Generating Reflexive Polytopes via Sequence Modeling

Bernt Ivar Utstøl Nødland
Norwegian Defence Research Establishment (FFI)
Instituttveien 20, 2007 Kjeller
Norway
bernt-ivar-utstol.nodland@ffi.no

Abstract

We train neural network sequence models to generate smooth reflexive lattice polytopes. We demonstrate that they can generate mathematical objects satisfying various geometric properties. We use the completeness of our datasets to give evidence that the models have learned some underlying structure of the data.

1 Introduction

Recently the capabilites of deep generative models has increased significantly. Inspired by these improvements we train sequence models implemented as transformers and recurrent neural networks to generate lattice polytopes. Each polytope in our training data is defined by a sequence of integers. The polytopes all share several global geometric properties that we check to evaluate the quality of the generated samples. We show that the sequence models can successfully generate lattice polytopes satisfying these properties.

The datasets are complete in the sense that they include every single example (up to changes of coordinate) satisfying the required properties, thus we are in the special situation of being able to train a generative model on the set of all true examples. This enables us to study to what extent the models are memorizing the training data or are learning some underlying structure of the data: For generated samples satisfying the required properties we can check whether the data defining the sample is identical to its representation in the training data, or if it is a coordinate change of it, although this is computationally expensive to do at scale. We show evidence that the models are not merely memorizing examples, but are learning some underlying structure.

One motivation for training generative models to generate polytopes was to see whether we could generate a counterexample to Oda's conjecture which states that all smooth polytopes are normal [Oda08]. We did not manage to find such a counterexample. Another application of training generative models on mathematical datasets is that it gives a way of automatically evaluating generative models by checking whether generated samples satisfy the chosen mathematical properties.

2 Data

The dataset we use are that of all smooth reflexive convex lattice polytopes of dimension 7 and of dimension 8 (there are only finitely many in each dimension), which is a dataset that already exists in the mathematical literature [Øb07].

Each polytope in the dataset satisfy the properties of being a lattice polytope, compact, reflexive, smooth and normal. These properties are all properties of the global polytope and are not checkable merely from local structure, hence we hypothesize it will be challenging to generate objects satisfying them for deep learning models. Since the dataset is complete, it is also true that any generated example satisfying all these properties will be in the training data, up to change of coordinates. We represent

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on Math-AI.

$$
\begin{array}{rl}
0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1 & x_7 \geq -1 \\
0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ -1 & -x_7 \geq -1 \\
-1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0 & -x_1 \geq -1 \\
0\ -1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0 & -x_2 \geq -1 \\
0\ \ 0\ -1\ \ 0\ \ 0\ \ 0\ \ 0 & -x_3 \geq -1 \\
0\ \ 0\ \ 0\ -1\ \ 0\ \ 0\ \ 0 & -x_4 \geq -1 \\
0\ \ 0\ \ 0\ \ 0\ -1\ \ 0\ \ 0 & -x_5 \geq -1 \\
0\ \ 0\ \ 0\ \ 0\ \ 0\ -1\ \ 0 & -x_6 \geq -1 \\
1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 6 & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + 6x_7 \geq -1
\end{array}
$$

Figure 1: An example from the 7d dataset in the hyperplane representation. A row of numbers from the left corresponds to coefficients of one defining inequality in $\mathbb{R}^7$ with coordinates $x_1, x_2, \cdots, x_7$.

a polytope by the equations of its defining hyperplanes and check the properties: compact, lattice polyhedron, smooth, and normal (reflexive is, given this choice of data representation, equivalent to compact lattice polyhedron).

Alternatively we could also represent a polytope by its vertices. We also train generative models on a subset of the 7d dataset where we give the polytope via vertices. By construction all samples will then automatically be lattice polytopes. The properties we test for under this formulation is therefore: reflexive, smooth, and normal. Training on these two different data representations can be considered analogous to stating the same semantic content in different languages.

3 Experiments

3.1 Implementation

We train recurrent networks with one or two LSTM layers [HS97]. We use an embedding dimension of 4 and hidden dimension 256 for each of the LSTM layers.

We also train transformer models [VSP+17]. This is a neural network which was designed for sequences, that utilizes an attention mechanism, as well as positional encodings, to help the model learn dependencies between different parts of a sequence. We use an encoder only transformer, in other words a small version of the GPT model [RHSS18]. We choose an embedding dimension of 128 and use 4 multi-attention heads and stack 4 such encoder layers to get the full model.

