# ECC: Platform-Independent Energy-Constrained Deep Neural Network Compression via a Bilinear Regression Model

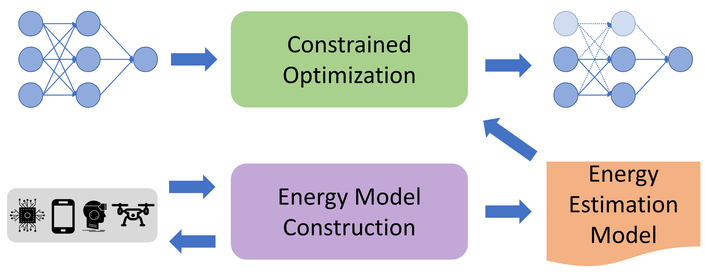
Haichuan Yang[1], Yuhao Zhu[1], Ji Liu[2,1]
[1]University of Rochester, [2]Kwai AI Lab at Seattle

## Motivation


- DNNs are increasingly being deployed on energy-constrained devices;
- Here, we compressed DNN to meet given energy budgets across different platforms.

## Overview

ECC framework overview:




The overall objective:

$$\min_{\mathcal{W}} \ell(\mathcal{W})$$

DNN parameters ← $\mathcal{W}$ ; $\ell(\mathcal{W})$ → DNN training Loss

$$\text{s.t. } \phi\left(\mathbf{w}^{(j)}\right) \le s_j, \quad j = 1,2,\dots|\mathcal{U}|$$

$$\mathcal{E}(\mathbf{s}) \le E_{\text{budget}}$$

$\phi\left(\mathbf{w}^{(j)}\right)$: #nonzero channels; $j = 1,2,\dots|\mathcal{U}|$: convolution layers; $\mathcal{E}(\mathbf{s})$: energy consumption

## Energy Estimation Model

**Bilinear Model**: approximate the energy consumption. Assumption: For each layer $j$, the energy estimation can be modeled by the interaction between $s_j$ and $s_{j+1}$:

$$\hat{\mathcal{E}}(\mathbf{s}) = a_0 + \sum_{j=1}^{|\mathcal{U}|} a_j s_j s_{j+1}$$

Estimate the energy model via a data-driven approach:

$$\min_{a_0,a_a,\dots,a_{|\mathcal{U}|}\ge 0} \mathrm{E}_{\mathbf{s}}\left[\left(\hat{\mathcal{E}}(\mathbf{s}) - \mathcal{E}(\mathbf{s})\right)^2\right]$$

## Optimization Algorithm

**ADMM reformulation:**

$$\min_{\mathcal{W},\mathbf{s}} \max_{z\ge 0,\mathbf{y}\ge \mathbf{0}} \ell(\mathcal{W}) + \underbrace{\mathcal{L}_1(\mathcal{W},\mathbf{s},\mathbf{y})}_{\phi(\mathbf{w}^{(j)}) \le s_j} + \underbrace{\mathcal{L}_2(\mathbf{s},z)}_{\hat{\mathcal{E}}(\mathbf{s}) \le E_{\text{budget}}}$$

$$\mathcal{L}_1 := \frac{\rho_1}{2}\sum_j \left[\phi\left(w^{(j)}\right) - s_j\right]_+^2 + \sum_j y_j\left(\phi\left(w^{(j)}\right) - s_j\right)$$

$$\mathcal{L}_2 := \frac{\rho_2}{2}\left[\hat{\mathcal{E}}(\mathbf{s}) - E_{\text{budget}}\right]_+^2 + z\left(\hat{\mathcal{E}}(\mathbf{s}) - E_{\text{budget}}\right)$$

Update $\mathcal{W}$:

$$\mathcal{W} \leftarrow \operatorname*{argmin}_{\mathcal{W}} \ell(\mathcal{W}) + \mathcal{L}_1(\mathcal{W},\mathbf{s},\mathbf{y})$$

