Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

Authors:

Nitish Srivastava
Geoffrey Hinton
Alex Krizhevsky
Ilya Sutskever
Ruslan Salakhutdinov

Presented by:

Christoph Schmidl
4226887
c.schmidl@student.ru.nl

17.05.2018
Research Seminar Data Science

Radboud University

IN DEI NOMINE FELICITER

Outline

1. Authors
2. General Information
3. Summary
4. Review
5. Discussion

Authors

Nitish Srivastava

- Publication years: 2012 - 2015
- Publication count: 7
- Citation count: 521

- Master’s Thesis: “Improving Neural Networks with Dropout”, University of Toronto, January 2013
- Received his PhD in 2017 under supervision of Geoffrey Hinton and Ruslan Salakhutdinov: “Deep Learning Models for Unsupervised and Transfer Learning”
- Other publications are related to Deep Boltzmann Machines

Authors

Geoffrey Everest Hinton

- Publication years: 1983 - 2017
- Publication count: 168
- Citation count: 7380

- The “Godfather of Deep Learning”
- Co-invented Boltzmann machines
- Works for Google and the University of Toronto
- Famous for his image-recognition milestone in Imagenet challenge 2012 and his work on Artificial Neural Networks in general
- “Geoffrey Hinton spent 30 years on an idea many other scientists dismissed”

Authors

Alex Krizhevsky

- Publication years: 2011 - 2017
- Publication count: 4
- Citation count: 1625

- Invented “AlexNet” with Geoffrey Hinton and Ilya Sutskever
- “AlexNet” contained 8 layers: 5 convolutional and 3 fully connected layers
- Co-Author of the paper “ImageNet Classification with Deep Convolutional Neural Networks”, 2012
  - Relies on Dropout

Authors

Ilya Sutskever

- Publication years: 2008 - 2017
- Publication count: 24
- Citation count: 2469

- Co-founder and Research Director of OpenAI
- Research Scientist at the Google Brain Team
- Co-founder of DNNresearch
- Postdoc in Stanford with Andrew Ng’s group

Authors

Ruslan Salakhutdinov

- Publication years: 2002 - 2016
- Publication count: 67
- Citation count: 2114

- Associate Professor of Computer Science at the Carnegie Mellon University
- Director of AI Research at Apple

General Information

- Written in 2012
- Submitted to the Journal of Machine Learning Research in 2013
- Published in 2014

- 388 citations in total
  - Between 2014 and 2018 (still relevant)
- 36 references
  - Yann LeCun, Director of AI Research at Facebook
  - Ian Goodfellow, Staff Research Scientist at Google Brain
    - Inventor of Generative Adversarial Networks

Summary

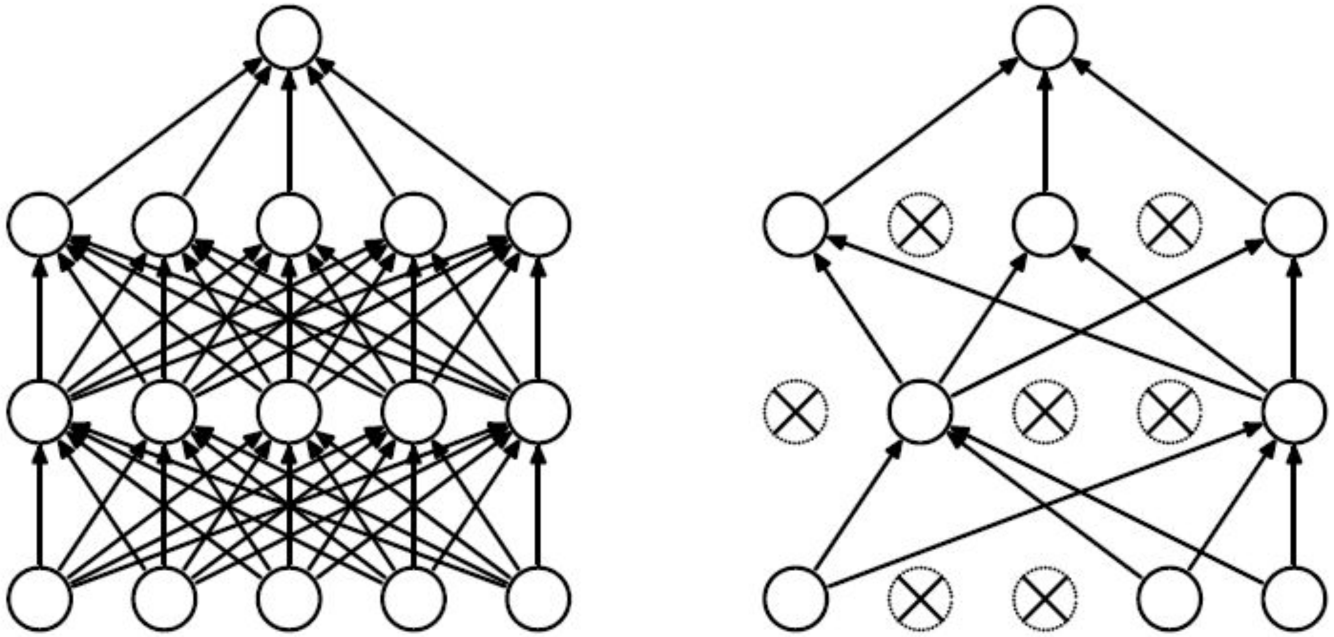
- Introducing a method to prevent overfitting by randomly dropping units along with their connections

- Major improvements over other regularization methods
  - Stop training on certain criteria
  - Weight penalties like L1 and L2 regularization

