Deep Learning Knot Theory

Arik Wilbert

Over the last decade, we have seen dramatic breakthroughs in what machine learning can do for us. Tasks that once seemed impossible for a computer to perform are now possible. Prominent examples of such tasks include image or speech recognition. In this project, we propose to use deep learning (a part of machine learning) to tackle difficult problems in pure mathematics, specifically knot theory. More precisely, the proposed project consists of three parts (see Subprojects 1, 2, and 3 below). The first two parts consist of training an artificial neural network to predict two important properties of knots. Once parts 1 and 2 are completed, the results will be applied to tackling an important open problem in knot theory; the question whether the Jones polynomial detects the unknot.

The applicant has a background in pure mathematics (algebra, geometry, topology) focusing on knot theory. We request $5,000 summer salary, so the applicant has the resources to spend the summer on expanding his knowledge on state-of-the art methods in deep learning, gather preliminary data (i.e., create a database of knot diagrams), train the neural networks, and devise follow-up projects that are suitable for student participation. Note that the applicant has already started building a machine learning community at the University of South Alabama by participating in an informal seminar that was organized by faculty and students in the Department of Mathematics and Statistics in Spring 2022.

1. Description of the project

1.1. **Machine Learning.** In order to understand the basic idea behind machine learning, take a look at the left picture in Figure 1 below.

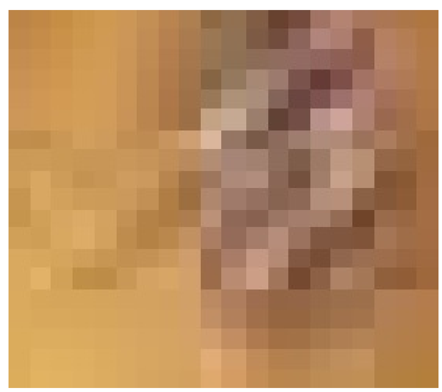

FIGURE 1. A jaguar (left picture) and a close-up of its fur (right picture).

It took the reader only a fraction of a second to determine that this picture shows the sculpture of a waving jaguar wearing a football uniform. If we zoom in on the fur of the jaguar (see the picture on the right in Figure 1), we realize that the picture of the jaguar is in fact made up of thousands of little colored squares. Isn't it remarkable how fast the human brain can process this information and decide that this grid consisting of thousands of colored squares assembles to depict a jaguar?

The part of our brain that makes us recognize the picture is the visual cortex. The visual cortex consists of millions of neurons. Modeling neural networks is the key idea behind what is called "deep learning". In order to explain this in more detail, the reader is referred to Figure 2 below which contains an example of a model of a neural network.

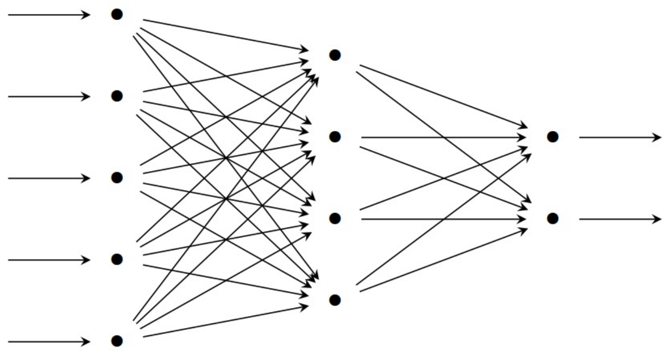

FIGURE 2. A model of a neural network.

The black dots represent the neurons. Each neuron can be thought of as a unit with several inputs and outputs connected to other neurons. In reality, neurons "fire" by emitting a charge along their axon. In our model, incoming and outgoing charges are encoded as real numbers. The neural network depicted above takes in five real numbers (as indicated by the five horizontal arrows on the left) and spits out two real numbers. Mathematically speaking, the neural network in the picture is a function $f\colon \mathbb{R}^5 \to \mathbb{R}^2$.

