Efficient Ensemble Sparse Convolutional Neural Network with Dynamic Batch Size

Shen Zheng, Liwei Wang & Gaurav Gupta,
Department of Mathematics, Wenzhou Kean University

Introduction

- Background

CONVOLUTIONAL NEURAL NETWORK IS HOT!

LeNet -> AlexNet -> VGG -> GoogLeNet -> ResNet->....

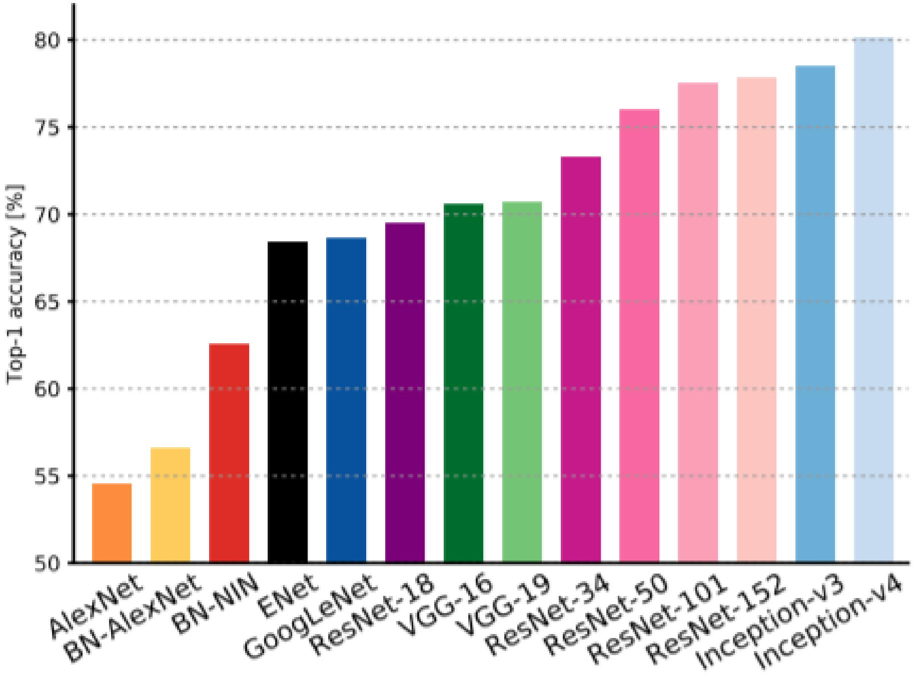
(LeCun,1998) (Krizhevsky, 2012) (Simonyan, 2014) (Szegedy 2014) (He, 2015)

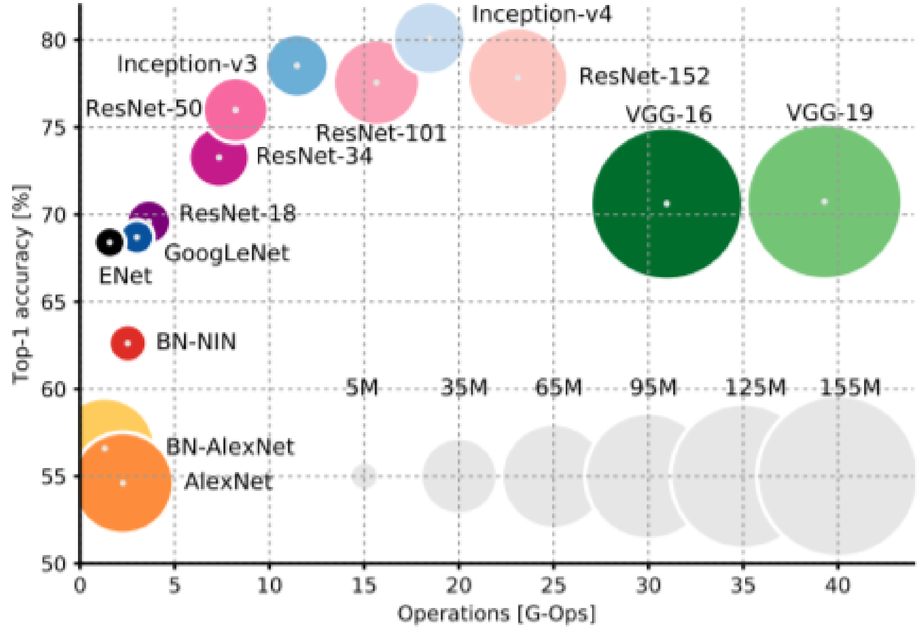
Visual Recognition, Speech Recognition, and Natural Language Processing

Introduction

- Problems

SLOW FOR COMPUTATION!





