COMP9444
Neural Networks and Deep Learning

2b. Generalization

Textbook, Sections 5.1-5.2, 7.11-12

Outline

- Supervised Learning (5.1)
- Ockham's Razor (5.2)
- Training, Validation and Test Error
- Dropout (7.11-12)

Types of Learning (5.1)

- Supervised Learning
  - agent is presented with examples of inputs and their target outputs
- Reinforcement Learning
  - agent is not presented with target outputs, but is given a reward signal, which it aims to maximize
- Unsupervised Learning
  - agent is only presented with the inputs themselves, and aims to find structure in these inputs

Supervised Learning

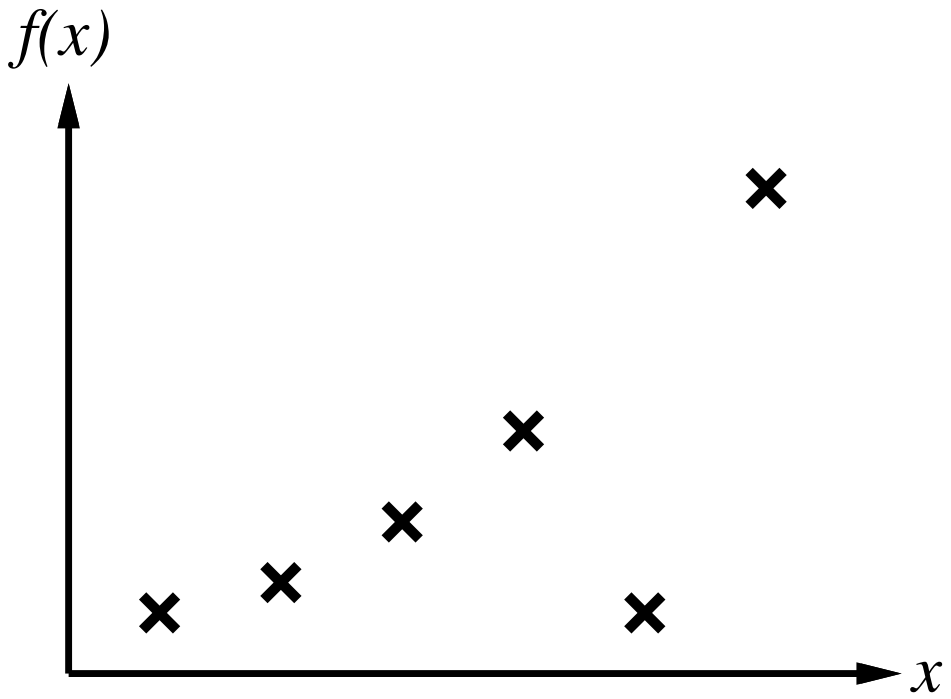
- we have a training set and a test set, each consisting of a set of items; for each item, a number of input attributes and a target value are specified.
- the aim is to predict the target value, based on the input attributes.
- agent is presented with the input and target output for each item in the training set; it must then predict the output for each item in the test set
- various learning paradigms are available:
  - Neural Network
  - Decision Tree
  - Support Vector Machine, etc.

Supervised Learning – Issues

- framework (decision tree, neural network, SVM, etc.)
- representation (of inputs and outputs)
- pre-processing / post-processing
- training method (perceptron learning, backpropagation, etc.)
- generalization (avoid over-fitting)
- evaluation (separate training and testing sets)

