INQUEST

SCOSH

DROPOUT

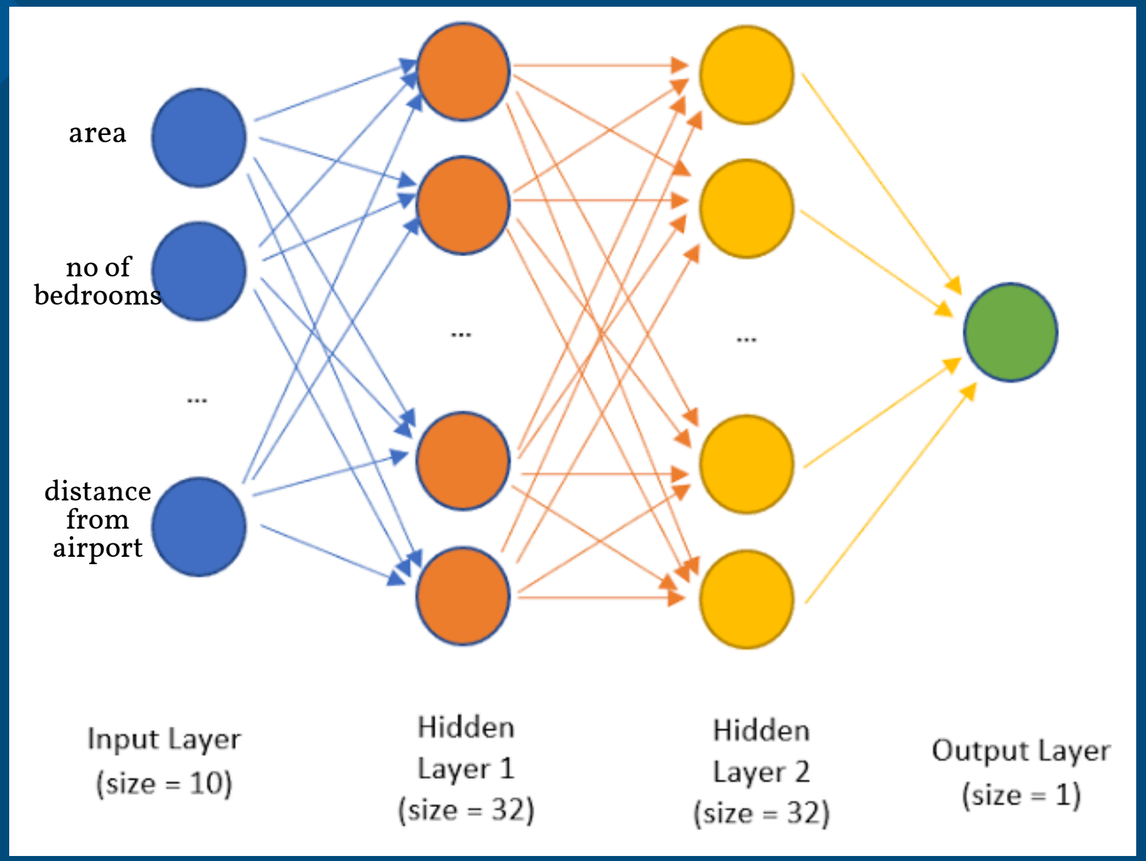
(A simple way to prevent neural network from overfitting)

