Towards Geographic Aware Neural Networks for Geospatial Vector Data: A Case Study on Land Use and Land Cover Classification

Marvin Mc Cutchan[1,2], Ioannis Giannopoulos[1,2]

[1]TU Wien, Gußhausstraße 27-29, Vienna, 1040, Austria
[2]Institute of Advanced Research in Artificial Intelligence (IARAI), Untere Viaduktgasse 16, Vienna, 1030, Austria

Abstract

Remotely sensed imagery is a well-established data source for spatial predictions, however,it comes with some disadvantages. Here, we explore how another data source could be used for spatial predictions, namely, geospatial vector data, by looking at a specific case study: LULC classification. We show how vector data can be encoded for an artificial neural network, making it geographicly aware and enable it to predict LULC classes. We use two different encoding schemes as well as two different artificial neural network architectures. Our results suggest that geospatial vector data can be used for LULC classification and that the type of encoding and artificial neural network plays a significant role.

Keywords

Geospatial vector data, Deep Learning, Vector encoding, Geospatial Semantics,

1. Introduction

A series of spatial prediction tasks, such as land use land cover (LULC) classifications, are based on remotely sensed imagery as input data. Electromagnetic reflectance of the physical surface enables to gain knowledge of the Earth and to base prediction models based on it. However, using remotely sensed imagery comes with two major disadvantages: (1) Electromagnetic reflectance does not capture all relevant geographic phenomena, such as land use, and may even be inconsistent across multiple sensor types and scenes [1]. (2) The raster model introduces two assumptions into the model: That the world can be tessellated in a regular manner and is homogeneous within a pixel [2]. As such, the raster model also does not maintain topological characteristics of real-world objects [3]. In this work we probe the usage of another data source for this purpose, namely geospatial vector data, which is expressed in geometric primitives (polygon, line strings, points) and attributes (e.g. type of object, height, etc.) describing corresponding geographic objects (geo-objects).

Geospatial vector data has attributes that make it a challenging task to be used as input for a deep neural network. A deep neural network requires structured and well-defined data, whereas vector data is highly irregular. For example, geo-objects might be scattered irregularly in a region and contain descriptions such as *coffee shop*, or, *supermarket*. To be able to create a deep learning model, the artificial neural network (ANN) architecture needs to be able to deal with this irregular geographic and semantic data. Here, we use LULC classification as a case study to explore how geospatial vector data can be used in combination with ANNs for spatial prediction tasks. To be able to deal with the irregular nature of geospatial vector data, we employ two encoding techniques which transform the vector input data to a feature space that can be used by an ANN. The outcome of this work is two-fold: (1) We show that geospatial vector data can be used for LULC classification; (2) we show that the type of encoding used for transforming geospatial vector data as well as the ANN architecture have an impact on the final classification performance. The first section briefly discusses the problems which appear when using geospatial vector data as input for ANNs. Then, we describe the experiments carried out and provide the preliminary results. In the end, we discuss our findings and outline future research.

2. Geospatial Vector Data and Artificial Neural Networks

Unlike raster data, (geospatial) vector data is irregular. A raster dataset tessellates a region of interest (ROI) into pixel, which is an ideal data model to be used for various deep learning architectures, as it is already formatted in a desired regular format that can be forwarded to an ANN right away. In contrast, geo-objects, such as a *house*, a *street*, a *garden*, a *river*, etc. are described as geometric

CDCEO 2021: 1st Workshop on Complex Data Challenges in Earth Observation, November 1, 2021, Virtual Event, QLD, Australia.
marvin.mccutchan@geo.tuwien.ac.at (M. M. Cutchan); ioannis.giannopoulos@geo.tuwien.ac.at (I. Giannopoulos)
0000-0003-3364-4163 (M. M. Cutchan); 0000-0002-2556-5230 (I. Giannopoulos)

primitives. A polygon for example might describe the shape of a house, a linestring a street, or, a point geometry the location of a point of interest. Next to these geometric descriptions, geo-objects contain attributes, which describe the corresponding geo-object. Such attributes can be, for example, the type of geo-object, such as a *coffee shop*, a *bar*, or a *university*. This data model may allow for a precise description of geographic and extensive auxiliary information (attributes) but is irregular and cannot be used in its native form by ANNs. This irregularity has two aspects: (1) a geometric and (2) a semantic aspect.

