SOLVING DEEP NEURAL NETWORKS WITH ORDINARY DIFFERENTIAL EQUATIONS FOR REMOTELY SENSED HYPERSPECTRAL IMAGE CLASSIFICATION

[1]*M. E. Paoletti, Student Member, IEEE*, [1]*J. M. Haut, Student Member, IEEE*,
[1]*J. Plaza, Senior Member, IEEE*, [1]*A. Plaza, Fellow, IEEE*
[1]Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, E-10003 Cáceres, Spain. (e-mail: mpaoletti@unex.es)

ABSTRACT

Deep neural networks (DNNs) have revolutionized the way remotely sensed hyperspectral image (HSI) data are managed and processed. For instance, residual networks (ResNets) have achieved high classification accuracy by applying sequential transformations (layer by layer) on the input HSI data, obtaining highly discriminative data representations. However, these models are quite complex, with significant requirements in terms of memory resulting from the large number of parameters that they need to learn, which also leads to potential overfitting issues. In this work, we specifically address the aforementioned problem by re-interpreting a DNN (the ResNet) as a continuous transformation, instead of the traditional (discrete) step-by-step approach. To achieve this, we combine ordinary differential equations (ODEs) with DNN architectures for the first time in the HSI data classification literature. This allows us to perform remotely sensed HSI data classification in an efficient way in terms of number of parameters. Our experimental results, conducted using two well-known HSI data sets, indicate that the inclusion of ODEs in the architecture of DNNs offers significant advantages when processing and classifying this kind of high-dimensional data, achieving better performance even with less training data.

Index Terms— Deep neural networks (DDNs), hyperspectral images (HSIs), residual networks (ResNets), ordinary differential equations (ODEs).

This paper was supported by Ministerio de Educación (Resolución de 26 de diciembre de 2014 y de 19 de noviembre de 2015, de la Secretaría de Estado de Educación, Formación Profesional y Universidades, por la que se convocan ayudas para la formación de profesorado universitario, de los subprogramas de Formación y de Movilidad incluidos en el Programa Estatal de Promoción del Talento y su Empleabilidad, en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016. This work has also been supported by Junta de Extremadura (decreto 14/2018, ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR18060) and by MINECO project TIN2015-63646-C5-5-R.

1. INTRODUCTION

The application of imaging spectroscopy to Earth Observation and remote sensing problems allows for the acquisition of high-dimensional HSI data cubes composed by hundreds of observations at narrow spectral wavelengths. As a result, each pixel in the HSI contains a detailed spectral signature that represents the observed image object(s). This information uniquely characterizes each element of the HSI, which is very useful for classification purposes. Specifically, classification consists of assigning to each pixel $\mathbf{x}_i \in \mathbb{R}^{n_{bands}}$ of the HSI dataset $\mathbf{X} \equiv \{\mathbf{x}_1, \cdots, \mathbf{x}_{n_{samples}}\} \in \mathbb{R}^{n_{samples} \times n_{bands}}$ a unique label $y_i = \{1, \cdots, K\}$, extracted from a set of K possible categories, creating pairs of $\{\mathbf{x}_i, y_i\}_{i=1}^{n_{samples}}$. In this sense, the classification process aims to approximate the function $f(\cdot, \theta)$ which, depending on parameters θ, maps the data of $\mathcal{X} \subset \mathbb{R}^{n_{samples}}$ (for instance, a HSI data set $\mathbf{X}$) to those categories/labels contained in $\mathcal{Y}$, i.e., $f : \mathcal{X} \rightarrow \mathcal{Y}$.

Among several techniques developed for the efficient classification of HSI data, artificial neural networks (ANNs) have been a very useful tool for the analysis of these high-dimensional images because of their great flexibility in terms of available architectures and learning modes, in addition to their capacity to extract representative features and their ability to discover non-linear relationships in the data [1].

