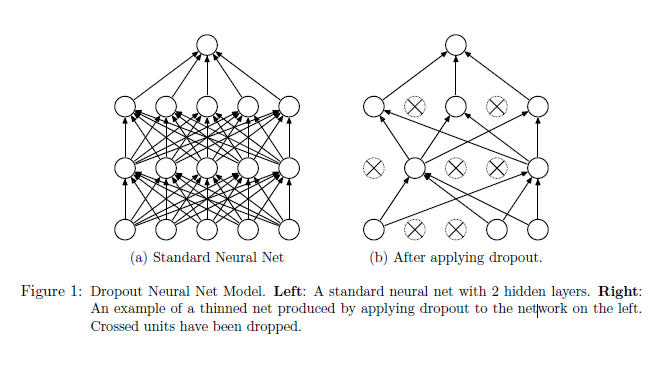
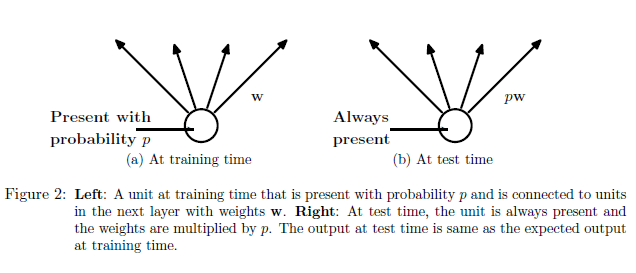
Dropout: A Simple Way to Prevent Neural Networks from Overfitting

1. Introduction

The term “dropout” refers to dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections, as shown in Figure 1.





2. Motivation

A closely related, but slightly different motivation for dropout comes from thinking

about successful conspiracies.

3. Related Work

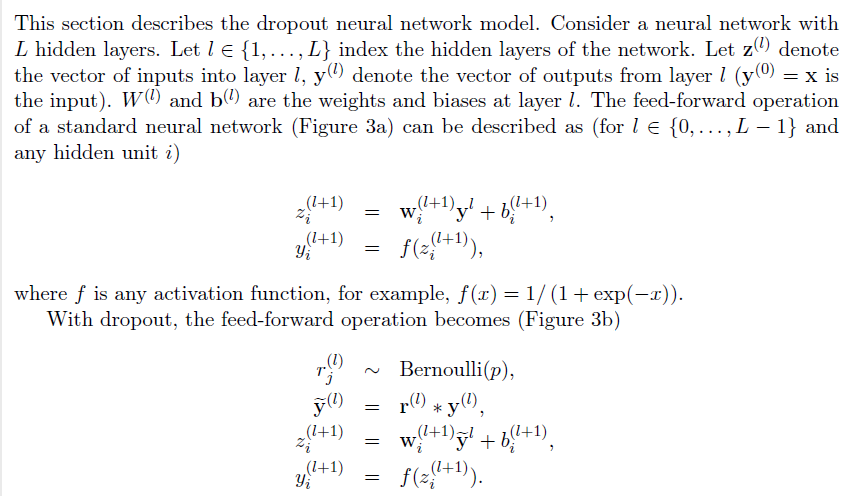
Dropout can be interpreted as a way of regularizing a neural network by adding noise to