Ad 1: Training samples would have a varying number of geo-objects. ANNs however require input vectors with a fixed number of input features. Furthermore, geo-objects come in three different geometric primitives – ANNs can also not deal with this aspect, as they do not have the capabilities to represent this geometric information without loss.

Ad 2: Semantics of a geo-object, which describe it, are often of nominal scaling (e.g. *restaurant*, or *university*). However, an ANN requires numerical input, not a nominal one. One may simply transform each nominal value to a token number. However, this would introduce a flaw into the ANN-based model as the distance between any two token values will introduce a wrong semantic similarity in feature space. For example, *coffee shop* might have a value of 128, *restaurant* a value of 1076 and *school* a token value of 203. These values would suggest that a *coffee shop* is semantically more similar in feature space to *school* than to *restaurant*. Such a tokenization of nominal values is a standard procedure in natural language processing (NLP) [4] and can be used to train language related models by embedding words. However, geographic data, which is used here, does not exhibit a sequential character such as a sentence observed in NLP. Therefore methods from NLP are not suitable to model the geographic distribution of semantic features. The encodings used here overcome this issue and cherish the two dimensional (geographic) distribution of the semantics.

In this work, we used two encoding schemes which enable us to transform geospatial vector data into a feature space that tackles these two problems. As such, we take geospatial vector data from LinkedGeoData[1], which describes each geo-object semantically using an ontology, and aim at predicting LULC over Austria. Prior work suggests that geospatial semantics are highly associated with LULC classes [5]. The ground truth is obtained from CORINE (Level 2)[2].

[1] http://linkedgeodata.org/

[2] https://land.copernicus.eu/pan-european/corine-land-cover

3. Methodology

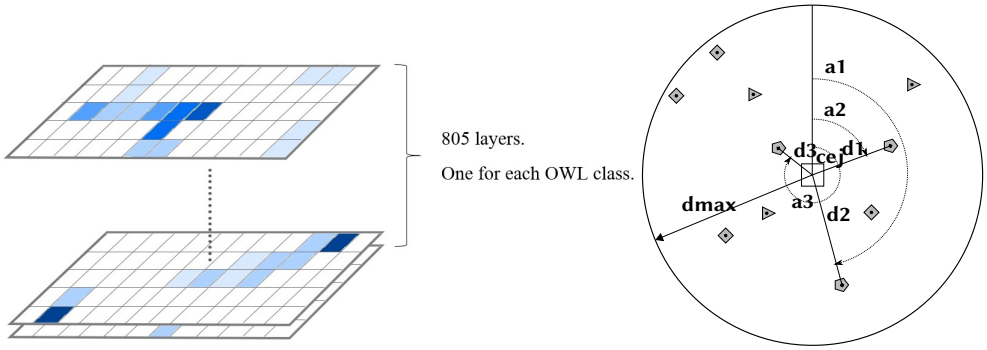
We used two input datasets for the classification experiments over Austria, which is the ROI: (1) Geospatial vector data from LinkedGeoData; (2) CORINE LULC raster data. There are 13 different LULC classes in Austria, defined by CORINE Level 2, described by grid-cells of size 100m x 100m. For each LULC class, 12,000 grid-cells were selected randomly, each of which serves as ground truth for the corresponding sample. Then, for each of the grid-cells, geospatial data from LinkedGeoData was obtained for the area 1km around the corresponding cell center. This data would then ultimately be used to predict the LULC class of the corresponding LULC grid-cell. However, in order two be able to do so, the vector data was transformed. Two encoding schemes were used for this purpose, the Geospatial Configuration Matrix [6], and an encoding that was developed within the realm of this work. We denote this encoding as the *Channel Encoding*.

3.1. The Geospatial Configuration Matrix

The Geospatial Configuration Matrix (GSCM) computes local statistics of every present class of geo-objects (see figure 1b for an illustration). It represents the mean, minimum, maximum of the distances and azimuths of all geo-objects of a particular type, relative to the LULC grid-cell center. For example, all geo-objects of type *restaurant* around a LULC grid-cell (distance is 1km) are obtained, then, all azimuths and distances between the restaurants and the LULC grid-cell are computed. Afterward, the mean, minimum and the maximum of these distances and azimuths were computed, which provides 6 features with regards to class *restaurant*. There are 805 geo-object classes within the ROI, which were not defined by us, but by the ontology of LinkedGeoData. As a result, the GSCM provides 4830 (6 features x 805 potential classes) features for each sample. LinkedGeoData describes the geo-objects with an Web Ontology Language (OWL) ontology. As such, each geo-object can be an instance of multiple classes. For example, a *pet-shop* is also a *shop* as well as an *amenity*.

