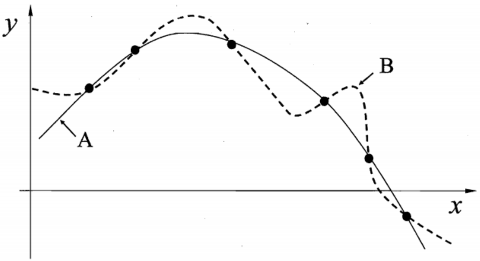
Inductive Biases due to Dropout

Shihua Zhang

October 27, 2021

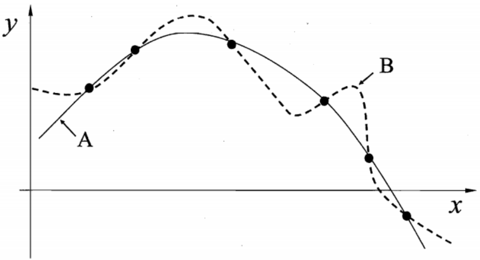
Inductive Bias

- Given a finite dataset, there are many possible solutions to the learning problem.
- They exhibit equally “good” performance on the training points.



Inductive Bias

- Given a finite dataset, there are many possible solutions to the learning problem.
- They exhibit equally “good” performance on the training points.



- How to select the ones for better generalization?
- The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict unseen data.

Inductive Biases due to Algorithmic Regularization

Several regularization strategies help to generalize in deep learning:

- Explicit regularization on objectives
  - ℓ_1 regularization
  - ℓ_2 regularization

Inductive Biases due to Algorithmic Regularization

Several regularization strategies help to generalize in deep learning:

- Explicit regularization on objectives
  - ℓ_1 regularization
  - ℓ_2 regularization
- Heurisitic regularization techniques
  - Early stopping of back-propagation [Caruana et al., 2001]
  - Batch normalization [Ioffe and Szegedy, 2015]
  - Dropout [Srivastava et al., 2014]

Inductive Biases due to Algorithmic Regularization

Several regularization strategies help to generalize in deep learning:

- Explicit regularization on objectives
  - ℓ_1 regularization
  - ℓ_2 regularization
- Heurisitic regularization techniques
  - Early stopping of back-propagation [Caruana et al., 2001]
  - Batch normalization [Ioffe and Szegedy, 2015]
  - Dropout [Srivastava et al., 2014]

Today, we focus on the inductive biases due to Dropout.

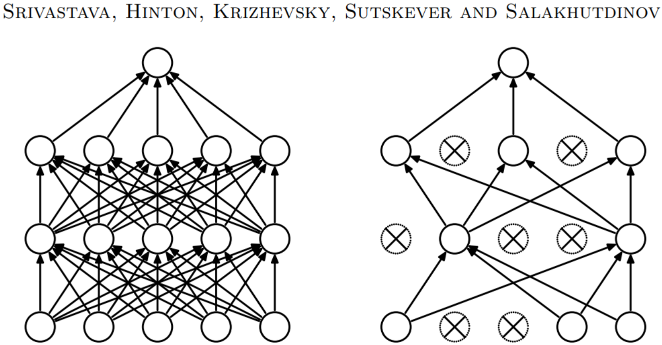
Overview

- Introduction to Dropout
- Matrix Sensing with Dropout
  - Gaussian sensing matrices
  - Matrix completion
- Deep Neural Networks with Dropout
- Landscape of the Optimization Problem
  - Implicit bias in local optima
  - Landscape properties

Outline

Dropout

- A popular algorithmic heuristic with limited formal understanding.
- Key idea: randomly drop units of DNN during training.
- Motivation: as a way to break "co-adaptation" [Srivastava et al., 2014].



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

Training with Dropout

- With dropout, the feed-forward operation becomes

$$\mathrm{B}_{ii} \sim \frac{1}{1-p}\text{Bernoulli}(1-p) \quad \text{i.i.d.}$$
$$z_i^{(l+1)} = \mathrm{W}_i^{(l+1)}\mathrm{B}y^{(l)} + b_i^{(l+1)}$$
$$y_i^{(l+1)} = \sigma\left(z_i^{(l+1)}\right)$$

Stochastic gradient descent

- For each training case in a mini-batch, sample a thinned network
- Do forward and back propagation on this thinned network
- The gradients for each parameter are averaged over the training cases in each mini-batch

Experiments on Image Data Sets

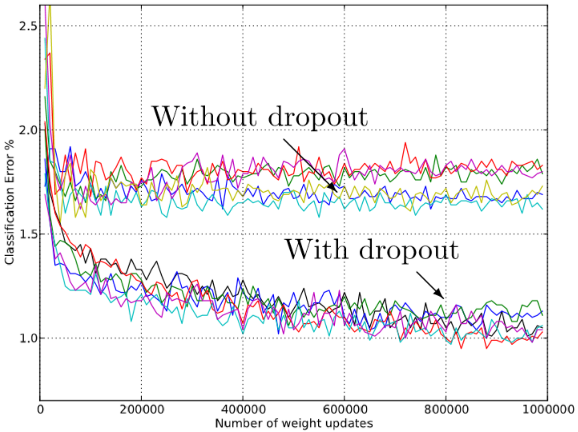


Figure: Test error for different architectures with dropout [Srivastava et al., 2014].

