Chapter 10
Artificial Intelligence

Eric Guérin, Orhun Aydin and Ali Mahdavi-Amiri

Abstract In this chapter, we provide an overview of different artificial intelligence (AI) and machine learning (ML) techniques and discuss how these techniques have been employed in managing geospatial data sets as they pertain to Digital Earth. We introduce statistical ML methods that are frequently used in spatial problems and their applications. We discuss generative models, one of the hottest topics in ML, to illustrate the possibility of generating new data sets that can be used to train data analysis methods or to create new possibilities for Digital Earth such as virtual reality or augmented reality. We finish the chapter with a discussion of deep learning methods that have high predictive power and have shown great promise in data analysis of geospatial data sets provided by Digital Earth.

Keywords Artificial intelligence · Machine learning · Generative models · Statistical data analysis

10.1 Introduction

Earth and its associated data sets are massive. Various forms of geospatial data sets are constantly accumulated and captured by different forms of sensors and devices (Mahdavi-Amiri et al. 2015). Managing such an immense data set is a challenge. As a result, many automated techniques have been designed to process geospatial data sets with minimal human interference. Since manual involvement should be minimal, the machines should be capable of processing data and delivering meaningful information to the users. With advancements in machine learning, processing

E. Guérin
LIRIS, Université de Lyon, Lyon, France

O. Aydin
ESRI, Los Angeles, Los Angeles, USA

University of Southern California, Los Angeles, USA

A. Mahdavi-Amiri (✉)
Department of Computing Science, Simon Fraser University, Burnaby, Canada
e-mail: amahdavi@sfu.ca

geospatial data sets has significantly improved. In this chapter, we discuss artificial intelligence and machine learning techniques that have been useful to manage and process geospatial data sets. Because the processing of geospatial data can also be a source of knowledge, some methods use existing data to generate and synthesize new data.

We start by discussing some traditional and statistical approaches in machine learning and then present more recent learning techniques employed for geospatial data sets. Traditional methods include predefined models such as linear regression, PCA, SVD, active contour, and SVM, in which the model is fixed and the learning is based on an optimization. We also briefly discuss evolutionary and agent-based methods and autoencoders as traditional methods that can be deep or shallow. We then discuss more recent deep learning techniques, including reinforcement learning, deep convolutional networks and generative models such as variational autoencoders and generative adversarial networks. In this chapter, we describe some applications of these machine learning techniques to handle geospatial data sets that are the main content of Digital Earth. In the future, a dynamic Digital Earth that can use such techniques to work with geospatial datasets is extremely practical. Currently, such methods are sparsely used on very specific Digital Earth data sets. We imagine that a more advanced Digital Earth will use state-of-the-art machine learning techniques much more than they are currently used.

10.2   Traditional and Statistical Machine Learning

Inferring patterns and forming relationships using artificial intelligence require knowledge of some characteristics of the phenomena/system of interest. One of the early approaches to enabling artificial intelligence for complex problems was to create knowledge bases that contain explicit sets of rules and associations, also known as ontology (Gruber 1993). For data pertaining to Earth system modeling, different niche knowledge bases were designed by various authors (McCarthy 1988; Rizzoli and Young 1997). The knowledge base approach to artificial intelligence required expert input to define the rules and associations. In addition, the expert knowledge had to be represented in a "computable form" (Sowa 2000), posing a bottleneck for these approaches. For spatially varying, complex phenomena, ontology representations were defined for Earth's subsystems such as in environmental modeling and planning (Cortés et al. 2001), and ecological reasoning (Rykiel 1989). General spatial and GIS knowledge bases were proposed by various authors (Kuipers 1996; Egenhofer and Mark 1995; Fonseca et al. 2002).

Despite the plethora of niche knowledge bases, knowledge base artificial intelligence requires assertions and ground truths (Lenat 1995), which can conflict with observations (Goodfellow et al. 2016). Numerous attempts to address this limitation have been presented by various authors, such as defining hierarchical (Kuipers 1996), or location/problem-tailored knowledge bases (Rizzoli and Young 1997).

Statistical machine learning alleviates the limitations of the knowledge-based approach to artificial intelligence and discovers rules and patterns from the data directly without explicit supervision (Goodfellow et al. 2016). In the case of statistical learning, patterns and rules from an unknown underlying process are defined for descriptive, predictive and prescriptive analytics.

Applications of statistical learning to understand and forecast natural and human phenomena are evaluated with respect to the components of the general definition of machine learning (Mitchell 1997). Mitchell's (1997) definition is as follows:

> A computer program is said to learn from experience [D] with respect to some class of tasks T and performance measure [Q], if its performance at tasks in T, as measured by [Q], improves with experience [D].

Machine learning methods are broadly grouped into supervised and unsupervised methods. Supervised machine learning methods experience modeled phenomena through so-called labeled training data. Labels in the training data correspond to the target variable to be predicted, either quantitative (regression) or qualitative (classification). Training data consists of predictors and their corresponding predictand. Thus, supervised machine learning methods learn relationships in the data through experiencing input/output pairs.

Unsupervised machine learning methods discover patterns in the data without supervision or explicit rules. Clustering is one of the most common unsupervised machine learning methods for geospatial datasets.

10.2.1 Supervised Learning

Supervised learning aims to define a relationship between r predictor variables, denoted by $X = (X_1, X_2, \ldots, X_r)$, and e predictands, $Y = (Y_1, Y_2, \ldots, Y_e)$. Supervised learning can be posed as a density estimation problem (Hastie et al. 2001):

$$P(Y|X) = P(Y, X)/P(X) \qquad (10.1)$$

where $P(Y|X)$ is the conditional probability density of observing the predictand given the predictors, $P(Y, X)$ is the joint probability distribution of the predictand and predictors, and $P(X)$ is the marginal probability distribution of the predictors. Using Mitchell's (1997) description, the performance Q can be quantified using a loss function $\mathcal{L}$ where, for a given method and set of parameters Θ, a location function, $\mu(x)$, is minimized (Hastie et al. 2001) in Eq. 10.2.

$$\mu(x) = argmin_{\Theta} E_{Y|X} \mathcal{L}(Y, \Theta) \qquad (10.2)$$

For a given Θ, a supervised machine learning method predicts the values at X as $\hat{y}$. The loss function, $\mathcal{L}$, quantifies the error between $\hat{y}$ and the training data y.

Some examples of supervised machine learning methods as they pertain to geospatial analysis are given in the following subsection.

10.2.1.1 Random Forest

Random forest is a framework for nonparametric estimation in which both classification and regression can be performed (Breiman 2001). It has gained popularity in numerous geospatial applications due to its flexibility in accommodating different types of inputs (categorical or continuous) and its ability to model complex relationships in the data.

Random forest addresses the overfitting limitation of **c**lassification **a**nd **r**egression **t**rees (CART). Random forest uses bootstrap aggregating, also known as bagging, to create subsets of the training data by sampling with replacement to build different CARTs (Breiman 1996). Each of the CARTs that make up the forest predict, or vote, for a given data point of x and the forest returns the majority vote in a classification or the average forest prediction for a regression. The voting scheme of random forest allows for complex relationships to be captured in the data that might not be possible otherwise. A pictorial summary of a random forest classifier for classifying a successful retail store (one) or an unsuccessful one (zero) with respect to its distance to the nearest highway exit and the number of brands it carries is given in Fig. 10.1.
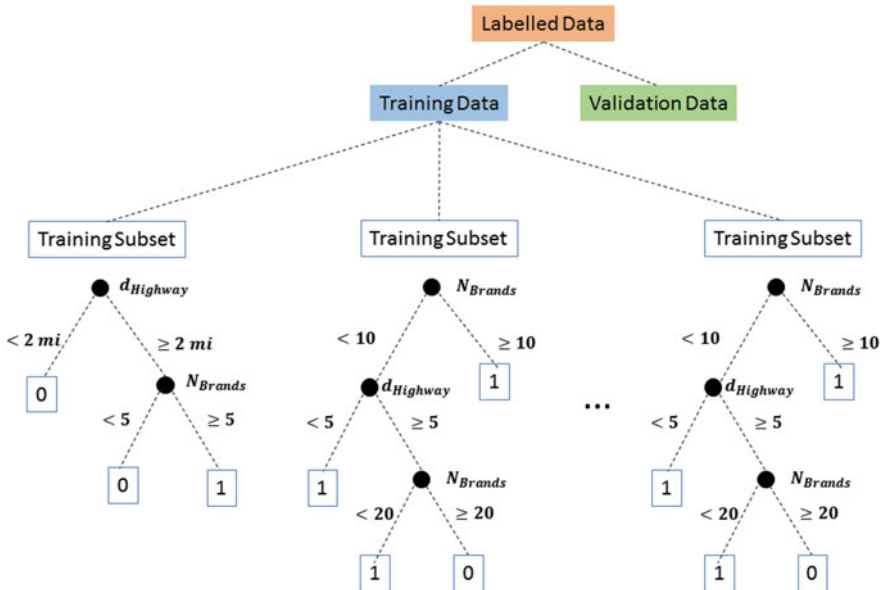