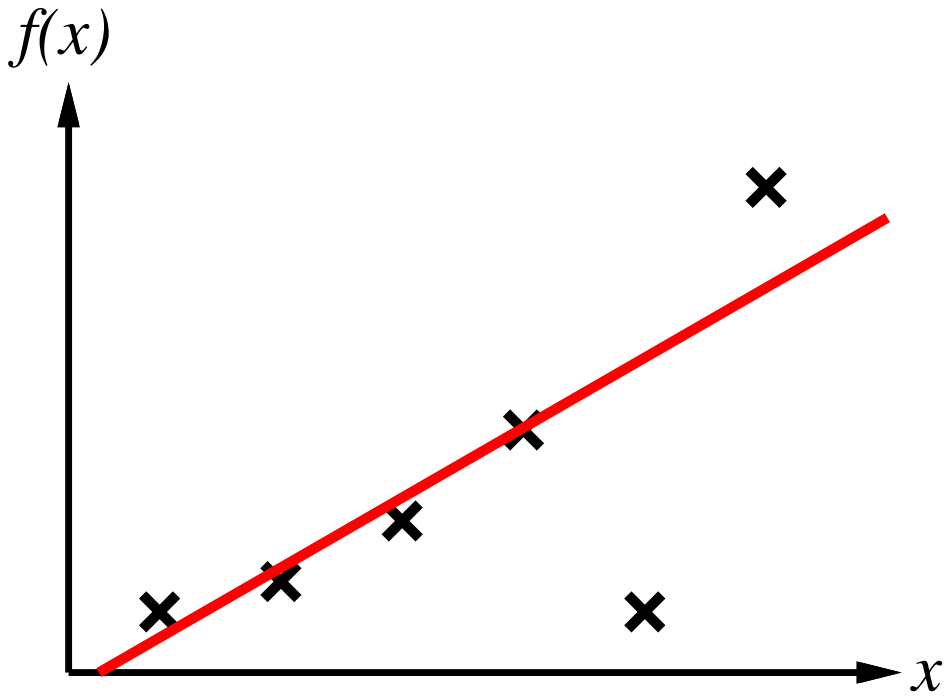
Curve Fitting

Which curve gives the "best fit" to these data?



Curve Fitting

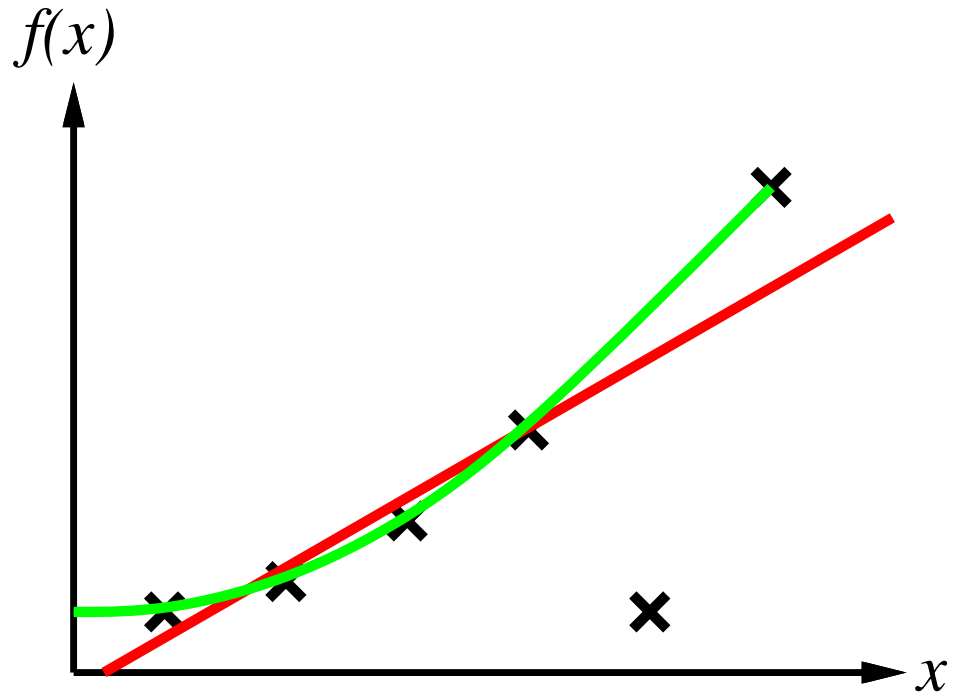
Which curve gives the "best fit" to these data?



straight line?

