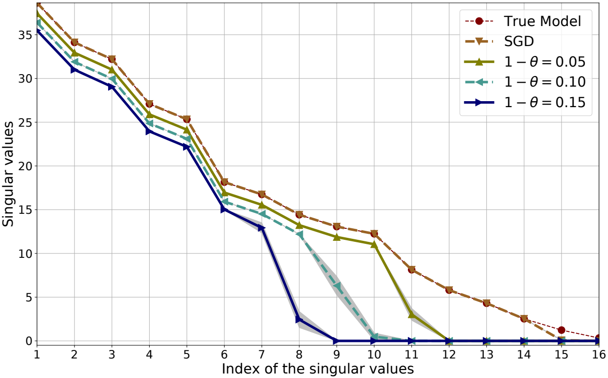
Empirical Observation

- 3-layer network with width/input/output dimensionality = 20.



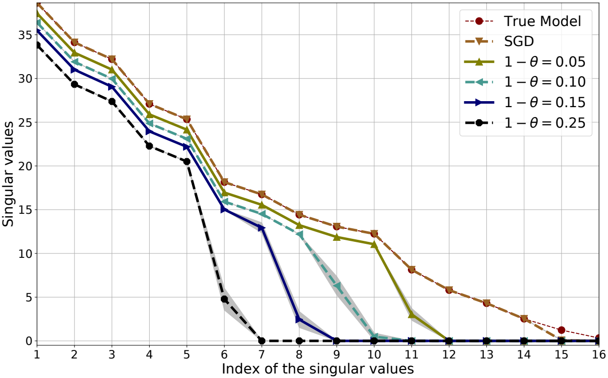
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



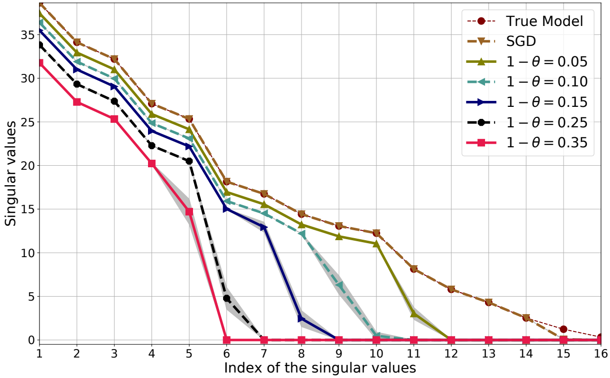
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



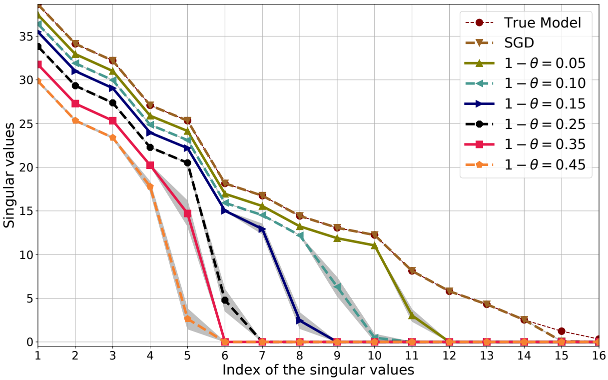
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



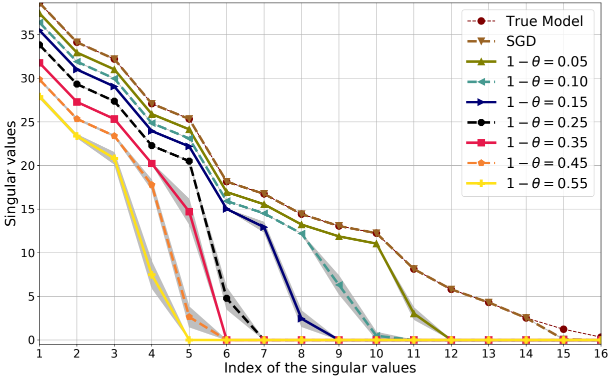
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



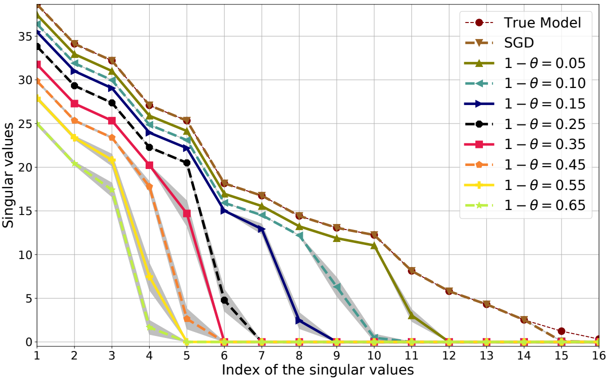
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



