477

VARIATIONAL BEAUTY OF SPACE

Machine Intuition and Non-Linear Neural Aggregations

TASOS VAROUDIS; ALAN PENN

Bartlett School of Architecture, UCL, UK

ABSTRACT

Spatial networks have long been known for their internal spatial order or beauty (Hillier, 2007) and in the field spatial computation and analytics we employ a number of graph based methodologies in order to understand or extract intrinsic attributes of the urban fabric around us. In this research we take a complete different approach by looking at urban structure through the use of deep convolutional variational autoencoders with interesting results.

Autoencoders are an unsupervised learning technique in which we employ neural networks for the task of representation learning. Specifically, a neural network architecture imposes a bottleneck in the network which forces a compressed and generalization of the knowledge representation of the structure of space. This non-linear compression and subsequent reconstruction creates a unique set of features that are inherent of urban space. Our network is multiple layers deep in order to be able to encode basic spatial complexity and is build based on a convolutional network architecture which is inspired by biological processes similar to the connectivity pattern between neurons that resembles the organization of the animal visual cortex. Artificial neurons respond to real urban networks of London in a restricted region of the visual field, which partially overlap such that they cover the entire convolutional visual field. Convolutional layers apply a convolution operation to the input, passing the result to the next layer. Each convolutional neuron processes data only for its receptive field but the cascading nature of our network build a knowledge of more complex spatial relations as data move deeper. While convolutional neural networks are extensively used in supervised image classification the work presented in this papers is completely unsupervised.

In this paper we present the complete architecture of the deep convolutional network that was trained for months using data from the city of London, concluding with two significant outcomes extracted from the variational encoding and decoding processes. A non-linear clustering, or neural aggregation, of urban space based on the learned features and a generative urban network output based on the variational synthesis we call machine intuition.

KEYWORDS

Urban networks, autoencoders, unsupervised learning, convolutional neural networks, machine learning.

1. INTRODUCTION

There is a long tradition for human kind debating beauty, design, machine thinking and creativity. While we don't pretend to have a complete answer, one thing that we can see is the relation between a Platonic 'generalised description' of objects, or the 'essence' of them and the way that modern 'thinking machines' work (Sparkes, 1996; Tandy, 1997, LeCun et al., 1998b).

The motivation for this research was our own fascination for how our brain, after years of urban analytics and design, subliminally generates a number of urban networks, some of them 'optimised' for space syntax performance, every time we look at a road central line map. We wanted to break down this intuitive process and with the help of modern artificial intelligence try to reconstruct it from simplified parts.

Artificial intelligence (AI) had until today three somewhat distinct generations, artificial intelligence in cybernetics around the 1940s-1960s, connectionism in the 1980s-1990s, and the current rapid expansion under the terms machine learning and deep learning, beginning in 2006. Initially researchers worked on and solved problems that are intellectually difficult for humans but relatively straight-forward to encode in a sequence of mathematical steps. While extremely fast instruction-based computations are something that computational machines are good, the true challenge for artificial intelligence is solving concepts easy for people to perform but hard to describe formally. Problems like these are solved through intuition or some automated process that we can't easily describe or break down to simple steps, like recognising the model of a car from an image. The presented research is exclusively focused on these, hard to describe, 'intuitive problems' from the perspective of a spatial or urban designer by employing computational learning algorithms intended to be models of biological learning of the brain, the artificial neural networks (ANN).

The Neocognitron project (Fukushima, 1980) introduced a powerful model, in which a large number of computational units can become intelligent via the interactions with each other similar to how the brain functions. The focus was on processing images and the biological inspiration was the structure of the visual system of mammals. Today this idea has become the basis for the convolutional network (LeCun et al., 1998b) and visual processing through modern convolutional networks for object recognition (DiCarlo, 2013) is widely used by neuroscientists and computing engineers. For our research it's obvious that the concept of visually perceiving spatial complexity of urban networks will be used extensively.

