**Questions Pytorch LSTM – Multivariate Timeseries prediction**

I built a model take takes 2 variables (price, total volume) as input and that outputs a prediction for a specified number of timesteps ahead for both variables.

**Q1**  
How does the loss function consider both variables? Or, put differently, is the loss function minimized based on both variables? Should I expect the inclusion of an extra variable (Volume) to have a positive influence on the prediction of Price?

**Q2**  
What exactly is a pytorch tensor? How is it different from arrays and lists (and, come to speak of it, how is a np array different from a list?) Why do we have all these different data structures/what is the advantage of having the different data structures?

**Q3**How well can I reasonably expect an LSTM to work on this kind of data? Shld it be able to outperform ARIMA?

**Q4**How do I implement one-step ahead forecasting? (This I shld be able to solve relatively simply)

**Q5**what would be the influence of training for more epochs?

**Q6**How does the loss function work? We now use MSELoss. What kind of different loss functions exist? Why is it for some epochs small and *afterwards* bigger again?

**Q7**Should I preprocess my input by taking a MA or smth so that the noise is less strong?

**Q8**  
Can I tell the LSTM that, although it receives multiple input vars, it should focus on optimizing one of them (Price)?

**Q9**What is the influence of the hidden layer size? (now 100)