Deep Learning and Neural Networks: an infinite dimensional perspective

Luca Galimberti (NTNU)

Abstract

The universal approximation theorem shows that any continuous function from $\mathbb{R}^n$ to $\mathbb{R}$ can be approximated arbitrary well with a one layer neural network. More precisely, for fixed continuous function $\sigma : \mathbb{R} \to \mathbb{R}$ and $a \in \mathbb{R}^n, \ell, b \in \mathbb{R}$, a *neuron* is a function $\mathcal{N}_{\ell,a,b} \in C(\mathbb{R}^n; \mathbb{R})$ defined by $x \mapsto \ell\sigma(a^\top x + b)$. The universal approximation theorem states conditions on the *activation function* σ such that the linear space of functions generated by the neurons

$$\mathfrak{N}(\sigma) := \operatorname{span}\{\mathcal{N}_{\ell,a,b}; \ell, b \in \mathbb{R}, a \in \mathbb{R}^n\}$$

is dense with respect to the topology of uniform convergence on compacts.

In this talk we present some novel results obtained by Fred Espen Benth (UiO), Nils Detering (University of California Santa Barbara) and myself on abstract neural networks and deep learning. More precisely, we derive an approximation result for continuous functions from a suitable class of infinite dimensional vector spaces into $\mathbb{R}$, which generalizes the universal approximation theorem above.

While such an approximation result might be of interest in its own, from a practical perspective it is not clear how the functions $\mathcal{N}_{\ell,A,b}$, which now involve infinite dimensional quantities, can actually be programmed. We therefore address the question of approximating the maps $\mathcal{N}_{\ell,A,b}$ by finite dimensional, easy to calculate quantities. The resulting neural network has an architecture similar to classical neural networks, with the exception that the activation function is now multidimensional. Also in this case, the approximating function is easy to implement and allows for fast computation and fitting. Finally, we also derive the approximation property for deep neural networks with a given fixed number of layers.

Few applications geared toward derivative pricing and numerical solutions of parabolic partial differential equations will be outlined.