Outline

Matrix Sensing

- Recover a matrix $\mathrm{M}_* \in \mathbb{R}^{d_2 \times d_0}$, with rank $r_* := \mathrm{Rank}\left(\mathrm{M}_*\right)$
- Given $y_i = \left\langle \mathrm{M}_*, \ \mathrm{A}^{(i)} \right\rangle$, for matrices $\mathrm{A}^{(1)}, \cdots, \mathrm{A}^{(n)}$, $n \ll d_2 d_0$
- Represent M in the factorized form and solve:

$$\underset{\mathrm{U}\in\mathbb{R}^{d_2\times d_1},\ \mathrm{V}\in\mathbb{R}^{d_0\times d_1}}{\text{minimize}} \widehat{L}(\mathrm{U}, \mathrm{V}) := \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \left\langle \mathrm{U}\mathrm{V}^\top, \mathrm{A}^{(i)} \right\rangle\right)^2 \tag{1}$$

Matrix Sensing

- Recover a matrix $\mathrm{M}_* \in \mathbb{R}^{d_2 \times d_0}$, with rank $r_* := \text{Rank}(\mathrm{M}_*)$
- Given $y_i = \left\langle \mathrm{M}_*, \ \mathrm{A}^{(i)} \right\rangle$, for matrices $\mathrm{A}^{(1)}, \cdots, \mathrm{A}^{(n)}$, $n \ll d_2 d_0$
- Represent M in the factorized form and solve:

$$\underset{\mathrm{U} \in \mathbb{R}^{d_2 \times d_1},\ \mathrm{V} \in \mathbb{R}^{d_0 \times d_1}}{\text{minimize}} \widehat{L}(\mathrm{U}, \mathrm{V}) := \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \left\langle \mathrm{U}\mathrm{V}^\top, \mathrm{A}^{(i)} \right\rangle \right)^2 \tag{1}$$

- Dropout as an instance of SGD:

$$\widehat{L}_{\text{drop}}(\mathrm{U}, \mathrm{V}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathrm{B}} \left(y_i - \left\langle \mathrm{U}\mathrm{B}\mathrm{V}^\top, \mathrm{A}^{(i)} \right\rangle \right)^2 \tag{2}$$

where diagonal matrix B has $\mathrm{B}_{jj} \sim \frac{1}{1-p} \text{Ber}(1-p)$

Explicit Regularizer

- Key: Dropout explicitly regularizes the empirical objective

$$\widehat{L}_{\text{drop}}(\mathrm{U}, \mathrm{V}) = \widehat{L}(\mathrm{U}, \mathrm{V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U}, \mathrm{V}) \tag{3}$$

where $\widehat{R}(\mathrm{U}, \mathrm{V}) = \sum_{j=1}^{d_1} \frac{1}{n} \sum_{i=1}^{n} \left(\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j\right)^2$, data dependent

Explicit Regularizer

- **Key**: Dropout **explicitly regularizes** the empirical objective

$$\widehat{L}_{\text{drop}}(\mathrm{U}, \mathrm{V}) = \widehat{L}(\mathrm{U}, \mathrm{V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U}, \mathrm{V}) \tag{3}$$

where $\widehat{R}(\mathrm{U}, \mathrm{V}) = \sum_{j=1}^{d_1} \frac{1}{n} \sum_{i=1}^{n} \left(\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j\right)^2$, data dependent

- **Proof**. Consider one of the summands in the Dropout objective.

$$\mathbb{E}_{\mathrm{B}}\left[\left(y_i - \left\langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \right\rangle\right)^2\right] = \left(\mathbb{E}_{\mathrm{B}}\left[y_i - \left\langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \right\rangle\right]\right)^2 + \mathrm{Var}\left(y_i - \left\langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \right\rangle\right)$$

- Note that $\mathbb{E}\left[\mathrm{B}_{jj}\right] = 1$ and $\mathrm{Var}\left(\mathrm{B}_{jj}\right) = \frac{p}{1-p}$, the first term on the right side is equal to $\left(y_i - \left\langle \mathrm{UV}^\top, \mathrm{A}^{(i)} \right\rangle\right)^2$.

Explicit Regularizer

- For the second term we have

$$
\begin{aligned}
\mathrm{Var}\left(y_i - \left\langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \right\rangle\right) &= \mathrm{Var}\left(\left\langle \mathrm{UBV}^\top, \mathrm{A}^{(i)} \right\rangle\right) \\
&= \mathrm{Var}\left(\left\langle \mathrm{B}, \mathrm{U}^\top \mathrm{A}^{(i)} \mathrm{V} \right\rangle\right) \\
&= \mathrm{Var}\left(\sum_{j=1}^{d_1} \mathrm{B}_{jj} \mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j\right) \\
&= \sum_{j=1}^{d_1} \left(\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j\right)^2 \mathrm{Var}\left(\mathrm{B}_{jj}\right) \\
&= \frac{p}{1-p} \sum_{j=1}^{d_1} \left(\mathrm{u}_j^\top \mathrm{A}^{(i)} \mathrm{v}_j\right)^2
\end{aligned}
$$

Induced Regularizer

- Thus,

$$\widehat{L}_{\text{drop}} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \left\langle \mathrm{UV}^\top, \mathrm{A}^{(i)}\right\rangle\right)^2 + \frac{1}{n}\sum_{i=1}^{n}\frac{p}{1-p}\sum_{j=1}^{d_1}\left(\mathrm{u}_j^\top \mathrm{A}^{(i)}\mathrm{v}_j\right)^2$$
$$= \widehat{L}(\mathrm{U},\mathrm{V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U},\mathrm{V})$$

