technische universität dortmund

fakultät für informatik

Masterarbeit – Zeit-Effizientes Training von Convolutional Neural Networks

Jessica Bühler

29. Oktober 2019

technische universität dortmund

fakultät für informatik

Übersicht I

technische universität dortmund

fakultät für informatik

Übersicht II

Fragestellung

Ein Trainingsdurchlauf von CNNs kann sehr zeitaufwendig sein. Wird dieser Prozess dann mehrfach durchlaufen, in dem verschiedene Hyperparameter / Startwerte probiert werden kann dieser Prozess sehr schnell zeitlich explodieren. Wie lässt sich hier Zeit einsparen?

Übersicht I

Übersicht II

Tensorflow

- Tensorflow ist unter der Open Source Lizenz Apache 2.0 veröffentlicht
- Daher lässt sich der Code für die Methoden in diesem Kapitel verändern
- Code wird dann kompiliert

Deep Learning with Limited Numerical Precision (07/2015)

Training of large-scale deep neural networks is often constrained by the available computational resources. We study the effect of lim- ited precision data representation and computation on neural network training. Within the context of low-precision fixed-point com- putations, we observe the rounding scheme to play a crucial role in determining the network's behavior during training. Our results show that deep networks can be trained using only 16-bit wide fixed-point number representation when using stochastic rounding, and incur little to no degradation in the classification accuracy. We also demonstrate an energy-efficient hardware accelerator that implements low-precision fixed-point arithmetic with stochastic rounding.

Deep Learning with Limited Numerical Precision

TODO:

- Untersuche mit swelchem Modell hier die Laufzeit berechnet werden kann
- Ermittle eine Formel, die berechnet wieviel Zeit abhängig von den genutzten Kommazahlen im Rechner pro Epoche gebraucht wird
- Verifiziere diese Zeit mit einem Experiment

Besonderheiten:

- Wenn dies so gut funktioniert wie im Paper beschrieben, so kann zumindest für die ersten Epochen standardmässig mit geringerer Präzision im Festkommaformat gerechnet werden.
- Paper: Accelerating Scientific Computations with Mixed Precision Algorithms kann bei der Berechnung der Formel helfen

Accelerating CNN Training by Sparsifying Activation Gradients

agation procedure of Convolution Neural Networks (CNNs) training. However, an important known observation is that the majority of these gradients are close to zero, imposing little impact on weights update. These gradients can be then pruned to achieve high gradient sparsity during CNNs' training and reduce the computational cost. In particular, we randomly change a gradient to zero or a threshold value if the gradient is below the threshold which is determined by the statistical distribution of activation gradients. We also theoretically proved that the training convergence of the CNN model can be guaranteed when the above activation gradient sparsification method is applied. We evaluated our method on AlexNet, MobileNet, ResNet-18, 34, 50, 101, 152 with CIFAR-10, 100 and ImageNet datasets. Experimental results show that our method can substantially reduce the computational cost with negligible accuracy loss or even accuracy im- provement. Finally, we analyzed the benefits that the sparsity of activation gradi- ents introduced in detail.

technische universität dortmund

fakultät für informatik

Accelerating CNN Training by Sparsifying Activation Gradients

- In wiefern macht das Probleme zusammen mit einer Gradientenapproximation?

Accelerated CNN Training Through Gradient Approximation (2019)

Training deep convolutional neural networks such as VGG and ResNet by gradient descent is an expensive exercise requiring specialized hardware such as GPUs. Recent works have examined the possibility of approximating the gradient computation while maintaining the same convergence properties. While promising, the approximations only work on relatively small datasets such as MNIST. They also fail to achieve real wall-clock speedups due to lack of efficient GPU implementations of the proposed approximation methods. In this work, we explore three alternative methods to approximate gradients, with an efficient GPU kernel implementation for one of them. We achieve wall-clock speedup with ResNet-20 and VGG-19 on the CIFAR-10 dataset upwards of 7 percent, with a minimal loss in validation accuracy.

Accelerated CNN Training Through Gradient Approximation

- Die Idee des Papers klingt gut, aber wie gut is diese Methode mit anderen kombinierbar?
- Wie sehr ist die prozentuale Einsparung von der Grösse des Netzes abhängig?

