Machine Intelligence 1 **Exercise Sheet 5**
WS 2017/18, Obermayer/Augustin due: 22.11.2017 at 23:55

Validation & Regularization

Exercise T5.1: Validation (tutorial)

(a) What is validation and why is it needed?

(b) What is the difference between *overfitting* and *underfitting*?

(c) Discuss the techniques *test set method* and *cross validation* to perform validation.

(d) How can hyperparameters (e.g. number of layers/neurons, regularization strength) of a model be selected using these techniques, and how can the resulting model be validated?

Exercise T5.2: Regularization (tutorial)

(a) What is the effect of the following alternative regularization terms, when minimizing the total training cost function ("risk"), $R(\underline{\mathbf{w}}) = E^T(\underline{\mathbf{w}}) + \lambda\, E^R(\underline{\mathbf{w}})$ for d-dim. params. $\underline{\mathbf{w}}$?

$$E^R(\underline{\mathbf{w}}) = \frac{1}{2p}\,||\underline{\mathbf{w}}||_2^2 = \frac{1}{2p}\sum_{i=1}^{d} w_i^2 \qquad (L_2 \text{ norm regularization: "weight decay"})$$

$$E^R(\underline{\mathbf{w}}) = \frac{1}{p}\,||\underline{\mathbf{w}}||_1 = \frac{1}{p}\sum_{i=1}^{d} |w_i| \qquad (L_1 \text{ norm regularization: "sparsify" / "Lasso"})$$

(b) What is the weight parameter vector $\underline{\mathbf{w}}$ with minimal risk $R(\underline{\mathbf{w}})$ for a linear neuron with a quadratic training cost function and weight decay regularization?

Exercise T5.3: Nonlinear basis functions (tutorial)

Instead of dealing with deep neural networks, many machine learning approaches use a linear neuron on input samples $\underline{\mathbf{x}}$, which are "expanded" by non-linear basis functions $\phi_i(\underline{\mathbf{x}})$, i.e., $y(\underline{\mathbf{x}}) = \sum_{i=1}^{d} w_i\, \phi_i(\underline{\mathbf{x}})$. In the lecture, the functions ϕ_i are *radial basis functions*, but here we want to discuss the set of all monomials up to some order.

(a) What are monomials and how is a linear combination thereof called?

(b) Monomials can grow very large for big inputs. To standardize the input space, one often *spheres* the data before expansion. How is "whitening" or "sphering" performed?

(c) Monomial basis functions can be regularized by weight decay.

(d) What is the weight parameter vector $\underline{\mathbf{w}}$ with minimal risk $R(\underline{\mathbf{w}})$ for a linear neuron with basis functions ϕ_i with a quadratic training cost function and weight decay regularization?

Exercise H5.1: Cross-validation (homework, 10 points)

This exercise asks you to assess the impact of a quadratic regularization penalty on the parameter estimates for a linear connectionist neuron to solve a regression task with a quadratic cost function.

Data: The file `TrainingRidge.csv` contains the *training set*, which are 200 observations $\{(\underline{\mathbf{x}}^{(\alpha)}, y_T^{(\alpha)})\}$. The two input variables for each observation $\underline{\mathbf{x}}^{(\alpha)} = (x_1^{(\alpha)}, x_2^{(\alpha)})^T$ are contained in the first 2 columns. The target values (labels) $y_T^{(\alpha)}$ are contained in the last column. The second file `ValidationRidge.csv` contains 1476 combinations $\{(\underline{\mathbf{x}}^{(\beta)}, y_T^{(\beta)})\}$ with $\underline{\mathbf{x}}^{(\beta)} = (x_1^{(\beta)}, x_2^{(\beta)})^T$ – a 36×41 grid – as a *validation set* in the same format.

(a) (3 point) The observations are from a large space. Monomials (see below) can grow very large for bigger inputs. Perform *whitening* (sphering) of the training data, such that the resulting input samples are decorrelated, have zero mean and unit variance. Plot the sphered training and validation sets in two scatter-plots, where the color of the markers represents the associated label. Use the same sphering transformation as obtained from the eigendecomposition of the centered *training* data's covariance matrix also to sphere the validation set (i.e., do not compute a separate sphering transformation for that purpose).
Remember that the sphered data is given by $\{\underline{\mathbf{x}}_{\text{sphered}}^{(\alpha)}\}_{\alpha=1}^{p}$ with $\underline{\mathbf{x}}_{\text{sphered}}^{(\alpha)} = \underline{\mathbf{\Lambda}}^{-\frac{1}{2}} \underline{\mathbf{E}}^T \underline{\mathbf{x}}_{\text{centered}}^{(\alpha)}$.
Here $\underline{\mathbf{x}}_{\text{centered}}^{(\alpha)} = \underline{\mathbf{x}}^{(\alpha)} - \bar{\underline{\mathbf{x}}}$ denotes the centered data point α (w.r.t. the center of the training data $\bar{\underline{\mathbf{x}}} = \frac{1}{p}\sum_{\alpha=1}^{p} \underline{\mathbf{x}}^{(\alpha)}$), $\underline{\mathbf{E}} = (\underline{\mathbf{e}}_1, \ldots, \underline{\mathbf{e}}_N)$ is the eigenvector matrix and $\underline{\mathbf{\Lambda}} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ is the eigenvalue matrix for the eigendecomposition ($\underline{\mathbf{C}}\,\underline{\mathbf{e}}_i = \lambda_i \underline{\mathbf{e}}_i$) of the centered training data's covariance matrix $\underline{\mathbf{C}}$ with $C_{ij} = \frac{1}{p}\sum_{\alpha=1}^{p} x_{\text{centered},i}^{(\alpha)}\, x_{\text{centered},j}^{(\alpha)}$.

