Dropout: A Simple Way to Prevent Neural Network from Overfitting

JMLR 2014

Juneyoung Yoon
EECS of GIST College

Paper Information

Paper : Dropout: A Simple Way to Prevent Neural Networks from Overfitting

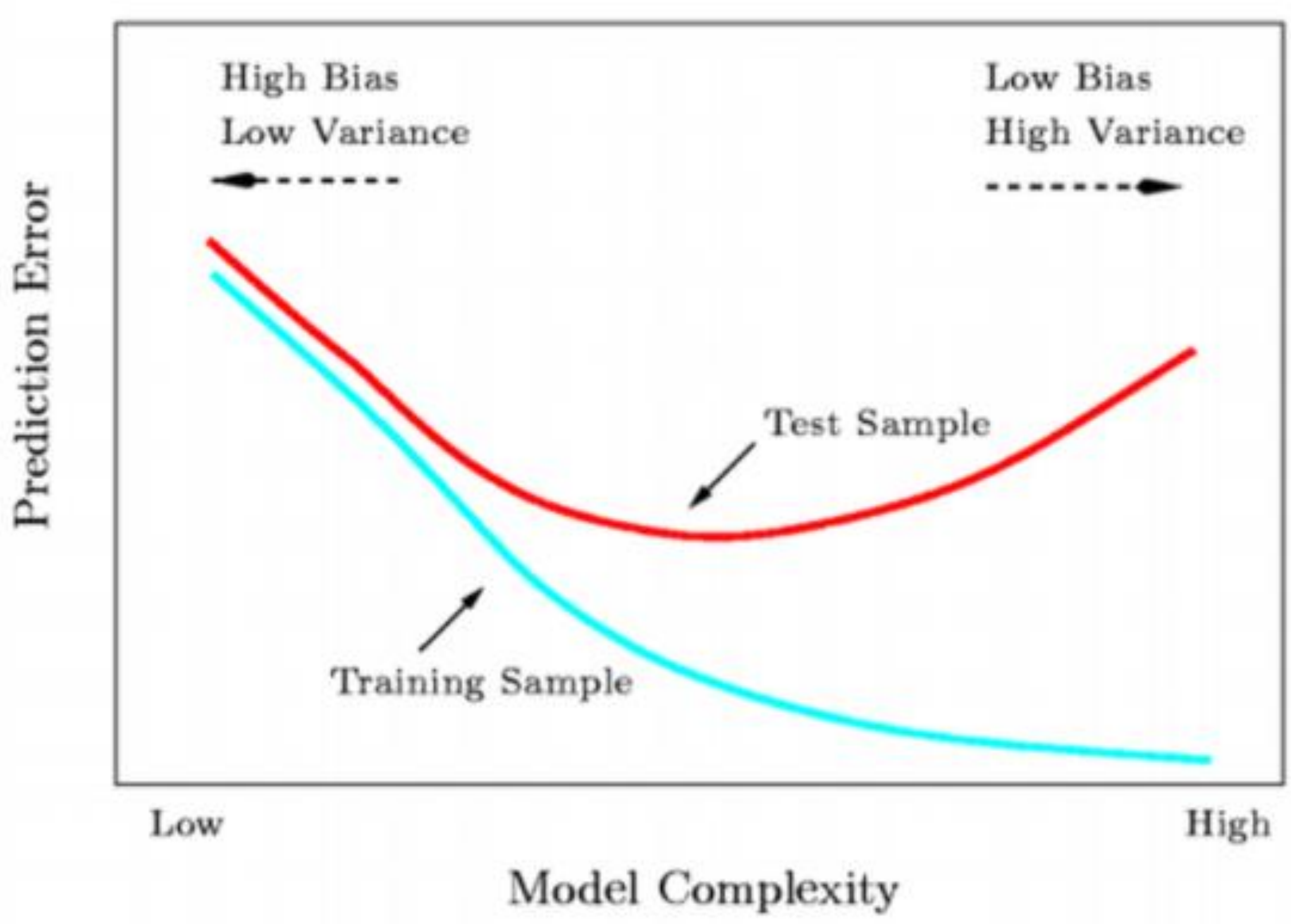
Authors : Nitish Srivastava, et. al. 4

Journal : Journal of Machine Learning Research (JMLR 2014)

