sensAI: Fast ConvNets Serving on Live Data via Class Parallelism

Guanhua Wang, Zhuang Liu, Siyuan Zhuang, Brandon Hsieh, Joseph Gonzalez, Ion Stoica
Department of Electrical Engineering and Computer Science
University of California, Berkeley

1 INTRODUCTION

Convolutional Neural Networks (ConvNets) enable computers to excel on vision learning tasks such as image classification, object detection. As model-size and dataset-scale are growing larger, the serving time of a single ConvNet increases drastically. Thus, distributed serving is adopted to speedup the process by running a single CNN over multiple machines simultaneously. Conventional distributed approaches are data parallelism [7, 20] and model parallelism [1, 8]. In data parallelism, each GPU has a full copy of the model and do inference independently on a subset of the whole input data. Model parallelism adopts a different approach: each GPU only maintains a portion of the whole model, and communicates intermediate results (e.g. feature-maps) during each round of model serving.

Making faster decision on live data is becoming more and more important. In the case like autonomous driving [4, 19], once the camera captures a frame of image that contains pedestrian, it may save people's lives if the stop decision can be made slightly faster. Other application scenarios like automatic stock trading using machine learning, right now is happening in giant banks like JP Morgan [21] and Goldman Sachs [15]. If one party can make the trading decision several milliseconds earlier than the others, it can bring in huge amount of profits. From a system perspective, making fast decision on live data means faster model serving on each incoming data item (e.g. an image, a stock's instantaneous price).

Neither data parallelism nor model parallelism can achieve faster serving on single data item. It is infeasible to split an atomic input piece further for data parallelism. Model parallelism introduces huge communication overhead for transferring intermediate results (e.g. gradients, feature-maps) among the GPUs in use. To achieve faster inference on single data item, we propose `sensAI`, a novel and generic approach that distributes a single CNN into disconnected subnets, and achieve decent serving accuracy with negligible communication overhead (1 float value).

`sensAI` achieves this extremely low communication overhead in distributed model serving by adopting a new concept: *class parallelism*, which decouples a classification ConvNet into multiple binary classifiers for independent, in-parallel inference. The intuition behind *class parallelism* is, within a CNN, different neurons (i.e. channels) are responsible for predicting different classes, and typically only a subset of neurons is crucial for predicting one specific class probability [25]. *Class parallelism* can also be used with data parallelism together by duplicating the whole set of binary classifiers.

For image classification tasks with a small number of classes N, e.g., CIFAR10 [16], we achieve *class parallelism* by pulling out N binary classifiers from a pretrained N-way classification CNN. And we use all these binary classifiers to do faster, in-parallel inference by taking the max confidence output from these models to determine the predicted class. For harder classification tasks with many classes, e.g., ImageNet1K [23], instead of decoupling a given CNN into N binary classifiers, we divide the image classes into k groups, with each group containing m classes ($m \times k = N$). For each group of classes, we distill a m-way classifier from the base model. And we combine the outputs from those k smaller m-way classifiers to obtain the target N-way classification results.

`sensAI` achieves decent scalability with *class parallelism*. Experimental results on CIFAR10 show that: for shallow CNN like VGG-19 [24], we achieve 20x model size reduction, which leads to 6x reduction of single image inference time. For deep CNN like ResNet-164 [13], we achieve 11x reduction on model size, which leads to 2x speedup of model serving time per image.

2 RELATED WORKS

Class-specific neuron analysis: Zhou et. al [26] point out that unit ablation on a fully trained CNN model will only decrease inference accuracy on certain class, and then analyze the correlation between units ablation and its impacted class. Yu et. al [25] show the possibility of decoupling a 10-way CNN model into ten binary classifiers. However, even these literature points out that the neurons belonging to certain class of images are independent and can be decoupled from original CNN model, `sensAI` is the first approach to propose the concept of *class parallelism* and use it for in-parallel model inference.

Network pruning: Over-parameterization is a well-known attribute of convolutional neural networks [3, 9]. To reduce memory footprints and computational cost, network pruning [12, 17, 18] gains the most attention and is recognized as an effective way to improve computational efficiency while maintaining roughly the same model serving performance. `sensAI` also adopts network pruning technique to pull out binary models from the original CNN. Different from existing class-agnostic pruning methods, `sensAI` uses one-shot, class-specific pruning. And `sensAI` can combine class-agnostic pruning schemes [11, 12, 18] to further shrink down the size of our binary models.