Alfredo Canziani, Adam Paszke, Eugenio Culurciello, 2017

Introduction

- Existing Solutions/ Related Work

1. Network Pruning & Convolutional accelerator

->FFT Conv. (Mathieu, 2013)

->Winograd Conv. Operation (Winograd, 1980; Lavin, 2015)

->Pruning & Retraining (Liu, 2016)

->Replace Conv. with Winograd Conv. Layers (Li, 2017)

-> Move ReLU into Winograd domain (Liu, 2018)

2. Activation Functions (not in this pre.)

Introduction

- Existing Solutions/ Related Work

3. Batch Sizes

->Large Batch Size” : Generalization Gap ” (Krizhevsky, 2014)

->” Sharp Minima ” (Keskar, 2016)

-> ” Generalization Gap” : Insufficient updates (Hoffer, 2016)

->Increasing the Learning Rate & Momentum (Smith, 2017)

-> Learning Rate Warm Up (Goyal, 2017)

Introduction

- Our Solutions

Weighted Average Stacking &

Network Pruning &

Winograd-ReLU Convolution &

DYNAMIC BATCH SIZE ALGORITHM

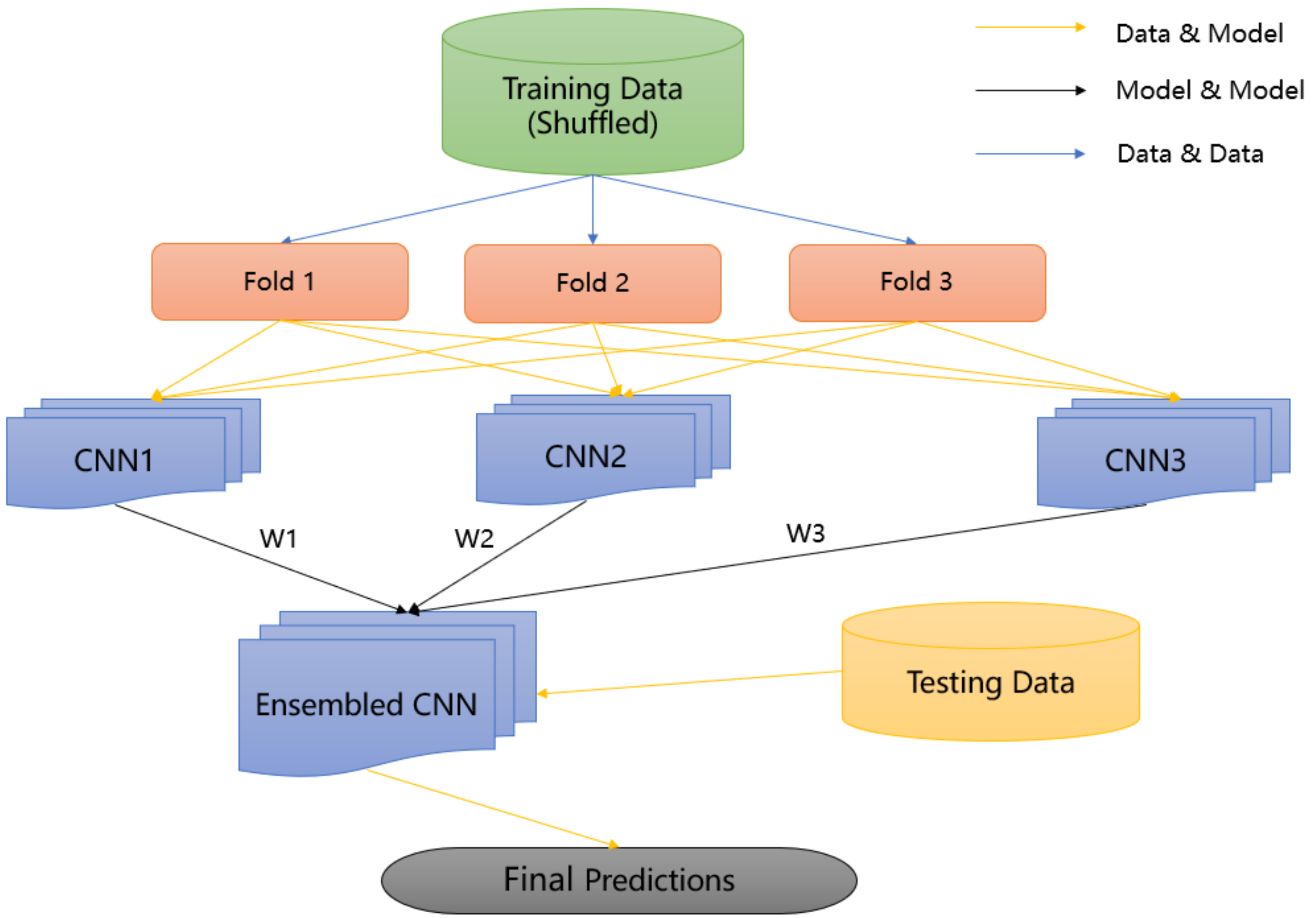
Methodology

- Model Architecture

-> Separate Train & Test

-> Stacked CNNs

-> Weighted Average



Methodology

- Training Procedures

  – Warm Up the Learning Rate gradually from 0.01 to 0.02, for the beginning 10 % of the total epochs.
  – Increase the learning rate by a multiplier of 2 every n epoch until validation accuracy falls, keeping momentum coefficient fixed. Linearly Scale the batch size to the learning rate.
  – Increase the momentum coefficient, keeping learning rate fixed. Scale the batch size to momentum coefficient.
  – Stop the above action until reaching maximum batch size, which is determined by three restrictions: GPU memory limits, non-decreasing validation accuracy and linear scaling rule constraints ($B << N/10$)
  – If validation accuracy does not improve for five consecutive epochs, decrease the learning rate by a multiplier of 0.1.

Methodology

- Algorithms

Algorithm 1 Mini-Batch SGDM with Dynamic Batch Size.

Require: Learning rate η, batch size B ,momentum coefficient m, numbers of steps T, number of data points N , loss function $f(\theta)$.

for $t \in [1, T]$ **do**

$\quad B_{\max} = \text{Round\_\&\_Min}\left(B_{\text{capacity}}\ , \frac{N}{10}\right)$

$\quad B_{min} = B_0$

$\quad B = \text{Round\_\&\_Clip}\left(\frac{\eta(1-m_0)}{\eta_0(1-m)} B_0, B_{\min}, B_{\max}\right)$

