Geotech Geol Eng (2022) 40:735–750
https://doi.org/10.1007/s10706-021-01921-w

ORIGINAL PAPER

An Exploratory Investigation into Image-Data-Driven Deep Learning for Stability Analysis of Geosystems

Zhen Liu · Shiyan Hu · Ye Sun · Behnam Azmoon

Received: 12 December 2019 / Accepted: 24 June 2021 / Published online: 25 August 2021

Abstract This study investigates image-data-driven deep learning in the stability analysis of geosystems. For the purpose, the recent breakthrough in computer vision represented by the Convolutional Neural Network (CNN), which was later used as a core technique in developing Google's AlphaGo, was studied for its capacity in assessing the stability of retaining walls. The concept used in the famous Dogs vs. Cats Kaggle challenge, in which machine learning algorithms are used to classify whether an image contains a dog or a cat, was employed. A CNN was used to analyze images for retaining walls to tell whether a wall is "cat" (safe) or "dog" (failed). For quantitative analysis, 2D images for retaining walls, organized as datasets of sizes from 500 to 200,000, were generated using a stochastic method and labeled using a traditional mechanistic method. An accuracy of 97.94% was achieved for predicting whether the retaining wall is safe via binary classifications with the CNN. Testing via the analysis of 20,000 additional images, which were independent and identically distributed, confirmed the results. Further investigations into the dataset sizes and computational power yielded quantitative insights into the influence of data and computing resources on the application of deep learning in the stability analysis of geosystems. The study, for the first time, proves the feasibility of stability analysis of geosystems with image data and provides a potential big data solution for geotechnical engineering as well as other civil engineering areas.

Keywords Image data · Deep learning · Stability analysis of geosystems · Convolutional neural network · Big data

Z. Liu (✉) · B. Azmoon
Department of Civil, Environmental, and Geospatial Engineering, Michigan Technological University, 1400 Townsend Drive, Dillman 201F, MI 49931 Houghton, USA
e-mail: zhenl@mtu.edu

B. Azmoon
e-mail: bazmoon@mtu.edu

S. Hu
Electronics and Computer Science, University of Southampton, ECS University of Southampton University Road, SO17 1BJ Southampton, UK
e-mail: S.Hu@soton.ac.uk

Y. Sun
Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, 1400 Townsend Drive, MEEM 926, Houghton, MI 49931, USA
e-mail: yes@mtu.edu

1 Introduction

This paper presents an exploratory effort for connecting geotechnical engineering to Artificial Intelligence (AI)–one area that is significantly changing our lives in recent years. Since the breakthrough in 2006

(Hinton and Salakhutdinov 2006), deep learning with Deep Neural Networks (DNN) has been helping machines to gain intelligence to outperform humans in many aspects: from text processing (Collobert and Weston 2008), speech recognition (Dahl et al. 2012; Hinton et al. 2012) and driverless cars (Nguyen et al. 2015) to disease diagnosis (Fakoor et al. 2013; Li et al. 2014) and stock market forecasting (Ding et al. 2015). In a recent triumph, Google's DNN program, AlphaGo, beat a top Go professional (Silver et al. 2016). These remarkable advances in AI are deemed as the 4th industrial revolution by many people (Kelnar 2016) and are reshaping many areas of modern society including scientific research (Ohlsson 2011). One of the biggest changes is the shift of focus from models to data (Kitchin 2014). Deep learning enables knowledge extraction from massive amounts of data, making the data outweigh the machine (or model). This shift happens in this era of big data, when data are explosively expanding due to the widespread application of low-cost sensors, electronic devices, and large-volume storage (Manyika et al. 2011). In geotechnical engineering, there have been incalculable and explosively increasing data from Geophysical Information Systems (GIS), weather stations, field monitoring, case studies and lab testing, the majority of which are collected/stored as images and live images (videos). But unfortunately, these data are mostly underutilized because they cannot be analyzed with the traditional model-driven methods in geotechnical engineering.