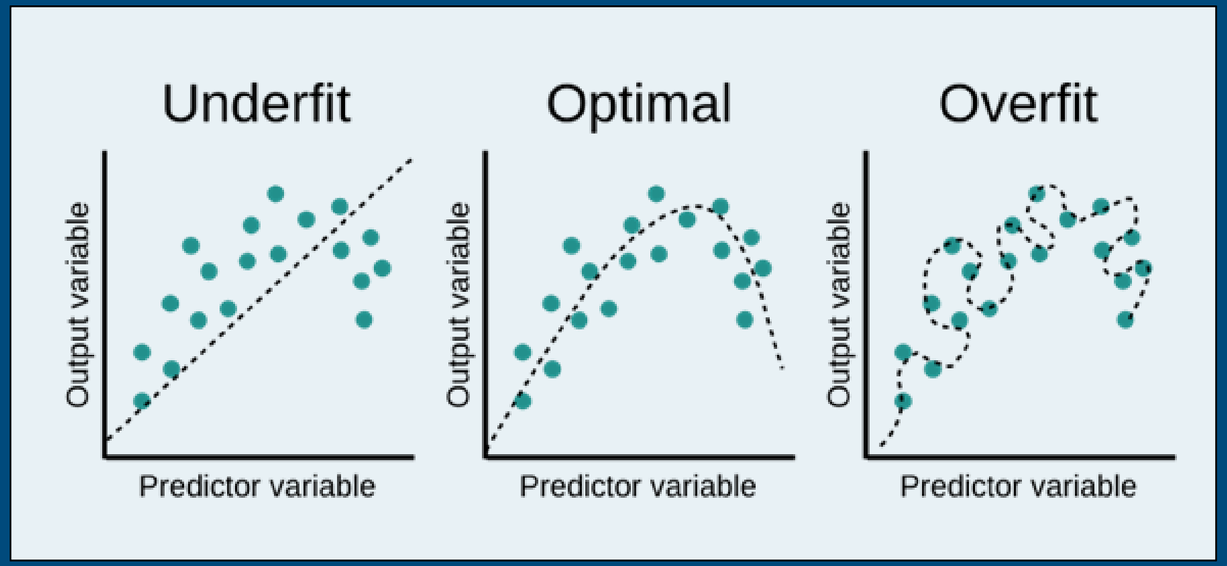
TEAM : EIGEN

Member1: Gaurav Rohilla
Member2: Khushi Jha
Member3: Priyansi

WHY WE NEED DROPOUT?