- Expected regularizer: $R(\mathrm{U},\mathrm{V}) := \mathbb{E}_{\mathrm{A}}[\widehat{R}(\mathrm{U},\mathrm{V})]$
- Induced regularizer: Consider the factors with the minimal value of $R(\mathrm{U},\mathrm{V})$ among all that yield the same empirical loss

$$\Theta(\mathrm{M}) := \min_{\mathrm{UV}^\top = \mathrm{M}} R(\mathrm{U},\mathrm{V})$$

Gaussian Matrix Sensing

- Assume that the entries of the sensing matrices are iid as standard Gaussian, i.e., $\mathrm{A}_{k\ell}^{(i)} \sim \mathcal{N}(0, 1)$.
- Hint: The induced regularizer due to Dropout provides the nuclear-norm regularization:

$$\Theta(\mathrm{M}) := \min_{\mathrm{U}\mathrm{V}^\top = \mathrm{M}} R(\mathrm{U}, \mathrm{V}) = \frac{1}{d_1} \|\mathrm{M}\|_*^2$$

Gaussian Matrix Sensing

- Assume that the entries of the sensing matrices are iid as standard Gaussian, i.e., $\mathrm{A}^{(i)}_{k\ell} \sim \mathcal{N}(0, 1)$.
- Hint: The induced regularizer due to Dropout provides the nuclear-norm regularization:

$$\Theta(\mathrm{M}) := \min_{\mathrm{U}\mathrm{V}^\top = \mathrm{M}} R(\mathrm{U}, \mathrm{V}) = \frac{1}{d_1} \|\mathrm{M}\|_*^2$$

- For any pair of factors (U, V), the expected regularizer is

$$R(\mathrm{U}, \mathrm{V}) = \sum_{i=1}^{d_1} \mathbb{E}_{\mathrm{A}} \left[\left(\mathrm{u}_i^\top \mathrm{A} \mathrm{v}_i \right)^2 \right] = \sum_{i=1}^{d_1} \|\mathrm{u}_i\|^2 \|\mathrm{v}_i\|^2$$

Gaussian Matrix Sensing

- By Cauchy-Schwartz inequality

$$
\begin{aligned}
R(\mathrm{U}, \mathrm{V}) = \sum_{i=1}^{d_1} \|\mathrm{u}_i\|^2 \|\mathrm{v}_i\|^2 &\geqslant \frac{1}{d_1} \left(\sum_{i=1}^{d_1} \|\mathrm{u}_i\| \|\mathrm{v}_i\| \right)^2 \\
&= \frac{1}{d_1} \left(\sum_{i=1}^{d_1} \left\|\mathrm{u}_i \mathrm{v}_i^\top\right\|_* \right)^2 \\
&\geqslant \frac{1}{d_1} \left(\left\| \sum_{i=1}^{d_1} \mathrm{u}_i \mathrm{v}_i^\top \right\|_* \right)^2 = \frac{1}{d_1} \left\|\mathrm{U}\mathrm{V}^\top\right\|_*^2
\end{aligned}
$$

  where the equality follows because for any pair of vectors a, b, it holds that $\left\|\mathrm{a}\mathrm{b}^\top\right\|_* = \left\|\mathrm{a}\mathrm{b}^\top\right\|_F = \|\mathrm{a}\|\|\mathrm{b}\|$
- Lower bound can be achieved for all (U, V) s.t.

$$
\|u_i\| \|v_i\| = \frac{1}{d_1} \left\|\mathrm{U}\mathrm{V}^\top\right\|_*, \forall i
$$

Gaussian Matrix Sensing

Based on the following result on (U, V) [Mianjy et al., 2018]:

Theorem 1

For any pair of matrices $\mathrm{U} \in \mathbb{R}^{d_2 \times d_1}$, $\mathrm{V} \in \mathbb{R}^{d_0 \times d_1}$*, there exists a rotation matrix* Q *such that matrices* $\widetilde{\mathrm{U}} := \mathrm{UQ}$, $\widetilde{\mathrm{V}} := \mathrm{VQ}$ *satisfy* $\|\widetilde{\mathrm{u}}_i\| \, \|\widetilde{\mathrm{v}}_i\| = \frac{1}{d_1} \left\| \mathrm{UV}^\top \right\|_*$*, for all* $i \in [d_1]$.

Gaussian Matrix Sensing

Based on the following result on (U, V) [Mianjy et al., 2018]:

Theorem 1

For any pair of matrices $\mathrm{U} \in \mathbb{R}^{d_2 \times d_1}$, $\mathrm{V} \in \mathbb{R}^{d_0 \times d_1}$, there exists a rotation matrix Q such that matrices $\widetilde{\mathrm{U}} := \mathrm{UQ}$, $\widetilde{\mathrm{V}} := \mathrm{VQ}$ satisfy $\|\widetilde{\mathrm{u}}_i\| \|\widetilde{\mathrm{v}}_i\| = \frac{1}{d_1} \left\|\mathrm{UV}^\top\right\|_*$, for all $i \in [d_1]$.

$$
\begin{aligned}
R(\mathrm{UQ}, \mathrm{VQ}) &= \sum_{i=1}^{d_1} \|\mathrm{Uq}_i\|^2 \|\mathrm{Vq}_i\|^2 \\
&= \sum_{i=1}^{d_1} \frac{1}{d_1^2} \left\|\mathrm{UV}^\top\right\|_*^2 \\
&= \frac{1}{d_1} \left\|\mathrm{UV}^\top\right\|_*^2
\end{aligned}
$$

