

The Bias-Variance Tradeoff

Given:

- the true function we want to approximate

$$f = f(\mathbf{x})$$

- the data set for training

$$D = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\} \text{ where } t = f + \varepsilon \text{ and } E\{\varepsilon\} = 0$$

- given D , we train an arbitrary neural network to approximate the function f by

$$y = g(\mathbf{x}, \mathbf{w})$$

The mean-squared error of this networks is:

$$MSE = \frac{1}{N} \sum_{i=1}^N (t_i - y_i)^2$$

To assess the effectiveness of the network, we want to know the expectation of the MSE if we test the network on arbitrarily many test points drawn from the unknown function.

$$E\{MSE\} = E\left\{\frac{1}{N} \sum_{i=1}^N (t_i - y_i)^2\right\} = \frac{1}{N} \sum_{i=1}^N E\{(t_i - y_i)^2\}$$

Let's investigate the expectation inside the sum, with a little "augmentation trick":

$$\begin{aligned} E\{(t_i - y_i)^2\} &= E\{(t_i - f_i + f_i - y_i)^2\} \\ &= E\{(t_i - f_i)^2\} + E\{(f_i - y_i)^2\} + 2E\{(f_i - y_i)(t_i - f_i)\} \\ &= E\{\varepsilon^2\} + E\{(f_i - y_i)^2\} + 2(E\{f_i t_i\} - E\{f_i^2\} - E\{y_i t_i\} + E\{y_i f_i\}) \end{aligned}$$

Note : $E\{f_i t_i\} = f_i^2$ since f is deterministic and $E\{t_i\} = f_i$

: $E\{f_i^2\} = f_i^2$ since f is deterministic

: $E\{y_i t_i\} = E\{y_i(f_i + \varepsilon)\} = E\{y_i f_i + y_i \varepsilon\} = E\{y_i f_i\} + 0$

: (the last term is zero because the noise in the infinite test set over which

we take the expectation is probabilistically independent of the NN

prediction). Thus the last term in the expectation above cancels to zero.

$$E\{(t_i - y_i)^2\} = E\{\varepsilon^2\} + E\{(f_i - y_i)^2\}$$

Thus the MSE can be decomposed in expectation into the variance of the noise and the MSE between the true function and the predicted values. This term can be further composed with the same augmentation trick as above.

$$\begin{aligned}
 E\{(f_i - y_i)^2\} &= E\{(f_i - E\{y_i\} + E\{y_i\} - y_i)^2\} \\
 &= E\{(f_i - E\{y_i\})^2\} + E\{(E\{y_i\} - y_i)^2\} + 2E\{(E\{y_i\} - y_i)(f_i - E\{y_i\})\} \\
 &= bias^2 + Var\{y_i\} + 2(E\{f_i E\{y_i\}\} - E\{E\{y_i\}^2\} - E\{y_i f_i\} + E\{y_i E\{y_i\}\})
 \end{aligned}$$

Note : $E\{f_i E\{y_i\}\} = f_i E\{y_i\}$ since f is deterministic and $E\{E\{z\}\} = z$

$$: E\{E\{y_i\}^2\} = E\{y_i\}^2 \text{ since } E\{E\{z\}\} = z$$

$$: E\{y_i f_i\} = f_i E\{y_i\}$$

$$: E\{y_i E\{y_i\}\} = E\{y_i\}^2$$

: Thus the last term in the expectation above cancels to zero.

$$E\{(f_i - y_i)^2\} = bias^2 + Var\{y_i\}$$

Thus the decomposition of the MSE in expectation becomes:

$$E\{(t_i - y_i)^2\} = Var\{noise\} + bias^2 + Var\{y_i\}$$

Note that the variance of the noise can not be minimized; it is independent of the neural network. Thus in order to minimize the MSE, we need to minimize both the bias and the variance. However, this is not trivial to do this. For instance, just neglecting the input data and predicting the output somehow (e.g., just a constant), would definitely minimize the variance of our predictions: they would be always the same, thus the variance would be zero—but the bias of our estimate (i.e., the amount we are off the real function) would be tremendously large. On the other hand, the neural network could perfectly interpolate the training data, i.e., it predict $y=t$ for every data point. This will make the bias term vanish entirely, since the $E(y)=f$ (insert this above into the squared bias term to verify this), but the variance term will become equal to the variance of the noise, which may be significant (see also Bishop Chapter 9 and the Geman et al. Paper). In general, finding an optimal bias-variance tradeoff is hard, but acceptable solutions can be found, e.g., by means of cross validation or regularization.