Curve Fitting

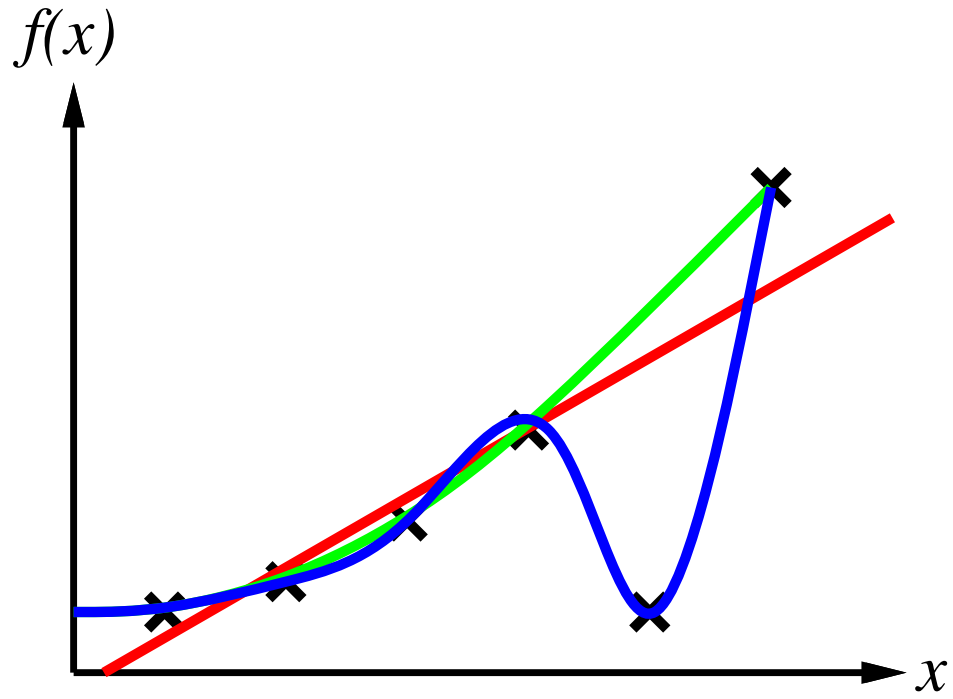
Which curve gives the "best fit" to these data?



parabola?

Curve Fitting

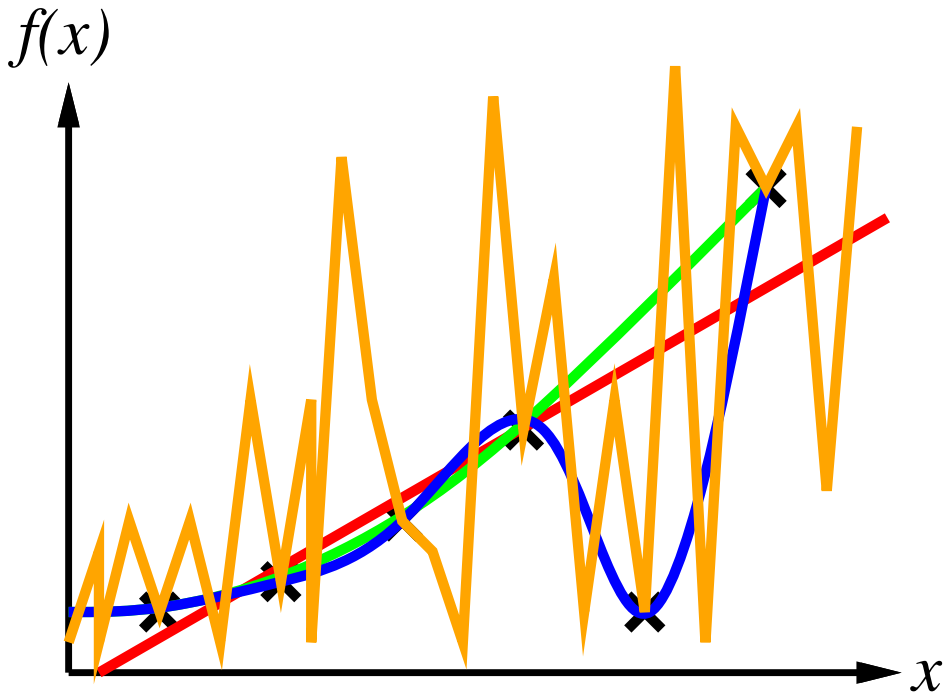
Which curve gives the "best fit" to these data?