The models are trained to predict the next token in the dataset using cross-entropy loss. We generate examples by sampling from the output probabilities until we generate the end-of-sequence token. The sizes of the models are chosen so that the transformer and the 2-layer LSTM models will have approximately the same number of parameters.

3.2 Generating 8d polytopes

Our most fundamental experiments are training sequence models on the 8d dataset. After training a model we sample 1000 polytopes from every model and use the open source computer algebra software systems Macaulay2 [GS] and polymake [GJ00], [AGH+17] to check all the properties in question. For the generated polytopes satisfying all of the global properties we check whether they are exact copies of their representative in the training set. We also check whether they are the same as their representative up to reordering of rows. The defining inequalities is in reality a set and not a sequence, thus reordering the rows is the simplest form of coordinate change.

3.3 Baseline

To compare the results with a baseline we also make a simple model that generate samples based on sampling from the histogram of tokens following the previous n tokens, where n is some natural number (we choose $n = 10$ in our experiments). In particular this is by construction a local model.

Therefore it cannot use the complete history to generate the next token, hence it cannot know the total length of the sequence, which leads to artificially short samples. We remedy this by replacing the newline token to also include the line number, thus the model cannot generate the end of sequence token before at least $d^2 + d$ steps (since there are no such samples in the dataset).

3.4 Training on half the dataset

We also train models on half of the 7d and 8d datasets (chosen randomly) and generate polytopes from the models. Then we check whether the polytopes generated by these models are to a large extent from the half it was trained on, or whether they are equally from both halves. These tests give some indication whether the models are simply memorizing the dataset, or whether they learn some underlying structures.

3.5 Training on convex hull representation

We also train models on the convex hull representation of the dataset and compare the performance to that of the original hyperplane representation. For these experiments we use the 7d dataset, since the samples are longer in the convex hull representation, and thus in this case require more computational resources both to train the models and to check the properties of generated samples.

4 Results

We give results for the two different LSTM networks that have 1 or 2 hidden layers, which we call L1 and L2, the Transformer network called T and the baseline model called B. The properties we check for are abbreviated normal (N), smooth(S), lattice polyhedron(L), compact(C), and reflexive(R).

4.1 Generating polytopes and baseline

In Table 1 we see the how many of the generated samples satisfied the different global properties for the experiments on the full 8d dataset. In Table 2 we see how many generated examples satisfied some choices of several properties at once. In Table 3 we see how many of the correctly generated samples that equalled their representative in the training set precisely and up to permutation of rows.

We see that the models successfully manage to generate many examples with all of the correct properties. This shows that the models manage to pick up on global properties. The baseline using only local properties performs significantly worse than the neural models with almost no correct samples. We also see that the transformer models generally perform better than the LSTM models. Unsurprisingly some properties are shown to be much easier to understand than others, with smoothness being the hardest.

We also observe that almost none of the generated samples are identical to their representative in the dataset. This gives evidence that the model is not simply memorizing the data, but rather learning some underlying structure. However, a substantial portion (roughly 1 in 5) of the samples are equal to their representative up to permutation of rows. This proportion is way higher than one would expect from a random permutation. This could be an indication that the model does perform some amount of memorization. However, the examples in the training data are generated via the algorithm of [Øb07] which means there are systematic patterns in how the polytopes are represented in the training data. Hence it is not probable that a random permutation is a reasonable comparison, and thus we are not able to draw very clear conclusions from this.

Somewhat surprisingly the models trained on the hyperplane representation generated no examples of compact lattice polytopes that were not normal. We wonder whether understanding why might reveal some underlying structure, that could help in proving that all smooth lattice polytopes are normal?

4.2 Half of 8d dataset

For the models trained on half the 8d dataset the results for the properties of generated samples is seen in Table 4 and of intersections of properties in Table 5.

Model	L1	L2	T	B
All	238	260	404	2
C	827	840	909	21
L	712	722	832	412
S	270	304	440	178
N	590	613	755	2

Table 1: How many of the 1000 generated 8d samples which satisfies the properties.

Model	L1	L2	T	B
C+L	590	613	755	2
C+L+N	590	613	755	2
C+not S	589	580	505	19
C+L +not S	352	353	351	0

Table 2: Intersections of the properties of the 1000 generated 8d polytopes.

Model	L1	L2	T	B
Copy	0/238	0/260	0/404	0/2
Perm	36/238	58/260	77/404	0/2

Table 3: Fraction of generated polytopes that are copy/permutation of their representative in the dataset.