2 Background

2.1 Traditional Neural Network Application in Geotechnical Engineering

The precursor of deep learning is the Artificial Neural Networks (ANNs)–a topic that has been extensively studied in geotechnical engineering. ANNs are a form of AI attempting to mimic the behavior of the human brain and nervous system (Shahin et al. 2001). A surge of interest was sparked in the late 1980s and widely spread in 1990s in geotechnical engineering. ANNs were applied to almost all aspects of geotechnical engineering with some degree of success (Park et al. 2011; Sakellariou and Ferentinou 2005; Shahin et al. 2001, 2008). Specifically, ANNs have been used extensively for modeling the load capacities of deep foundations (Ahmad et al. 2007; Ardalan et al. 2009; Das and Basudhar 2006; Goh et al. 2005; Hanna et al. 2004; Shahin 2010; Shahin and Jaksa 2009), ground anchors (Shahin and Jaksa 2005, 2006), behavior of shallow foundations (Chen et al. 2006; Kuo et al. 2009; Padmini et al. 2008; Provenzano et al. 2004; Shahin et al. 2005, 2003, 2002), constitutive relationships of soils (Banimahd et al. 2005; Basheer 2002; Fu et al. 2007; Gao et al. 2004; Habibagahi and Bamdad 2003; Hashash et al. 2004; Lefik and Schrefler 2003; Najjar and Huang 2007; Shahin and Indraratna 2006), shear strength and stress history (Baykasoğlu et al. 2008; Byeon et al. 2006; Çelik and Tan 2005; Kaya 2009; Kurup and Dudani 2002; Lee et al. 2003; Narendra et al. 2006; Yang and Rosenbaum 2002), swelling pressure (Ashayeri and Yasrebi 2009; Erzin 2007), compaction and permeability (Abdel-Rahman 2008; Das et al. 2011; Sinha and Wang 2008), soil composition and classification (Bhattacharya and Solomatine 2006; Kurup and Griffin 2006), dynamic properties (Garcıa et al. 2006; Kaya 2016; Romo and Garcıa 2003), dams (Kim and Kim 2008; Yu et al. 2007), blasting (Lu 2005), mining (Singh and Singh 2005), environmental geotechnics (Shang et al. 2004), rock mechanics (Gokceoglu et al. 2004; Ma et al. 2006; Maji and Sitharam 2008; Millar and Hudson 1994; Singh et al. 2007, 2005; Sitharam et al. 2008), site characterization (Caglar and Arman 2007), tunnels and underground openings (Alimoradi et al. 2008; Benardos and Kaliampakos 2004; Chen et al. 2009; Neaupane and Achet 2004a, b; Neaupane and Adhikari 2006; Yoo and Kim 2007), slope stability and landslides (Cho 2009; Deng and Lee 2001; Erzin and Cetin 2013; Farrokhzad et al. 2011; Ferentinou and Sakellariou 2007; Goh and Kulhawy 2003; Kanungo et al. 2006; Kostić et al. 2016; Mayoraz and Vulliet 2002; Neaupane and Achet 2004a, b; Neaupane and Achet 2004a, b; Shangguan et al. 2009; Wang et al. 2014; Wang and Sassa 2006), and subsurface characterization (Gangopadhyay et al. 1999; Moayedi and Hayati 2018; Samui and Sitharam 2010). Traditional ANNs have also been applied to the stability analysis of earth-retaining structures on lateral deflection of walls (Chua and Goh 2005; Goh et al. 1995; Hopmans et al. 2002; Kung et al. 2007), lateral thrust and its point of application (Yildiz et al. 2010), risk of serviceability (Goh and Kulhawy 2005), internal stability (Kasa et al. 2011), frequency (Heidari

2011), optimal design (Gowda et al. 2012), and seismic response (Ozturk 2014).

Despite the interest and success, the application of ANNs as well as other AI approaches in geotechnical engineering has been stagnating. This can be seen from the conclusive remarks of the two widely-cited review papers of Shahin and his co-workers, which summarized the practice before and after 2001 (Shahin et al. 2001, 2008). The 2005 review paper was presented to "attract more geotechnical engineers to pay better attention to this promising tool" while the 2009 paper commented that the "neural networks for the time being might be treated as a complement to conventional computing techniques rather than as an alternative". The advantage of ANNs is its ability in handling pattern recognition and modeling of non-linear relationships of multivariate dynamic systems (Sakellariou and Ferentinou 2005), especially those complex engineering problems that are beyond the computational capability of traditional research approaches (Shahin et al. 2008). The disadvantage is the concerns in robustness, transparency, extrapolation, and uncertainty. In summary, the AI tide represented by ANNs did not replace traditional geotechnical methods or even become widely-accepted alternatives. The reason is most likely that there were not many problems that ANNs can handle and traditional geotechnical methods cannot, while ANNs are less connected to physics and did not show a satisfactory level of intelligence back then.

