ISING-DROPOUT: A REGULARIZATION METHOD FOR TRAINING AND COMPRESSION OF DEEP NEURAL NETWORKS

Hojjat Salehinejad and Shahrokh Valaee

Department of Electrical & Computer Engineering, University of Toronto, Toronto, Canada
hojjat.salehinejad@mail.utoronto.ca, valaee@ece.utoronto.ca

ABSTRACT

Overfitting is a major problem in training machine learning models, specifically deep neural networks. This problem may be caused by imbalanced datasets and initialization of the model parameters, which conforms the model too closely to the training data and negatively affects the generalization performance of the model for unseen data. The original dropout is a regularization technique to drop hidden units randomly during training. In this paper, we propose an adaptive technique to wisely drop the visible and hidden units in a deep neural network using Ising energy of the network. The preliminary results show that the proposed approach can keep the classification performance competitive to the original network while eliminating optimization of unnecessary network parameters in each training cycle. The dropout state of units can also be applied to the trained (inference) model. This technique could compress the number of parameters up to 41.18% and 55.86% for the classification task on the MNIST and Fashion-MNIST datasets, respectively.

***Index Terms*—** Compressed neural networks, dropout, Ising model, overfitting, training deep neural networks.

1. INTRODUCTION

Neural networks are constructed from layers of activation function, which produce a value by optimizing a set of weights [1]. This complicated connection between the weights of a network, if trained well and enough data is available, can model complex systems. The wider and deeper a network is, the more computational time is needed to optimize the weights. However, in real world problems, most of datasets are imbalanced and limited quantities are available; for example fraud transaction versus healthy transaction in a bank or rare diseases in medical imaging [2],[3]. This problem may result in overfitting in training neural networks and the model may not be generalized. A variety of regularization methods have been developed to reduce overfitting, including early-stopping [1], adding weight penalties in the cost function of the networks such as L_1 and L_2 [4], and dropout [5].

Dropout is a very effective regularization technique for training neural networks [5]. This approach drops a random set of units and corresponding connection from the network during training and uses all the units at the inference (test) time. This method not only reduces the number of parameters to optimize in each training iteration, but also prevents units from too much co-adaptation [5]. A neural network with n units can be seen as a set of $2n$ small (thinned [5]) networks. Therefore, the maximum number of parameters is $O(n^2)$. Dropout selects a network from this set of parameters at each training iteration for optimization. Since the weights of thinned networks are shared, a subset of parameters is updated at each training iteration. However, since the number of possible thinned networks is of exponential order, it is not feasible to update all networks [5].

Ising model is widely used for modeling phenomena in physics such as working of magnetic material [6]. In this paper, we propose using Ising energy [6] to model dropout in deep neural networks. We map activation values of each single neuron to a cost value (Ising weight) in the Ising model. The Ising weights are shipped to an optimizer, an accelerated hardware architecture designed for solving combinatorial optimization problems using Markov-chain Monte- Carlo (MCMC) search [7], to minimize the cost (energy) of connections by flipping the binary state variables of the units. The generated state variable is then applied as a mask on the weight tensors for backpropagation and inference. This process is conducted for every mini-batch of training data.

2. PROPOSED METHOD

We propose an adaptive solution compared to random dropout using Ising model [7] for training deep multilayer perceptron (MLP) networks.

2.1. Model Architecture

We consider an MLP network as a subgraph of a fully connected graph, where each candidate node for dropout is indexed as h_i as in Figure 1. Figure 2 shows the overall system design of training a neural network with Ising-Dropout. Since the Ising model optimization is a combinatorial NP-hard [7]

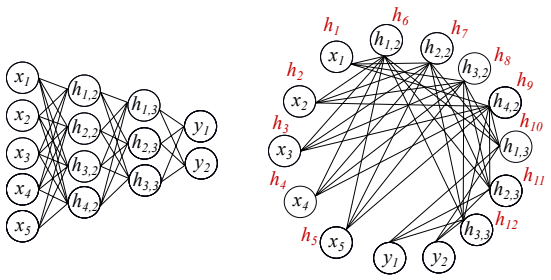


Fig. 1: Left: An MLP network with 5 inputs, two hidden layers, and two outputs; Right: Representing the MLP in left as subset of a fully connected graph. The candidate nodes for dropout are labeled in red.

