T-61.5030 Advanced course in neural computing

Solutions for exercise 3

1. Let us denote $\mathbf{d} = [d_1\ d_2 \dots d_N]^T$ and $\mathbf{y} = [y_1\ y_2 \dots y_N]^T$, where $y_i = \sum w_k F_k(\mathbf{x}_i)$, and $\mathbf{w} = [w_1\ w_2 \dots w_K]^T$. If $\mathbf{F}$ is a matrix whose element on the ith row and kth column contains the value $F_k(\mathbf{x}_i)$, we can write $\mathbf{y} = \mathbf{F}\mathbf{w}$. The sum of squared errors can then be written compactly as

$$\mathcal{E} = (\mathbf{d}-\mathbf{y})^T(\mathbf{d}-\mathbf{y}) = (\mathbf{d}-\mathbf{F}\mathbf{w})^T(\mathbf{d}-\mathbf{F}\mathbf{w}) = \mathbf{d}^T\mathbf{d} - 2\mathbf{w}^T\mathbf{F}^T\mathbf{d} + \mathbf{w}^T\mathbf{F}^T\mathbf{F}\mathbf{w}\,.$$

The gradient with respect to $\mathbf{w}$ vanishes at the minimum point. Solving for $\mathbf{w}$ then yields

$$\nabla_{\mathbf{w}}\mathcal{E} = -2\mathbf{F}^T\mathbf{d} + 2\mathbf{F}^T\mathbf{F}\mathbf{w} = 0 \;\Rightarrow\; \mathbf{w} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\mathbf{d}\,.$$

2. Averaging over K experts does not change the bias: for a given input $\mathbf{x}$, we have

$$(f(\mathbf{x}) - E_{\mathcal{T}}[y])^2 = \left(f(\mathbf{x}) - E_{\mathcal{T}}\left[\sum_{k=1}^{K}\frac{F_k(\mathbf{x})}{K}\right]\right)^2 = \left(f(\mathbf{x}) - \sum_{k=1}^{K}\frac{E_{\mathcal{T}}\left[F(\mathbf{x})\right]}{K}\right)^2$$
$$= (f(\mathbf{x}) - E_{\mathcal{T}}[F(\mathbf{x})])^2$$

where $f(\mathbf{x})$ denotes the desired function value and $\mathcal{T}$ denotes the training set; the second equality follows because the experts are identical, and thus have an identical expected output, here denoted $E_{\mathcal{T}}[F(\mathbf{x})]$.

The variance decreases by factor $1/K$ if the learning of different experts is independent because variances of independent variables add up and multiplying with $w_k = 1/K$ decreases the variance by factor $1/K^2$. In more detail, we have:

$$E_{\mathcal{T}}[(y - E_{\mathcal{T}}[y])^2] = E_{\mathcal{T}}\left[\left(\sum_{k=1}^{K}F_k(\mathbf{x})/K - E_{\mathcal{T}}[\sum_{k=1}^{K}F_k(\mathbf{x})/K]\right)^2\right]$$

$$= E_{\mathcal{T}}\left[\left(\sum_{k=1}^{K}(v_k(\mathbf{x}))/K\right)^2\right] = E_{\mathcal{T}}\left[\sum_{k,l=1}^{K}v_k(\mathbf{x})v_l(\mathbf{x})\right]/K^2$$

$$= \sum_{k=1}^{K}E_{\mathcal{T}}[v_k^2(\mathbf{x})]/K^2 + \sum_{k,l=1,k\neq l}^{K}E_{\mathcal{T}}[v_k(\mathbf{x})]E_{\mathcal{T}}[v_l(\mathbf{x})]/K^2 = \sum_{k=1}^{K}E_{\mathcal{T}}[v^2(\mathbf{x})]/K^2 = E_{\mathcal{T}}[v^2(\mathbf{x})]/K$$

where $v_k(\mathbf{x}) = F_k(\mathbf{x}) - E_{\mathcal{T}}[F_k(\mathbf{x})]$; with this notation, $E_{\mathcal{T}}[v_k^2(\mathbf{x})]$ is the variance of expert k. If the learning of the different experts is independent, the v_k are independent random variables, which yields the fourth equality. The fifth follows because $E_{\mathcal{T}}[v_k(\mathbf{x})] = E_{\mathcal{T}}[F_k(\mathbf{x})] - E_{\mathcal{T}}[F_k(\mathbf{x})] = 0$, and because identical experts have identical variances, here denoted $E_{\mathcal{T}}[v^2(\mathbf{x})]$.