2.2 Recent Breakthroughs in Deep Learning and Image Recognition

Most of the studies summarized in the above subsection were built on Artificial Neural Network (ANN) techniques which had matured before the 1990s. This surge of interest quickly evaporated with the slump ANN research in the late 1990s. With the ascent of Support Vector Machines and the failure of backpropagation, ANN research enters a dark time in the early 2000s, a time when even the founders of deep learning, i.e., LeCun and Hinton, had their papers and grant proposals routinely rejected due to their subject being neural networks. As a result, as Hinton tells, they hatched a "conspiracy": "rebrand" the frowned-upon field of neural nets with "Deep Learning". However, the turning point was not the name, but a breakthrough published in Science in 2006 (Hinton and Salakhutdinov 2006), in which Hinton and his co-workers proved the possibility of training deep networks using the restricted Boltzmann machine. Real momentum for the deep learning snowball was gained because of the work in computer vision initiated in (Krizhevsky et al. 2012), which beat other machine learning methods in the ILSVRC-2012 computer vision competition. Since then, such competitions have been predominated by deep learning algorithms. The core idea of the deep learning in computer vision, or more specifically, image recognition/classification, is the use of a Convolutional Neural Net with pooling and convolution layers, variations in the input data, very efficient GPU implementation, ReLU neurons, and dropout layers. These life-changing breakthroughs were soon sensed by many industry giants such as Microsoft, Google, and IBM, which significantly promoted the development of deep learning (Hinton et al. 2012). This innovation started from computer vision and speech recognition soon swept other areas such as driverless cars, natural language processing, drug discovery and toxicology, biomedical informatics, customer relationship management, and stock prediction.

Despite its new birth, this innovation has been becoming a force changing the world, including many research areas. Taking computer linguistics for example, deep learning hit this area as a "tsunami" in the 2015 Natural Language Processing (NLP) conference (Manning 2016)–this major computer linguistics conference was predominated by deep learning within one year. Engineering disciplines have been significantly impacted by computer science innovations, which can be seen from NSF's promotion of new concepts such as "big data", "cyber-physical system", and "smart city". Based on these facts, one may be wondering "when will deep learning especially the above image-data-driven machine learning technique majorly impact engineering disciplines?" and "what a role will geotechnical engineering play in this process?".

2.3 Trend and Difficulty in Data-Driven Studies

Data is another reason that deep learning is becoming a world-changing force (Chen and Lin 2014). Deep learning models or machines, especially those DNNs dealing with image data, enable learning from image and video data, which account for a major part of the "big data" (Kaisler et al. 2013; Wu et al. 2014).

Learning from these data significantly improves the machines' intelligence. The knowledge gained from the data has been surpassing that from the improvements of the machines. As a result, data have become a major resource. For instance, big data trends have been driving the future of businesses, leading to the wide acceptance of and transitions to data-driven marketing and data-driven companies. In industry giants such as Google, "data drives 'everything'" (Metz 2009).

All the above changes have been occurring in the background of "big data explosion" (Lohr 2012). In geotechnical engineering, there exists a great amount of GIS data and more and more data created by remote sensing techniques. Additionally, the decreasing costs of cameras and storage devices help popularize the documentation of testing and construction in terms of images and videos. Despite the explosive occurrence of such data, our primary research tools, such as traditional physically-based analytical and numerical methods, only take a few inputs. Thus, most of the data mentioned above are underutilized or totally wasted. From another angle, these data, including the knowledge contained in the data, are barely extracted or utilized.

3 Methodology

3.1 Overview

The goal of this study is to understand the image-data-driven deep learning in geosystems with an exploratory investigation into the stability analysis of retaining walls. To achieve the goal, the recent breakthrough in computer vision (Russakovsky et al. 2015), which was later used as one of the core techniques in the development of Google's AlphaGo (Moyer 2016), is studied for its capacity in assessing the stability of a typical geosystem, i.e., retaining walls. The core concept enabling the breakthrough, Convolutional Neural Networks (CNN), which helped machines surpass humans in visual classification capacity, is used to process the big data in geotechnical engineering, which primarily consist of images and live images (videos) and cannot be analyzed using traditional geotechnical methods. As shown in Fig. 1, the concept used in the famous Dogs vs. Cats Kaggle challenge, in which machine learning algorithms are used to classify whether an image contains a dog or a cat, is employed. A CNN is employed to analyze images for a retaining wall to tell whether the wall is a "cat" (safe) or a "dog" (failed). For quantitative analysis, 2D images for retaining walls are generated using a stochastic method and analyzed using a traditional mechanistic method for labeling. These labeled image data are used as input to train CNNs for supervised learning. The trained CNN is tested against another independent set of data generated in the same way as the training data. The study is proposed based on the hypothesis that telling a failed structure from a safe one is not essentially different from telling a dog from a cat in images in deep learning.

3.2 Convolutional Neural Networks

CNN, which are also frequently called ConvNets, are a special type of feed-forward network. CNNs are inspired by the connectivity pattern between neurons in the animal visual cortex. Each cortical neuron responds to stimuli only in a restricted region of the visual field, i.e., receptive field. The receptive fields of different neurons partially overlap to cover the entire visual field, i.e. the image. Such an organization empowers the CNN with an ability to extract features automatically: each neuron captures the local features in the image area that it can see, and such features can be extracted into a higher-level of features by another layer of neurons. As shown in Fig. 2, a CNN first identifies line features and then further extracts shapes. This overcomes one major drawback in traditional image classification algorithms, in which a lot of effort is required for manual pre-processing. In other words, the CNN learns the filters that are hand-engineered in traditional algorithms. This independence from prior knowledge and the save of human effort in the feature design are a major advantage. In addition, CNNs can contain special layers such as convolutional layers and pooling layers, allowing the networks to encode certain image properties. Due to the above characteristics, CNNs exhibit exceptional performance in visual recognition tasks.

