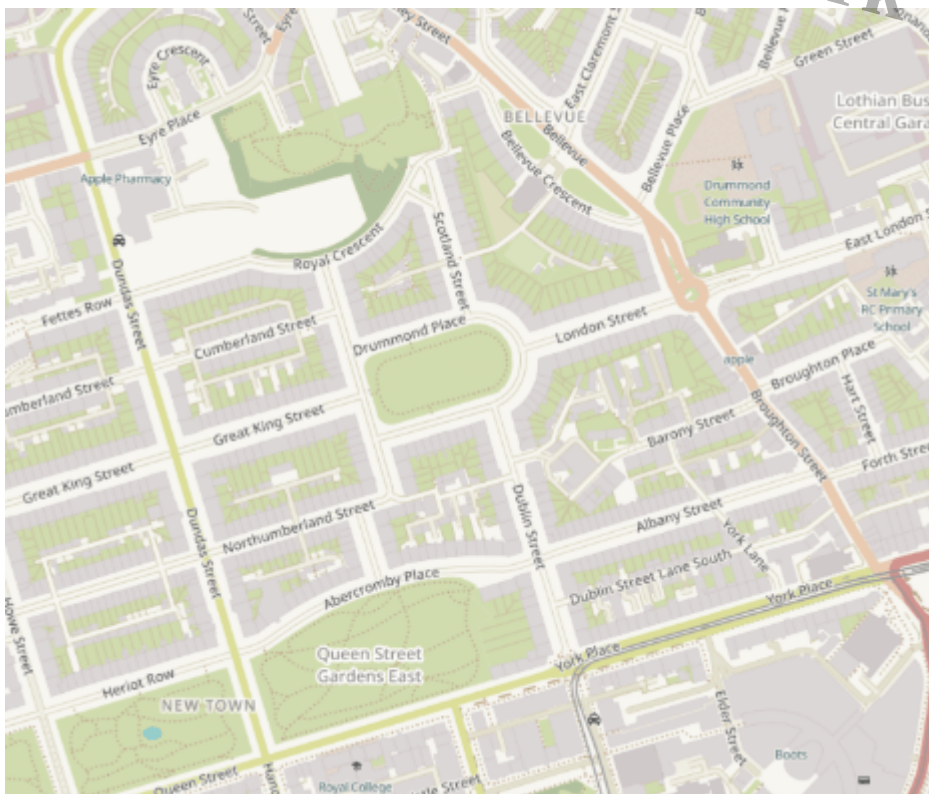




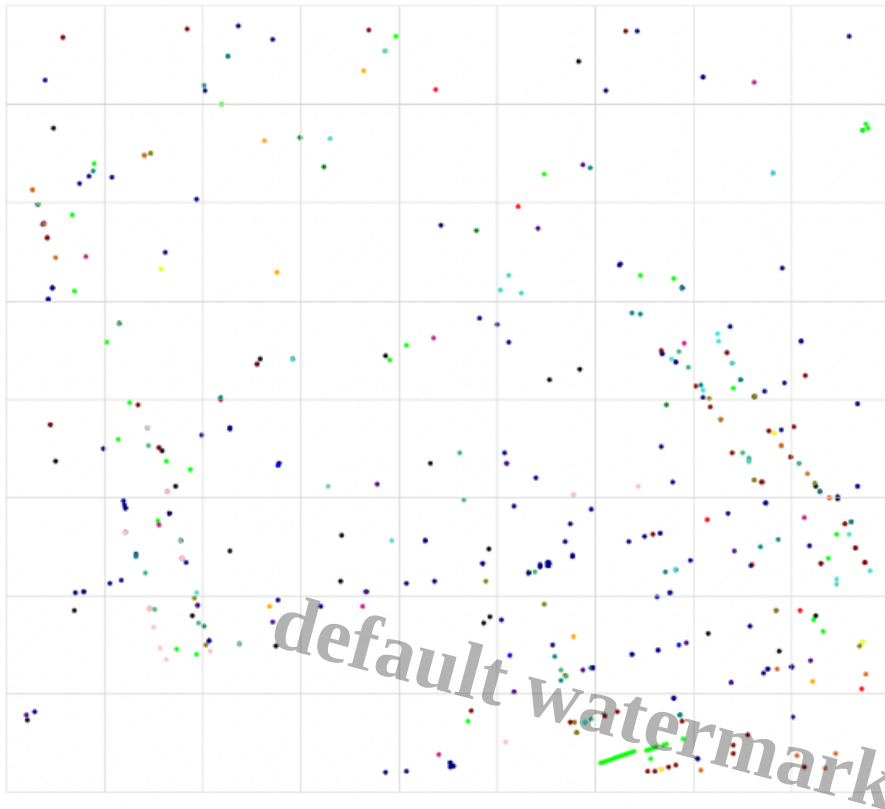
The neural neighbourhood

Description

In a previous post ([AI makes AI](#)) I described an an auto-associative neural network with an example using made-up data. Hereâ??s a more sophisticated version with actual urban data. Consider this fragment of a city map.