Checking which of the approximately 800 000 examples a generated polytope is equivalent to is computationally expensive so we cannot do this on large scale. However, we can quickly check the examples that are equal to their original representative up to permutation of rows, seen in Table 6. To get more substantial numbers we also train a transformer on half the 7d dataset and generate 1000 polytopes, of which 319 are correct. For all of these we manage to check which training sample a generated sample is equivalent to and find that 179/319 (approximately 56 %) of generated samples are from the half the model was trained on.

The experiments using half the data, while not having enough data to be certain, indicate that the models are actually learning something about the data, rather than simply memorizing: The fraction of correctly generated samples that are in the training data are only slightly exceeding 50 % (56 % for the largest set of examples we managed to check). If the model did a large amount of memorization one would expect to see a large majority of samples from the half the models were trained on.

4.3 Convex hull representation

We train Transformer and 2-layer LSTM on the subset of 7d polytopes that are less than 800 tokens in the convex hull representation and on the hyperplane representation of the same subset. In Table 7 we see the results for the convex hull representation, while Table 8 shows the corresponding results for the hyperplane representation.

When given the dataset in the convex hull representation the models generate less correct samples than they do in the hyperplane representation. The LSTM model in particular almost did not generate any examples with the correct properties. An obvious reason for this is that the sequences are longer.

Model	L1	L2	T
All	187	244	361
C	783	834	908
L	667	706	851
S	237	282	396
N	522	590	772

Table 4: Properties of the 1000 generated samples trained on half 8d data.

Model	L1	L2	T
C+L	522	590	772
C+L+N	522	590	772
C+not S	596	590	547
C+L+not S	335	346	411

Table 5: Intersections of properties for 1000 generated samples for half 8d data.

Model	L1	L2	T
Half 8d	15/28	16/37	42/73

Table 6: Among generated polytopes that up to permutation equal its representative in the data, this table shows the fraction which was in the training half.

Model	L2	T
All	7	285
R	7	285
S	7	286
N	27	410

Table 7: Properties of 1000 generated samples of 7d convex hull dataset.

Model	L2	T
All	216	547
C	660	909
L	616	850
S	288	583
N	421	784

Table 8: Properties of 1000 generated samples of 7d hyperplane dataset.

However these models also had some factors in their favour, since the space of possible objects are smaller.

Using the convex hull representation we can search for a counterexample: Since all 7d/8d smooth reflexive polytopes are known to be normal, a counterexample has to be smooth, non-reflexive and non-normal, and generating such a polytope (given our data representation) is only possible in the convex hull representation. We did not find any counterexample. While unsuccessful, we think that generative models might in the future be used to search for counterexamples in mathematics.

5 Conclusion

We have shown that neural network sequence models can generate lattice polytopes with geometric properties. Transformer models generate more correct samples than LSTM models and models learn more from the hyperplane representation than the convex hull representation. We have given evidence that indicates that the models are learning some underlying structure of the data.

References

[AGH+17] Benjamin Assarf, Ewgenij Gawrilow, Katrin Herr, Michael Joswig, Benjamin Lorenz, Andreas Paffenholz, and Thomas Rehn. Computing convex hulls and counting integer points with `polymake`. *Math. Program. Comput.*, 9(1):1–38, 2017.

[BGT97] Winfried Bruns, Joseph Gubeladze, and Ngô Viêt Trung. Normal polytopes, triangulations, and Koszul algebras. *J. Reine Angew. Math.*, 485:123–160, 1997.

[Cas06] Cinzia Casagrande. The number of vertices of a Fano polytope. *Ann. Inst. Fourier (Grenoble)*, 56(1):121–130, 2006.

[CLS11] David A. Cox, John B. Little, and Henry K. Schenck. *Toric varieties*, volume 124 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2011.

[GJ00] Ewgenij Gawrilow and Michael Joswig. `polymake`: a framework for analyzing convex polytopes. In *Polytopes—combinatorics and computation (Oberwolfach, 1997)*, volume 29 of *DMV Sem.*, pages 43–73. Birkhäuser, Basel, 2000.

[GS] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at `http://www.math.uiuc.edu/Macaulay2/`.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[KS00] Maximilian Kreuzer and Harald Skarke. Complete classification of reflexive polyhedra in four dimensions. *Adv. Theor. Math. Phys.*, 4, 03 2000.

[Oda08] Tadao Oda. Problems on minkowski sums of convex lattice polytopes, 2008.