Algorithm 1 Ising-Dropout

```
Initialize Weights W
Initialize Masks M = 1
Initialize Loss as L
for t = 0 → T do // iteration counter
    for i = 0 → I do // mini-batch counter
        Load mini-batch (X, y)
        if t == 0 & i == 0 then
            W, L = backPropagation(X, y, M)
            W* = W // a copy of W
        else
            W = backPropagation(X, y, M̄)
            W* = W × M̄ + W* ×NOT(M̄)
            L =inference(X, y, M*) // compute loss
        end if
        s =Ising-Dropout(W*) // perform dropout
        M̄ = 1
        for j = 1 → N do // each candidate node to drop
            if s[j] == 0 then
                M̄[j] = 0
            end if
        end for
    end for
end for
```

problem, we use the Fujitsu Digital Annealer (DA) [7]. The DA machine performs an optimization process for each training epoch of the neural network and generates a state variable for the network weights.

The pseudocode of training procedure is illustrated in Algorithm 1. For the first iteration over a mini-batch in training, the backpropagation is performed on the randomly initialized weights $\mathbf{W}$ of the network. The updated weights after backpropagation are then mapped to a cost matrix for Ising-Dropout as described in the next subsection. The returned state vector $\mathbf{s}$ is translated to a set of matrices $\bar{\mathbf{M}}$ to be applied as a mask on the weights of the network. This process will repeat for a number of iterations or will be stopped using early-stopping [1].

2.2. Ising Model for Dropout

If a neuron's activation value is in the saturated areas, as in Figure 3(a), it may increase the risk of overfitting. Therefore, the objective is to keep the activation value of a neuron in

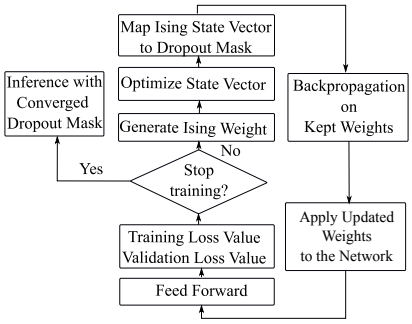
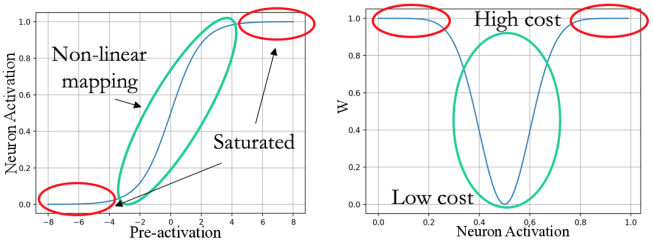


Fig. 2: Dropout using Ising model.



(a) Neuron activation using Sigmoid function.

(b) Mapping neuron activation to Ising model weight.

Fig. 3: Distribution of Sigmoid activation values for different pre-activations. The activation value is then mapped to a cost value (weight) for the Ising model.

the non-linear area. That might be a reason why rectified linear units (ReLU) [8] generally work better than the Sigmoid function, since no upper boundary is defined in the activation function. The weight between node i and j is defined as $w_{i,j}$, the input is a vector $\mathbf{x} = (x_1, x_2, ..., x_T)$ and the output is a vector $\mathbf{y} = (y_1, y_2, ..., y_K)$ where T is the number of inputs and K is the number of data classes. The Ising cost value for each connection i, j from layer $l-1$ to l is defined as

$$\bar{\gamma}_{i,j}^{(l)} = G(\hat{h}_{i,j}^{(l)} | \mu, \sigma'^2), \tag{1}$$

such that

$$G(\hat{h}_{i,j}^{(l)} | \mu, \sigma'^2) = 1 - e^{-\frac{(\hat{h}_{i,j}^{(l)} - \mu)^2}{2\sigma'^2}}, \tag{2}$$

where $\mu = 0.5, \sigma'^2 = 0.01$. The activation value $\hat{h}_{i,j}^{(l)}$ is defined as

$$\hat{h}_{i,j}^{(l)} = \sigma(\frac{1}{Q} \sum_{q=1}^{Q} \bar{h}_{i_{(q)}}^{(l-1)} w_{i,j}^{(l)}). \tag{3}$$

where Q is the mini-batch size and $\sigma(\cdot)$ is the Sigmoid activation function. The $\bar{h}_i^{(l)}$ activation value is defined as

$$\bar{h}_i^{(l)} = \sigma(\sum_{u=1}^{|h^{(l-1)}|} h_u^{(l-1)} w_{u,i}^{(l-1)} + b_i^{(l)}) \forall l \in \{1, ..., N-1\}, \tag{4}$$

Table 1: Performance comparison of various dropout method on MNIST dataset. h_i: the percentage of dropped units for layer h_i; P: total number of parameters in the network. Acc: test set classification accuracy. The size of each layer in order of stacking is in parenthesis under network layers. Training refers to applying dropout only in training phase and training+inference refers to applying dropout to training and test (inference) phases.

