Neural Network Models of Language (II)

1.0 Multi-layer Perceptrons

1.1 Nonlinear Networks

Linear networks are limited in that they can represent some types of associations, but not others. One way to increase the power of the network is to add a nonlinearity – usually some type of threshold function – to transform the output. Among other things, this allows the creation of multi-layer networks. These are called multi-layer perceptrons (MLPs).

The network in the figure above follows the equations: \( h = f(W_1v) \), and \( u = f(W_2h) \). Or, we could write this as one longer equation: \( u = f(W_2f(W_1v)) \).

The transfer function allows the network to make decisions, by replacing graded input with all-or-none activation.

### Step Function

\[
    f(x) = \begin{cases} 
        1, & \text{if } x > \theta \\
        0, & \text{otherwise} 
    \end{cases}
\]

### Sigmoid Function

\[
    f(x) = \frac{1}{1 + e^{-(x-\theta)/T}}
\]
The result is a network that is able to learn internal representations.

1.2 Backpropagation Learning

Learning in nonlinear networks is similar in principle to learning in linear networks. The error function is defined in just the same way. And changes to the weights are negatively proportional to the derivative of the error function with respect to the weights.

There are two important differences, however. First, because the output involves a nonlinear function, we need to know the derivative of this function. This means that functions such as the step function are not useful in learning situations, because they are not differentiable everywhere. Sigmoid functions are commonly used because they are smooth.

Second, there is more than one set of weights because there is more than one layer. Thus, a separate derivative must be calculated for each set of weights.

2.0 Neural Networks for Language

2.1 Rumelhart & Norman’s Model of Past Tense Formation
2.2 A Preliminary Model of Grammaticality Judgements
2.3 Jordan’s Phonological Model
2.4 Elman’s Grammar Model