Prediciting Calabi-Yau Threefold Topology from the Kreuzer-Starke List of 4D Reflexive Polytopes

Ben Campbell[a], Per Berglund[a], Vishnu Jejjala[b]

[a]University of New Hampshire [b]University of the Witwatersrand, Johannesburg, South Africa

String Theory

- String theory is a physical framework that unifies quantum mechanics with general relativity, i.e. a theory of quantum gravity.
- There are different formulations of the theory, but each requires at least a $(9+1)$-dimensional spacetime stucture to be mathematically consistent. In order for string theory to agree with the experimentally verified (3+1)-dimensional spacetime, 6 of the 9 spactial dimensions must be "compactified." These compactified dimensions take the form of a Calabi-Yau 3-fold, a three complex dimensional manifold.
- It has been shown that there is a lower bound of 10^{500} different compactifications.

Machine Learning

- Can be used to classify data according to known or unknown patterns
- Dataset is split into two groups:
  - A training set to calculate gradients and adjust internal parameters accordingly to minimize a loss function.
  - A testing set to generate predictions and access accuracy.
- There are many different machine learning paradigms. For a classification problem, we use a fully-connected network with rectified non-linearity.
  - Consists of N vectors, called hidden layers, which are chained together via linear transformations and non-linear functions:

$$\rho_n(W^n a^{n-1} + b^n), \quad \text{with } \rho_n \text{ some non-linear function.}$$

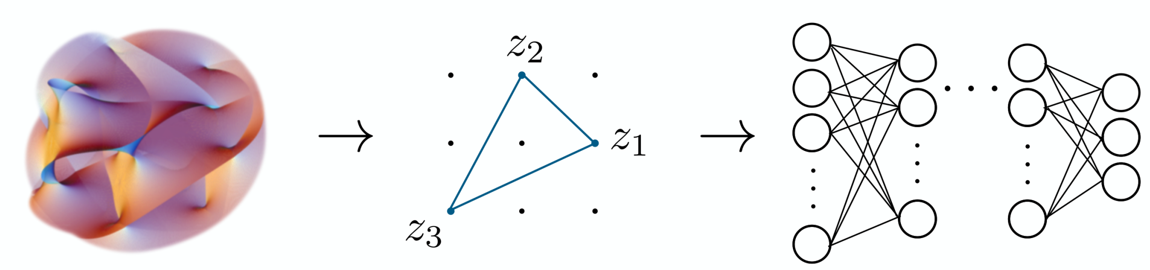
  - An input layer a^0 feeds into the first hidden layer (a^1) and the last hidden layer (a^N) feeds into an output layer a^{N+1}, where predictions are read off from.
  - It is often useful for the output layer to represent a probability distribution, in which case the output layer is normalized.

$$\implies \text{Pred}(z_{\text{input}}) = \sigma(\rho(W^n a^{n-1} + b^n)), \quad \text{with } a^0 = z_{\text{input}}$$

where $\sigma(z_i) = \frac{e^{z_i}}{\sum_{l=1}^{k} e^{z_i}}$, $\rho(z_i) = \max(0, z_i)$, and $a^{n-1} = \rho(W^{n-1}a^{n-2} + b^{n-1})$

Compactification as a Machine Learning Problem

A Calabi-Yau n-fold can be uniquely mapped to an $(n+1)$-dimensional reflexive polytope Δ^* representing a particular type of compact $(n+1)$-dimensional manifold known as a toric variety:



$$(z_1)^3 + (z_2)^3 + (z_3)^3 = 0$$

$$\Rightarrow (z_1, z_2, z_3) \sim (\lambda z_1, \lambda z_2, \lambda z_3)$$

Our model:

- Three models are each trained to learn a different topological property of the Calabi-Yau 3-fold corresponding to the input polytope: the Euler number χ, and the Hodge numbers $h^{1,1}$ and $h^{2,1}$.
  - Topologically speaking, $h^{1,1}$ and $h^{2,1}$ are invariants which count the different types of holes in the Calabi-Yau 3-fold, while χ counts the difference between these holes. That is, $\chi = 2(h^{1,1} - h^{2,1})$.
- Models have 5 - 10 hidden layers with 362 - 1000 neurons per layer.
- 136 input neurons and 480 - 961 output neurons, depending on ontology.

4-dimensional Reflexive Polytopes

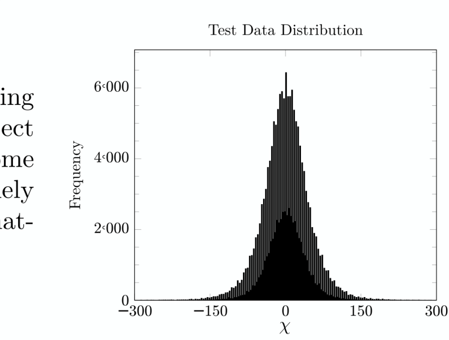
The phenomenologically relevant case are the 4D polytopes which represent the Calabi-Yau 3-folds of string theory (or, the curled up 6 real space dimensions).

