The Neural Code

Summer term 2022, Christian Leibold, Yuk-Hoi Yiu

Problem Set 7 **—Empirical rsik Minimization /KKT—** **28.06.2022**

Due date: 07.07.2022

---

1. (Unfalsifiable Model) Load the time series data from `data.txt` (First row: Time points, Second row: amplitudes).

(a) Plot the signal

(b) Find the Fourier coefficients that describe the signal and reconstruct the signal from its Fourier coefficients (FFT won't work since the time points are not equally sampled, use the Riemann sum approximation of the continuous Fourier integral prepared in `FT.py`).

(c) Plot the original and the Fourier-reconstructed signal on top of each other to show that they match very well.

(d) Take 100 random downsamples of the data, each containing a fraction f of of the data points and construct the Fourier coefficients in the same way as in b,c. Compute the training error (L_2 loss) and the test error from the prediction of the remaining $1-f$ data points (given the Fourier coefficients obtained from the training sample). Use $f = 0.5, 0.75, 0.9, 0.95$.

(e) What makes the model unfalsifiable?

2. (Polynomial SRM)

Load the data `data.txt` from the previous problem. A simple way of fitting polynomial models is to use multi-linear regression with powers t^n of the time variable t added as regressors (see previous sheet).

(a) Write a function that fits the timeseries with a polynomial of order 8 using `statsmodels.regression.linear_model.OLS`.

(b) Compute the training error (L_2 loss) as a function of the polynomial order. Explain your result.

(c) Take 100 random downsamples of the data, each containing a fraction f of of the data points and fit the model. Compute the training error and the test error from the prediction of the remaining $1-f$ data points (given the model parameters obtained from the training sample). Use $f = 0.5, 0.75, 0.9, 0.95$.

(d) What is the optimal model complexity?

3. (Capacity)

(a) Write a function that generates 200 random data points in N dimensions, where N is a function argument and randomly assign binary labels to the data points.

(b) Use logistic regression (`sklearn.linear_model.LogisticRegression`) to seperate the data points for increasing $N > 2, 3, ...$ and monitor the error rate as a function of N.