4th order polynomial?

Curve Fitting

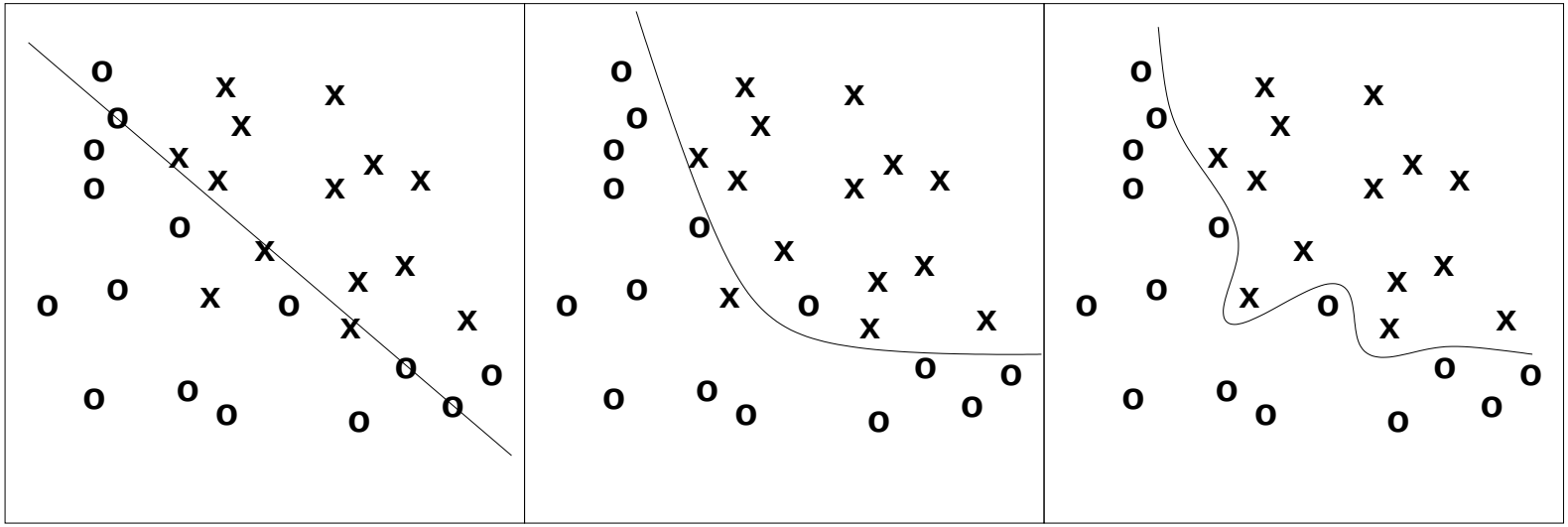
Which curve gives the "best fit" to these data?



Something else?

Ockham's Razor (5.2)

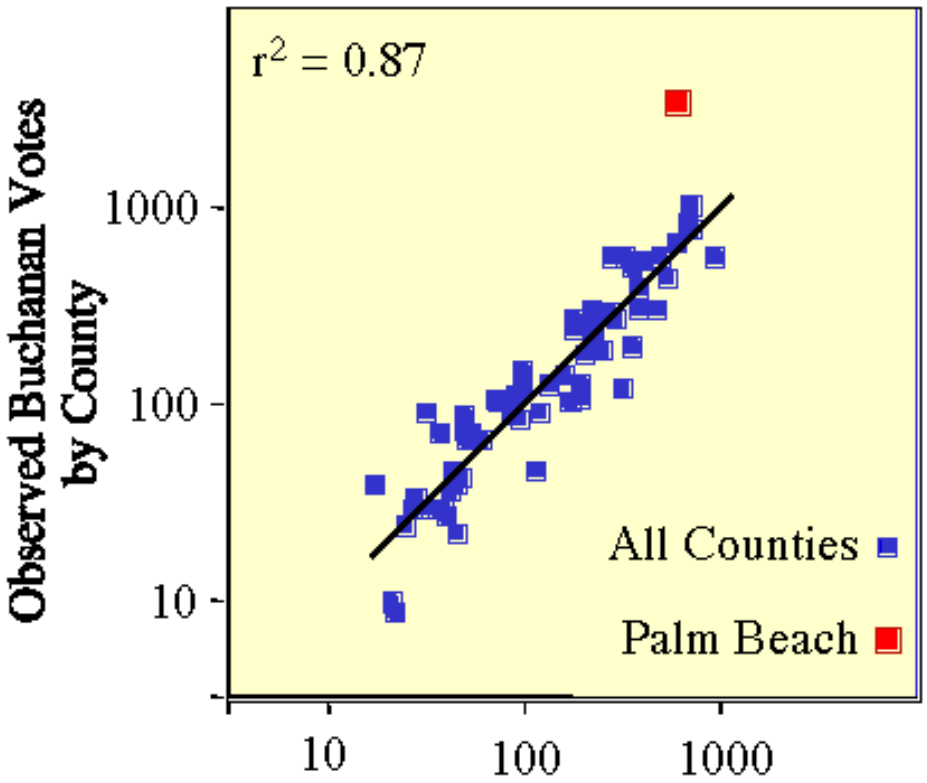
"The most likely hypothesis is the simplest one consistent with the data."



