Learning Neural Networks with Adaptive Regularization

Han Zhao*†, Yao-Hung Hubert Tsai*†, Ruslan Salakhutdinov†, Geoffrey J. Gordan†‡
†Carnegie Mellon University, ‡Microsoft Research Montreal (*equal contribution)
https://github.com/yaohungt/Adaptive-Regularization-Neural-Network

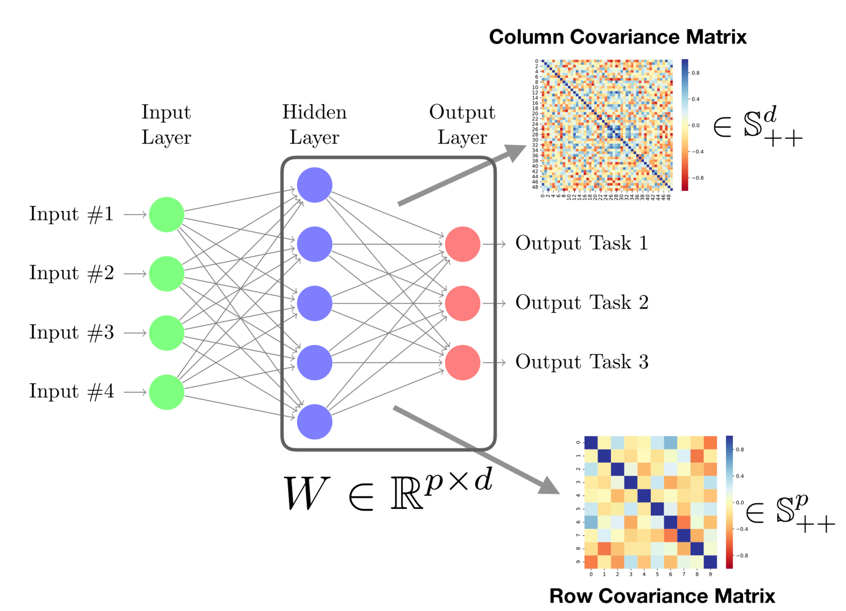
Overview

Q: How to effectively regularize neural networks given few available training data?

- **AdaReg**: an approximate empirical Bayes method for regularizing NN training on small datasets
- Learn the preconditioning matrix adaptively from the data
- Significant improvement in terms of spectral norm and stable rank, leading to smaller generalization error

Motivation

Two-layer Neural Network



Hidden layer $\mathbf{h}\in\mathbb{R}^p$ with output $\hat y\in\mathbb{R}$ for regression

$$\hat y = \mathbf{a}^\top\mathbf{h},\ \mathbf{h}=g(W\mathbf{x}),\ W\in\mathbb{R}^{p\times d}$$

Loss function

$$\ell(W,\mathbf{a}) = \frac{1}{2}(\hat y - y)^2$$

Update for weight matrix W

$$W \leftarrow W - \gamma(\hat y - y)(\mathbf{a}\circ\mathbf{h}')\mathbf{x}^\top$$

$\mathbf{h}'$: component-wise derivative of $\mathbf{h}$ w.r.t. its input

$(\hat y - y)(\mathbf{a}\circ\mathbf{h}')\mathbf{x}^\top$: gradient is always rank-1

→ rows/columns of W are correlated!

A: Update each row/column by taking information from other rows/columns into consideration!

Prior of Parameters

Multivariate Normal Distribution

$$W\sim\mathcal{MN}(\mathbf{0}_{p\times d},\Sigma_r,\Sigma_c)$$

$\Sigma_r\in\mathbb{S}^p_{++},\Sigma_c\in\mathbb{S}^d_{++}$: row and column covariance matrices

$$p(W\mid\Sigma_r,\Sigma_c) = \frac{\exp\left(-\mathrm{Tr}(\Sigma_r^{-1}W\Sigma_c^{-1}W^\top)/2\right)}{(2\pi)^{pd/2}\det(\Sigma_r)^{d/2}\det(\Sigma_c)^{p/2}}$$

Determine the Parameters in the Prior

1) Empirical Bayes: estimate the parameters of the prior from data

$$\hat\Sigma_r,\hat\Sigma_c = \arg\max_{\Sigma_r,\Sigma_c} p(\mathcal{D}\mid\Sigma_r,\Sigma_c) = \arg\max_{\Sigma_r,\Sigma_c}\int p(\mathcal{D}\mid W)\,p(W\mid\Sigma_r,\Sigma_c)\,dW$$

Intractable.