$\quad B = Stepwise(B)$

$\quad g_t = \frac{1}{B} \sum_{i=1}^{B} \nabla f(\theta_i)$

$\quad v_t = m v_{t-1} + \eta g_t$

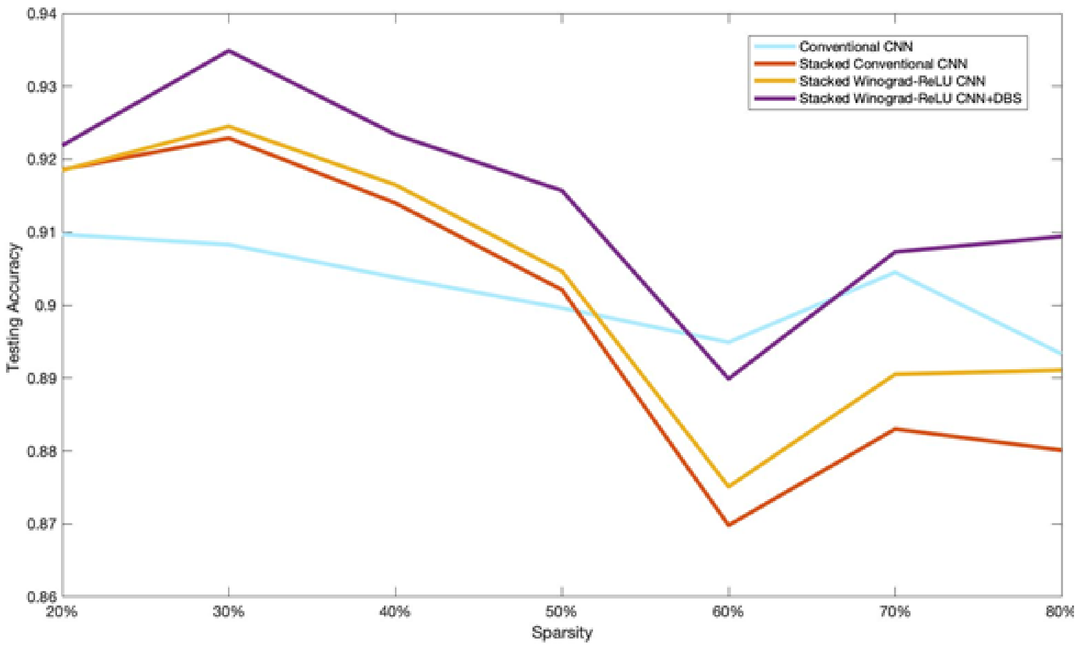
$\quad \theta_t = \theta_{t-1} - v_t$

end for

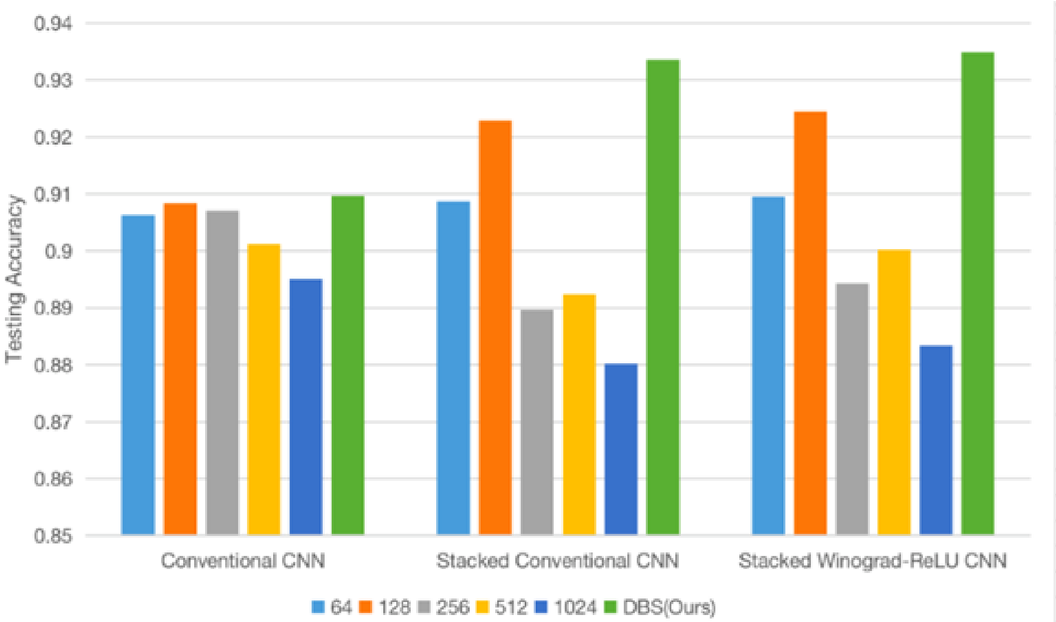
return B, θ_t

Experiment

- AlexNet + FASHION-MNIST (Xiao, 2017)



(a)



(b)

Experiment Results

- AlexNet + FASHION-MNIST

	C-CNN		SC-CNN		SWR-CNN	
Batch Size	Time	Speed	Time	Speed	Time	Speed
64	20	0.85x	53	0.32x	24	0.71x
128	12	1.42x	37	0.46x	17	1.00x
256	7	2.43x	28	0.61x	13	1.31x
512	5	3.4x	22	0.77x	10	1.70x
1024	3	5.67x	21	0.81x	8	2.13x
DBS(Ours)	6	2.83x	24	0.71x	11	1.55x