- Applied to supervised learning tasks
  - Vision (MNIST, CIFAR-10, CIFAR-100, SVHN, ImageNet)
  - Speech Recognition (TIMIT)
  - Document Classification (Reuters-RCV1)
  - Computational Biology (Alternative Splicing Data set)

Summary - Key Idea

- Drop units along with their connections by a certain probability
- Prevent heavy co-adaptations
- Provides a way of approximately combining exponentially many different neural network architectures: “thinned networks”



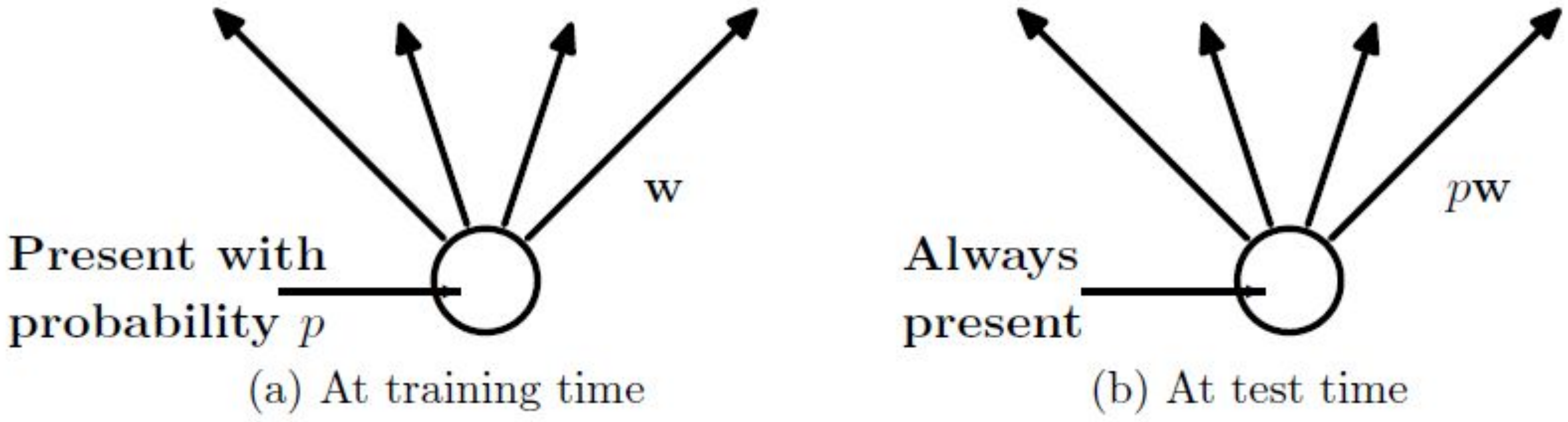
(a) Standard Neural Net

(b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Summary - Key Idea

- p = probability to retain a unit
- Training a network with dropout = Training a collection of 2^n thinned networks with extensive weight sharing
- Testing a network with dropout = Simple approximate averaging method using one network



(a) At training time

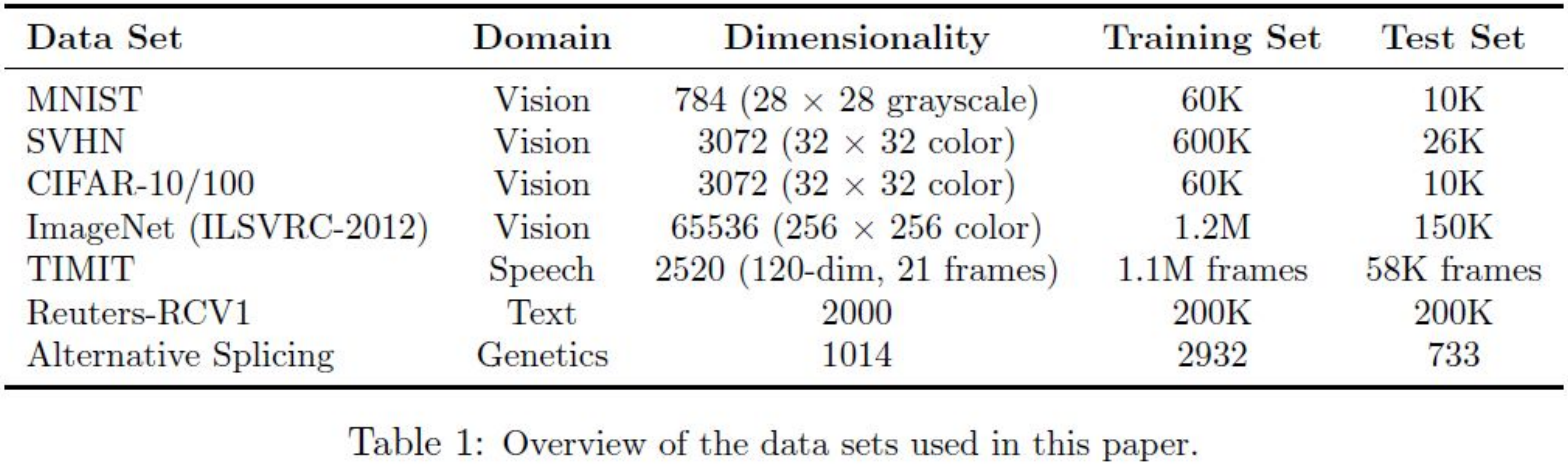
(b) At test time

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 **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.