2) Iterative maximization of the joint distribution: sequence of MAP

$$W^{(t+1)} = \arg\max_W \log p(\mathcal{D}\mid W) + \log p(W\mid\Sigma_r^{(t)},\Sigma_c^{(t)})$$

$$\Sigma_r^{(t+1)} = \arg\max_{\Sigma_r} \log p(\mathcal{D}\mid W^{(t+1)}) + \log p(W^{(t+1)}\mid\Sigma_r,\Sigma_c^{(t)})$$

$$\Sigma_c^{(t+1)} = \arg\max_{\Sigma_c} \log p(\mathcal{D}\mid W^{(t+1)}) + \log p(W^{(t+1)}\mid\Sigma_r^{(t+1)},\Sigma_c)$$

Approximate empirical Bayes and tractable.

Optimization

$$\min_{W,\mathbf{a}}\min_{\Omega_r,\Omega_c}\ \frac{1}{2n}\sum_{i\in[n]}(\hat y(\mathbf{x}_i;W,\mathbf{a})-y_i)^2 + \lambda\|\Omega_r^{1/2}W\Omega_c^{1/2}\|_F^2 - \lambda\left(d\log\det(\Omega_r)+p\log\det(\Omega_c)\right)$$

subject to $uI_p\preceq\Omega_r\preceq vI_p,\ uI_d\preceq\Omega_c\preceq vI_d$

preconditioned matrix: $\|\Omega_r^{1/2}W\Omega_c^{1/2}\|_F^2 = \|(\Omega_r^{1/2}\otimes\Omega_c^{1/2})\mathrm{vec}(W)\|_2^2$

$\Omega_r := \Sigma_r^{-1},\Omega_c := \Sigma_c^{-1}$: precision matrices

Solving Ω_r

$$\min_{\Omega_r}\ \mathrm{Tr}(\Omega_r W\Omega_c W^T) - d\log\det(\Omega_r) + \mathbb{I}_{\mathcal{C}}(\Omega_r)$$ with $\mathcal{C}:=\{A\in\mathbb{S}^p_{++}\mid uI_p\preceq A\preceq vI_p\}$

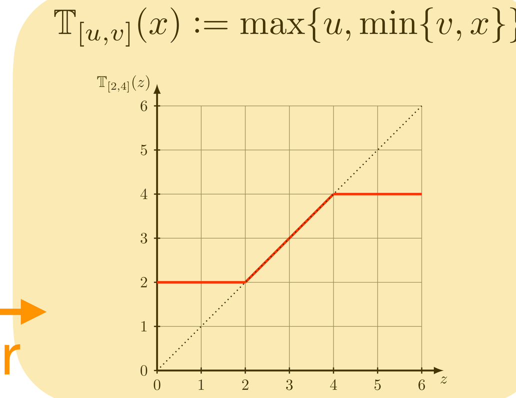
→ $0\in\partial\left(\frac{1}{d}\mathrm{Tr}(\Omega_r W\Omega_c W^T) - \log\det(\Omega_r)+\mathbb{I}_{\mathcal{C}}(\Omega_r)\right) = W\Omega_c W^T/d - \Omega_r^{-1} + \mathcal{N}_{\mathcal{C}}(\Omega_r)$ → $W\Omega_c W^T/d - \Omega_r^{-1}\in\mathcal{N}_{\mathcal{C}}(\Omega_r^{-1})$

→ Optimal Ω_r^{-1} is the Euclidean projection of $W\Omega_c W^\top/d$ onto $\mathcal{C}$