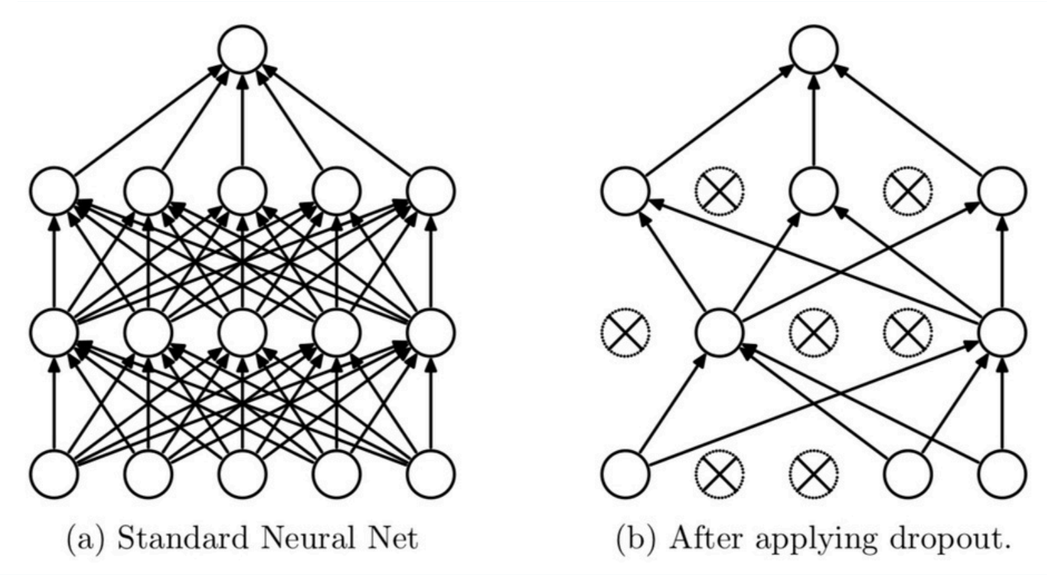
Example: House Price Prediction

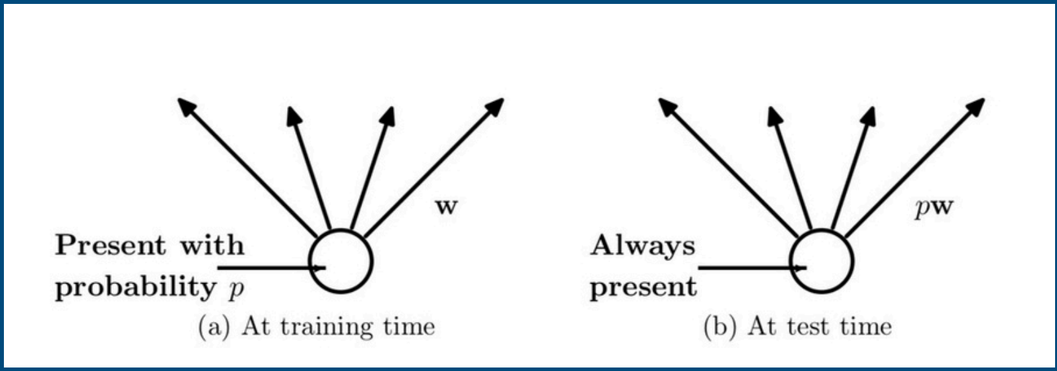


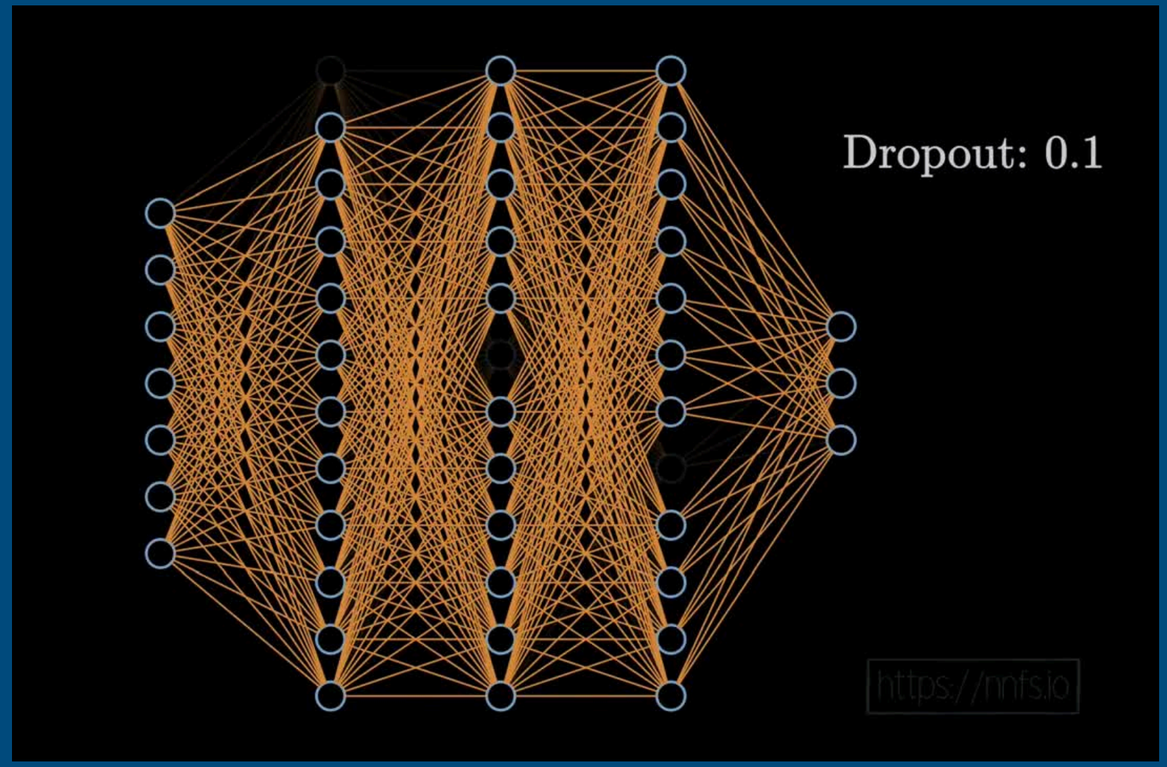


The idea of dropout came to stop neurons from ***relying too much on each other*** (co-adaptation).By randomly dropping neurons, the model learns independent, stronger features, which ***reduces overfitting***.