Summary - Learning Dropout Nets

- Dropout Neural Networks can be training using SGD
  - For each training case in a mini-batch, a thinned network is sampled
  - Forward and backpropagation are solely done on this thinned network
- Max-norm regularization especially useful for dropout
  - Constraining the norm of incoming weight vector to constant c
- Dropout can be used to finetune nets that have been pre-trained by
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines
  - Autoencoders

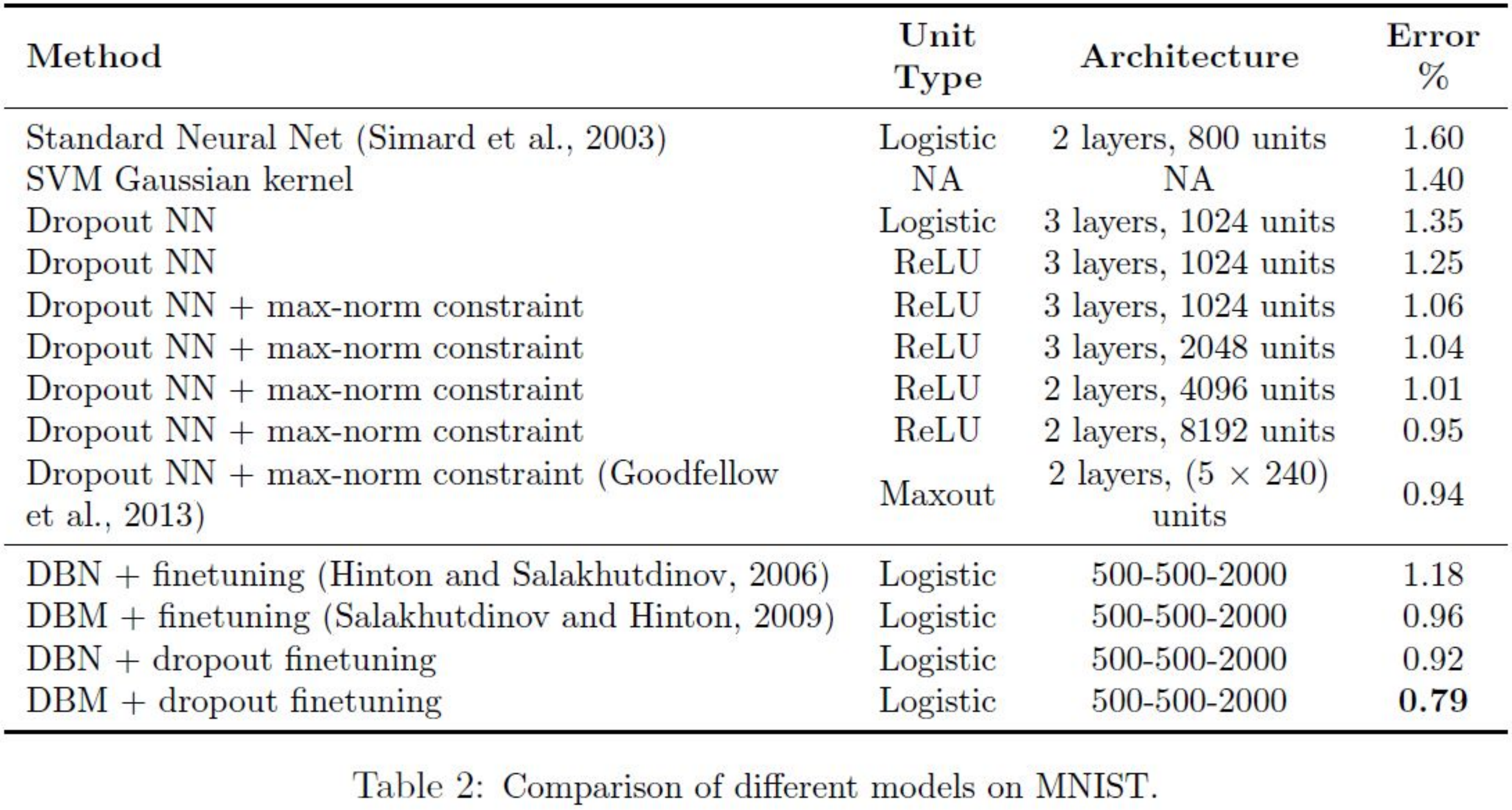
Summary - Used Datasets



Data Set	Domain	Dimensionality	Training Set	Test Set
MNIST	Vision	784 (28 × 28 grayscale)	60K	10K
SVHN	Vision	3072 (32 × 32 color)	600K	26K
CIFAR-10/100	Vision	3072 (32 × 32 color)	60K	10K
ImageNet (ILSVRC-2012)	Vision	65536 (256 × 256 color)	1.2M	150K
TIMIT	Speech	2520 (120-dim, 21 frames)	1.1M frames	58K frames
Reuters-RCV1	Text	2000	200K	200K
Alternative Splicing	Genetics	1014	2932	733

Table 1: Overview of the data sets used in this paper.