3.3 Architecture

As shown in Fig. 3, a CNN is usually composed of an input layer, multiple hidden layers, and an output layer. These hidden layers typically consist of convolutional layers, pooling layers, ReLU layers, fully

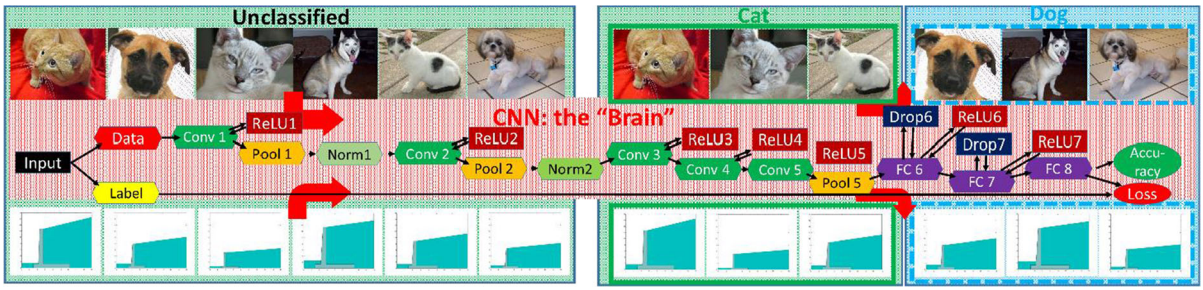

Fig. 1 Concept of stability analysis with binary classification using deep learning

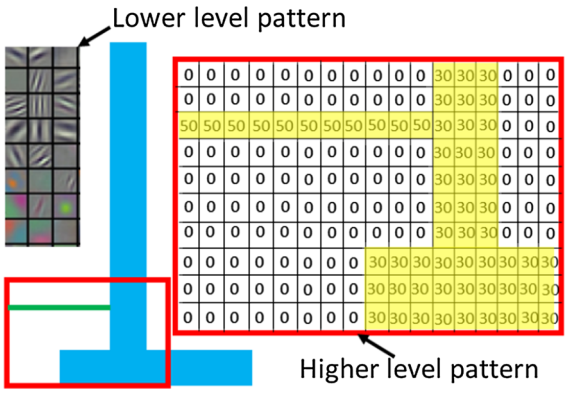

Fig. 2 Schematic of feature extraction in CNN

connected layers, and normalization layers. The convolutional, pooling, and ReLU layers act as learnable feature extractors, while the fully connected layers act as a machine learning classifier. Furthermore, the early layers (close to input) of the network encode generic patterns of the images, while later layers encode the detailed patterns of the images. It is noted that only the convolutional layers and fully-connected layers have weights. Learning these weights is the primary goal of the training phase. A CNN with weights learned from labeled samples is a trained model, which can be used to classify/predict unlabeled samples (or called instances). This subsection for architecture explains the key components of the CNN used in this study.

3.3.1 Convolutional Layer

Theoretically, fully connected feedforward neural networks (without convolutional and pooling layers) can also be used to learn features and classify image data. But, this use is still not realistic at this time because this architecture requires a number of neurons to learn from pixels. The required computation cost would be too high to afford due to the pixels numbers of normal image data. The convolutional layers help solve this problem by reducing the number of free parameters and consequently allowing the network to be deeper with fewer parameters (Aghdam and Heravi 2017). Attributed to this innovation, the CNN resolves the vanishing or exploding gradients problem in training deep neural networks with many layers.

As shown in Fig. 4, a convolutional layer consists of a set of learnable filters (or called neurons or kernels) that are slid over the image spatially. Mathematically, this corresponds to computing dot products between the entries of the filter, i.e., the matrix in