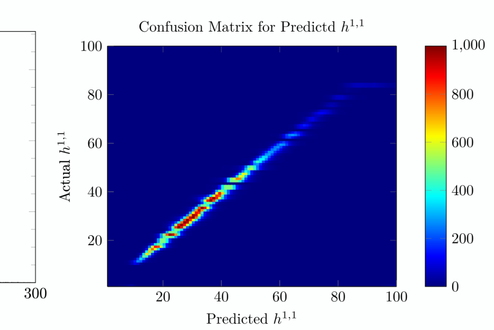
- There are 473,800,776 reflexive polytopes in 4D
- Many share the same topological features
- ~150 GB of raw vertex data (parsed and compressed to ~15 GB)
- Vertex data is used as input to the model, along with other polytope features (e.g. number of vertices, number of lattice points, etc)

Results

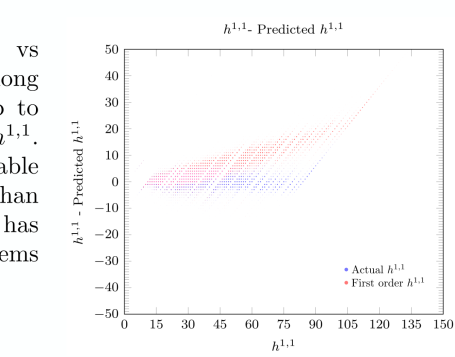
Three models are trained to classify the polytopes by the Euler number χ and the Hodge numbers $h^{1,1}$ and $h^{2,1}$ of the corresponding Calabi-Yau 3-fold. All the models preform roughly the same at an accuracy of 35% - 40%.

The distribution of testing data is plotted with respect to χ which reveals some interesting features, namely the two seperate, alternating distributions.





The error in actual $h^{1,1}$ vs predicted $h^{1,1}$ is plotted along with the error of $h^{1,1}$ up to first order vs predicted $h^{1,1}$. This shows the model is able to predict $h^{1,1}$ to better than first order (second order has yet to be looked at, but seems promising).

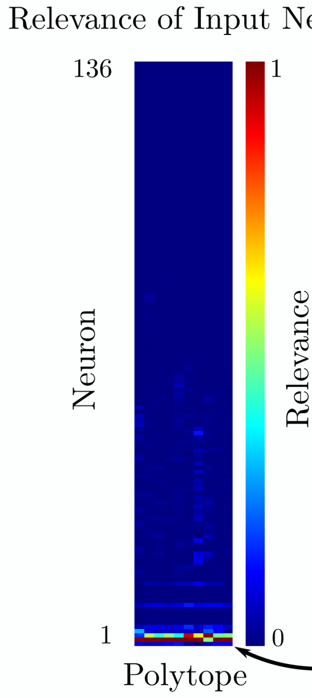


A confusion matrix of the prediction of $h^{1,1}$ is plotted which shows actual vs predicted $h^{1,1}$ with color indicating model frequency. A model with 100% accuracy would have a straight line with slope unity. Our model shows an overall decent match to the perfect case, with highest accuracy occuring in the $h^{1,1} = 20$ to $h^{1,1} = 40$ range. Color off of the diagonal is indicative of false positives and false negatives.

Ontology	Layers	Neuron Density	Output Size	Accuracy	Actual − Predicted
χ	5	1000	961	36.45%	-0.33631 ± 6.63236
$h^{1,1}$	3	362	480	40.62%	0.58364 ± 6.63748
$h^{2,1}$	3	362	480	34.54%	0.83766 ± 7.66044

Ontology	Leading Order - Predicted
χ	-0.343800 ± 12.79859
$h^{1,1}$	5.075965 ± 9.70828
$h^{2,1}$	5.327715 ± 10.722364

Relevance of Input Neurons



We measure the relevancy of each input by propogating the output back through the network accoring to the rule

$$R_j^{m-1} = \sum_k \frac{W_{jk}^m a_j^{m-1} + \frac{1}{n_j} b_k^m}{\sum_{l=1}^{k} W_{lk}^m a_l^{m-1} + b_k^m} R_k^m$$

where R_k^m denotes the value of the k^{th} neuron in the m^{th} layer, and n_j denotes the number of neurons (neuron density) in the j^{th} layer.

The most relevant input neurons for these 10 polytopes are only the first few. This is where the feature data about the polytopes is kept, in particular, the most relevant neuron is the number of points in the dual polytope while the vertex data is barely being used. In the future, we will probably begin focusing on the use of feature data like the number of points exclusively.

References

[1] M. Kreuzer, H. Skarke. Classification of Reflexive Polyhedra in Three Dimensions. (1998).
[2] Bao, Jiakang et al. Lectures on the Calabi-Yau Landscape. (2020).
[3] Mehta, Pankaj et al. A high-bias, low-variance introduction to Machine Learning for physicists. Physics
[4] He, Yang-Hui. The Calabi-Yau Landscape: from Geometry, to Physics, to Machine-Learning. (2018).
[5] Douglas, Michael R. Statistics of string vacua. (2003).
[6] Jessica Craven, Vishnu Jejjala, and Arjun Kar.Disentangling a Deep Learned VolumeFormula. 2020. arXi
[7] Bouchard, Vincent. Toric Geometry and String Theory. 2006. arXiv:hep-th/0609123

University of New Hampshire