3. In ensemble averaging the weights are fixed (see exercise problem 2). In Bayesian learning, the networks are averaged over the posterior probabilities of the models. In Bayesian learning the averaging is done over the probability distribution while in ensemble learning it is done over the output. These are not equivalent in general, but in some cases they are roughly same (see answer for problem 4.3 next week). Ensemble learning can therefore be seen as an approximation to the Bayesian averaging.

4. Assume that N_i samples are needed to train expert E_i so that the resulting error rate is $\epsilon_i < 0.5$, i=1,2,3. Note that if we had $\epsilon_i > 0.5$ we could simply switch each response to obtain error rate $1 - \epsilon_i < 0.5$. Then N_1 data samples are required to train the first expert.

   To train expert E_2, we need N_2 samples such that half of them are correctly classified by E_1 and half are misclassified. We must therefore run data by E_1 until it has misclassified $N_2/2$ samples and correctly classified $N_2/2$ samples. Since E_1 has error rate $\epsilon_1 < 1/2$, this requires $\max(\frac{N_2}{2\epsilon_1}, \frac{N_2}{2(1-\epsilon_1)}) = \frac{N_2}{2\epsilon_1}$ samples on average. (Note: this assumes the misclassified samples are evenly distributed in the data. It would be more realistic to consider the classifications as a stochastic process and compute how long it takes on average to produce the required numbers of correctly and incorrectly classified samples. The answer that would yield is

$$\sum_{k=0}^{\infty} \left((N_2 + k) \binom{N_2 + k - 1}{N_2/2 - 1} \left(\epsilon_1^{N_2/2} (1-\epsilon_1)^{N_2/2+k} + \epsilon_1^{N_2/2+k} (1-\epsilon_1)^{N_2/2} \right) \right)$$

   which can be approximated numerically.)

   To train expert E_3, we need N_3 samples for which the first two experts disagree. In a binary classification problem the probability of disagreement is

$$p(E_1 \ \textit{disagrees with} \ E_2) = p(E_1 \ \textit{correct}, E_2 \ \textit{incorrect}) + p(E_1 \ \textit{incorrect}, E_2 \ \textit{correct}) \ .$$

   Assuming that the responses of E_1 and E_2 are independent, this reduces to

$$\begin{aligned} p(E_1 \ \textit{disagrees with} \ E_2) &= p(E_1 \ \textit{correct}) p(E_2 \ \textit{incorrect}) + p(E_1 \ \textit{incorrect}) p(E_2 \ \textit{correct}) \\ &= (1-\epsilon_1)\epsilon_2 + \epsilon_1(1-\epsilon_2) = \epsilon_1 + \epsilon_2 - 2\epsilon_1\epsilon_2 \ . \end{aligned}$$

   From this we can compute the number of samples required to train E_3 (assuming the samples for which E_1 and E_2 disagree are evenly distributed).

   If samples aren't reused in training, the total number of samples required to train the committee machine is

$$N_{committee} = N_1 + \frac{N_2}{2\epsilon_1} + \frac{N_3}{\epsilon_1 + \epsilon_2 - 2\epsilon_1\epsilon_2} \ .$$

   If the experts are identical, this simplifies to

$$N_{committee} = N \left(1 + \frac{1}{2\epsilon} + \frac{1}{2\epsilon(1-\epsilon)} \right) \ .$$

   Schapire (1990) proved the following bound for the error rate of the committee machine:

$$\epsilon_{committee} \leq g(\epsilon) = 3\epsilon^2 - 2\epsilon^3 \ .$$

   The graph of this bound is shown in Haykin, Fig. 7.3. Each value of the bound $g(\epsilon)$ corresponds to a certain ϵ, from which one can calculate the required value of $N_{committee}$.