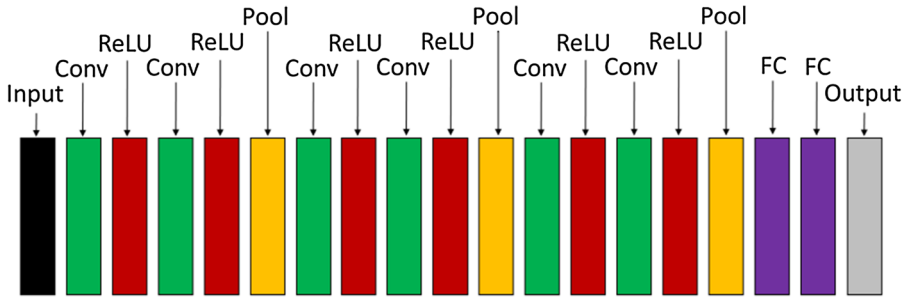

Fig. 3 Architecture of a typical CNN

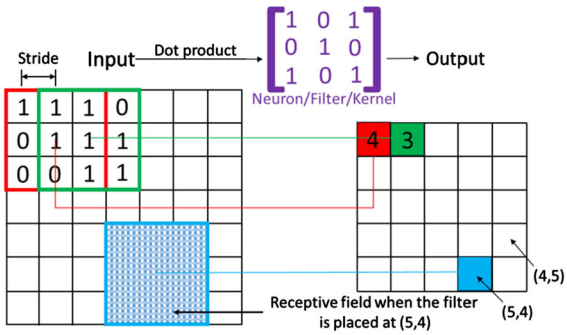

Fig. 4 Neurons of a convolutional layer connected to their receptive fields

Fig. 4, and different regions of the same size from the input image (Stanford's deep learning tutorial). The entries of the filter are the weights of the convolutional layer. Therefore, if the filter sliding over different image regions was viewed as different filters, the filters in the convolutional layer are essentially identical. These filters will be activated when they identify the same specific structures (features) in the images, e.g., lines, depending on the size of the image regions that the filers are applied to. It is worthwhile to mention that the filters should extend to the full depth of the input image. Thus, for a colored image, the filter should have a depth of 3 to cover all 3 color channels (red, green, blue) of the image. The mathematical convolution operation emulates the response of an individual neuron to visual stimuli (Glauner 2015).

3.3.2 Pooling Layer

CNNs may include local or global pooling layers, which combine the outputs of multiple neurons from one layer into a single neuron in the next layer (Ciresan et al. 2011; Krizhevsky et al. 2012). Two common types of pooling layers are the max pooling and average pooling. The max pooling uses the maximum value from a cluster of neurons as the output, as illustrated in Fig. 5, while the average pooling uses the average value. Mathematically, pooling is a non-linear down-sampling. The use of a pooling layer can progressively reduce the spatial size of the image data, and consequently, reduce the number of parameters and computation operations for the network. Also, pooling can also remove redundant specific features and consequently help control over-fitting. This study adopts the max pooling, and detailed

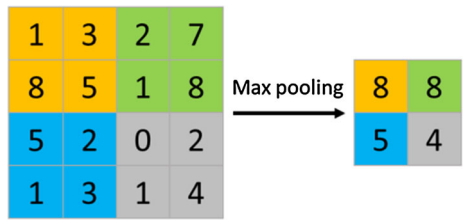

Fig. 5 Mathematical operations in max pooling using a 2 × 2 filter and a stride of 2

parameters for these pooling layers will be presented later.

3.4 Geosystem Stability Analysis with CNN Classifier

Many existing CNNs can be applied to analyze the image data of geosystems with minor modifications. Since there have been no attempts at applying this relatively new computer vision technique to stability analysis of geosystems. There are still several issues that significantly differentiate the application of CNNs in geosystems from the normal use of CNNs in computer science and thus are critical to the success of such efforts. Described in this sub-section are three critical issues that were learned in this pioneering study. The detailed setup of learning algorithms will be introduced in the next section for deep learning and results.

3.4.1 Labeled Data

As explained in the introduction section, data are a major driving force for deep learning. No study has been reported on the use of image data for stability analysis of geosystems. Therefore, it is unclear how to obtain and utilize such data. In this study, the major purpose is to prove the feasibility of applying image-data-driven deep learning to geosystems, so the focus was placed on clean image data with 2D profiles of retaining walls generated by a computer. In this way, a large amount of high-quality data can be obtained without being distracted by less-relevant image pre-processing such as de-noising and thus can focus on the validation of the concept.

As shown in Figs. 2d and 6 images for cantilever retaining wall systems with different geometries were constructed using random numbers generated by

MATLAB. As shown in Fig. 6, the retaining wall and the soil bodies on its two sides were generated in a 18 m × 18 m rectangular region (canvas). The parameters in Table 1 were used to generate the image data to reach a compromise between the data variability and the classification balance (between failed and safe). As the first attempt at image-data-based deep learning for the stability analysis of geosystems, this study is focused on the geometry and thus adopts fixed material properties, which will be explored in the follow-up studies. The cohesion, internal friction angle, and density of the soil are 30 kPa (c), 30° (ϕ), and 18.5 kN/m^3 (γ), respectively; and the density of the retaining wall (assumed to be made of concrete) is 23.5 kN/m^3 (γ_c).

