KalmanNet For ECG Analysis
Fetal Heart Rate Estimation

Mehdi Bakka Guy Revach

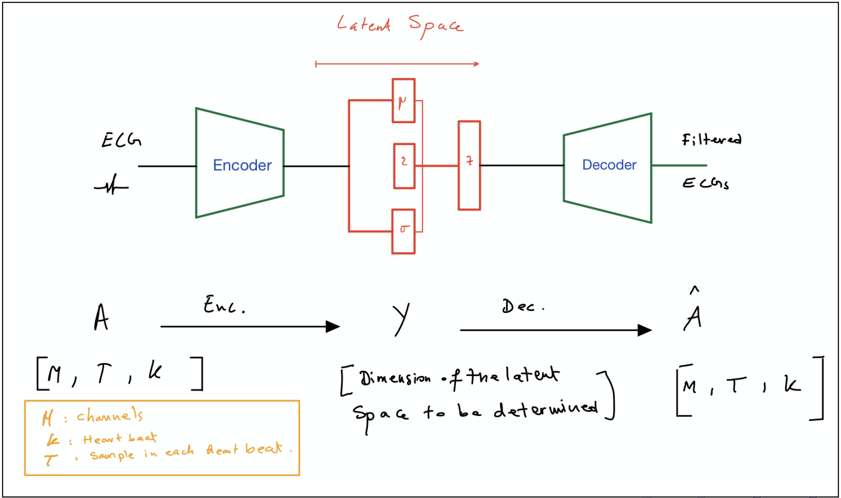
Signal and Information Processing Laboratory (ISI)
ETHZ

December,2021

Agenda

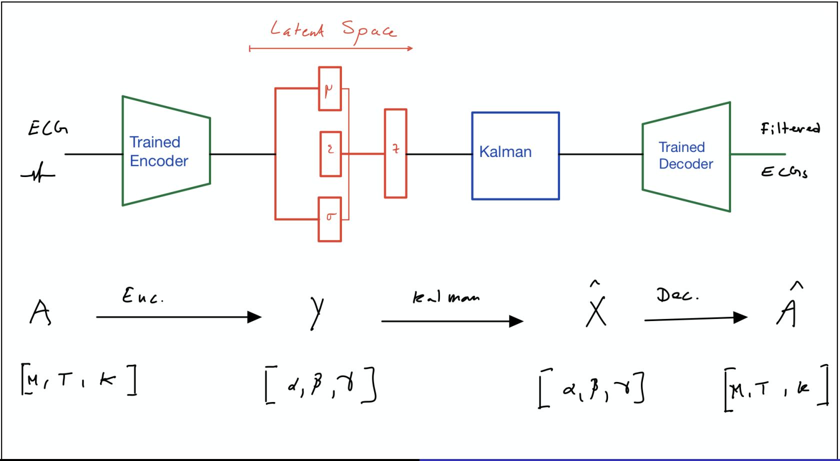
Variational Auto Encoder

- First we will train the VAC using Mean square error plus the KL divergence as a loss function.



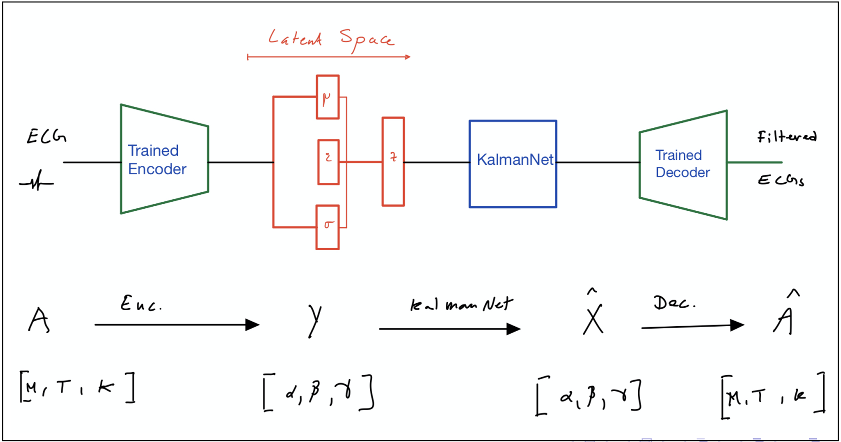
Trained VAC and Kalman Filter

- We will use the previously trained VAC, and add a model based Kalman filter between the encoder and the decoder to do the filtering in the latent space.
- Note that the dimension of the latent space is going to be chosen smaller than the dimension of the original signals. Hence the Kalman filtering would be faster.