[Paf17] Andreas Paffenholz. polydb: A database for polytopes and related objects, 2017.

[RHSS18] Alec Radford, Karthik Harasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[Øb07] Mikkel Øbro. *Classification of smooth Fano polytopes*. PhD thesis, University of Aarhus, 2007. available at pure.au.dk/portal/files/41742384/imfphd2008moe.pdf.

A Mathematical background on polytopes

Here we give a short introduction to the mathematics of polytopes and their properties, required to understand the datasets we use.

Definition 1 *A polyhedron is a subset of $\mathbb{R}^d$ defined by a finite number of linear inequalities. In other words a set of the form*

$$\{x \in \mathbb{R}^d | b + Ax \geq 0\},$$

for some matrix A and vector b. A polyhedron is called a lattice polyhedron if all vertices of P are integer points.

The dimension of a polyhedron is defined as the dimension of the smallest linear subspace containing the polyhedron. A polyhedron is called convex if it forms a convex subset of $\mathbb{R}^d$.

Definition 2 *A polyhedron which is compact (meaning it has finite volume) is called a polytope.*

In this paper we will study polyhedra up to lattice equivalence. In other words we consider two polyhedra P and Q as isomorphic if they are equal after some affine coordinate change of $\mathbb{R}^d$ that equals a translation term plus an element of $GL_d(\mathbb{Z})$. In other words a sum of an integer translation and a change of basis matrix with integer entries having determinant equal to ± 1.

Convex lattice polytopes has received a lot of attention in the intersection of toric geometry and string theory. To explain this connection we need to define the following property of a polytope:

Definition 3 *To a lattice polyhedron P one can associate the dual polyhedron $P^\vee$ defined by:*

$$P^\vee = \{u \in \mathbb{R}^d | x \cdot u \geq -1 \text{ for all } x \in P\},$$

where $\cdot$ is the ordinary scalar product of vectors. A lattice polyhedron containing 0 as an interior point is called reflexive if the associated dual polyhedron is also a lattice polyhedron.

For a lattice polytope the condition that $P^\vee$ is also a lattice polytope implies that 0 is the unique interior integer point of P. To a reflexive polytope one can associate a toric Fano variety. Inside these toric Fano varieties one can construct so-called Calabi-Yau manifolds. According to superstring theory the universe is a 10-dimensional manifold, where four dimensions are space and time dimensions, while the remaining six dimensions corresponds to a three-dimensional complex Calabi-Yau manifold (three complex dimensions corresponds to six real dimensions). The conjectural relationship between a Calabi-Yau and the dual Calabi-Yau associated to the dual polytope is what is known as *mirror symmetry*. Because almost all known examples of three-dimensional Calabi-Yau manifolds live inside four-dimensional toric fano manifolds, string theorists have been very interested in studying four-dimensional reflexive polytopes. This is the background for the construction of the Kreuzer-Skarke database of all reflexive polytopes of dimensions 3 and 4. Parallel to this development there has also been significant interest in reflexive polytopes in the mathematics community.

Definition 4 *A lattice polyhedron is called smooth if for any vertex the minimal integer points of the emanating edges form a $\mathbb{Z}$-basis for the integer points of $\mathbb{R}^d$. In other words, there has to be exactly d edges emanating from each vertex and the matrix containing the minimal integer vectors of these edges has determinant ± 1. This condition is equivalent to the condition that the toric variety associated to the polyhedron is a smooth manifold.*

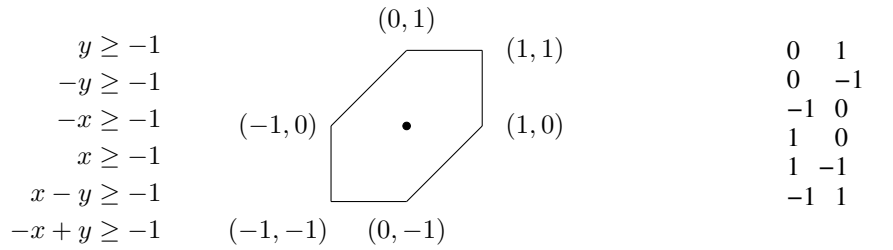

Figure 2: Left: defining inequalities. Center: polytope drawn with vertices. Right: how we store inequalities.