3.2. The *Channel Encoding*

Another encoding scheme, denoted as the *Channel Encoding*, was developed within this work (see figure 1a). The focus of this encoding is to represent the geographic distribution of the geo-objects while avoiding the problem of a wrong semantic similarity. For this purpose, an image stack having a width and height of 1km, 805 channels and a pixel resolution of 10m x 10m was created for each sample. Each channel represents the geographic distribution of one OWL class of geo-objects, as each



(a) The *Channel Encoding* scheme. Each layer in the image stack represents the geographic distribution of geo-objects of a particular OWL class (e.g. *pet-shop*). Each pixel represents the number of geographic objects of this OWL class that appear within the corresponding pixel. Here, dark blue pixels denote higher numbers whereas light blue pixels denote a lower number of geo-objects.

(b) The GSCM computes six local statistics for each class of geo-objects around a grid-cell cej (which is obtained from CORINE level 2). The distance threshold around a grid-cell is denoted d_{max} and has to be defined. Here, we use a threshold of 1km only. In this figure, the different shapes denote different classes of geo-objects. For example, all pentagons could be of class Restaurant. Azimuths (a_1 to a_3) and distances (d_1 to d_3) are used to calculate six descriptive statistical values for this class of geo-objects. Source of image: [6]

Figure 1: The two different encodings used in this work.

pixel value is the number of geo-objects of this particular class which are present within the same pixel.

3.3. The artificial neural network architectures

Within this work we employ two different ANN architectures: The MLP-Mixer [7] and the Perceiver architecture [8]. The MLP Mixer architecture is based on fully connected layers only, however, they are applied layer-wise and pixel-wise. As such, it is a competitive alternative to convolutional neural networks and visual transformer networks, which are usually utilized for image vision tasks. The Perceiver architecture has two main characteristics: (1) It uses Transformer mechanisms which, in addition, use a reduced latent space as a bottleneck. This enables to speed up the computation. (2) It uses a parametrized Fourier feature encodings which enables to retrieve attentions, based on relative positions. As such, the Perceiver architecture is designed as a general-purpose ANN and was utilized for image data, sound data, and 3D point clouds [8].

3.4. The experiments

Once each encoding was performed for the input data, they were forwarded to the two ANNs used in this work, namely, the MLP-Mixer and Perceiver network. Finally, we calculated the overall accuracy as well as kappa coefficient for each of these four experiments (two encodings times two networks). Please note that the hyperparameters for each encoding were the same. In order to find a set of suitable hyper parameter, a random search was performed, as literature suggest its superiority over a systematic search, such as a grid search [9]. The chosen hyper parameters can be seen in table 1.

4. Results, Analysis, and Discussion

Table 2 shows the results of the accuracy assessment of the four experiments. Using the *Channel Encoding* and the Perceiver yielded the highest accuracy of 72.0% with a kappa coefficient of 0.69. However, using the *Channel Encoding* yielded the lowest accuracy once it was used with the MLP-mixer network. In contrast, using the GSCM yielded the highest accuracies once used with the MLP-Mixer (overall accuracy of 70.2% and kappa coefficient of 0.68). The Perceiver architecture and *Channel Encoding* may have yielded the best results as the Perceiver is a general-purpose ANN which can handle a high number of channels. Additionally, the Perceiver uses a Fourier feature encoding, which is parametrized. As such, it can compare values based on relative positions. This capability enables the network to learn meaningful spatial relationships which are present within the entire extent

Table 1
The hyperparameters of the MLP-Mixer (left) and Perceiver (right) architecture used in this work. The train and test split of the data was 80% and 20%, respectively. The hyperparameters were searched for in a random search manner.

MLP-Mixer	
Latent dimensions	128
Number of MLP-Mixer modules	6
dropout	0.60

Perceiver	
latent dimensions	512
cross heads	1
attention dropout	0.4
fully connected dropout	0.4
latent heads	4
number of iterations	10

Table 2
The results of the accuracy assessment of the experiments. The geographic extend used for each sample was the same (1 km).

