

Exercise for Introduction to Machine Learning

Session 2

Advanced materials processing with intelligent systems

Explain intuitively why a thinner but deeper network (i.e., many layers, each with few parameters) should perform better than a shallower and thicker one (i.e., with few layers, but many parameters per layer) — given that they have the same number of parameters.

Normalization is the process of changing the signal's range to a predefined one (i.e., $[0,1]$, or $[-1,1]$).

Mark the correct answer(s):

1. It should be avoided as it "destroys" the information about the amplitude
2. It helps gradient descent to converge faster
3. Allows better handling of multi-modal signals
4. It is not necessary if we use an optimization technique that adapts the learning rate per parameter.

What downsides can normalization bring? Conversely, what are the advantages the standardization brings over normalization?

Normalization is sensitive to outliers. How can you make it more robust?

The learning rate is probably the most critical hyper-parameters for most ML algorithms. Intuitively, a larger learning rate should be preferred to a smaller one for faster convergence. Nonetheless, large learning rates can make the optimization unstable and even leading to divergence. How can you combine the advantages of a large learning rate (fast convergence) with the ones of a smaller learning rate (stability)?

A very popular optimization algorithm doesn't consider the magnitude of the gradient but only its sign. Think about how you can develop an optimization algorithm that doesn't consider the magnitude of the gradient but only its sign. What are the advantages?

As seen in class, the primary outcome of Deep Learning is to "automatically" learn to extract unconventional features from the raw data, which allows developing an end-to-end training approach (from raw data to outcomes). When can this technique be helpful, and when does it make sense to manually engineer some features to feed the model instead of using the raw data as input?