REU 2025 Kernel-KAN Model Architecture Proposal

Nash Rickert

May 30, 2025

1 Summary and Motivation

This paper proposes investigating a model architecture for classifying hyperspectral data that combines preprocessing data with a kernel before passing the transformed data through a Kolmogorov-Arnold Network (KAN). CNN architectures have proven to be effective in handling existing hyperspectral datasets but they are large and cumbersome, which makes them difficult to apply in an FPGA setting. On the other hand, KANs have shown to have potential for similar classification tasks despite doing so without any spatial encoding, but they lag behind the CNN in accuracy. By learning nonlinear functions directly instead of weights to be applied to fixed activations, a KAN is likely much more suitable for the hardware setting of real-time classification due to smaller model size. By combining a kernel frontend to a KAN, we hope that the spatial awareness will boost the effectiveness of the KAN while retaining the size and performance benefits of that particular model architecture. The KAN has also been shown to work better with lower dimension data, which the kernel passthrough naturally helps achieve. Notably, one of the benefits of a KAN was the fact that it could perform truly real-time classification of hyperspectral data due to doing classification by pixel. Implementing kernel preprocessing eliminates this benefit and it will need to be judged whether potential increases in classification accuracy is a worthwhile tradeoff.

2 Expectations

A large CNN was able to achieve roughly 99.5% accuracy on the Indian Pines dataset. A KAN trained by Dirk Kaiser as a part of our research project was able to achieve roughly 80% accuracy on the same dataset. Our hope and expectation is that a small kernel frontend will boost the accuracy of the KAN network to be comparable to that of the CNN while retaining the size benefits of the KAN model architecture.

3 Utility

By learning nonlinear functions directly instead of weights for fixed activations, a KAN is potentially able to approximate functions with similar efficacy to a traditional multilayer-perceptron (MLP) with far fewer nodes in the model. Additionally, the potential complexity of such functions is not necessarily problematic for us because we can implement the functions as lookup tables inside of the FPGAs. The KAN also has interpretability benefits, as the specific functions it learns might tell us something about the actual problem we are trying to solve. For example, our KAN might discover the wavelengths that are important for certain classification problems. Unlike in a traditional MLP, these wavelengths might be discoverable to us by investigating the model more closely.

4 Additional Considerations

There are several additional questions to think about when considering using a KAN for our classification. Perhaps the most important is the ability to prune the learned functions to shrink our model even further. If many of the learned functions behave similarly, then perhaps we can reduce our model to several basis functions which approximate the behavior of the others. In this case, we will only need to program these basis functions into our FPGAs, and other functions can redirect into them. Possible methods to do this could be k-means clustering, PCA, or the built-in pruning of the KAN libary (although the latter does not seem particularly effective). It will also be important to understand the meaning and importance of polynomial order, smoothness at polynomial connections, and anchor point complexity when building our KAN as these hyperparameters might affect the overall effectiveness and interpretatibility of our mode.