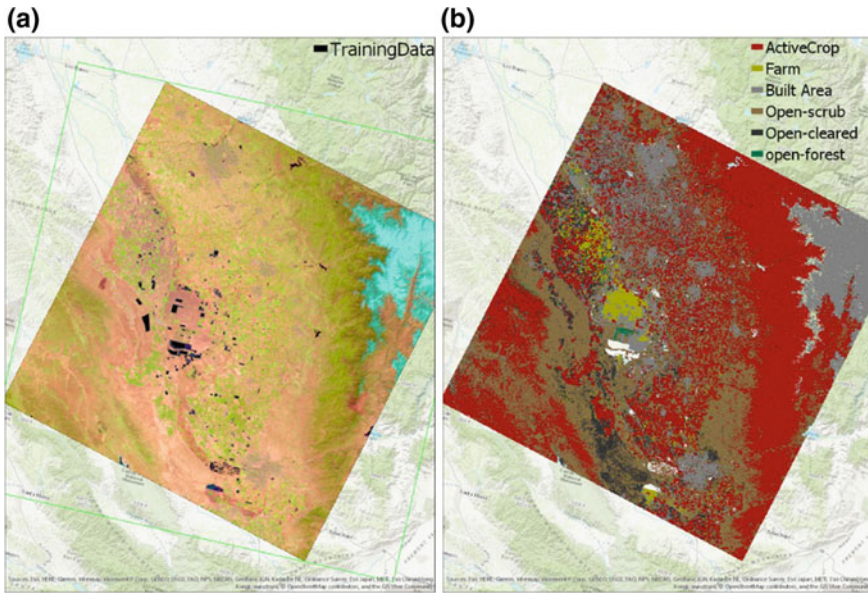**Fig. 10.1** Cartoon representation of a random forest classifier

Fig. 10.2 a Satellite image over southern California, with training data marked with black polygons **b** classified land coverage map using random forest

Note that every tree experiences different subsets of training data and their structures are different from one another. The voting scheme allows for capturing underlying patterns in the data by defining complex relationships captured in a large ensemble of trees rather than a single tree.

In geospatial problems, various random forest classifiers are used in a wide range of problems, including land cover classification (Gislason et al. 2006) and ecological modeling (Cutler et al. 2007). In land cover classification, random forest speeds up classification of land use by forming a relationship between the satellite image RGB value and the type of land it corresponds to. In this case, the training data consists of tagged locations at which the land cover and RGB values are known. An example of the random forest classifier output for land use classification is given in Fig. 10.2.

In Fig. 10.2, a small number of farms and areas around them were used as training data (marked with black polygons). The training set that consists of 300 farms was used within the random forest classifier to define land use in southern California.

10.2.1.2   Geographically Weighted Regression

Geographically **w**eighted **r**egression (GWR) provides a statistical framework for incorporating spatial dependency within a linear regression system (Fotheringham

Fig. 10.3 Conceptual depiction of GWR. Regression is performed for the orange point with a red circle defining the neighborhood
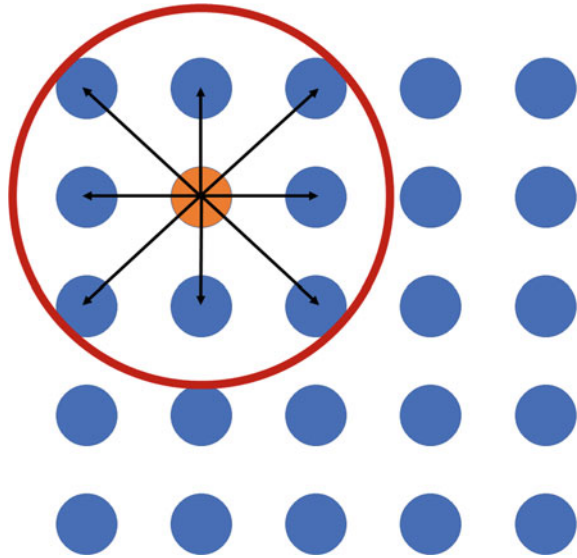
et al. 2003). GWR provides spatial extensions to ordinary least squares and general-ized linear models (Nelder and Wedderburn 1972) such as geographically weighted logistic regression. GWR is depicted conceptually in Fig. 10.3.

Figure 10.3 illustrates a regression system solved within the neighborhood (red circle) for the location indicated in orange. First, GWR defines a weighting scheme to determine spatial weights for the neighbors, and the predictors X at every location (blue) are weighted with respect to their distance to the location for which the regres-sion is performed (orange). The geographically weighted linear system of equations solved at a point i can be expressed as follows:

$$\hat{\beta}(u_i, v_i) = \left(\mathbf{X}^T \mathbf{W}(u_i, v_i) \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{W}(u_i, v_i) \mathbf{Y} \tag{10.3}$$

where $\hat{\beta}(u_i, v_i)$ is the coefficient matrix for the predictors $\mathbf{X}$ at location i. $\mathbf{W}(u_i, v_i)$ is a diagonal weighting matrix that contains geographic weights on its diagonal elements for neighbors inside the neighborhood window (red circle in Fig. 10.3), and $\mathbf{Y}$ contains the variable being predicted. Note that the linear system above is similar to the general linear regression system given in Eq. 10.4.

$$\hat{\beta} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{Y} \tag{10.4}$$

where $\hat{\beta}$ is defined globally for the entire dataset. The geographic weights are inversely weighted with respect to the distance. Thus, the weights have large values for neighbors close to the regression location i. Different weighting schemes and neighborhood definitions are possible; the reader is encouraged to explore seminal work on this topic (Fotheringham et al. 2003).

Spatial representation via a weighting scheme can give GWR high predictive power for geospatial datasets in which a strong spatial autocorrelation is observed. The impact of incorporating spatial relationships in the regression model is demonstrated by comparing GWR with a nonspatial supervised machine learning method. In this example, GWR is juxtaposed against a random forest predictor for a problem with strong spatial autocorrelation in the data. Statistical climate downscaling (Wilby and Wigley 1997) was performed with GWR and a random forest regressor. Statistical downscaling calibrates the output of a global circulation model (GCM) to observed climate data such as temperature or precipitation. In this example, climate downscaling for the lower 48 US states; a regression model can be defined between 19 predictors (from GCM) and the observed average temperature. The regression model can be used to predict the average temperature for the entire lower 48 states. A random forest predictor can be trained using the observed average temperature and simulated GCM variables. The GWR model is formed using only 3 of the independent predictors due to the collinearity restriction of GWR. Below are the predicted average temperature profiles.

Note that the average temperature profile estimated in Fig. 10.4a depicts the patterns of temperature change captured in Fig. 10.4b. Even though fewer predictors are used in the GWR than in the random forest regressor, large-scale patterns in the temperature profile changes are captured. The GWR model in Fig. 10.4a was also compared to a random forest regressor model trained using the same three predictors. In that case, the GWR returned a mean-squared error that was 60% of that of the random forest regressor.

10.2.1.3 SVM

Support vector machine (SVM) is a supervised nonparametric statistical learning method (Corinna and Vapnik 1995). In its original form, the method comprises a set of labeled data instances and the SVM attempts to find a hyperplane that separates the dataset into a discrete predefined number of classes as consistently as possible for the training data (see Fig. 10.5) (Vapnik 1979). It is possible to generalize SVM to nonlinear kernels such as radial basis functions to learn and classify data sets with higher complexity (Schölkopf and Smola 2002).