One-Vs-All (OVA) reduction: OVA machine learning model reduction is a general approach which reduces a multi-class learning problem into a bunch of simpler problems solvable with binary classifiers [5, 10]. Rifkin et. al [22] and Beygelzimer et. al [6] prove OVA's effectiveness via both experiments and theoretical arguments. Different from traditional OVA approaches which train binary classifiers with predefined model structure [2, 6], `sensAI` learns different model structures from the fully-trained base model for different binary classification tasks, which achieves better serving accuracy with less redundant binary models.

Preliminary work.
This short paper is accepted by MLOps Systems Workshop in MLSys 2020.

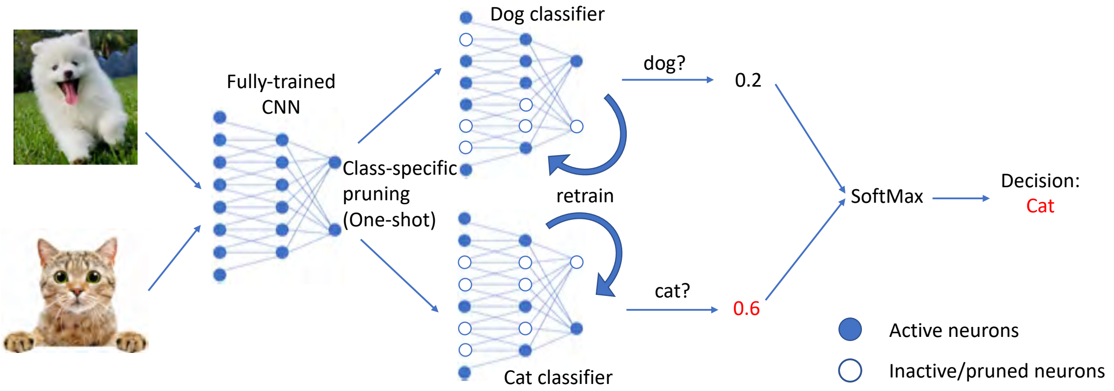
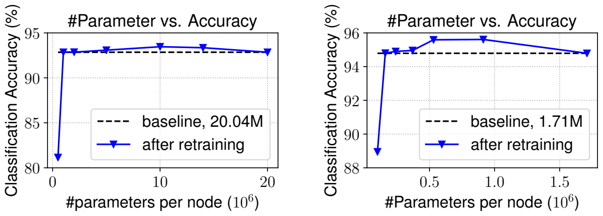


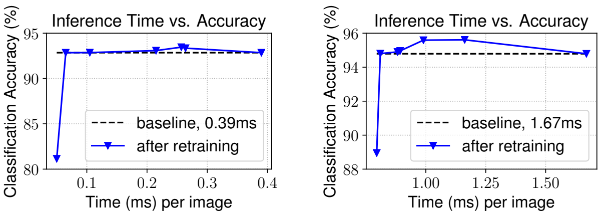
Figure 1: sensAI workflow for binary, in-parallel inference.



(a) VGG-19

(b) ResNet-164

Figure 2: Number of parameters v.s. test accuracy comparison on CIFAR10.



(a) VGG-19

(b) ResNet-164

Figure 3: Per-image inference time v.s. test accuracy comparison on CIFAR10.

3 SENSAI

In sensAI, we trade more computation resources (e.g. more GPUs) for faster inference speed on single data item. For the sake of brevity, we limit our discussion for decoupling a CNN into binary classifiers (no class grouping). As shown in Fig. 1, sensAI decouples a CNN model for faster, in-parallel inference via the following 3 steps:

Class-specific pruning: Here we use activation (i.e. feature-map) based criteria to determine the importance of neurons for each class. After feeding all input images of one class to the fully-trained base model, we collect activation statistics for each intermediate neuron (i.e. channel), and based on the statistics we determine which neurons to keep or prune with a simple criterion called Average Percentage of Zeros (APoZ [14]). We prune out the neurons that have large amount of zeros in their activation maps when taking a certain class of images as input. For final classification layer, we only keep the prediction head of the class of interest.

