

---

## Week 7 Homework

---

Samuel Cuthbertson

SAMUEL.CUTHBERTSON@COLORADO.EDU

### 1. Answers to Part A

#### 1.1. What is the most common way of choosing the number of nodes in the hidden layer?

Trial and error. There are rules of thumb, like that the number of nodes *should* lie between the number of input and the number of output nodes, but trial and error is how we find that mystery number.

#### 1.2. How is overfitting typically handled in neural networks, and how does this contrast with approaches typically taken with linear classifiers?

In neural networks, overfitting is typically combated with early stopping. This is in direct contrast to regularization in linear classifiers, but has the same effect: They both keep weights from growing too large. Early stopping just does it by stopping training, while regularization directly penalizes large weights.

#### 1.3. Describe the difference between backpropagation and forward propagation.

Back propagation is the process of propagating the error from the output layer of nodes back through all previous layers and applying the loss function to each layer. Forward propagation is actually running the network with a given input, and finding the error which you apply through back propagation. In other words, forward propagation is testing and back propagation is training/learning.

#### 1.4. What is the vanishing gradient problem, and when is it actually a problem?

The vanishing gradient problem applies to how, in a large multi-layered neural network, the earlier hidden layers learn much slower than the later layers. This is only ever actually a problem when using a function such as sigmoid or tanh which "squashes" a large portion of inputs into a similar output.

#### 1.5. Name three disadvantages of using neural networks vs. using a simpler classifier.

Overfitting, difficulty retraining, and an unclear VC-Dimension.

### 2. Analysis for Part B

For both datasets, the parameter I "tweaked" was the number of hidden layers. I compared against a logistic loss function, and used 10-fold cross validation. All the data is in the table below.

Number of Hidden Layers	Accuracy on Iris	Accuracy on Sports
1	0.24511715827975422	0.9990234375
3	0.9951170688727871	0.9990234375
5	0.9677733221324161	0.9990234375
7	0.9677733221324161	0.9990234375
25	0.8662108342396095	0.9990234375

Number of Hidden Layers	Logistic on Iris	Logistic on Sports
1	0.874023333308287	0.9990234375
3	0.9755858212010935	0.9990234375
5	0.9667967597488314	0.9990234375
7	0.874023333308287	0.9990234375
25	0.934570201090537	0.9990234375

For the Iris data, it's easy to see how the neural network begins to overfit at greater than 3 layers. Likewise, it's apparent that logistic is struggling with this little data and has poor precision. Even with 10-fold cross validation, it fails to find a consistent accuracy.

For sports, I'm not at all sure what went wrong. Either there is such a clear and distinguishable boundary that even a neural network with 1 layer can find it as well as a neural network with 25 hidden layers, or something is wrong with my testing script. However, after spending the last couple hours debugging it, I still can't find any fault. I've attached my testing script in the directory `./vw` for checking my work.