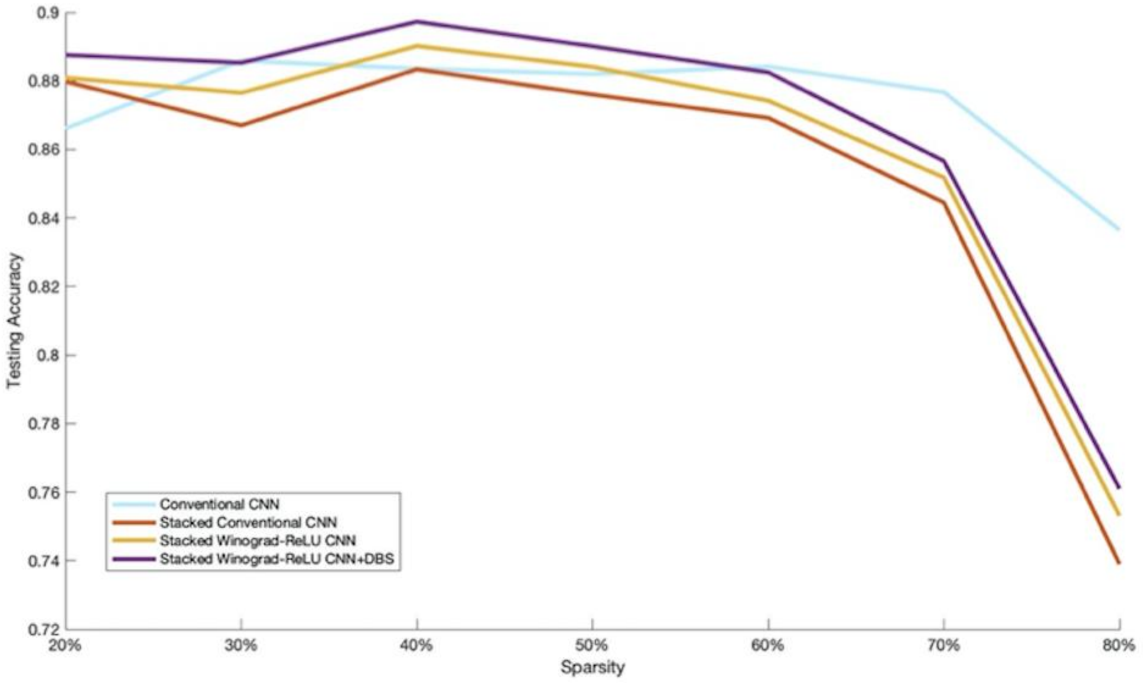
Table 2: Computational Speed for Different AlexNet models on FASHION-MNIST with Different Batch Sizes

	C-CNN		SC-CNN		SWR-CNN		SWR-CNN + DBS (ours)	
Sparsity	Time	Speed	Time	Speed	Time	Speed	Time	Speed
20%	12	1.42x	37	0.46	17	1.00x	11	1.55x
30%	12	1.42x	37	0.46	17	1.00x	11	1.55x
40%	12	1.42x	36	0.47	16	1.06x	11	1.55x
50%	11	1.55x	37	0.46	17	1.00x	11	1.55x
60%	11	1.55x	37	0.46	16	1.06x	10	1.7x
70%	11	1.55x	36	0.47	16	1.06x	10	1.7x
80%	11	1.55x	37	0.46	16	1.06x	10	1.7x
Overall	11.42	1.49x	36.71	0.46x	16.43	1.03x	10.57	1.61x

