Inference and Discovery in Remote Sensing Data with Features Extracted using Deep Networks

Isabel Sargent[1][2], Jonathon Hare[2], David Young[2], Olivia Wilson[1], Charis Doidge[1], David Holland[1], and Peter M Atkinson[3]

[1] Ordnance Survey, 4 Adanac Drive, Southampton, SO16 0AS, UK
[2] University of Southampton, Highfield, Southampton, SO17 1BJ
[3] Lancaster University, Bailrigg, Lancaster LA1 4YW

Abstract. We aim to develop a process by which we can extract generic features from aerial image data that can both be used to infer the presence of objects and characteristics and to discover new ways of representing the landscape. We investigate the fine-tuning of a 50-layer ResNet deep convolutional neural network that was pre-trained with ImageNet data and extracted features at several layers throughout these pre-trained and the fine-tuned networks. These features were applied to several supervised classification problems, obtaining a significant correlation between the classification accuracy and layer number. Visualising the activation of the networks' nodes found that fine-tuning had not achieved coherent representations at later layers. We conclude that we need to train with considerably more varied data but that, even without fine tuning, features derived from a deep network can produce better classification results than with image data alone.

Keywords: Remote sensing, Deep learning, Feature extraction

1 Introduction

To serve its public task and meet customers' requirements, Ordnance Survey, Britain's mapping agency, interprets the landscape to create digital representations portraying and characterising human-made (e.g. pylons, buildings, roads) and natural (e.g. rivers, moorland, boulder fields) real-world objects for a wide range of applications such as routing, asset management, planning and geospatial modelling. Increasingly diverse and subtle objects and landscape characteristics are required such as the location of hedgerows or the age of buildings. In a rapidly changing commercial environment, it is essential that mapping agencies build approaches that can respond quickly to customers' changing needs - both in response to customers' requests and in anticipation of their future requirements.

Both field survey and remote sensing survey are used to create and maintain detailed mapping products. The majority of information extraction from remote sensing data is done so by expert interpreters using manual processes. For example, digital plotting using stereo imagery is employed to define the perimeter of real-world objects. With better instrumentation and the pressure to improve data currency, the volume of data being acquired is increasing. Clearly, manual capture methods will struggle to scale with increased data and demand. A few, rules-based, automatic approaches are also used but as data and products develop, these rules need to be manually updated, usually at considerable cost.

Machine learning offers an approach that allows models to develop as the real world, data and customer needs change. Recent breakthroughs in image interpretation have used deep learning [9, 8], which has the ability to extract "hierarchies of representations" [4]. However, most applications focus on only the the final layers of the network: either training the network to find the classes of interest [10, 2] or using the features extracted by the penultimate layer as inputs to a shallow learning algorithm [7, 6].

Given the complexity of the model being learned [12], deep networks require a considerable amount of data. Adequate labelled data are rarely available and so it is impractical to train a deep model for each customer requirement. Instead, we aim to extract, from remote sensing data, features that are generic to our existing and future inference problems. Our hypothesis is that we can decode the signatures of human activities and non-human processes that have shaped the landscape - Bengio *et al.*'s "underlying explanatory factors hidden in the observed data" [1] - to extract descriptors of the landscape. These descriptors can then be applied as input features to infer the presence of real-world objects or landscape characteristics . As well as this *inference* goal, we conjecture that these features will serve a second, *discovery*, goal by providing new ways of describing the landscape. For example, the features may represent the era in which regions were developed (as is evident in the layout of roads and buildings) or they may pertain to the risk of flood inundation (as results in identifiable patterns of vegetation). This presented work focuses on our inference goal by testing extracted features against a set of classification problems. We also begin to address our discovery goal by interrogating the weights in the trained networks.

2 Approach

We used the 3-band aerial imagery that makes up our OS MasterMap® Imagery Layer (Imagery Layer) product. These images are orthorectified to 25 cm spatial resolution and are available for all of Great Britain. We also have a topographic vector product, OS MasterMap® Topography Layer (Topography Layer), that portrays real-world objects, such as buildings, roads and fields, as area, line and point vectors with a range of descriptive attributes. Because the aerial imagery is orthorectified using detailed terrain and object height data, Topography Layer has a strong correspondence to Imagery Layer.

In essence, our problem is one of unsupervised learning in that we want to transform our input data in such a way that draws out factors that we have only loosely defined in advance - the underlying explanatory factors, or descriptors, of the landscape. Unsupervised targets can be difficult to specify and so we opted to set a supervised target for training - Topography Layer data. Our assumption is that this target will 'guide' training towards forming a hierarchy of representations of the factors that, in combination, define the landscape as described in Topography Layer.