2. URBAN NETWORKS, VISUAL PERSEPTION AND MACHINE LEARNING

The primary artificial neural network (ANN) model architecture that we use for this research is based on autoencoders. Autoencoders are an unsupervised learning model for training neural networks for the task of representation learning through the process of passing data via a restricted bottleneck. This bottlenecked design forces a compressed representation of the original input to be learned. In essence, an autoencoder is a neural network that is trained to attempt to copy its input to its output and composed from a pair of two connected networks, an encoder and a decoder. An encoder network takes an input, and converts it into a smaller, dense representation, which the decoder network can use to convert it back to the original input. Because the model is forced to prioritize which aspects of the input should be copied, it often learns useful properties of the data and discards irrelevant parts. While autoencoders have a long history (Bourlard and Kamp, 1988; Hinton and Zemel, 1994) and were traditionally used for dimensionality reduction or feature learning, our work is more aligned with recent theoretical developments. We have generalized the idea of an encoder-decoder beyond deterministic functions and we use the stochastic mapping or latent variable models that enable the use of autoencoders as generative models (Kingma, 2013; Rezende et al., 2014). With autoencoders (AE) we try to exploit the idea that learned data structures cluster around a low-dimensional manifold, or for more complex problems, a small set of such manifolds.

In basic terms, we train an autoencoder by engaging two main computational forces one of which tries to counter the other. The AE tries to learn a representation of a training example, in our case an image of a street network configuration of London (figure 3) that could be approximately recovered through the decoder part of the network. This action is directly opposed to regularisation-penalty, which is a designed constrain in the network that limits the capacity of the AE as we move towards its centre (figure 1). Copying straight the input to the output would be useless as well as losing all important information because of a badly implemented bottleneck. A fine-tuned combination is useful because we can force the hidden representation to capture information about the structure of the data-generating distribution. The important principle is that the autoencoder can afford to represent only the variations that are needed to reconstruct training examples.

Because of its unusual, bottlenecked design the very early successful research on autoencoders has been in dimensionality reduction. Hinton and Salakhutdinov (2006) successfully tested the deep learning and representation learning capacity of autoencoders. Deep autoencoders outperformed traditional Principal Component Analysis (PCA) and their internal 'learned' representation was also qualitatively easier to interpret with their manifolds well separated in clusters. One part of our work links directly to the fact that neural networks, and in extension AEs, are non-linear and their dimensionality reduction capabilities are more efficient than a PCA (figure 1). In the later parts of our work we present early hints of the generalisation capabilities of mapping urban network complexity to the lower-dimensional space through autoencoders.

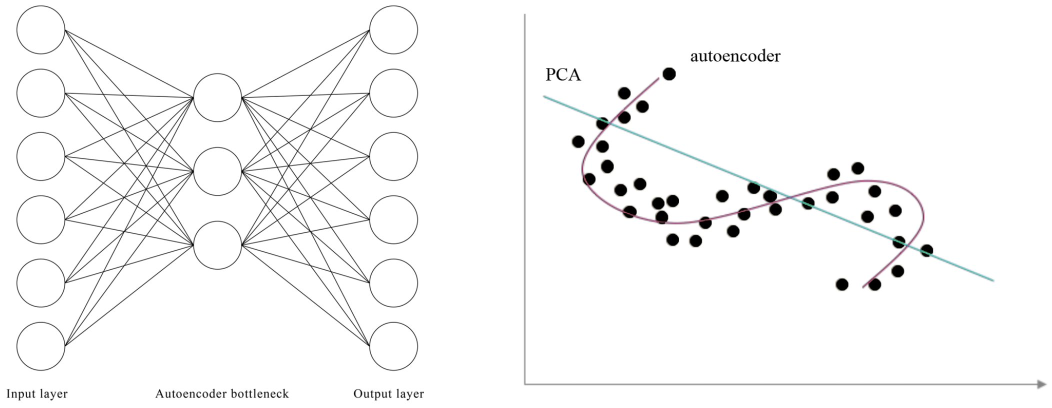

Figure 1: Simple autoencoder design (left). Non-linear model-representation capacity of autoencoders versus PCA.

For our pilot research, the generalisation capabilities and the non-linearity is important as the complexity of urban configurations, even in the form of a small image with a handful of streets (figure 3), is very complex. The generating distribution in most cases can't be 'compressed' into a single low-dimensional manifold. A simple encoder part of an AE would try to learn a local translation from input to encoded space. In our pilot case, manifolds can have very complicated structures that can be difficult to capture from only local interpolations. Therefore, we not only started building deeper models (Hinton and Salakhutdinov, 2006), with multiple layers (i.e deep learning), but also explore a specific form of autoencoders called Variational Autoencoders (VAE).