As studied and discussed by Mountrakis et al. (2011), SVMs have been extensively employed in remote sensing and geospatial data analysis due to their ability to use small training data sets, often resulting a higher classification accuracy than the traditional methods (Mantero et al. 2005). For instance, SVM has been used in road extraction from IKONOS imagery by (Huang and Zhang 2009) assessing the influence of the slope/aspect of the terrain on the forest classification accuracy (Huang et al. 2008), a crop classification task (Wilson et al. 2004), and many more factors.
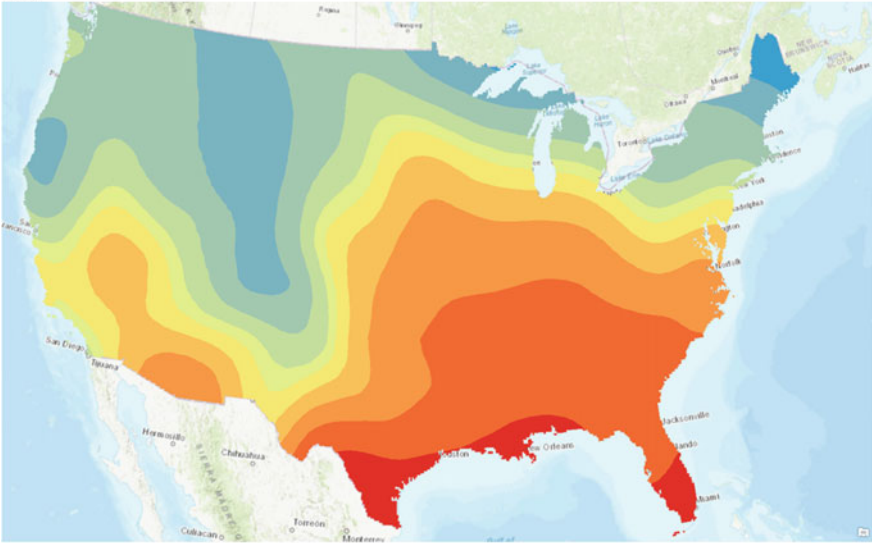
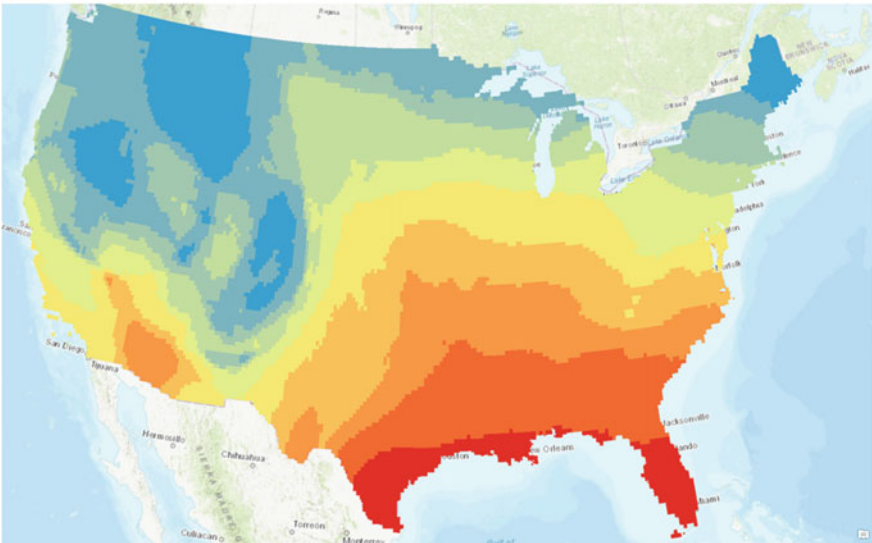**Fig. 10.4** **a** Downscaled temperature profile using GWR **b** downscaled temperature profile using a random forest regressor
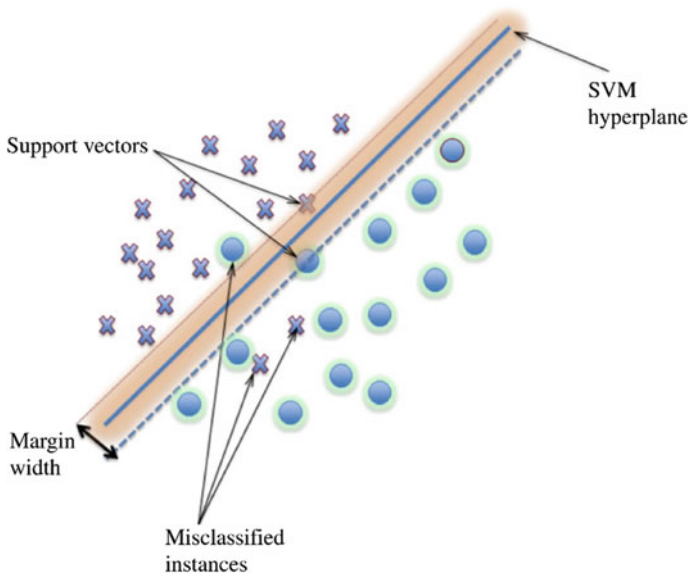
Fig. 10.5 SVM attempts to distinguish two categories of data by a hyperplane. Image from Mountrakis et al. (2011)

10.2.1.4 Active Contours and Active Shapes

Active contours or snakes have been developed with the aim of finding important features in an image by fitting a curve to the edges and lines of an image (Kass et al. 1988). Active contours are a set of energy-minimizing splines that are guided by external forces from the image. Snakes have been used extensively in geospatial image processing to detect features such as roads and buildings.

Active contours were later extended to active shapes to accommodate specific patterns in a set of objects and identify only those that are present in the training data (Cootes et al. 1995). In essence, they are very similar to active contours, but active shapes can only deform and fit the data that is consistent with the training set. Both active shapes and active contours have been extensively used in different applications of remote sensing and geoscience, such as object extraction (Liu et al. 2013), lane detection (Heij et al. 2004), and road extraction (see Fig. 10.6) (Kumar et al. 2017; Laptev 1997).

10.2.2 Unsupervised Learning

Unsupervised learning aims to infer the distribution of $P(X)$ in Eq. 10.1. Unlike supervised learning, $P(Y|X)$ or $P(X, Y)$ is not employed (Hastie et al. 2001). Thus,

Fig. 10.6 Active contours used to extract roads. Image taken form Laptev (1997)

unsupervised learning does not utilize any training dataset that contains information on $P(X, Y)$. One of the most common uses of unsupervised learning in geospatial analysis is in defining clusters and regions. These two terms differ, as clustering refers to defining groups based on value similarity in the data whereas regionalization performs clustering under spatial constraints (Duque et al. 2007). Both of these unsupervised learning approaches have wide applications (Duque et al. 2007; Hastie et al. 2001; Mitchell 1997; von Luxburg 2010). Most clustering and regionalization methods require definition of k, the number of clusters to divide X into. There are extensive surveys of clustering and regionalization in the literature for readers to refer to (Duque et al. 2007; Jain et al. 1999).

10.2.2.1 SKATER Algorithm

As discussed in Chap. 8, the K-means algorithm (Macqueen 1967) aims to partition X into k groups and minimize the intergroup dissimilarity with the assumption that minimal intergroup dissimilarity corresponds to distinct groups. K-means seeks to create groups that consist of similar elements, ensuring that dissimilar elements are assigned to different groups. Mathematically:

$$\mu(x) = argmin_C \sum_{i=1}^{k} \sum_{x \in c_i} \left\| x - \bar{C}_\iota \right\|^2 \tag{10.4}$$

where $C = \{C_1, C_2, \ldots, C_k\}$ is the group of clusters, with a cluster c_m consisting of a subset of X and $c_1 \cup c_2 \cup \cdots \cup c_k = X$. K-means has various uses in geospatial analysis, including detecting patterns in traffic accidents (Anderson 2009), analyzing landslides (Keefer 2000) and creating labels by clustering topo-climatic data (Burrough et al. 2001).

The SKATER algorithm is a regionalization algorithm that imposes graph-based spatial constraints on the k-means algorithm (Assunção et al. 2006). Unlike Lloyd's algorithm (Lloyd 1982), SKATER only assigns spatially contiguous and similar objects to the same cluster. Regionalization has vast uses in geospatial analysis,

including analysis of gerrymandering, healthcare services (Church and Barker 1998) and resource allocation (Or and Pierskalla 1979).

Clustering and regionalization were applied to the same dataset to juxtapose the types of patterns they expose in the data and the resulting understanding gained using these two methods. The average temperature in the United States in June 2012 was used. The resulting clusters and regions are displayed below.

The regionalization and clustering results in Fig. 10.7 show similarities in the overall temperature patterns, which change N-S in the eastern portion of the US and W-E in the western portion. Notably, the k-means result in Fig. 10.7b displays isolated patches whereas the regionalization result has spatially contiguous regions. Due to the constrained optimization scheme to satisfy the spatial constraints, the regions defined by regionalization have a higher variance than those in the k-means result. However, both maps display similarities in the temperature and the extent to which these similarities can be aggregated into homogeneous zones.

10.2.2.2   Autoencoders

Another very useful and common machine learning technique is autoencoders (Rumelhart et al. 1985). In an autoencoder, the data passes through a bottleneck, where the bottleneck is a lower representation of the same data. Autoencoders are made of two neural networks called the encoder and decoder (Fig. 10.8). The encoder receives data D, maps it to a lower space and obtains L; a decoder receives L, maps it back to the same dimension of D and obtains D'. The distance between D and D', which is called the reconstruction loss, should be minimized. A direct application of autoencoders is in compression, in which one can reduce the dimension of D to L and work with L and the decoder instead of the data D in its native resolution. Autoencoders have also been used in geospatial applications to find water bodies (Zhiyin et al. 2015) or denoise satellite images (Liang et al. 2017).

Machine learning techniques are not limited to the list of applications and methods provided here. Several variations of these methods as well as many other standalone techniques have been successfully employed in the Digital Earth, geoscience and remote sensing fields. For a more in-depth and comprehensive study, refer to the work of Lary et al. (2016).