Total citation: 25,230

Main Problem

- Overfitting



Previous Research

- Regularization
  - L1/L2 regularization
  - Adding noise to hidden layer

- Data augmentation
  - Affine Transform
  - Elastic Distortion

Dropout

Model combination can improve the performance

But

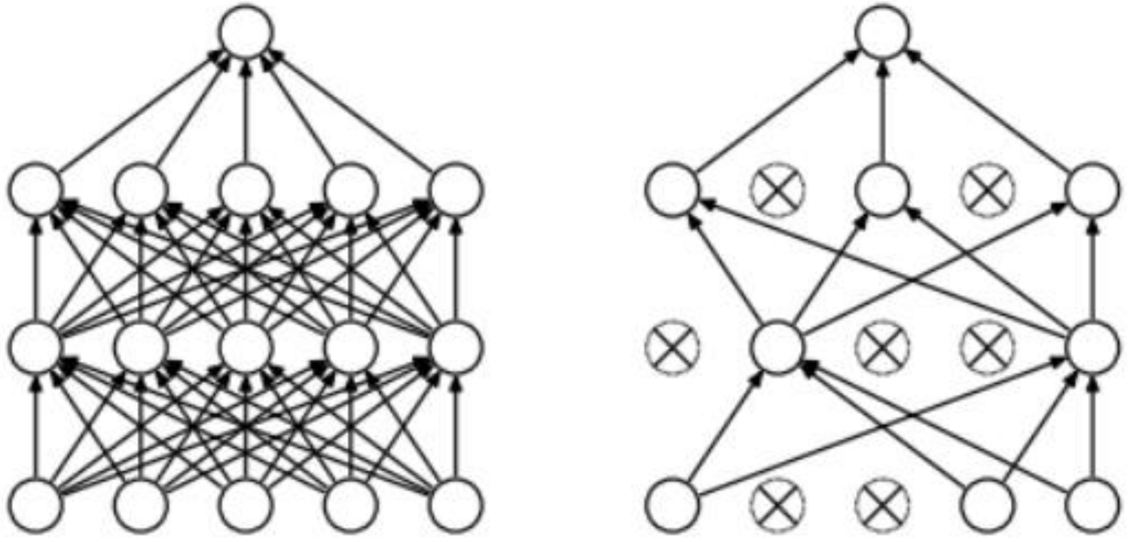
1. Each model should have **different architectures** or be **trained in different data**
2. Training each large network **requires a lot of computation**

\+ Large network **require large amount of data**

Dropout is designed to address both these issues

Dropout

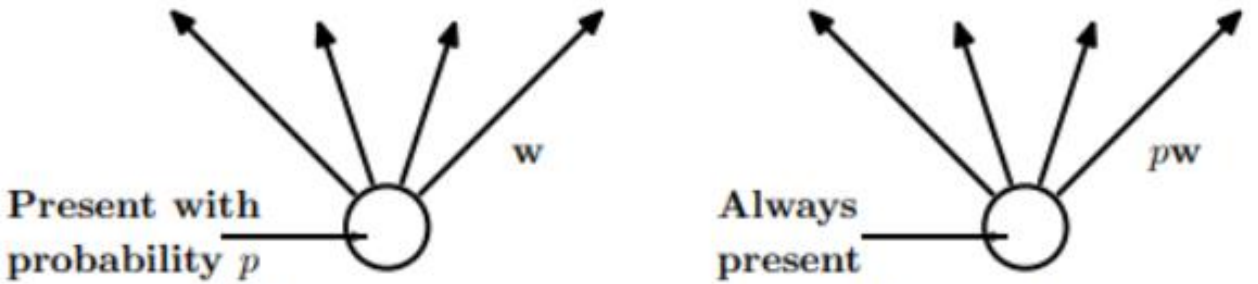
- Prevents overfitting and provide way to combining different neural network
    - It randomly drops(removes) unit from the network
    - All the units survived dropout is called **thinned network**
    - A neural network that has n units, can collect 2^n possible thinned network



(a) Standard Neural Net (b) After applying dropout.

Dropout

- Network shared weight can be combined into single network(test time)
  - **Present retained with probability *p*** is connected to units in next layer during training
  - **Outgoing weight is multiplied by *p*** at test time
  - It leads to lower generalization error compare to other regularization methods



(a) At training time (b) At test time

Figure 2: **Left**: A unit at training time that is present with probability p and is connected to units in the next layer with weights $\mathbf{w}$. **Right**: At test time, the unit is always present and the weights are multiplied by p. The output at test time is same as the expected output at training time.

Dropout

- Model
  - Neural network with L hidden Layers. $l \in \{1, \dots, L\}$
  - $\boldsymbol{z}^{(l)}$: vector of inputs into layer l
  - $\boldsymbol{y}^{(l)}$: vector from layer l
  - $W^{(l)}$: weights bias at layer l, $b^{(l)}$: bias at layer l
  - f : any activation function

$\boldsymbol{r}^{(l)}$: vector of independent Bernoulli random variable
$\widetilde{\boldsymbol{y}}^{(l)}$: thinned outputs