For a more concrete example, consider the task of recognizing handwritten digits. In this scenario, we would like to feed the model a picture of a digit. Just like the picture of the jaguar, any such picture consists of pixels (colored squares). A pixel is determined by a grayscale value between 0 and 255. Assuming the picture consists of 784 pixels, the data we can feed the computer model is a matrix (rectangular grid) of 784 numbers between 0 and 255.

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

FIGURE 3. A picture of a digit encoded in a pixel matrix.

In this context, the the neural network's job is to produce ten real numbers which we interpret as a probability distribution on $\{0, 1, \ldots, 9\}$. In other words, the neural network tells us with what probability it thinks that the handwritten number is a $0, 1, \ldots, 9$, respectively.

Put into a more mathematical framework, the goal is to learn a certain function $f\colon \mathbb{R}^{784} \to \mathbb{R}^{10}$ which takes in a pixel matrix and then outputs the associated probability distribution. In order to achieve this goal we have to "train" the neural network. This process boils down to providing the model with the correct answers for a large amount of input data. Based on this, the neural network then approximates the function f.

1.2. **Knot Theory.** In order to apply machine learning in knot theory (a subbranch of a mathematical discipline called topology), we first need to understand what a mathematician means by a "knot". In order to create a knot, take a piece of string or rope, tie a knot in it, and then glue the ends together. Note that joining the ends of the rope is what differentiates a "mathematical" knot from a knot one would usually encounter in everyday life.

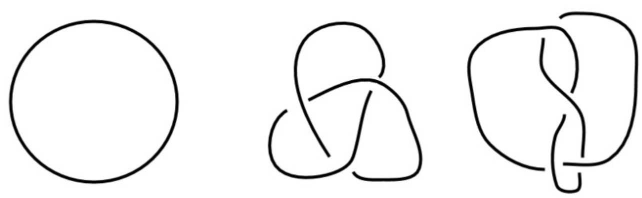

Figure 4. Three examples of knots.

An easy but important example of a knot is the unknot. The unknot can be drawn with no crossings, and is also called a trivial knot. It is the simplest of all knots (see the leftmost knot in Figure 4).

The fundamental problem of knot theory is to determine whether two given knots can be rearranged (without cutting) to be exactly alike. A special case of this problem is the following question. Given a knot, is it the unknot? Take for example the knot in Figure 5 below.

Figure 5. A tangled up unknot.

Is it possible to untangle this knot so that it looks like the unknot? The answer is in fact yes. But the human brain struggles to make such a decision within a short amount of time.

Subproject 1. Train a neural network to decide if the picture of a knot represents the unknot.

How can one decide if two given knots are different? Note that this is a difficult task. Being unable to transform one knot into the other does not prove that it is impossible. Maybe we are just not clever enough and need to try harder. In order to solve this problem, mathematicians have come up with ways to assign so-called “invariants” to knots. One of these invariants is the famous Jones polynomial. This polynomial was invented by Vaughan Jones in 1984. Jones later received the Fields Medal (the mathematician’s analog of the Nobel prize) for his groundbreaking discovery.

Recall that a polynomial is simply a formal expression of the form $-X^{-2} + 1 - 2X + 5X^3$. It is a finite sum of powers of X (note that we also allow negative powers!), where each power of X is additionally allowed to have a coefficient (a real number) in front of it. The Jones polynomial is a way of assigning such a polynomial to a knot. For example, the unknot gets assigned the polynomial 1, the second knot in Figure 4 gets assigned the polynomial $-X^{-4} + X^{-3} + X^{-1}$, and the third knot in Figure 4 has Jones polynomial $X^{-2} - X^{-1} + 1 - X + X^2$. There is a simple rule (which can easily be taught to high school students) to calculate the Jones polynomial of a knot. To keep this proposal nontechnical, we omit to explain how it works exactly.

What makes the Jones polynomial so useful is the following crucial property. If one draws two different pictures of the same knot, and then uses the algorithm to compute the respective Jones polynomials, one ends up with the same answer. For example, if one were to use the algorithm to compute the Jones polynomial for the knot in Figure 5, one would end up getting 1, since it can in fact be untangled to produce the unknot (which has Jones polynomial 1). The upshot of this construction is the following. If you have two complicated pictures of a knot and the respective Jones polynomials turn out to be different, then the two knots must in fact be different and we cannot transform one into the other.