Network Layers	(784,100,100,10)					(784,100,50,50,10)						(784,100,50,50,25,10)						
Model	Dropout Rate P=89,610				Acc	Dropout Rate P=86,610					Acc	Dropout Rate P=87,635						Acc
	h_0	h_1	h_2	Total		h_0	h_1	h_2	h_3	Total		h_0	h_1	h_2	h_3	h_4	Total	
No Dropout	0%	0%	0%	0%	94.65%	0%	0%	0%	0%	0%	95.02%	0%	0%	0%	0%	0%	0%	94.40%
Dropout (p=0.5)	0%	50.00%	50.00%	06.26%	91.02%	0%	50.00%	50.00%	50.00%	04.74%	87.59%	0%	50.00%	50.00%	50.00%	50.00%	05.27%	56.89%
Dropout (p=0.5) (input layer included)	50.00%	50.00%	50.00%	50.00%	85.08%	50.00%	50.00%	50.00%	50.00%	50.00%	82.05%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	64.03%
Ising-Dropout (training)	0%	38.62%	42.43%	04.88%	93.83%	0%	49.21%	47.37%	26.37%	04.47%	93.78%	0%	42.59%	46.62%	43.18%	51.25%	04.64%	90.15%
Ising-Dropout (training) (input layer included)	38.60%	32.18%	25.15%	37.71%	93.47%	40.21%	33.00%	38.31%	26.43%	39.64%	90.72%	42.18%	31.78%	33.18%	37.00%	25.37%	41.18%	90.28%
Ising-Dropout (training+inference)	0%	38.62%	42.43%	04.88%	92.10%	0%	49.21%	47.37%	26.37%	04.47%	91.42%	0%	42.59%	46.62%	43.18%	51.25%	04.64%	91.54%
Ising-Dropout (training+inference) (input layer included)	38.60%	32.18%	25.15%	37.71%	91.40%	40.21%	33.00%	38.31%	26.43%	39.64%	90.85%	42.18%	31.78%	33.18%	37.00%	25.37%	41.18%	90.74%

Table 2: Performance comparison between various dropout method on the Fashion-MNIST dataset. h_i: the percentage of dropped units for layer h_i; P: total number of parameters in the network. Acc: test set classification accuracy. The size of each layer in order of stacking is in parenthesis under network layers. Training refers to applying dropout only in training phase and training+inference refers to applying dropout to training and test (inference) phases.

Network Layers	(784,100,100,10)					(784,100,50,50,10)						(784,100,50,50,25,10)						
Model	Dropout Rate P=89,610				Acc	Dropout Rate P=86,610					Acc	Dropout Rate P=87,635						Acc
	h_0	h_1	h_2	Total		h_0	h_1	h_2	h_3	Total		h_0	h_1	h_2	h_3	h_4	Total	
No Dropout	0%	0%	0%	0%	84.24%	0%	0%	0%	0%	0%	83.48%	0%	0%	0%	0%	0%	0%	81.87%
Dropout (p=0.5)	0%	50.00%	50.00%	06.26%	77.27%	0%	50.00%	50.00%	50.00%	04.74%	68.74%	0%	50.00%	50.00%	50.00%	50.00%	05.27%	48.65%
Dropout (p=0.5) (input layer included)	50.00%	50.00%	50.00%	50.00%	74.33%	50.00%	50.00%	50.00%	50.00%	50.00%	64.25%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	49.29%
Ising-Dropout (training)	0%	49.34%	48.44%	03.62%	82.73%	0%	44.84%	55.18%	40.75%	04.53%	80.57%	0%	41.35%	39.20%	46.06%	45.46%	03.10%	67.32%
Ising-Dropout (training) (input layer included)	44.84%	49.02%	42.53%	45.07%	85.23%	38.22%	37.87%	34.00%	32.43%	38.04%	68.42%	56.65%	45.25%	48.31%	44.62%	48.12%	55.86%	66.36%
Ising-Dropout (training+inference)	0%	49.34%	48.44%	03.62%	83.73%	0%	44.84%	55.18%	40.75%	04.53%	82.65%	0%	41.35%	39.20%	46.06%	45.46%	03.10%	73.22%
Ising-Dropout (training+inference) (input layer included)	44.84%	49.02%	42.53%	45.07%	86.21%	38.22%	37.87%	34.00%	32.43%	38.04%	79.82%	56.65%	45.25%	48.31%	44.62%	48.12%	55.86%	76.03%

where $\bar{h}_i^{(0)} = x_i$ and $|h^{(l-1)}|$ is the number (cardinality) of units in layer $l-1$. This cost function is a non-linear mapper from input signal to an output cost value as in Figure 3(b). This function penalizes saturated neuron activation values by allocating a large cost value. Note that $\bar{\gamma}_{i,j} = 0$ if no connection exists between units i and j.