Network	Task	OA (dmax = 1km)	Kappa (dmax = 1km)
MLP-Mixer[7]	GSCM	70.2 %	0.68
	Channel Encoding	42.2%	0.37
Perceiver[8]	GSCM	65.1 %	0.62
	Channel Encoding	**72.0%**	**0.69**

of the *Channel Encoding*. In contrast, the MLP-Mixer does not apply such strategies which can potentially explain the poor performance when the *Channel Encoding* is used with it and might suggest that attention mechanisms combined with learnable position encoding can significantly help to detect meaningful spatial relationships. Another characteristic of the Perceiver might have improved the accuracies, once the *Channel Encoding* was used: it is permutation invariant to its input. As such, certain spatial constellations may have appeared in a transformed manner (e.g. rotated or shifted) but still be captured as related and meaningful. A potential explanation for the MLP-Mixer to have a lower classification performance when using the *Channel Encoding* than when using the GSCM, could be the following: The MLP-Mixer model transforms every patch of the input into a latent space, using a multilayer perceptron (MLP). This patch size is set by the number of channels (805 in the case of the *Channel Encoding*) and a defined width and height. Here, we used the entire extent of 2km (two times 1km) as width and height, covering the entire extend of a sample. While it might be more difficult for the MLP to compute a meaningful latent representation for a bigger patch than for a smaller one, a smaller patch size might hamper to find meaningful spatial relationships. Thus, the MLP-Mixer model might not have been able to compute meaningful latent representations as the input patches were too big. In this work we explored only one dmax threshold (1km). However, [6] found that this value can have a significant impact on the performance of the spatial prediction model. As such, we would expect that the LULC classification performance would change here too, once dmax is changed. However, we leave the exploration of this hyper parameter to future research.

5. Conclusions and Future Research

These preliminary results suggest that geospatial vector data can be used for a spatial prediction task like LULC classification. They also indicate that the way how geospatial vector data is encoded plays a crucial role. Also, the type of network plays an important role: The Perceiver network was able to learn meaningful features from the *Channel Encoding*, which was not the case with the MLP-Mixer. In this work, we described a first step towards how geospatial vector data can be encoded for using it with deep neural networks. As such, future research can explore novel ways how vector data can be efficiently encoded. Furthermore, ANN architectures that do not require a prior encoding could be explored. In this work we explored using vector data alone, however, future research could focus on how vector data and imagery, such as hyperspectral [10] or multispectral optical imagery [11], could be fused into one enhanced feature space.

References

[1] A. J. Comber, R. A. Wadsworth, P. F. Fisher, Using semantics to clarify the conceptual confusion between land cover and land use: the example of 'forest', Journal of Land Use Science 3 (2008) 185–198. doi:10.1080/17474230802434187.

[2] P. Fisher, The pixel: A snare and a delusion, International Journal of Remote Sensing 18 (1997) 679–685. doi:10.1080/014311697219015.

[3] S. Winter, Unified Behavior of Vector and Raster Representation, University of Technology Vienna, 2000. URL: https://books.google.at/books?id=QfGwcQAACAAJ.

[4] J. J. Webster, C. Kit, Tokenization as the initial phase in nlp, in: COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics, 1992.

[5] M. M. Cutchan, I. Giannopoulos, Geospatial Semantics for Spatial Prediction, in: S. Winter, A. Griffin, M. Sester (Eds.), 10th International Conference on Geographic Information Science (GIScience 2018), volume 114 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018, pp. 45:1–45:6. doi:10.4230/LIPIcs.GISCIENCE.2018.45.

[6] M. Mc Cutchan, S. Özdal Oktay, I. Giannopoulos, Semantic-based urban growth prediction, Transactions in GIS 24 (2020) 1482–1503. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12655. doi:https://doi.org/10.1111/tgis.12655.

[7] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, A. Dosovitskiy, Mlp-mixer: An all-mlp architecture for vision (2021). URL: https://arxiv.org/abs/2105.01601. arXiv:2105.01601.

[8] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, J. Carreira, Perceiver: General perception with iterative attention (2021). URL: https://arxiv.org/abs/2103.03206. arXiv:2103.03206.

[9] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[10] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE Transactions on Geoscience and Remote Sensing 54 (2016) 6232–6251. doi:10.1109/TGRS.2016.2584107.

[11] Y. Chen, C. Li, P. Ghamisi, X. Jia, Y. Gu, Deep fusion of remote sensing data for accurate classification, IEEE Geoscience and Remote Sensing Letters 14 (2017) 1253–1257. doi:10.1109/LGRS.2017.2704625.