10.2.3   Dimension Reduction

There have been extensive efforts to learn the patterns and forms that data sets contain. It is possible to predict the behavior of a data set and/or compress the data set into a more compact form for transmission, storage, and retrieval. In addition to autoencoders that can be used for dimensionality reduction, one of the easiest methods for compression and dimensionality reduction for a given data set and subsequent prediction of its behavior for unknown data points is *linear regression*.
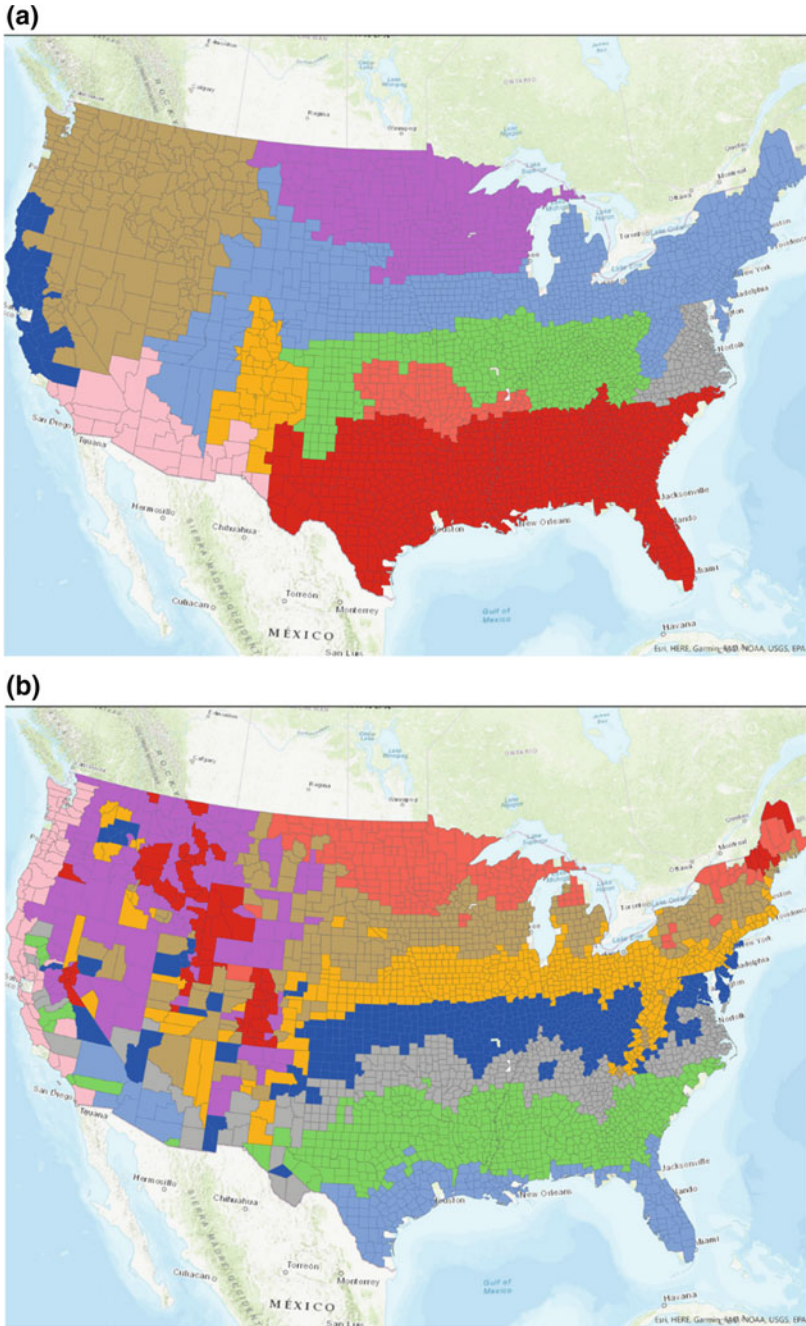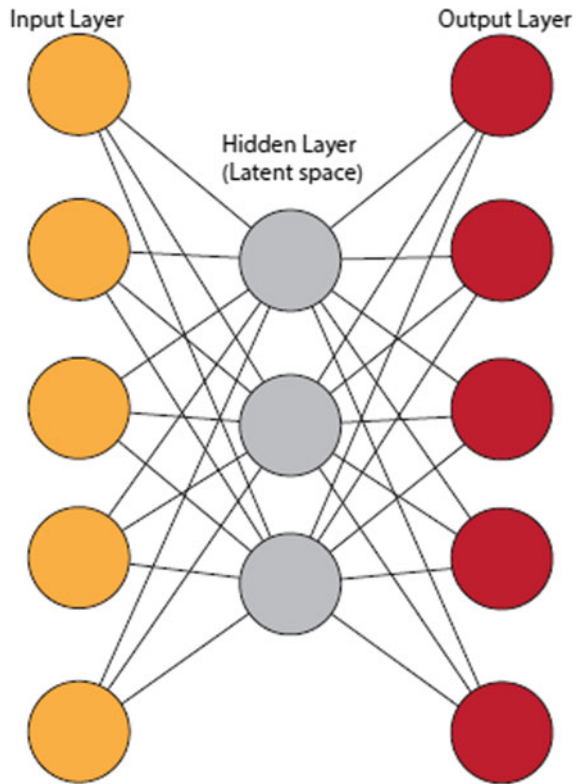
Fig. 10.7 **a** Temperature regions defined by SKATER **b** temperature regions defined by k-means

Fig. 10.8 The autoencoder passes the data (yellow neurons) through an encoder to learn a lower dimension (hidden/latent space; gray neurons) representation of the data. The decoder attempts to reconstruct the data (red neurons) as closely as possible to the given data

In the 2D case of this method, the data points attain two coordinates, and the line that best represents these data sets is considered as the model representative of the data. The best representation can have different meanings, including the line that has the smallest least square distance with all the data points. Regression, linear or nonlinear, has been a great tool to analyze spatial data. Belae et al. (2010) provided a survey of regression techniques used to represent and analyze spatial datasets. For Digital Earth platforms, Mahdavi-Amiri et al. (2018) combined regression with a wavelet to transmit quantitative datasets on a discrete global grid system (DGGS).

10.2.3.1   PCA

Another form of linear representation of a data set is principal component analysis (PCA). In this representation, the covariance matrix of the data is initially formed by applying the inner product of a data matrix A in its transpose $\left(Cov = A^T A\right)$. The eigenvectors of the covariance matrix, λ_i, represent the main trends of the data. If we have a data set forming an ellipsoid in 2D, the eigenvectors are the two main axes of the ellipsoid. Figure 10.9 represents PCA in 2D. PCA has been extensively used in

Fig. 10.9 PCA finds the main trends of the data. The data points illustrated in yellow have two main trends x′ and y′ that are the eigenvectors associated with the largest eigenvalues of the covariance of the data
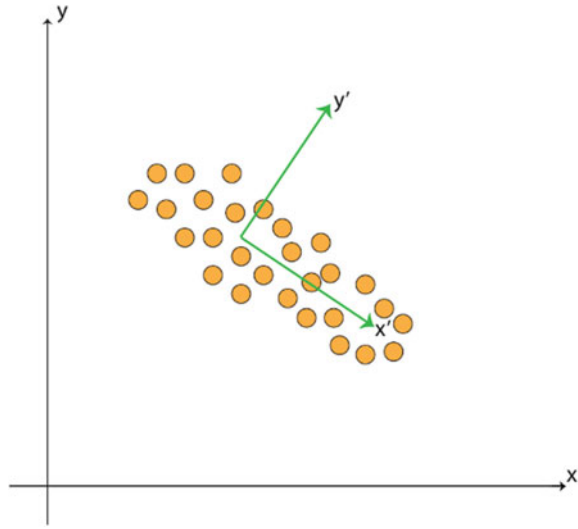
many applications including computer graphics, computer vision, and data science. PCA has been used in different applications related to geospatial data representation and geospatial data analysis (Demšar et al. 2013). For instance, PCA has been successfully used to study drought areas (Gocic and Trajkovic 2014), evaluate water quality (Parinet et al. 2004), and distinguish vegetation (Panda et al. 2009).

10.2.3.2 SVD

Singular value decomposition (SVD) is a decomposition that reveals important information about a matrix. In SVD, a matrix A is decomposed into the form USV^T, in which U and V are two rotation matrices and S is a diagonal scale matrix with values called the singular values, σ_i, of matrix A. There is a direct connection between PCA and SVD because the singular values of the singular value decomposition of data matrix A are the square root of the eigenvalues of the covariance matrix that is found in PCA $\left(\sigma_i = \sqrt{\lambda_i}\right)$. To compress or denoise data, it is possible to zero out small eigenvalues obtained by SVD and keep important portions of the data. SVD has been extensively employed in image processing applications (Sadek 2012). It has also been used in geospatial applications. For instance, Wieland and Dalchow (2009) used SVD to detect landscape forms, and Dvorsky et al. (2009) used SVD to determine the similarity between maps.

10.2.3.3   Evolutionary and Agent-Based Techniques

Evolutionary and Agent-based techniques have also been extensively used to perform analyses of geospatial data sets. Two important algorithms are genetic algorithms (GAs) and ant colony optimization (ACO).