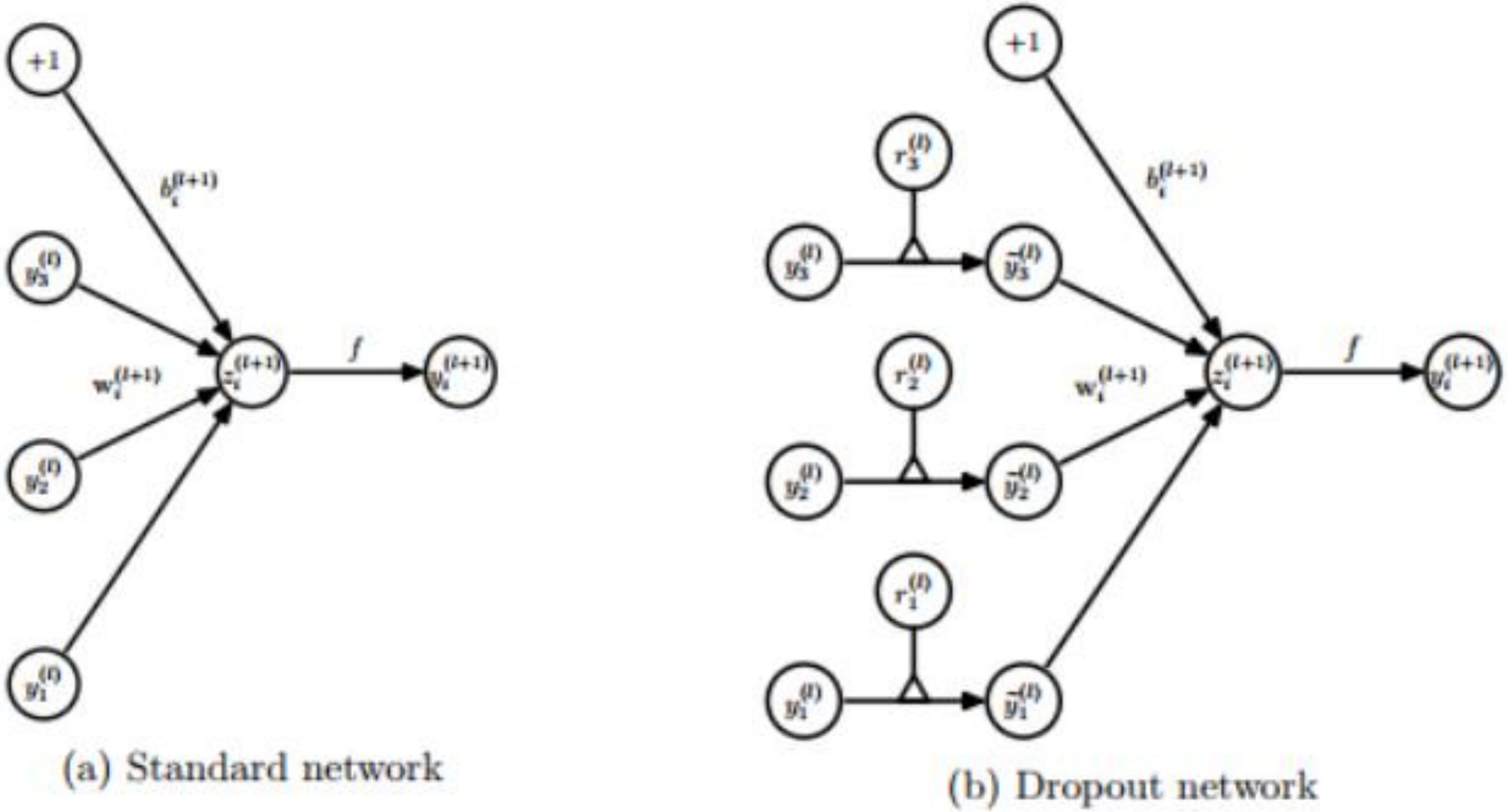
$$
\begin{aligned}
z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}, \\
y_i^{(l+1)} &= f(z_i^{(l+1)}),
\end{aligned}
$$

$$
\begin{aligned}
r_j^{(l)} &\sim \text{Bernoulli}(p), \\
\widetilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\
z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\widetilde{\mathbf{y}}^l + b_i^{(l+1)}, \\
y_i^{(l+1)} &= f(z_i^{(l+1)}).
\end{aligned}
$$

