Manifold Approximation in Hilbert spaces with Compositional Polynomial Networks - Bensalah 2025

Ismail • 5 Aug 2026

We would like to approximate a high dimensional set M in a Hilbert space X by some lower dimensional manifold M_n of dimension n, using an auto-encoder.

We start with an illustrative example. Consider the KdV equation

$$\frac{\partial u}{\partial t} + 4u\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0 \qquad \text{in } [-\pi, \pi] \times [0, 1]$$

which descibes the propagation of a solition in one dimensional space with periodic boudnary conditions. Let $M = \{u(t) : t \in [0, 1]\}$ be the set we wish to approximate. In practice the equation is solved on a space=time grid, meaning we take for example a 256-point discretization of the space domain and a 5000-point discretization of the time domain (+1=5001 for the initial condition) to be our solution snapshots, and we wish to approximate this high dimensional set of size 256×5001. For each point in time, we need to figure out 256 values for u, meaning the solution depends on 256 parameters. The goal is to find an auto-encoder that gives a lower dimensional estimate for u, meaning with much fewer parameters than 256 that encapsulates all the necessary information about the 256 points up to some prescribed tolerance with respect to some error measure - usually L^2/mean-squared or $\sup$/worst-case.

The idea could be explained by the following.

One basic approach is to use a purely linear approximation. Meaning we approximate M (or the discretized set) by a lower dimensional linear space of a large enough dimension such that all the values in M are well approximated after going through the auto-encoder. That is, in a Hilbert space, our auto-encoder is just the linear map P_{M_n} which projects onto the optimal lower-dimensional subspace M_n of dimension n. In practice, this is achieved through principal component analysis (PCA) performed on the matrix discretizing the set M and taking as a basis for the linear subspace the first n principal components associated with the largest n singular values. The problem with this approach is that for highly-nonlinear manifolds, the number of dimensions needed to a get a high precision is often high, and sometimes infeasible computationaly. This is in particular the case for problems dominated by transport (e.g. KdV).

Since using a purely linear approach is not a great idea, we seek to nonlinearize. For this purpose we first acknowledge that the auto-encoder is a two step operation, where the values in M first are encoded in a lower dimensional space $\mathbb{R}^n$ by an encoder E. Then a decoder D maps the set of n parameters associated with each value $E(u)$ back to the Hilbert space, hoping to get a good enough approximation $D(E(u))$ of u. The auto-encoder is associated with the pair (E, D). In the case of a linear approximation, both E and D are linear. If we wish to nonlinearize, we are faced with three choices: we only use a nonlinear encoder, we only use a nonlinear decoder, or we set both to be nonlinear. If we only consider linear decoders, then the image of the decoder will be a linear space in X, yielding the same error scenario as in the fully-linear. If we consider a nonlinear endcoder, then imposing stability on the approximation operator yields a difficult-to-implement optimal nonlinear encoder - associated with NP-hard optimization problems. A linear encoder is nevertheless easy to understand: it is simply projecting onto the n (optimal) basis elements of the parameter space. This leaves us with the choice of a linear encoder and a nonlinear decoder. The encoder gives us n parameters, and we use the optimal basis elements to span an n dimensional linear space X_n inside X, and the decoder adds a nonlinear "correction" using the first n parameters so that the final approximation set, $Mn = \{D(a) : a \in Im(E)\}$, is an n-dimensional manifold living in a potentially higher dimensional linear space $X_N, N \geq n$, depending on how many nonlinear corrections are added. The final expression for our decoder would look like:

$$D(a) = \sum_{i=1}^{n} a_i \varphi_i + \sum_{i=n+1}^{N} g_i(a) \varphi_i$$

where the g_i are nonlinear functions of the parameter vector.

The question now is how do we decide on N? Where do we get these nice basis elements? How many should we consider for the encoder and how many for the decoder?

To start, N should be determined such that the projection error onto X_N is controlled in some way. That is, we pick the first N eigenvalues of the operator associated with PCA such that $\sqrt{\sum_{i>N} \lambda_i} \leq C(\epsilon)$ for some error threshold $C(\epsilon)$ that guarantees the desired precision. Our nice basis will then be the corresponding N principal components. Here is where we should pay attention: choosing the first n principal components as the basis for the encoder will NOT give the optimal approximation - this was demonestrated. One must instead opt for an adaptive approach, choosing those components whose coefficients won't be approximated well by our nonlinear approximations - in that case we just use

the coefficient that is the constant projection coefficient (i.e. consider that direction in linear approximation). Thus, there will be a learning phase, performed on a training set where we not only construct g_i but also check for which principal component directions should be considered for the nonliear part and which for the linear part. In our earlier example, we can take the values $t_i [0 , 0.2] $ to learn the indiced of directions to be considered in the linear set and the coefficients g_i.

Now comes the contribution of the paper: we wish to choose good functions g_i. Our nice choice comes from an observation.

> A coefficient $a_i(u)$ for $i > n$ may have a highly nonlinear relation with the first n coefficients $a = E(u)$ but a much smoother relation when expressed in terms of a and additional coefficients $a_j(u)$ with $n < j < i$. This observation suggests the following compositional structure of the decoder's functions $g_i(a) = f_i(a, (g_j(a))_{n<j\leq n_i})$.

(Bensalah et al. 2025)

The nonlinear functions proposed, as suggested by the titles, are polynomials. There are different options for the space from which these polynomials could come, but ultimetly the final polynomial is that one that minimizes some error from the projections coefficients (on the sample data).

Therefore the coefficients will be given by $g_i(a) = f_i((g_j(a))_{j\in S_i})$, where S_i is the set of indices associated with g_i that we constructed adaptively.

That's all we need to know.

The method is compared to other sota methods and outperforms them all in error and numeber of parameters. For our KdV equation, with degree 5 polynomials form an appropriate polynomial space with only 43 PCA components, an error of 10^{-5} is achieved using only two parameters!

Bensalah, Antoine, Anthony Nouy, and Joel Soffo. 2025. *Nonlinear Manifold Approximation Using Compositional Polynomial Networks*. https://arxiv.org/abs/2502.05088.