In GAs, a set of random solutions is initially produced and these solutions are considered parents to make a new generation of solutions based on three rules: **Selection** rules that select parents based on their fitness, **Crossover** rules that combine two parents to generate children for the next generation, and **Mutation** rules that apply random changes to parents to form children (Mitchel 1998). GAs have been used in many applications in geospatial data analysis such as road detection (Jeon et al. 2002) and satellite image segmentation (Mohanta and Binapani 2011).

ACO is an optimization technique that works based in an agent-based environment. In this stochastic environment, the ants are agents that walk over a certain solution path and leave a track called a pheromone. Paths with more pheromone are usually more optimal (shortest) than others, and they attract more agents. A classic problem that can be solved by ACO is the travelling salesman problem. ACO has been successfully employed to solve other types of hard problems including those involving geospatial data analysis. For instance, ACO has been used for path planning considering traffic (Hsiao et al. 2004) and road extraction from raster data sets (Maboudi et al. 2017).

10.3   Deep Learning

When a large amount of data is involved and/or a complex model for representing the data is used, it is common to employ deep learning methods (Goodfellow et al. 2016). Digital earth data represents a massive amount of data, for example, high-precision digital elevation models or aerial photography. Because the rules that produce this kind of data are very complex and involve many natural or human processes, it can be difficult to apply standard learning models or algorithms and retain this complexity. Thus, the deep models described in this section are relevant.

10.3.1   Convolutional Networks

Deep learning has been popularized by image processing applications. In this context, the processed data is arranged into a regular grid and is adapted to so-called convolutional layers. Data extracted from Digital Earth can be of this nature by construction. For example, raster data such as digital elevation models or aerial photography images are already arranged into regular grids and can be processed out of the box with convolutional layers. Convolutional neural networks rely on the fact that the same processing can be applied to different parts of the image. Traditional
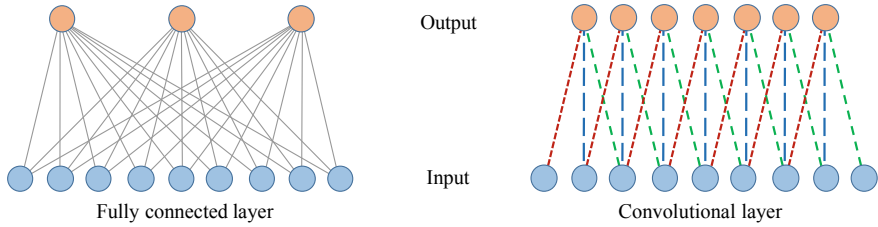
Fig. 10.10 Convolutional layers use fewer coefficients and are spatialized

fully connected schemes for neural network layers use many coefficients that can be spared with convolutional layers and used in other features. Figure 10.10 compares the principle of a convolutional layer to that of a traditionally fully connected layer. Both examples show an input of size 9. While a fully connected layer uses 27 coefficients to produce an output of size 3, the convolutional layer can produce 9 outputs from only 3 different coefficients. This means that the same feature extraction is performed but at different locations, which is relatively close to traditional convolution in the discrete domain.

Recently, a convolutional network was used to infer the super-resolution of a digital elevation model by using aerial photography (Argudo et al. 2018). Figure 10.11 shows the architecture of this network. This work comes from the observation that publicly available high-resolution DEMs (resolution less than 2 m) do not cover the full Earth whereas it is possible to find high-resolution imagery (orthophotos) with good coverage of the Earth. Many applications require a fine resolution for the DEM, and Argudo et al. proposed inserting details into a coarse DEM using inferred information drawn from the high-resolution orthophoto of the same footprint (Fig. 10.12). Basically, the method produces a DEM with 2 m precision from a DEM with 15 m precision and an orthophoto with 1 m precision. To produce this result, a fully convolutional network was used.

In the literature, a full system to automatically infer street addresses from satellite imagery was proposed (Demir et al. 2018a). One step that must be performed is the extraction of roads from the satellite images. This was done using a modified
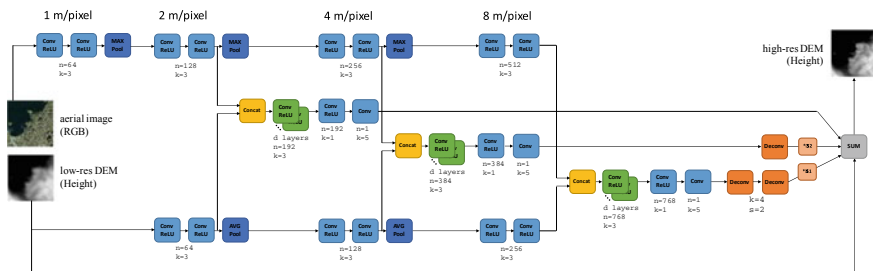

Fig. 10.11 A fully convolutional network was used to infer the high-resolution DEM from its coarse version and the high-resolution orthophoto (courtesy of O. Argudo et al.)
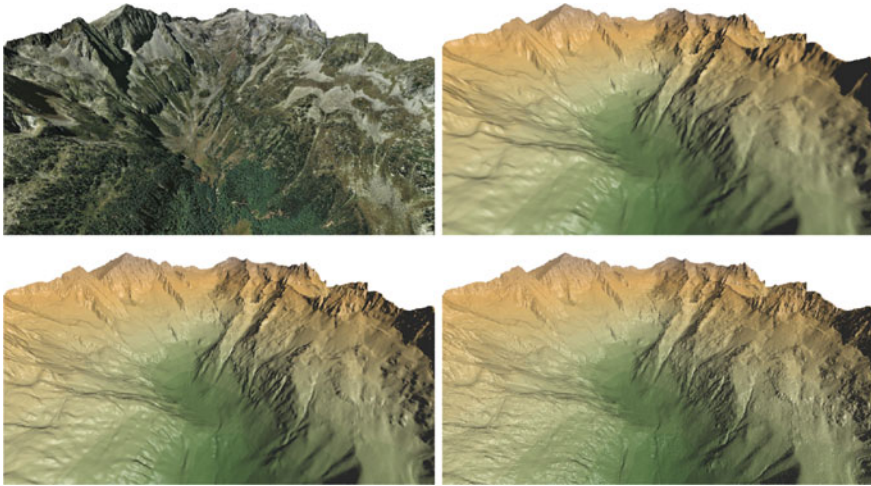
Fig. 10.12 Super-resolution of a 15 m precision DEM (top right) using an orthophoto (top left). Result (bottom left) and the ground truth reference (bottom right) (courtesy of O. Argudo et al.)

version of SegNet, a convolutional network primarily used for image segmentation. In this architecture, the input and output resolutions are identical, and the network consists of several encoder layers that decrease the resolution followed by decoders that increase the resolution. The network is trained using manually labeled 192 × 192 pixel images, in which a binary road mask is associated with each pixel of the image to indicate if the pixel belongs to a road or not. Figure 10.13 shows an example of the results obtained in automatic extraction of the road information compared with the ground truth.

More generally, automatic processing of satellite images with a deep learning approach appears to be very efficient in segmentation and feature extraction. The DeepGlobe project (http://deepglobe.org) aims to challenge authors to use deep learning for three applications: road extraction, building detection and land cover classification (Demir et al. 2018b).

Fig. 10.13 Automatic extraction of the road mask (right) from the satellite image (left), compared with the ground truth road network (center) (courtesy of I. Demir et al.)

Fig. 10.14 The schematic of
a recurrent neural network
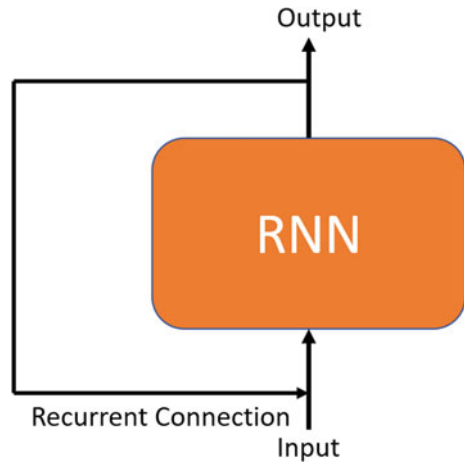
10.3.2 Recurrent Neural Networks

While convolutional neural networks and dense neural networks work well for static
data in which there is no sense of time, a **r**ecurrent **n**eural **n**etwork (RNN) (Jain and
Medsker 1999) processes data by iterating through the input elements and maintain-
ing a state that contains information relative to what it has seen until then. An RNN
is a neural network with an internal loop (see Fig. 10.14). The state of the RNN is
updated between processing independent sequences; therefore, we still consider one
data sequence as a single data point in the network. The difference is that this data
point is not processed in a single step as opposed to those in dense or convolutional
neural networks. In an RNN, the network internally loops over sequence elements
until it learns the flow of the data. An RNN is helpful when dealing with a temporal
data set. In geospatial data analysis, an RNN has been recently applied in interest-
ing applications such as correction of satellite image classification (Maggiori et al.
2017) and land cover classification (Ienco et al. 2017). Since many types of geospa-
tial data sets such as weather, satellite images, or seasonal animal behavior have
timing attached to them, we expect that RNNs will be widely used in the analysis of
geospatial data sets in the near future and that Digital Earth will benefit from such
networks.