Advances in deep learning have allowed the implementation of deeper and complex ANNs, called deep neural networks (DNNs), which are composed by a hierarchy of multiple layers in which the l-th layer applies a transformation to the input data $\mathbf{x}^{(l)}$, its weights $\mathbf{W}(l)$ and biases $b^{(l)}$, to finally pass the result through a non-linear activation function $\mathcal{H}(\cdot)$:

$$\begin{aligned} \mathbf{a}^{(l+1)} &= \mathbf{W}^{(l)} \cdot \mathbf{x}^{(l)} + b^{(l)} \\ \mathbf{x}^{(l+1)} &= \mathcal{H}(\mathbf{a}^{(l+1)}) \end{aligned} \tag{1}$$

In a classification context, function $f(\cdot, \theta)$ can be replaced by the concatenation of those affine linear transformations and point-wise nonlinearities defined by Eq. (1) at each hidden layer, which can be considered as nonlinear functions, while θ comprises all network's parameters.

Eq. (1) can be applied *directly* (as in traditional fully-connected architectures such as multilayer perceptrons [2]),

in the form of a *time series* (as in recurrent neural networks [3]), or included in a *kernel operation* of a convolutional neural network (CNN) [4]. In addition, the introduction of skip- and residual-connections allows for the development of more complex architectures, in which grouped layers conform entire blocks of mapping data [5]. In particular, residual neural networks (ResNets) [5] group several operation layers and non-linear activation functions into blocks, called residual units, whose inputs and outputs are connected through a residual connection that helps to propagate the information from previous layers to the rest of the network. In this sense, for the l-th residual unit, Eq. (1) can be reformulated as follows:

$$\begin{aligned} \mathbf{a}^{(l+1)} &= \mathbf{x}^{(l)} + \mathcal{F}\left(\mathcal{W}^{(l)}, \mathbf{x}^{(l)}, \mathcal{B}^{(l)}\right) \\ \mathbf{x}^{(l+1)} &= \mathcal{H}(\mathbf{a}^{(l+1)}) \end{aligned} \tag{2}$$

where $\mathcal{F}(\cdot)$ represents all the operations applied over the input data $\mathbf{x}^{(l)}$, i.e., the additive residual mapping function, and $\mathcal{W}^{(l)}$ and $\mathcal{B}^{(l)}$ are the weights and biases, respectively, of the layers involved in each residual block. The ResNet can be interpreted as a discrete sequence of L hidden blocks, where the data flow propagates through the residual units until an abstract representation of the original input data is obtained. In this sense, the number of trainable parameters depends directly on L. This fact leads memory consumption to grow linearly in $\mathcal{O}(L)$ order, which also implies the need to use more data to properly train the model (overfitting problem).

With the aim of developing a DNN with constant and lower memory cost, and a significantly reduced number of trainable parameters (thus effectively dealing with overfitting issues), this work re-interprets the traditional ResNet as a continuous transformation [6], considering an architecture with $L \to \infty$ and very small step-size. In this context, Eq. (2) is interpreted as the Euler discretization of a continuous transformation in any time-step, from t to $t+1$ [7], where the first value $\mathbf{x}^{(0)} = \mathbf{x}$ is the input to the original network. Following the aforementioned principles, ResNet can be considered as an ordinary differential equation (ODE), whose successive evaluations obtain hidden states from the input data, until a desired level of precision is reached by the classifier.

In summary, this work proposes (for the first time in the literature) the implementation of a continuous-depth neural network with a parameterized ODE [6] which is specifically designed for the classification of HSI data. Section 2 introduces our newly developed methodology, while section 3 provides a detailed discussion of the results obtained using two widely-used HSI data sets. Section 4 concludes the paper with some remarks and hints at plausible future research lines.

