Assignment 1

Memoryless Systems

442.022 Nonlinear Signal Processing, Practical
Summer Term 2021

Christian Toth, christian.toth@tugraz.at
Signal Processing and Speech Communication Laboratory
Graz University of Technology

Last updated: April 19, 2021

Overview

This assignment deals with the topics universal approximators and system identification, harmonic analysis, statistical analysis, and propagation of uncertainty in the context of memoryless systems. Each presented problem is divided into tasks, for each of which we indicate its number of achievable points. Additionally, some tasks are marked with a ★. These starred tasks are intended to be bonus tasks that add points to the nominal total of 100 points for the assignment. For this assignment you can earn up to 40 bonus points, which roughly corresponds to an improvement of one grade for the entire course.

Anyways, which tasks (starred or unstarred) you choose to work on and consequently, how many points you can possibly achieve, is up to you. Given the diverse backgrounds of the participants this should provide you with some freedom to work on what is most interesting and useful to you!

Submission Guidelines

Deadline: 06 May 2021, 11:59 p.m.
Format: Report as .pdf and code as .zip via the TeachCenter.
Late Submission Policy: Late submission within the first, second and third day after the deadline reduces the achieved number of points to 85%, 70% and 50% respectively. Submissions late by more than three days will not be accepted and thus no points can be achieved for the assignment in such a case.

For further information see also the course repository.

Universal Approximators and System Identification

Problem 1

In this problem we want to compare different universal approximators for memoryless systems. For this purpose, consider the setup depicted in Figure 1 where we want to identify the static nonlinearity $f(\cdot)$. In `/data/csv/1_1_system_identification/` you will find a training set $X_{train} = \{(x_i, y_i)\}_{i=1\ldots N_{train}}$ and test set $X_{test} = \{(x_i, y_i)\}_{i=1\ldots N_{test}}$ containing $N_{train} = 50$ respectively $N_{test} = 30$ i.i.d.[1] samples, where $x_i \sim \mathcal{U}(0, 1)$ and $y_i = f(x_i) + \nu_i$ with $\nu_i \sim \mathcal{N}(0, \sigma_\nu^2)$.

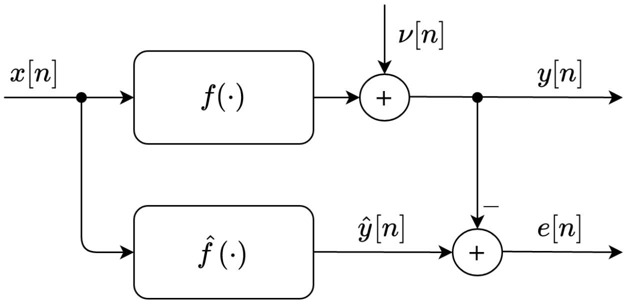

Figure 1: System identification setup for Problem 1

Task a) (8 Points) Use the training set to find a polynomial model

$$\hat{f}(x) = \sum_{p=0}^{P} \alpha_p x^p \tag{1}$$

of order P and parameters $\theta = \{\alpha_p\}_{p=1}^{P}$ that are optimal w.r.t. the mean squared error (MSE)

$$J(\theta, P) = \frac{1}{N} \sum_{e=0}^{N-1} e_i^2 \tag{2}$$

as a loss function. Choose a suitable model order P by performing a hyperparameter search, i.e. fit and compare models of increasing order up to $P = 25$. For the fitted models plot the MSE achieved on the training and test set as a function of the model order. Answer the following questions.

- What do you notice when you compare the training and test set MSE curves?
- Which model order performs best (i.e. minimizes the MSE) on the training and test set respectively? Report their MSE scores and include a plot of your approximated systems $\hat{f}$ over the support $x \in [0, 1]$ with training and test set samples.

[1]independent and identically distributed

- Which model would you choose to predict new samples that are neither in your training nor in your test set? Explain your reasoning.

Task b) (3 Points, ★) In practice you do not want to do model evaluation with your test set because the test set score should give you a realistic estimate of how you can expect your model to perform after deployment. Once you make a selection based on your test set your estimate will be overconfident. To avoid this problem, split your training set into a (smaller) training set and validation set (disjoint) and select a suitable model order based on the validation set performance. Answer the following questions.