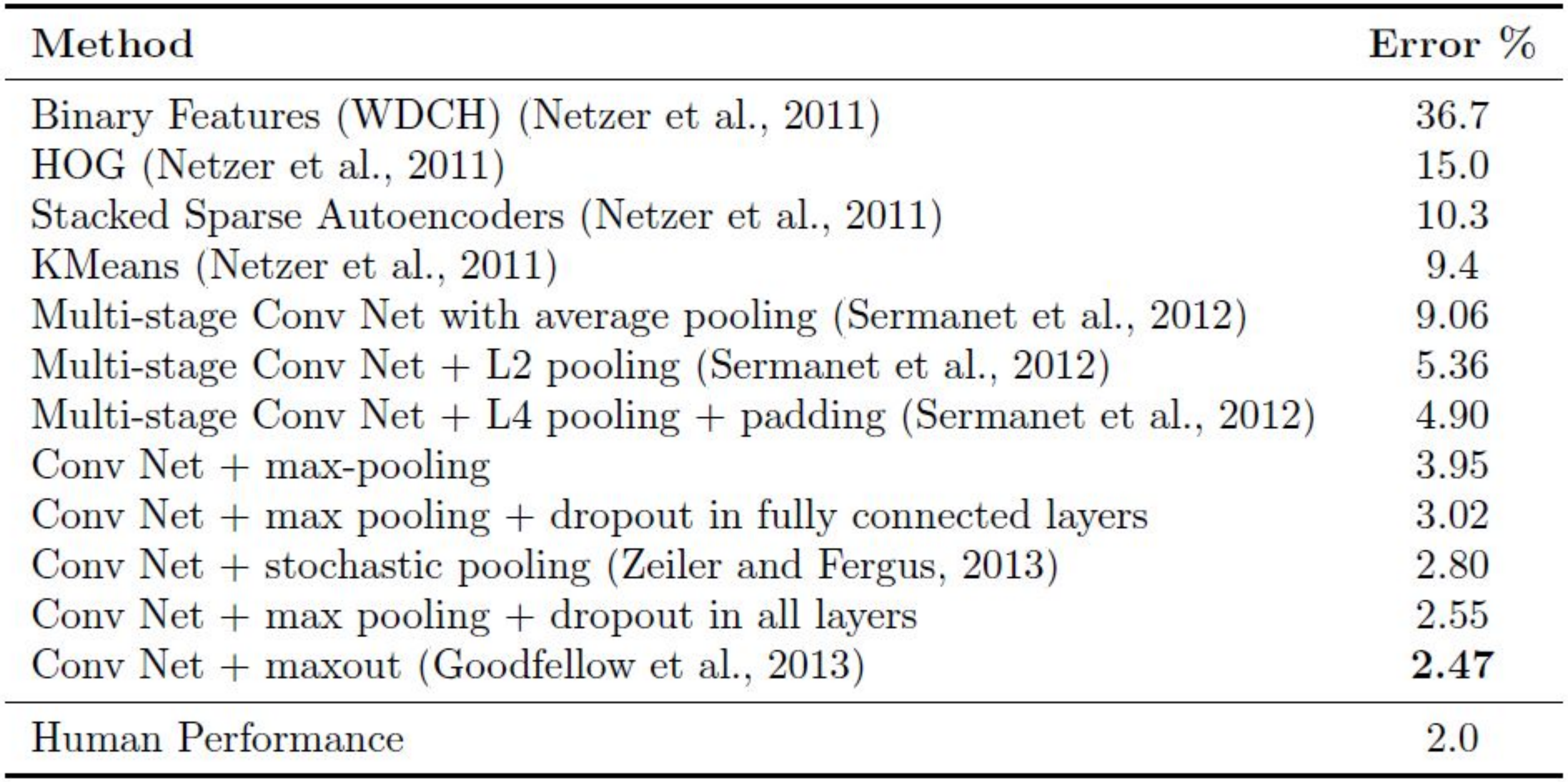
Summary - MNIST Results



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.

Summary - Street View House Numbers Results

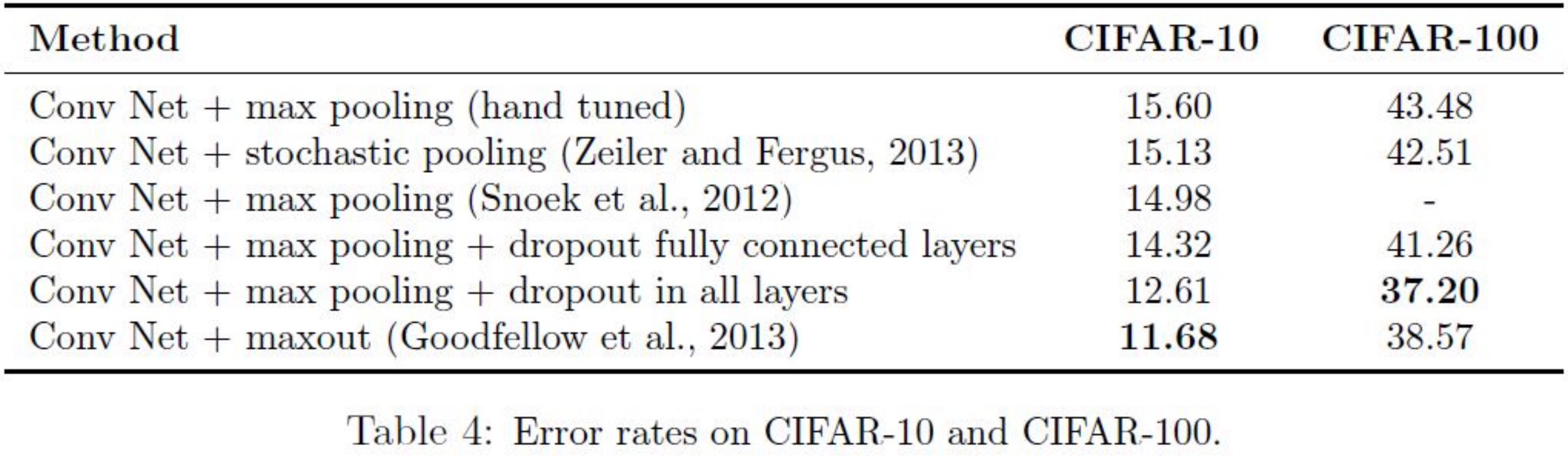


Method	Error %
Binary Features (WDCH) (Netzer et al., 2011)	36.7
HOG (Netzer et al., 2011)	15.0
Stacked Sparse Autoencoders (Netzer et al., 2011)	10.3
KMeans (Netzer et al., 2011)	9.4
Multi-stage Conv Net with average pooling (Sermanet et al., 2012)	9.06
Multi-stage Conv Net + L2 pooling (Sermanet et 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.