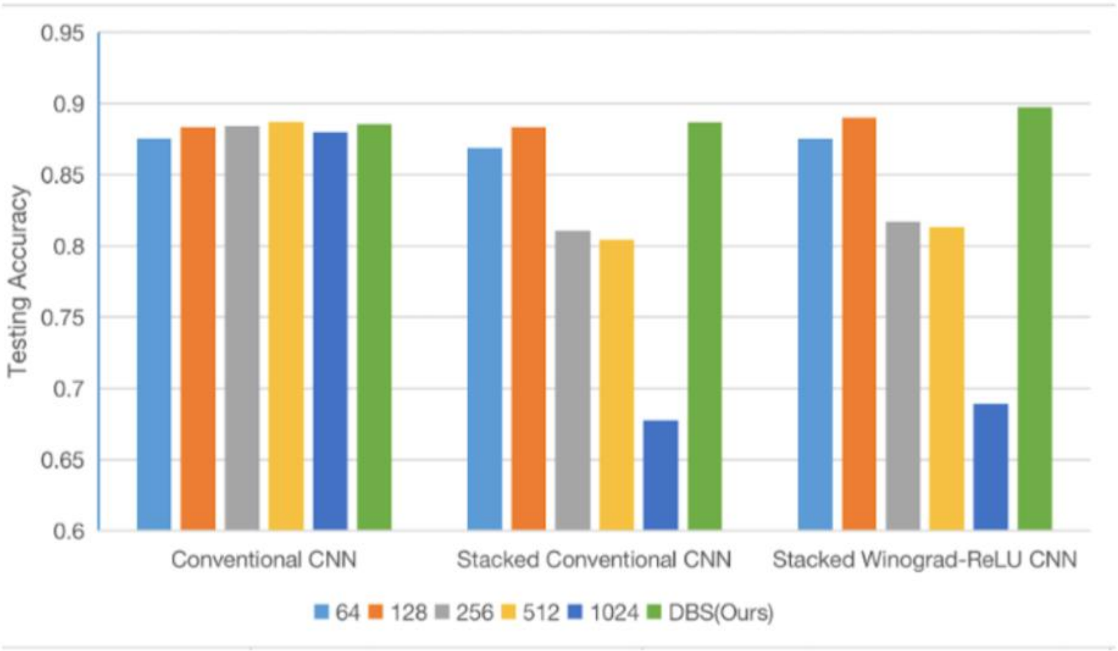
Table 1: Computational Speed for Different AlexNet models on FASHION-MNIST at Different Sparsity

Experiment Results

- VGG + CIFAR10 (Krizhevsky, 2009)



(a)



(b)

Experiment Results

- VGG + CIFAR10

	C-CNN		SC-CNN		SWR-CNN	
Batch Size	Time	Speed	Time	Speed	Time	Speed
64	28	0.93x	62	0.42x	28	0.93x
128	18	1.44x	40	0.65x	18	1.44x
256	13	2.00x	32	0.81x	15	1.73x
512	11	2.36x	28	0.93x	13	2.00x
1024	9	2.89x	27	0.96x	12	2.17x
DBS(Ours)	13	2.00x	29	0.90x	14	1.86x

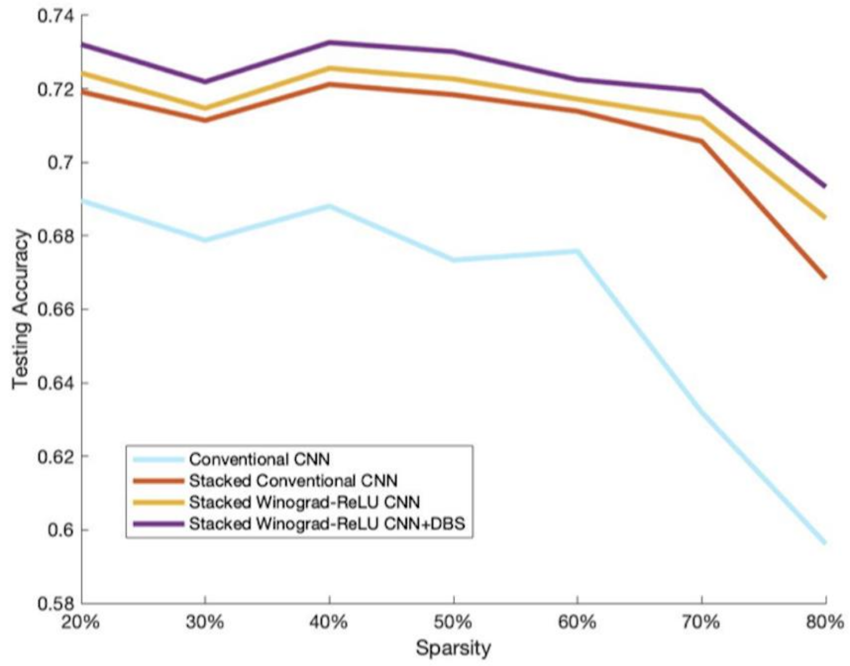
Table 4: Computational Speed for Different VGG models on CIFAR-10 with Different Batch Sizes