- What do you notice when you compare the training and validation set MSE curves? Include a plot in your report.
- Which model order performs best on the validation set? Report the test set MSE score for this model and plot your approximated system as in Task a).
- How did you choose the split size of your validation set? What is in favor of a large/small validation set? Explain your reasoning.

Task c) (8 Points) Consider now a Gaussian radial basis function (RBF) model $\hat{f}$

$$\hat{f}(x) = \alpha_0 + \sum_{p=1}^{P} \alpha_p \mathrm{e}^{-\frac{(x-c_p)^2}{2w_p^2}} \tag{3}$$

of order P. The model is linear in the scaling coefficients $\{\alpha_p\}$ but nonlinear in the centers $\{c_p\}$ and widths $\{w_p\}$. To circumvent this problem place the centers of a given model equally spaced on the support of x and empirically choose a single, global width parameter $w_1 = w_2 = \cdots = w$. Using your experience from Task a) and Task b) approximate the true system $f(\cdot)$. Compare the resulting RBF model to your polynomial model by reporting their achieved MSE and plotting the models over the support of x. Make sure to have the same training setup (e.g. validation set) for both models. Answer the following questions.

- How does the model order influence the RBF models' performance? Can you choose a model order that is too high?
- How does the width parameter influence the RBF models' performance? Is it better to choose a small or large width?
- How does the choice of width and model order interact/depend on each other?
- Think about a multi-variate system, e.g. $x \in \mathbb{R}^n$ with $n = 100$, what could be potential problems when we try to choose the RBF centers and widths by hand?

Task d) (5 Points, ★) In Task c) we only fitted the scaling coefficients and chose the RBF centers and width parameters statically. Now the question is: can we actually learn optimal center and width parameters? A standard (although not necessarily the best) way to do nonlinear optimization is gradient descent (GD). Consider the following tasks.

- Represent the RBF model in (3) as a feed-forward neural network, i.e. give it's weight matrices, activation functions in terms of the RBF parameters (widths, centers and scaling coefficients) and sketch a network for P=3.
- Compute the gradients of the widths, centers and scaling coefficients analytically. (Optional)
- Implement an RBF-NN model in PyTorch and train the model via GD. Choose a suitable model order empirically (you don't have to do a hyperparameter search). Compare it to the model with statically chosen centers and widths by reporting MSE on the test set and plotting the models. Which model performs better? What would you expect?
- What other methods for finding the center and width parameters are there? Do a short literature search and report the method(s) you found. Make sure to reference your sources properly.

Task e) (9 Points) Approximate the true system $f(\cdot)$ now with a feed-forward neural network. You can choose the architecture empirically so try to play around with different activation functions, number of layers, layer sizes, etc. Make sure to document your findings and final design in the report. Compare your model to your previously trained RBF and polynomial models by reporting their achieved MSE and plotting the models over the support of x.

Task f) (5 Points) Conclude your findings from your experiments, i.e. in the given scenario which function approximator works best and why? What are their strengths and weaknesses, and under what circumstances would you prefer to use one over the other? Explain your reasoning (it is ok to argue based on intuitions).

Harmonic Analysis and Equalization

Problem 2

An arbitrary memoryless system

$$y[n] = f(x[n]) \tag{4}$$

can be approximated with its K^{th}-order Taylor series expansion

$$\hat{f}(x[n]) = \sum_{k=0}^{K} \frac{1}{k!} \left. \frac{d^{(k)} f(z)}{dz^k} \right|_{z=c} (x[n] - c)^k \tag{5}$$

around the center c. Consider a sinusoidal input signal $x[n] = A\cos(\theta_0 n)$. Then any third order approximation will assume the generic form

$$\hat{f}(x[n]) = \alpha_0 + \alpha_1 \cos(\theta_0 n) + \alpha_2 \cos(2 \cdot \theta_0 n) + \alpha_3 \cos(3 \cdot \theta_0 n) \tag{6}$$

where the coefficients $\alpha_0, \ldots, \alpha_3$ depend on the expansion center c, the signal amplitude A, and the function $f(\cdot)$ and its derivatives.

Task a) (6 Points) Analytically derive the coefficients $\alpha_0, \ldots, \alpha_3$ for a generic system $f(x[n])$. How can these coefficients be interpreted? **Hint:** The identities $\cos^2\phi = \frac{1}{2}(1 + \cos(2\phi))$ and $\cos^3\phi = \frac{1}{4}(3\cos\phi + \cos(3\phi))$ will be useful.