inadequate good compromise over-fitting

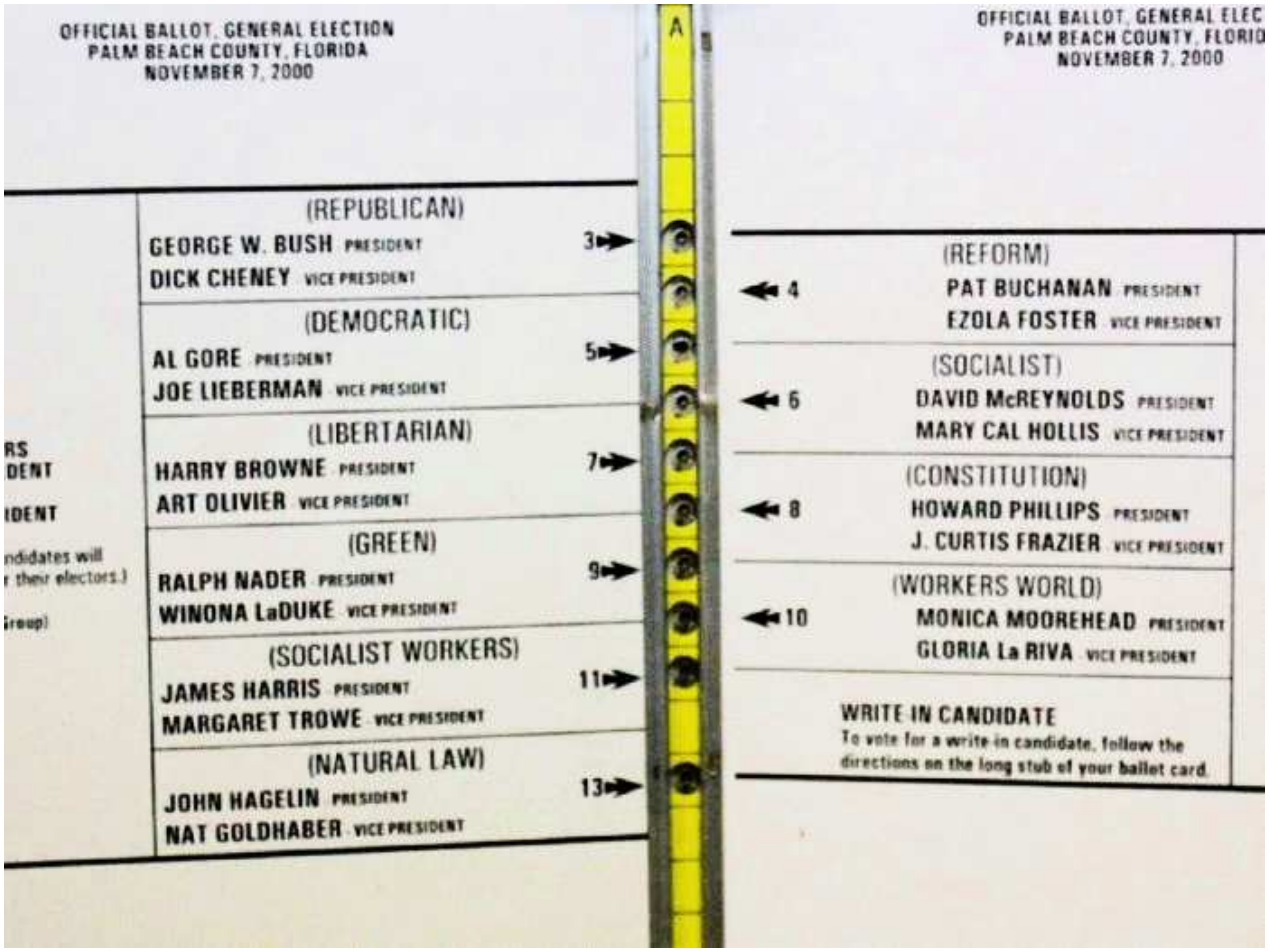
Since there can be noise in the measurements, in practice need to make a tradeoff between simplicity of the hypothesis and how well it fits the data.