	C-CNN		SC-CNN		SWR-CNN		SWR-CNN +DBS (ours)	
Sparsity	Time	Speed	Time	Speed	Time	Speed	Time	Speed
20%	19	1.37x	41	0.63x	18	1.44x	15	1.73x
30%	19	1.37x	41	0.63x	18	1.44x	14	1.86x
40%	18	1.44x	40	0.65x	18	1.44x	14	1.86x
50%	18	1.44x	40	0.65x	18	1.44x	14	1.86x
60%	18	1.44x	40	0.65x	18	1.44x	13	2.00x
70%	18	1.44x	39	0.67x	17	1.53x	13	2.00x
80%	18	1.44x	39	0.67x	17	1.53x	12	2.17x
Overall	18.3	1.42x	40.0	0.65x	17.71	1.47x	13.57	1.92x

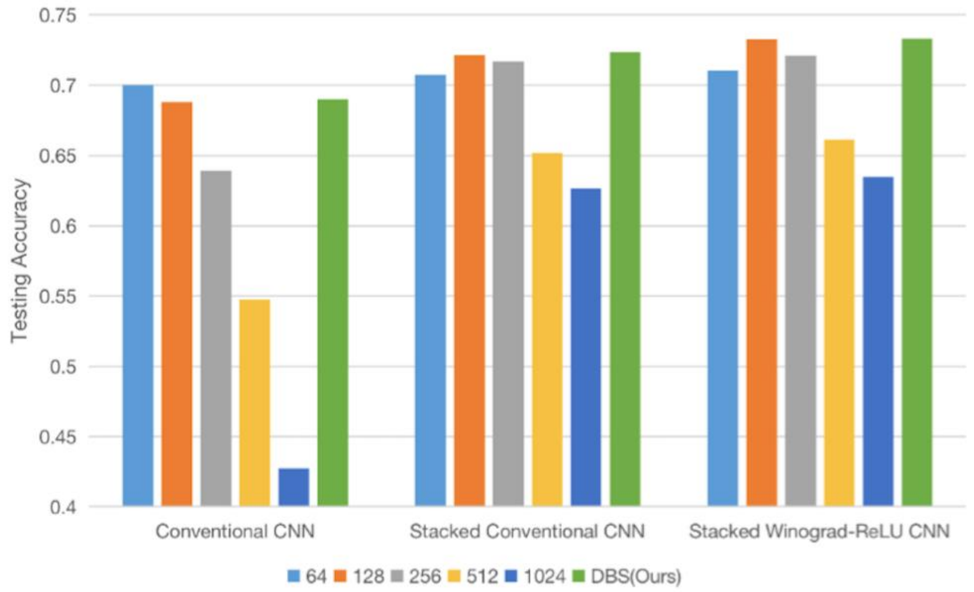
Table 3: Computational speed for different VGG models on CIFAR-10 at different sparsity

Experiment Results

- ResNet + CIFAR100 (Krizhevsky, 2009)



(a)



(b)

Experiment Results

- ResNet + CIFAR100

	C-CNN		SC-CNN		SWR-CNN	
Batch Size	Time	Speed	Time	Speed	Time	Speed
64	49	1.10x	90	0.60x	53	1.02x
128	29	1.86x	49	1.10x	20	2.70x
256	20	2.70x	34	1.59x	14	3.86x
512	14	3.86x	25	2.16x	11	4.91x
1024	12	4.50x	22	2.45x	10	5.40x
DBS(Ours)	18	3.00x	29	1.86x	13	4.15x