Now consider the system

$$y(x[n]) = \sigma(x[n]) = \frac{1}{1 + \mathrm{e}^{-x[n]}}. \tag{7}$$

Task b) (6 Points) Reuse your results from Task a) to derive the third order Taylor approximation around the centers $c \in \{0, \ln 2\}$ for the system given in (7).

Task c) (8 Points) Verify your results for both approximations from Task b) numerically for the input signal parameters $A \in \{1, 3\}$ and $\theta_0 = \frac{2\pi}{5}$. In your report include plots comparing the true and approximated systems with respect to (i) their input/output curves, and (ii) their frequency spectra given the amplitude A and expansion center c. Discuss the impact of the expansion center and the signal amplitude on your results.

Statistical Analysis

Problem 3

Consider the random variable X with density

$$p_X(x) = \frac{\lambda}{2} \mathrm{e}^{-\lambda|x|}, \quad x \in [-\infty, \infty].$$

Task a) (8 Points) Find the corresponding characteristic function $\Phi(\omega)$.

Task b) (5 Points, ★) Show that all odd-order moments of X are zero. **Hint:** Exploit the fact that p_X is an even function and the constraint that this implies for the moments!

Problem 4

Let X, Y and Z be random variables, where $Z = X + Y$, and assume that X and Y are statistically independent, i.e., their joint distribution factorizes as $p_{X,Y} = p_X \cdot p_Y$.

Task a) (7 Points) Show that $p_Z = p_Y * p_X$, where the asterisk denotes convolution, i.e.,

$$p_Z(z) = \int_x p_X(x) \cdot p_Y(z - x) dx \tag{8}$$

$$= \int_y p_Y(y) \cdot p_X(z - y) dy \tag{9}$$

Task b) (3 Points, ★) Show or refute that the cumulants of X and Y are additive, i.e., the cumulants of Z are given as $c_p^Z = c_p^X + c_p^Y$.

Propagation of Uncertainty and Normalizing Flows

Problem 5

Consider a very simplified version of an (indoor) localization task, i.e., we want to estimate the position of an agent in a certain area around a base station as depicted in Figure 2.

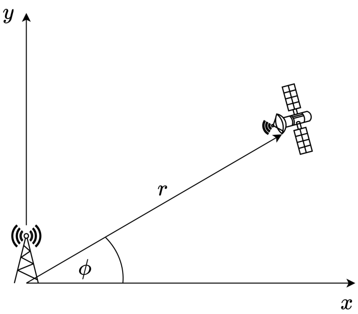

Figure 2: Sketch of the application scenario in Problem 5.

Specifically, the agent is located at position $\boldsymbol{z} = \begin{pmatrix} x & y \end{pmatrix}^T$ (in Cartesian coordinates) and sends a signal to the base station located at $\boldsymbol{z_0} = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$. From this signal the base station estimates the polar coordinates $\hat{\boldsymbol{w}} = \begin{pmatrix} \hat{r} & \hat{\phi} \end{pmatrix}^T$ of the agent. We model the uncertainty in this measurement process as zero-mean [2], additive noise $\boldsymbol{\nu} \sim p_{\boldsymbol{\nu}}$ corrupting the true polar coordinates $\boldsymbol{w}$ which are readily computed by the nonlinear transformation

$$\boldsymbol{w} = f(\boldsymbol{z}) = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \arctan \frac{y}{x} \end{pmatrix}. \tag{10}$$

Hence, the measurement process is described as

$$\hat{\boldsymbol{w}} = \boldsymbol{w} + \boldsymbol{\nu} = f(\boldsymbol{z}) + \boldsymbol{\nu}. \tag{11}$$

Finally, from the measurement $\hat{\boldsymbol{w}}$ we want to estimate the agent's position in Cartesian coordinates by simply applying the inverse transformation

$$\hat{\boldsymbol{z}} = f^{-1}(\hat{\boldsymbol{w}}) = \begin{pmatrix} r \cdot \cos\phi \\ r \cdot \sin\phi \end{pmatrix}. \tag{12}$$

The whole measurement and estimation process is depicted in Figure 3.