Outliers



[faculty.washington.edu/mtbrett]

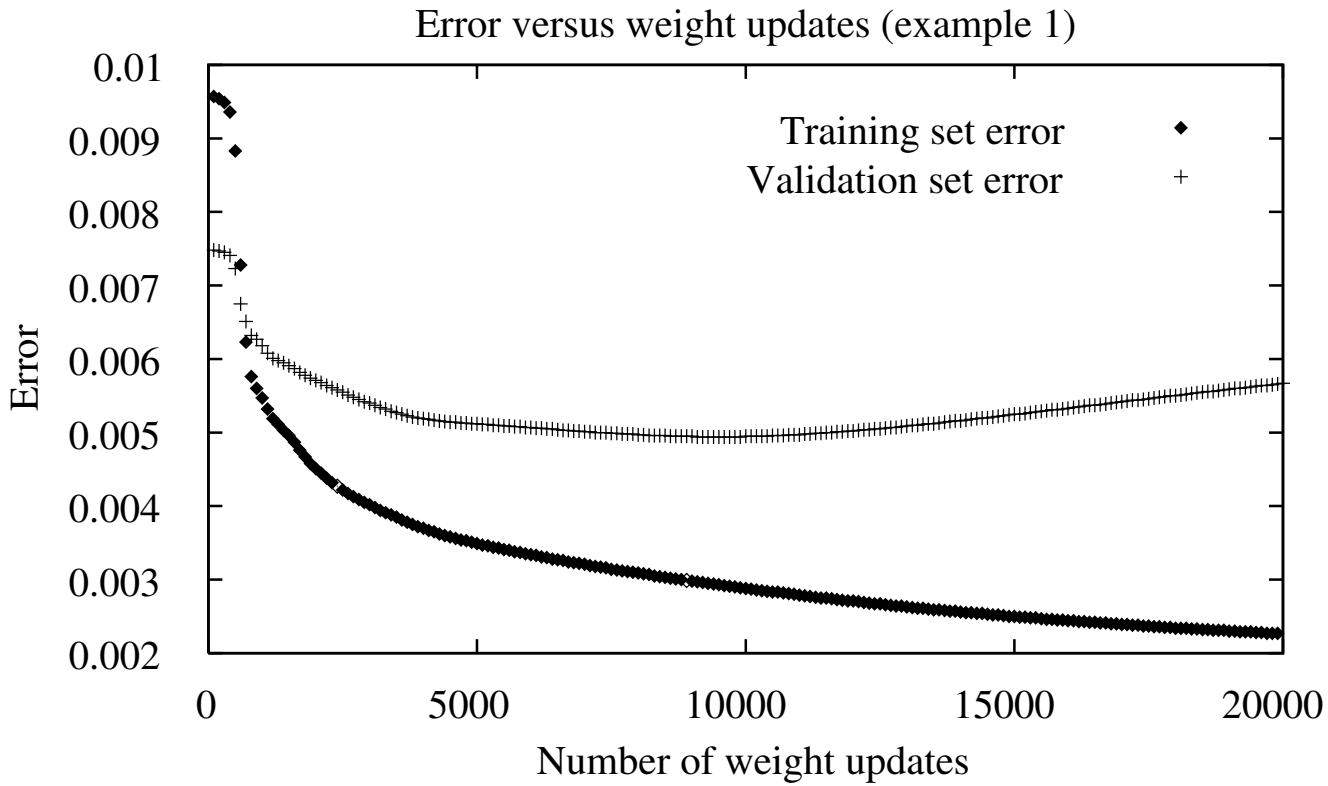
Butterfly Ballot



Ways to Avoid Overfitting in Neural Netorks

- Limit the number of hidden nodes or connections
- Limit the number of training epochs (weight updates)