WHY WE NEED DROPOUT?



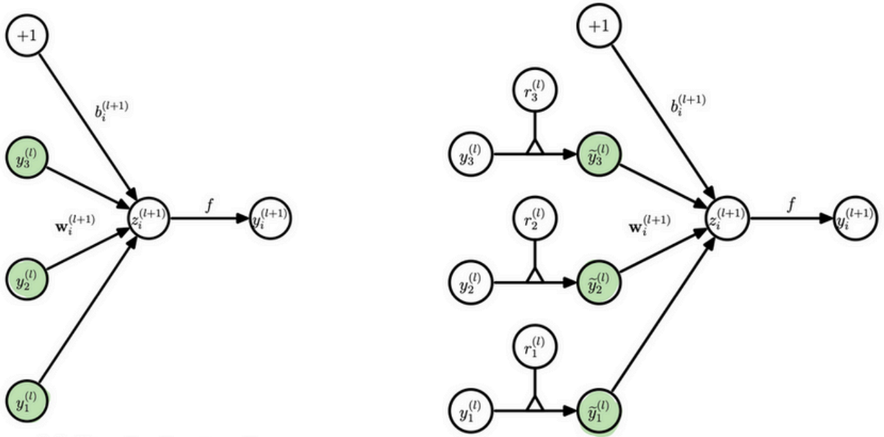




Each training step uses a new ***thinned network***

n units can be seen as 2^n possible smaller networks

MODEL DESCRIPTION



(a) Standard network (b) Dropout network

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

where f is any activation function, for example, $f(x) = 1/(1+\exp(-x))$.
With dropout, the feed-forward operation becomes (Figure 3b)

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

1933

l : Index of hidden layers

L: Total no of hidden layers

W(l) : Weight matrix at layer l

b(l) :Bias vector at layer l

z(l) : Input vector to layer l

y(l) : Output vector from layer l

r(l) : For any layer is a vector of independent bernaulli random variable

Comparision of dropouts

with Standard regularization methods

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 combined with max norm regularization gives the best result