- A maxout layer is simply a layer where the activation function is the max of the inputs

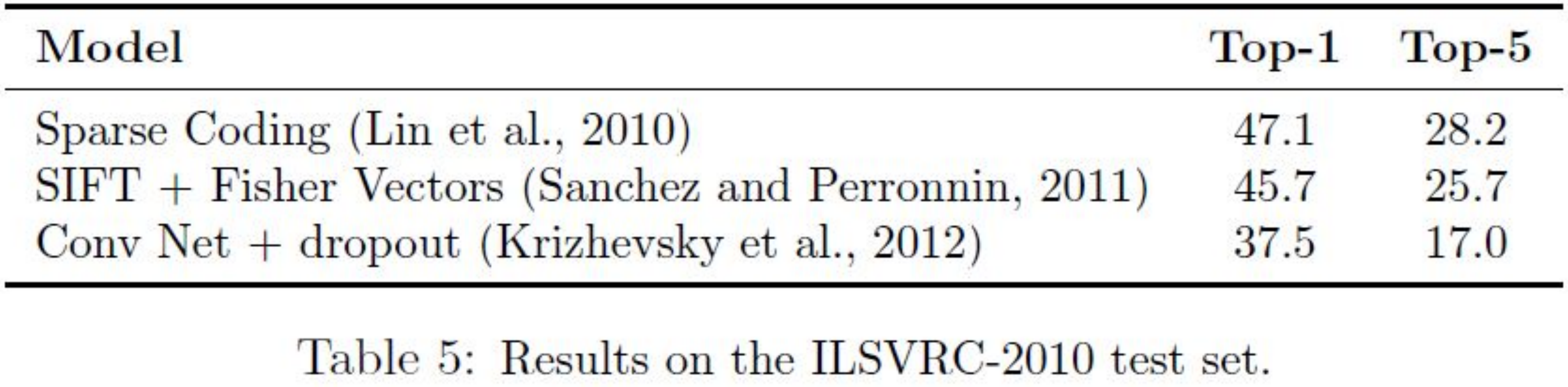
Summary - CIFAR-10 and CIFAR-100 Results



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

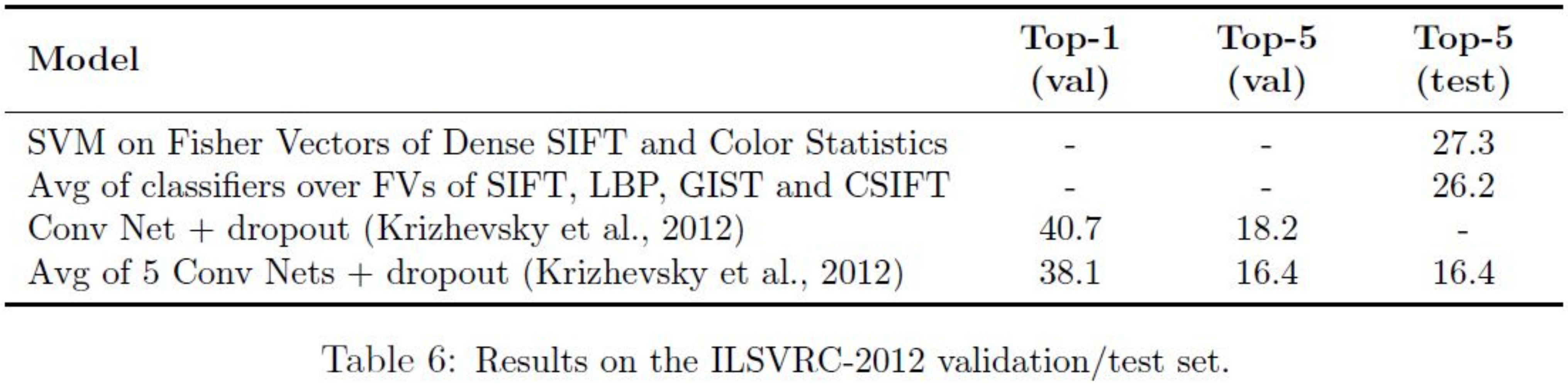
Table 4: Error rates on CIFAR-10 and CIFAR-100.

Summary - ImageNet Results



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

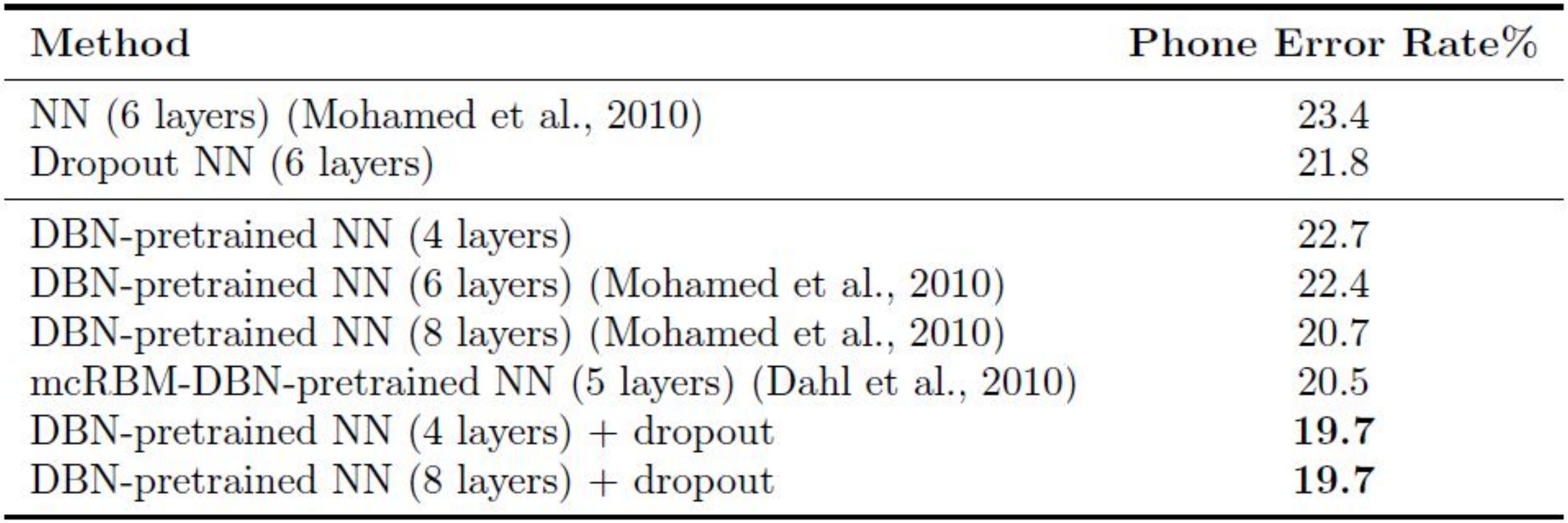
Table 5: Results on the ILSVRC-2010 test set.



