Improved Dropout for Shallow and Deep Learning

[1]Zhe Li, [2]Boqing Gong, [1]Tianbao Yang

[1]The University of Iowa, [2]University of Central Florida

Main contribution

- Proposed a multinomial dropout for shallow learning.
- Demonstrated that this proposed distribution-dependent dropout leads to a faster convergence and a smaller generalization error through the risk bound analysis.
- Proposed an efficient evolutional dropout for deep learning.
- Justified the proposed dropouts for both shallow and deep learning empirically.

Problem Setup

- Let $(\mathbf{x}, y)$ denote a feature vector and a label, where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathcal{Y}$.
- Denote by $\mathcal{P}$ the joint distribution of $(\mathbf{x}, y)$ and by $\mathcal{D}$ the marginal distribution of $\mathbf{x}$.
- The goal is to learn a linear prediction function ($f(x) = \mathbf{w}^\top \mathbf{x}$) that minimizes the expected risk (considering loss function $\ell(\cdot, y)$):

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{L}(\mathbf{w}) \triangleq \mathrm{E}_{\mathcal{P}}[\ell(\mathbf{w}^\top \mathbf{x}, y)] \qquad (1)$$

- Denote by $\boldsymbol{\epsilon} \sim \mathcal{M}$ a dropout noise vector of dimension d.
- The corrupted feature vector is given by $\widehat{\mathbf{x}} = \mathbf{x} \circ \boldsymbol{\epsilon}$, where the operator $\circ$ represents the element-wise multiplication.
- Denote by $\widehat{\mathcal{P}}$ the joint distribution of the new data $(\widehat{\mathbf{x}}, y)$ and by $\widehat{\mathcal{D}}$ the marginal distribution of $\widehat{\mathbf{x}}$.
- With the corrupted data, the risk minimization becomes

$$\min_{\mathbf{w}\in\mathbb{R}^d} \widehat{\mathcal{L}}(\mathbf{w}) \triangleq \mathrm{E}_{\widehat{\mathcal{P}}}[\ell(\mathbf{w}^\top(\mathbf{x}\circ\boldsymbol{\epsilon}), y)] \qquad (2)$$

Learning with Multinomial Dropout

Definition 1. A **multinomial dropout** is defined as $\widehat{\mathbf{x}} = \mathbf{x} \circ \boldsymbol{\epsilon}$, where $\epsilon_i = \frac{m_i}{kp_i}, i \in [d]$ and $\{m_1, \ldots, m_d\}$ follow a multinomial distribution $Mult(p_1, \ldots, p_d; k)$ with $\sum_{i=1}^d p_i = 1$ and $p_i \geq 0$.

Proposition 1. f $\ell(z, y) = \log(1 + \exp(-yz))$, then

$$\mathrm{E}_{\widehat{\mathcal{P}}}[\ell(\mathbf{w}^\top \widehat{\mathbf{x}}, y)] = \mathrm{E}_{\mathcal{P}}[\ell(\mathbf{w}^\top \mathbf{x}, y)] + R_{\mathcal{D},\mathcal{M}}(\mathbf{w})$$

where $\mathcal{M}$ denotes the distribution of $\boldsymbol{\epsilon}$ and $R_{\mathcal{D},\mathcal{M}}(\mathbf{w}) = \mathrm{E}_{\mathcal{D},\mathcal{M}}\left[\log \frac{\exp(\mathbf{w}^\top \frac{\mathbf{x}\circ\boldsymbol{\epsilon}}{2}) + \exp(-\mathbf{w}^\top \frac{\mathbf{x}\circ\boldsymbol{\epsilon}}{2})}{\exp(\mathbf{w}^\top \mathbf{x}/2) + \exp(-\mathbf{w}^\top \mathbf{x}/2)}\right]$. Dropout is a data-dependent regularizer

Learning with Multinomial Dropout:

