Toy Model

Hudson Chen

June 13

Dataset-1:

Y = sin(x), with zero-mean, 0.09 variance gaussian noise.

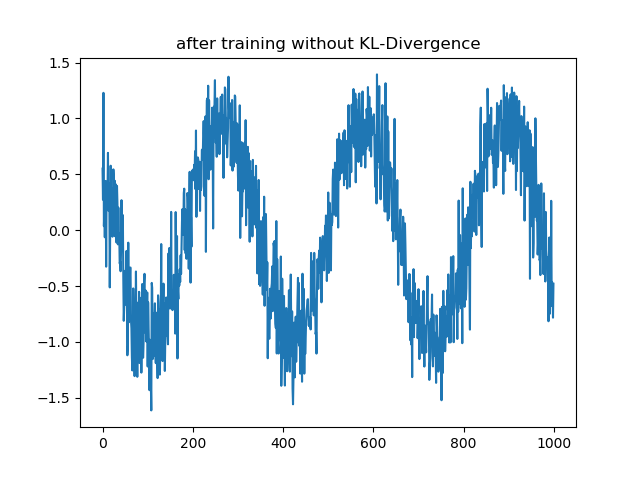
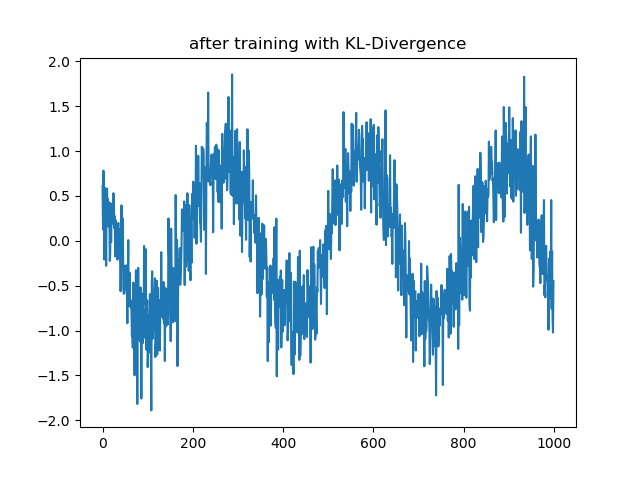
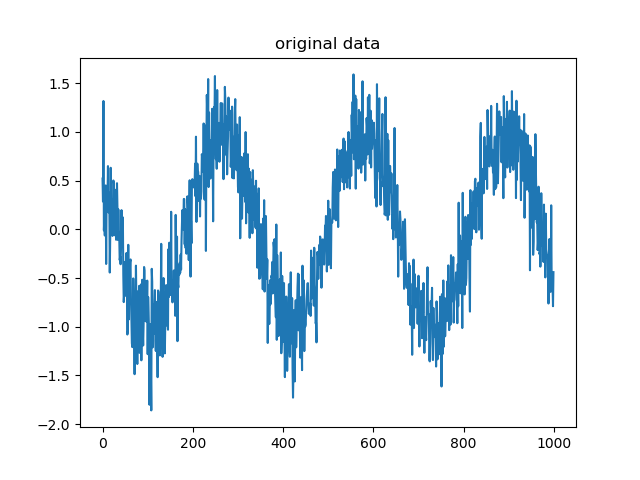
Dataset-2:

Y = 10 \* sin(x), with unit gaussian noise.

Model Architecture:

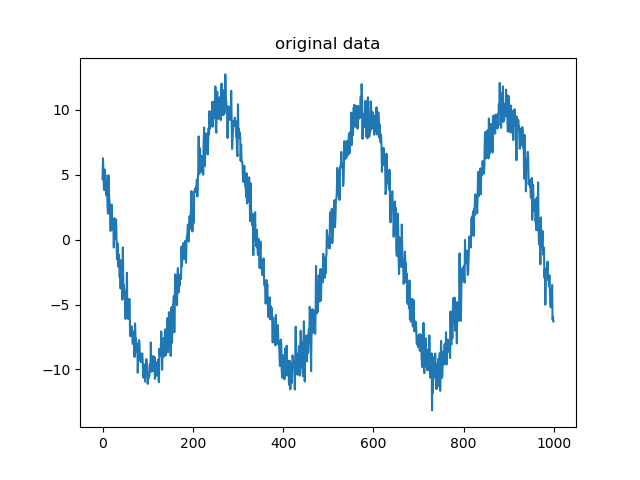
Two layer model with 200 units in the first layer and 100 units in the second layer, using Tanh as a non-linear activation function. The output is mean and log variance. Prior is unit gaussian. The model assumes Gaussian likelihood.

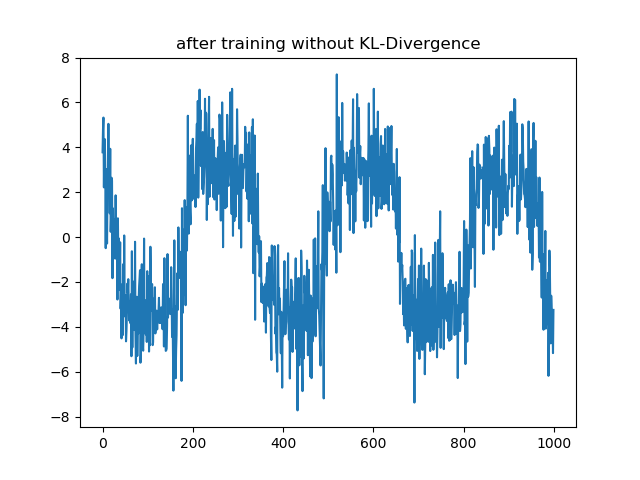
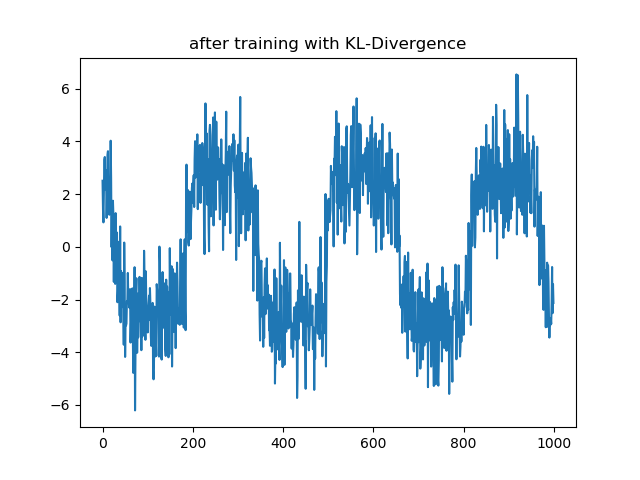
Training with dataset one:



We can see that KL divergence term doesn’t make a big difference. That’s because the original data distribution is close to the prior, so maximizing likelihood has the same effect as minimizing KL divergence.

Training with dataset two:





We can see that KL divergence term here makes a big difference. That’s because the original data distribution is far from the prior, so the KL divergence term pushed the data to move close to unit gaussian, thus deviating from the true distribution.

Question:

1. The model is having bad performance on fitting y = 10 \* sin(x). My guess is that it has something to do with model initialization. So next time, if we are trying to use the pytorch package, it would be better to rescale the input data to range [-1,1].

Code on github: <https://github.com/hudsonchen/toy_BNN>