10.3.3 Variational Autoencoder

Deep neural networks are useful to analyze data sets and are also helpful in generating
new data sets. It is possible to consider two deep neural networks as the encoder and
decoder of an autoencoder and produce a latent space that represents the data. Using
only L and an encoder, we can reproduce a lossy representation of D. However, it

is not possible to pick a vector in L and expect to reproduce a meaningful result by feeding it to the encoder because the distribution of L is unknown if autoencoders are used. In **v**ariational **a**uto**e**ncoders (VAEs) (Kingma and Welling 2014), in addition to the compression loss, another loss is minimized that forces the L to be a Gaussian distribution. Thus, VAEs can be used as a generative neural network in which one can sample the Gaussian distribution and feed it to the encoder to generate a new shape that does not necessarily belong to the training data set. Although VAEs have potential to generate data and learn low-dimensional data for geospatial data sets, VAEs have not been extensively tested for geospatial data analysis and generation.

10.3.4 Generative Adversarial Networks (GANs)

Similar to VAEs, **g**enerative **a**dversarial **n**etworks (GANs) (Goodfellow et al. 2014) are also generative models. GANs consist of a pair of networks that have two different and adversarial roles. These networks have a convolutional architecture and are often complex to retain the complexity of the underlying models. The first network is a generator that we denote as G, which attempts to generate the best result, for example, an image. Then, the second network takes the image as input and tries to infer if it is a generated image or not. This second network is called a discriminator and we denote it as D. Both G and D are trained alternatively. The objective of G is to fool D whereas D aims to avoid being fooled by G. The strength of this kind of adversarial formalism is that it is equivalent to use of a very complex function to train the generator G (encoded into the discriminator), far more complex than traditional distance would be.

Conditional **G**ANs (cGANs) are GANs with a particular setup in which the discriminator is trained to recognize the matching between an input image A and an output B whereas a traditional GAN only tests the plausibility of the output without any knowledge of the input. The training principle of a cGAN is explained in Fig. 10.15.
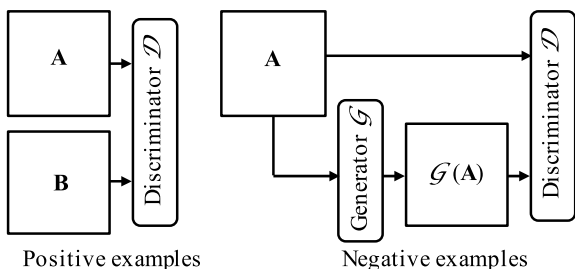

Fig. 10.15 cGAN principle: a training pair (A, B) is used to learn positive examples. For negative examples, only A is used together with the generator to form the pair (A, G(A))

Conditional GANs have recently been used to automatically generate digital elevation models from user sketches (Guérin et al. 2017). The user sketches the river network, the crests and some altitude cues and obtains a plausible terrain that matches the given constraints, based on a training dataset made of sketch/terrain pairs. The method consists of building such a dataset by extracting the sketch from a real-world terrain. The difficulty of this kind of setup is to automatically build a sketch that is compatible with user sketches, i.e., similar to what a user would draw. Building a sketch that is too close to the terrain features will force the user to draw very precisely, which is not relevant in a sketching context but would be useful in a reconstruction process. The digital elevation model must be simplified to produce simpler features. In their work, Guérin et al. propose initially downsampling the digital elevation model and then smoothing it. This coarse digital elevation model is then processed by a flow simulation, from which the skeleton is extracted. The same process is applied to extract ridges. This feature extraction is illustrated in Fig. 10.16.

The training dataset is formed of pairs that describe the matching between the sketch and the terrain. Figure 10.17 gives examples of such pairs. To create a more pliable terrain synthesizer, the sketches randomly include one, two or the three features among the river lines, crest lines and altitude cues.

Figure 10.18 shows examples of outputs produced by the DEM generator from sketches. The results were obtained by using training from a DEM extracted from the NASA SRTM dataset at 1 arc-second from different locations in the United
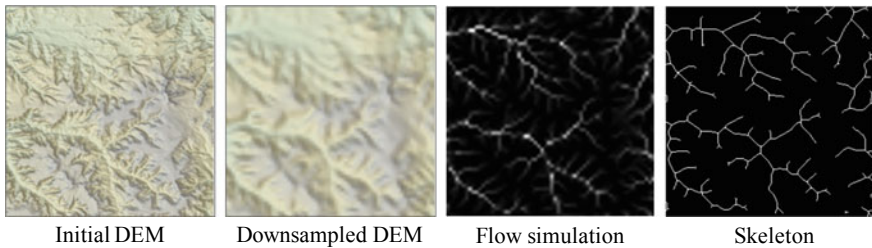

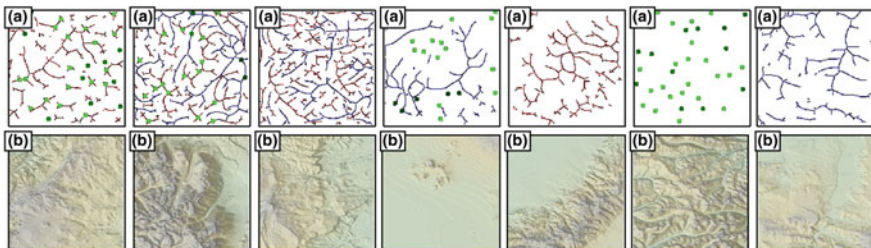| Initial DEM | Downsampled DEM | Flow simulation | Skeleton |

Fig. 10.16 Training database examples

Fig. 10.17 Training database examples. Training pairs are formed by a sketch (**a**) and an associated DEM (**b**). Sketches can feature river lines (blue), crests (red) and altitude cues (green)
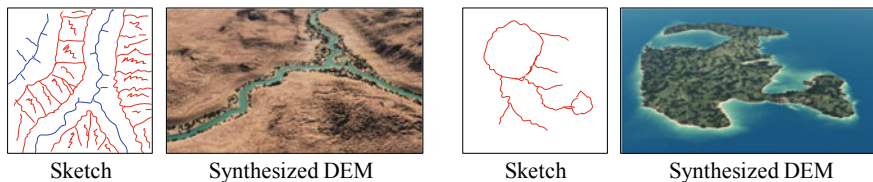
Fig. 10.18 Examples of generated digital elevation models from simple sketches. A canyon generated using river and crest lines (left). A volcanic island generated using only crest lines (right)

States. In the same article, the authors proposed the use of the same principle to automatically generate digital elevation models from a single level set sketch. They also described examples of automatic void filling in digital elevation models. Finally, because cGANs can embed very complex models, they used it to mimic an erosion process.

10.3.5 Dictionary-Based Approaches

Approaches based on base function decompositions have intrinsic limitations. Base functions are usually used because they have orthogonality properties that lead to an efficient decomposition. Selecting the base can be difficult because it heavily depends on the nature of the signal. Thus, it can be a viable option to use dictionary-based descriptions. A signal is represented as a linear combination of atoms from a dictionary. Atoms do not need to have special properties such as orthogonality. They are typically chosen directly from the data by picking the most representative signals or by using an optimization. A survey of dictionary-based methods for 3D modeling was conducted by Lescoat et al. (2018). One of the applications of dictionary-based modeling is called sparse modeling, which adds an additional constraint on the number of atoms used to represent the final signal, called *sparsity*.

10.3.5.1 Dictionary Decomposition

Given a dictionary, the decomposition of a signal consists of finding the best atom, i.e., the atom that maximizes the projection. Then, the same process is applied iteratively to the residual until reaching the target sparsity. This process is called matching pursuit and was introduced by Mallat and Zhang (1993). This decomposition algorithm was further improved by Cai and Wang (2011) by introducing the *Orthogonal Matching Pursuit* (OMP) algorithm. The main difference is that the best decomposition of the already-found atoms is recomputed after each new atom is found.

Fig. 10.19 An example of terrain amplification that adds plausible details from a given exemplar using a dictionary-based approach. The original terrain had a precision of 1 km, and successive amplifications by a factor of 4 increase the precision to 4 m

10.3.5.2 Dictionary Optimization

One aim of dictionary based approaches is to find a dictionary that is adapted to a given context or set of signals. This can be done by an optimization process. One goal of this optimization is to minimize the reconstruction error, for example, by computing an L^2 distance between the reconstructed signal and the original. It is common to add a constraint on the type of decomposition, for example, by setting a maximum sparsity. Unfortunately, the optimization problem under this type of constraint is too difficult to solve in an optimal way. Heuristics have been proposed that lead to good results with a relatively low cost. K-SVD is one of these algorithms (Aharon et al. 2006), which consists of iterating between two steps. The first step consists of optimizing the decomposition, which can be done using a standard OMP algorithm. The second step optimizes the dictionary with respect to the previously computed decomposition. The two steps are repeated until a number of iterations is reached or a given error is obtained.