Variational autoencoders (VAE) provide a probabilistic way for encoding an observation in latent space (Everitt, 1984; Kingma, 2013; Rezende et al., 2014). Thus, rather than building an encoder that outputs a single value to describe each latent state feature, the encoder describes a probability distribution for each latent feature (figure 2). Another important element of VAEs, which is not present in simple AEs, is that their encoded latent spaces are by definition continuous, allowing easy random sampling and interpolation. We extensively used this feature later in our work and is considered part of the generative modelling strengths of VAEs. In order for the encoder to generate this probabilistic model the output is not one vector of numbers like AEs but one vector of means and another vector of standard deviations from that mean. This stochastic generation implies, that even for the same input, while the mean and standard deviations remain the same, the actual encoding will vary a small amount on every single pass. In space syntax terms, or urban configurations logic, this can translate to having a number of similar road networks that exhibit the same analytical values (integration or choice for example) as described with traditional space syntax analytics.

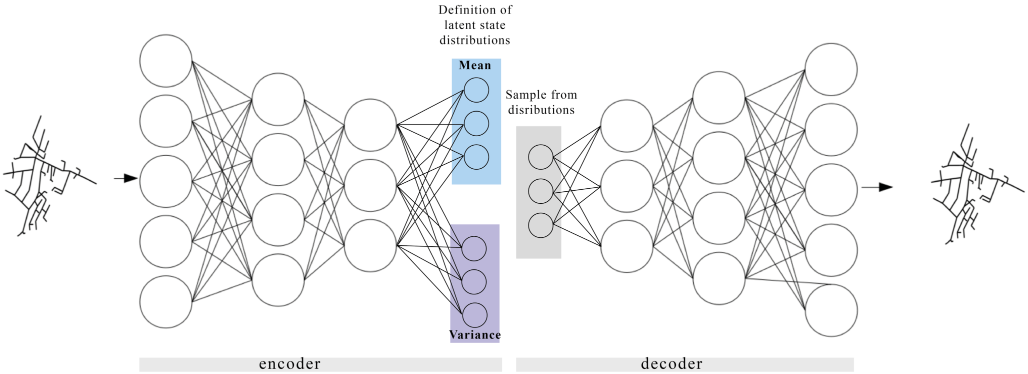

Figure 2: Building blocks of a variational autoencoder.

Next we addressed the most fundamental building block of our model, which deals with the idea of learning from images of existing road network examples similar to how urban designers observe the world. Images are a matrix of values corresponding to the intensity of light (white for highest intensity, black for lowest intensity) at each pixel value and in order to feed a 2D image into a network we need to 'flatten' it into a vector, thus losing all spatial relationships within the 2D data. Preserving the spatial arrangement of features (pixels) is important not only because we will be using images in our training set, as many computer scientists do, but also because in our space syntax field spatial configuration is fundamental as we deal with maps of two-dimensional arrangements of streets.

Convolutional neural networks (LeCun, 1989) are a tailor-made architecture of neural networks for processing data that have a grid-like topology similar to the images that we are working with. Most of the latest successes in machine learning and artificial intelligence are heavily dependant on convolutional neural networks (CNNs), like self-driving cars (Bojarski et. al. 2016). Our intuitive connection to convolutional methods comes from our space syntax and architecture knowledge. Visual perception ideas from Gibson (1979) and the use of visual material for teaching was the initial inspiration in trying to understand how we could build an artificial system that can learn spatial configurations. The first convolutional networks began with pure neuro-scientific experiments before any computational models were developed. They were first designed in order to understand how mammal's vision system works (Hubel and Wiesel, 1959). They recorded the activity of individual neurons in cats and observed how neurons in the cat's brain responded to images projected in precise locations on a screen. They found that the early visual system responded most strongly to very specific patterns of light, like vertical lines, and not at all in other situations. These findings were a strong influence in contemporary deep learning models.