Algorithm 1 Block Coordinate Descent for Adaptive Regularization

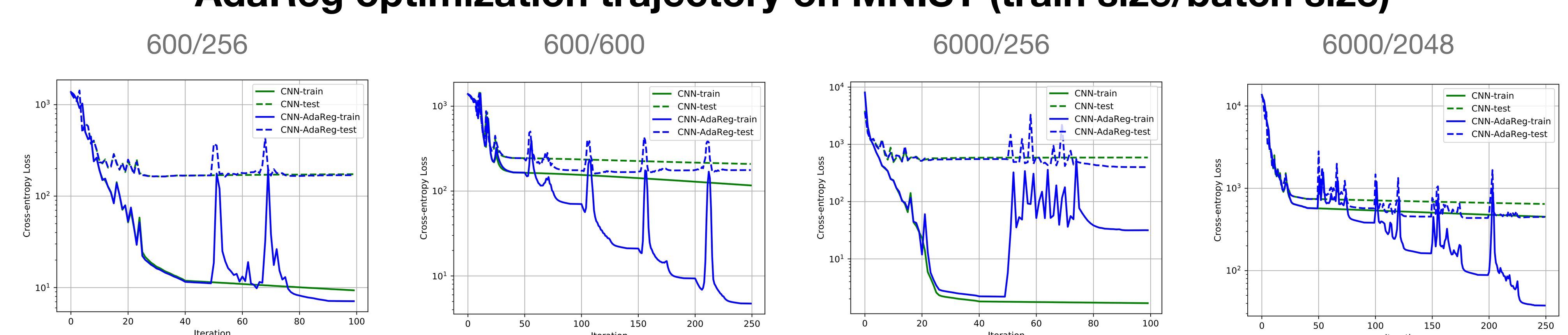
```
Input: Initial value φ^(0) := {a^(0), W^(0)}, Ω_r^(0) ∈ S^p_{++} and Ω_c^(0) ∈ S^d_{++}, first-order optimization algorithm 𝔄.
  for t = 1, ..., ∞ until convergence do
    Fix Ω_r^(t-1), Ω_c^(t-1), optimize φ^(t) by backpropagation and algorithm 𝔄
    Ω_r^(t) ← InvThreshold(W^(t) Ω_c^(t-1) W^(t)T, d, u, v)
    Ω_c^(t) ← InvThreshold(W^(t)T Ω_r^(t) W^(t), p, u, v)
  end for
  procedure InvThreshold(Δ, m, u, v)
    Compute SVD: Q diag(r) Q^T = SVD(Δ)
    Hard thresholding r' ← T_[u,v](m/r)
    return Q diag(r') Q^T
  end procedure
```

thresholding operator: $\mathbb{T}_{[u,v]}(x) := \max\{u,\min\{v,x\}\}$

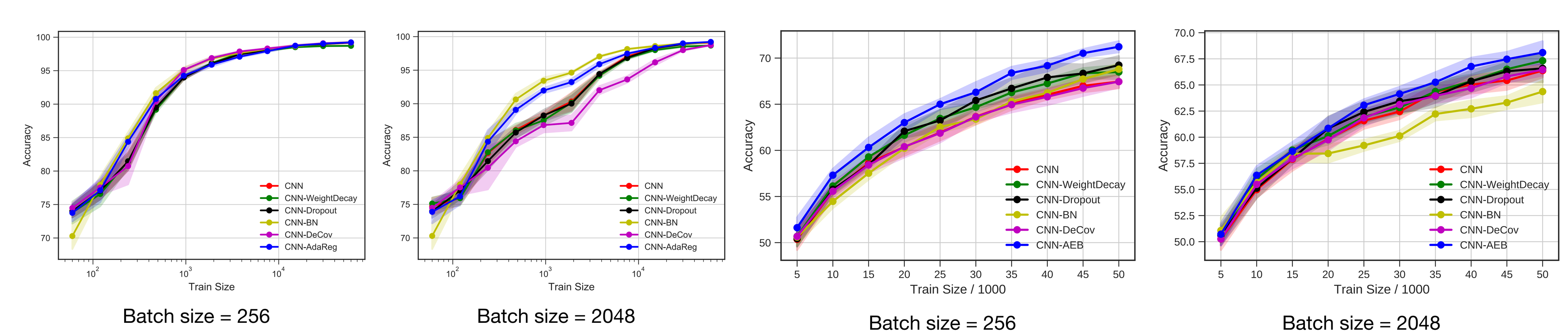


Experiments

AdaReg optimization trajectory on MNIST (train size/batch size)