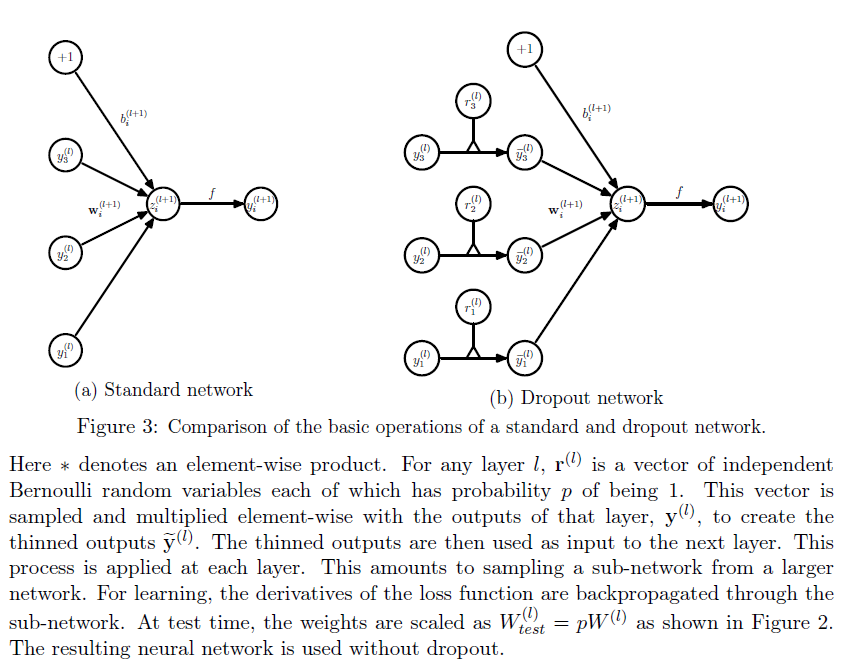
its hidden units.

In dropout, we minimize the loss function stochastically under a noise distribution.

This can be seen as minimizing an expected loss function.

4. Model Description





5. Learning Dropout Nets

5.1 Backpropagation

One particular form of regularization was found to be especially useful for dropout— constraining the norm of the incoming weight vector at each hidden unit to be upper bounded by a fixed constant c.

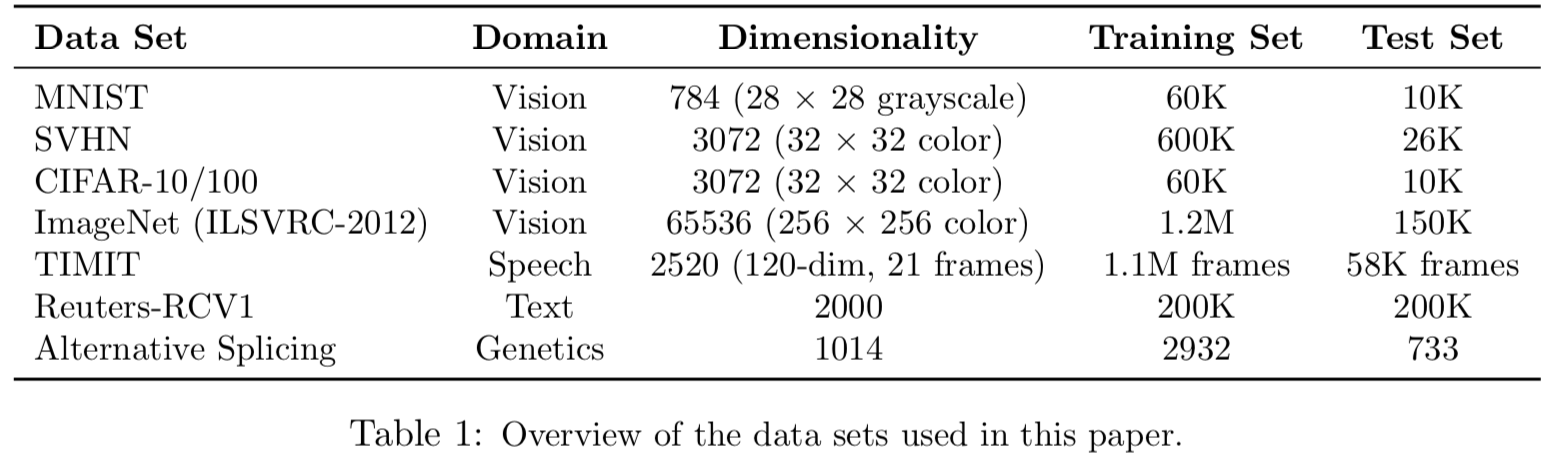
This is also called max-norm regularization since it implies that the maximum value that the norm of any weight can take is c.

The constant c is a tunable hyperparameter, which is determined using a validation set.

