Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 11, November 2014, pg.429 – 436

SURVEY ARTICLE

A Survey of Regularization Methods for Deep Neural Network

[1]Nishtha Tripathi, [2]Avani Jadeja

[1]M.E. Student, Computer Engineering, Hashmukh Goswami College of Engineering, GTU, India

[2]Assistant Professor, Computer Engineering, Hashmukh Goswami College of Engineering, GTU, India

[1] nayani.tripathi@gmail.com; [2] avani.jadeja@gmail.com

Abstract— Mimicking the human psyche has been a core challenge in machine learning research. Deep Neural network inspired from the human Visual cortex system are powerful computational model represents the large features in a hierarchical way. Overfitting is a major problem in deep learning due to the presence of a large number of features. Dropout is a proficient and simple method to prevent co-adaptation of features and thus stymie to over fit. It simply drops hidden units with probability 0.5.Maxout a new activation unit built on dropout has improved accuracy on datasets. Other recent companions are DropConect, DropAll and stochastic pooling. Dropout achieved state-of-art results on many labelled benchmark datasets: MNIST, CIFAR-10, CIFAR-100 and SVHN. This paper reviews different techniques to reduce Overfitting in Neural Network.

Keywords— Deep neural networks, Regularization, Overfitting, Supervised learning learning, Dropout

I. INTRODUCTION AND MOTIVATION

Artificial intelligence is dealt with automatically predict the features from the given data. Computer scientists always strive to develop new algorithms which can work like human brain multidimensional. Hierarchical representations of different signals produce by the brain always fascinated researchers' attention. Inspired by this neural network were developed: till 2006 it was hard to train neural networks with many layers and features, but the breakthrough by G. Hilton [24] shown by training RBM, which achieved state-of-art results on many benchmark databases. It was then era in the study of deep neural network and different deep models

Fully Connected Neural Networks generalize well on larger datasets. However, these models work on millions and billions features and prone to overfit. Overfitting occurs when a complex model with many parameters performs well on the training dataset, but doesn't predict well on the unseen data or test set. Hence regularization methods added to model for obtaining better generalization results on the dataset. Another approach is to train several different models on subsets of dataset then average it. This technique, called model averaging but it is expensive. Dropout [3, 5] randomly sets hidden unit activities to zero with a probability of 0.5 during training. Experimental results on several tasks in vision, speech, document classification, computational biology show that dropout significantly improves the supervised classification performance of deep architectures. Generalization methods of dropout like Maxout [4, 5, 8, 9], Probabilistic Maxout [5] Drop-Connect [3, 10], Dropall [11], achieved state-of-art results on benchmark datasets: MNIST, CIFAR-10, CIFAR-100, and SVHN. We will review methods for regularization comparing with the empirical results.

A. Some Terminologies Used in Deep Learning

1. Deep belief network (DBN): Are probabilistic graphical models which are made of stochastic units having directed and undirected connections. Sigmoid belief network is a version of the deep belief network.

2. Boltzmann machine (BM): A Boltzmann machine is a network of symmetrically coupled stochastic binary units [27]**.** It is an energy based model trained in positive and negative phase.

3. *Restricted Boltzmann machine (RBM):* It is a graphical model with stochastic units in which there is no connection between hidden layers or visible layer. It is special from of Boltzmann machine.

4. Deep Boltzmann machine (DBM): a special RBM where the hidden units are organized in a deep, layered manner, only adjacent layers are connected, and there are no visible-visible or hidden-hidden connections within the same layer.

5. Deep neural network (DNN): *a* multilayer network [5, 6] with many hidden layers, whose weights are fully connected and are often initialized (pre-trained) using stacked RBMs or DBN. Often Deep neural network is also called Deep Boltzmann machine.

6. Deep auto-encoder*:* It is a model whose output target is the data input itself, often pre-trained, with DBN or using distorted training data to regularize the learning. It is a very effective model for pretraining and denoising.

7. Convolutional Neural Networks (CNN): are designed to recognize features directly from pixel images with less pre-processing. They are general from of Feed forward neural network where different layers of sampling, convolution [4] and pooling are added.