A convolution layer can be simply defined as a) a 'window' by which we examine a subset of the 2D image, b) the operation of scanning the entire image looking through this window. This window can be parameterized to look for specific features within an image, like lines or simple shapes. This window is technically called a filter, since it produces an output filter that focuses solely on the regions of the image that exhibits the feature it was searching for. The final output of a single convolution is called a feature map. Similar to how traditional neural networks use weights and activations, we project the convolutional filter onto the image with a combination and activation between the filter and the pixel values. Our filters are not explicitly defined and are instead parameterized in the network design phase and then we let the network learn the best filters to use during training.

During our research into building an ANN that can learn a very basic understanding of the urban spatial configuration, we experimented with stacked layers of convolutions with a single activation and we observed how they perform better by learning more intricate patterns within the features mapped in the previous layer. This allowed our model to identify general patterns in early layers, than focus on the patterns within patterns in later layers. Convolutional filters are indeed capable of considering locality of features and the spatial relations (in terms of the 2D image). Intuitively, it seems similar to spatial analysts investigating multi-centre models of a city in low radii, that builds into a stronger single centre network where all local centres are merged into one large radii analysis. Internally the convolutional network learns to use earlier layers to extract low-level features and then combine these features into more rich representations of the data. Because later layers of the network are more likely to contain specialized feature maps, as we move deeper into the layers of our network we also increase the number of total filters or feature maps, a common practice in the ANN research community.

While designing and building the core model as a fine-tuned assemble of the neural ideas discussed above was the ultimate goal, the correct representation of the input dataset played an equally important role. The ultimate battle in the next two sections was between building an interesting model that can also be trained in reasonable time. In this process the size and shape of the input training set change a number of time. This reasonable time ended up being a number of months with a high-end single GPU computer.

3. DATASET AND SPATIAL REPRESENTATION

The main focus of our work was to understand the urban spatial complexity in terms of neural representation knowledge and build a pilot experiment. For this reason, we generated a dataset consisting of 300000 images created from parts of London's metropolitan street network as depicted in figure 3. The process of generating the dataset had the following steps: a) we selected every street segment individually, b) for every segment we run an analysis of step depth with depthmapX (Varoudis, 2012), c) created a subnetwork with the closest 100 segments, and d) created a black and white image centred around the 'central' segment from step 'a'. Grayscale images have a single value for each pixel (one colour channel) and for our machine learning experiments we rescaled the light intensity values to be bound between 0.0 and 1.0, as it's traditionally done. This collection of subnetworks is likely to have a different width, height and orientation so it needs to be centred and white padded before it's saved as an image.

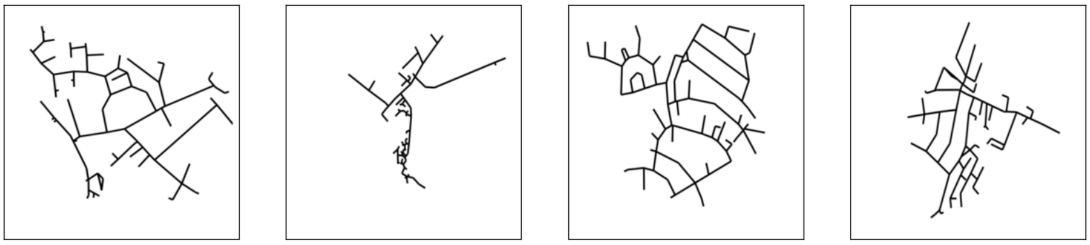

Figure 3: Examples from the training set. Four subnetworks extracted from London.

The method of generating the images 'by moving one segment at a time' also introduces visual overlaps between images and a continuity of the spatial complexity description. Something that our own intuition found positive for the artificial learning that follows. This overlapping of local neighbourhood map in conjunction with the convolutional filter mechanism of a) 'scanning with overlaps' and b) 'stacking feature maps' was positive for our project. Finally, the images were saved in a two-dimensional lossless format with a single colour channel so a pixel can go from white to completely black and later for training efficiency the images (2d) were converted and packed in a numerical three-dimensional array (3d) with the third dimension representing the multiple examples.

