LAURA THESING, LMU Munich

Which networks can be learned by an algorithm? – Expressivity meets Turing in Deep Learning

Deep learning with neural networks celebrates a big success in a wide range of applications, with performance exceeding what many thought would be possible. However, we are also aware of the inherent instabilities to small perturbations of the input. Methods to show these instabilities with adversarial attacks were first introduced for image classification tasks and were later extended to malware detection, sound detection, text analysis, medical fraud, and inverse problems. The instabilities might be surprising when considering the results about the expressivity of neural networks that show that the class is rich enough for continuous network approximations. Already the universal approximation theorem shows that all continuous functions can be approximated with neural networks. Hence given a continuous problem it can be approximated by stable neural networks. Therefore, the question arises: Why are the learned network approximations typically unstable when stable neural networks exist? One aspect that is often overlooked is the computability of neural networks. Indeed, there are examples where stable neural networks exist, but no algorithm can compute them. This fact leads to the questions about which networks are computable functions and for which networks do we have an algorithm that can map from point samples to the approximation. We will elaborate on these questions in this talk.