Machine Learning: Methods & Current Landscape in Architectural Practice

Introduction

Artificial intelligence, or AI, has become a standard science fiction movie trope at this point, leading to many misconceptions and in some cases, even fear. Put simply, artificial intelligence is development of computer programs to complete tasks normally requiring human input. One leading field of research within AI is machine learning (ML). Machine learning is a subset of AI in the same way that a square is a quadrilateral but not necessarily vice versa. Machine learning and artificial intelligence have found vastly increasing uses throughout every facet of human life, regardless of whether we, as individuals, notice its impact or not. However, this ubiquity has not lended itself to the use of learning algorithms within the field of architecture. Design tends to be a heavily conservative profession when it comes to the adoption of emerging methods and technologies, but architecture has consistently had an avant-garde wing of researchers and practitioners ready to take the leap.

The vast majority of existing ML research and development has been in two contrasting methods: supervised and unsupervised learning. Both techniques involve feeding vast datasets into a machine learning model in order to 'train' the algorithm to accomplish a relatively simple (for a human) task. Supervised learning is built upon training the model using explicitly labeled data in order to predict the *classification* of inputs. On the other hand, unsupervised learning is a means of training a machine learning model using entirely unlabeled data in order to discover insights and/or hidden patterns in the data.

The "hello world!" equivalent, or initial learning exercise for many data science students when first introduced to machine learning consists of training an algorithm to identify handwritten digits (0-9). This is the classic example of a supervised learning training method; however, beyond classification lies methods of regression as well. Regression is a method to define a mathematical equation that defines

the general alignment of data. These methods range from as simple as linear regression, a slope

determining function taught in most high school mathematics curriculums, all the way up to the much

more rigorous K-nearest neighbor- a means of organizing data spatially by association.

Available Tools and Frameworks

Beyond the various descriptors for methods, it is important to discuss the availability of tools-

for users anywhere between computer scientists and architects. At the most basic level, machine

learning tools are built using high level programming languages such as R and Python. These languages

are among the most prevalent and easiest to understand (high level refers to compounding levels of

abstraction in code above the base layer of machine code, unintelligible to humans). To build a machine

learning model via programming from scratch is a naïve endeavor for an architect, let alone a data

scientist. Instead, existing libraries (sets of premade, usually open-source functions one can import into

scripts) and frameworks have been created and are maintained by professionals and corporations.

Above: SciKit Learn and Google TensorFlow (1)*.*

Google operates and maintains TensorFlow, one of the largest and most utilized machine

learning libraries in the world- Tensorflow is used by companies from Twitter to PayPal. The main draw

to TensorFlow is its ability to allow the computer it is run on to utilize the graphics processing unit

(GPU), heavily increasing processing power and thereby training speed (2). Further, TensorFlow offers essentially what are the building blocks of creating and training learning algorithms in the form of low-level functions built into its library. TensorFlow especially shines in implementing neural networks- a further specialized method used by a type of machine learning called deep learning. Deep learning mimics the human brain's network of neurons by creating and training a map of artificial neural networks (ANNs, for short). Training a model, whether a neural network or any other of hundreds of subsets within machine learning essentially boils down to optimizing the tolerances and biases of each connection between each node within the model. These tolerances and biases refer to varying slopes and intercepts of functions representing each connection- each time a new handwritten number, for example, is run through the model, every tolerance/bias is tweaked with respect to every other value. This requirement of raw computing power is the bottleneck to advancements in the field of learning algorithms and AI.

In contrast to TensorFlow, SciKit Learn is a framework for machine learning- another level of ease and abstraction built upon the previous. SciKit Learn contains many prebuilt implementations of machine learning algorithms. This is valuable in situations where the model a practitioner is after is relatively simple and one cannot spend too much time waiting while training the model. Regression and classification especially are where SciKit Learn shines due to the shift in importance to concepts more representable with pure math, as opposed to the (again) identification of handwritten digits. However, as the onus of this paper is on the implementation of machine learning workflows into architectural practice, I have concluded that as powerful as the above two options are, architects do not reliably have the time nor expertise to utilize either tool. Enter Lunchbox ML for Grasshopper.
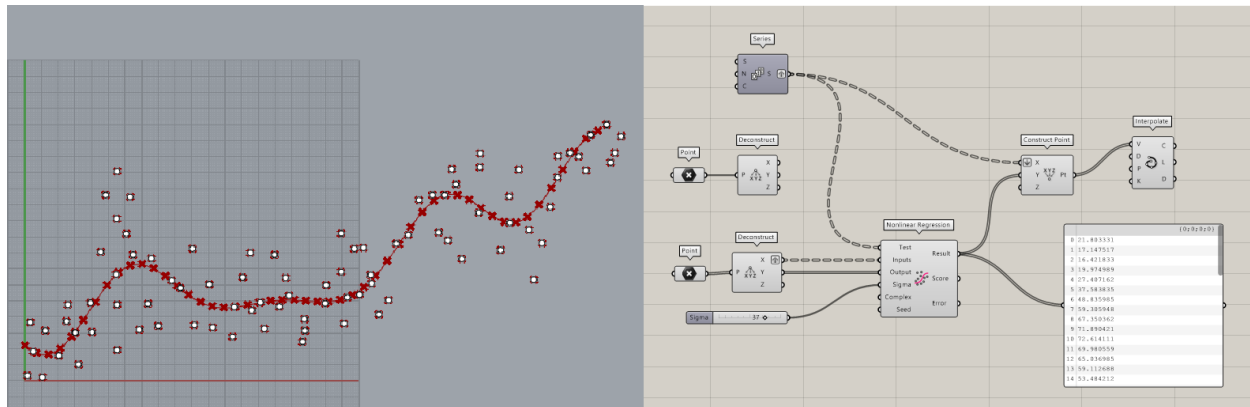
Figure 1: Non-linear regression using Lunchbox ML