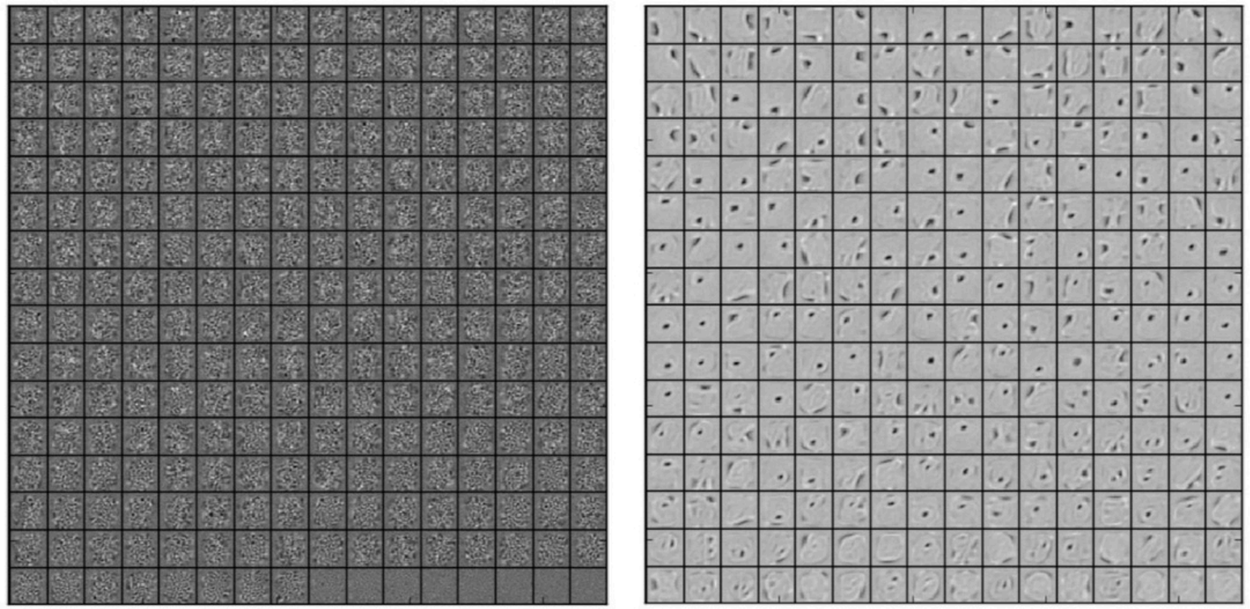
with Bayesian neural network

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.

we found that Bayesian neural nets perform better than dropout. However, we see that dropout improves significantly upon the performance of standard neural networks and outperforms all other methods.

Effects of Dropouts on Features



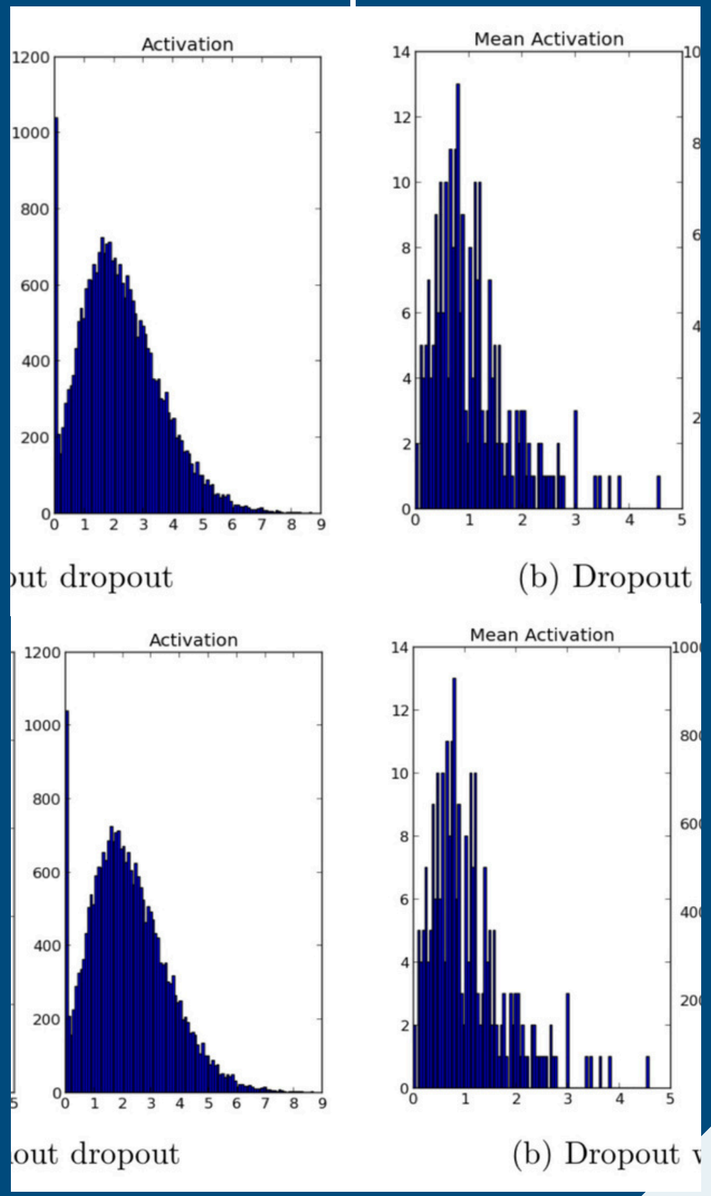
(a) Without dropout

(b) Dropout with $p = 0.5$.