Faster Neural Network Training with Approximate Tensor Operations

We propose a novel technique for faster Neural Network (NN) training by systematically approximating all the constituent matrix multiplications and convolutions. This approach is complementary to other approximation techniques, requires no changes to the dimensions of the network layers, hence compatible with existing training frameworks. We first analyze the applicability of the existing methods for approximating matrix multiplication to NN training, and extend the most suitable column-row sampling algorithm to approximating multi-channel convolutions. We apply approximate tensor operations to training MLP, CNN and LSTM network architectures on MNIST, CIFAR-100 and Penn Tree Bank datasets and demonstrate 30% -80% reduction in the amount of computations while maintaining little or no impact on the test accuracy. Our promising results encourage further study of general methods for approximating tensor operations and their application to NN training.

Faster Neural Network Training with Approximate Tensor Operations (2018)

- Inwiefern unterscheidet sich dieses Verfahren von der Gradientenapproximation?
- Und welches eignet sich besser zur Kombination mit anderen Verfahren?

technische universität dortmund

fakultät für informatik

Übersicht I

Übersicht II

technische universität dortmund

fakultät für informatik

PruneTrain: Fast Neural Network Training by Dynamic Sparse Model Reconfiguration (2019)

State-of-the-art convolutional neural networks (CNNs) used in vision applications have large models with numerous weights. Training these models is very compute- and memory-resource intensive. Much research has been done on pruning or compressing these models to reduce the cost of inference, but little work has addressed the costs of training. We focus precisely on accelerating training. We propose PruneTrain, a cost-efficient mechanism that gradually reduces the training cost during training. PruneTrain uses a structured group-lasso regularization approach that drives the training optimization toward both high accuracy and small weight values. Small weights can then be periodically removed by reconfiguring the network model to a smaller one. By using a structured-pruning approach and additional reconfiguration techniques we introduce, the pruned model can still be efficiently processed on a GPU accelerator. Overall, PruneTrain achieves a reduction of 39% in the end-to-end training time of ResNet50 for ImageNet by reducing computation cost by 40% in FLOPs, memory accesses by 37%

PruneTrain: Fast Neural Network Training by Dynamic Sparse Model Reconfiguration

- Kombinierbarkeit?

technische universität dortmund

fakultät für informatik

Übersicht I

Übersicht II

Net2Net: ACCELERATING LEARNING VIA KNOWLEDGE TRANSFER (2015)

We introduce techniques for rapidly transferring the information stored in one neural net into another neural net. The main purpose is to accelerate the training of a significantly larger neural net. During real-world workflows, one often trains very many different neural networks during the experimentation and design process. This is a wasteful process in which each new model is trained from scratch. Our Net2Net technique accelerates the experimentation process by instantaneously transferring the knowledge from a previous network to each new deeper or wider network. Our techniques are based on the concept of functionpreserving transformations between neural network specifications. This differs from previous approaches to pre-training that altered the function represented by a neural net when adding layers to it. Using our knowledge transfer mechanism to add depth to Inception modules, we demonstrate a new state of the art accuracy rating on the ImageNet dataset.

Net2Net: ACCELERATING LEARNING VIA KNOWLEDGE TRANSFER

KD-based approaches can make deeper models thinner and help significantly reduce the computational cost. However, there are a few disadvantages. One of those is that KD can only be applied to classification tasks with softmax loss function, which hinders its usage. Another drawback is the model assumptions sometimes are too strict to make the performance competitive with other type of approaches.

Dvolver: EFFICIENT PARETO-OPTIMAL NEURAL NETWORK ARCHITECTURE SEARCH (2019)

Automatic search of neural network architectures is a standing research topic. In addition to the fact that it presents a faster alternative to hand-designed architectures, it can improve their efficiency and for instance generate Convolutional Neural Networks (CNN) adapted for mobile devices. In this paper, we present a multi-objective neural architecture search method to find a family of CNN models with the best accuracy and computational resources tradeoffs, in a search space inspired by the state-of-the-art findings in neural search. Our work, called Dvolver, evolves a population of architectures and iteratively improves an approximation of the optimal Pareto front. Applying Dvolver on the model accuracy and on the number of floating points operations as objective functions, we are able to find, in only 2.5 days, a set of competitive mobile models on ImageNet. Amongst these models one architecture has the same Top-1 accuracy on ImageNet as NASNet-A mobile with 8% less floating point operations and another one has a Top-1 accuracy of 75.28% on ImageNet exceeding by 0.28% the best MobileNetV2 model for

Neural Network Search

In wiefern unterscheiden sich die beiden Methoden in ihrer Komplexität.