Standard Neural Network **Dropout Neural Network**

Dropout

- Model

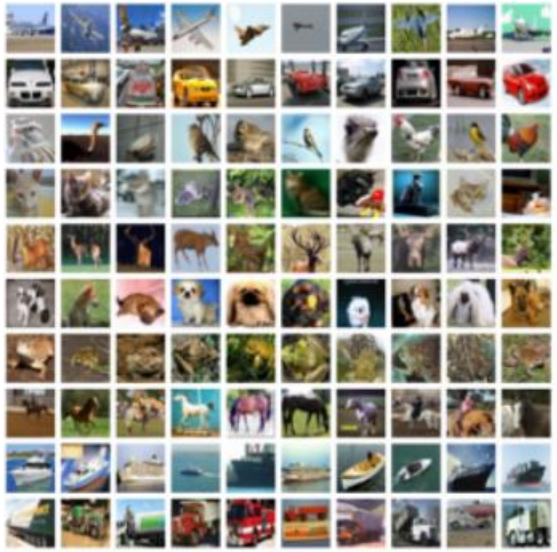


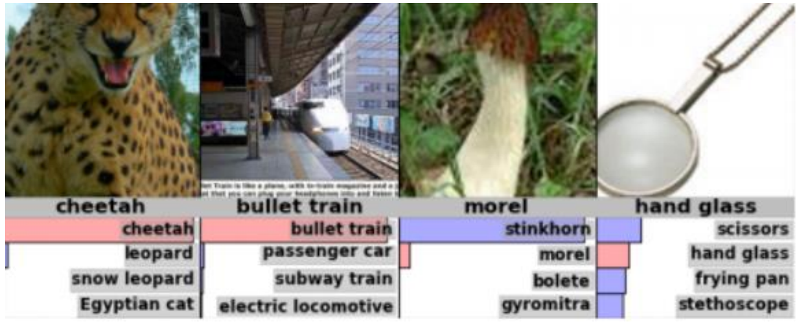
Dropout

- Training dropout neural nets
  - Regularization like momentum, annealed learning rates, L1/L2 weight decay, works well
  - **Max-norm** is especially useful for dropout neural nets
  - Max-norm: optimize weight $\boldsymbol{w}$ under constraint $||\boldsymbol{w}||_{\mathbf{2}} \leq c$ (c is a tunable hyperparameter)
  - Other regularization methods provides **boost over just using dropout**

Dropout

- Experiment
  - On **Image Data Sets**
    - MNIST
    - Street View House Numbers(SVHN)
    - CIFAR-10, CIFAR-100
    - ImageNet
  - On **Voice Data**
    - TIMIT
  - On a **Text Data Set**
    - Reuters-RCV1





Dropout

- Experiment
  - On **Image Data Sets**
    - MNIST

Method	Unit Type	Architecture	Error %
Standard Neural Net (Simard et al., 2003)	Logistic	2 layers, 800 units	1.60
SVM Gaussian kernel	NA	NA	1.40
Dropout NN	Logistic	3 layers, 1024 units	1.35
Dropout NN	ReLU	3 layers, 1024 units	1.25
Dropout NN + max-norm constraint	ReLU	3 layers, 1024 units	1.06
Dropout NN + max-norm constraint	ReLU	3 layers, 2048 units	1.04
Dropout NN + max-norm constraint	ReLU	2 layers, 4096 units	1.01
Dropout NN + max-norm constraint	ReLU	2 layers, 8192 units	0.95
Dropout NN + max-norm constraint (Goodfellow et al., 2013)	Maxout	2 layers, (5×240) units	0.94
DBN + finetuning (Hinton and Salakhutdinov, 2006)	Logistic	500-500-2000	1.18
DBM + finetuning (Salakhutdinov and Hinton, 2009)	Logistic	500-500-2000	0.96
DBN + dropout finetuning	Logistic	500-500-2000	0.92
DBM + dropout finetuning	Logistic	500-500-2000	**0.79**

Table 2: Comparison of different models on MNIST.

The MNIST data set consists of 28×28 pixel handwritten digit images. The task is to classify the images into 10 digit classes. Table 2 compares the performance of dropout with other techniques. The best performing neural networks for the permutation invariant