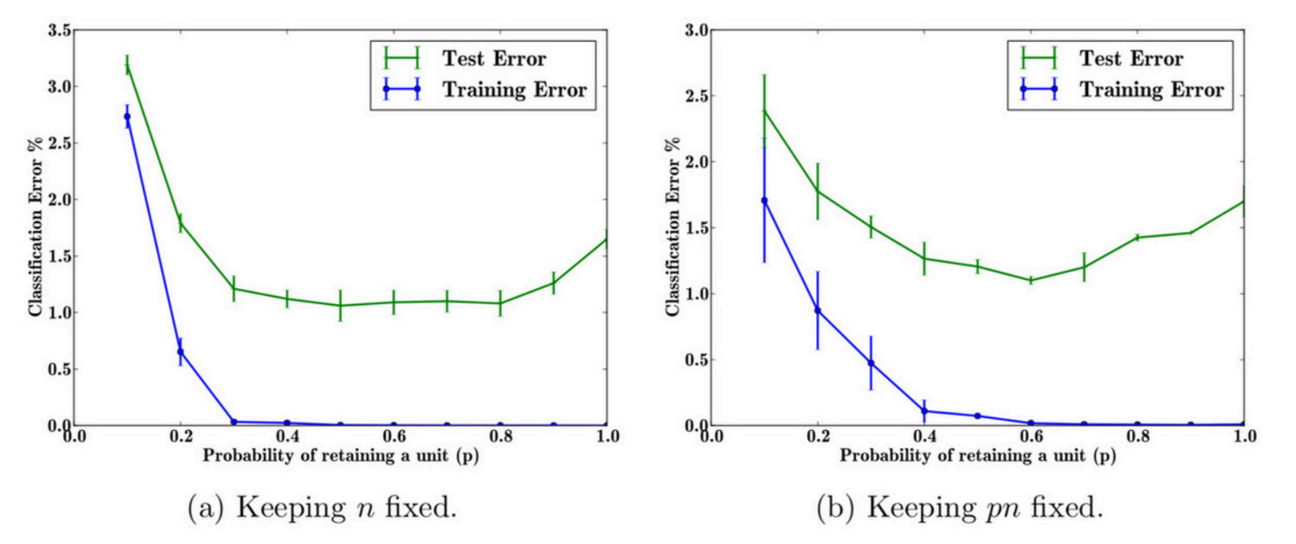
Without dropout, neurons co-adapt and rely on each other's corrections. Dropout forces each neuron to work independently → learns stronger, more meaningful features.

Example: On MNIST, features with dropout look like edges/strokes, while without dropout they look less interpretable.

