Machine Intelligence 1 | **Exercise Sheet 5**

WS 2021/22, Obermayer/Kashef | due: 29.11.2021 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. parameters $\underline{\mathbf{w}}$?

$$
\begin{aligned}
E^R_{[\underline{\mathbf{w}}]} &= \tfrac{1}{2p}\, ||\underline{\mathbf{w}}||_2^2 = \tfrac{1}{2p} \sum_{i=1}^{d} w_i^2 \qquad (L_2 \text{ norm regularization: “weight decay”}) \\
E^R_{[\underline{\mathbf{w}}]} &= \tfrac{1}{p}\, ||\underline{\mathbf{w}}||_1 = \tfrac{1}{p} \sum_{i=1}^{d} |w_i| \qquad (L_1 \text{ norm regularization: “sparsify” / “Lasso”})
\end{aligned}
$$

(b) What is the optimal 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)

In order to fit highly non-linear functions, many machine learning approaches use a linear neuron on an alternate representation of the input samples $\underline{\mathbf{x}}$. This representation is an "expansion" of $\underline{\mathbf{x}}$ by non-linear basis functions $\phi_i(\underline{\mathbf{x}})$, i.e., $y(\underline{\mathbf{x}}) = \sum_{i=1}^{d} w_i\, \phi_i(\underline{\mathbf{x}})$. Here we want to discuss the set of all monomials up to some order.

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

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

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

(d) What is the optimal 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 regularization penalty on the parameters of a linear connectionist neuron to solve a regression task with a quadratic cost function. We will only consider a quadratic regularization term for this exercise.

The Data:

The file `TrainingRidge.csv` contains the *training set*, with 200 observations and corresponding target values (ground truth/labels) $\{(\underline{\mathbf{x}}^{(\alpha)}, y_T^{(\alpha)})\}$. The two input variables for each observation $\underline{\mathbf{x}}^{(\alpha)} = (x_1^{(\alpha)}, x_2^{(\alpha)})^\top$ appear in the first 2 columns. The target values $y_T^{(\alpha)}$ appear in the last column.

The data contained in the second file `ValidationRidge.csv` serves as the *validation set*. It follows the same format as above. The *validation set* contains 1476 pairs $\{(\underline{\mathbf{x}}^{(\beta)}, y_T^{(\beta)})\}$. The values of $\underline{\mathbf{x}}^{(\beta)} = (x_1^{(\beta)}, x_2^{(\beta)})^\top$ form a 36×41 grid in input space.

(a) (3 point) **Preprocessing**: Monomials (see details below in (b)) can grow very large in magnitude for bigger input values. Perform *sphering* of the training data, such that the resulting input samples are decorrelated, have zero mean and unit variance. The sphered data is given by

$$\{\underline{\mathbf{x}}_{\text{sphered}}^{(\alpha)}\}_{\alpha=1}^{p} \quad \text{with} \quad \underline{\mathbf{x}}_{\text{sphered}}^{(\alpha)} = \underbrace{\underline{\mathbf{\Lambda}}^{-\frac{1}{2}} \underline{\mathbf{E}}^\top}_{\substack{\text{sphering} \\ \text{transformation}}} \underline{\mathbf{x}}_{\text{centered}}^{(\alpha)} .$$

Here

$\underline{\mathbf{x}}_{\text{centered}}^{(\alpha)} = \underline{\mathbf{x}}^{(\alpha)} - \langle \underline{\mathbf{x}} \rangle$ denotes the centered data point α w.r.t. the center of the training data $\langle \underline{\mathbf{x}} \rangle = \frac{1}{p} \sum_{\alpha=1}^{p} \underline{\mathbf{x}}^{(\alpha)}$,

$\underline{\mathbf{E}} = (\underline{\mathbf{e}}_1, \ldots, \underline{\mathbf{e}}_N)$ is the eigenvector matrix and $\underline{\mathbf{\Lambda}} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ is the eigenvalue matrix for the eigendecomposition

$$\underline{\mathbf{C}}\,\underline{\mathbf{e}}_i = \lambda_i \underline{\mathbf{e}}_i$$

of the covariance matrix $\underline{\mathbf{C}}$ with $C_{ij} = \frac{1}{p} \sum_{\alpha=1}^{p} x_{\text{centered},i}^{(\alpha)} \, x_{\text{centered},j}^{(\alpha)}$.