Labeled training datasets with 500, 1000, 2000, 5000, 10,000, 20,000, 400,000, 800,000, 120,000, and 200,000 images were generated with the above geometries for labeling, i.e., safe and failed. For each training dataset, 5/6 of the images were used for training and the rest 1/6 were used for validation (also called cross-validation in many deep learning publications). 5000 additional images were generated in the same way for testing. All the image data for training, validation, and testing were generated independently.

The retaining walls in the generated images were analyzed using Rankine's pressure theory for their stability against overturning (Das 2015). The images were labeled as safe or failed, depending on weather ratio between the stabilizing momentum, M_s, and overturning momentum, M_o, is greater than 1.

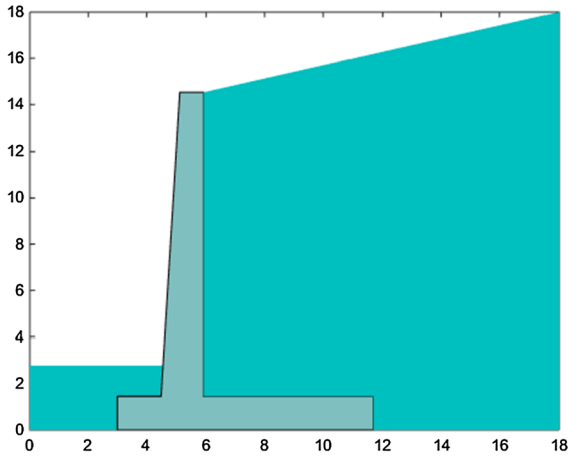

Fig. 6 Example of image data used in deep learning

$$SF = \frac{M_s}{M_o} \tag{1}$$

$$\begin{aligned} M_s = {} & \gamma B_h (H - H_b)\left(B - \frac{B_h}{2}\right) + \frac{1}{2}\gamma B_h (H_1 - H) \\ & \left(B - \frac{B_h}{3}\right) + \gamma_c B_{s1}(H - H_b)\left(B_t + B_{s2} - \frac{1}{2}B_{s1}\right) \\ & + \frac{1}{2}\gamma_c (B_{s2} - B_{s1})(H - H_b)\left(Bt + (B_{s2} - B_{s1})\frac{2}{3}\right) \\ & + \gamma_c B H_b \frac{B}{2} + P_p \frac{D}{3} + P_a \sin(\alpha) B \end{aligned} \tag{2}$$

$$M_o = P_a \cos(\alpha)\frac{H_1}{3} \tag{3}$$

where $P_a = \gamma H_1^2 K_a/2 - 2cH_1\sqrt{K_a}$ and $P_p = \gamma H_1^2 K_p/2 + 2cH_1\sqrt{K_p}$ are active and passive Rankine's forces, respectively, in which $K_a = \tan(\pi/4 - \phi/2)^2$ and $K_p = \tan(\pi/4 + \phi/2)^2$ are Rankine's active and passive pressure coefficients, respectively. No safety factors were introduced in this pure mechanistic analysis. An assumption underneath the use of the above theory is that the theory has no model bias especially any potential bias/uncertainty in the calculation of Rankine's earth pressure coefficients.

3.4.2 *Transfer Learning*

Another critical technique to ensure the success of the image-data-driven deep learning in geosystems is transfer learning. It was found that, if the CNN is trained from scratch, that is, with initial weights generated from random numbers, the accuracy would remain at a very low level, e.g., less than 70%. The transfer learning technique was proven to be a very practical and powerful technique for building geosystem classifiers.

The idea behind transfer learning is analogous to cooking: for cuisine, it will be much more efficient to buy household food ingredients from groceries stores instead of growing the vegetables and hunting for the meats. Transfer learning starts the training from a trained model on a different dataset and adapts the trained model to the problem to be addressed. There are two strategies for transfer learning. The first strategy is to fine-tune the trained model, either the whole network or some layers, directly on the new

Table 1 Parameters for generating the image data of retaining walls

Parameter	Description	Formulation
H	height of the wall	$4 + 11\rho$
B	length of the base slab	$(0.5 + 0.4\rho)H$
B_t	length of the toe	$0.1\,H$
B_{s2}	width of the bottom of the stem	0.1 H
I	slope of the front surface of the stem	0.2 + 0.03 ρ
B_{s1}	width of the top of the stem	$B_{S2} - H \cdot I_s$
B_h	width of the heel	$B - B_t - B_{S2}$
α	angle of the soil behind the wall	20 ρ
H_b	height of the base	0.1 H
D	height of soil in front of the toe	$H_b + 0.1 \cdot \rho \cdot H (D \leq 0.6)$

ρ is a random number between 0 and 1 generated by MATLAB. The parameters and design correlations are from Das (2015)

dataset via backpropagation. The second strategy is to use a trained model as a fixed feature extractor: replacing the fully-connected layers with new ones while fixing the other layers (as feature extractors) in training for new models. This study adopted the first strategy and an existing well-trained CNN, i.e., bvlc_reference_caffenet (Caffe's Model Zoo library), which was trained on the ImageNet dataset consisting of millions of images across 1000 categories.