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.

Summary - TIMIT Results

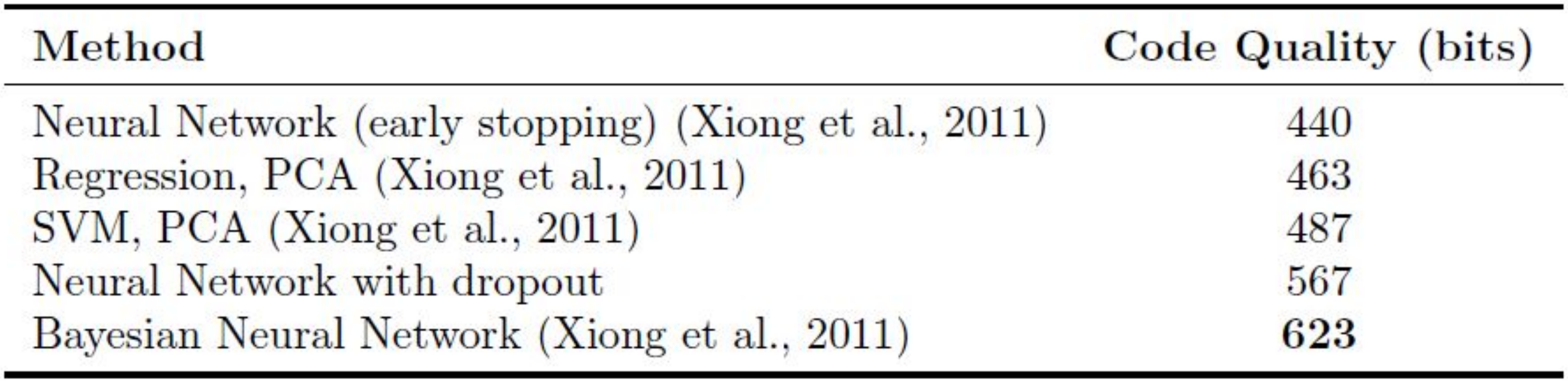


Method	Phone Error Rate%
NN (6 layers) (Mohamed et al., 2010)	23.4
Dropout NN (6 layers)	21.8
DBN-pretrained NN (4 layers)	22.7
DBN-pretrained NN (6 layers) (Mohamed et al., 2010)	22.4
DBN-pretrained NN (8 layers) (Mohamed et al., 2010)	20.7
mcRBM-DBN-pretrained NN (5 layers) (Dahl et al., 2010)	20.5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