(b) (2 point) A single linear neuron is not able to predict the target labels very well. To increase the representational power of the model class, *expand* the sphered 2-dimensional input samples to all possible *monomials* up to degree 9. Here, a monomial of order k corresponds to a term $x_1^l x_2^m$ with $l+m=k$. The model should contain all 55 terms $x_1^l x_2^m$ with $l+m=k$ for $k = 0, 1, ..., 9$. These monomials can be enumerated by $i = 1, \ldots, d = 55$ defining $\phi_i(\underline{\mathbf{x}})$. The prediction function for the quadratic cost measure $E^T(\underline{\mathbf{w}})$ is given by

$$y(\underline{\mathbf{x}}; \underline{\mathbf{w}}) \;=\; \underline{\mathbf{w}}^\top \underline{\mathbf{x}}\,, \qquad \text{with} \qquad \underline{\mathbf{w}} \;=\; \left(\underline{\mathbf{\Phi}}\,\underline{\mathbf{\Phi}}^\top\right)^{-1} \underline{\mathbf{\Phi}}\,\underline{\mathbf{y}}^T$$

with input matrix $\underline{\mathbf{\Phi}} \in \mathbb{R}^{d,p}$ [having components $\Phi_{i,\alpha} = \phi_i(\underline{\mathbf{x}}^{(\alpha)})$] and a label vector $\underline{\mathbf{y}} \in \mathbb{R}^{1,p}$ (with components $y_T^{(\alpha)}$). Plot on the validation set the first 10 monomials $\phi_i(\underline{\mathbf{x}})$ ($0 \le k \le 3$) as well as the predicted function $y(\underline{\mathbf{x}})$ either as a scatter plot or as a 36×41 image, where the colors indicate the labels.

(c) (3 points) To avoid over-fitting of a linear neuron with the polynomial expansion of (b), the solution must be regularized with a weight-decay term, i.e., the risk $R(\underline{\mathbf{w}}) = E^T(\underline{\mathbf{w}}) + \lambda \frac{1}{2p} ||\underline{\mathbf{w}}||_2^2$ has to be minimized. For a regularization strength $\lambda > 0$, an input matrix $\underline{\mathbf{\Phi}} \in \mathbb{R}^{d,p}$ and a label vector $\underline{\mathbf{y}} \in \mathbb{R}^{1,p}$ (as above), the prediction function is

$$y(\underline{\mathbf{x}}; \underline{\mathbf{w}}) \;=\; \underline{\mathbf{w}}^\top \underline{\mathbf{x}}\,, \qquad \text{with} \qquad \underline{\mathbf{w}} \;=\; \left(\underline{\mathbf{\Phi}}\,\underline{\mathbf{\Phi}}^\top + \lambda \underline{\mathbf{I}}\right)^{-1} \underline{\mathbf{\Phi}}\,\underline{\mathbf{y}}^T\,,$$

where $\underline{\mathbf{I}}$ denotes the identity matrix.
To find the best regularization constant, perform a 10-fold cross-validation with the *training set* for all $\lambda \in \{10^z \,|\, z \in \{-4, -3.9, -3.8, \ldots, 3.9, 4\}\}$. Plot the resulting average and standard deviation of the MSE (mean squared error, i.e., average quadratic cost of the predictions) over all folds against λ (as an error-bar plot with a logarithmic x-axis for λ). Also

plot (similarly to (b)) the true labels of the validation set next to the predicted labels for the best regularization parameter (λ_T), which minimizes the average MSE over all folds .

(d) (2 points) To compare these empirical estimates of bias and variance with the true generalization error, repeat (c) with the polynomial expansion of the *validation set*. Is the best lambda (λ_G) different from λ_T? Compare the learned function in (c) with functions that are learned with λ_G on (i) the training set and (ii) the validation set.

Total 10 points.