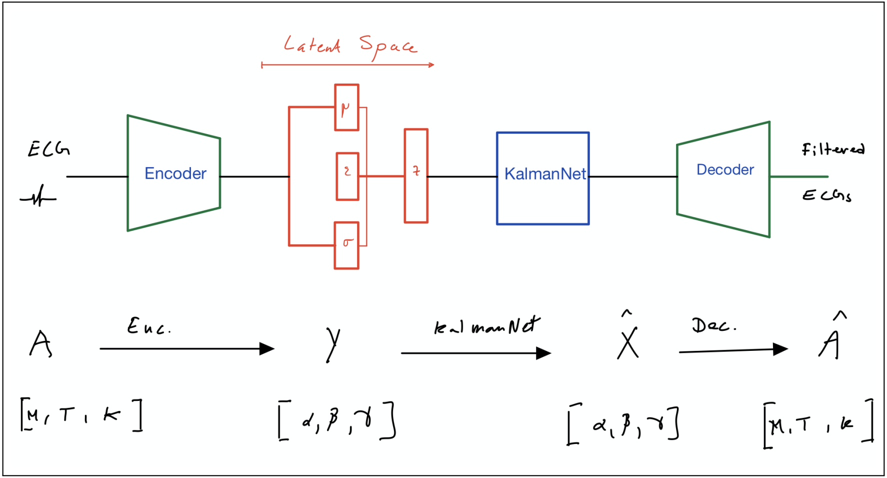
Trained VAC and KalmanNet

- We will use the previously trained VAC, but this time with a KalmanNet filter between the encoder and the decoder to do the filtering in the latent space.
- Note that only the the KalmanNet parameters are going to be trained, since we are using a already trained VAC.



VAC and KalmanNet trained end-to-end.

- One final approach would be to train the model end-to-end (ie VAC parameters and KalmanNet parameters) at a time.



Compare With Rik and al paper.

- Finnally we are going to compare the performance of both ’Trained VAC and KalmanNet’ and the ’VAC and KalmanNet end-to-end’ with the results found in Rik et al paper, using the same evaluation metric as in the paper (ie Normalized MSE).

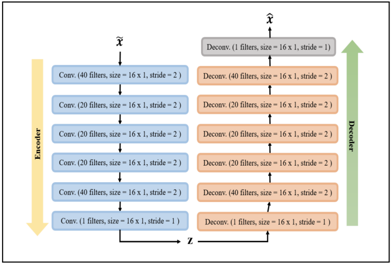
Opening to further research.

- Fetal ECG signals are very noisy signals. Thus a pre-processing step is important, where noise and maternal ECGs are removed , and heartbeats are detected. One question that could be subject of investigation is : Can VAC be used to encode only relevant features from raw fetal ECG signals without any need to pre-processing.

Train a VAC : Architecture and Input data

1. A VAC with the following architecture has been trained has been trained.

- Only 5 measurements from the data sets has been used, more can be used later for better results.
- Every measurement consists of 6 channels with around 80000 samples each ie (6,80000).
- The measurement are concatenated to form a big data of size (6, 5*80000).
- Then Finally each channel is being divided into windows of 1024 samples and fed to the VAC. In fact, the VAC input has as dimension (1024,1).
- The Latent space dimension is chosen to be 100.