Task a) (6 Points) Derive an expression for the conditional density $p_{\hat{\boldsymbol{w}}|\boldsymbol{z}}$ of the measurements $\hat{\boldsymbol{w}}$ when observing the true position $\boldsymbol{z}$. **Hint:** Make use of the "change-of-variable" formula

[2] The expected value of the noise vanishes, i.e., $\mathbb{E}_{\boldsymbol{\nu}}(\boldsymbol{\nu}) = 0$.

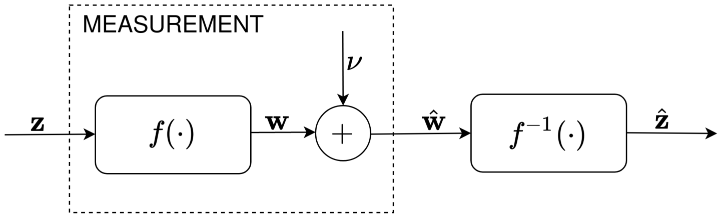

Figure 3: Measurement and estimation process of the localization in Problem 5.

and/or the convolution property of the delta distribution $\int \delta(\xi - \xi_0) g(\xi) = g(\xi_0)$ (what is $p_{\hat{\boldsymbol{w}}|\boldsymbol{w},\boldsymbol{\nu}}$?).

Task b) (4 Points) Compute the conditional expectation $\mathbb{E}_{\hat{\boldsymbol{w}}|\boldsymbol{z}}(\hat{\boldsymbol{w}})$ of the measurements for a given true position. Is this an unbiased estimate for the true polar coordinates $\boldsymbol{w}$?

Task c) (10 Points) Assume now further that the components of the noise vector $\boldsymbol{\nu} = \begin{pmatrix} \nu_r & \nu_\phi \end{pmatrix}^T$ are independently distributed, i.e., $\nu_r \sim p_{\nu_r}$, $\nu_\phi \sim p_{\nu_\phi}$ and $p_{\boldsymbol{\nu}} = p_{\nu_r} \cdot p_{\nu_\phi}$. Compute the conditional expectation $\mathbb{E}_{\hat{\boldsymbol{w}}|\boldsymbol{z}}(f^{-1}(\hat{\boldsymbol{w}}))$ of estimated position for a given true position. Is this an unbiased estimate for the true coordinates $\boldsymbol{z} = f^{-1}(\boldsymbol{w})$? Remark: It is easier to compute this expectation in the polar coordinate space.

Task d) (4 Points, ★) Assume now that you get multiple i.i.d. measurements $\{\hat{\boldsymbol{w}}_n\}_{n=1}^N$ for the same agent position $\boldsymbol{z}$ and you can choose between to estimators

$$\hat{\boldsymbol{z}}_{\boldsymbol{A}} = f^{-1}\left(\frac{1}{N} \sum_n \hat{\boldsymbol{w}}_n\right) \tag{13}$$

and

$$\hat{\boldsymbol{z}}_{\boldsymbol{B}} = \frac{1}{N} \sum_n f^{-1}(\hat{\boldsymbol{w}}_n). \tag{14}$$

Which one would you choose and why? What assumptions about the noise model are crucial for your decision? **Hint:** You can easily verify your answer with a small simulation.

Problem 6

We now want to employ normalizing flows to learn (multimodal) multivariate distributions. To visualize the distributions easily we stick to bivariate distributions. More specifically, we take as ground truth distribution a binary image and assign all white pixels zero and all black pixels equal probability. For the following tasks, code for loading and converting images, as well as for sampling from a binary image as ground truth distribution is provided in `./notebooks/assignments/1-6-flows.ipynb` . You can either use the provided images `./data/img/thumb.png` or `./data/img/up.png` , or find your own fancy image online (think about attributions!).

Task a) (15 Points) Use Pyro and PyTorch to implement a coupling flow (e.g. a spline coupling flow) and train the flow on samples drawn from the ground truth distribution. Report your architecture, hyperparameters and the number of samples you used for training. Report also the achieved log-likelihood. Further, include a scatter plot with 5000 samples drawn from and a log-likelihood plot of your learned distribution.

Task b) (15 Points, ★) Implement an autoregressive flow (e.g. a spline autoregressive or a neural autoregressive flow) and repeat the experiments from Task a) and report your results.

Task c) (5 Points, ★) Briefly compare the two types of flows you implemented in Task a) and Task b). What are their advantages/disadvantages? Did you encounter any of these during you work? Report any other interesting findings you made!