Summary - Alternative Splicing Data Set Results

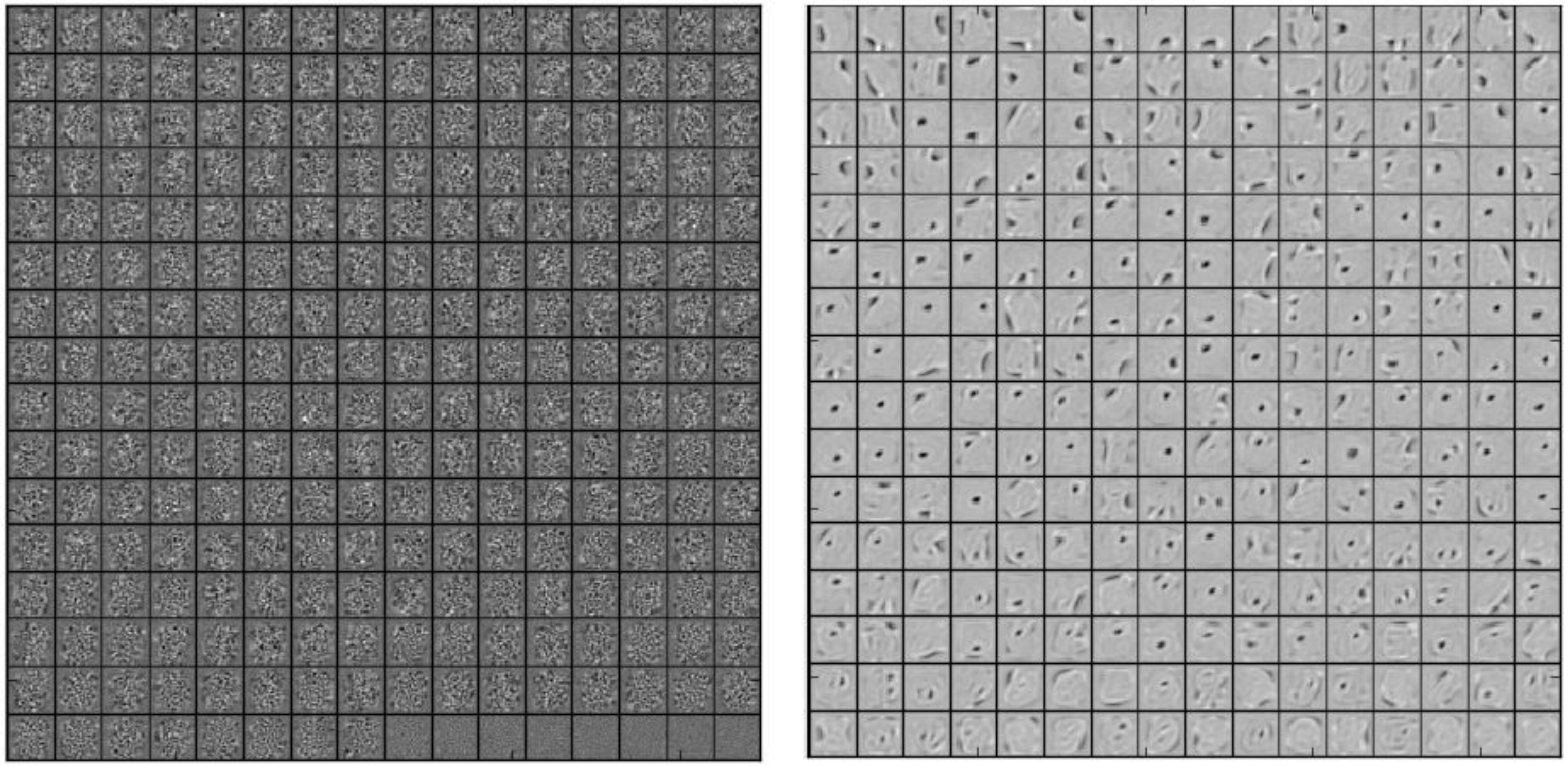
- Comparison with Bayesian Neural Networks
  - Bayesian Neural Networks perform better



Method	Code Quality (bits)
Neural Network (early stopping) (Xiong et al., 2011)	440
Regression, PCA (Xiong et al., 2011)	463
SVM, PCA (Xiong et al., 2011)	487
Neural Network with dropout	567
Bayesian Neural Network (Xiong et al., 2011)	**623**

Table 8: Results on the Alternative Splicing Data Set.

Summary - Dropout Effects

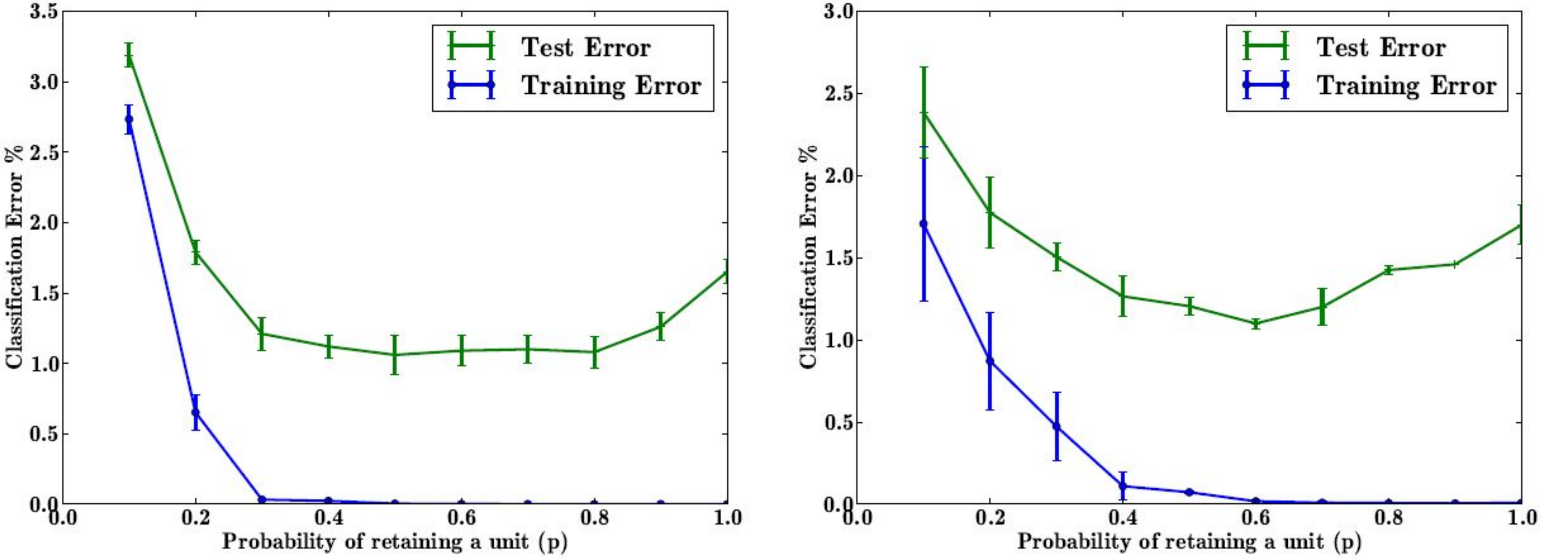


(a) Without dropout

(b) Dropout with $p = 0.5$.

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

Summary - Dropout Effects



(a) Keeping n fixed.

(b) Keeping pn fixed.

Figure 9: Effect of changing dropout rates on MNIST.