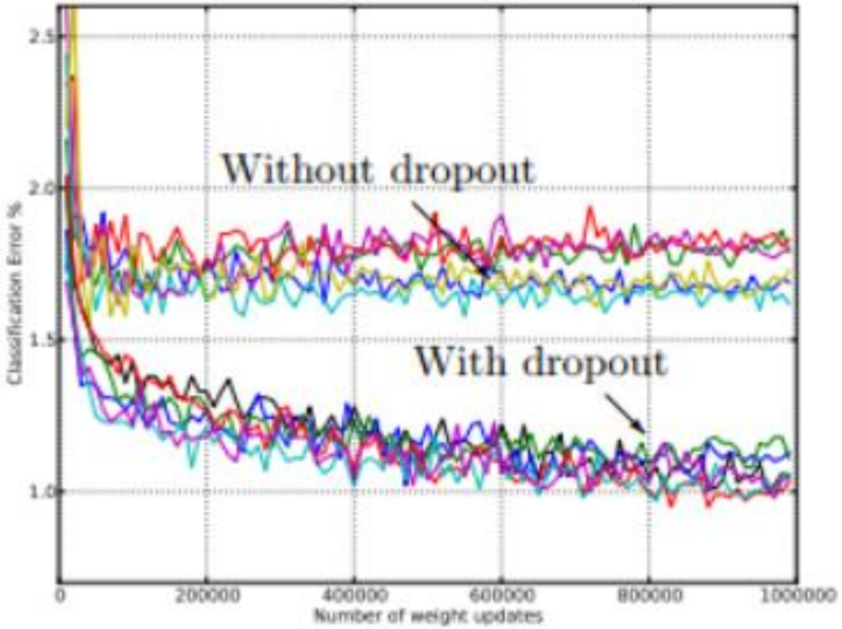


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

Dropout

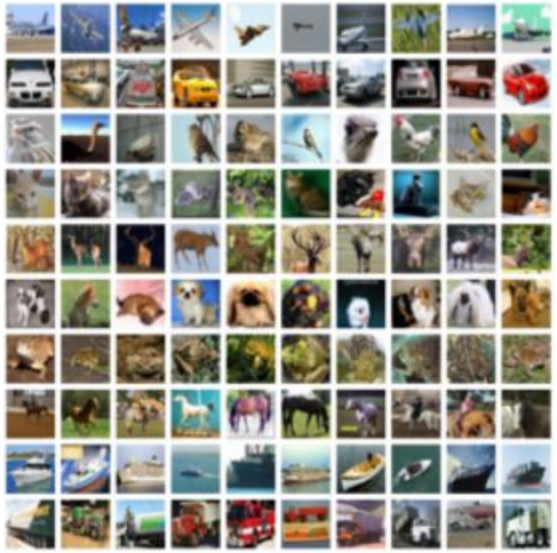
- Experiment
  - On **Image Data Sets**
    - Street View House Numbers(SVHN)

Method	Error %
Binary Features (WDCH) (Netzer et al., 2011)	36.7
HOG (Netzer et al., 2011)	15.0
Stacked Sparse Autoencoders (Netzer et al., 2011)	10.3
KMeans (Netzer et al., 2011)	9.4
Multi-stage Conv Net with average pooling (Sermanet et al., 2012)	9.06
Multi-stage Conv Net + L2 pooling (Sermanet et al., 2012)	5.36
Multi-stage Conv Net + L4 pooling + padding (Sermanet et al., 2012)	4.90
Conv Net + max-pooling	3.95
Conv Net + max pooling + dropout in fully connected layers	3.02
Conv Net + stochastic pooling (Zeiler and Fergus, 2013)	2.80
Conv Net + max pooling + dropout in all layers	2.55
Conv Net + maxout (Goodfellow et al., 2013)	**2.47**
Human Performance	2.0

Table 3: Results on the Street View House Numbers data set.

Dropout

- Experiment
  - On **Image Data Sets**
    - CIFAR-10, CIFAR-100

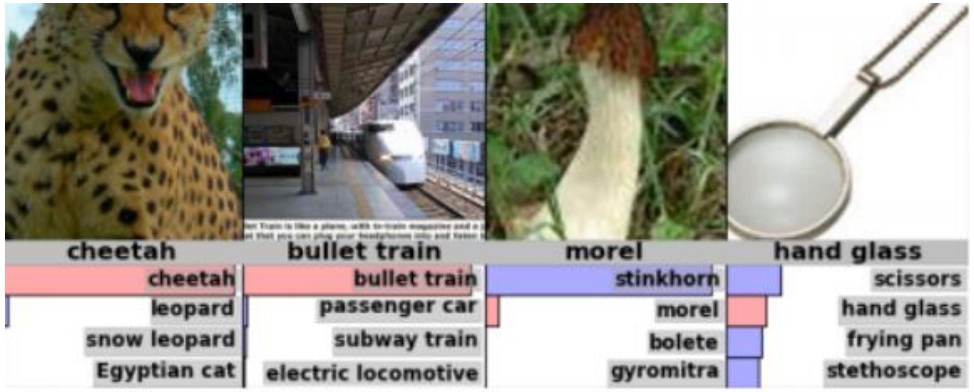


Method	CIFAR-10	CIFAR-100
Conv Net + max pooling (hand tuned)	15.60	43.48
Conv Net + stochastic pooling (Zeiler and Fergus, 2013)	15.13	42.51
Conv Net + max pooling (Snoek et al., 2012)	14.98	-
Conv Net + max pooling + dropout fully connected layers	14.32	41.26
Conv Net + max pooling + dropout in all layers	12.61	**37.20**
Conv Net + maxout (Goodfellow et al., 2013)	**11.68**	38.57

Table 4: Error rates on CIFAR-10 and CIFAR-100.

Dropout

- Experiment
  - On **Image Data Sets**
    - ImageNet



Model	Top-1	Top-5
Sparse Coding (Lin et al., 2010)	47.1	28.2
SIFT + Fisher Vectors (Sanchez and Perronnin, 2011)	45.7	25.7
Conv Net + dropout (Krizhevsky et al., 2012)	37.5	17.0

Table 5: Results on the ILSVRC-2010 test set.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

Table 6: Results on the ILSVRC-2012 validation/test set.

Dropout

- Experiment
  - On **Voice Data**
    - TIMIT
    - Recordings from 680 speakers covering 8 major dialects of American English

Method	Phone Error Rate%
NN (6 layers) (Mohamed et al., 2010)	23.4
Dropout NN (6 layers)	21.8
DBN-pretrained NN (4 layers)	22.7
DBN-pretrained NN (6 layers) (Mohamed et al., 2010)	22.4
DBN-pretrained NN (8 layers) (Mohamed et al., 2010)	20.7
mcRBM-DBN-pretrained NN (5 layers) (Dahl et al., 2010)	20.5
DBN-pretrained NN (4 layers) + dropout	**19.7**
DBN-pretrained NN (8 layers) + dropout	**19.7**

Table 7: Phone error rate on the TIMIT core test set.