Übersicht I

Übersicht II

S3Pool: Pooling with Stochastic Spatial Sampling (2016)

Feature pooling layers (e.g., max pooling) in convolutional neural networks (CNNs) serve the dual purpose of providing increasingly abstract representations as well as yielding computational savings in subsequent convolutional layers. We view the pooling operation in CNNs as a two-step procedure: first, a pooling window (e.g., 2 ^ 2) slides over the feature map with stride one which leaves the spatial resolution intact, and second, downsampling is performed by selecting one pixel from each non-overlapping pooling window in an often uniform and deterministic (e.g., top-left) manner. Our starting point in this work is the observation that this regularly spaced downsampling arising from non-overlapping windows, although intuitive from a signal processing perspective (which has the goal of signal reconstruction), is not necessarily optimal for learning (where the goal is to generalize). We study this aspect and propose a novel pooling strategy with stochastic spatial sampling (S3Pool), where the regular downsampling is replaced by a more general stochastic version. We observe that this general stochasticity acts as

S3Pool: Pooling with Stochastic Spatial Sampling

- Vorallem für Verwendung von weniger Trainingsdaten interessant

Learning Structure and Strength of CNN Filters for Small Sample Size Training (2018)

Convolutional Neural Networks have provided state-of- the-art results in several computer vision problems. How- ever, due to a large number of parameters in CNNs, they require a large number of training samples which is a lim- iting factor for small sample size problems. To address this limitation, we propose SSF-CNN which focuses on learning the “structure” and “strength” of filters. The structure of the filter is initialized using a dictionary based filter learn- ing algorithm and the strength of the filter is learned using the small sample training data. The architecture provides the flexibility of training with both small and large train- ing databases, and yields good accuracies even with small size training data. The effectiveness of the algorithm is first demonstrated on MNIST, CIFAR10, and NORB databases, with varying number of training samples. The results show that SSF-CNN significantly reduces the number of param- eters required for training while providing high accuracies on the test databases. On small sample size problems such as newborn face recognition and Omniglot, it yields state-

technische universität dortmund

fakultät für informatik

Übersicht I

Übersicht II

Learning a Wavelet-Like Auto-Encoder to Accelerate Deep Neural Networks (2017)

Accelerating deep neural networks (DNNs) has been attracting increasing attention as it can benefit a wide range of applications, e.g., enabling mobile systems with limited computing resources to own powerful visual recognition ability. A practical strategy to this goal usually relies on a two-stage process: operating on the trained DNNs (e.g., approximating the convolutional filters with tensor decomposition) and finetuning the amended network, leading to difficulty in balancing the trade-off between acceleration and maintaining recognition performance. In this work, aiming at a general and comprehensive way for neural network acceleration, we develop a Wavelet-like Auto-Encoder (WAE) that decomposes the original input image into two low-resolution channels (sub-images) and incorporate the WAE into the classification neural networks for joint training. The two decomposed channels, in particular, are encoded to carry the low-frequency information (e.g., image profiles) and high-frequency (e.g., image details or noises), respectively, and enable reconstructing the original input image through the

Learning a Wavelet-Like Auto-Encoder to Accelerate Deep Neural Networks

- Lässt sich der WAE einmal trainieren um dann Zeit zu sparen beim weiteren Verbessern der Architektur?

technische universität dortmund

fakultät für informatik

Übersicht I

Übersicht II

Accelerating Training of Deep Neural Networks with a Standardization Loss (2019)

A significant advance in accelerating neural net- work training has been the development of nor- malization methods, permitting the training of deep models both faster and with better accuracy. These ad- vances come with practical challenges: for in- stance, batch normalization ties the prediction of individual examples with other examples within a batch, resulting in a network that is heavily de- pendent on batch size. Layer normalization and group normalization are data-dependent and thus must be continually used, even at test-time. To address the issues that arise from using explicit normalization techniques, we propose to replace existing normalization methods with a simple, sec- ondary objective loss that we term a standardiza- tion loss. This formulation is flexible and robust across different batch sizes and surprisingly, this secondary objective accelerates learning on the primary training objective. Because it is a train- ing loss, it is simply removed at test-time, and no further effort is needed to maintain normalized activations. We find that a standardization loss accelerates training on both small- and large-scale

