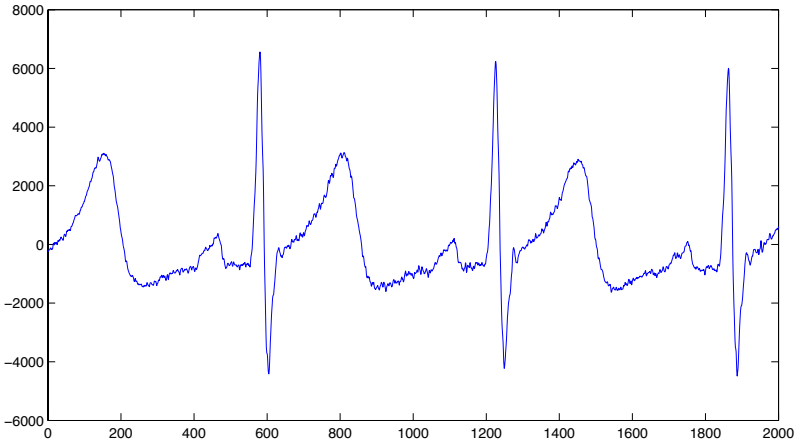
Machine Learning Spring 2016, Homework 4 – Preparatory Microproject for Final Project

In the remainder of this semester I want you to work on a rather advanced final project, which has been designed by Lee Cheng (http://www.masys.url.tw/AU/AU.htm) and is nicely documented at http://www.masys.url.tw/AU/2015SP/BMSD-D/HW/HWfinal-Maternal_Fetal_ECG/ProjectDescription.htm . This project concerns isolating an unborn child's ECG signal from ECG recordings taken from the mother's thorax and abdomen. As you can see from the references list at the end of that webpage, this is a heavily investigated signal processing task. We will apply the (comparatively modest) powers of adaptive linear signal processing to this task.

We will not exactly follow the project definition on that webpage. I will think of a suitably constrained project version in a week or so. However, to get going, this HW #4 asks you to carry out a simple subtask, just in order to get used to these data and to implement a basic LMS algorithm.

The task for this preparatory microproject is to use a linear filter to *predict* a scalar signal (an ECG trace channel taken from the mother's thorax –).

Data: download the Matlab data file `fecg.mat` from that webpage. It contains five ECG signals (`abdomen1`, `abdomen2`, `abdomen3`, `thorax1`, `thorax2`) recorded from a pregnant woman. For this HW we only use `thorax1`. This is a signal that essentially carries only the mother's ECG signal with no component of the child's. The signal is a timeseries of 20,000 points. Here is a plot of the first 2000 points for a visual impression:



Task: Use the LMS algorithm to learn/adapt a filter which predicts this signal. Concretely, if the input $x(n)$ signal is the `thorax1` signal (you may scale / shift it), then the desired output $y(n)$ is the input (in your scaled / shifted form) at time $n + 100$. In other words, the teacher signal is $d(n) = x(n + 100)$.

In more detail, you should set up an LMS based adaptive filter which is driven by $x(n)$, where $n = 1, ..., 20000 – 100 + 1$. At start time, the filter weights should be

initialized to all zeros. Toward the end of this run, the filter output $y(n)$ should come as close to $x(n + 100)$ as you can manage.

Hints: Finding a good value for the adaptation rate λ is one of the keys for good performance. You may also want to experiment with setting λ to greater values initially (for fast tracking), later to lower values (for smaller residual error). Another key for success is finding a good filter length.

Deliverables: This time, for this initial warmup with a short processing time, I don't expect the "lab report" style report that I usually want to see (we'll save that for the full project that comes as next and final homework). For a grade of 2.0, it is enough to submit just your code and a type-set pdf document that reports your LMS settings (filter length, adaptation rate) and displays your results from your favourite run graphically in four plots:

1. A plot that shows the filter output $y(n)$ overlaid on the teacher $d(n)$
2. A plot that shows the squared error $\varepsilon(n)^2$, or rather, for a more instructive rendering, the log10 of $\varepsilon(n)^2$. You will find that this is a very jittery curve. It is instructive to also plot into the same diagram a smoothed version of this curve.
3. A plot showing the development of the norm of weight vectors $\mathbf{w}(n)$.
4. A plot showing the final weight vector.

Please prepare your plots such that the axes labels are readable, and explain the plots in brief caption lines.

Finally, report the *normalized root mean square error* (NRMSE) obtained from the mismatch between the teacher $d(n)$ and the filter output $y(n)$ for all timesteps after $n = 5000$ (discarding the first timesteps because they have a larger error due to the filter weights adapting away from the initialization). The NRMSE between two timeseries $y(n)$ and $d(n)$ is defined by

$$NRMSE(\{d(n)\},\{y(n)\}) = \sqrt{\frac{\frac{1}{L}\sum_{i=1}^{L}(d(i)-y(i))^2}{\mathrm{var}(d)}} = \sqrt{\frac{\frac{1}{L}\sum_{i=1}^{L}(d(i)-y(i))^2}{\sum_{i=1}^{L}(d(i)-\frac{1}{L}\sum_{j=1}^{L}d(j))^2}},$$

where L is the length of the timeseries. In words, the NRMSE is the square root of the mean square error normalized by the variance of the reference. Reporting the NRMSE instead of the MSE is often preferable because the NRMSE is invariant under scalings of the data (note that you may have scaled the ECG recording).

Grading: if you get everything set up technically correct and present the four plots for a single adaptation run – grade 2.0. Better grades or even bonus points for more detailed analyses, systematic efforts to improve prediction accuracy, and insightful discussion (if you look closely at plots, there are many peculiarities that deserve an interpretation).

As usual, send your code and document to x.he@jacobs-university and myself (h.jaeger@jacobs-university.de). **Deadline** is Monday April 25, 23:59 hrs.