Matrix Completion

- Matrix completion (MC) can be formulated as a special case of matrix sensing with sensing matrices being random indicator matrices.
- Let $\mathrm{A}^{(j)}$ be an indicator matrix whose (i, k)-th element is selected randomly with probability $\mathrm{p}(i), \mathrm{q}(k)$, then

$$\Theta(\mathrm{M}) = \frac{1}{d_1} \left\| \sqrt{\mathrm{diag}(\mathrm{p})} \mathrm{U}\mathrm{V}^\top \sqrt{\mathrm{diag}(\mathrm{q})} \right\|_*^2 \qquad \text{(weighted trace-norm)}$$

- The weighted trace-norm or nuclear norm has been studied by [Salakhutdinov and Srebro, 2010][Foygel et al., 2011]

Matrix Completion

- Matrix completion (MC) can be formulated as a special case of matrix sensing with sensing matrices being random indicator matrices.
- Let $\mathrm{A}^{(j)}$ be an indicator matrix whose (i, k)-th element is selected randomly with probability $\mathrm{p}(i), \mathrm{q}(k)$, then

$$\Theta(\mathrm{M}) = \frac{1}{d_1} \left\| \sqrt{\mathrm{diag}(\mathrm{p})} \mathrm{UV}^\top \sqrt{\mathrm{diag}(\mathrm{q})} \right\|_*^2 \quad \text{(weighted trace-norm)}$$

- The weighted trace-norm or nuclear norm has been studied by [Salakhutdinov and Srebro, 2010][Foygel et al., 2011]
- Key: A generalization bound for MC with dropout in terms of the value of the explicit regularizer at the minimum of the empirical problem [Arora et al., 2021].

A generalization bound for Matrix Completion

Theorem 2 ([Arora et al., 2021])

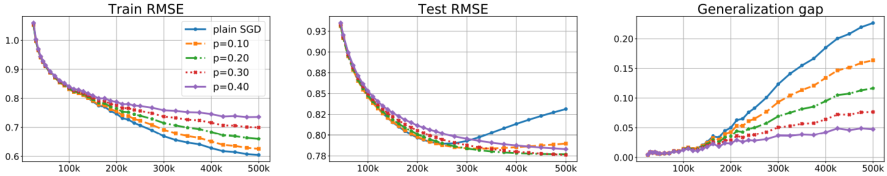
Assume that $d_2 \geqslant d_0$ *and* $\|\mathrm{M}_*\| \leqslant 1$. *Furthermore, assume that* $\min_{i,k} p(i) q(k) \geqslant \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$. *Let* (U, V) *be a minimizer of the dropout objective in equation (3). Let* α *be such that* $R(\mathrm{U}, \mathrm{V}) \leqslant \alpha / d_1$. *Then, for any* $\delta \in (0, 1)$, *the following generalization bounds holds with probability at least* $1 - \delta$ *over a sample of size* n *:*

$$L\left(g\left(\mathrm{UV}^\top\right)\right) \leqslant \widehat{L}(\mathrm{U}, \mathrm{V}) + 8\sqrt{\frac{2\alpha d_2 \log\left(d_2\right) + \frac{1}{4}\log(2/\delta)}{n}}$$

where $g(\mathrm{M})$ *thresholds* M *at* ± 1, *i.e.* $g(\mathrm{M})(i, j) = \max\{-1, \min\{1, \mathrm{M}(i, j)\}\}$, *and* $L\left(g\left(\mathrm{UV}^\top\right)\right) := \mathbb{E}(y - \left\langle g\left(\mathrm{UV}^\top\right), \mathrm{A}\right\rangle)^2$ *is the true risk of* $g\left(\mathrm{UV}^\top\right)$

Empirical Results on Matrix Completion

MovieLens dataset: 10M ratings for 11K movies by 72K users.



- The training error, test error, and generalization gap for plain SGD and dropout with different p as a function of the number of iterations.
- Intuitively, a larger dropout rate p results in a smaller α.

Empirical Results on Matrix Completion

MovieLens dataset: 10M ratings for 11K movies by 72K users.

	plain SGD		dropout			
width	last iterate	best iterate	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$
$d_1 = 30$	0.8041	0.7938	0.7805	0.785	0.7991	0.8186
$d_1 = 70$	0.8315	0.7897	0.7899	0.7771	0.7763	0.7833
$d_1 = 110$	0.8431	0.7873	0.7988	0.7813	0.7742	0.7743
$d_1 = 150$	0.8472	0.7858	0.8042	0.7852	0.7756	0.7722
$d_1 = 190$	0.8473	0.7844	0.8069	0.7879	0.7772	0.772

Figure: Test RMSE of plain SGD as well as the dropout algorithm with various dropout rates for various factorization sizes.[Arora et al., 2021]

- Dropout performance improves with the size of the parametrization.
- SGD has worse generalization even for best iterate picked on test data.

Outline

Regression with Deep Neural Networks

- $\mathcal{X} \subseteq \mathbb{R}^{d_0}$, $\mathcal{Y} \subseteq [-1, 1]^{d_2}$, $\mathcal{D}$ is an (unknown) distribution on $\mathcal{X} \times \mathcal{Y}$
- 2-layers neural networks parameterized by w

$$f_{\mathrm{w}}(\mathrm{x}) = \mathrm{U}\sigma\left(\mathrm{V}^{\top}\mathrm{x}\right)$$

where $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}$, $\mathrm{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$.

