Questions and answers - Lukas Heinrich Lecture 3

The following questions were submitted through Google Form. Some may have been answered in the Q&A session already. Nevertheless, we request our lecturers to provide written answers here for the benefit of those who could not attend that session. Thank you!

Slide 41. I have a follow-up question about the VAE and normalizing flow. I think by letting the latent space dimension of a VAE to be the same as the input space, the VAE can map the complex distributions shown on slide 41 to Gaussian, which means VAE can achieve the same thing as normalizing flow. (Please correct me if I'm wrong, and if I'm wrong, could you let me know why VAE cannot achieve it?). And VAE would be less expensive than the normalizing flow, so I'm wondering in what scenario that we cannot replace normalizing flow by VAE. I really don't see the point of imposing an invertible constraint on normalizing flow…it just makes the mapping more expensive to train…

LH: the VAE does not have invertible/bijective functions (not least because the dimensionality is changing) so they don't do exactly the same job as a NF.

The VAE loss is a bound on log p(x) while a NF gives you an exact expression for log p(x), so whenever you need the latter you can't use a VAE

Slide 31. It seems that variational autoencoders work better and are more usable than the standard autoencoder. Is the standard autoencoder used in any applications? Is there an advantage using standard autoencoder or limitation on using variational autoencoder?

LH: For pure compression, where you don't care about the distribution (i.e. for generative modelling) in the latent space, a AE might be sufficient

Slide not specified. I'm recently working in an ill-posed inverse problem (self-supervised learning with CNN), and I was surprised to find that the Adam gives a better result than second-order methods such as inexact gaussian-newton method--although the later method converges faster. I think it might partly because of the implicit regularization effect of gradient-based method (https://arxiv.org/abs/2210.07082). Do you think the structure of CNN itself would introduce an implicit regularization effect as well? I haven't

seen any literature on this question yet, but if you have any recommendations on this topic, it will be very much appreciated.

End of talk. I'm a bit confused by your explanation for double descent. Is the point that a larger model has a larger space of possible functions it can represent, or that it is easier to optimize, or something else? Isn't a simple fully connected network already a universal function approximator, in which case a larger model wouldn't have a larger "hypothesis space" if I understand the term correctly.

LH: a simple net is only a UFA in the limit of arbitrary many neurons (see Lecture 1&2)
any finite network is also a finite function approximator