4. TESTED NEURAL MODEL

Throughout the exploration of different neural techniques we tested a large number of models, from simpler autoencoders to non-convolutional variational examples, all with varying number of layers (depth). Our final model could be called a "Deep Convolutional Variational Autoencoder". It encapsulates all the step-by-step experimentation presented in the easier parts of the paper and produced some promising results.

The general concept is that a) the first step in encoding the visual logic of an urban configuration is dealt by a number of cascading two-dimensional convolutional layers with increasing number of filters, as we move to deeper layers, b) the knowledge translation to a latent space happens through a variational model encoding the mean and deviation from the mean for every feature and c) the decoding process for the creation of the output example is a reverse convolution or deconvolution network (Zeiler et. al., 2010) were an image is produced and tested against the input. In detail the network design is 6 layers deep before we reach the end of the encoder, while we reverse the layers for the decoder. Using deep encoders and decoders offers many advantages as depth can exponentially reduce the computational cost of representing some functions and can decrease the amount of training data needed to learn some functions. Hinton and Salakhutdinov tested deep autoencoders with better knowledge compression results than corresponding shallow or linear autoencoders (Hinton and Salakhutdinov, 2006). Furthermore, our entire network is trained as a whole. The parameters of the model are trained via two loss functions: a reconstruction loss forcing the decoded samples to match the initial inputs, mean-squared error in our case, and the Kullback-Liebler divergence (KL), which can measure how closely the variables match a unit Gaussian distribution (Yu et. al., 2013) between the learned latent distribution and the prior input distribution, acting as a regularization term which helps in learning nicely defined latent spaces and reduces overfitting.

All the experiments were implemented mainly in Python by using well established open source tools and libraries like Google's Tensorflow. The training process was very lengthy and time consuming and for the presented example a stable state reached after the epoch 10000 that took a single GPU computer over a month to compute (one epoch equals to passing all the images of the subnetworks of London through the training process). While the scope of this paper is to present a good preliminary example of our work, we have to admit that networks of this design can be very unstable and we had a large number of failed attempts.

5. NEURAL AGGRIGATION / CLUSTERING

The first result of our neural exploration into urban spatial configurations comes from the combined power of autoencoders and variational inference. As presented earlier, the aim of our encoder network was to produce a non-linear knowledge compression from existing spatial configuration in a meaningful way. Driven by our interest in urban morphology and the patterns of urban systems we wanted to be able to produce a visualisation that matches our understanding of urban configurations. Following the training of the model as a whole we disconnected the decoder part of the network, froze the weights of all neurons and used the encoder part in order to output the encoded values of a random set of images from the London dataset. Figure 4 (left) presents the generated latent space that our network has learned. Every point represents a learned latent representation of a single urban subnetwork of London. By extracting nearby point's original input image we see that our network maintains the morphological similarity of nearby encodings on the local scale (figure 4, right). While these are very preliminary results, our intuition sees a similarity in the 'perpendicular' street network of the two images as opposed to the more 'organic – tree structure' of the other pair of street network images. In essence we see an efficient method for learning a latent space representation for spatial morphology clustering. These results are even more encouraging because our model was trained as a standard variational autoencoder with no injection of any special clustering parameters. The equilibrium observed in figure 4 was achieved by the combination of the cluster-forming nature of the reconstruction loss and the distinctive packing nature of KL divergence. We expect future research to include training end-to-end networks where the decoder part also includes parameters to evaluate clustering accuracy in order to explicitly test the clustering capabilities of our design.