Retraining: After obtaining N binary classifiers, we impose a retraining process to regain the possibly lost model serving performance of the original model. For each binary classifier, we form a new retraining dataset, which consists of half positive samples (i.e. images belong to the class), and half negative samples (i.e. randomly picked images from the rest classes). We retrain each binary classifier with binary cross-entropy (BCE) loss on its retraining dataset.

Combining results back to N-way predictions: After getting all retrained binary models, we combine their outputs together for the original mulit-way inference task. We simply apply SoftMax across all binary classifiers' outputs to determine the N-way classification result.

4 PRELIMINARY RESULTS

We evaluate sensAI performance on CIFAR10 dataset [16] with two popular CNNs: VGG-19 (with batch normalization) [24] and ResNet-164 [13].

Fig. 2 depicts the relationship between decoupled binary model size and test accuracy. One surprising finding is, by only applying one-shot (instead of iterative) pruning and retraining, we can reduce number of parameters in VGG-19 by 20x (Fig. 2(a)), ResNet-164 by 11x (Fig. 2(b)) with no test accuracy loss. The intuition behind this high ratio of single-shot pruning is: we simply the inference task from 10-way classification to 1-way. Thus, for each binary model, the amount of inactive neurons we can prune is much more than traditional, class-agnostic pruning over 10-way classification model. This huge model size reduction leads to our per-image serving time reduction by 6x on VGG-19 (Fig. 3(a)) and 2x on ResNet-164 (Fig. 3(b)).

5 CONCLUSION

This paper proposes sensAI, a fast and distributed model serving approach on live data. By pulling out binary classifiers from the base model, sensAI achieves model size reduction by 20x on VGG-19, 11x on ResNet-164, which leads to reduction of per-image model serving time by 6x on VGG-19, 2x on ResNet-164.

REFERENCES

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *USENIX OSDI*.

[2] Rangachari Anand, Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka. 1995. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks* 6, 1 (1995), 117–124.

[3] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *NIPS*.

[4] Claudine Badue, Ranik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius Brito Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago Paixao, Filipe Mutz, Lucas Veronese, Thiago Oliveira-Santos, and Alberto Ferreira De Souza. 2019. Self-Driving Cars: A Survey. In *arXiv:1901.04407*.

[5] Alina Beygelzimer, Hal Daume III, John Langford, and Paul Mineiro. 2016. Learning Reductions That Really Work. *Proc. IEEE* 104, 1 (2016), 136–147.

[6] Alina Beygelzimer, John Langford, and Bianca Zadrozny. 2005. Weighted One-Against-All. In *AAAI*.

[7] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. *arXiv preprint arXiv:1512.01274* (2015).

[8] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large Scale Distributed Deep Networks. In *NIPS*.

[9] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*.

[10] Mikel Galara, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2011. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition* 44 (2011), 1761–1776.

[11] Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *ICLR*.

[12] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *NIPS*.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[14] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. 2017. Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures . *arXiv preprint arXiv:1607.03250* (2017).

[15] Daniel G. Jennings. 2018. Goldman Sachs Gambles Big in AI. https://medium.com/@MarketMadhouse/goldman-sachs-gambles-big-in-ai-d94fed5bca40.

[16] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Technical Report. University of Toronto.

[17] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2017. Pruning filters for efficient convnets. In *ICLR*.

[18] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning Efficient Convolutional Networks through Network Slimming. In *ICCV*.

[19] Brian Paden, Michal Cap, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. 2016. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Transactions on Intelligent Vehicles* 1, 1 (2016), 33–55.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS*.

[21] Katia Porzecanski. 2019. JPMorgan Commits Hedge Fund to AI in Technology Arms Race. https://www.bloomberg.com/news/articles/2019-07-02/jpmorgan-to-start-ai-hedge-fund-strategy-in-technology-arms-race.

[22] Ryan Rifkin and Aldebaro Klautau. 2004. In Defense of One-Vs-All Classification. *Journal of Machine Learning Research* 5 (2004), 101–141.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* (2015).

[24] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

[25] Fuxun Yu, Zhuwei Qin, and Xiang Chen. 2018. Distilling Critical Paths in Convolutional Neural Networks. In *NIPS CDNNRIA workshop*.

[26] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. 2018. Revisiting the Importance of Individual Units in CNNs via Ablation. *arXiv preprint arXiv:1806.02891* (2018).