Effects of Dropouts on Sparsity



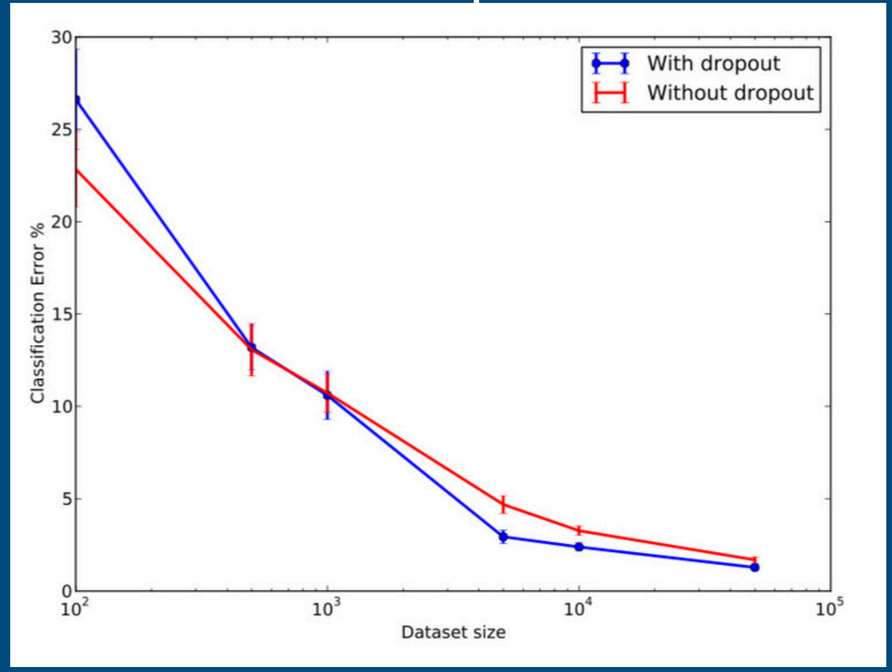
Effects of Dropout Rate



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

- Too low p → very few active units → underfitting.
- Too high p (close to 1) → very little dropout→ overfitting risk.
- Best results usually when p ≈ 0.5 for hidden layers and 0.8 for input layer.

Effects of Dataset Size



- On very small datasets, dropout doesn't help much (model can still overfit).
- With very large datasets, the benefit decreases since overfitting is less of a problem.
- There's a "***sweet spot***" where dropout works best → medium-sized data with risk of overfitting.

Exerimental results:

