

[Get started](#)[Open in app](#)

Flawnson Tong

[Follow](#)

1K Followers

[About](#)

What is Geometric Deep Learning?

Deep Learning 🤖 on graphs and in 3D



Flawnson Tong Apr 18, 2019 · 9 min read



Follow [my Twitter](#) and join the [Geometric Deep Learning subreddit](#) for latest updates in the space.

Get started

Open in app



In 1D or 2D world. All that we can observe exists in 3D, and our data should reflect that. It's about time machine learning gets to our level.

Images, text, audio, and many others are all euclidean data.

7

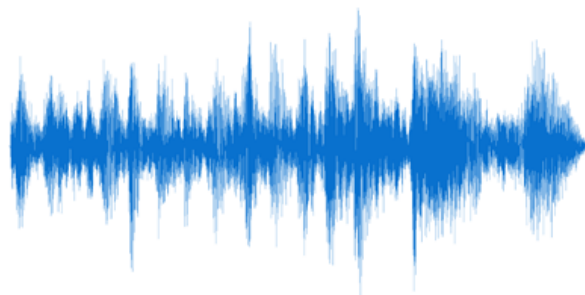
Numbers



Images

The quick brown
fox jumps over
the lazy dog

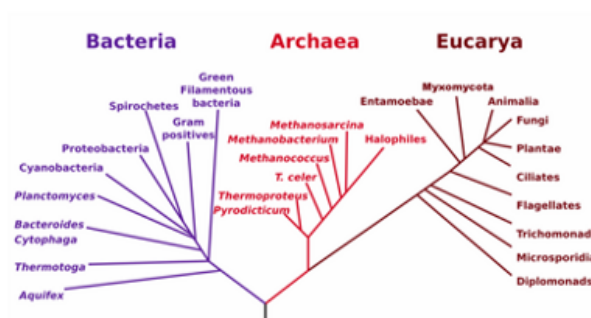
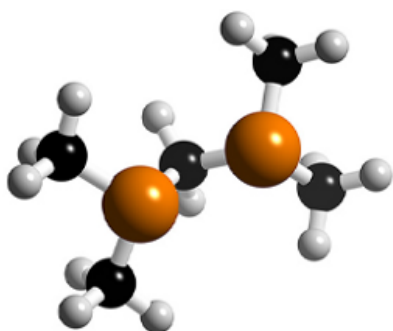
Text

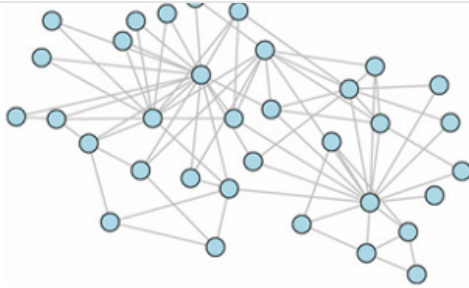
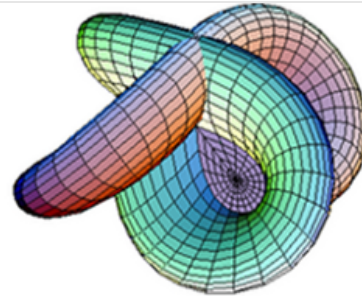


Audio

Euclidean Data

Non-euclidean data can represent more complex items and concepts with more accuracy than 1D or 2D representation:



[Get started](#)[Open in app](#)**Networks****Manifolds**

Non-Euclidean Data

When we represent things in a non-euclidean way, we are giving it an **inductive bias**. This is based on the intuition that, given data of an arbitrary type, format, and size, one can prioritize the model to learn certain patterns by **changing the structure of that data**. In the majority of current research pursuits and literature, the inductive bias that is used is **relational**.

Building on this intuition, **Geometric Deep Learning (GDL)** is the niche field under the umbrella of deep learning that aims to build neural networks that can learn from non-euclidean data.

The prime example of a non-euclidean datatype is a graph. **Graphs** are a type of data structure that consists of **nodes** (entities) that are connected with **edges** (relationships). This abstract data structure can be used to model almost anything.

We want to be able to learn from graphs because:

Graphs allow us to represent individual features, while also providing information regarding relationships and structure.

There are various types of graphs, each with a set of rules, properties, and possible actions. **Graph theory** is the study of graphs and what we can learn from them. This will be covered in the next part of this series.

