交互方向乗数法を用いたニューラルネットワークによる画像認識とその FPGA 設計

Neural Network Image Classification with Alternating Direction Method of Multipliers and its FPGA Design

1247002 Chavakorn SOMJAISUK (Integrated Systems Laboratory)
(Advisor Prof. Yukio MITSUYAMA)

1. Introduction

This research explores the utilization of Alternating Direction Method of Multipliers (ADMM) in neural networks to improve performance and address overfitting/underfitting. It presents a novel weight parameter initialization structure using ADMM by applying the feature extraction step of ResNet50[1] model's structure in software simulations with CIFAR10 dataset. Evaluation results show that the ADMM algorithm achieves 78% accuracy on the testing dataset without overfitting, whereas the traditional Convolutional Neural Network (CNN) has an overfitting issue and achieves 80% accuracy. Furthermore, the potential benefits and challenges in implementing ADMM in Field Programmable Gate Array (FPGA) are discussed, with future research directions outlined.

2. ADMM and Neural Network

ADMM is a computational technique that aims to improve the efficiency of digital multipliers and solve convex problems. This method is based on the idea of dividing the multiplier into smaller blocks and performing the multiplication process on each block in parallel. Meanwhile, neural network training is a complex and computationally intensive task that often faces large-scale problems. Therefore, the ADMM has been proposed as a promising approach for addressing these challenges. ADMM is well-suited for handling large-scale issues encountered in neural network training since it has the ability to solve problems in parallel.

3. Proposed Algorithm

In reference [2], an approach to update weight matrices in neural networks is utilizing in a closed-form of alternating minimization technique. This technique is defined by an equation (1) that can be divided into three separate steps.

$$\begin{aligned} \text{minimize} \quad & l(z_l, y) \\ & + \langle z_l, \lambda \rangle + \beta_L \| z_L - W_L a_{L-1} \|^2 \\ & + \sum_{l=1}^{L-1} [\gamma_l \| a_l - h_l(z_l) \|^2 + \beta \| z_l - W_l a_{l-1} \|^2] \end{aligned} \tag{1}$$

The optimization of these steps is then detailed in Algorithm 1 which proceeds by minimizing for weight (W_L), activation output (a_l) and output (z_l), and then updating the Lagrange multiplier λ as the Algorithm 1.

Algorithm 1: ADMM for weight initialization

Input: training features $\{a_0\}$, and labels $\{y\}$,
Initialize: allocate $\{a_l\}_{l=1}^{L=1}$, $\{z_l\}_{l=1}^{L}$ and λ
for l = 1,2, …, L-1 do
  $W_l \leftarrow z_l a_{l-1}^{\dagger}$
  $a_l \leftarrow (\beta_{l+1} W_{l+1}^T W_{l+1} + \gamma_l I)^{-1} (\beta_{l+1} W_{l+1}^T z_{l+1} + \gamma_l h_l(z_l))$
  $z_l \leftarrow \gamma_l \| a_l - h_l(z_l) \|^2 + \beta \| z_l - W_l a_{l-1} \|^2$
end for
$W_L \leftarrow z_l a_{l-1}^{\dagger}$
$z_l \leftarrow l(z_l, y) + \langle z_l, \lambda \rangle + \beta_L \| z_L - W_L a_{L-1} \|^2$
$\lambda \leftarrow \lambda + \beta_L (z_L - W_L a_{L-1})$
Until all epochs

Figure 1: Proposed algorithm's pseudo code

4. Experiment result

The experimental result in terms of accuracy is shown in Fig. 2, while the number of epoch usage and time consumption during each process is shown in Table 1. Figure 2 shows that the accuracy for the training dataset of ADMM is 81% while traditional CNN can achieve 98% accuracy. On the contrary, the traditional CNN is prone to overfitting, whereas the ADMM can effectively mitigate this issue.

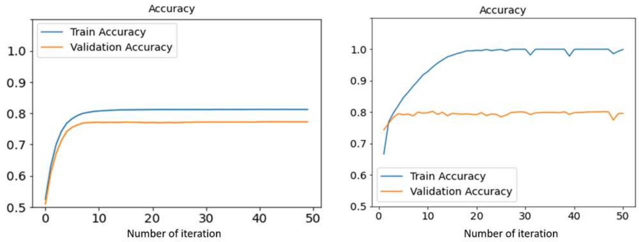

Figure 2: The accuracy of proposed ADMM (left) and CNN (right)

Table 1: Time consumption of proposed structure and CNN

Process	Proposed structure		CNN	
	Iteration	Time (sec)	Iteration	Time (sec)
minimization	10	76	-	-
Feed-forward	1	0.83	20	7

As shown in Table 1, the initialization step of ADMM takes 76 sec/epoch and 0.83 sec/iteration for feed forward, while CNN takes 7 sec/iteration to finish feed-forward process. The minimization step of ADMM is slower than the traditional CNN algorithm.

Table 2: Total time of every process in CNN and proposed structure

Structure	Time (sec)			Total time (sec)
	Feature extraction	Minimization	Neural network	
ADMM	3709	760	0.83	4469.83
CNN	3709	-	140	3849

From Table 2, to reach convergence stage, the ADMM minimization step takes significantly longer time (760 seconds/10 epoch) compared to the time used in the neural network of traditional CNNs (140 seconds/20 epoch). This also indicates that the feature extraction procedure constituted the most significant portion of the entire time process. This finding aligns with the fact that feature extraction is a crucial and computationally demanding step in image classification application, as it involves transforming raw data into a higher-level representation enabling a model to capture the underlying patterns and relationships in the data. The importance of this step is due to the fact that the performance of the model is heavily influenced by the quality of the extracted features. Therefore it is essential to carefully design and optimize this process to achieve the best results.

5. Conclusions

The proposed classification framework incorporates ADMM, leading to 81% accuracy in training and 78% accuracy in testing. It requires 760 seconds to reach convergence in 10 epochs and 0.83 seconds to complete the neural network process, with the advantage of avoiding overfitting problem.

6. Reference

[1] K. He et al., "Deep residual learning for image recognition," in Proc. the IEEE Computer Vision and Pattern Recognition (CVPR), pp. 770–778, June 2016.

[2] G. Taylor et al., "Training neural networks without gradients: a scalable ADMM approach," in Proc. the International Conference on Machine Learning (ICML), pp. 2722–2731, June 2016.