- Dropout
- Weight Decay (Week 3)
- Data Augmentation (Week 4)
- Regularization

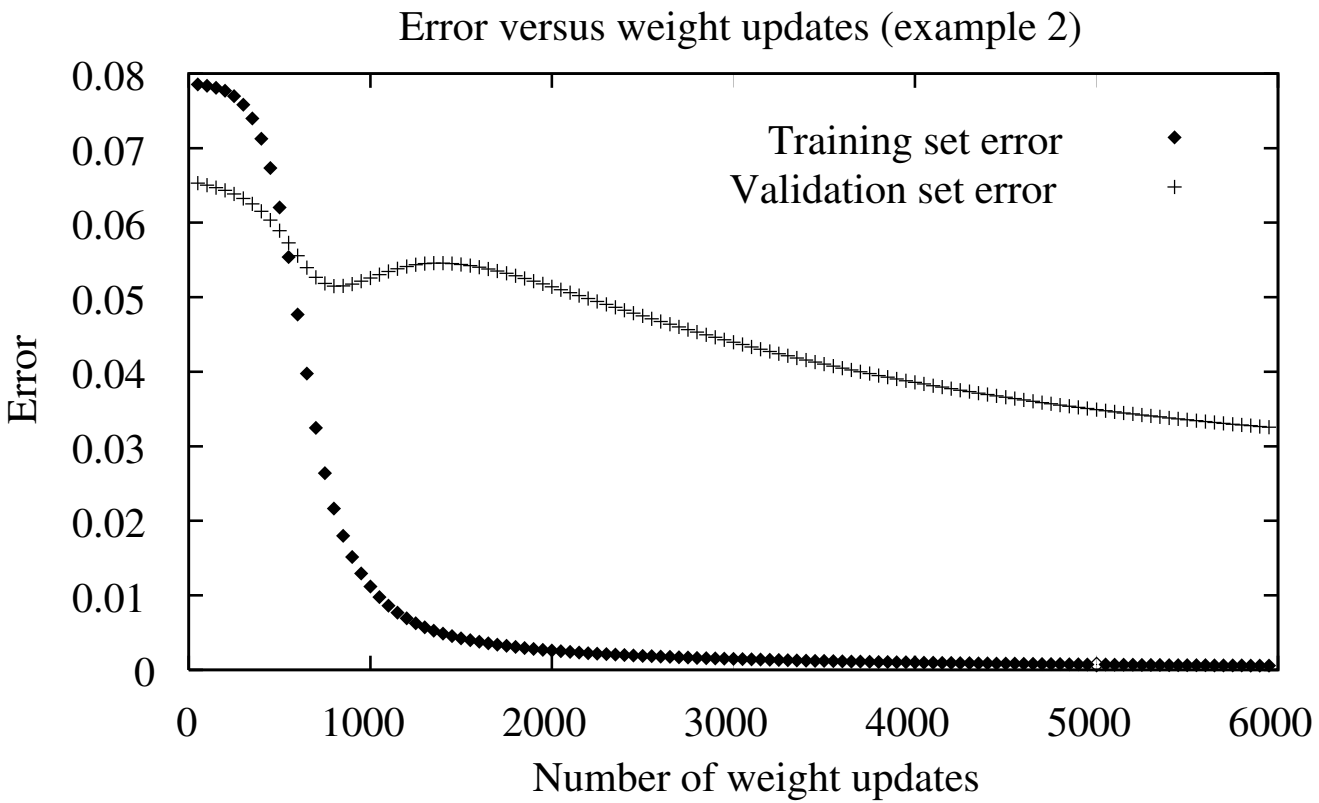
Training, Validation and Test Error



x-axis could be number of weight updates, hidden nodes, dropout, etc.