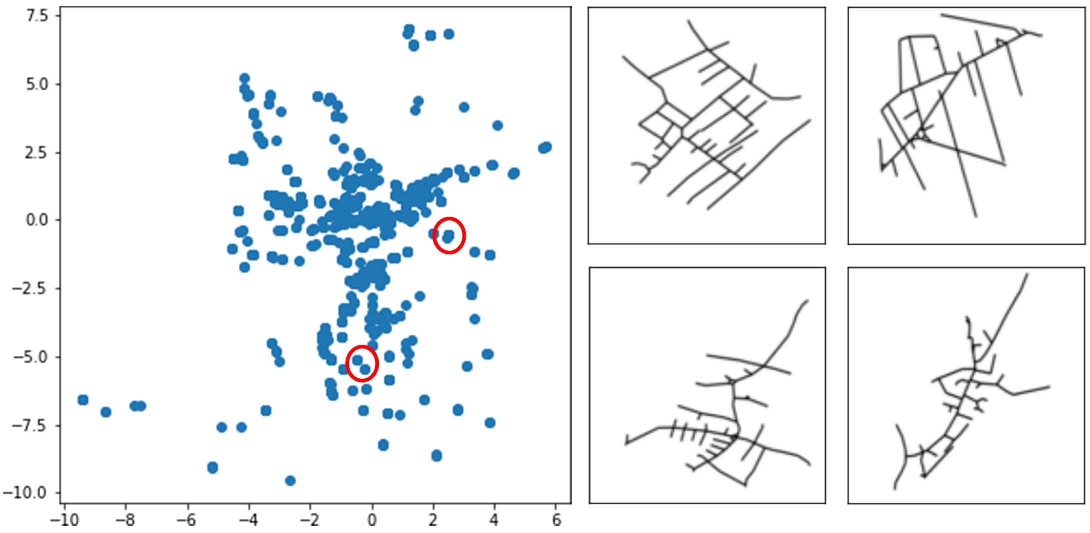

Figure 4: Representation of a subset of London's sub networks in latent space (left). Clusters of morphological similarities (right).

6. MACHINE INTUITION / GENERATION

The second part of our research focused on the generative properties of our model. In the previous sections we described how VAEs' latent space is a continuous definition and not fragmented as with other AEs. The generated latent vectors roughly follow a Gaussian distribution because of the KL divergence influence during the training phase. This allows the implementation of random sampling and interpolation by simply generating a vector of two-dimensional random numbers originating from a Gaussian distribution. In order to construct this sampling generator we took the trained network and use the decoder part as a stand-alone model. The decoder network takes two values, representing the "X, Y" coordinates of the learned latent space and outputs an image which is the product of the 6 cascading layers of deconvolutions (figure 5). This image, depending on success of the training and modelling process vary from a random grey-scale assortment of pixels to a perfect reconstruction of the input map (theoretically possible).

As discussed in the dataset generation section above our London training set is comprised of 300000 images representing black and white snapshots of the cities' urban street morphology. Each image or datapoint has thousands of dimensions (i.e. pixels), and our "Deep Convolutional Variational Autoencoder" model's task was to capture the morphological inter-dependencies between pixel groups (segment lines, arraignments etc.) and organise the learned knowledge into cascading meaningful generalised representations.