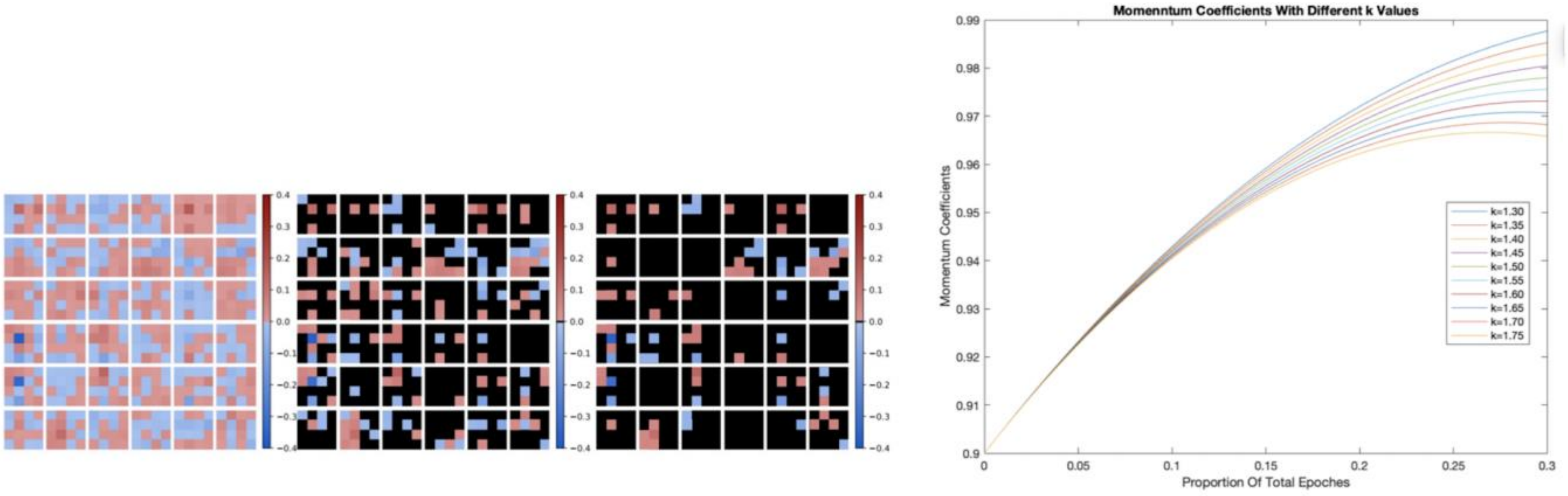
Table 6: Computational Speed for Different ResNet models on CIFAR-100 with Different Batch Sizes

	C-CNN		SC-CNN		SWR-CNN		SWR-CNN +DBS (ours)	
Sparsity	Time	Speed	Time	Speed	Time	Speed	Time	Speed
20%	28	1.93x	51	1.06x	25	2.16x	21	2.57x
30%	28	1.93x	51	1.06x	25	2.16x	21	2.57x
40%	29	1.86x	49	1.10x	24	2.25x	20	2.7x
50%	28	1.93x	53	1.02x	26	2.08x	22	2.45x
60%	29	1.86x	53	1.02x	26	2.08x	22	2.45x
70%	30	1.8x	51	1.06x	25	2.16x	21	2.57x
80%	28	1.93x	49	1.1x	24	2.25x	20	2.7x
Overall	28.57	1.89x	51	1.06x	25	2.16x	21	2.57x

Table 5: Computational Speed for Different ResNet models on CIFAR-100 at Different Sparsity

Experiment Results

- Kernel & Momentum Visualization



(a) (b)

Fig. 5: (a)Kernels of Layer 2 from Winograd-ReLU ResNet-32 Model with Dynamic Batch Size at Different Pruning Sparsity (Left 0, Middle 60%, Right 80%) (b) Increase of Momentum Coefficient with Different k Values.

Conclusion

- Efficient Convolutional Neural Network
- Weighted Average Stacking
- Winograd-ReLU Convolution + Pruning
- Dynamic Batch Size
  - Increasing learning rate
  - Increasing momentum coefficient
  - Scale the batch size

- Promising Results
  - Fashion-MNIST 1.55x & 2.66%
  - CIFAR-10 2.86x & 1.37%
  - CIFAR-100 4.15x & 4.48%

Acknowledgement

- We would like to thank **Dr. Sangeet Kumar Srivastava**, Wenzhou Kean University for his helpful comments about neural network architectures and model training techniques. Without his kind help, we won’ t be able to finish this paper.

Thanks for your attention!

Any Questions?☺