I overlaid a 100 metre grid carving the neighbourhood into 72 cells. Within each of these cells lies a combination of items (features), each belonging to a distinct category. For instance, the cell at the bottom left of the gridded map includes a business office, a hotel, and a historical commemorative plaque. I took this data from the Ordnance Survey module in [Digimap](#).



With help from ChatGPT4 I wrote a program in Python to create and access a neural network to identify patterns in this data.

My goal here was to demonstrate parallels between urban mapping techniques and natural language processing. To that end I have utilized a neural network to model the mapped structure of this neighbourhood.

A neural network is a computer program that manipulates values in a network data structure to encapsulate and reproduce patterns in data. To do so, the network is trained on collections of data, such as combinations of features that are in close proximity to one another.

Here, every grid cell acts as a training data instance for a neural network, analogous to a sentence in a natural language processing training set. The aspiration is that the trained model will uncover patterns within the mapped data that might not be immediately obvious, revealing compatibilities, outliers, and clustering trends that expose the character of different parts of the neighbourhood.

That said, my simple demonstration, and the programs behind it, are framed in terms of the relationships between words and two-dimensional data points. The results will be probabilities positioned against words. This is a long way from the poetic interpretations of urban places in Calvino’s *Invisible Cities*, but I’ll defer such sophistications till later.

One hundred metre grid cells are fairly coarse for this kind of map analysis. A finer grid would allow more detail. In the demonstration here I refined the method by recruiting a grid offset to effectively double the number of datapoints in the training data and to increase precision.

Here is part of the tabulation of the content of each grid cell. These are the grid cells, numbered 0-72. The columns in the table indicate whether or not the feature is present in that cell. As well as truncating the table I have simplified it. Behind each of these shaded squares is a number, e.g., if there are 3 hotels in a grid cell then the number in the table will be 3; if there is 1 bus stop the number will be 1.

Cell ID	ATM	Bar	Bus stop	Business	Café	Commemorative	Community	Education	Electric service	Entertainment	Gallery	Hairdressing	Health	Hotel	Open space	Post box	Religious	Restaurant	Retail	Sport	WiFi
0																					
1																					
2																					
3																					
5																					
6																					
7																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
17																					
18																					
19																					
20																					
21																					
22																					

After training the neural network on these cells it can predict likely combinations in any cell. For example, if a grid cell contains a health facility and a public open space then it is likely to also contain a bus stop, a business and a sports facility.

Considering my method of training the neural network on overlapping cells thatâ??s the same as saying that if thereâ??s a health facility near (within a 100 metre grid cell) to an open space, they are also likely to be near a bus stop, a business and a sports facility.

Here I show similar combination sets. The strength of the shading indicates the degree of certainty of the prediction that the features are in close proximity. The values attached to the predicted features are numerical, and the presence or absence of features is also a product of setting cut-off values.



It's of note that a trained neural network will always give a result. So, if an initial combination of features did not exist in the training set, the network will still generate a plausible combination consistent with the patterns it has developed in its network parameters. Hence the claim that neural networks are able to produce plausible outputs from inputs it has not been exposed to in its training set. That's relevant as we think about the application of these techniques to natural language generation.

What I have illustrated here is a kind of proximity modelling via a neural network. This exercise in training a neural network model on features is similar to applications of neural network training for natural language processing. The features in my demonstration are words (or tokens) arranged spatially in two dimensions. Imagine the grid cells as sentences in a document, and the features are the words within those sentences. In my neighbourhood map these words are arranged spatially in two dimensions. In text generation via automated natural language processing, the proximities are linear, i.e., adjacencies in one dimensional space.

Category

1. Artificial Intelligence

Tags

1. chatgpt
2. map
3. neural network
4. ordnance survey
5. Python
6. urban data

Date Created

September 23, 2023

Author

rcoyne99

default watermark