Deliverables: Plot the sphered training and validation sets using two separate scatter-plots. Color the points according to their label.

Important:

- Use the same $\langle \underline{\mathbf{x}} \rangle$ computed from the training data for centering the validation data.
- Use the same sphering transformation obtained from the eigendecomposition of the centered *training* data's covariance matrix to sphere the validation set (i.e., do not compute a separate sphering transformation for the validation set).

(b) (2 points) **Feature Expansion**: 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 2D input 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 which feeds into the quadratic cost measure $E^T_{[\underline{\mathbf{w}}]}$ is given by

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

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 $\underline{\mathbf{y}}_T \in \mathbb{R}^{1,p}$ (with components $y_T^{(\alpha)}$).

Deliverables: Using the validation set, produce the following plots:

(i) The first 10 monomials $\phi_i(\underline{\mathbf{x}})$ ($i \in [0, 9]$) as a function of x_1, x_2. Visalize each monomial separately. You can visualize each monomial by using either a scatter plot or a 36×41 "heatmap"[1].

(ii) The predicted function $y(\underline{\mathbf{x}}; \underline{\mathbf{w}})$ as a function of x_1, x_2, also as a scatter plot or "heatmap" where the colors indicate the prediction value.

(c) (3 points) To avoid over-fitting when using the polynomial expansion above, we apply regularization using a weight-decay term, i.e., the risk $R_{[\underline{\mathbf{w}}]} = E^T_{[\underline{\mathbf{w}}]} + \lambda \frac{1}{2} ||\underline{\mathbf{w}}||_2^2$ 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 $\underline{\mathbf{y}}_T \in \mathbb{R}^{1,p}$ (as above), the prediction function is

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

where $\underline{\mathbf{I}}$ denotes the identity matrix.
To find the best value for the regularization coefficient, perform a 10-fold cross-validation with the *training set* for all $\lambda \in \{10^z \,|\, z \in \{-4, -3.9, -3.8, \ldots, 3.9, 4\}\}$. Each fold splits the original training set into a smaller training set and a *test* set.
Deliverables:

(i) Plot the average and standard deviation of the MSE (mean squared error, i.e., average quadratic cost of the predictions) over the *test* set for all folds against λ (as an error-bar plot with a logarithmic x-axis for λ).

(ii) Idenitfy the value of the best regularization coefficient λ_T^*, which has the minimal average MSE over all test folds.

(iii) Train the model using the entire original training set regularized by λ_T^*. Plot the true labels of the *validation set* alongside your model's predictions. What is the MSE of the model on the validation set?

(d) (2 points) To compare these empirical estimates of bias and variance with the true generalization error, repeat (c) with the same polynomial expansion of the *validation set*. That is:

(i) Replace your original training set with the validation set and treat this as your new training set.

[1] $36 \times 41 = 1476$ is the number of observations in the validation set.

(ii) Keep the same sphering transformation as before. Reuse the same matrix of eigenvectors $\underline{\mathbf{E}}$ and matrix of eigenvalues $\underline{\mathbf{\Lambda}}$ that you used to preprocess the data in (a).

(iii) Perform the same expansion as in (b) nd cross validation in order to identify the best regularization coefficient $\rightsquigarrow \lambda_G^*$ using this data.

(iv) What is the MSE of the model on the entire original *validation set*? Keep in mind that this is data you actually used for training the model.

(e) Is λ_G^* different from λ_T^*? Compare by plotting the function learned in (c) using λ_T^* with the function that is learned in (d) using λ_G^* on

(i) the original training set
and

(ii) the original validation set.

Total 10 points.