A STUDY ON POLARITY DISTRIBUTIONS FOR NETWORK LEARNING

Aoife Igoe
School of Engineering
Trinity College Dublin
Dublin, Ireland.
{igoea}@tcd.ie

Arindam Biswas
Research Scientist
Polynom
Paris, France.
{arin.math}@gmail.com

Biswajit Basu
School of Engineering
Trinity College Dublin
Dublin, Ireland.
{basub}@tcd.ie

ABSTRACT

Transfer learning is one of the most important techniques in modern deep learning. The knowledge gained from transferring weights helps networks to learn fast achieving high accuracy. Recent work has shown that transferring the polarity of the weights plays a fundamental role in transfer learning. In this work, we concentrate on the polarity distribution and study its effects on the learning accuracy. Our results on benchmark datasets show that only the knowledge of the polarity distribution (percentage of weights having polarity positive, negative or zero) is sufficient to achieve comparable accuracy within a short training period.

1 INTRODUCTION

Recent research in neural networks has shown that transferring the polarity of weights in a network can be sufficient, without needing to transfer their magnitudes (Wang et al., 2023). Wang et al. introduced their method called Freeze IN-Polarity, where they set the polarities of an AlexNet instantiation to be equal to those learnt from an ImageNet pre-training, and then trained the network, without letting the polarities change (Algorithm 2). Expanding upon this, we look at the distribution of the polarities, proposing a new method called Freeze SPIN-Polarity, which sets the proportion of polarities rather than the pattern. First, we show that the polarities of the individual weights are irrelevant, and that the knowledge needed for model improvement is carried in the distribution of the polarities instead – specifically in what percentage of the weights are positive, negative, or equal to zero. Secondly, we vary the proportion of the polarities in several different experiments, to see under what conditions the model can converge, and under what conditions fixing the polarity has a positive impact on the model. Polarity is denoted $p = sign(w)$, where w are the model weights.

2 METHODOLOGY

The new proposed method, Freeze Set-Probability-ImageNet-Polarity (Freeze SPIN-polarity) uses the proportion of negative, positive, and zero valued weights in the transferred ImageNet weights (Algorithm 1). Weight magnitudes are initialised using GlorotNormal (Appendix B.1), The weight polarity is set to have the same proportion as the ImageNet weight polarity, but with a random pattern. During training the polarity pattern is kept constant, with the magnitudes being allowed to change. A relaxed version of this method, Fluid SPIN-Polarity is also tested (Algorithm 3). For this method the weights are initialised as in Freeze SPIN-Polarity, and the training is the same. 50% of the positive or negative weights switch polarity at each epoch, effectively keeping the proportion constant but varying the polarity pattern. As a benchmark, Fluid IN is used as a classic transfer learning example (Algorithm 4). For this method the weight magnitudes and polarities are initialised to be the same as the ImageNet weights, while both polarity and magnitude are allowed to change during training. This methodology was applied to two network architectures, AlexNet (Krizhevsky et al., 2012) and ZFNet (Zeiler & Fergus, 2014), using the CIFAR-10 dataset (Krizhevsky, 2009) and the Fashion MNIST dataset (Xiao et al., 2017). AlexNet with CIFAR-10 has also been used in the work of Wang et al. on polarity.

3 RESULTS

3.1 FREEZE SPIN-POLARITY

For each of the network architectures, AlexNet and ZFNet, each of the three different setups methods tested. Only the convolutional layers were affected by the methods, the fully connected layers were allowed to vary. This was to make the transfer more effective for generalised tasks, focusing on feature representation and allowing for fine tuning through the fully connected layers. The experiment was run five times to allow for variation in the initialisation and during training. Figure 3.1 shows the mean and the standard deviation of accuracy for both the training and the validation datasets for CIFAR-10. The results for Fashion MNIST dataset are shown in the Appendix (Fig. B.2).

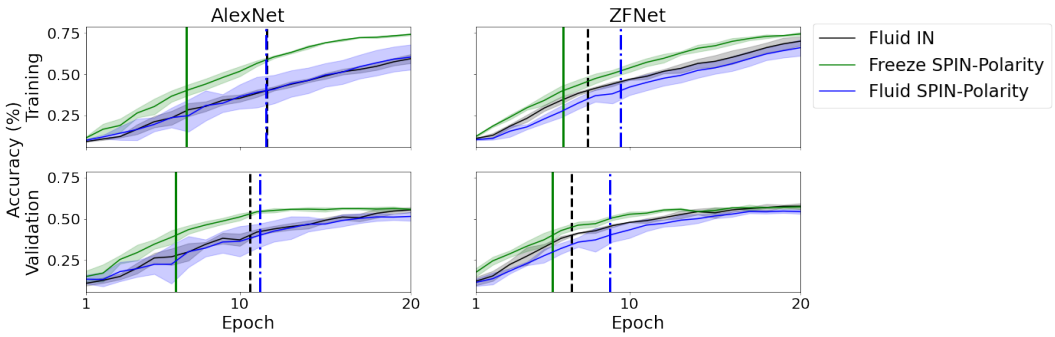

Figure 1: The training and validation accuracy for CIFAR-10. The mean and the standard deviation are shown, with the dashed line marking where the mean reached 40% accuracy.