The computational complexity of the usual algorithm for computing the Jones polynomial depends exponentially on the number of crossings of the knot. Thus, it is not feasible to calculate the Jones polynomial for knots with many crossings using the classical algorithms. This motivates to try a new machine learning approach to the problem of computing it.

Subproject 2. Train a neural network to produce the Jones polynomial given the picture of a knot.

2. Importance of the project

There is an ongoing discussion if AI will eventually replace human beings. The author of this proposal takes the standpoint that machine learning can serve as a valuable tool for fundamental research in the 21st century. But it will not render the human researcher superfluous. The point of using machine learning in mathematical research is not to replace the mathematician. Instead, it should be thought of as a modern tool similar to an undergraduate who thinks of his calculator as a valuable tool in solving calculus problems. More specifically, we believe that machine learning will be useful in generating and verifying conjectures. However, note that it will still be up to the mathematician to then prove these conjectures.

Based on the general considerations in the previous paragraph, we propose to use deep learning to specifically study the following open question in knot theory.

Open Question. *Does the Jones polynomial detect the unknot?*

Recall that if two knot diagrams have different Jones polynomials, then the knots must be different. How about the converse? Suppose you have a complicated picture of a knot and its Jones polynomial tuns out to be 1 (the Jones polynomial of the unknot). Does this imply that the knot can be untangled? This is what we mean when we ask if the Jones polynomial “detects the unknot”. It is arguably one of the most important unanswered questions about the Jones polynomial. There has been little progress towards answering it since the discovery of the polynomial in the 80’s. This is why we propose to use machine learning as a new tool to tackle it.

Subproject 3. Use the results from Subprojects 1 and 2 to prove that the Jones polynomial detects the unknot or come up with a counterexample.

Upon the successful completion of Subprojects 1 and 2, we can use the resulting neural networks to compute the Jones for a large number of complicated knots (Subproject 2) and at the same time decide if the knot is the unknot (Subproject 1). This will provide an efficient way to search for counterexamples to the conjecture that the Jones polynomial detects the unknot.

3. Anticipated Outcomes

The applicant is confident that this is a feasible project which can be completed over the summer. The most time-consuming part will be the creation of a data base of knot diagrams to be used in training the neural networks for Subprojects 1 and 2. The actual training should only take a couple of days. The applicant intends to use existing software libraries (such as TensorFlow) to create and train the neural networks. Depending on the size of the data, the applicant will resort to using the Alabama Supercomputer Center as an additional resource for training. Subproject 3 is more open-ended and leaves potential for future research. While the proposed project is self-contained, it could easily be expanded into a larger long-term project.

We expect the outcomes of the proposed research to lead to results publishable in peer-reviewed journals. As a consequence, this will also make the applicant more competitive when applying for external grants in the future.

If expanded into a long-term project, this would positively impact the University of South Alabama in at least two ways.

(1) The applicant sees potential for future collaboration on related projects with other departments. Knot theory has many applications in diverse fields such as biology (DNA-protein interaction, topoisomerase enzymes), medicine (mechanisms of antibiotic resistance), chemistry (molecular chirality), and physics (quantum entanglement, quantum computing). As such, this project could provide a basis for future interdisciplinary research beyond the College of Arts and Sciences including the School of Computing, the School of Engineering, and the College of Medicine.

(2) The successful completion of the proposed project provides a basis for student research on related projects in the future. Note that once a training set of knot diagrams has been created, this training set can be used to train other neural networks to investigate other questions about knots. We believe that research projects on the interface of knot theory and deep learning are particularly well-suited for student involvement. Note that knot theory is a branch of mathematics that does not require a sophisticated, abstract machinery to be understood. As such, it is widely accessible to undergraduate students. Students participating in this project will not only be able to discover deep facts about pure mathematics, but they will also gain valuable job skills that will prepare them to excel in their careers in the data-driven 21st century. This would very well align with the LevelUP initiative that will shape the University of South Alabama in the coming years.