Consider the function

$$f(x_1, x_2, x_3, x_4) = \frac{x_1}{1+x_1} + \frac{x_2}{1+x_2} + \frac{x_3}{1+x_3} + \frac{x_4}{1+x_4}$$

defined on $[0, 2]$. Assume that the input-output data for the function $f(\boldsymbol{x})$ are randomly generated on $[0, 2]$ with a uniformly distributed noise $\mathcal{U}(a, b)$, where $\mathcal{U}(a, b)$ represents the uniformly distributed variable on the interval $[a, b]$. That is, for each input $\boldsymbol{x}$, the output $y = f(x_1, x_2, x_3, x_4) + \mathcal{U}(a, b)$.

We produce 2000 training input-output data $\{(\boldsymbol{x}_i, y_i) | i = 1, 2, \cdots, 2000\}$ with noise $\mathcal{U}(a, b)$ for the function $f(\boldsymbol{x})$. The backpropagation learning algorithm may obtain an NN (4 input neurons, 6 hidden neurons, 1 output neuron) to approximate the function $f(\boldsymbol{x})$ according to the noise data.

Let $F(\boldsymbol{x}, \boldsymbol{w}^*)$ be the output of mapping implemented by the NN. We generate 1000 test noise data $\{(\boldsymbol{x}'_i, y'_i) | i = 1, 2, \cdots, 1000\}$, then we have the errors shown in Table 1.

Table 1: Errors

Noise	$\frac{1}{2}\sum_{i=1}^{1000} \lvert F(\boldsymbol{x}'_i, \boldsymbol{w}^*) - f(\boldsymbol{x}'_i)\rvert^2$	$\frac{1}{2}\sum_{i=1}^{1000} \lvert y'_i - f(\boldsymbol{x}'_i)\rvert^2$
$\mathcal{U}(-0.05, 0.05)$	0.362	0.389
$\mathcal{U}(-0.10, 0.10)$	1.333	1.643
$\mathcal{U}(-0.20, 0.20)$	4.208	6.226
$\mathcal{U}(-0.30, 0.30)$	7.306	14.01
$\mathcal{U}(-0.40, 0.40)$	14.74	24.90

Note that the errors in the first column are less than that in the second column. This means that the trained NN can compensate for the error of the noise training data.