Lunchbox (3) is a free and open-source toolkit of many incredibly useful features largely missing from Grasshopper: paneling, Excel/CSV input and output, simple generative tools for structure, and most importantly here- machine learning. Lunchbox allows architects and designers to run anything from 2-dimensional regression models by utilizing Grasshopper and Rhino as an XY plane graph (Figure 1) to creating neural networks trained to identify specific typologies of room or structure in 3D. Within the associated research project for this course, work was primarily done and tested within the Rhino/Grasshopper/Lunchbox environment. The above example of non-linear regression using Lunchbox shows a simple means of finding the fit curve between points placed at random by the user; however, it is not difficult to imagine use cases in which this could be implemented to understand a 3D scan or GIS data, either represented as a point cloud or similar.

Another simple example of Lunchbox is shown in Figure 2: using the Gaussian Mix component to isolate groupings of panels with similar attributes. This ML model simply takes in the means to sort objects (here being areas and perimeters of each individual panel) and the number of groups wished for by the user. The algorithm then groups panel sections accordingly and colors them via a vanilla Grasshopper gradient component (see below). However, beyond these candid examples shown it is important to step back and view professional work done at the intersection of architecture and learning algorithms.
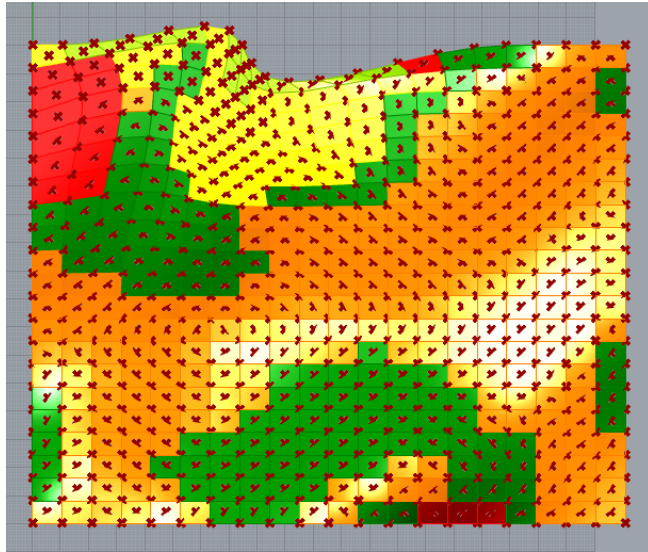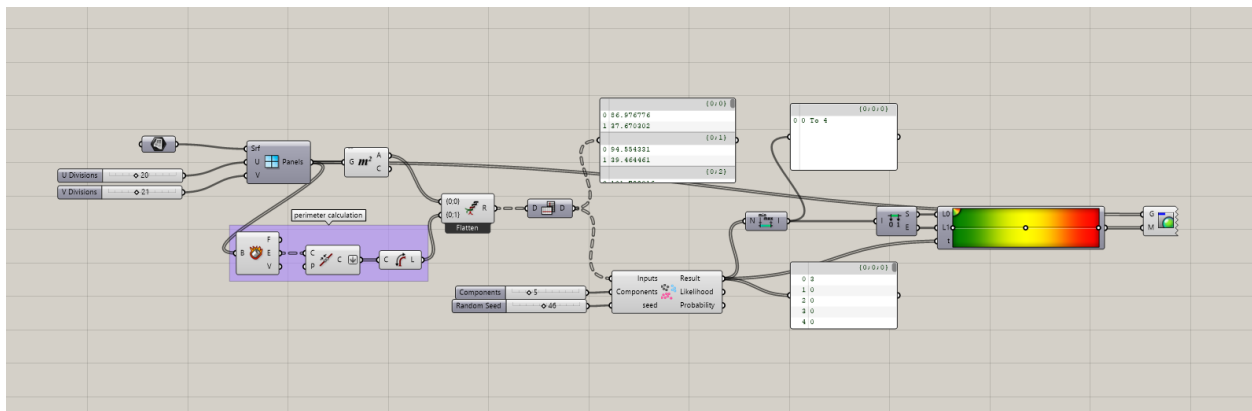
4