- Squared ℓ_2 loss, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, with $\ell(\mathrm{y}, \mathrm{y}') = \|\mathrm{y} - \mathrm{y}'\|^2$

Regression with Deep Neural Networks

- $\mathcal{X} \subseteq \mathbb{R}^{d_0}$, $\mathcal{Y} \subseteq [-1, 1]^{d_2}$, $\mathcal{D}$ is an (unknown) distribution on $\mathcal{X} \times \mathcal{Y}$
- 2-layers neural networks parameterized by w

$$f_{\mathrm{w}}(\mathrm{x}) = \mathrm{U}\sigma\left(\mathrm{V}^{\top}\mathrm{x}\right)$$

where $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}$, $\mathrm{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$.

- Squared ℓ_2 loss, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, with $\ell(\mathrm{y}, \mathrm{y}') = \|\mathrm{y} - \mathrm{y}'\|^2$
- Goal: find a hypothesis $f_{\mathrm{w}} : \mathcal{X} \to \mathcal{Y}$, with a small

$$L(\mathrm{w}) := \mathbb{E}_{\mathcal{D}}\left[\ell\left(f_{\mathrm{w}}(\mathrm{x}), \mathrm{y}\right)\right] \qquad \text{(population risk)}$$

- Given n samples $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{n} \sim \mathcal{D}^n$ drawn i.i.d. from $\mathcal{D}$

Regression with Deep Neural Networks

- $\mathcal{X} \subseteq \mathbb{R}^{d_0}$, $\mathcal{Y} \subseteq [-1, 1]^{d_2}$, $\mathcal{D}$ is an (unknown) distribution on $\mathcal{X} \times \mathcal{Y}$
- 2-layers neural networks parameterized by w

$$f_{\mathrm{w}}(\mathrm{x}) = \mathrm{U}\sigma\left(\mathrm{V}^{\top}\mathrm{x}\right)$$

  where $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}$, $\mathrm{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$.
- Squared ℓ_2 loss, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, with $\ell(\mathrm{y}, \mathrm{y}') = \|\mathrm{y} - \mathrm{y}'\|^2$
- Goal: find a hypothesis $f_{\mathrm{w}} : \mathcal{X} \to \mathcal{Y}$, with a small

$$L(\mathrm{w}) := \mathbb{E}_{\mathcal{D}}\left[\ell\left(f_{\mathrm{w}}(\mathrm{x}), \mathrm{y}\right)\right] \qquad \text{(population risk)}$$

- Given n samples $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{n} \sim \mathcal{D}^n$ drawn i.i.d. from $\mathcal{D}$
- ERM: minimize

$$\widehat{L}(\mathrm{w}) := \frac{1}{n}\sum_{i=1}^{n}\left[\|\mathrm{y}_i - f_{\mathrm{w}}(\mathrm{x}_i)\|^2\right] \qquad \text{(empirical risk)}$$

Dropout in Deep Neural Networks

- Dropout as SGD iterates – the dropout objective:

$$\widehat{L}_{\text{drop}}(w) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathrm{B}} \left\| \mathrm{y}_i - \mathrm{UB}\sigma\left(\mathrm{V}^\top \mathrm{x}_i\right)\right\|^2$$

where $\mathrm{B}_{ii} \sim \frac{1}{1-p}\text{Bern}(1-p), i \in [d_1]$.

- We seek to understand the explicit regularizer due to dropout:

$$\widehat{R}(\mathrm{w}) := \widehat{L}_{\text{drop}}(\mathrm{w}) - \widehat{L}(\mathrm{w}) \quad \text{(explicit regularizer)}$$

- Denote the output of the i-th hidden node on input x by $\mathrm{a}_i(\mathrm{x})$; $\mathrm{a}(\mathrm{x}) \in \mathbb{R}^{d_1}$ denotes the activation of the hidden layer on input x.
- Rewrite the Dropout objective as

$$\widehat{L}_{\text{drop}}(w) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathrm{B}} \left\| \mathrm{y}_i - \mathrm{UBa}(\mathrm{x}_i)\right\|^2.$$

Dropout Regularizer in Deep Regression

- The explicit regularizer due to dropout is

$$\widehat{R}(\mathrm{w}) = \lambda \sum_{j=1}^{d_1} \left\|\mathrm{u}_j\right\|^2 \widehat{a}_j^2, \quad \widehat{a}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_j\left(x_i\right)^2}$$

where $\lambda = \frac{p}{1-p}$ is the regularization parameter.

Dropout Regularizer in Deep Regression

- The explicit regularizer due to dropout is

$$\widehat{R}(\mathrm{w}) = \lambda \sum_{j=1}^{d_1} \left\|\mathrm{u}_j\right\|^2 \widehat{a}_j^2, \quad \widehat{a}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_j\left(x_i\right)^2}$$

  where $\lambda = \frac{p}{1-p}$ is the regularization parameter.
- Consider ReLU activations and input distributions that are symmetric and isotropic, i.e., $\mathbb{P}_{\mathcal{X}}(\mathrm{x}) = \mathbb{P}_{\mathcal{X}}(-\mathrm{x})$ and $\mathrm{C} = \mathbb{E}\left[\mathrm{x}\mathrm{x}^\top\right] = \mathrm{I}$. Then the expected regularizer due to dropout is given as

$$R(\mathrm{w}) := \mathbb{E}[\widehat{R}(\mathrm{w})] = \frac{\lambda}{2} \sum_{i_0, i_1, i_2 = 1}^{d_0, d_1, d_2} \mathrm{U}\left(i_2, i_1\right)^2 \mathrm{V}\left(i_1, i_0\right)^2$$

- It is a data-dependent variant of the ℓ_2 path-norm of the network [Neyshabur et al., 2015].
- It can yield capacity control in deep learning.