Several applications of sparse modeling with terrains have been proposed by Guérin et al. (2016) and Argudo et al. (2018). The terrain is decomposed into patches that compose input signals. A so-called amplification process is used to introduce plausible details into the terrain by mapping between low-resolution and hi-resolution atoms. The dictionary is drawn from an exemplar terrain at high resolution and automatically transformed into low resolution by a trivial downsampling process. The amplification algorithm simply decomposes the patches from a given terrain in the low-resolution dictionary and uses the corresponding high-resolution atoms to reconstruct it. Because the dictionary has been extracted from real terrain, the added details are plausible and realistic, as shown in Fig. 10.19.

10.3.6 Reinforcement Learning

Reinforcement **l**earning (RL) is a powerful learning method in dynamic environments (Sutton and Barto 1998). In RL, there is usually an agent in an environment and the agent receives rewards based on its actions. The final goal is to learn how to take actions to maximize the rewards. At any time t, an environment is defined by states S_t in which an agent can take action A_t and change the environment state to S_{t+1}. When
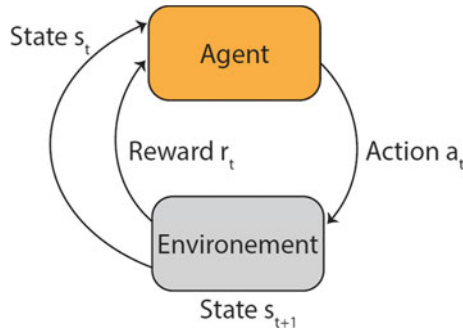
Fig. 10.20 An agent receives state s_t, performs an action and receives reward r_t from the environment. The state of the environment changes to s_(t+1). This process continues until a terminal state is achieved

the agent takes action A_t, the environment receives a reward r_t. These iterations continue until the environment reaches a terminal state (Fig. 10.20). Examples of applications that RL can be extremely useful for are games or robot locomotion in which more points and more stable states are the rewards of the game and locomotion environments, respectively.

RL has also been used in applications in GIS and geospatial data analysis. For instance, RL has been used to model land cover changes (Bone and Dragicevic 2009). With recent advances in RL and the growth of computational power, we expect that RL will receive more attention from the GIS and Digital Earth communities. For instance, one application of RL can be to simulate the behavior of endangered species in different simulated environments.

10.4 Discussion

In the past, machine learning has seen hypes and winter seasons. It started with symbolic AI in the 1960s, which claimed the ability to make machines with intelligence comparable to an average human being in less than a decade. However, people soon realized that they were far from reaching that point. In the 1980s, with the rise of *expert systems*, similar hype was seen in the area of machine learning, followed by a winter season due to the lack of generality of expert systems and their high maintenance costs (Chollet 2017). Recently, deep learning methods became popular again and showed great success in different areas of computer science including geospatial analysis, which is an important portion of Digital Earth platforms. Deep learning will likely continue to grow and be applied more in this field, especially because of the availability of computational power and big data sets that help create more powerful models. However, deep learning cannot solve all problems. For instance, current deep learning models are unable to solve problems that require reasoning or long-term planning (Chollet 2017). Deep learning models work extremely well in

mapping an input to a desired output with very little human-level knowledge about the input or output and their effect on industry and science will probably remain for a very long time. There is plenty of discussion about the future of deep learning and AI, notably by its great pioneers such as Lecun et al. (2015) and in the European perspective on AI (Craglia et al. 2018).

Artificial intelligence and particularly machine learning and deep learning have great potential to contribute to the generation, analysis, and management of geospatial data sets. Digital Earth should benefit from such opportunities, as a place holder to represent such data sets and a platform to analyze them. Since Digital Earth is constantly receiving geospatial data sets, a successful Digital Earth should use reliable, fast, and comprehensive techniques to manage and make use of such data. Deep Learning techniques show promise in these directions. However, there are still issues in their use in Digital Earth platforms that must be addressed. In the following sections, we discuss some of these issues.

10.4.1 Reproducibility

If a technique such as a deep neural network produces particular results, such results should be reproduceable by others. Placing the code on GitHub and providing free access to data sets have been helpful for this issue. However, there are still some issues, especially when the data are owned by a company or the network was designed by an industrial team. In particular neural network architectures, randomness can be included, usually to improve the training. When this randomness is also present in the operational network, it can disrupt the reproducibility of results.

10.4.2 Ownership and Fairness

Ownership of artifacts provided by machine learning techniques is also heavily under question. If a person with almost no knowledge about a network takes information from available sources, modifies a few parameters, takes data from an available source and produces something unique or obtains a certain analysis, who is the owner of such results? The data owner, developer of the network, or the person who combined these ingredients? In more serious scenarios, who is at fault when a system that works based on machine learning techniques makes a catastrophic mistake or performs a discriminatory action that may involve racism or sexism? Another question is whether data sets and computation power are available to everyone, i.e., do we have "data democratization"? Fortunately, the wealth of free access data sets and code bases along with cheap computational power such as Amazon Web Services (AWS) have resolved some of these issues but we are still far from perfect.

10.4.3 Accountability

Due to the nature of some algorithms involved in machine learning, it usually cannot be used in contexts where accountability is a strong constraint. This is especially the case with deep neural networks where a lot of information is hidden in the layers, which can lead to unexpected and unwanted results. Conversely, traditional machine learning methods such as linear regressions or PCA are very reliable even if they are limited in terms of applications. Reasonably, one could consider using deep learning methods only when traditional methods fail or are lacking.

10.5 Conclusion

In conclusion, we provided a sampling of artificial intelligence techniques and their applications in geospatial data generation, analysis, and management. We discussed how AI can be beneficial for generating new terrain data sets, identifying roads and analyzing various geospatial data sets such as satellite imagery. AI techniques and deep learning methods appear very promising. Extensive research on these topics will likely make them even more suitable for use in different domains including geospatial analysis and Digital Earth. However, these techniques are unfortunately standalone and have not been integrated into a Digital Earth platform that makes use of such techniques. Appropriate artificial intelligence techniques should be meticulously included in Digital Earth, considering their pros and cons including fairness and bias to provide interactive, comprehensive and meaningful analysis to users.

References

Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311–4322

Anderson TK (2009) Kernel density estimation and K-means clustering to profile road accident hotspots. Accid Anal Prev 41(3):359–364

Argudo O, Chica A, Andujar C (2018) Terrain super-resolution through aerial imagery and fully convolutional networks. Comput Gr Forum 37(2):101–110

Assunção RM, Neves MC, Câmara G et al (2006) Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. Int J Geogr Inf Sci 20(7):797–811

Beale CM, Lennon JJ, Yearsley JM et al (2010) Regression analysis of spatial data. Ecol Lett 13(2):246–264

Bone C, Dragićević S (2009) Defining transition rules with reinforcement learning for modeling land cover change. Simulation 85(5):291–305

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Bunge W (1966) Gerrymandering, geography, and grouping. Geogr Rev 56(2):256–263

Burrough PA, Wilson JP, van Gaans PFM et al (2001) Fuzzy k-means classification of topo-climatic data as an aid to forest mapping in the Greater Yellowstone Area, USA. Landsc Ecol 16(6):523–546

Cai TT, Wang L (2011) Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Trans Inf Theory 57(7):4680–4688

Chollet F (2017) Deep learning with python. Manning Publications Company, Shelter Island

Church J, Barker P (1998) Regionalization of health services in Canada: a critical perspective. Int J Health Serv 28(3):467–486

Cootes TF, Taylor CJ, Cooper DH et al (1995) Active shape models-their training and application. Comput Vis Image Underst 61(1):38–59

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Cortés U, Sangüesa MS-MR, Comas J et al (2001) Knowledge management in environmental decision support systems. AI Commun 14(1):3–12

Craglia M, Annoni A, Benczur P et al (2018) Artificial intelligence: a european perspective. Publications Office of the European Union, Luxenbourg

Cutler DR, Edwards Jr. TC, Beard KH et al (2007) Random forests for classification in ecology. Ecology 88(11):2783–2792

Demir İ, Hughes F, Raj A et al (2018a) Generative street addresses from satellite imagery. ISPRS Int J Geo-Inf 7(3):84

Demir I, Koperski K, Lindenbaum D et al (2018b) DeepGlobe 2018: a challenge to parse the Earth through satellite images. Comput Vis Pattern Recognit. https://arxiv.org/abs/1805.06561 Accessed 2 Aug 2019.