From Imagery Layer, we extracted overlapping patches of 224 × 224 pixels, corresponding to a square of 56 m × 56 m on the ground. Each patch was labelled with attribution taken from the vector feature in Topography Layer that overlaid its centre. To achieve this, we combined the 'Theme' attributes for the vector feature into a string resulting in the following 22 classes: {Roads Tracks And Paths; Land; Water; Rail; Buildings; Structures; Heritage And Antiquities; Land,Water; Rail,Roads Tracks And Paths; Buildings,Structures; Roads Tracks And Paths,Structures; Land,Structures; Land,Roads Tracks And Paths; Roads Tracks And Paths,Rail; Structures,Water; Water,Structures; Rail,Structures; Water,Land; Roads Tracks And Paths,Water; Land,Rail; Heritage And Antiquities,Land; Buildings,Roads Tracks And Paths}.

To investigate the features learned at depth, we chose to adapt a 50-layer ResNet [5] (ResNet50) that had been trained on the ImageNet dataset [11] (weights available in Keras [3]) by performing a fine-tuning operation to enable the network to better learn internal representations of our aerial imagery. With limited processing capacity, fine-tuning allowed the re-use of learned low-level image features, such as edge and colour filters, focussing the computational effort on tailoring the network to a new data domain. Fine-tuning was performed by fixing all layers of the network except the last one for 50 epochs, and then training all layers for a further 50 epochs. We used a stochastic gradient descent optimiser with an initial learning rate of 1e-4 and momentum of 0.8 for fine-tuning. Each epoch consisted of approximately 1.2 million image-class pairs sampled from the Southampton area in the South of the UK (containing a mix of water, urban and rural settings). The training pairs were sampled randomly against the same underlying distribution as the training region. Because of the vast size of the data used for training it is extremely unlikely that the network saw the same training instance more than once during the entire training process. With a batch size of 32, training the last layer alone took slightly over 9000s per epoch on a single Titan X GPU, and training the entire network took around 23000s for each epoch. Overall training accuracy was 84.3% and validation accuracy (on 16000 image-class pairs taken from a region that was not used during training) was 78.2%.

Towards our inference goal, we performed a series of trials of the features extracted from the ResNet50 networks for a small labelled dataset from Lincolnshire in the East Midlands of the UK. Three different classification problems were investigated: 1) finding inland water; 2) finding roads and tracks; and 3) differentiating metalled roads and tracks, unmetalled roads and tracks and a mixture of other classes. These classification problems were selected from a wider set of manually labelled data because they were particularly difficult for our rules-based approaches. We did not perform any further training of the deep networks for these trials. It was noted that they were similar, but not identical, to classes in the target data.

A patch of 224 × 224 pixels, centred on the location of the class label, was extracted from the image data. For each classification problem, an 'other' class was drawn from labelled patches not currently being used (these included classes such as ‘scrub’, ‘solar panels’). The patches were shuffled and balanced such that the same number of patches was available for each class, including the ‘other’ class, in each trial. The numbers of examples were approximately 100, 90 and 40 for problems 1), 2) and 3), respectively. Each patch was forward-propagated through both the ImageNet and fine-tuned ResNet50 networks and the maximum activation at each node was returned forming a feature set for each selected layer. For comparison, we also created a feature set from values taken directly from the central 12 by 12 pixels of each patch, which resulted in a vector of similar magnitude to feature sets from the later layers of the deep networks. Feature sets were input to linear support vector machine classifiers, which were trained against each of the 3 classification problems. For each classification problem and feature set combination, training was performed over 10 different folds of the data. For 5 of these, the regularization parameter, C, was tuned using 10 folds of a separate verification dataset. For the other 5 folds, the C parameter was set to 1.0. For each test, the average classification accuracy was taken over the 5 folds of the data. The resulting accuracies are compared in Figure 1. Towards our discovery goal, we studied the nodes’ receptive fields by visualising the parts of the data that most activated each node using a similar method to [14].

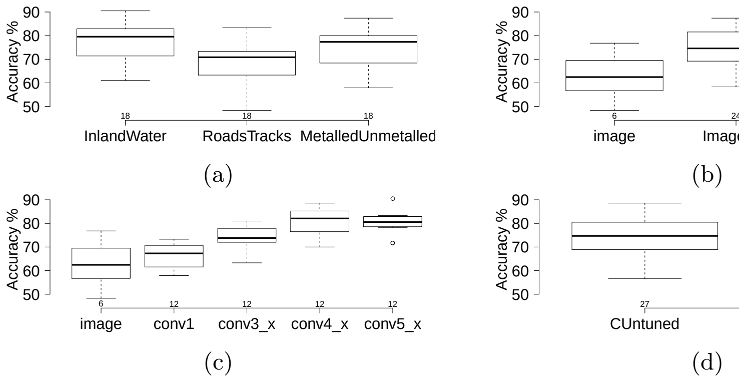

Fig. 1: Boxplots showing the accuracy against (1a) the classification problem, (1b) the network weights, (1c) the layer from which the features were extracted and (1d) whether or not the C parameter was tuned. In (1b) and (1c), ‘image’ refers to the trials using pixel values as input features. ‘conv1’, ‘conv3_x’, ‘conv4_x’ and ‘conv5_x’ are regions increasingly deep within the network as described in [11]. The number of trials for each plot is given above the x-axis. Outliers are represented by dots.