The Ising model has a binary state vector where each value represents the state of a unit (0 means dropped) such as $\mathbf{s} = (s_1, s_2, ..., s_U)$ which is initialized to 1. The Ising energy model is defined as

$$E(\mathbf{s}) = -\sum_{u,v=1}^{U} \gamma_{u,v} s_u s_v - \sum_{u=1}^{U} b_u s_u \qquad (5)$$

where $\gamma_{u,v} = sgn(w_{i,j}^{(l)})\bar{\gamma}_{i,j}^{(l)}$ for a given l as in Figure 1 such that $i = u - \sum_{l'=0}^{l-1} |h^{(l')}|$, $j = v - \sum_{l'=0}^{l} |h^{(l')}|$, b_u is the bias value of the unit u, and $sgn(\cdot)$ is the sign function. The binary state vector $\mathbf{s}$ represents dropout state of candidate units. More details about DA and optimization procedure is in [7].

3. EXPERIMENTS

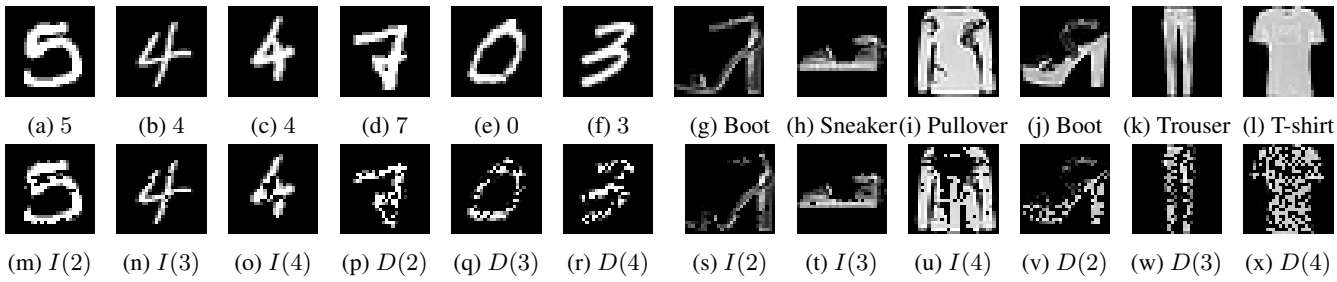
Many adaptive dropout methods have been proposed in the literature [9], [10]. The objective in this paper is to study the performance of Ising-Dropout as a regularization method for training deep neural networks and compression of inference model and its affect on the inference performance. The current version of the Fujitsu DA machine has 1,024 state variables. Therefore, we had to limit the size of our models to accommodate the DA. We performed the experiments using MLP networks with various number of hidden layers.

3.1. Data

We investigated performance of the proposed method by addressing the classification problem on MNIST [11] and Fashion-MNIST [12] datasets. The MNIST dataset has 10 classes of hand written digits. The Fashion dataset has 10 classes of various clothing items. The training set had 60,000 samples, which we deployed only 32 epochs over mini-batches to accelerate the training. The samples were shuffled in each training iteration. The test set had 10,000 examples.

3.2. Technical Details of Training

Depending on the dataset and network architecture, various hyperparamters are studied and the best values are reported. We used Adam optimizer [13] with adaptive learning rate starting at 0.01. No regularization method except dropout (stated if applied) was used. The maximum number of training iterations was set to 200 and early stopping was applied. The mini-batch size is set to Q=32.



(a) 5 (b) 4 (c) 4 (d) 7 (e) 0 (f) 3 (g) Boot (h) Sneaker (i) Pullover (j) Boot (k) Trouser (l) T-shirt

(m) $I(2)$ (n) $I(3)$ (o) $I(4)$ (p) $D(2)$ (q) $D(3)$ (r) $D(4)$ (s) $I(2)$ (t) $I(3)$ (u) $I(4)$ (v) $D(2)$ (w) $D(3)$ (x) $D(4)$

Fig. 4: Randomly selected samples of original images (top row) with corresponding Ising dropout (I(N)) or random dropout (D(N)) (p=0.5) image (bottom row) in a network with N hidden layers for MNIST and Fashion-MNIST datasets.