Function Class Learned by Dropout

- Let $d_2 = 1$, we focus on the following distribution-dependent class

$$\mathcal{F}_\alpha := \left\{ f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{u}^\top \sigma\left(\mathrm{V}^\top \mathrm{x}\right), \sum_{i=1}^{d_1} |u_i|\, a_i \leqslant \alpha \right\}$$

  where $a_i^2 := \mathbb{E}_{\mathrm{x}}\left[\widehat{a}_i^2\right] = \mathbb{E}_{\mathrm{x}}\left[a_i(\mathrm{x})^2\right]$
- We argue that networks trained with dropout belong to the class $\mathcal{F}_\alpha$ (for a small value of α).

Function Class Learned by Dropout

- Let $d_2 = 1$, we focus on the following distribution-dependent class

$$\mathcal{F}_\alpha := \left\{ f_\mathrm{w} : \mathrm{x} \mapsto \mathrm{u}^\top \sigma\left(\mathrm{V}^\top \mathrm{x}\right), \sum_{i=1}^{d_1} |u_i|\, a_i \leqslant \alpha \right\}$$

  where $a_i^2 := \mathbb{E}_\mathrm{x}\left[\widehat{a}_i^2\right] = \mathbb{E}_\mathrm{x}\left[a_i(\mathrm{x})^2\right]$

- We argue that networks trained with dropout belong to the class $\mathcal{F}_\alpha$ (for a small value of α).

- By Cauchy-Schwartz inequality,

$$\sum_{i=1}^{d_1} |u_i|\, a_i \leqslant \sqrt{d_1 \sum_{i=1}^{d_1} |u_i|^2\, a_i^2} = \sqrt{d_1 \frac{1}{\lambda} R(\mathrm{w})}$$

  Thus, for a fixed width, dropout controls the function class $\mathcal{F}_\alpha$.

Function Class Learned by Dropout

- This inequality is loose if a small subset of hidden nodes $\mathcal{J} \subset [d_1]$ "co-adapt" in a way that the other hidden nodes (i.e., all $j \in [d_1] \setminus \mathcal{J}$) are almost inactive, i.e. $u_j a_j \approx 0$.
- By minimizing the expected regularizer, dropout is biased towards networks where the gap between $\frac{1}{d_1}\left(\sum_{i=1}^{d_1} |u_i|\, a_i\right)^2$ and $R(\mathrm{w})$ is small, which in turn happens if

$$|u_i|\, a_i \approx \left|u_j\right| a_j, \forall i, j \in [d_1]\,.$$

- Dropout breaks "co-adaptation" by promoting solutions with nearly equal contribution from hidden neurons.

Function Class Learned by Dropout

- This inequality is loose if a small subset of hidden nodes $\mathcal{J} \subset [d_1]$ "co-adapt" in a way that the other hidden nodes (i.e., all $j \in [d_1] \setminus \mathcal{J}$) are almost inactive, i.e. $u_j a_j \approx 0$.
- By minimizing the expected regularizer, dropout is biased towards networks where the gap between $\frac{1}{d_1}\left(\sum_{i=1}^{d_1} |u_i|\, a_i\right)^2$ and $R(\mathrm{w})$ is small, which in turn happens if

$$|u_i|\, a_i \approx \left|u_j\right| a_j, \forall i, j \in [d_1]\,.$$

- Dropout breaks "co-adaptation" by promoting solutions with nearly equal contribution from hidden neurons.
- Next, under mild condition on the input distribution, a generalization bound can be derived.

Bound on the Rademacher Complexity

Assumption 1 (β-retentive)

The marginal input distribution is β-retentive for some $\beta \in (0, 1/2]$, if for any non-zero vector $\mathrm{v} \in \mathbb{R}^d$, it holds that $\mathbb{E}\sigma\left(\mathrm{v}^\top \mathrm{x}\right)^2 \geqslant \beta \mathbb{E}\left(\mathrm{v}^\top \mathrm{x}\right)^2$.

- Mahalanobis norm: $\|\mathrm{X}\|_{\mathrm{C}^\dagger}^2 = \sum_{i=1}^n \mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i$.

Theorem 3

For any sample $S = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n$ of size n,

$$\mathfrak{R}_S\left(\mathcal{F}_\alpha\right) \leqslant \frac{2\alpha \|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n\sqrt{\beta}}$$

Furthermore, it holds for the expected Rademacher complexity that $\mathfrak{R}_n\left(\mathcal{F}_\alpha\right) \leqslant 2\alpha\sqrt{\frac{\mathrm{Rank}(\mathrm{C})}{\beta n}}$.

Generalization Bounds

- Dropout regularizer directly controls the value of α, thereby controlling the Rademacher complexity in Theorem 3.
- Let $g_{\mathrm{w}}(\cdot) := \max\{-1, \min\{1, f_{\mathrm{w}}(\cdot)\}\}$ project the network output f_{w} onto the range $[-1, 1]$. We have the following generalization gurantees for g_{w} based on Theorem 3.

Theorem 4

For any $f_{\mathrm{w}} \in \mathcal{F}_\alpha$, for any $\delta \in (0, 1)$, the following generalization bound holds with probability at least $1 - \delta$ over a sample S of size n

$$L(g_{\mathrm{w}}) \leqslant \widehat{L}(g_{\mathrm{w}}) + \frac{16\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{\sqrt{\beta}n} + 12\sqrt{\frac{\log(2/\delta)}{2n}}$$

Experimental Results

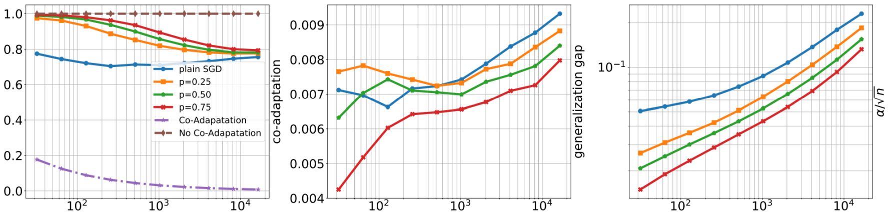


Figure: "co-adaptation", generalization gap and $\alpha/\sqrt{n}$ as a function of the width of networks trained with dropout on MNIST. The trained 2-layer networks achieve 100% training accuracy [Arora et al., 2021]

.