- Give the initial solution $\mathbf{w}_1$.
- Update the model at t^{th} iteration: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(\mathbf{w}_t^\top(\mathbf{x}_t \circ \boldsymbol{\epsilon}_t), y_t)$
- Output the final solution: $\widehat{\mathbf{w}}_n = \frac{1}{n}\sum_{t=1}^n \mathbf{w}_t$

Improved Dropout for Shallow Learning

Risk Bound of $\widehat{\mathbf{w}}_n$ in Expectation

Theorem 1: Let $\mathcal{L}(\mathbf{w})$ be the expected risk of $\mathbf{w}$ defined in (1). Assume $\mathrm{E}_{\widehat{\mathcal{D}}}[\|\mathbf{x}\circ\boldsymbol{\epsilon}\|_2^2] \leq B^2$ and $\ell(z, y)$ is convex and G-Lipschitz continuous. For any $\|\mathbf{w}_*\|_2 \leq r$, by appropriately choosing η, we can have

$$\mathrm{E}[\mathcal{L}(\widehat{\mathbf{w}}_n) + R_{\mathcal{D},\mathcal{M}}(\widehat{\mathbf{w}}_n)] \leq \mathcal{L}(\mathbf{w}_*) + R_{\mathcal{D},\mathcal{M}}(\mathbf{w}_*) + \frac{GBr}{\sqrt{n}}$$

Theoretically, minimizing the term $\mathrm{E}_{\widehat{\mathcal{D}}}[\|\mathbf{x}\circ\boldsymbol{\epsilon}\|_2^2]$ and the relaxed upper bound of term $R_{\mathcal{D},\mathcal{M}}(\mathbf{w}_*)$ yields the optimal sampling probabilities:

$$p_i^* = \frac{\sqrt{\mathrm{E}_{\mathcal{D}}[x_i^2]}}{\sum_{j=1}^d \sqrt{\mathrm{E}_{\mathcal{D}}[x_j^2]}}, i = 1, \ldots, d \qquad (3)$$

Practically, we use the empirical second-order statistics to compute the probabilities:

$$p_i = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^n [[\mathbf{x}_j]]_i^2}}{\sum_{i'=1}^d \sqrt{\frac{1}{n}\sum_{j=1}^n [[\mathbf{x}_j]]_{i'}^2}}, i = 1, \ldots, d \qquad (4)$$

Improved Dropout for Deep Learning

Let $X^l = (\mathbf{x}_1^l, \ldots, \mathbf{x}_m^l)$ denote the outputs of the l^{th} layer for a mini-batch of m examples, calculate the probabilities for dropout by

$$p_i^l = \frac{\sqrt{\frac{1}{m}\sum_{j=1}^m [[\mathbf{x}_j^l]]_i^2}}{\sum_{i'=1}^d \sqrt{\frac{1}{m}\sum_{j=1}^m [[\mathbf{x}_j^l]]_{i'}^2}}, i = 1, \ldots, d \qquad (5)$$

Evolutional Dropout for Deep Learning

Input: a batch of outputs of a layer: $X^l = (\mathbf{x}_1^l, \ldots, \mathbf{x}_m^l)$ and dropout level parameter $k \in [0, d]$

Output: $\widehat{X}^l = X^l \circ \Sigma^l$

Compute sampling probabilities by (5)

For $j = 1, \ldots, m$

Sample $\mathbf{m}_j^l \sim Mult(p_1^l, \ldots, p_d^l; k)$

Construct $\boldsymbol{\epsilon}_j^l = \frac{\mathbf{m}_j^l}{k\mathbf{p}^l} \in \mathbb{R}^d$, where $\mathbf{p}^l = (p_1^l, \ldots, p_d^l)^\top$

Let $\Sigma^l = (\boldsymbol{\epsilon}_1^l, \ldots, \boldsymbol{\epsilon}_m^l)$ and compute $\widehat{X}^l = X^l \circ \Sigma^l$

Figure 1: Evolutional Dropout applied to a layer over a mini-batch

Remark 1. Similar to Batch Normalization, evolutional dropout can also address the internal covariace shift issue by adapting the sampling probabilities to the evolving distribution of layers' output.