Figure 2: Gaussian mix by panel attributes

Figure 3: Gaussian mix Grasshopper definition

Case Studies

Alisa Andrasek is a leading user of machine learning workflows in architectural practice, both

from a formal and analytical standpoint. As a member of Wonderlab, working with students at the

Bartlett School of Architecture at the University College of London, Andrasek helped create *Morphocyte*

(2015), a project showcasing so-called "counter-intuitive aesthetics" (4). Morphocyte consists of a

cellular division algorithm, which generates an incredible number of design iterations largely following

the biological logic of cell reproduction and cancer growths. To sort through the sheer number of results, a supervised learning program was implemented in which the designers point out instances they expected or preferred before then allowing the learning algorithm to find similar, yet unexpected designs. Andrasek previously had submitted a competition entry for a large 40,000 m^2 port terminal in Kaohsiung, Taiwan titled *Fissures,* utilizing an algorithm generating a vast selection of fractal geometries rendered in glass (5). The designs were then sifted through and narrowed down to a winner using a similar method as *Morphocyte,* utilizing a learning algorithm to find designs with traits specified by Andrasek. Further, the ML algorithm purposefully took selected designs and distorted the results, skewing ever further into the chaotic nature of fractal geometry (6).

On the analytical side of ML usage in architecture and the related field of urban design and planning, much work has been done in the analysis of housing data. A study was conducted in 2019 based in China to attempt to understand the reasons for renters choosing specific locations to live (7). Researchers utilized machine learning models trained using data harvested from online housing rental websites (OHRWs), like the US-based Zillow, to begin to understand the reasoning behind location, price, and how inequities in housing are derived. Further, data showed that growth rate in most Chinese cities is greatest on the outskirts, implying those most at-risk for homelessness or other housing inequalities suffer most from planning decisions due to increased commute/walk distances and times to job hubs in central districts. Results from this study were implemented in Shenzhen as a methodology to inform the creation of equitable housing policy and was deemed effective.

Final Thoughts

Seeing the success of early forays into utilizing machine learning in the realm of architecture, it is important to note that this is only the beginning of a great paradigm shift within the profession. The next decade may well be the decade in which computational workflows become the norm inside of

firms, large and small. However, currently most firms are lucky to have a singular designer with

knowledge beyond the surface level of Grasshopper or Dynamo; but, as time marches onwards more

and more students will be graduating with technical expertise in advanced simulation, analysis, and

optimization methods. Given the lack of this specialized knowledge at present, it is worth noting that ML

implementations using Grasshopper plugins such as Lunchbox are especially useful for less technically

inclined users. While, on the other hand, those willing to do a deep dive will likely find the options

beyond to be more suitable. Python- or R-based models are best for high-detail or highly repeatable

design and analytic scenarios, where Grasshopper shines as an on-the-fly solution. The former code

implementations have the additional benefit of being operating system agnostic, whereas portions of

Lunchbox do not function correctly on MacOS vs Windows, for example. Downsides to learning

algorithms, such as the requirement of vast datasets to conduct training are decreasing as time goes on,

primarily due to the increasing recognition of the value of big data. Perhaps in the future, design firms

may even pool together their knowledge, contributing in an almost crowdfunding manner to improve

the state of the built environment, bringing architecture closer to an exact science.

References

1. TensorFlow vs. Scikit-Learn: How Do They Compare? [Internet]. Springboard. 2021. Available from: https://www.springboard.com/library/machine-learning-engineering/scikit-learn-vs-tensorflow/

2. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-scale machine learning on heterogeneous systems [Internet]. 2015. Available from: https://www.tensorflow.org/

3. Lunchbox [Internet]. Proving Ground. Available from: https://provingground.io/tools/lunchbox/

4. Andrasek A. In search of the unseen: Towards superhuman intuition. Archit Des. 2019;89(5):112–9.

5. Andrasek A. Fissures Port [Internet]. Available from: https://www.alisaandrasek.com/projects/fissures-port

6. Bottazzi R. Learning algorithms, design, and computed space. Enq ARCC J Archit Res. 2019;16(2):6–17.

7. Hu L, He S, Han Z, Xiao H, Su S, Weng M, et al. Monitoring housing rental prices based on social media:An integrated approach of machine-learning algorithms and hedonic modeling to inform equitable housing policies. Land Use Policy. 2019;82:657–73.