Dropout

- Experiment
  - On a **Text Data Set**
    - Reuters-RCV1
    - Collection of 800,000 newswire article from Reuters
    - Not use dropout (error rate 31.05%)
    - Use dropout (error rate 29.62%)

Dropout

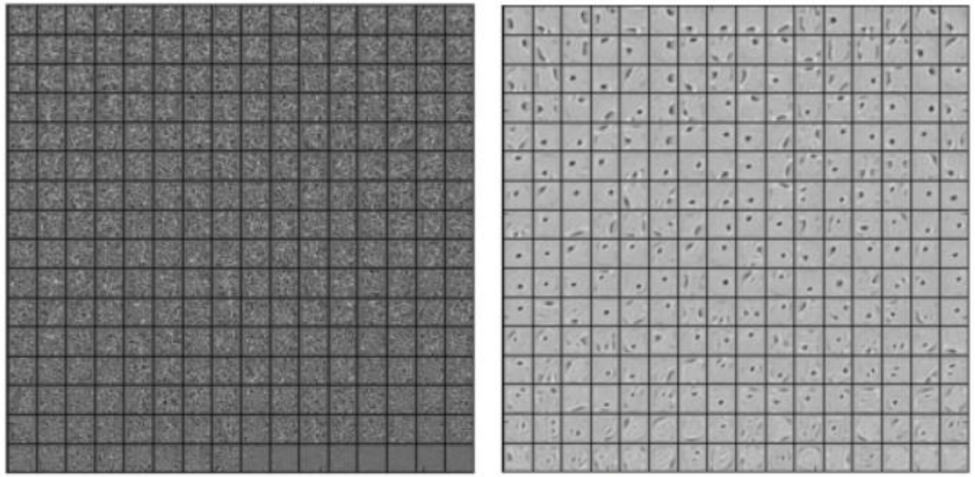
- Comparison with Standard Regularizers

Method	Test Classification error %
L2	1.62
L2 + L1 applied towards the end of training	1.60
L2 + KL-sparsity	1.55
Max-norm	1.35
Dropout + L2	1.25
Dropout + Max-norm	**1.05**

Table 9: Comparison of different regularization methods on MNIST.

Dropout

- Effect on Features & Sparsity
  - Feature
  - Dropout prevents co-adaption by making the presence of other hidden units unreliable
  - By using dropout, the hidden units are detected well
  - It is probably the main reason why it has lower generalization error

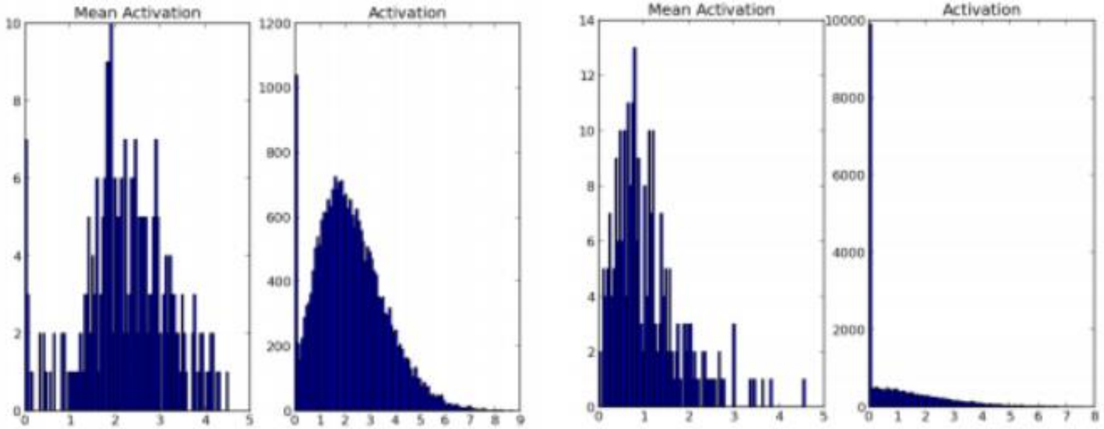


(a) Without dropout (b) Dropout with $p = 0.5$.

Figure 7: Features learned on MNIST with one hidden layer autoencoders having 256 rectified linear units.