Experimental Results

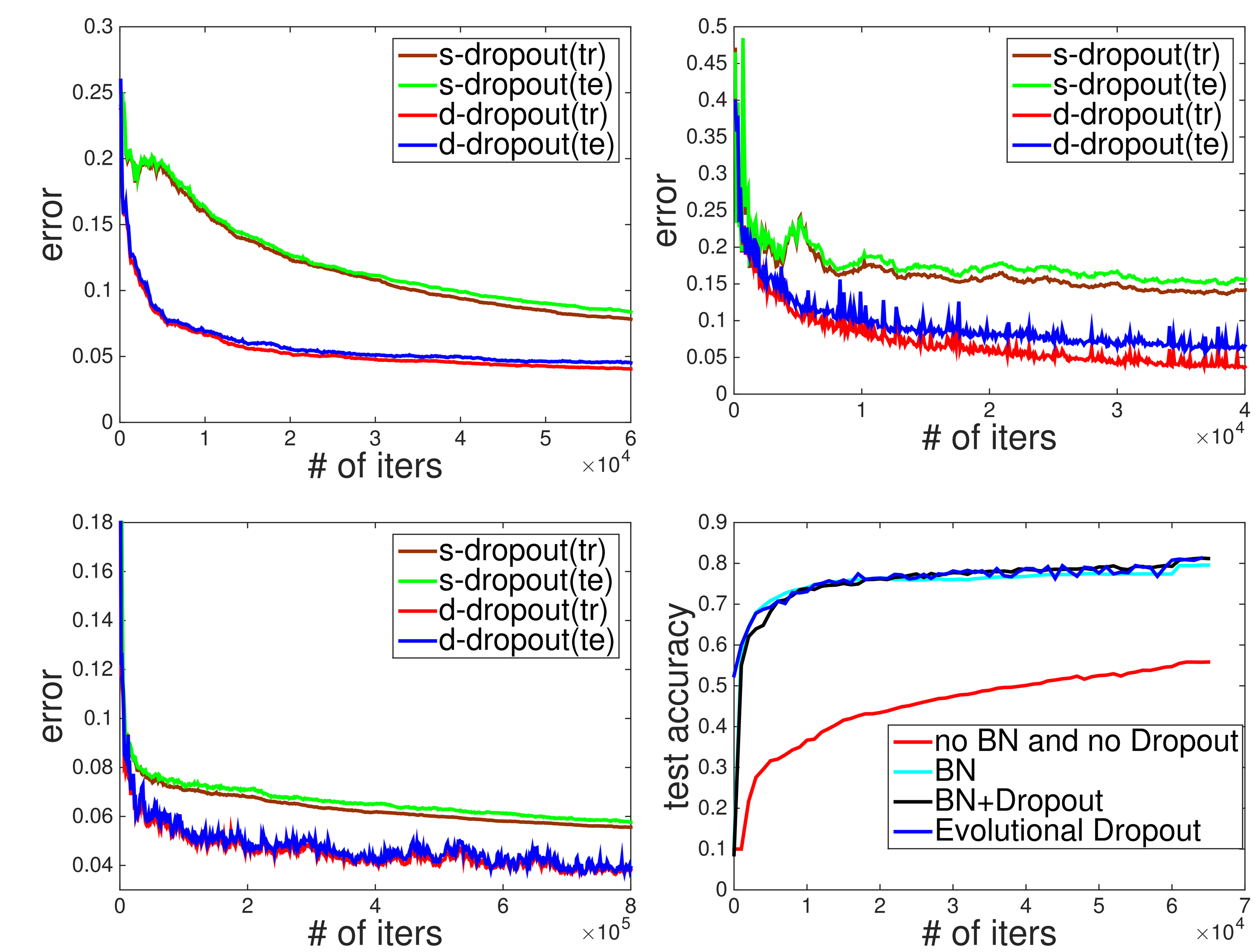

Figure 2: data-dependent dropout vs. standard dropout on three datasets (real-sim, news20 and RCV1) for logistic regression; Lower Right Corner: Evolutional dropout vs BN on CIFAR-10.