Examples of Geometric Deep Learning

[Get started](#)[Open in app](#)

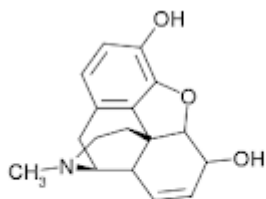
Molecular Modeling and learning

For a concrete example of how **Graph Learning** can improve existing machine learning tasks we can look at the computational sciences.

One of the bottlenecks in computational chemistry, biology, and physics is the representation concepts, entities, and interactions. The nature of science is empirical and is therefore the result of many external factors and relationships. Here are some examples of where this is most obvious:

- Protein interaction networks
- Neural networks
- Molecules
- Feynman diagrams
- Cosmological maps

Our current methods of representing these concepts computationally can be considered “lossy”, since we lose a lot of valuable information. For example, using a simplified-molecular-input-line-entry-system (SMILE) string to represent molecules is easy to compute, but at the expense of structural information of the molecule.



Morphine

SMILES: CN1CCC23C4C1CC5=C2C(=C(C=C5)O)OC3C(C=C4)O

A molecule and its SMILE string

By treating atoms as nodes, and bonds as edges, we can save structural information that can be used downstream in prediction or classification.

[Get started](#)[Open in app](#)

3D Modeling and Learning

As an example of how **Geometric Deep Learning** lets us learn from datatypes never used before, consider a person posing for a camera:

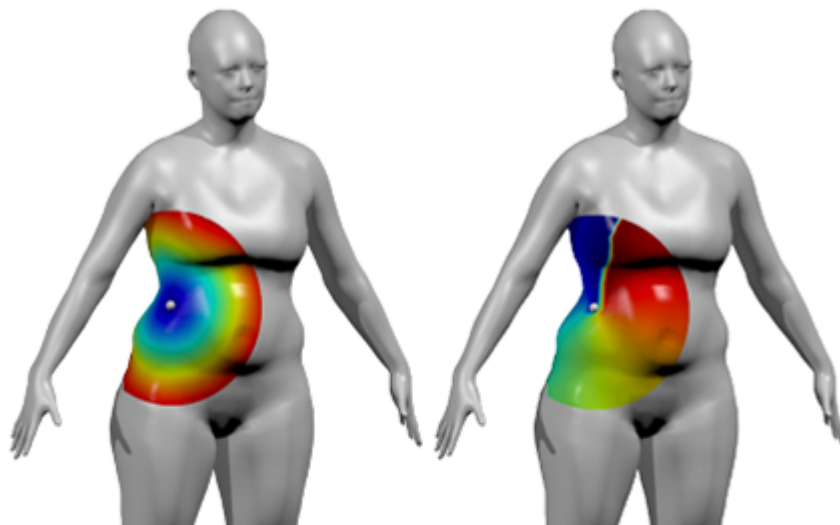


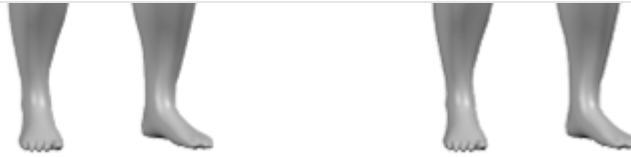
People pose for a 2D image capture (Courtesy of the MoNet team)

This image is 2D, although in our minds, we are aware that it represents a 3D person. Our current algorithms, namely Convolutional Neural Networks (CNN), are pretty good at predicting labels like the person posing and/or the kinds of poses given only a 2D image. **The difficulty arises when poses become extreme and the angle is no longer fixed.** Often times there may be clothing or objects in an image that obstruct the view of an algorithm, making it difficult to predict the pose.

Now imagine a 3D model of this same person making poses:

The CNN can now be run on the 3D object itself rather than a 2D image of the object.



[Get started](#)[Open in app](#)

Obviously, this is a 3D model as shown on a 2D screen but use your imagination! (Courtesy of the MoNet team)