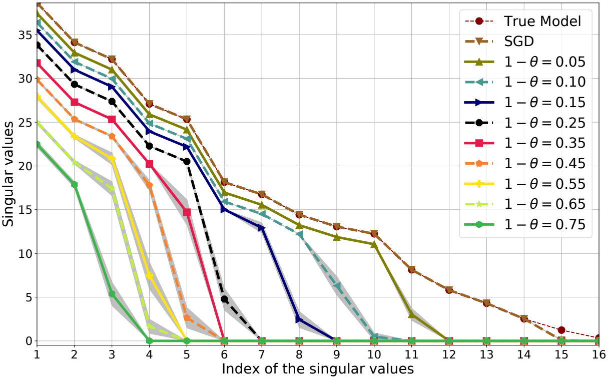
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



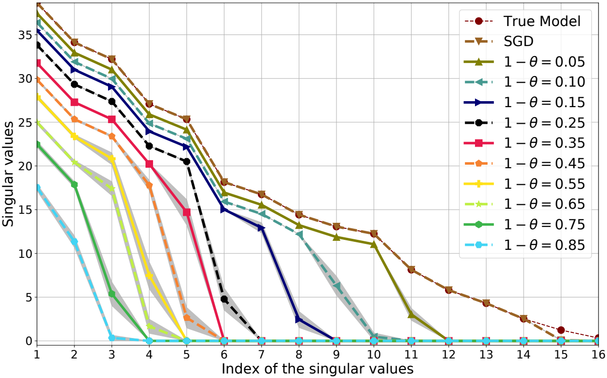
Empirical Observation

- 3-layer network with width/input/output dimensionality $= 20$.



Empirical Observation

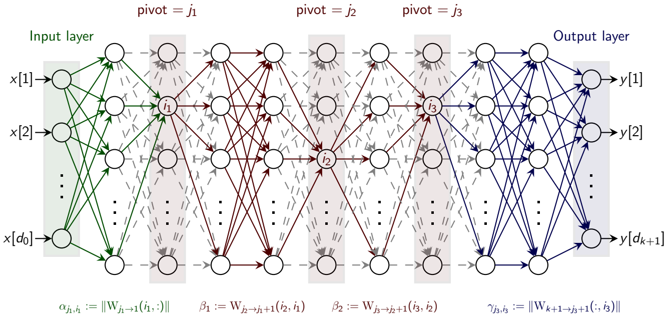
- 3-layer network with width/input/output dimensionality = 20.



Main Results

Explicit Regularizer

Give full characterization of $R(\mathrm{w}) := L_\theta(\mathrm{w}) - L(\mathrm{w})$



Main Results

Explicit Regularizer

Give full characterization of $R(\mathrm{w}) := L_\theta(\mathrm{w}) - L(\mathrm{w})$

Induced Regularizer

$$\Theta(\mathrm{M}) := \min_{f_{\mathrm{w}} = \mathrm{M}} R(\mathrm{w})$$

Main Results

Explicit Regularizer

Give full characterization of $R(\mathrm{w}) := L_\theta(\mathrm{w}) - L(\mathrm{w})$

Induced Regularizer

$$\Theta(\mathrm{M}) := \min_{f_{\mathrm{w}}=\mathrm{M}} R(\mathrm{w})$$

- Multi-dimensional output

$$\Theta^{**}(f_{\mathrm{w}}) = \nu_{\{d_i\}} \|f_{\mathrm{w}}\|_*^2$$

Main Results

Explicit Regularizer

Give full characterization of $R(\mathrm{w}) := L_\theta(\mathrm{w}) - L(\mathrm{w})$

Induced Regularizer

$$\Theta(\mathrm{M}) := \min_{f_\mathrm{w}=\mathrm{M}} R(\mathrm{w})$$

- Multi-dimensional output

$$\Theta^{**}(f_\mathrm{w}) = \nu_{\{d_i\}} \|f_\mathrm{w}\|_*^2$$

- One-dimensional output

$$\Theta(f_\mathrm{w}) = \Theta^{**}(f_\mathrm{w}) = \nu_{\{d_i\}} \|f_\mathrm{w}\|^2$$

Main Results

Explicit Regularizer

Give full characterization of $R(\mathrm{w}) := L_\theta(\mathrm{w}) - L(\mathrm{w})$

Induced Regularizer

$$\Theta(\mathrm{M}) := \min_{f_\mathrm{w}=\mathrm{M}} R(\mathrm{w})$$

- Multi-dimensional output

$$\Theta^{**}(f_\mathrm{w}) = \nu_{\{d_i\}} \|f_\mathrm{w}\|_*^2$$

- One-dimensional output

$$\Theta(f_\mathrm{w}) = \Theta^{**}(f_\mathrm{w}) = \nu_{\{d_i\}} \|f_\mathrm{w}\|^2$$

Effective Regularization Parameter

$\nu_{\{d_i\}}$ increases with depth and decreases with width

deeper and narrower networks are more biased towards low-rank solutions

Thanks for your attention!

Stop by Poster 79 for more information.