3.4.3 Deep Learning with GPU

Another factor critical to the success of the image-data-driven deep learning in geosystems is the use of GPU, which is also a key to the recent advances in computer vision. CNNs require large datasets and a lot of computational time to train. Although training small CNNs for simple problems with normal CPUs could be possible, our computing experiments indicated that a GPU is essential. It takes days to train our CNN with an 8-core Intel i7-4790 3.6 GHz CPU using 4000 images. However, the time is shortened to less than one hour with an entry-level GPU, i.e., NVIDIA K620, and to several minutes with advanced GPU, i.e., NVIDIA K80. The deep learning framework used in this study, i.e., Caffe (Jia et al. 2014), provides options for GPU computation.

4 Deep Learning and Results

This section presents details for building, training, validating, and post-processing a CNN using an open-source deep learning framework, Caffe, and Python. The results will also be delivered and discussed.

4.1 Procedure

Caffe is a deep learning framework developed by the Berkeley Vision and Learning Center, which comes with a C++, a Python, and a MATLAB interface. The Python interface was adopted and most of the deep learning work was conducted in a Linux environment (Red Hat distribution). The following are the five steps in deep learning with a CNN using Caffe.

1. Data preparation: The images as input are first cleaned and stored in a format that can be used by Caffe. A Python script was developed to handle both the pre-processing and storage of the images.
2. Model definition: A CNN is defined by specifying its architecture parameters in an architecture configuration file with the extension of.prototxt.
3. Solver definition: The training of CNN using techniques such as back-propaga2. tion is by nature an optimization process. The parameters of this optimizer, or called solver, were defined in a model configuration file with the extension of.prototxt.
4. Model training: Training consisting of alternate supervised learning and validation is carried out in iterations by executing one Caffe command in a shell (e.g., Bash in Linux). The trained model in terms of weights is stored in a file with the extension of.caffemodel.
5. Prediction and Testing: The trained model in.caffemodel files is utilized to make predictions of data independent of the training data. If such data are labeled, then the predictions can be used to validate the trained model. A Python script is written to automate the prediction process.

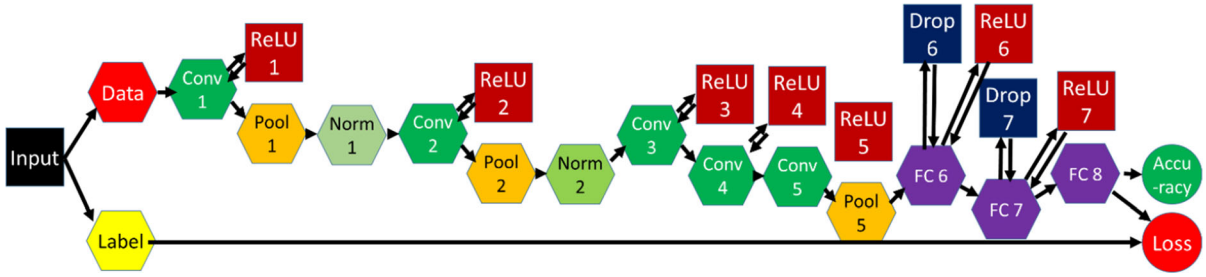

Fig. 7 Architecture of the CNN used in this study

4.2 Data Preparation

First, the image data with labels, i.e., safe or failed, was generated for both training/validation and prediction/testing, using the geometries and random numbers given in the previous section (Labeled Data Subsection). The retaining wall region and the soil region were filled with different colors. The edge of the retaining wall region used a different color. The two types of images, i.e., training (including cross-validation) and testing (prediction), were stored in two different folders.

Next, the image data were histogram-equalized and resized. In detail, histogram equalization was applied to all the training images (three channels) for adjusting the contrast of the images. The images were then resized to 227 × 227 pixels. The training data were divided into two sets: 5/6 of the training images for training and the test 1/6 for validation. The training, validation, and testing data were stored as separate LMDB databases, which are a common format of input data for Caffe.

Next, a Python module from the Python interface of Caffe, i.e., compute_image_mean, was utilized to obtain the mean image of the training data. Then, the mean image was subtracted from each input image to ensure that every feature pixel has zero mean. This is a common preprocessing step in supervised machine learning with image data to help improve the deep learning outcome. The pre-defined mean values of the R, G, B channels are 104, 117, and 123, respectively.

