Artificial Neural Networks

Prof. Dr. Sen Cheng

Oct 28, 2019

Problem Set 4: Model Selection/ Regularization

Tutors: Olya Hakobyan (olya.hakobyan@rub.de), José Donoso (jose.donoso@rub.de)

Further Reading: publications etc, book chapters

1. The holdout method

   (a) Load the file *'04_model_selection_data.npy'*. The first and second columns of the data refer to the predictors and targets, respectively. Split the dataset into training and validation sets using the *train_test_split* function from the *sklearn.model_selection* package. Use a ratio of 80:20 and set the parameter *random_state* of the *train_test_split* function to a fixed value.

   (b) Fit polynomials of degrees N to the data (say $N \in [1,6]$) using the package *sklearn*. To generate the polynomial features, use the class *PolynomialFeatures* from *sklearn.preprocessing*. Plot the fitting lines along with the training data points. Plot the *mean squared error* vs. the model complexity for the training and validation sets. Describe the resulting fits in terms of bias and variance. Which model seems to best fit the data?

   (c) When using different random states do you always have the same best fit model? Why/Why not?

2. k-fold cross-validation

   (a) Use k-fold cross-validation to determine the optimal model for the data in exercise 1. You can use the *KFold* function from *sklearn.model_selection* package to obtain indices for the training and validation sets. Although cross-validation uses all available data for both training and validation, it doesn't use all possible combinations of the data points. Therefore, to check the stability of your estimation, shuffle the data for each split by setting the parameter shuffle in *KFold* function to True.

   (b) Run the regression for $k \in [2,4,5,10,20]$. For each k, plot the training and validation performances vs model complexity and select the model with the best performance. Repeat and see for which k do you get the most stable pattern of training and test errors.

3. Regularization

   (a) Run regression using the best polynomial degree that you've identified in the last exercise. Then, use ridge regression for 9-degree polynomial features with different regularization strengths (λ). Use *Ridge* and *Lasso* classes from *sklearn.linear_model*.

   Note: for the current data the range of λ should be very small: $\lambda \in [0, 0.003]$.

   (b) Plot the training and validation error vs λ for the ridge regression as well as the error from the best fit model for comparison. How does the regularization affect the model performance?

   (c) Plot the the polynomial coefficients vs λ.

   (d) Repeat (a-c) for Lasso regression. Does the λ have the same influence as in ridge regression? Is the number of nonzero coefficients what you would have expected?