Over the 54 tests, the classification accuracy averaged 73.8%. It is evident in 1b that using features derived from the deep networks increases classification accuracy but that there is only a small improvement with fine-tuning. In 1c, deeper features tend to result in higher classification accuracy and a large correlation (r=.70) was found between layer number and classification accuracy.

The early layers of the network responded to our data as would be expected for any image data, having nodes that are activated by edges and colours and, at intermediary layers, particular shapes such as circles. However, although we observed a divergence between the receptive fields in the ImageNet and the fine-tuned network, at later layers, no discernable label could be applied to the activations of the nodes, even with fine-tuned weights.

3 Discussion and Conclusions

We have initiated research into extracting generic features from remote sensing data and applied these to our inference and discovery goals. Topography Layer provided labels for a large training dataset. However, the chosen 22 classes were poorly balanced resulting in few examples for some classes. For future work, we are developing a more balanced set of labels based on Topography Layer.

The large correlation between accuracy and layer number is evidence that more useful features are learned deeper within the network. Even features taken from deep layers of the network trained only with ImageNet achieved promising classification accuracies. Our investigation of the receptive fields demonstrated that early layers of the network represented generic image features, yet we were not able to interpret the representations at later layers even following fine-tuning. Further, it is likely that concepts are represented as a combination of activations within the network (and not just high activations). Future research will therefore investigate how the whole layer represents the input data using techniques such as clustering and dimensionality reduction on outputs at each layer.

Most image datasets applied to deep learning, such as the ImageNet challenge data, comprise scenes in which the labelled objects are well framed within the view. Even aerial image benchmark datasets, such as UC Merced Land Use Dataset [13], feature objects centred within the frame. In contrast, region- and country-wide aerial imagery comprise continuous real-world features that occur with equal probability anywhere within the view, at any orientation. Further, the kinds of real-world objects that remote sensing is often concerned with (roads, fields, buildings, etc.) are extremely variable in scale and shape. One way of interpreting a trained CNN is as a set of templates that represent the most commonly encountered structure within the dataset. The variation in position, orientation, scale and shape presents a particular problem for feature learning from remote sensing data. Thus, the training data for our classification problems were not typical of most patches from remote sensing data because the objects were centred in the patch, even for the 'other' class. Whilst the classification accuracy within these tests was promising, when we applied the classifier to whole images the results were noisy and demonstrated that more typical training examples are needed to develop a usable inference tool for our data. This principle is also pertinent to training and fine-tuning a deep network and may explain why fine-tuning did not result in interpretable representations in later layers.

To date, we have not extracted the underlying explanatory factors that we desire from our remote sensing data. We conclude that greater consideration of the training data is required to ensure that datasets, for both deep networks and shallow inference networks, protray real-world objects with the full variance of position, orientation, scale and shape.

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35(8), 1798–1828 (2013)
2. Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L.: Land use classification in remote sensing images by convolutional neural networks. ArXiv e-prints abs/1508.00092 (2015), `http://arxiv.org/abs/1508.00092`
3. Chollet, F., et al.: Keras. `https://github.com/fchollet/keras` (2015)
4. Deng, L., Yu, D.: Deep learning: Methods and applications. Tech. rep., Microsoft Research Lab - Redmond (May 2014), `https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/`
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Hu, F., Xia, G.S., Hu, J., Zhang, L.: Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sensing 7(11), 14680–14707 (2015)
7. Huang, F.J., LeCun, Y.: Large-scale learning with svm and convolutional nets for generic object categorization. In: Proceedings of Computer Vision and Pattern Recognition Conference (2006)
8. Le, Q.V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., Ng, A.Y.: Building high-level features using large scale unsupervised learning. In: Proceedings of the Twenty-Ninth International Conference on Machine Learning. Edinburgh, Scotland (2012)
9. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning. pp. 609–616 (2009), `http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf`
10. Mnih, V., Hinton, G.E.: Learning to label aerial images from noisy data. In: International Conference on Machine Learning (2012)
11. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115, 211—252 (2015)
12. Vapnik, V.N.: Statistical learning theory. New York: Wiley (1998)
13. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS) (2010)
14. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I. pp. 818–833. Springer International Publishing, Cham (2014)