2. METHODOLOGY

A first-order ODE can be expressed as the initial value problem (IVP) of the form $\frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = f(t, \mathbf{h}(t))$ with initial condition $\mathbf{h}(t_0) = \mathbf{h}_0$, being $t \in \{0, \cdots, T\}$ an observation interval and $f(t, \mathbf{h}(t))$ a known function. The solution of this ODE is the value of the unknown function $\mathbf{h}(t) = \mathbf{h}_t$ at each point t. Geometrically, $\mathbf{h}_{i+1}$ at point t_{i+1} can be approximated through the tangent line based on the previous point as $\mathbf{h}_{i+1} = \mathbf{h}_i + f(t_i, \mathbf{h}_i) \cdot (t_{i+1} - t_i)$. In this context, the Euler method gives a solution for $\mathbf{h}(t)$, assuming that the i-th observation point is related with the first one as $t_i = t_0 + \alpha \cdot i$, being α a step-size. This assumption leads to the fact that each point is related to the immediately preceding one through the step-size: $t_{i+1} = t_i + \alpha$, which can be included in the previous equation to obtain a final expression $\mathbf{h}_{i+1} = \mathbf{h}_i + \alpha \cdot f(t_i, \mathbf{h}_i)$. Comparing this expression with Eq. (2), it can be observed that the residual mapping unit is a special case of the Euler discretization method, where the step-size is set to $\alpha = 1$ and the known function $f(\cdot)$ is parameterized by the weights and biases of the l-th block, with $l \in \{0, \cdots, L\}$, being l_0 the input layer. The aforementioned observations can be extrapolated in order to re-define the ResNet under a continuous interpretation as follows:

$$\begin{aligned} \mathbf{h}_{i+1} &= \mathbf{h}_i + f(t_i, \mathbf{h}_i, \theta_i) \\ \text{where } \frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} &= f(t, \mathbf{h}(t), \theta) \text{ with } \mathbf{h}(t_0) = \mathbf{x} \end{aligned} \tag{3}$$

where the traditional residual units have been replaced by a parameterized ODE [6], being the weights and biases defined by θ and the neural topology by $f(\cdot)$, while the output is the hidden state $\mathbf{h}(t_i)$ at time t_i. Under these assumptions, the traditional block-by-block performance (which depends on L) is eventually replaced by $\tilde{L}$ evaluations of Eq. (3), which can be easily carried out by any off-the-shelf ODE solver:

$$\mathbf{h}_{i+1} = \mathit{ODEsolver}(\mathbf{h}(t_i), f, t_i, t_{i+1}, \theta) \tag{4}$$

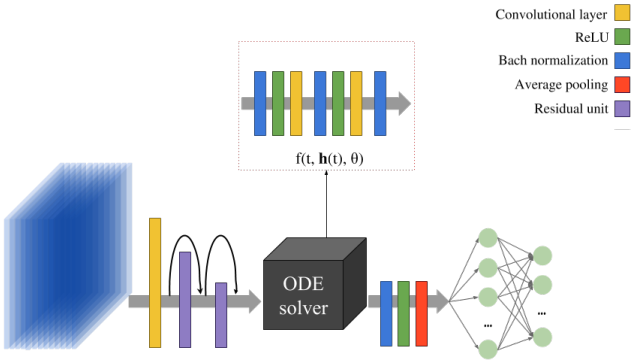
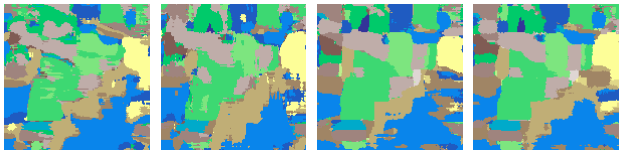


Fig. 1. Architecture of the proposed ODE-net for HSI data classification. It is composed by three well-differentiated parts: i) a pre-processing step that extracts low-level feature representations from the original HSI patches, which feed the ii) ODE solver, and whose output state is employed to perform iii) the final classification, implemented by fully-connected layers.



a) ResNet (75.18%) *b*) **ODE-net (79.98%)** *c*) ResNet (98.29%) *d*) **ODE-net (98.45%)**

Fig. 2. Classification maps for Indian Pines (IP) dataset, with 3% [a) and b)] and 15% [c) and d)] of training data. Note that the overall classification accuracies are shown in brackets, and the best result is highlighted in bold typeface.