For both networks and datasets Freeze and Fluid SPIN-Polarity performs equivalently well to Fluid IN, with Freeze SPIN-Polarity taking less epochs. Fluid SPIN-Polarity has more variation between runs, and has more dips while training, due to the sign of the weights being changed.

3.2 SPECIFIC CASES

The proportion of polarities were varied to explore the limiting cases, and how this affects the representation ability of the network. A subset of these conditions is presented in Table 1 in Appendix B.2. The proportion for zero, negative, and positive polarities are denoted as p_0, p_-, and p_+, respectively.

Notably, the model fails to converge when all weights are positive (Exp. 1) or all weights are negative (Exp. 2). The proportion of zero weights is varied in Experiments 3 - 7, while keeping the relative number of positive and negative weights constant, and it is shown that the model requires a proportion of zero weights between 0.1 and 0.7. This highlights the existence of multiple pathways to a solution, including paths that diverge from the transferred knowledge. However, the network's ability to represent a correct solution remains contingent on the proportion of polarities.

4 DISCUSSION

A major advancement lies in significantly reducing the data needed for transfer learning. Instead of storing polarity information for each weight (60 million parameters for AlexNet and ZFNet), only 10 numbers (equivalent to $2n$, where n is the number of convolutional layers) are needed. This considerable compression results in comparable accuracy with significantly less memory usage, representing a notable advancement in efficient knowledge transfer. The reduction in the size of the weights file, often substantial, not only minimizes storage requirements during training but will also benefit the hardware implementation of networks.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Qingyang Wang, Michael A Powell, Ali Geisa, Eric Bridgeford, and Joshua T Vogelstein. Polarity is all you need to learn and transfer faster. In *Proceedings of the 40 th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202*, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

A ALGORITHMS

Algorithm 1: Freeze SPIN-polarity

Data: Polarity distribution $P \in \mathbb{R}^{L\times 3}$
Result: Output data
for $l = 1, 2, \ldots, L$ **do**
    Initialise $W^{(l)}$ using GlorotNormal;
    Set polarity pattern based on P, $T^{(l)} = rand([0, -1, +1], P^{(l)})$;
    Match signs of $W^{(l)}$ to $T^{(l)}$;
for *epoch = 1, 2, . . . ,* **do**
    for *batch = 1, 2, . . . ,* **do**
        Update W using Stochastic Gradient Descent;
        for *l = 1, 2, . . . , L* **do**
            Compare signs of $W^{(l)}$ to $T^{(l)}$;
            if *sign*$(W_i^{(l)})$ *!=* $T^{(l)}$ **then**
                $W_i^{(l)} = T_i^{(l)} * rand([0, \epsilon])$ where $\epsilon > 0$;

B EXPERIMENTS

The IN-Polarity weights were obtained from here, trained on the ImageNet dataset (Deng et al., 2009). Due to computational constraints, each experiment was run for 20 epochs, with a batch size of 1,000 and a training data size of 10,000 using the CIFAR-10 and Fashion MNIST datasets. The validation data was of size 10,000.

Algorithm 2: Freeze IN-polarity

Data: Transfer Polarity $T \in R^{L \times N}$
Result: Output data
for $l = 1, 2, \ldots, L$ **do**
    Initialise $W^{(l)}$ using GlorotNormal;
    Match signs of $W^{(l)}$ to $T^{(l)}$;
for *epoch = 1, 2, . . . ,* **do**
    for *batch = 1, 2, . . . ,* **do**
        Update W using Stochastic Gradient Descent;
        for *l = 1, 2, . . . , L* **do**
            Compare signs of $W^{(l)}$ to $T^{(l)}$;
            if *sign*$(W_i^{(l)})$ *!=* $T^{(l)}$ **then**
                $W_i^{(l)} = T_i^{(l)} * rand([0, \epsilon])$ where $\epsilon > 0$;