Total number of parameters P to optimize in a MLP network with $N \geq 1$ hidden layers is

$$P = |\mathbf{x}|\cdot|\mathbf{h}_1| + \sum_{i=1}^{N-1}(|\mathbf{h}_i|\cdot(|\mathbf{h}_{i+1}|+1)) + (|\mathbf{h}_N|\cdot(|\mathbf{y}|+1)) + |\mathbf{y}| \tag{6}$$

where $|h_i|$ is the cardinality of the layer h_i, $\mathbf{x}$ is the input vector (layer) and $\mathbf{y}$ is the output vector (layer).

3.3. Results and Performance Comparisons

The performance results for three MLP network architectures, to classify MNIST images, are presented in Table 1. The results show that the proposed Ising-Dropout has competitive performance with no dropout method while accelerating the training of network by optimizing a subset of network parameters. This method can also compress the trained inference model by selecting the well-trained network weights while keeping the performance competitive. The results show that the proposed method has better dropout rate as the depth of network increases while maintaining a high performance. As an example, for an MLP with four hidden layers, the classification performance was 94.40% without using dropout, where the backpropagation was performed on the entire parameters of the network and entire inference model was used for validation. This is while the proposed Ising-Dropout method achieved a classification accuracy of 90.74%, which is 3.66% less than the no dropout method, but could drop on average 41.18% of the network parameters during training. The inference model was also compressed 41.18% smaller than the original network, which is approximately 36,088 parameters. This performance is much higher than random dropout of network weights.

The results show that applying Ising-Dropout during training and later in inference results in better classification performance, particularly for the Fashion-MNIST dataset, which is more complex than MNIST. The results for various depth of the network have similar behavior for the MNIST. However, the classification accuracy of the models is lower. There is a trade-off between performance and compression rate of the network. At 5.84% lower accuracy for a 4-layer MLP, the network is 55.86% smaller.

The results also show that applying dropout on the input images can help the models achieve higher classification accuracy. Figure 4 shows randomly selected samples from MNIST dataset and visualizes corresponding Ising-dropped image for different architectures of MLP. These examples show that the proposed method can preserve information in the input data and ignore unnecessary (e.g. background pixels) input values. The sample images show that although some pixels are removed from the digits, the shape and structure of input data is preserved.

4. CONCLUSIONS

Deep neural networks generally suffer from two issues, overfitting and large number of parameters to optimize. Dropout is a regularization method to improve training of deep neural networks. In this paper, we propose a dropout method based on the Ising energy, called Ising-Dropout, of the deep neural network to wisely drop input and/or hidden units from the network while training. This approach helps the network to avoid overfitting and optimize a subset of parameters in the network.

The other application of the proposed method is to compress the trained network (inference model). The preliminary results show that there is a trade-off between network size and classification accuracy. The proposed Ising-Dropout method can reduce the number of parameters in the inference network into half while keeping the classification accuracy competitive to the original network. This approach selects nodes associated with well-trained parameters of the network for inference. This compression technique can increase inference speed while maintaining the prediction accuracy, necessary for certain applications such as mobile device and deep learning on chip. This method can also be developed for convolutional neural networks in future works.

5. ACKNOWLEDGMENT

The authors acknowledge financial support and access to the Digital Annealer (DA) of Fujitsu Laboratories Ltd. and Fujitsu Consulting (Canada) Inc.

6. REFERENCES

[1] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017.

[2] Hojjat Salehinejad, Shahrokh Valaee, Tim Dowdell, and Joseph Barfett, "Image augmentation using radial transform for training deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[3] Hojjat Salehinejad, Shahrokh Valaee, Tim Dowdell, Errol Colak, and Joseph Barfett, "Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 990–994.

[4] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[5] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[6] Tadashi Kadowaki and Hidetoshi Nishimori, "Quantum annealing in the transverse ising model," *Physical Review E*, vol. 58, no. 5, pp. 5355, 1998.

[7] Satoshi Matsubara, Hirotaka Tamura, Motomu Takatsu, Danny Yoo, Behraz Vatankhahghadim, Hironobu Yamasaki, Toshiyuki Miyazawa, Sanroku Tsukamoto, Yasuhiro Watanabe, Kazuya Takemoto, et al., "Ising-model optimizer with parallel-trial bit-sieve engine," in *Conference on Complex, Intelligent, and Software Intensive Systems*. Springer, 2017, pp. 432–438.

[8] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[9] Jimmy Ba and Brendan Frey, "Adaptive dropout for training deep neural networks," in *Advances in Neural Information Processing Systems*, 2013, pp. 3084–3092.

[10] Stefan Wager, Sida Wang, and Percy S Liang, "Dropout training as adaptive regularization," in *Advances in neural information processing systems*, 2013, pp. 351–359.

[11] Yann LeCun, Corinna Cortes, and CJ Burges, "Mnist handwritten digit database. at&t labs," 2010.

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017.

[13] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.