- Increasing the dropout rate results in less co-adaptation empirically.
- Increasing dropout rate decreases the generalization gap.
- The bound of the Rademacher complexity is predictive on the generalization gap.

Outline

Goal

- Dropout is a first-order method and the landscape of the Dropout objective (e.g., Problem (4)) is highly non-convex.
- Can perhaps only hope to find a local minimum, that too provided if the problem has no degenerate saddle points [Ge et al., 2015].
- Therefore, the following questions are expected:
  - What is the implicit bias of dropout in terms of local minima?
  - Do local minima share anything with global minima structurally?
  - Can dropout find a local optimum?

Problem Setup

- We focus on the case of single hidden layer linear autoencoders with tied weights, i.e. $\mathrm{U} = \mathrm{V}$.

$$\mathcal{H}_r := \left\{ h_{\mathrm{U}} : \mathrm{x} \mapsto \mathrm{U}\mathrm{U}^\top \mathrm{x}, \mathrm{U} \in \mathbb{R}^{d_0 \times d_1} \right\}$$

- Assume that the input distribution is isotropic, i.e. $\mathrm{C}_{\mathrm{x}} = \mathbb{E}\left[\mathrm{x}\mathrm{x}^\top\right] = \mathrm{I}$
- The population risk reduces to

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathrm{y} - \mathrm{U}\mathrm{U}^\top \mathrm{x}\right\|^2\right] &= \mathrm{Tr}\left(\mathrm{C}_{\mathrm{y}}\right) - 2\left\langle \mathrm{C}_{\mathrm{yx}}, \mathrm{U}\mathrm{U}^\top \right\rangle + \left\|\mathrm{U}\mathrm{U}^\top\right\|_F^2 \\
&= \left\|\mathrm{M} - \mathrm{U}\mathrm{U}^\top\right\|_F^2 + \mathrm{Tr}\left(\mathrm{C}_{\mathrm{y}}\right) - \|\mathrm{M}\|_F^2
\end{aligned}$$

where $\mathrm{M} = \frac{C_{yx} + C_{xy}}{2}$.

Problem Setup

- Ignoring the terms that are independent of the weight matrix U, the goal is to minimize $L(\mathrm{U}) = \left\|\mathrm{M} - \mathrm{U}\mathrm{U}^\top\right\|_F^2$.

- Solving the following problem with Dropout:

$$\min_{\mathrm{U} \in \mathbb{R}^{d_0 \times d_1}} L_\theta(\mathrm{U}) := \left\|\mathrm{M} - \mathrm{U}\mathrm{U}^\top\right\|_F^2 + \lambda \underbrace{\sum_{i=1}^{d_1} \|\mathrm{u}_i\|^4}_{R(\mathrm{U})} \tag{4}$$

Implicit Bias in Local Optima

- $L(\mathrm{U})$ is rotation invariant, i.e. for any rotation matrix Q

$$L(\mathrm{UQ}) = \left\|\mathrm{M} - \mathrm{UQQ}^\top\mathrm{U}^\top\right\|_F^2 = L(\mathrm{U}), \quad \mathrm{Q}^\top\mathrm{Q} = \mathrm{QQ}^\top = \mathrm{I}$$

But the regularizer is not rotation invariant.

Implicit Bias in Local Optima

- $L(\mathrm{U})$ is rotation invariant, i.e. for any rotation matrix Q

$$L(\mathrm{UQ}) = \left\| \mathrm{M} - \mathrm{UQQ}^\top \mathrm{U}^\top \right\|_F^2 = L(\mathrm{U}), \quad \mathrm{Q}^\top \mathrm{Q} = \mathrm{QQ}^\top = \mathrm{I}$$

  But the regularizer is not rotation invariant.

- By Cauchy-Schwartz inequality, we have

$$R(\mathrm{U}) = \lambda \sum_{i=1}^{d_1} \|\mathrm{u}_i\|^4 \geqslant \frac{\lambda}{d_1} \|\mathrm{U}\|_F^4$$

  with equality iff all the columns of U have equal norms (equalized).

- If the weight matrix U were not equalized, one can design a rotation matrix Q that UQ has a smaller regularizer, hence the objective.

Implicit Bias in Local Optima – Theorem

If U is not equalized, then any ϵ-neighborhood of U contains a point with dropout objective strictly smaller than $L_\theta(\mathrm{U})$.

Theorem 5 ([Mianjy et al., 2018])

All local minima of Problem (4) are equalized, i.e. if U *is a local optimum, then* $\|\mathrm{u}_i\| = \|\mathrm{u}_j\| \; \forall i, j \in [r]$.

- Dropout tends to give equal weights to all hidden nodes
- No matter how small the dropout rate – all local minima become equalized.