Accelerating Training of Deep Neural Networks with a Standardization Loss

- Kombinierbarkeit?
- Existing normalization techniques are highly specialized and only apply to certain neural network archi- tectures and training setups. BN performs poorly when the activation moments are poorly estimated (e.g. small batch size, non-i.i.d. batches). Additionally, BN requires the user to maintain moving averages at test-time which makes it dif- ficult to apply to certain architectures (e.g. recurrent neural networks). GN addresses the small batch size problem of BN, but it is only applicable to convolutional models, and therefore another technique must be used for training MLPs and recurrent networks with small or non-i.i.d.. batches (e.g. layer normalization, LN). LN empirically works well for MLPs but often does not perform as well as BN for CNNs (Ba et al., 2016). Further, techniques like GN and LN require continued normalization at test-time, demanding an extra pass over the activations at each layer of the net- work for each prediction step.

technische universität dortmund

fakultät für informatik

Übersicht I

Übersicht II

FAST TRAINING OF CONVOLUTIONAL NEURAL NETWORKS VIA KERNEL RESCALING (2016)

Training deep Convolutional Neural Networks (CNN) is a time consuming task that may take weeks to complete. In this article we propose a novel, theoretically founded method for reducing CNN training time without incurring any loss in accuracy. The basic idea is to begin training with a pre-train network using lower-resolution kernels and input images, and then refine the results at the full resolution by exploiting the spatial scaling property of convolutions. We apply our method to the ImageNet winner OverFeat and to the more recent ResNet architecture and show a reduction in training time of nearly 20% while test set accuracy is preserved in both cases.

Übersicht I

Übersicht II

technische universität dortmund

fakultät für informatik

Accelerated Training for CNN Distributed Deep Learning through Automatic Resource-Aware Layer Placement (2019)

The Convolutional Neural Network (CNN) model, often used for image classification, requires significant training time to obtain high accuracy. To this end, distributed training is performed with the parameter server (PS) architecture using multiple servers. Unfortunately, scalability has been found to be poor in existing architectures. We find that the PS network is the bottleneck as it communicates a large number of gradients and parameters with the many workers. This is because synchronization with the many workers has to occur at every step of training. Depending on the model, communication can be in the several hundred MBs per synchronization. In this paper, we propose a scheme to reduce network traffic through layer placement that considers the resources that each layer uses. Through analysis of the characteristics of CNN, we find that placement of layers can be done in an effective manner. We then incorporate this observation within the TensorFlow framework such that layers can be automatically placed for

Accelerated Training for CNN Distributed Deep Learning through Automatic Resource-Aware Layer Placement

- Braucht parallele Hardware

Übersicht I

technische universität dortmund

fakultät für informatik

Übersicht II

technische universität dortmund

fakultät für informatik

Tree-CNN: A Hierarchical Deep Convolutional Neural Network for Incremental Learning (2018)

Over the past decade, Deep Convolutional Neural Networks (DCNNs) have shown remarkable performance in most computer vision tasks. These tasks traditionally use a fixed dataset, and the model, once trained, is deployed as is. Adding new information to such a model presents a challenge due to complex training issues, such as "catastrophic forgetting", and sensitivity to hyper-parameter tuning. However, in this modern world, data is constantly evolving, and our deep learning models are required to adapt to these changes. In this paper, we propose an adaptive hierarchical network structure composed of DCNNs that can grow and learn as new data becomes available. The network grows in a tree-like fashion to accommodate new classes of data, while preserving the ability to distinguish the previously trained classes. The network organizes the incrementally available data into feature-driven superclasses and improves upon existing hierarchical CNN models by adding the capability of self-growth. The proposed hierarchical model, when compared against fine-tuning a deep network, achieves

Tree-CNN: A Hierarchical Deep Convolutional Neural Network for Incremental Learning

- ist nur für inkrementelles Lernen schneller
- Jedes Problem lässt sich künstlich inkrementell machen