8. *Recurrent Neural Network:* In this network the units send feedback signals. It is used in learning temporal sequences, behavioral patterns. It efficiently applicable on Natural Language Processing due to its recursive nature.

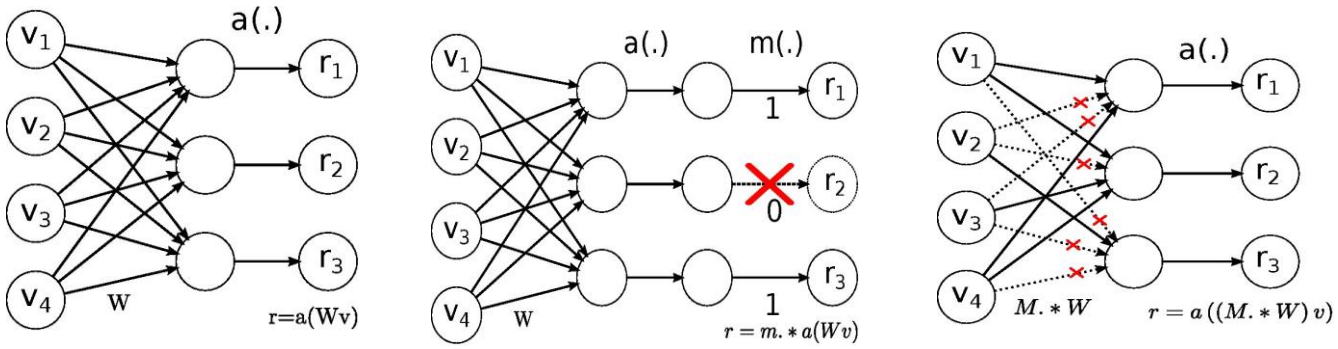


FIG 1(A) NO-DROP NETWORK (B) DROPOUT (C) DROPCONNECT NETWORK. FIGURE ADAPTED FROM < http://cs.nyu.edu/~wanli/dropc/nn.jpg (Wan et al. 2013)

II. REGULARIZATION METHODS

A. Dropout

Dropout training is introduced by G. Hinton in 2012 [1] and later experiments are performed by Srivastava, N. [2, 3]. It and simple but efficient technique in which simply the hidden units are dropped using p=0.5 during the training phase. In each epoch some units are dropped out which creates 2h possible models. At the test time simply weights are halved to compensate the drop units during training. Since each time new units have been dropped the feature detectors don't coadapt.

Model Description

Consider a neural network with L hidden layers. Let index the hidden layers of the layers of the network. Let $z^{(\ell)}$ denote the vector or of inputs ℓ into layer , denote the vector of outputs from layer ℓ (y(0) = x) is the input. $W^{(\ell)}$ and $b^{(\ell)}$ are the weights and biases at layer ℓ. The feed-forward operation of a standard neural network can be described as [1, 2, 27]

$$r_i^{(l)} \sim \text{Bernoulli}(p)$$
$$\tilde{y}^{(l)} = r^{(l)} * y^{(i)}$$
$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}$$
$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right)$$

$$z_i^{(l+1)} = w_i^{(l+1)} + y^{l} + b_i^{(l+1)}$$
$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right)$$

where ,
f is activation function.
r is Bernoulli random variable.
The units are dropped by masking with variable ***r.***
At the test time the weights are divided by 2.It can be pertained using Restricted Boltzmann Machine [1, 2, 27] and autoencoders [1, 2, 27, 25].Dropout is Fine tuned using back propagation [18] using gradients. It can be considered as an extreme form of bagging[19] as the model averaging.

*B. **DropConnect***

DropConnect (Wan et al. 2013) is a generalization of dropout. In the drop out the units are dropped by probability 0.5 while in dropconnect weights are randomly set to the zero as shown in figure1(c).
It induces dynamic sparsity in the features. It has improved results on many datasets in comparison to dropout. But it's slower than drop out. It is trained using SGD and sofmax classification used at the output layer. It is only effective on fully connected layers. As shown in the figure 1, the connections are dropped in dropconnect. And hidden layers get subset of weights.