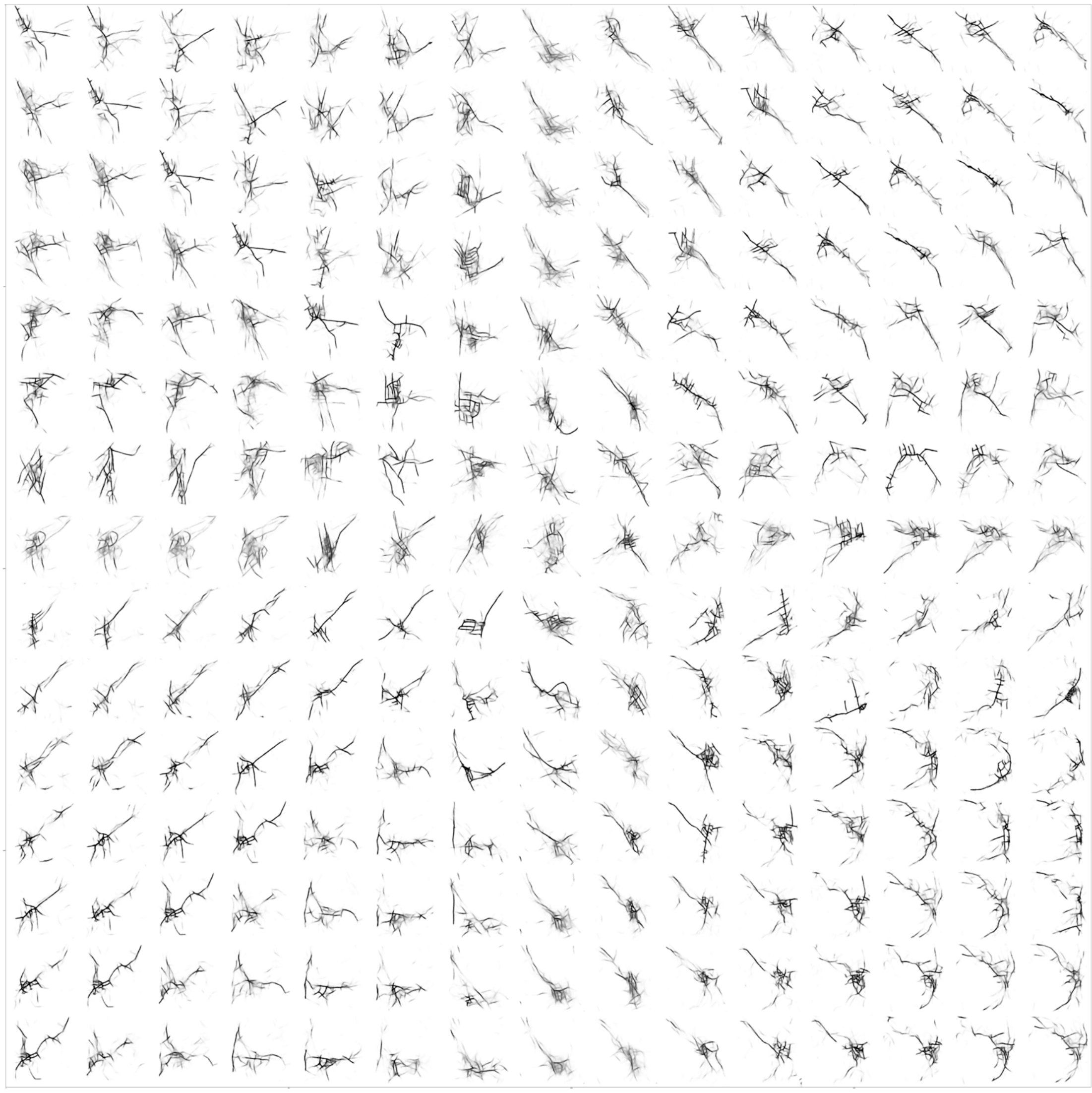

Figure 5: A set of generated new spatial configurations.

In figure 5 we present the generated output that is based on a random grid-based sampling of ‘X, Y’ coordinates of the continuous latent space, 15 by 15 samples. What we ultimately want from our machine intuition is not to produce examples exactly like the ones already in our training dataset, i.e. copies of the street network of London, but to synthesise new unseen images that encapsulate the spatial dynamics from the learned London morphology. From this preliminary output we can clearly see that the model can produce a large number of very interesting variations and also demonstrate that the intuitive morphological characteristics, which could be picked up by an urban analyst, are organised nicely in the learned latent space. The well structured morphological interrelations found in figure 5 and 6 confirm the potential for our neural model to generalise and learn a very basic understanding of urban morphology.