During the forward-pass, a state is obtained by Eq. (4), which is employed to calculate the loss function of the network $loss(ODEsolver(\mathbf{h}(t_i), f, t_i, t_{i+1}, \theta))$. This $loss$, implemented as the cross-entropy function, is back-propagated through the entire network, being the network parameters updated by the stochastic gradient descent.

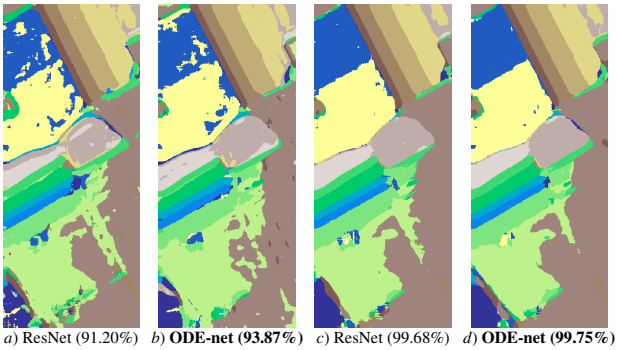
An important detail to keep in mind is the memory cost of the proposed network, which depends exclusively on the parameters managed by $f(\cdot)$ and is kept constant in each evaluation, leading to $\mathcal{O}(1)$ cost. In this sense, our newly proposed network for HSI data classification controls more efficiently the number of trainable parameters, which has a remarkable impact on the memory usage and on the overfitting of the model. As opposed to the classic ResNet, whose performance highly depends on the number of parameters to train and also on the amount of available training data to properly fine-tune these parameters, our proposal is able to obtain more robust results with any percentage of training data (even with a very low number of training samples), while the original model is hampered by its very high number of parameters. For illustrative purposes, Fig. 1 provides a graphical representation of the proposed architecture (called hereinafter ODE-net) for HSI data classification purposes.

3. EXPERIMENTAL RESULTS

3.1. Environment and Data Sets

All our experiments have been conducted on a 6th Generation Intel® Core™i7-6700K processor with 8 MB of cache and up to 4.20 GHz frequency (4 cores/8 way multitask processing), 40 GB of DDR4 RAM memory with serial speed of 2400MHz, a GPU NVIDIA GeForce GTX 1080 with 8 GB GDDR5X of video memory and 10 Gbps of memory frequency, a Toshiba DT01ACA HDD with 7200RPM and 2 TB of disk capacity, and an ASUS Z170 pro-gaming motherboard. Ubuntu 18.04 x64 and python have also been employed in our implementation.

Two widely used HSI data sets have been considered to conduct the experimental assessments: the Indian Pines (IP) and the Salinas Valley (SV) scenes, collected by the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) over Northwestern Indiana and the Salinas Valley in California, respectively. The first scene contains 145×145 samples



a) ResNet (91.20%) *b*) **ODE-net (93.87%)** *c*) ResNet (99.68%) *d*) **ODE-net (99.75%)**

Fig. 3. Classification maps for the Salinas Valley (SV) dataset with 1% [a) and b)] and 10% [c) and d)] of training data. Note that the overall classification accuracies are shown in brackets, and the best result is highlighted in bold typeface.

with 200 spectral bands (after the removal of noisy bands), spatial resolution of 20 meters per pixel, and spectral range from 0.2 to 2.4 microns. The second scene is composed by 512×217 pixels, with 204 spectral bands (after the removal of noisy bands) and spatial resolution of 3.7 meters per pixel. Both images comprise 16 land-cover classes. To assess the classification accuracies obtained over these two scenes, we use the overall (OA) and average (AA) accuracy, and the kappa coefficient. We also compute the number of required model parameters.