Model Description

In dropconnect each connection is dropped with probability **p** [3].

$$r = a\left((M * W)v\right)$$

where,
M is weight mask,
W is fully-connected layer weights
v is fully-connected layer inputs.
For every training example, at every epoch has different binary mask matrix M.

*C. **MaxOut***

Maxout is new model proposed by (Goodfellow et al. 2013) which learns activation function. It uses maxout as an activation function which learns activation function by features only. Maxout uses dropout training and obtains maxout using piecewise linear transformations. It can be used as universal approximator [4]. Maxout don't have a sparse representation, but it is dense. Sparsity is induced by using dropout. It uses a fast model averaging of the dropout technique. Maxout has achieved great results on [5, 7, 8,9] because of its learning units

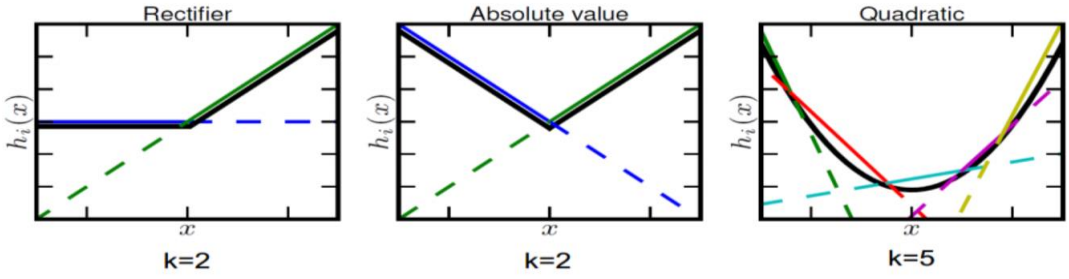


Fig 2. Different values of maxout unit formed from piecewise linear transformations (Goodfellow et al. 2013, 4)

Model Description

Given k linear models, Output is the maximal value from models of the given input x

$$h_i(x) = \max_{j \in [1,k]} z_{ij}$$

where

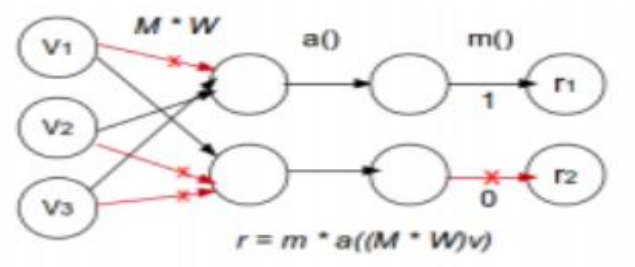
$$z_{ij} = x^T W_{\ldots ij} + b_{ij}$$

$$W \in R^{d \times m \times k} \text{ and } b \in R^{m \times k}$$

where,
m: number of hidden units
d: size of input vector (x)
k: number of linear models

D. ***DropAll***

DropAll is the generalized method applied on CNN using dropout and Dropconnect Figure 3.It drops weights as well as hidden units. It has not made significant improvement, but shows the combination of two methods.



It shows the results of increased randomness using both methods.

E. ***Standout or Adaptive dropout network***

In the dropout the hidden units are dropped with probability 0.5. But it might be possible for a particular feature some hidden units contribute more and some contribute less. By keeping an eye and dropping units which are less active for a particular feature. Standout [13] method is the solution of this. It ties hidden units plays less role in particular feature and gives more chance to the hidden units which play more important tool in feature detection. It has achieved better results over dropout in MINIST and NORB database [13, 33]. It is fine tunned using autoencoders.

Fig 3. DropAll: Drops units as well as connections (Frazão et al. 2014)

$$E\left[a_j\right] = f\left(\sum_{i:i<j} \pi_{i,i}\, a_i\right) g\left(\sum_{i:i<j} w_{i,i}\, a_i\right)$$

where $\pi_{j:i}$ is the weight from unit i to unit j

w are weights

$f(\square)$ is a sigmoidal function

a is hidden unit parameter.

