**Continuous Kernel Convolution For Sequential Data**

**Introduction**

CNN and RNN have suffered with limitations in the past with RNN’s main limitation being vanishing descent. While CNN overcomes the disappearing descent issue with back propagation through time, it has struggled with sequences of unknown size. Hence Continuous Kernel Convolution approach is proposed which overcomes the challenges faced by RNN and CNN. Vanishing gradient problem of recurrent networks can be solved by 3 methods. The first method uses gating mechanism which involves preserving information from the far past and enhance gradient flow by updating their hidden representation only when update gates are activated. The second method uses unitary recurrent units where unitary eigen values are used in recurrent connections. The third method avoids recurrent connections and back-propagation through time. Discrete convolutions cannot handle irregularly sampled data properly. This paper gives us a view on the applications for which continuous kernels are advantageous.

**Convolution operation**

Here a few representations [n] the set {0, 1, 2, . . . , n}are used.Bold capital and lowercase letters depict vectors and matrices, sub-indices are used to index vectors,and parentheses are used for time indexing. In Centered and Causal Convolutions the convolution is

defined as: (x ∗ ψ)(t) = ?=1 ?? σ ∫ R xc(τ )ψc(t − τ ) dτ The convolution is effectively performed between the input signal described as a sequence of finite length and a convolutional kernel.The convolutional kernel is commonly centered around the point of calculation t which is solved by providing a causal formulation to the convolution.

In Convolutions with Discrete Kernels the memory horizon NK must be defined a priori. In order to alleviate the limitations mentioned previously, it has been proposed to dilate the sequence parameterizing the convolutional kernel K by a factor η.

**Continuous Kernel Convolution**

Its seen in the paper,they have been able to construct global memory horizons without modifying the structure of the network or adding more parameters.CKConvs are able to handle irregularly sampled and partially observed data natively. To this end, it is sufficient to sample MLPψ at positions for which the input signal is known and perform the convolution operation with the sampled kernel. CKCNNs can be deployed at sampling rates different than those seen during training, and it can be trained on data with varying temporal resolutions. It is seen that linear recurrent units can be described as a CKConv with a particular family of convolutional kernels i.e, exponential functions.

Since exponentially growing gradients lead to divergence, the eigenvalues for converging architectures are often smaller than 1. This explains why the effective memory horizon of recurrent networks is so small.Linear recurrent units are a convolution between the input and a very specific class of convolutional kernels i.e, exponential functions.

The convolutional kernel MLPψ is parameterized by a conventional L-layer neural network. Haing a closer look at their approach it can be thought of as providing implicit neural representations to the unknown convolutional kernels ψ of a conventional convolutional

architecture. The function being implicitly represented, is uniformly distributed. Consequently,it is seen conventional initialization techniques lead to poor performance. The expressiveness of such an approximation is determined by the number of knots the basis provides, i.e., places where a non-linearity bends the space. Naturally, the better the placing of these knots at initialization, the faster the approximation may converge. For a spatially uniform distributed input, the knots should be uniformly distributed as well.

It is observed that finding an initialization with an exponential number of knots is a cumbersome and unstable procedure so to cater to that issue they have utilized an initialization procedure with which the total number of knots is equal to the number of neurons of the network.

It is observed ReLU networks show large difficulties in representing very nonlinear and non-smooth functions.Fourier transform states that any integrable function can be described as a linear combination of an infinite basis of phase-shifted sinusoidal functions.Intuitively,approximations via Sine networks can be seen in terms of an exponentially large Fourier-like basis. The exponential growth combined with the periodicity of sine allows for astonishingly good approximations,the more terms in a Fourier transform, the better the approximation becomes.

**Experiment**

The approach is validated across a large variety of tasks and against a large variety of existing models. They have parameterized all convolutional kernels as a 3-layered MLP with Sine non linearities. They have used the convolution theorem to speed up

convolution operations in the networks with the Fourier transform.

In Stress experiments, it is validated that the memory horizon of shallow CKCNNs is not restricted by architectural choices. It is seen that a shallow CKCNN solves both problems for all sequence lengths considered without structural modifications.

In Discrete sequence experiments, they validated the applicability of CKConvs for discrete sequence modeling tasks: sMNIST, pMNIST and sCIFAR10. It is observed that shallow CKCNNs outperform strong recurrent and convolutional models.A small CKCNN obtains state-of-the-art on sMNIST and wider CKCNN also increases the results on ther datasets.

In time-series modeling experiments, they have tried evaluating CKCNN on time series data and on longterm dependencies. It is seen that it performs well and has the ability to handle very long term dependencies.

In testing at different sampling rates experiments, they have considered data that is trained at different sampling rates. It is seen that the performance of CKCNNs remains relatively stable even for large sampling rate fluctuations and it even outperforms HiPPO.

At last they even explored the applicability of CKCNNs for irregularly-sampled data. It exhibits stable performance but does cross NCDEs.

**Conclusion**

Futher experiments indicate that the models do not benefit from larger depth, which suggests that CKCNNs do not rely on very deep features.MLPs parameterizing spatial functions should use sine nonlinearities. The kernels often contain frequency

components higher than the resolution of the grid used during training. CKCNNs can be executed in parallel, and thus can be much faster than recurrent networks.