Demšar U, Harris P, Brunsdon C et al (2013) Principal component analysis on spatial data: an overview. Ann Assoc Am Geogr 103(1):106–128

Duque JC, Ramos R, Suriñach J (2007) Supervised regionalization methods: a survey. Int Reg Sci Rev 30(3):195–220

Dvorský J, Snášel V, Voženílek V (2009) Map similarity testing using matrix decomposition. In: 2009 international conference on intelligent networking and collaborative systems, IEEE, Barcelona, 4–6 November 2009

Egenhofer MJ, Mark DM (1995) Naive geography. In: Frank AU, Kuhn W (eds) Spatial information theory A theoretical basis for GIS. Springer, Heidelberg, pp 1–15

Fonseca FT, Egenhofer MJ, Agouris P et al (2002) Using ontologies for integrated geographic information systems. Trans GIS 6(3):231–257

Fotheringham AS, Brunsdon C, Charlton M (2003) Geographically weighted regression: the analysis of spatially varying relationships. Wiley, Chichester

Gislason PO, Benediktsson JA, Sveinsson JR (2006) Random forests for land cover classification. Pattern Recognit Lett 27(4):294–300

Gocic M, Trajkovic S (2014) Spatiotemporal characteristics of drought in Serbia. J Hydrol 510:110–123

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Goodfellow I, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C et al (eds) Advances in neural information processing systems 27 (NIPS 2014). NIPS, Montreal, pp 2672–2680

Gruber TR (1993) Toward principles for the design of ontologies used for knowledge sharing? Int J Hum-Comput Stud 43(5):907–928

Guérin E, Digne J, Galin E et al (2016) Sparse representation of terrains for procedural modeling. Comput Gr Forum 35(2):177–187

Guérin É, Digne J, Galin É et al (2017) Interactive example-based terrain authoring with conditional generative adversarial networks. ACM Trans Gr 36(6):1–13

Hastie T, Tibshirani R, Friedman JH (2001) The elements of statistical learning: data mining, inference, and prediction. Springer, New York, NY

Heij R, Helgers M, Kockelkorn W et al (2004) Snakes for lane detection. Image Vis Comput 269–280

Hsiao Y-T, Chuang C-L, Chien C-C (2004) Ant colony optimization for best path planning. In: IEEE international symposium on communications and information technology, 2004. ISCIT 2004, IEEE, Sapporo, 26–29 October 2004

Huang H, Gong P, Clinton N et al (2008) Reduction of atmospheric and topographic effect on Landsat TM data for forest classification. Int J Remote Sens 29(19):5623–5642

Huang X, Zhang L (2009) Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. Int J Remote Sens 30(8):1977–1987

Ienco D, Gaetano R, Dupaquier C et al (2017) Land cover classification via multitemporal spatial data by deep recurrent neural networks. IEEE Geosci Remote Sens Lett 14(10):1685–1689

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323

Jeon B-K, Jang J-H, Hong K-S (2002) Road detection in spaceborne SAR images using a genetic algorithm. IEEE Trans Geosci Remote Sens 40(1):22–29

Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vis 1(4):321–331

Keefer DK (2000) Statistical analysis of an earthquake-induced landslide distribution — the 1989 Loma Prieta, California event. Eng Geol 58(3):231–249

Kingma D, Welling M (2014) Auto-encoding variational bayes. Mach Learn. arXiv preprint arXiv: 1312.6114

Kuipers BJ (1996) An ontological hierarchy for spatial knowledge. In: Proceedings of 10th international workshop on qualitative reasoning about physical systems. AAAI Press, Fallen Leaf, pp 113–120

Kumar P, Lewis P, McCarthy T (2017) The potential of active contour models in extracting roads from mobile laser scanning data. Infrastructures 2(3):1–16

Laptev I (1997) Road extraction based on snakes and sophisticated line extraction. Master's thesis, Royal Institute of Technology

Lary DJ, Alavi AH, Gandomi AH et al (2016) Machine learning in geosciences and remote sensing. Geosci Front 7(1):3–10

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

Lenat DB (1995) CYC: a large-scale investment in knowledge infrastructure. Commun ACM 38(11):33–38

Lescoat T, Ovsjanikov M, Memari P et al (2018) A survey on data-driven dictionary-based methods for 3D modeling. Comput Gr Forum 37(2):577–601

Liang P, Shi W, Zhang X (2017) Remote sensing image classification based on stacked denoising autoencoder. Remote Sens 10(1):16

Liu G, Sun X, Fu K et al (2013) Interactive geospatial object extraction in high resolution remote sensing images using shape-based global minimization active contour model. Pattern Recognit Lett 34(10):1186–1195

Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137

Maboudi M, Amini J, Hahn M et al (2017) Object-based road extraction from satellite images using ant colony optimization. Int J Remote Sens 38(1):179–198

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: le Cam LM, Neyman J (eds) Proceedings of the fifth berkeley symposium on mathematical statistics and probability, volume 1: statistics. University of California Press, Berkeley, pp 281–297

Maggiori E, Charpiat G, Tarabalka Y et al (2017) Recurrent neural networks to correct satellite image classification maps. IEEE Trans Geosci Remote Sens 55(9):4962–4971

Mahdavi-Amiri A, Alderson T, Samavati F (2015) A survey of digital earth. Comput Gr 53:95–117

Mahdavi-Amiri A, Alderson T, Samavati F (2018) Geospatial data organization methods with emphasis on aperture-3 hexagonal discrete global grid systems. Cartographica 54(1):30–50

Mallat SG, Zhang Z (1993) Matching pursuits with time-frequency dictionaries. IEEE Trans Signal Process 41(12):3397–3415

Mantero P, Moser G, Serpico SB (2005) Partially supervised classification of remote sensing images through SVM-based probability density estimation. IEEE Trans Geosci Remote Sens 43(3):559–570

McCarthy J (1988) Mathematical logic in artificial intelligence. Daedalus 117(1):297–311

Medsker L, Jain LC (1999) Recurrent neural networks: design and applications. CRC Press, Boca Raton

Mitchell T (1997) Machine learning. McGraw Hill, Burr Ridge

Mitchell M (1998) An introduction to genetic algorithms. MIT Press, Cambridge

Mohanta RK, Binapani S (2011) A review of genetic algorithm application for image segmentation Int J Comput Technol Appl 3(2):720–723

Mountrakis G, Im J, Ogole C (2011) Support vector machines in remote sensing: a review. ISPRS J Photogramm Remote Sens 66(3):247–259

Nelder JA, Wedderburn RWM (1972) Generalized linear models. J Royal Statist Soc Ser A (Gen) 135(3):370–384

Or I, Pierskalla WP (1979) A transportation location-allocation model for regional blood banking. AIIE Trans 11(2):86–95

Panda SS, Hoogenboom G, Paz J (2009) Distinguishing blueberry bushes from mixed vegetation land use using high resolution satellite imagery and geospatial techniques. Comput Electron Agric 67(1):51–58

Parinet B, Lhote A, Legube B (2004) Principal component analysis: an appropriate tool for water quality evaluation and management—Application to a tropical lake system. Ecol Model 178(3):295–311

Rizzoli AE, Young WJ (1997) Delivering environmental decision support systems: software tools and techniques. Environ Model Softw 12(2):237–249

Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Institute for Cognitive Science, University of California, San Diego, La Jolla

Rykiel EJ (1989) Artificial intelligence and expert systems in ecology and natural resource management. Ecol Model 46(1):3–8

Sadek R (2012) SVD based image processing applications: state of the art, contributions and research challenges. Int J Adv Comput Sci Appl 3(7):26–34

Schölkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge

Sowa JF (2000) Knowledge representation: logical, philosophical, and computational foundations. Brooks/Cole, Pacific Grove

Sutton R, Barto A (1998) Introduction to reinforcement learning. MIT Press, Cambridge

Vapnik V (1979) Estimation of dependences based on empirical data. Springer, New York

von Luxburg U (2010) Clustering stability: an overview. Found Trends Mach Learn 2(3):235–274

Wieland R, Dalchow C (2009) Detecting landscape forms using Fourier transformation and singular value decomposition (SVD). Comput Geosci 35(7):1409–1414

Wilby RL, Wigley TML (1997) Downscaling general circulation model output: a review of methods and limitations. Prog Phys Geogr: Earth Environ 21(4):530–548

Wilson MD, Ustin SL, Rocke DM (2004) Classification of contamination in salt marsh plants using hyperspectral reflectance. IEEE Trans Geosci Remote Sens 42(5):1088–1095

Zhiyin W, Long Y, Shengwei T et al (2015) Water body extraction method based on stacked autoencoder. J Comput Appl 35(9):2706–2709

Eric Guérin is Assistant Professor at INSA de Lyon, France. He received an engineering degree from INSA de Lyon in 1998 and a Ph.D. in Computer Science from Université Claude Bernard Lyon 1 in 2002. His research interests include procedural modeling of virtual worlds, natural phenomena and machine learning. He is head of the LIRIS/Geomod team since 2015.

Orhun Aydin is a researcher in Environmental Systems Research Institute and assistant professor in Spatial Sciences Institute at U.S.C. Prior to joining ESRI and USC, he obtained a Ph.D. and M.Sc. in Geostatistics from Stanford University. Orhun's research pertains to modelling, exploring and forecasting spatial and spatio-temporal phenomena and their applications for Earth Systems.

Ali Mahdavi-Amiri is a University Research Associate at Computing Science department in Simon Fraser University. His research interests are in Computer Graphics with specialty in Geometric Modeling, Geospatial Modeling and Visualization, Digital Fabrication, and Machine Learning. Ali has received several awards including NSERC Postdoctoral Fellowship, Graduate Achievement Award, Michael A. J. Sweeney Award, and Teaching Excellence Award.