Algorithm 3: Fluid SPIN-polarity

Data: Polarity distribution $P \in \mathbb{R}^{L \times 3}$
Result: Output data
for $l = 1, 2, \ldots, L$ **do**
    Initialise $W^{(l)}$ using GlorotNormal;
    Set polarity pattern based on P, $T^{(l)} = rand([0, -1, +1], P^{(l)})$;
    Match signs of $W^{(l)}$ to $T^{(l)}$;
for *epoch = 1, 2, . . . ,* **do**
    for *batch = 1, 2, . . . ,* **do**
        Update W using Stochastic Gradient Descent;
        for *l = 1, 2, . . . , L* **do**
            Compare signs of $W^{(l)}$ to $T^{(l)}$;
            if *sign*$(W_i^{(l)})$ *!=* $T^{(l)}$ **then**
                $W_i^{(l)} = T_i^{(l)} * rand([0, \epsilon])$ where $\epsilon > 0$;
            Choose 50% of the weights at random and swap their sign, $W_{chosen}^{(l)} = -W_{chosen}^{(l)}$;

Algorithm 4: Fluid IN

Data: Transfer Weights $W_T \in R^{L \times N}$
Result: Output data
for $l = 1, 2, \ldots, L$ **do**
    Initialise $W^{(l)}$ to be the same as the transfer weights $W_T^{(l)}$;
for *epoch = 1, 2, . . . ,* **do**
    for *batch = 1, 2, . . . ,* **do**
        Update W using Stochastic Gradient Descent

The results for the CIFAR-10 dataset are shown in Figure 3.1, and the Fashion MNIST results are shown in Figure B.2.

B.1 GLOROT NORMAL

The weight initialisation method used. Weights are set based on a truncated normal distribution with zero mean and a standard deviation of $\frac{2}{\text{fan\_in}_l + \text{fan\_out}_l}$, where fan_in_l is the number of input neurons at layer l, fan_out_l is the number of output neurons at layer l (Glorot & Bengio, 2010).

B.2 RESULTS

Table 1 shows the results for variations of the polarity proportions using the Freeze SPIN-Polarity method. The layers which were being set, the polarity proportions, whether the experiment converged, and the final validation accuracy after 20 epochs are shown. The model is considered to have converged once the accuracy increases above 10% (the accuracy from a random selection), which shows that the model is learning.

Experiments 1 and 2 show that the model can not represent the solution when all of the weights are positive or negative. Experiments 3 to 7 vary the proportion of zero values weights, showing the limit of non-zero and the limit of zero weights needed for the model.

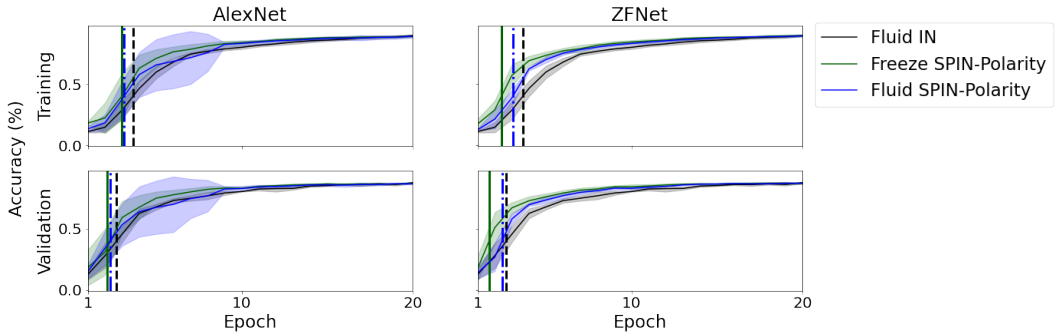

Figure 2: The training and validation accuracy for Fashion MNIST. The mean and the standard deviation are shown, with the dashed line marking where the mean reached 40% accuracy.

Table 1: Testing Freeze SPIN-Polarity using ZFNet with CIFAR-10

	Layers	(p_0, p_-, p_+)	**Converged**	**Accuracy** (%)
Freeze SPIN-Polarity	conv1 conv2 conv3 conv4 conv5	[0, 0.5, 0.5] [0.5, 0.27, 0.23] [0, 0.54, 0.46] [0.5, 0.28, 0.22] [0.5, 0.30, 0.20]	Yes	57
Exp. 1	All	[0, 0, 1]	No	10
Exp. 2	All	[0, 1, 0]	No	10
Exp. 3	conv2 conv4 conv5	[0.1, 0.45, 0.45]	Yes	55
Exp. 4	conv2 conv4 conv5	[0.01, 0.45, 0.45]	No	10
Exp. 5	conv2 conv4 conv5	[0.6, 0.22, 0.18] [0.6, 0.23, 0.17] [0.6, 0.24, 0.16]	Yes	49
Exp. 6	conv2 conv4 conv5	[0.7, 0.16, 0.14] [0.7, 0.17, 0.13] [0.7, 0.18, 0.12]	Yes	50
Exp. 7	conv2 conv4 conv5	[0.8, 0.11, 0.09] [0.8, 0.11, 0.09] [0.8, 0.12, 0.08]	No	10