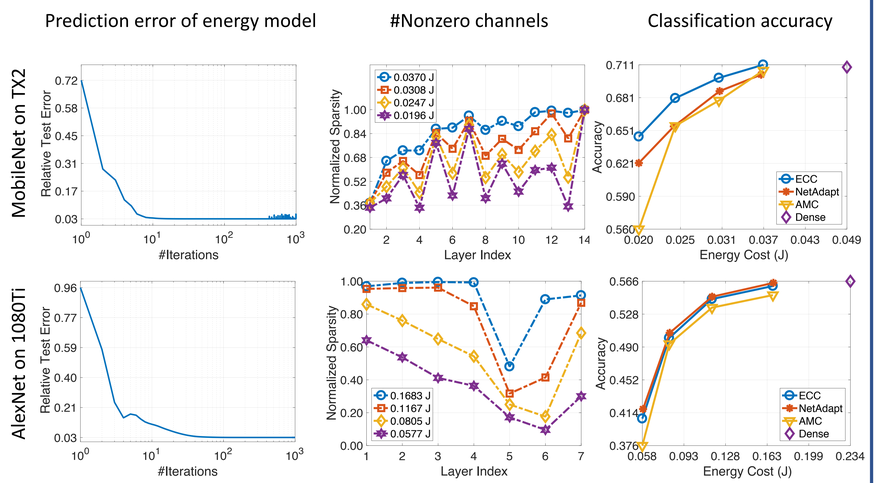
Update $\mathbf{s}$:

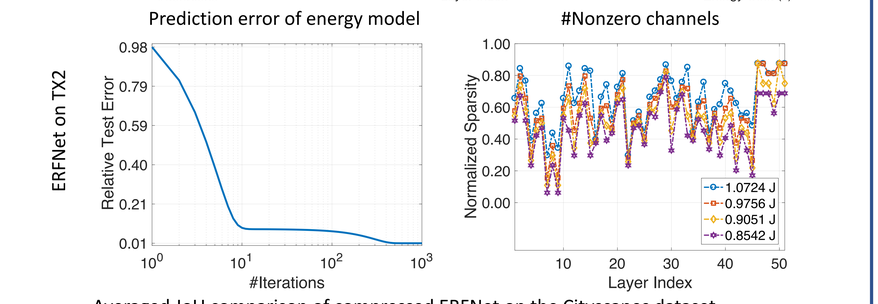
$$\mathbf{s} \leftarrow \mathbf{s} - \beta\nabla_{\mathbf{s}}\left(\mathcal{L}_1(\mathcal{W},\mathbf{s},\mathbf{y}) + \mathcal{L}_2(\mathbf{s},z)\right)$$

Update $\mathbf{y}, z$:

$$y_j \leftarrow \left[y_j + \rho_1\left(\phi\left(w^{(j)}\right) - s_j\right)\right]_+$$
$$z \leftarrow \left[z + \rho_2\left(\hat{\mathcal{E}}(\mathbf{s}) - E_{\text{budget}}\right)\right]_+$$

## Experiment Results




Averaged-IoU comparison of compressed ERFNet on the Cityscapes dataset

| Methods / Energy Budget | GTX 1080 Ti | | | | | Jetson TX2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1.3843 J | 1.4451 J | 1.5238 J | 1.6519 J | 1.9708 J | 0.8542 J | 0.9051 J | 0.9756 J | 1.0724 J | 1.3213 J |
| Dense | - | - | - | - | 0.722 | - | - | - | - | 0.722 |
| **ECC** | **0.6713** | **0.6830** | **0.7007** | **0.7163** | - | **0.6733** | **0.6914** | **0.7017** | **0.7183** | - |
| NetAdapt [39] | 0.6361 | 0.6523 | 0.6865 | 0.7114 | - | 0.6567 | 0.6708 | 0.6916 | 0.7084 | - |
| AMC [11] | 0.6340 | 0.6374 | 0.6749 | 0.6992 | - | 0.6344 | 0.6491 | 0.6685 | 0.6976 | - |

## Conclusion

- Proposed a bilinear energy consumption model of DNN inference;
- Leveraged the energy estimation model to formulate DNN compression as constrained optimization;
- Outperformed state of the arts.