A possible justification is that constraining weight vectors to lie inside a ball of fixed radius makes it possible to use a huge learning rate without the possibility of weights blowing up

5.2 Unsupervised Pretraining

6. Experimental Results



6.1 Results on Image Data Sets

6.2 Results on TIMIT

6.3 Results on a Text Data Set

6.4 Comparison with Bayesian Neural Networks

6.5 Comparison with Standard Regularizers

7. Salient Features

7.1 Effect on Features

7.2 Effect on Sparsity

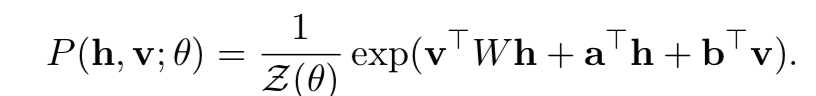
7.3 Effect of Dropout Rate

7.4 Effect of Data Set Size

7.5 Monte-Carlo Model Averaging vs. Weight Scaling

8. Dropout Restricted Boltzmann Machines

8.1 Model Description



Where θ = {W, a, b} represents the model parameters and Z is the partition function.

8.2 Learning Dropout RBMs

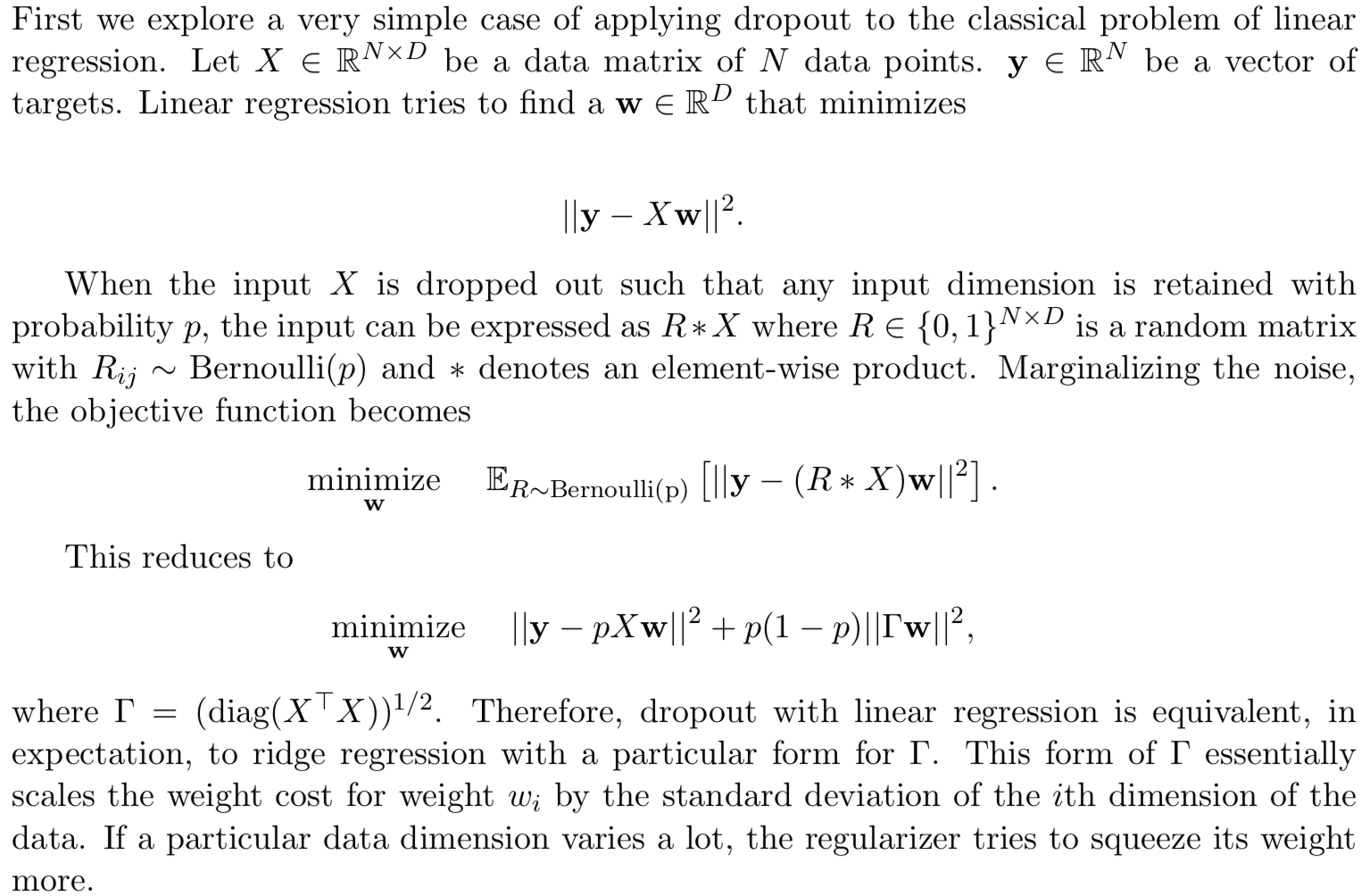
8.3 Effect on Features

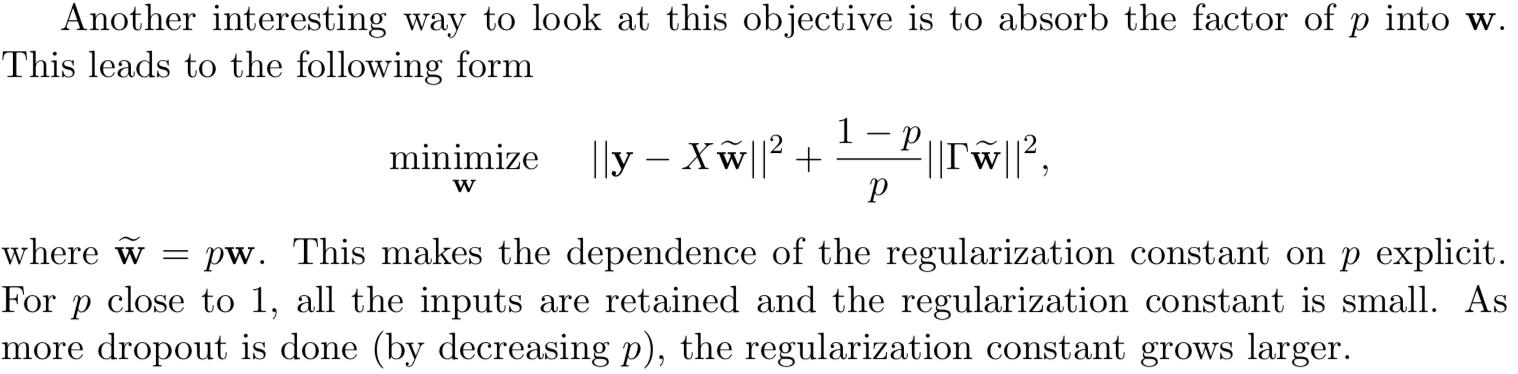
8.4 Effect on Sparsity

9. Marginalizing Dropout

Dropout can be seen as a way of adding noise to the states of hidden units in a neural network.

9.1 Linear Regression





9.2 Logistic Regression and Deep Networks

10. Multiplicative Gaussian Noise

Dropout involves multiplying hidden activations by Bernoulli distributed random variables which take the value 1 with probability p and 0 otherwise. This idea can be generalized by multiplying the activations with random variables drawn from other distributions. We recently discovered that multiplying by a random variable drawn from N (1, 1) works just as well, or perhaps better than using Bernoulli noise. This new form of dropout amounts to adding a Gaussian distributed random variable with zero mean and standard deviation equal to the activation of the unit.

Therefore, dropout can be seen as multiplying hi by a Bernoulli random variable rb that takes the value 1/p with probability p and 0 otherwise.

11. Conclusion

Dropout is a technique for improving neural networks by reducing overfitting.

One of the drawbacks of dropout is that it increases training time.