Regularization of Neural Network Fitting

Stephen Vardeman

Analytics Iowa LLC

ISU Statistics and IMSE

Penalized training error for a neural net

Suppose that coordinates of input vectors in a training set have been standardized and one wants to regularize the fitting of a neural net. One way of proceeding is to define a penalty function like

$$J(\mathbf{A}) = \sum_{h=0}^{H} \sum_{i,j} \left(a_{ij}^{h}\right)^2 \tag{1}$$

for $\mathbf{A}$ standing for the entire set of weights in $\mathbf{A}^0, \mathbf{A}^1, \ldots, \mathbf{A}^H$ (it is not completely clear whether one wants to include weights on the "bias" terms in the neural net sums in (1)) and seek to optimize the penalized total training loss

$$\sum_{i=1}^{N} L\left(\hat{\mathbf{f}}_{\mathbf{A}}(\mathbf{x}_i), y_i\right) + \lambda J(\mathbf{A}) = N \cdot \overline{\text{err}} + \lambda J(\mathbf{A}) \tag{2}$$

for a $\lambda > 0$.

Gradient descent for penalized fitting

By modifying the basic gradient descent recursion previously discussed to

$$a_{\text{new}} = a_{\text{current}} - \gamma \left(D \left(a_{\text{current}} \right) + 2\lambda a_{\text{current}} \right)$$

one arrives at an algorithm for optimizing the penalized training loss (2). Potentially, an appropriate value for λ might be chosen based on cross-validation.

Bayes analysis

Something that may at first seem quite different comes from a Bayesian point of view. For example, with a univariate regression model for outputs

$$y_i = f\left(\mathbf{x}_i|\mathbf{A}\right) + \epsilon_i$$

for the ϵ_i iid $\mathrm{N}\left(0, \sigma^2\right)$, a likelihood is simply

$$l\left(\mathbf{A}, \sigma^2\right) = \prod_{i=1}^{N} h\left(y_i|f\left(\mathbf{x}_i|\mathbf{A}\right), \sigma^2\right)$$

for $h\left(\cdot|\mu, \sigma^2\right)$ the normal pdf.

If then $g\left(\mathbf{A}, \sigma^2\right)$ specifies a prior distribution for $\mathbf{A}$ and σ^2, a posterior for $\left(\mathbf{A}, \sigma^2\right)$ has density proportional to

$$l\left(\mathbf{A}, \sigma^2\right) g\left(\mathbf{A}, \sigma^2\right)$$

Priors

For example, one might well assume that *a priori* the as are iid $\mathrm{N}(0, \eta^2)$ (where small η^2 will provide regularization and it is again unclear whether one wants to include the as corresponding to bias terms in such an assumption or to instead provide more diffuse priors for them, like improper "Uniform$(-\infty, \infty)$" or at least large variance normal ones). A standard improper prior for σ^2 is $\ln \sigma \sim$Uniform$(-\infty, \infty)$. In any case, whether improper or proper, abuse notation and write $g\left(\sigma^2\right)$ for a prior density for σ^2.

Log-posterior and penalized training error

Then with independent mean 0 variance η^2 priors for all the weights (except possibly the ones for bias terms that might be given Uniform$(-\infty, \infty)$ priors), $\ln\left(l\left(\mathbf{A}, \sigma^2\right) g\left(\mathbf{A}, \sigma^2\right)\right)$ is proportional to

$$-NK\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - f\left(\mathbf{x}_i|\mathbf{A}\right)\right)^2 - \frac{1}{2\eta^2}J(\mathbf{A}) + \ln g\left(\sigma^2\right)$$

$$= -NK\ln(\sigma) + \ln g\left(\sigma^2\right) - \frac{1}{\sigma^2}\left(\sum_{i=1}^{N}\left(y_i - f\left(\mathbf{x}_i|\mathbf{A}\right)\right)^2 + \frac{\sigma^2}{\eta^2}J(\mathbf{A})\right) \quad (3)$$

(flat improper priors for the bias weights correspond to the absence of terms for them in the sums for $J(\mathbf{A})$ in form (1)). This recalls display (2) and suggests that appropriate λ for regularization can be thought of as a variance ratio of "observation variance" and prior variance for the weights.

MCMC and maximum posterior density

It's clear how to define Metropolis-Hastings-within-Gibbs algorithms for sampling from $l\left(\mathbf{A}, \sigma^2\right) g\left(\mathbf{A}, \sigma^2\right)$. But typically the high dimensionality of the parameter space combined with the symmetry-derived multi-modality of the posterior will prevent one from running an MCMC algorithm long enough to fully detail the posterior But detailing the posterior may not really be necessary or even desirable. Rather, one might simply run the MCMC algorithm monitoring the values of $l\left(\mathbf{A}, \sigma^2\right) g\left(\mathbf{A}, \sigma^2\right)$ corresponding to the successive MCMC iterates. An MCMC algorithm will spend much of its time where the corresponding posterior density is large and we can expect that a long MCMC run will identify a nearly modal value for the posterior. Rather than averaging neural nets according to the posterior, one might instead use as a predictor a neural net corresponding to a parameter vector (at least locally) maximizing the posterior.

MCMC and maximum posterior density cont.

One might even take the parameter vector in an MCMC run with the largest $l\left(\mathbf{A}, \sigma^2\right) g\left(\mathbf{A}, \sigma^2\right)$ value and for a grid of σ^2 values around the empirical maximizer use the back-propagation algorithm modified to fully optimize

$$\sum_{i=1}^{N}\left(y_i - f\left(\mathbf{x}_i|\mathbf{A}\right)\right)^2 + \frac{\sigma^2}{\eta^2} J\left(\mathbf{A}\right)$$

over choices of $\mathbf{A}$. This, in turn, could be used with relationship (3) to perhaps improve somewhat the result of the MCMC "search."