Summary - Dropout Effects

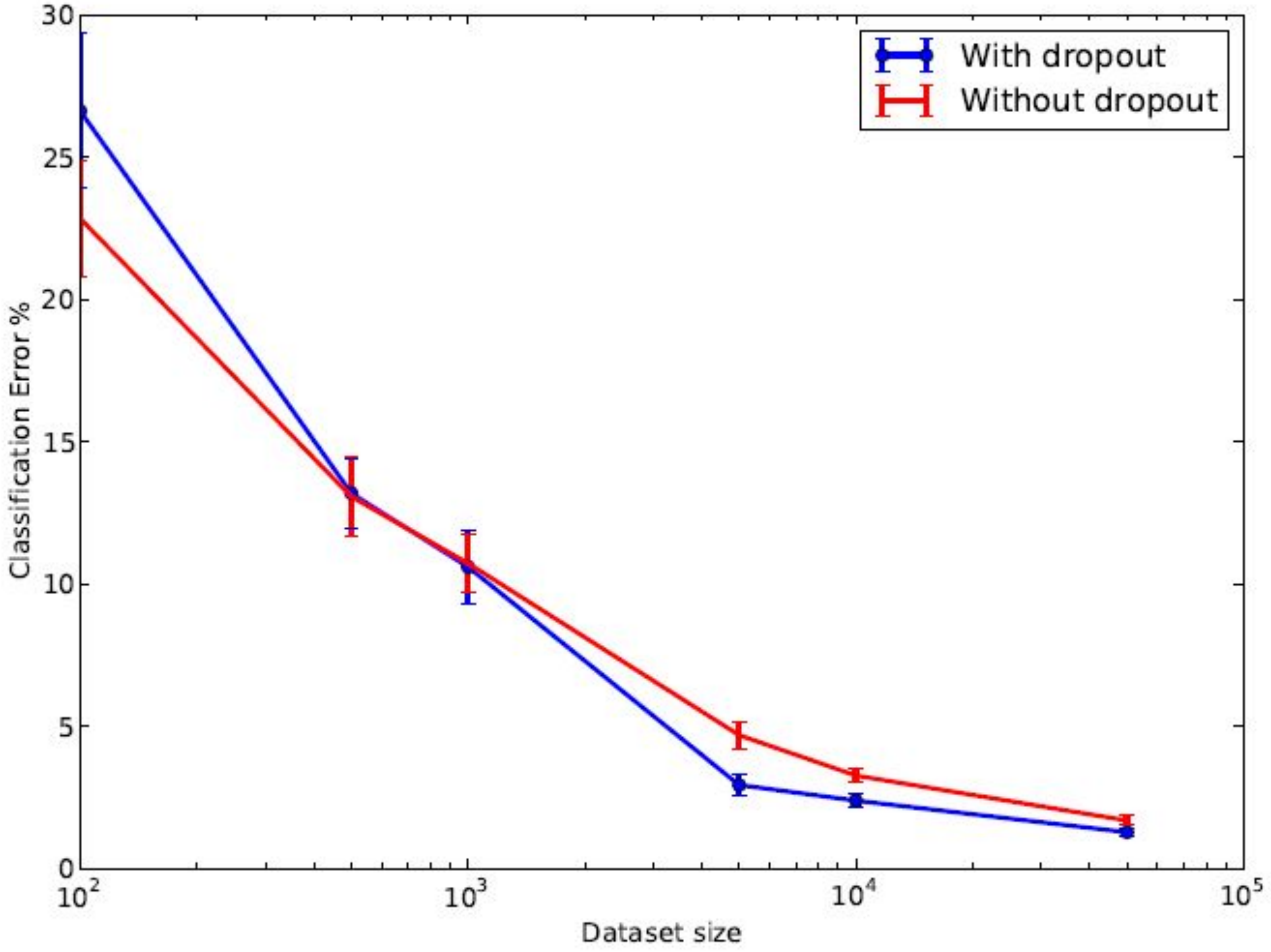


Figure 10: Effect of varying data set size.

Review

- Well written
- Simple but yet effective idea
- Experiments have been chosen to cover different domains
- Practical Guide in Appendix
- Contributed to the more famous paper “ImageNet Classification with Deep Convolutional Neural Networks”

Discussion

- How do you think Dropout compares to other techniques to prevent overfitting, like L2 regularization and Batch Normalization?
- When should Dropout be used over other techniques?
- Should we always use Dropout?
- Do you think the length of the paper was necessary to make a compelling case for the intrusive regularization method?
- It appears that Dropout is not used anymore in a number of large available networks such as Resnet, InceptionV3. Why do you think they do not use Dropout?

Discussion

- Given the paper “Understanding Deep Learning Requires Rethinking Generalization” from the last session, do you think that dropout actually prevents overfitting or might there be another reason why the generalization error is decreased?

- Newer articles hardly ever mention dropout anymore. Many deep supervised models are nowadays done with residual convolutional architectures that make use of batch normalization. Has dropout become redundant?

- Since the authors note that removing any convolutional layer resulted in inferior performance, can we expect better performance if we add even more convolutional layers?

Discussion

- A hyper parameter p with a high value can be interpreted as resulting in too complex co-adaptations between neurons. What could be an interpretation of a higher error associated with a low p value?

- The authors mention a speedup of Dropout is an interesting direction for future work. Have people tried improving Dropout and how much did it improve?

- The authors mention that the gradients that are being computed are not gradients of the final architecture. This increases training time as the network is effectively training a different network every time. Do you think there's a solution for this limitation? Perhaps there's some sort of technique for optimizing these gradient updates such that the training time decreases.

Discussion

- The authors mention that multiplicative Gaussian noise led to equal or better performance when the mean and variance matched a Bernoulli distribution. However, they do not provide any explanation for this finding. Do you have an idea of why this might be better?