Neural Networks and Classification

Consider a $K$-class classification problem. In place of a single $y$ taking values in $\{1, 2, \ldots, K\}$ one might define $K$ indicator variables

$$y_k = I[y = k]$$

and consider a vector neural network predictor $\hat{f} : \mathbb{R}^p \rightarrow [0, 1]^k$. This can be accomplished by using in place of identity functions $g_k(u) = u$ mapping linear combinations of the "last" layer of hidden variables to (not one, but $K$) outputs $\hat{y}_k$, sigmoidal (logistic) functions

$$g_k(u) = \frac{1}{1 + \exp(-u)}$$

Then (exactly as in Module 16) least squares fitting to the $K$ (now 0-1) response variables $y_k$ produces a very flexible method of simultaneously modeling the vector of $k$ indicators as a function of the input $x \in \mathbb{R}^p$. Now the vector of predictions

$$\begin{pmatrix}
\hat{y}_1 \\
\hat{y}_2 \\
\vdots \\
\hat{y}_K
\end{pmatrix} =
\begin{pmatrix}
\hat{f}_1(x) \\
\hat{f}_2(x) \\
\vdots \\
\hat{f}_K(x)
\end{pmatrix}$$

has entries in $(0, 1)$ but is typically not a probability vector since its entries usually won't sum to 1. But that is completely inconsequential as regards making a plausible classifier. Corresponding to the fitted neural network is the classifier

$$\hat{f}(x) = \arg \max_k \hat{f}_k(x)$$