Of a given dimension there are only finitely many reflexive polytopes, up to integer change of coordinates. Øbro developed an algorithm to find all smooth reflexive polytopes of a given dimension [Øb07] which was used to construct the database of all smooth lattice polytopes up to dimension 9. An additional property a lattice polytope can have is that of normality:

Definition 5 *A lattice polytope P is called normal if it has the property that all integer lattice points of any integer multiple lP is the sum of l integer lattice points from P.*

It is a well-known open conjecture that states that any smooth polytope is normal [Oda08]. For reflexive polytopes this has been checked up to dimension 8; thus all polytopes of dimension less than or equal to 8 in the database of smooth reflexive polytopes are normal.

Example 1 *In Fig. 2 we show an example of a 2-dimensional polytope. It can be described as the set of points satisfying the inequalities below. It is compact since it has finite area. It is a lattice polytope since all its vertices are integer points. We remark that the inequalities being defined by inequalities with integer coefficients are far from being sufficient to imply that the vertices are integer points. It is reflexive since it has only one interior integer point. It is normal since all 2-dimensional lattice polytopes are normal [BGT97, Proposition 1.2.4]. We store the data defining such a polytope in terms of the inequalities defining it, where we omit the constant -1, since the constant term always can be assumed to equal -1 by [CLS11, Theorem 8.3.4]. Equivalently we could also store it by recording the vertices: In that case we recover the polytope by taking the convex hull of the vertices.*

B Datasets of reflexive polytopes

We suggest two different datasets of polytopes that can be used to train deep generative models: The Kreuzer-Skarke database of relexive polytopes of dimension 4 [KS00] and the database of smooth reflexive polytopes of dimension up to 9 [Paf17]. In the latter case the dimensions 8 and 9 are probably the most reasonable to train large deep models on, since there in those cases are a large number of examples.

The 7d dataset contains 72256 samples, while the 8d dataset contains 749892 samples. Each coordinate entry is regarded as a token, thus "0" is a possible token (in fact the most common) but also "-2" is one token. One could have chosen alternative tokenizations, for instance by considering each inequality as a token or each indivudal character a token. The vocabulary size will be around 30 with our chosen convention. The max sequence length of the 7d dataset is 169 tokens and 219 tokens for the 8d dataset. If one would like to scale up the experiments of this paper we consider both the 9-dimensional smooth reflexive polytopes and the 4-dimensional reflexive polytopes to be interesting datasets, since these are significantly larger than our chosen datasets.

The properties of the polytopes cannot be seen independently of each other: If P is not a lattice polytope we automatically have that P is not smooth. If P is a lattice polytope defined by inequalities of the form $1 + w_i \geq 0$, then it will automatically be reflexive. If P is reflexive and smooth then it is automatically normal (for $\dim P \leq 8$). If P is not compact then normality is not defined, so we consider it as not normal. If P is smooth, then by conjecture P is always normal, although this is not proven in general.

B.1 Details on convex hull dataset

In the hyperplane representation, a reflexive polytope of dimension d can have at most $3d$ defining inequalities [Cas06, Theorem 3]. For the number of vertices there are no such simple upper bound, and in practice they are indeed much longer, the longest example in the 7d dataset having more than 3000 tokens. We choose the subset of the 7d dataset consisting of those examples that have less than 800 tokens in the convex hull representation, to avoid very long sequences. We train generative models on both this and the corresponding hyperplane dataset and compare performance on the two datasets. This dataset has 39737 examples, thus it is approximately half the 7d dataset.

B.2 Sensitivity to local changes

To give an indication of how volatile the properties are to local changes, we drew 1000 random examples from the dataset and did a random change of either changing a single 0 to ± 1 or changing a single ± 1 to a 0. The distribution of the properties in the modified polytopes were:

- Compact: 851
- Lattice polytope: 789
- Smooth: 227
- Normal: 642
- All: 147

Thus only 147 out of 1000 are still correct datasamples with a single small change. With more than one change even less polytopes will be correct. This indicates that a successful model has to understand global properties, since the modified samples are locally plausible examples of the dataset.

C Training details

All models are trained with one polytope being one data sample. We add special start-of-sequence and end-of-sequence tokens to each sample and pad to the fixed maximum length. We train until the loss is reasonably stable, which is somewhere in the range 100-800 epochs, depending on size of model and dataset. We train with the Adam optimizer with learning rate 0.001. The models are trained on a single Titan V GPU with approximately 110 hours of combined training time. Checking the global properties for one list of 1000 8d polytopes took about 6 hours using a single CPU. We do not do a search of hyperparameters, since the experiments are more intended as a proof of concept than as a claim about the best possible performance, thus we have chosen to keep things as vanilla as possible.