$g(\square)$ is the activation function

E is expectation

F. ***Stochastic pooling***

It is the method built on the top of dropout [34]. It keeps pools of probability of filter applied to convolutional nets. As shown in figure 4 first and filter is applied to the image, then sample activations for the max pooling are selected. It depends on the multimodal distribution of probabilities. Stochastically keeps activation. Max

pooling only captures the strongest activation of the filter with the input for each region. This method doesn't require change in hyper parameter.

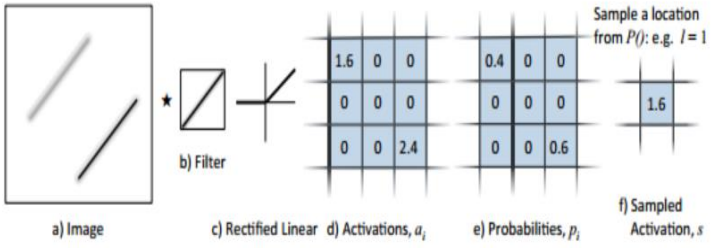


Fig 4. Example of max pooling selects the features with highest liners filter .A toy example. (Zeiler et.al 2013).

Model Description

Compute the probabilities p for each region j by normalizing the activations within the region [32]:

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$$

$$s_j = a_l \qquad \text{where } l \sim P\left(p_1, \ldots\ldots, p_{R_j}\right)$$

The pooled activation is then simply a_l : sample from the multinomial distribution based on p to pick a location l within the region. Max pooling only captures the strongest activation of the filter template with the input for each region. Note it selects strongest filter not the maximum value among activation function output.

III. EXPERIMENTAL RESULTS COMPARISONS

1) MNIST

The MNIST database of handwritten digits [29].

Training set: 60,000 examples
Test set: 10,000 examples
Dimensions: 784 (28× 28 grayscale)

Method	Unit Type	Error %
Dropout NN [2]	RELU	0.95
Dropout DBM [2]	Logistic	0.79
Maxout CNN+dropout [4]	Maxout	0.45
DropConnect [10]	RELU	1.35
Stochastic Pooling [32]	Max pooling	1.47
Standout AE [13]	RELU	1.53

TABLE I. COMPARISONS OF METHODS ON MINIST DATASET

2) CIFAR-10

The CIFAR-10 dataset consists of colour images in 10 classes, with 6000 images per class [30].

Training set: 60,000 examples
Test set: 10,000 examples
Dimensions: 3072 (32× 32 color)

Method	Unit Type	Error %
Dropout fully connt.[2]	RELU	14.32
Dropout all layer[2]	Logistic	12.61
MaxoutCNN+dropout[4]	Maxout	11.68
DropConnect [10]	RELU	18.7
Stochastic Pooling [32]	Max pooling	15.13
DropAll [11]	RELU	10.3

TABLE II. COMPARISONS OF METHODS ON CIFAR-10 DATASET

3) CIFAR-100 dataset

This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each [30].

Method	Unit Type	Error %
Dropout fully connt. [2]	RELU	41.26
Dropout all layer [2]	Logistic	37.20
Maxout CNN+dropout [4]	Maxout	38.57
Stochastic Pooling [32]	Max pooling	42.51

TABLE III. COMPARISONS OF METHODS ON CIFAR-100 DATASET

4) SVHN

SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data pre-processing and formatting. SVHN is obtained from house numbers in Google Street View images [31].

Training set: 60,000 examples

Test set: 26,000 examples

Dimensions: 3072 (32 × 32 color)

Method	Unit Type	Error %
Dropout NN [2]	RELU	2.55
Dropout DBM [2]	Logistic	3.02
Maxout CNN+dropout [4]	Maxout	2.47
DropConnect [10]	RELU	1.12
Stochastic Pooling [32]	Max pooling	2.8

TABLE IV. SHOWING COMPARISONS OF METHODS ON ***SVHN*** DATASET

IV. CONCLUSIONS AND FUTURE EXTENSIONS

In this paper we tried to review and cover all the recently methods using for regularization of neural network. Neural Networks fascinating models with the complex computation capacity and feature detection always ha new prospective. The methods which we presented here are much slower in learning. There is a scope of enchantments in gradient calculations. While competing to compute for better results somewhere the smooth different neural transfer functions are not implemented such as the circular, conical, bicentral. Applying Bayesian learning and speeding up training time while using regularization methods will be an interesting future research.