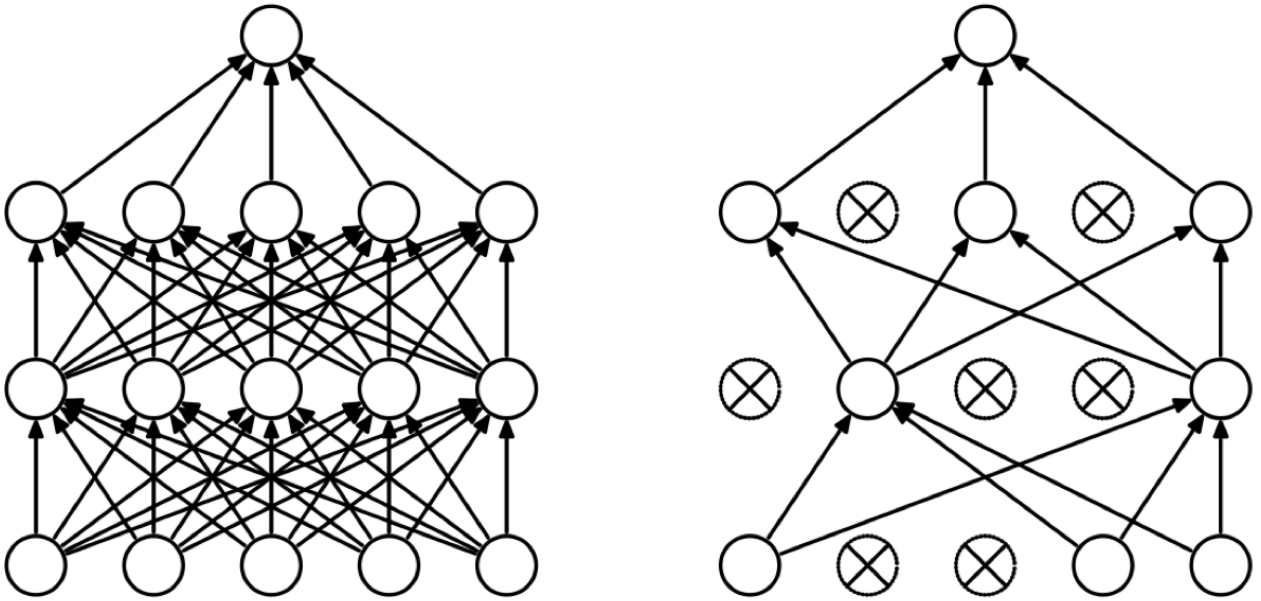
Training error decreases but validation and test error flatter or increase.

Be Careful about Early Stopping



We may need to continue training in order to be sure what is going on.

Dropout (7.12)



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

For each minibatch, randomly choose a subset of nodes to not be used.
Each node is chosen with some fixed probability (usually, one half).

Dropout (7.12)

When training is finished and the network is deployed, all nodes are used, but their activations are multiplied by the same probability that was used in the dropout.

Thus, the activation received by each unit is the average value of what it would have received during training.

Dropout forces the network to achieve redundancy because it must deal with situations where some features are missing.

Another way to view dropout is that it implicitly (and efficiently) simulates an ensemble of different architectures.

Ensembling (7.11)

Ensembling is a method where a number of different classifiers are trained on the same task, and the final class is decided by "voting" among them.

In order to benefit from ensembling, we need to have diversity in the different classifiers.

For example, we could train three neural networks with different architectures, three Support Vector Machines with different dimensions and kernels, as well as two other classifiers, and ensemble all of them to produce a final result.
(Kaggle Competition entries are often done in this way).

Dropout as an Implicit Ensemble

In the case of dropout, the same data are used each time but a different architecture is created by removing the nodes that are dropped.

The trick of multiplying the output of each node by the probability of dropout implicitly averages the output over all of these different models.