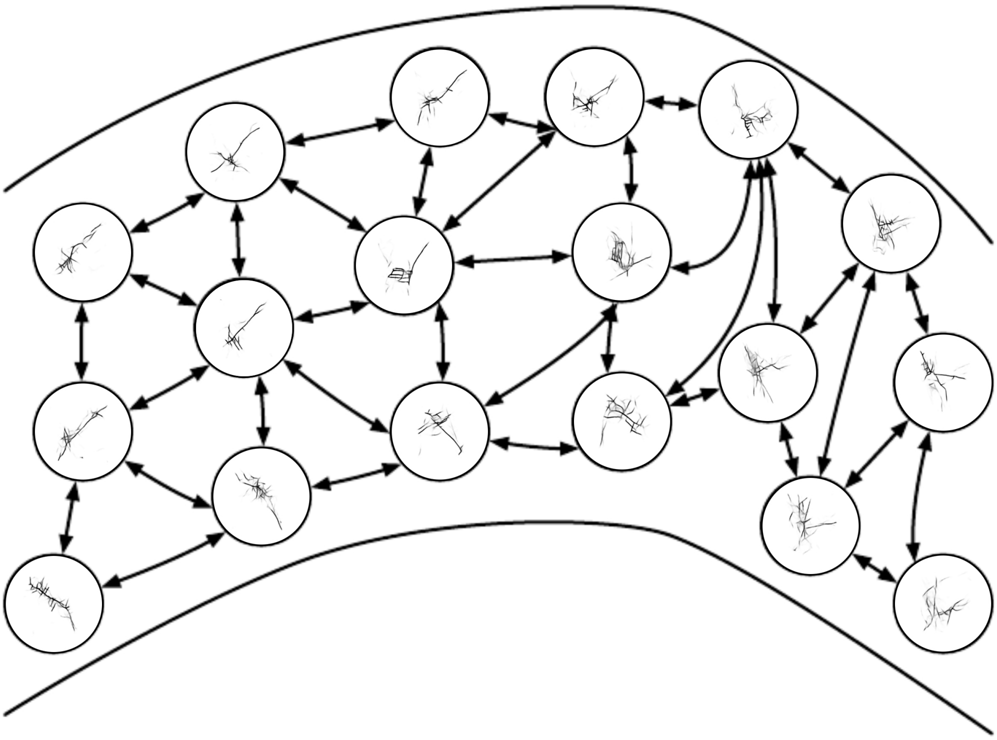

Figure 6: Sample mental manifold of urban spatial configurations from our neural model.

7. CONCLUSIONS

With our work we demonstrated how a deep convolutional variational neural model could be trained to learn inherent morphological characteristics of the urban street network. Using images of London street network broken up in small neighbourhoods we trained a model that was able to 'see' by imitating neural processes similar to our visual system and extract generalised representations, or knowledge, completely unsupervised.

The two outcomes, namely the neural clustering in latent space and the generative machine intuition, are highly interrelated but they both give us insights to two unexplored areas in the filed of spatial morphology. First, in experimenting with machine intuition in urban morphology and design, and second the research of how performance-based space syntax analytics can be hybridised with the help of neural networks. These focused explorations will also help to overcome and understand better the shortcomings that we had during the numerous failed experiments. The results presented in figure 5 and 6 give a clear view of the internal mental model of the trained neural network and could be understood intuitively. Although it could not be tested, the morphological mental evolutions depicted in figure 5 appear to be organic from a spatial designer's perspective.

8. ACKNOWLEDGEMENTS

The research wouldn't be possible without the assistance of Bartlett's IT team and the support by UK's Engineering and Physical Sciences Research Council (EPSRC) fund EP/M023583/1.

9. REFERENCES

Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. Biological Cybernetics, 59, 291–294.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.

DiCarlo, J. J. (2013). Mechanisms underlying visual object recognition: Humans vs. neurons vs. machines. NIPS Tutorial.

Everitt, BS (1984). An Introduction to Latent Variables Models. Chapman & Hall. ISBN 978-9401089548.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36, 193–202.

Gibson, J. (1979), The Ecological Approach to Visual Perception, Boston: Houghton Mifflin.

Hinton, G. E. and Zemel, R. S. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In NIPS'1993 .

Hinton, G. E. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507.

Hillier, Bill. Space is the machine: a configurational theory of architecture. Space Syntax, 2007.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurons in the cat's striate cortex. Journal of Physiology, 148, 574–591.

Kingma, D. P. (2013). Fast gradient-based inference with continuous latent variable models in auxiliary form. Technical report, arxiv:1306.0733.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Proceedings of the International Conference on Learning Representations (ICLR).

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998b). Gradient based learning applied to document recognition. Proc. IEEE.

LeCun, Y. (1989). Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In ICML'2014. Preprint: arXiv:1401.4082.

Sparkes, B. (1996). The Red and the Black: Studies in Greek Pottery. Routledge

Tandy, D. W. (1997). Works and Days: A Translation and Commentary for the Social Sciences. University of California Press.

Varoudis, T. (2012). depthmapX - Multi-Platform Spatial Network Analyses Software. OpenSource.

Yu D., Yao K., Su H., Li G. and F. Seide, KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 7893-7897.

Zeiler, Matthew D., Dilip Krishnan, Graham W. Taylor, and Rob Fergus. "Deconvolutional networks." (2010): 2528-2535.