3.2. Discussion of Results

To test the performance of our newly proposed ODE-net for HSI data, a comparison has been conducted with the popular ResNet using the two aforementioned scenes. It should be noted that the ResNet has been implemented by replacing the ODEsolver in Fig. 1 by six residual units. Two different training percentages have been employed for each scene: 3% and 15% (for IP) and 1% and 10% for SV. These percentages define the number of labeled samples that are randomly selected to create the training set, while the test set comprises all the remaining labeled samples.

Table 1 reports the classification results achieved by ResNet and our newly proposed ODE-net, obtained as the average of 5 experiments. It can be observed that, with enough training, our newly proposed ODE-net is able to reach very similar (and even slightly better) results in terms of OA than the traditional ResNet. However and most importantly, in the case that very limited training samples are available, our proposal is able to reach the best results employing only one third of the trainable parameters required by the ResNet. This demonstrates that, in addition to providing more robust results with fewer samples, our model also achieves a better use of memory resources, with an impressive reduction in the number of required trainable parameters.

For illustrative purposes, Figs. 2 and 3 show some of the

Table 1. Classification results obtained for the IP and SV data sets using different percentages of training samples.

	Indian Pines				Salinas Valley			
	3%		15%		1%		10%	
Class	ResNet	ODE-net	ResNet	ODE-net	ResNet	ODE-net	ResNet	ODE-net
1	8.41 ±8.9794	**21.14** ±16.4219	**95.13** ±5.8974	94.62 ±4.7901	90.07 ±4.5933	**93.59** ±4.9556	99.85 ±0.1732	**99.98** ±0.0253
2	73.55 ±5.4803	**77.36** ±4.8655	97.93 ±0.5693	**98.04** ±1.3882	88.68 ±5.4176	**92.68** ±3.8408	99.96 ±0.0600	**99.92** ±0.1118
3	61.49 ±14.2059	**68.34** ±8.1622	98.23 ±0.5575	**97.21** ±2.1993	78.80 ±14.1556	**91.29** ±8.1720	99.81 ±0.2193	**99.93** ±0.1665
4	63.36 ±21.8903	**69.08** ±9.5369	97.96 ±1.6717	**98.11** ±1.3859	98.27 ±2.1726	**99.17** ±0.7469	99.87 ±0.0813	**99.86** ±0.1429
5	67.82 ±11.6907	**72.95** ±5.2044	95.71 ±2.6678	**97.22** ±2.1684	98.16 ±1.3211	**98.79** ±0.9839	**99.80** ±0.1443	99.73 ±0.1782
6	76.00 ±9.7225	**80.48** ±7.2722	98.87 ±0.7967	**99.39** ±0.4774	98.88 ±0.7218	**99.40** ±0.6706	**99.97** ±0.0365	99.95 ±0.0595
7	0.74 ±2.2222	**11.48** ±18.2537	89.13 ±17.0898	**94.35** ±8.0288	94.87 ±3.6414	**97.48** ±1.4419	99.82 ±0.1576	**99.89** ±0.0847
8	94.82 ±7.9803	**93.84** ±7.5768	**99.93** ±0.1129	99.90 ±0.1970	89.98 ±2.6326	**90.62** ±1.8414	99.38 ±0.3279	**99.46** ±0.2733
9	0.53 ±1.5789	**5.79** ±12.1053	69.41 ±19.2956	**89.41** ±8.6453	96.19 ±2.3344	**98.46** ±0.7968	**99.99** ±0.0268	99.96 ±0.1129
10	58.70 ±18.5370	**65.25** ±4.7458	97.13 ±1.6627	**96.94** ±1.2801	91.87 ±4.7726	**95.55** ±2.3230	99.95 ±0.0551	**99.97** ±0.0320
11	85.49 ±4.6536	**88.61** ±2.3170	99.36 ±0.6392	**99.54** ±0.3739	93.53 ±4.6894	**94.36** ±1.9628	98.52 ±0.9781	**99.29** ±0.3507
12	71.08 ±9.3017	**72.14** ±6.4694	**97.70** ±1.2010	97.68 ±1.5948	95.32 ±2.1510	**96.69** ±1.6449	99.60 ±0.3991	**99.84** ±0.1155
13	79.14 ±17.7546	**90.51** ±6.7375	99.20 ±1.2902	**99.83** ±0.3680	96.92 ±1.9443	**97.85** ±1.2978	99.42 ±0.4998	**99.34** ±0.6287
14	90.91 ±3.4966	**95.60** ±2.6161	99.41 ±0.5582	**99.67** ±0.2920	90.56 ±5.6880	**96.52** ±2.5817	99.65 ±0.5155	**99.84** ±0.2379
15	64.20 ±11.0867	**76.18** ±12.1276	96.40 ±2.6988	**97.56** ±1.4998	82.81 ±3.1921	**86.46** ±3.5356	99.43 ±0.2990	**99.61** ±0.2081
16	2.56 ±6.6305	**34.67** ±31.0308	**94.18** ±2.6065	89.11 ±5.9156	86.07 ±5.2526	**93.28** ±1.8057	**99.68** ±0.2761	99.63 ±0.4261
OA	75.18 ±6.2376	**79.98** ±1.7644	98.29 ±0.4306	98.45 ±0.4583	91.20 ±1.2009	**93.87** ±0.5633	99.68 ±0.1240	**99.75** ±0.0723
AA	56.17 ±7.3123	**63.96** ±4.2821	95.35 ±1.7001	**96.79** ±1.0424	91.94 ±1.5623	**95.14** ±0.4933	99.67 ±0.1333	**99.76** ±0.0695
K	71.50 ±7.3363	**77.05** ±2.0522	98.05 ±0.4909	98.23 ±0.5232	90.19 ±1.3403	**93.18** ±0.6271	99.64 ±0.1381	**99.72** ±0.0805
Time(s)	**48.98** ±0.1925	106.25 ±0.1801	**78.07** ±0.2078	205.31 ±9.7192	**113.22** ±0.1221	347.84 ±0.4285	**223.65** ±0.3511	713.40 ±4.2058
Parameters	601872	**232080**	601872	**232080**	601872	**232080**	601872	**232080**