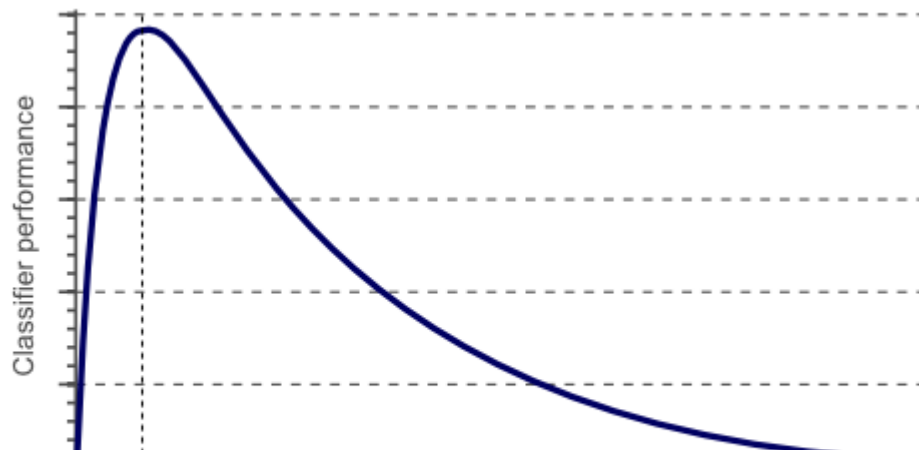
Instead of learning from a 2D representation, which restricts the data to a single perspective angle, imagine if we could run a convolution directly on the object itself. Analogously to traditional CNNs, **the kernel would pass through every “pixel” represented as a node in a point-cloud (basically a graph that wraps around the 3D object)**. Every corner and crevice on the 3D model would be covered and the information will be considered. In short, the difference between vanilla CNNs versus its Geometric equivalent is predicting the label of an object given a picture of it, versus predicting the label of an object given a 3D model of it.

As our 3D modelling, design, and printing technology improves, one could imagine how this would yield far more accurate and precise results.

The case of Dimensionality

Dimensions in the traditional sense

The notion of dimensionality is already commonly used in data science and machine learning, where **the number of “dimensions” correlates to the number of features/attributes per example/datapoint in a dataset.**



[Get started](#)[Open in app](#)

Optimal number of features

The Hughes phenomenon (Courtesy of Visiondummy)

While at first, the performance of machine learning algorithms spikes, after a certain number of features (dimensions), the performance levels off. This is known as **the curse of dimensionality**.

Geometric Deep Learning doesn't solve this problem. Rather, algorithms like graph convolutions reduce the performance penalties incurred when using datatypes that have a lot of features, since **relational data is considered via inductive bias and not as an additional feature**.

What we talk about when we talk about dimensionality

Dimensionality in Geometric Deep learning is just a question of data being used in training a neural network. Euclidean data obeys the rules of euclidean geometry, while non-euclidean data is loyal to non-euclidean geometry.

As explained by [this awesome StackExchange A.I stream post](#), Non-Euclidean geometry can be summed up with the phrase:

“the shortest path between 2 points isn't necessarily a straight line”.

Other strange rules include:

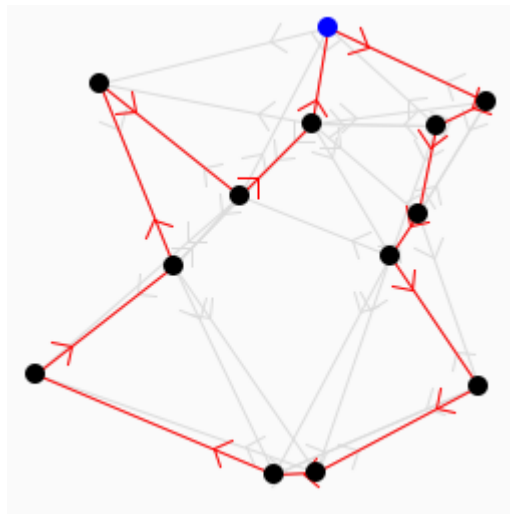
- Interior angles of triangles always add up to more than 180 degrees
- Parallel lines can meet, either infinitely or never
- Quadrilateral shapes can have curved lines as sides

There is an entire field of non-euclidean geometry which is another topic on its own. For a bit of an intuition-boost, take an image, one of the most popular euclidean datatypes.

[Get started](#)[Open in app](#)

Pixel art can be conceptualized as a collection of pixels positioned with left, right, up, and down operations

An image that is made up of pixels have a notion of left, right, up, and down. One can traverse the image by translating a function over the image recursively. This is exactly what a CNN does.



Graphs have no sense of cardinal direction; it is more abstract.

On a graph however, there is no notion of left, right, up, or down. There is just a node that is connected to an arbitrary number of nodes. A node can even be connected to itself.

Dimensions in the traditional sense of machine and deep learning still exist in the use of non-euclidean data for training neural networks. It is entirely possible to have many node features for example, where each feature is another “dimension”. But the term is rarely used in literature to represent this.

