mlp-mdl

1 Introduction

This document details the procedures used to run both **mlp-mdl-3h** and **mlp-mdl-vh**. It uses much of the notation introduced in Hinton and van Camp [1][1]. The two methods only differ in that mlp-mdl-3h uses a fixed number of hidden units, $n_h = 3$, while mlp-mdl-vh uses a variable number of hidden units, chosen using the following information argument: If each weight in the network takes on average r bits to specify and if training set, $\mathcal{T}$, contains $I_{\mathcal{T}}$ bits then the number of hidden units is chosen as:

$$n_h = \frac{I_{\mathcal{T}}/r - 1}{n_i + 2} \tag{1}$$

where a single output unit has been assumed and n_i is the number of input units. The information (in bits) of a training set is estimated as:

$$I_{\mathcal{T}} = 0.5 \log_2 \frac{\sigma_t^2}{\sigma_r^2}$$

where σ_t^2 is the variance of the targets and σ_r^2 the residual variance of a linear regression model.

[1] Available from the web page at http://www.cs.toronto.edu/~delve/methods/mlp-ese-1/home.html

2 Priors

The input to hidden weights are assigned a prior distribution $P_{IH}(w)$ and all remaining weights in the network are assigned a different prior $P_R(w)$. Both priors are of the form:

$$P(w) = \sum_i \pi_i P_i(w) \tag{2}$$

where π_i is the mixing proportion of zero mean Gaussian P_i. Having fixed the architecture, training proceeds using the modified MDL cost:

$$C(\alpha) =< E > +\alpha\hat{G} \tag{3}$$

where $\hat{G} = -\log \sum_i \pi_i e^{-G_i}$ and G_i is the Kullback Liebler distance between the posterior and prior distributions of Gaussian i. Both P_{IH} and P_R are initialized with a single Gaussian component having $\sigma = 0.15$. Up to 50 epochs of congugate gradient minimization are allowed before incrementing α using the schedule $\alpha = \{.025, .05, .15, .2, .25, .3, .4, .5, .6, .7, .8, .9, 1\}$. When α attains its MDL value of 1, we allow ten repetitions of 50 congugate gradient epochs with search direction resets between each repetition. A first set of test set predictions are made using the final network configuration.

The single component prior P_{IH} (with variance σ^2) is then split into 2 components, one with $\frac{\sigma}{2}$ and the other with 2σ. The standard deviations are frozen for 150 congugate gradient updates and then allowed to adapt for another 8 repetitions of 50 conjugate gradient epochs. A second set of test set predictions are again made using this network configuration.

3 Ensembles

The process is repeated four times for each training set, resulting in 8 sets of test set predictions. In the case of mlp-mdl-3h, the learning is simply repeated four times, while in the case of mlp-mdl-vh, learning is repeated for $r = \{0.5, 1, 2, 4\}$.

The final prediction for test case j, (O_j) is obtained by averaging the 8 individual predictions, o_{jk} :

$$
\begin{aligned}
O_j &= \sum_{k=1}^{8} \beta_k o_{jk} \\
\beta_k &= \frac{\exp(-(C_k - m)/T)}{\sum_l \exp(-(C_l - m)/T)}
\end{aligned}
$$

where the temperature, $T = 10$ and $m = 5$ for predictions made with the single Gaussian prior and $m = 10$ for the two Gaussian prior.

4 Invoking scripts

If you manage to build the software, all predictions for a task can be made by invoking the script *ensemble_sh*. This in turn calls the tcl script *ensemble* (both are supplied in the tar file). There are hard coded paths at the top of each of these scripts that will need to be changed to suit your setup.

References

[1] G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the*

the Sixth Annual Conference on Computational Learning Theory, pages 5–13, 1993.