(a)MNIST

0.1.1 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

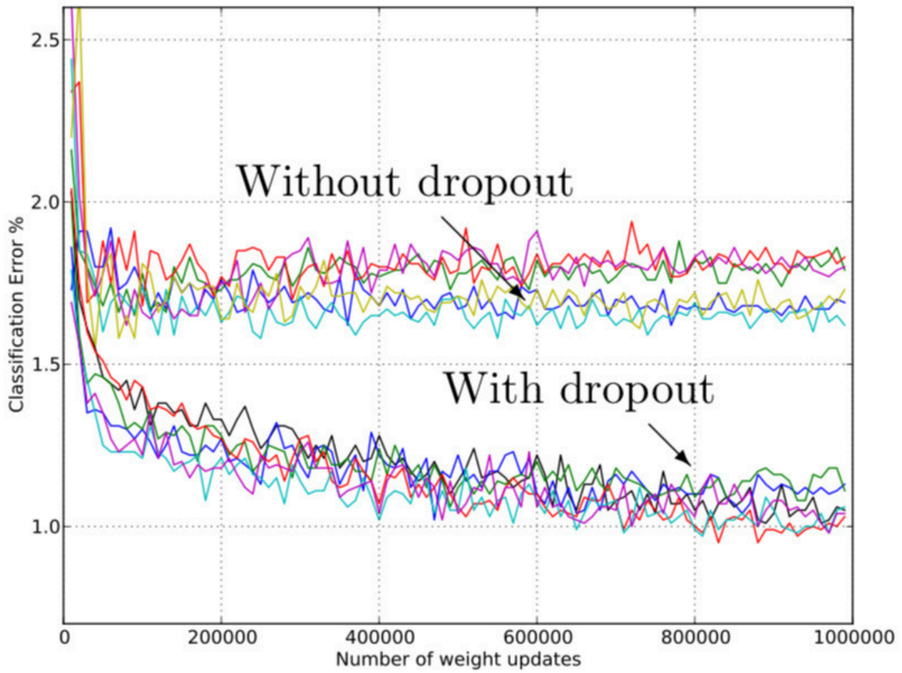


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

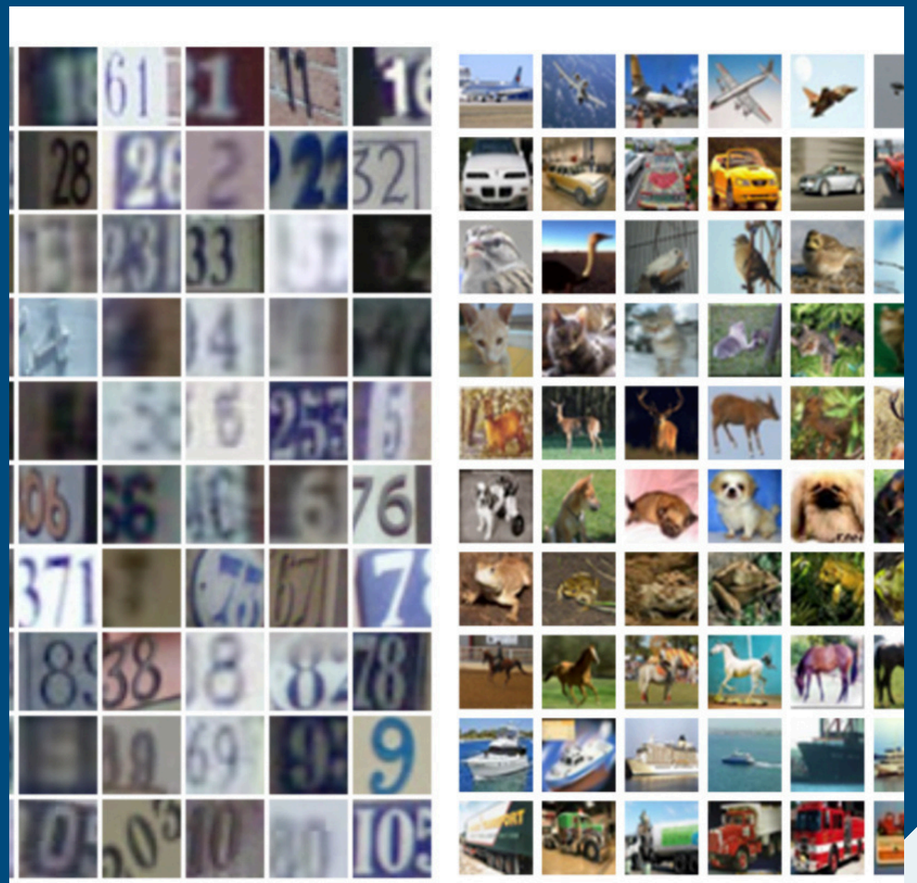
Dropout gives a huge improvement across all architectures, without using hyperparameters that were tuned specifically for each architecture.

Exerimental results:

(b)SVHN (street view house number)

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.



ıse Numbers (SVHN) (b) CIFA

Exerimental results:

(c)CIFAR-10 and CIFAR-100

ouse Numbers (SVHN) (b) CIFAR

Regularization Methods that use dropout achieve state-of-the-art results on SVHN, ImageNet, CIFAR-100 and MNIST. Dropout considerably improved the performance of standard neural nets on other data sets as well.

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.

Conclusion:

- *Dropout reduces overfitting by breaking neuron co-adaptations.*
- *Makes networks more robust and generalizable to unseen data.*
- *Achieved state-of-the-art results on MNIST, CIFAR-100, SVHN, and ImageNet.*
- *Limitation: Increases training time . A dropout neural network takes 2-3 times longer to train than standard neural network*
- *Trade-off: less overfitting vs. longer training time.*

"Speeding up dropout is an interesting direction for future work."

BIBLOGRAPHY

M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. In Proceedings of the 29th International Conference on Machine Learning, pages 767–774. ACM, 2012.

G. E. Dahl, M. Ranzato, A. Mohamed, and G. E. Hinton. Phone recognition with the mean-covariance restricted Boltzmann machine. In Advances in Neural Information Processing Systems 23, pages 469–477, 2010.

O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. Machine Learning, 81(2):149–178, 2010.

A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. In
Proceedings of the 23rd International Conference on Machine Learning, pages 353–360. ACM, 2006.

THANK YOU