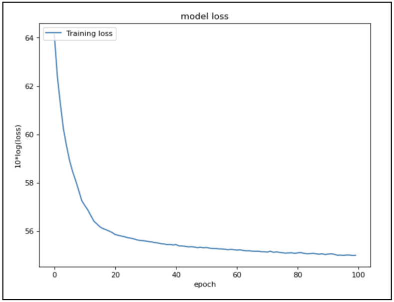


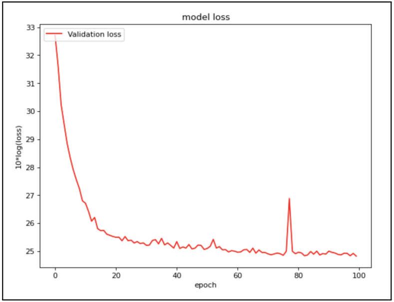
Train a VAC : Results

- Input shape [512,1]. Equivalent of 1 heartbeat in Rik's paper.
- Number of measurement used : 15 measurement file, ie 194112 heartbeat of shape [512,1].
- Data : 50% tarining, 20% Validation, 30% testing.
- Runing time : around 1h45min.
- Number of epochs = 100 epochs.

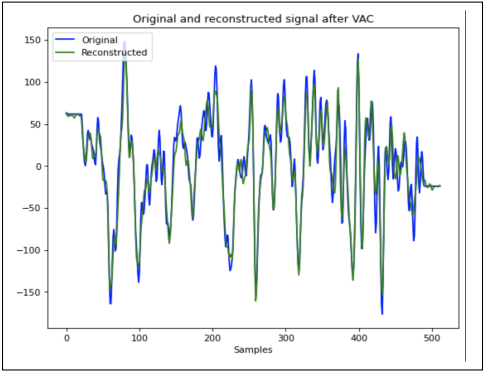
Train a VAC : Training vs Validation loss

- Training loss = KL divergence + MSE.
- Validation loss = MSE.

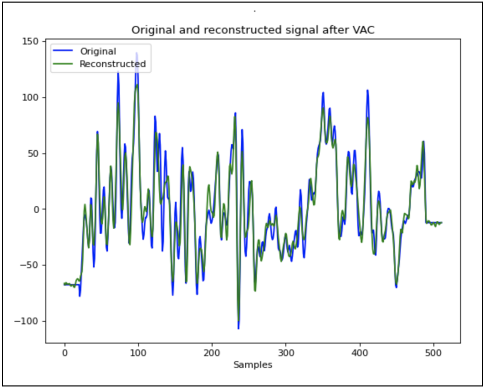




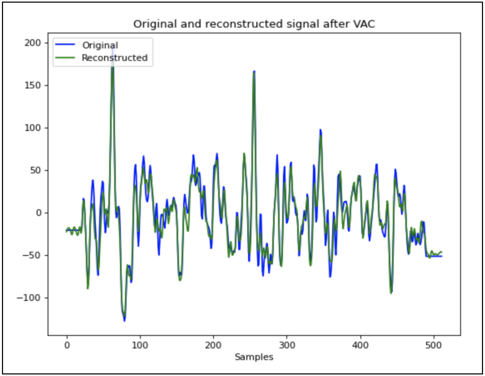
Train a VAC : Reconstruction examples from Testing Data.



Train a VAC : Reconstruction examples from Testing Data.



Train a VAC : Reconstruction examples from Testing Data.



- More figures are in the file: simple_vac_results.

Next steps:

In progress...

- Shape the input data to be as close a possible to the shape in Rik's paper.
- Test the compression limit, ie latent space dimension.
- Use a trained VAC with kalman net.

Summary

#Inputs	Latent Space	Epochs	learning rate	Kernel	Performanc
14999	100	100	0.0001	16x16	16.75
14999	60	100	0.0001	16x16	13.62
14999	25	100	0.0001	16x16	12.24
14999	10	100	0.0001	16x16	09.03
50468	100	100	0.0001	16x16	13.724
50468	50	100	0.0001	16x16	11.0618
50468	50	100	0.0001	16x20	13.095
50468	10	100	0.0001	6x40	?