MNIST **CIFAR-10**



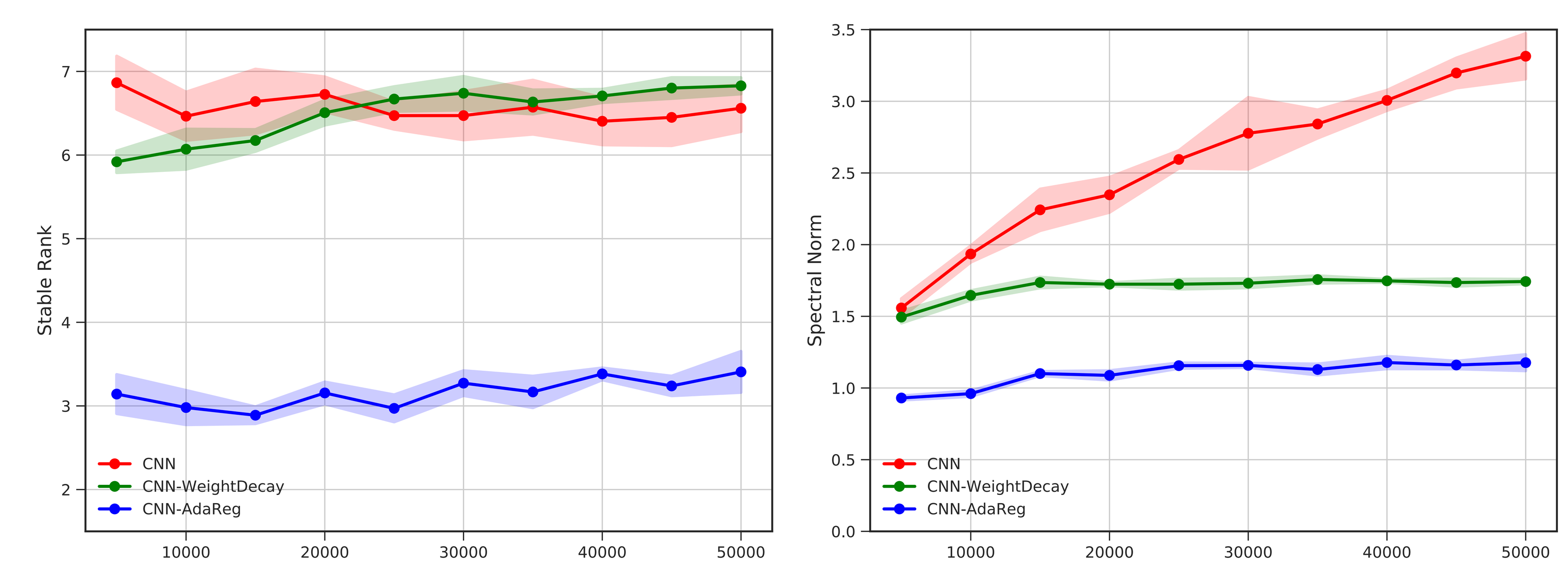
Batch size = 256 Batch size = 2048 Batch size = 256 Batch size = 2048

Stable Rank and Spectral Norm for Generalization Error (Neyshabur et al. ICLR'17)

Generalization Error $= O\left(\sqrt{\prod_{j=1}^{L}\|W_j\|_2^2\sum_{j=1}^{L}\mathrm{srank}(W_j)/n}\right)$

$\mathrm{srank}(W) := \|W\|_F^2/\|W\|_2^2$

$1\le\mathrm{srank}(W)\le\mathrm{rank}(W)$



CIFAR-10