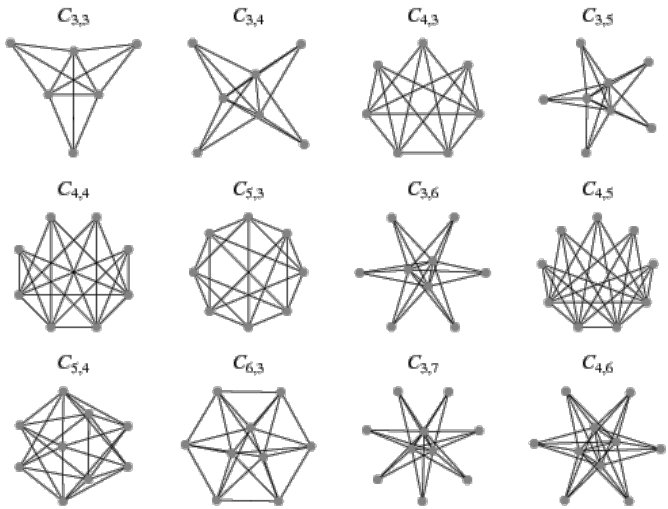
Adam Ferris

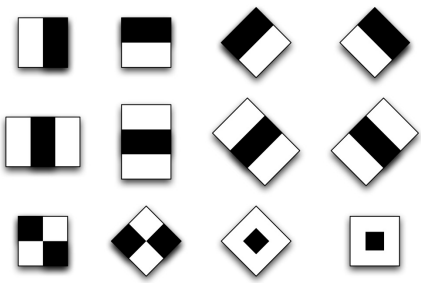


Unsupervised Learning

The title Unsupervised Learning takes its name from a type of training used for developing "models" from unlabeled data. There are a few approaches to analyzing huge datasets, but obviously unsupervised learning stands in opposition to supervised learning.

Supervised learning is where the data has been prestructured. For example, information in an Excel spreadsheet, or a set of images embedded with metadata and ordered a certain way. There are certain rules bounding the data, so the machine knows what is where and how to compare elements against each other.

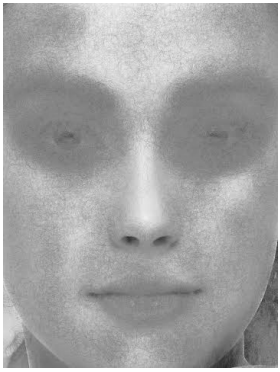
Unsupervised learning is when the data has no inherent structure. For example, a set of unrelated images found on the internet. The computer doesn't start out with any idea of what should be where, and has to evolve a model over time on its own.



When people talk about neural networks in computers, they are generally referring to unsupervised learning systems. These systems are often derived/ inspired by biological processes found in the brain. There are a series of input and output nodes, and successful ones are strengthened over time, whereas less successful ones are culled.

This kind of learning is often used to figure out what kind of similarities there are in data sets, essentially, the data scientists don't know the questions to ask about the data set, so they let the algorithm determine what is important / common.

The screen based piece is using something called principal component analysis (PCA). I don't quite understand the math on this, but you don't have to in order to wield the algorithm. The code takes a set of faces and averages them all, and then subtracts each face from the average to generate a new set of faces.



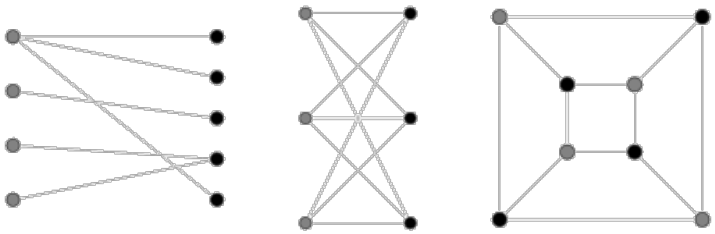
The final output images can be used as input for another face detection algorithm called Eigenfaces. Eigenfaces are sort of an outdated and clumsy way of doing face detection, but I find the output images really fascinating by-products of the research.

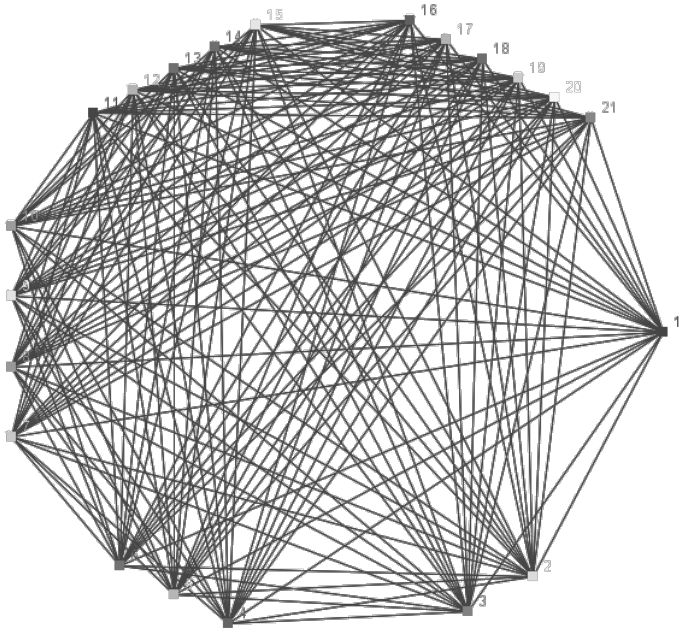
I've also been interested in this idea of abstract nonsense. This is a term used by mathematicians when describing certain ideas related to a branch of math called category theory. Category theory tries to define the structure of mathematical models in very broad strokes without being interested in the actual content of the equation at hand.

A proof in category theory might be called abstract nonsense because it lacks any kind of context given by the original equation, and as such can seem almost unrelated. I don't pretend to understand a lot of what's going on in many of these algorithms, but I can usually understand in general terms the purpose of what the algorithm is trying to do.

There is some kind of parallel here about using a tool even though you have no idea what's going on the inside. I think for many people computers are black boxes in this way.

At some level, the inner workings of most systems become seemingly arbitrary and chaotic. But this is really not the case; beneath that chaos is a complex and ordered system.




16
17
18
19
20
21
15
14
13
12
11
1
2
3