Implicit Bias in Local Optima – Illustration

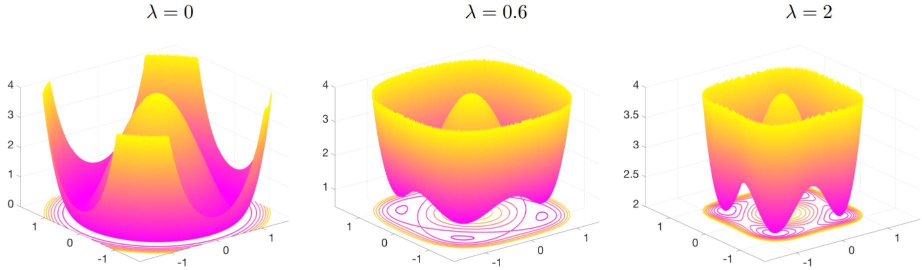


Figure: Optimization landscape for a single hidden-layer linear autoencoder network with dropout, for different regularization parameter λ.

- (Middle) All local minima are global, and are equalized, i.e. the weights are parallel to $(\pm 1, \pm 1)$.
- (Right) As λ increases, global optima shrink further.

Strict Saddle Point/Property

Definition 6 (Strict saddle point/property)

Let $f : \mathcal{U} \to \mathbb{R}$ be a twice differentiable function and let $U \in \mathcal{U}$ be a critical point of f.
Then, U is a **strict saddle point** of f if the Hessian of f at U has at least one negative eigenvalue, i.e. $\lambda_{\min}\left(\nabla^2 f(\mathrm{U})\right) < 0$.
Furthermore, f satisfies **strict saddle property** if all saddle points of f are strict saddle.

- Strict saddle property ensures that for any critical point U that is not a local optimum, the Hessian has a significant negative eigenvalue.
- SGD can escape saddle points and converge to a local minimum [Ge et al., 2015].

Landscape Properties

- For the special case of no dropout (i.e. $\lambda = 0$), Problem (4) has been shown to have no spurious local minima and satisfy strict saddle property ([Baldi and Hornik, 1989, Jin et al., 2017]).
- Question: Can the regularizer induced by dropout potentially introduce new spurious local minima and/or degenerate saddle points?
- The answer is no, at least when the dropout rate is sufficiently small.

Landscape properties

Theorem 7 ([Mianjy et al., 2018])

Let $r := \mathrm{Rank}(M)$. Assume that $d_1 \leqslant d_0$ and that the regularization parameter satisfies $\lambda < \frac{r\lambda_r(\mathrm{M})}{\left(\sum_{i=1}^{r} \lambda_i(\mathrm{M})\right) - r\lambda_r(\mathrm{M})}$. Then it holds for Problem (4) that

1. all local minima are global,

2. all saddle points are strict saddle points.

- The theorem guarantees that any critical point U that is not a global optimum is a strict saddle point.
- This property allows SGD to escape such saddle points.

Prook sketch

Lemma 8

All critical points of Problem (4) that are not equalized, are strict saddle points.

Lemma 9

Let $r := \mathrm{Rank}(M)$. *Assume that* $d_1 \leqslant d_0$ *and* $\lambda < \frac{r\lambda_r}{\sum_{i=1}^{p}(\lambda_i - \lambda_r)}$. *Then all equalized local minima are global. All other equalized critical points are strict saddle points.*

- Theorem 5 and lemma 8 show that non-equalized critical points are not local optima, they are strict saddle points.
- If λ is chosen appropriately, then all critical points that are not global optimum, are strict saddle points.

Summary

- Dropout is a popular regularization with limited understanding.
- Instantiate explicit forms of regularizers due to Dropout and how they provide capacity control in various machine learning problems:
  - Gaussian matrix sensing
  - Matrix completion
  - Deep learning
- For dropout problem (4) with sufficiently small dropout rate:
  - All local minima are equalized
  - All local minima are global
  - All saddle points are non-degenerate

References I

Arora, R., Bartlett, P., Mianjy, P., and Srebro, N. (2021).
Dropout: Explicit forms and capacity control.
In *International Conference on Machine Learning*, pages 351–361. PMLR.

Baldi, P. and Hornik, K. (1989).
Neural networks and principal component analysis: Learning from examples without local minima.
Neural networks, 2(1):53–58.

Caruana, R., Lawrence, S., and Giles, L. (2001).
Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping.
Advances in neural information processing systems, pages 402–408.

Foygel, R., Salakhutdinov, R., Shamir, O., and Srebro, N. (2011).
Learning with the weighted trace-norm under arbitrary sampling distributions.
arXiv preprint arXiv:1106.4251.

Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015).
Escaping from saddle points—online stochastic gradient for tensor decomposition.
In *Conference on learning theory*, pages 797–842. PMLR.

Ioffe, S. and Szegedy, C. (2015).
Batch normalization: Accelerating deep network training by reducing internal covariate shift.
In *International conference on machine learning*, pages 448–456. PMLR.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017).
How to escape saddle points efficiently.
In *International Conference on Machine Learning*, pages 1724–1732. PMLR.

References II

Mianjy, P., Arora, R., and Vidal, R. (2018).
On the implicit bias of dropout.
In *International Conference on Machine Learning*, pages 3540–3548. PMLR.

Neyshabur, B., Tomioka, R., and Srebro, N. (2015).
Norm-based capacity control in neural networks.
In *Conference on Learning Theory*, pages 1376–1401. PMLR.

Salakhutdinov, R. and Srebro, N. (2010).
Collaborative filtering in a non-uniform world: Learning with the weighted trace norm.
arXiv preprint arXiv:1002.2780.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014).
Dropout: A simple way to prevent neural networks from overfitting.
Journal of Machine Learning Research, 15(56):1929–1958.