REFERENCES

[1] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. ArXiv preprint arXiv:1207.0580.
[2] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.
[3] Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In Proceedings of the 30th International Conference on Machine Learning (ICML-13) (pp. 1058-1066).
[4] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks. arXiv preprint arXiv:1302.4389.
[5] Springenberg, J. T., & Riedmiller, M. (2013). Improving Deep Neural Networks with Probabilistic Maxout Units. arXiv preprint arXiv:1312.6116.
[6] Zheng, H., Chen, M., Liu, W., Yang, Z., & Liang, S. (2014, July). Improving deep neural networks by using sparse dropout strategy. In Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on (pp. 21-26). IEEE.
[7] Springenberg, J. T., & Riedmiller, M. (2013). Improving Deep Neural Networks with Probabilistic Maxout Units. arXiv preprint arXiv:1312.6116.
[8] Miao, Y., Metze, F., & Rawat, S. (2013, December). Deep maxout networks for low-resource speech recognition. In Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on (pp. 398-403). IEEE.
[9] Cai, M., Shi, Y., & Liu, J. (2013, December). Deep maxout neural networks for speech recognition. In Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on (pp. 291-296). IEEE.
[10] Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In Proceedings of the 30th International Conference on Machine Learning (ICML-13) (pp. 1058-1066).
[11] Frazão, X., & Alexandre, L. A. (2014). DropAll: Generalization of Two Convolutional Neural Network
[12] Regularization Methods. In Image Analysis and Recognition (pp. 282-289). Springer International Publishing.
[13] Ba, J., & Frey, B. (2013). Adaptive dropout for training deep neural networks. In Advances in Neural Information Processing Systems (pp. 3084-3092).
[14] Li, K., Huang, Z., Cheng, Y. C., & Lee, C. H. (2014, May). A maximal figure-of-merit learning approach to maximizing mean average precision with deep neural network based classifiers. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on (pp. 4503-4507). IEEE.
[15] Bengio, Y. (2009). Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1), 1-127.
[16] Sagarika Sahoo, Nishtha Tripathi, Avani Jadeja. Neural Tensor Networks for capturing new facts from knowledge bases. International Journal of Computer Science and Mobile Computing, Vol.3 Issue.10, October- 2014, pg. 860-863.
[17] Prechelt, L. (1998). Early stopping-but when?. In Neural Networks: Tricks of the trade (pp. 55-69). Springer Berlin Heidelberg.
[18] Caruana, R., Lawrence, S., & Giles, L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems, 402-408.
[19] Ha, K., Cho, S., & MacLachlan, D. (2005). Response models based on bagging neural networks. Journal of Interactive Marketing, 19(1), 17-30.
[20] Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural computation, 4(1), 1-58.
[21] Li Deng (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing, 3, e2 doi:10.1017/atsip.2013.9
[22] Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19, 153
[23] Hinton, G. E.; Sejnowski, T. J. (1986). D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, ed. "Learning and Relearning in Boltzmann Machines". Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations (Cambridge: MIT Press): 282–317.
[24] Hinton, Geoffrey, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18, no. 7 (2006): 1527-1554.
[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research, 11:3371–3408, 2010.
[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
[27] Srivastava, N. (2013). Improving neural networks with dropout (Master's Thesis University of Toronto).

[28] Frazão, X., & Alexandre, L. A. (2014). Weighted Convolutional Neural Network Ensemble. In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (pp. 674-681). Springer International Publishing.
[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998
[30] Krizhevsky, Alex and Hinton, Geoffrey. Learning multiplelayers of features from tiny images. Technical report,University of Toronto, 2009.
[31] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y. Ng Reading Digits in Natural Images with Unsupervised Feature Learning NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.
[32] Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557.
[33] Y. LeCun, F.J. Huang, L. Bottou, Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. CVPR 2004
[34] S. Haykin, Neural Networks - A Comprehensive Foundation, Maxwell MacMillian Int., New York, 1994.
[35] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