best classification maps obtained by the considered methods in our experiments. As we can observe, the classification maps provided by ResNet and our newly proposed ODE-net are quite similar, as already indicated by Table 1, when enough training is employed. However, when very few training samples are available, the ResNet is hampered by distortion noise in the obtained classification results, while our newly developed ODE-net consistently provides very good classification results regardless of the number of available training samples.

4. CONCLUSIONS AND FUTURE LINES

This work presents, for the first time in the literature, a continuous DNN for remotely sensed HSI data classification based on ODE solvers. The proposal demonstrates significant improvements in terms of memory consumption and parameter generation when compared with classical, discrete ResNet implementations, allowing an efficient reduction of the model's overfitting. Most importantly, ODE-net requires significantly less training samples to provide consistently good classification results. As future work, a deeper study of parameter $\tilde{L}$ on the accuracy of the network should be conducted. Also, a reduction of computation times will be pursued through the implementation of more computationally efficient solvers.

5. REFERENCES

[1] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 1, pp. 8–32, March 2017.

[2] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "An investigation on self-normalized deep neural networks for hyperspectral image classification," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 3607–3610.

[3] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, July 2017.

[4] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6440–6461, Nov 2018.

[5] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral-spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2018.

[6] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems 31*, pp. 6572–6583. Curran Associates, Inc., 2018.

[7] Lars Ruthotto and Eldad Haber, "Deep neural networks motivated by partial differential equations," *arXiv preprint arXiv:1804.04272*, 2018.