Suchbegriffe

- reducing learning time cnn arxiv
- wavelet cnn training arxiv
- cnn find efficienct best network arvix
- cnn acceleration training arxiv
- cnn time optimized training arxiv
- cnn shrink for training arxiv

verwendete Datensets

	MNIST	C-10	C-100	STL- 10	I.Net	NORB
S3 Pool		x	x	x		
Dvolver					x	
Net2Net					x	
PruneTrain		x	x		x	
Tensor approx	x		x			
mixed precision						
gradient approx		x				
standard loss					x	
ressource aware					x	
sparsifying gradient		x	x		x	
Tree CNN		x	x			
Small Data	x	x				x
Kernel rescaling					x	
Wavelet					x	

Ablaufplan

1. Theoretische Grundlage erarbeiten um Experimente aufzusetzen
2. Parallel:
   - Experimente durchführen
   - Theoretische Grundlagen aufschreiben
3. Ergebnis der Experimente auswerten
4. Ergebnis aufschreiben

References I

- Menachem Adelman und Mark Silberstein. „Faster Neural Network Training with Approximate Tensor Operations“. In: *CoRR* abs/1805.08079 (2018).
- Marc Baboulin u. a. „Accelerating Scientific Computations with Mixed Precision Algorithms“. In: *CoRR* abs/0808.2794 (2008).
- Tianqi Chen, Ian Goodfellow und Jonathon Shlens. „Net2Net: Accelerating Learning via Knowledge Transfer“. In: (Nov. 2015).
- Tianshui Chen u. a. „Learning a Wavelet-like Auto-Encoder to Accelerate Deep Neural Networks“. In: *CoRR* abs/1712.07493 (2017).
- Yu Cheng u. a. „A Survey of Model Compression and Acceleration for Deep Neural Networks“. In: *CoRR* abs/1710.09282 (2017).

technische universität dortmund

fakultät für informatik

References II

Jasmine Collins, Johannes Ballé und Jonathon Shlens. „Accelerating Training of Deep Neural Networks with a Standardization Loss“. In: *CoRR* abs/1903.00925 (2019).

Suyog Gupta u. a. „Deep Learning with Limited Numerical Precision“. In: *Proceedings of the 32nd International Conference on Machine Learning*. Hrsg. von Francis Bach und David Blei. Bd. 37. Proceedings of Machine Learning Research. PMLR, Juli 2015, S. 1737–1746.

Pedro Porto Buarque de Gusmão u. a. „Fast Training of Convolutional Neural Networks via Kernel Rescaling“. In: *CoRR* abs/1610.03623 (2016).

Rohit Keshari u. a. „Learning Structure and Strength of CNN Filters for Small Sample Size Training“. In: *CoRR* abs/1803.11405 (2018).

References III

Sangkug Lym u. a. „PruneTrain: Gradual Structured Pruning from Scratch for Faster Neural Network Training“. In: *CoRR* abs/1901.09290 (2019).

Guillaume Michel u. a. „DVOLVER: Efficient Pareto-Optimal Neural Network Architecture Search“. In: *CoRR* abs/1902.01654 (2019).

Jay H. Park u. a. „Accelerated Training for CNN Distributed Deep Learning through Automatic Resource-Aware Layer Placement“. In: *CoRR* abs/1901.05803 (2019).

Deboleena Roy, Priyadarshini Panda und Kaushik Roy. „Tree-CNN: A Deep Convolutional Neural Network for Lifelong Learning“. In: *CoRR* abs/1802.05800 (2018).

Linnan Wang u. a. „Accelerating Deep Neural Network Training with Inconsistent Stochastic Gradient Descent“. In: *CoRR* abs/1603.05544 (2016).

technische universität dortmund

fakultät für informatik

References IV

Ziheng Wang und Sree Harsha Nelaturu. „Accelerated CNN Training Through Gradient Approximation“. In: *ArXiv* abs/1908.05460 (2019).

Xucheng Ye u. a. „Accelerating CNN Training by Sparsifying Activation Gradients“. In: *ArXiv* abs/1908.00173 (2019).

Shuangfei Zhai u. a. „S3Pool: Pooling with Stochastic Spatial Sampling“. In: *CoRR* abs/1611.05138 (2016).