Dropout

- Effect on Features & Sparsity
  - Sparsity
  - Using Dropout has fewer hidden units that have high activations
  - The mean activation is also smaller for the dropout net



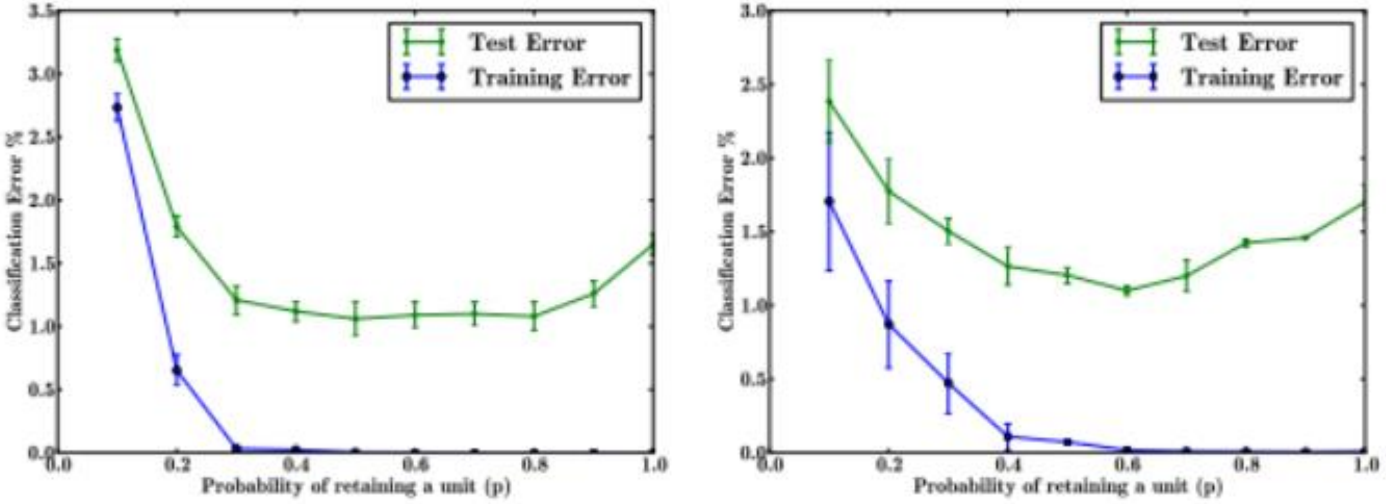
(a) Without dropout

(b) Dropout with $p = 0.5$.

Figure 8: Effect of dropout on sparsity. ReLUs were used for both models. **Left**: The histogram of mean activations shows that most units have a mean activation of about 2.0. The histogram of activations shows a huge mode away from zero. Clearly, a large fraction of units have high activation. **Right**: The histogram of mean activations shows that most units have a smaller mean mean activation of about 0.7. The histogram of activations shows a sharp peak at zero. Very few units have high activation.

Dropout

- Effect of Dropout Rate
  - Dropout Rate(tunable hyperparameter p – the probability of retaining a unit in the net)
  - 1. The number of units is held constant
  - 2. The number of hidden units changed so that the expected number of hidden units that will be retained after dropout is held constant()



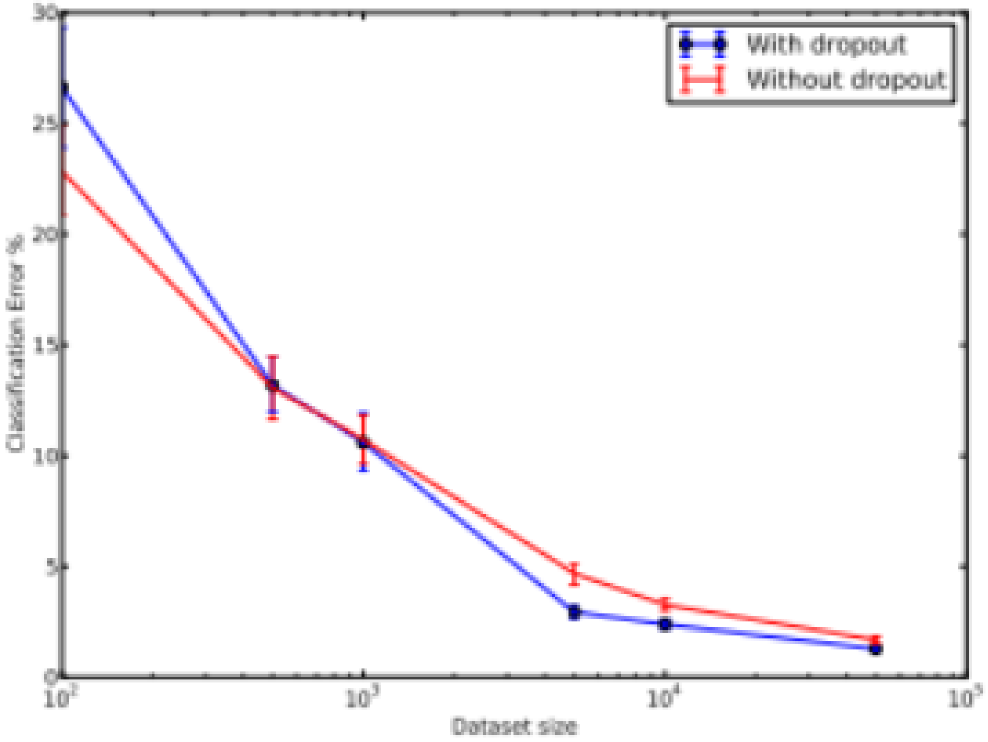
(a) Keeping n fixed. (b) Keeping pn fixed.