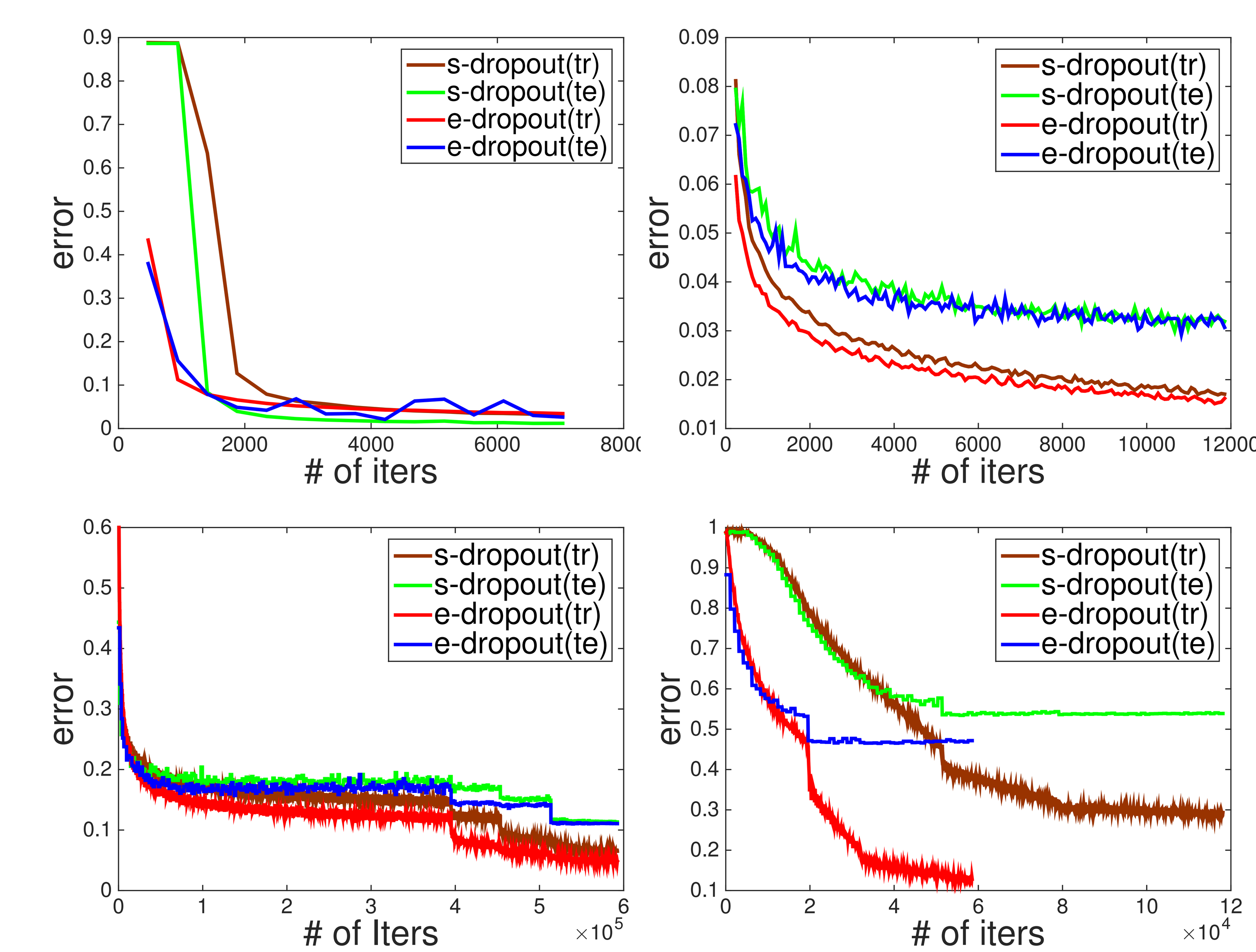

Figure 3: Evolutional dropout vs. standard dropout on four benchmark datasets (MNIST, SVHN, CIFAR-10 and CIFAR-100) for deep learning