[Get started](#)[Open in app](#)

handful of popular algorithms. Each algorithm roughly specializes in a specific datatype. Just as **RNNs** were built for time-dependent data and **CNNs** for image-type data, Graph neural networks (**GNNs**) are a type of Geometric Deep Learning algorithm built for graphs and networks. As said by Graham Ganssle, Head of Data Science at Expero:

Graph convolutional networks are the best thing since sliced bread because they allow algos to analyze information in its native form rather than requiring an arbitrary representation of that same information in lower dimensional space which destroys the relationship between the data samples thus negating your conclusions. — [Graham Ganssle](#) (in a Tweet)

Graph convolutional networks, or GCNs is to a Graph neural networks what CNNs are to Vanilla neural networks.

The implication of this new method makes a big difference; we are no longer forced to leave behind important information in a dataset. Information like **structure, relationships, and connections**, which are integral to some of the most important data-giving tasks and industries like transportation, social media, and protein networks.

In short, the field of Geometric Deep Learning has **3 main contributions**:

1. We can make use of non-euclidean data
2. We can maximize on the information from the data we collect
3. We can use this data to teach machine learning algorithms

[Get started](#)[Open in app](#)

We argue for making generalization a top priority for AI, and advocate for embracing integrative approaches which draw on ideas from human cognition, traditional computer science, standard engineering practice, and modern deep learning. — DeepMind, Google Brain, MIT, and the University of Edinburgh

In other words, we have much to expect from Geometric Deep Learning.

In Essence

So what is Geometric Deep Learning?

Geometric Deep Learning is a niche in Deep Learning that aims to generalize neural network models to non-Euclidean domains such as graphs and manifolds.

The notion of relationships, connections, and shared properties is a concept that is naturally occurring in humans and nature. Understanding and learning from these connections is something we take for granted. Geometric Deep Learning is significant because it allows us to take advantage of **data with inherent relationships, connections, and shared properties.**

Key Takeaways

- The Euclidean domain and non-euclidean domain have different rules that are followed; **data in each domain specializes in certain formats** (image, text vs graphs, manifolds) and convey differing amounts of information

[Get started](#)[Open in app](#)

predictions and classifications on **relational datatypes**

- The difference between traditional Deep Learning and Geometric Deep Learning can be illustrated by imagining the accuracy between **scanning an image of a person versus scanning the surface of the person themselves.**
- In traditional Deep Learning, dimensionality is directly correlated with the number of features in the data whereas in Geometric Deep Learning, it **refers to the type of the data itself, not the number of features it has.**

One of the reasons I set out to write about Geometric Deep Learning because there is hardly any entry-level resources, tutorial, or guides for this relatively new niche. With that ultimate goal in mind, I am writing a series of articles all on Graph Learning. All articles can be accessed here:

[Graph Learning — Part 0: Graph Theory and Deep Learning prerequisites](#)

[Graph Learning — Part 1: Overview of Graph Embedding for Deep Learning](#)

[Graph Learning — Part 2: Graph Convolutional Networks for GDL](#)

UPDATE: Nov 20th, 2020

The field has changed and grown a lot since this article was written, and I've learned a lot over the past year.

Geometric Deep Learning can now be found being used in the front lines of companies like [Pinterest](#), [Twitter](#), [Uber](#), [Accenture](#), and the list goes on to cover just about every company with a finger in the field of ML. In academia, Graph Representation Learning and related terms constantly top the list of topics in discussion for conferences from [NeurIPS](#) to [ICML](#) (1, 2). We're beginning to see [cross-domain comparisons](#) and novel models with novel applications.

It was already difficult to summarize GDL merely a year ago and it's futile to do it now. But I feel compelled to try again anyway and obligated to write a more accurate (and updated) summary for an ever-increasing audience because I really love what I do, and I

[Get started](#)[Open in app](#)

Need to see more content like this?

Follow [my Twitter](#) and join the [Geometric Deep Learning subreddit](#) for latest updates in the space.

Follow me on [LinkedIn](#), [Facebook](#), [Instagram](#), and of course, [Medium](#) for more content.

All my content is on [my website](#) and all my projects are on [GitHub](#)

I'm always looking to meet new people, collaborate, or learn something new so feel free to reach out to flawnsontong1@gmail.com

Upwards and onwards, always and only 🚀

Some rights reserved ⓘ

[Machine Learning](#)[Artificial Intelligence](#)[Startup](#)[Entrepreneurship](#)[Technology](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