1. n is fixed
   - It becomes flat when $0.4 \le p \le 0,8$
2. pn is fixed
   - values of p close to 0.6 perform best
   - use default value 0.5
   - have to increase units to do dropout and lower the effect of underfitting

Dropout

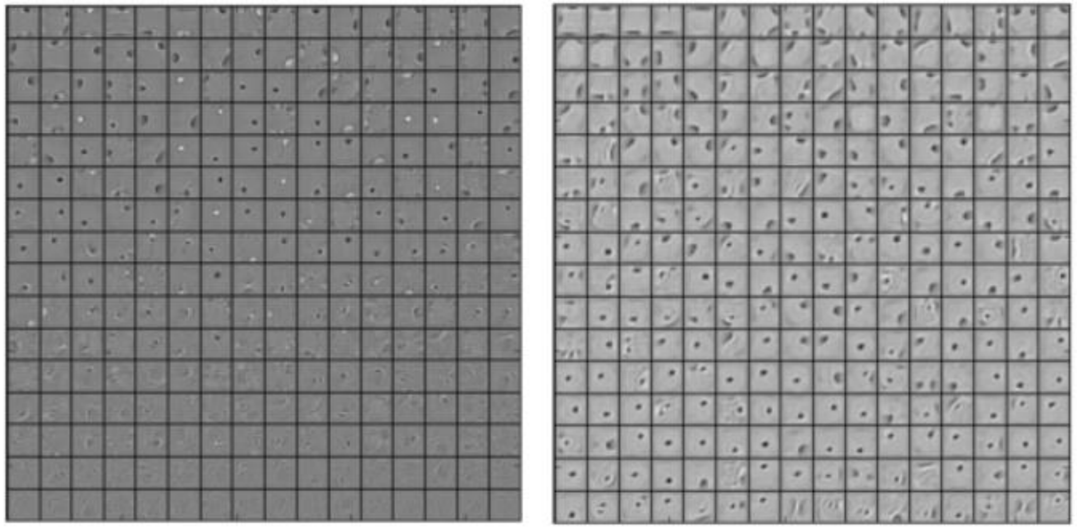
- Effect of Data Size
  - The network was given data sets of size 100,500,1K,5K,10K and 50K chosen randomly from the MNIST

  - Extremely small data doesn't improve
  - As the size of data increase, dropout works well(more than 1K)
  - If data size gets large enough the effect of dropout gets smaller



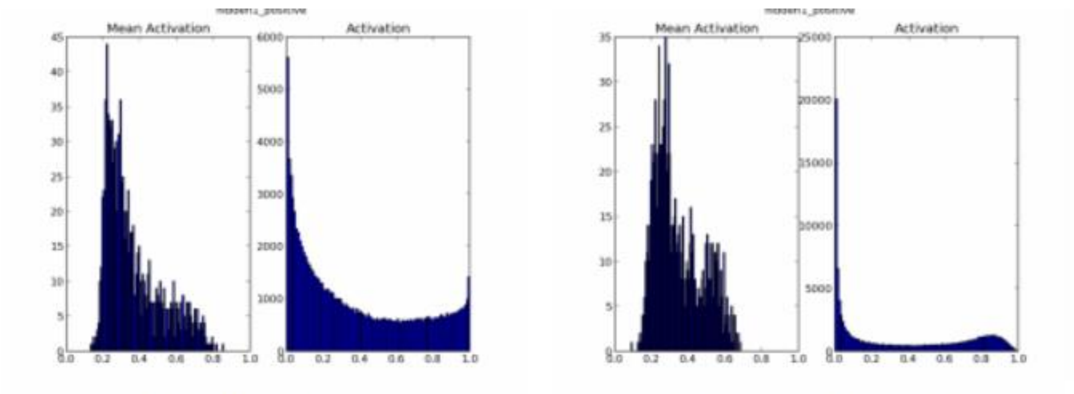
Dropout

- Dropout in Restricted Boltzmann machine(RBM)
  - One of graphical probabilistic model
  - Dropout sharpen the feature and hidden unit activation gets more sparse



(a) Without dropout (b) Dropout with $p = 0.5$.

Figure 12: Features learned on MNIST by 256 hidden unit RBMs. The features are ordered by L2 norm.



(a) Without dropout (b) Dropout with $p = 0.5$.

Figure 13: Effect of dropout on sparsity. **Left**: The activation histogram shows that a large number of units have activations away from zero. **Right**: A large number of units have activations close to zero and very few units have high activation.

Dropout

- Conclusion
  - Dropout improve the performance of neural nets in a wide variety
  - Dropout considerably improved the performance of standard neural nets
  - Dropout is an effective way to reduce Overfitting