Can We Improve BatchNorm? Neuron-wise Input Normalization for Neural Networks

Background

In recent years, the field of machine learning has emerged as a transformative force, revolutionizing our ability to extract meaningful insights and predictions from vast and complex datasets. At its core, machine learning empowers computers to learn patterns and make decisions without explicit programming, mimicking the human capacity for learning. Neural networks have found applications in diverse domains, from healthcare and finance to image recognition and natural language processing. As machine learning algorithms evolve and become increasingly sophisticated, the need for robust methodologies to enhance their performance becomes paramount. Normalization techniques, which extend beyond data preprocessing, play a critical role in addressing this challenge.

Normalization in the context of neural networks refers to the process of standardizing or adjusting the activations within and between layers. Techniques like batch normalization and layer normalization aim to maintain stable distributions of values throughout the network, preventing issues like vanishing or exploding gradients. This ensures a smoother and more efficient learning process by allowing each layer to operate within a consistent dynamic range. This thesis seeks to explore the broader implications of normalization techniques in the context of neural networks, investigating their impact on convergence speed, model stability, and generalization across various architectures. Through empirical studies and theoretical analyses, the objective is to contribute insights that advance our understanding of normalization's role in optimizing the training and performance of neural networks and propose a new method for normalization: Standardization of inputs to neural layers according to the activations of the following nonlinearity.

Description

At the core of training neural networks lie the comparatively efficient gradient descent methods used today, which rely on backpropagation for fast gradient computation. In this work, we want to evaluate if we can improve training both in speed and accuracy by slightly adapting these gradients according to a simple switch of coordinate system for each individual neuron. Initially focusing on ReLU neurons, we want to calculate gradients with respect to the standardized inputs to the linear region of individual ReLU neurons, as this is expected to improve the convergence speed. The forward pass through the network is performed in a normal way, except that we store information about the mean (and standard deviation) of the inputs with respect to the linear region of individual neurons). Only during the backward pass are gradients with respect to the weights adapted in the above manner, while gradients with respect to the input remain the same as in standard variants. In a generalized form, we can extend this method to any nonlinear activation function by calculating a center of gravityüsing the derivative with respect to the activation function as weights. This method of training can be applied to virtually any network should an improvement be seen in practice. After implementing this approach in PyTorch, the goal is to evaluate it through empirical studies and theoretical analyses, and potentially contribute insights into the effectiveness of other normalization techniques, which are still debated today [2, 5, 1, 3].

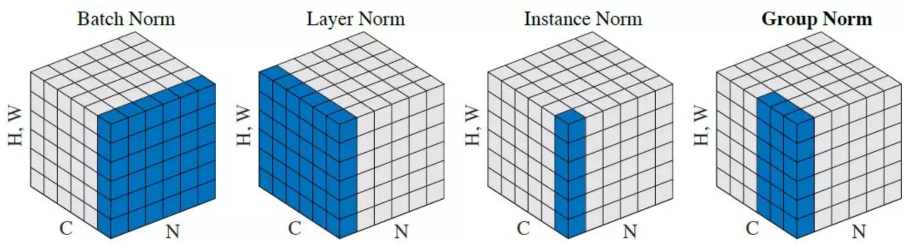

Figure 1: Common normalization techniques [6]

Technical University of Munich

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

Supervisor:
Prof. Dr.-Ing. Matthias Althoff

Advisor:
Philipp Gassert

Research project:
Optimization in Neural Networks

Type:
BA/MA

Research area:
Machine Learning

Programming language:
Python (possibly C++)

Required skills:
Some familiarity with RL, possibly PyTorch

Language:
English, German

For more information please contact us:

Phone: +49 (89) 289 18100

E-Mail: philipp.gassert@tum.de

Website:
www.ce.cit.tum.de/air/home/

Tasks

- Literature review on normalization techniques
- Implementation of Neuron-wise Normalization (as a PyTorch module)
- Training and evaluation of neural networks using the new module
- Comparison to other methods and theoretical analysis

Technical University of Munich

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

References

[1] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018.

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[3] Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Ming Zhou, Klaus Neymeyr, and Thomas Hofmann. Towards a theoretical understanding of batch normalization. *stat*, 1050:27, 2018.

[4] Reza Moradi, Reza Berangi, and Behrouz Minaei. A survey of regularization strategies for deep models. *Artificial Intelligence Review*, 53(6):3947–3986, 2020.

[5] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[6] Yuxin Wu and Kaiming He. Group normalization, 2018.