4.3 Model Definition in Caffe

In this study, attempts were made to obtain a model using a CNN with an architecture shown in Fig. 7. As can be seen, this CNN, which is named GeoNet, contains a deep net consisting of multiple convolutional (Conv), ReLU, pooling (Pool), normalization, drop-out (Drop), and fully-connected (FC) layers. The arrows between layers show the direction of the data flow. The architecture was developed from a popular CNN model called AlexNet (Krizhevsky et al. 2012). This architecture was proven to be suitable for the binary classification tasks in this study, while more recent CNNs with deeper and wider architectures such as ResNet and Inception can be employed for more complex tasks.

The parameters of GeoNet was stored in a file with the extension of.prototxt. Table 2 lists the major parameters for different layers.

4.4 Solver Definition in Caffe

The solver in Caffe specifies the parameters for optimization. These parameters were defined in a.prototxt file. In detail, the solver adopted an adaptive learning rate method, AdaDelta (Zeiler 2012) for the optimization, and the initial learning rate was fixed at 0.001. The weight decay term for preventing overfitting is 0.0005 (weight_decay). The maximum training iteration number was set to 40,000 (max_iter). If this value is too small, convergence cannot be guaranteed; if it is too large, oscillations may be produced, leading to a waste of training time. Other parameters such as display frequency (display) and result recording frequency (snapshot) do not impact the training and

Table 2 Major model configuration parameters

Layer	Description
Conv1	output number: 96, kernel size: 4, stride: 4
Pool1	type: Max, kernel size: 3, stride: 2
Norm1	type: LRN, local size: 5, alpha: 0.0001, beta: 0.75
Conv2	output number: 256, pad: 2, kernel size: 5, group number: 2
Pool2	type: Max, kernel size: 3, stride: 2
Norm1	type: LRN, local size: 5, alpha: 0.0001, beta: 0.75
Conv3	output number: 384, pad: 1, kernel size: 3
Conv4	output number: 384, pad: 1, kernel size: 3, group number: 2
Conv5	output number = 256, pad: 1, kernel size: 3, group number: 2
Pool5	type: Max, kernel size: 3, stride: 2
Fc6	type: InnerProduct, output number: 4096, weight filler type: Gaussian, std: 0.005
Drop6	dropout_ratio: 0.5
Fc7	type: InnerProduct, output number: 4096, weight filler type: Gaussian, std: 0.005
Drop6	dropout_ratio: 0.5
Fc7	type: InnerProduct, output number: 2, weight filler type: Gaussian, std: 0.01
Accuracy	type: accuracy
Loss	type: SoftmaxWithLoss

The default values in Caffe were used if not specified. The batch sizes for training, validation, and testing are both 40 for NVIDIA K620 GPU (constrained by the available GPU memory) and 40 and 400 for NVIDIA K80 GPU

validation results, and thus, can be chosen based on the needs and available resources. The solver mode is GPU (solver_mode).

4.5 Training

With the model and solver defined in the above subsections, the training (including validation) process was started with the "train" tool provided by Caffe via command line in Bash as follows: "path_to_train train –solver path_to_solver/solver_name.prototxt 2 > &1 | tee path_to_log/log_name.log". The model parameters were defined in a file named "solver_name.prototxt". This files also contains the paths for both the model definition file and the result file. Depending on the result recording frequency, the trained model, i.e., weights for neurons in the network, were saved as a file in the path and name that was specified in the solver file, appended with the iteration number. For example, if the snapshot_prefix was defined as path_to_result/model_name, then the trained model after 2000 iterations would be stored in the file "model_name_2000" in the folder of "path_to_result".

As mentioned in the previous section, if deep learning is conducted using the above data, model, and solver, it is very difficult to obtain good training results. A much more efficient way is to start training from a trained model, which was not necessarily trained with our data or even image data containing the same type of object. This is because most images share similar low-level features such as lines. Here a very useful fine-tuning technique, transfer learning, was utilized. The transfer learning in this study was conducted with bvlc_reference_caffenet from Caffe's repository, Model Zoon, which has been widely used by researchers and machine learning practitioners to share their trained models. The model bvlc_reference_caffenet was trained with the ImageNet dataset which contains millions of images across 1000 categories. The command was very similar to that for training from scratch and can also be used to restart a training process from any previously-stored model. The model configuration file used in the transfer learning was identical to that in the original training process.

The training alternates between training and validation iterations. The loss in the training and the loss and accuracy in the validation were saved to the log file at the frequency specified in the solver configuration file (via model parameters "display" and "test_interval"). The variations of the loss and the accuracy can be plotted in real time during the training (including validation) process using Python scripts or plotted manually after the training is finished. A script file was developed based on open-source code CaffePlot.py.

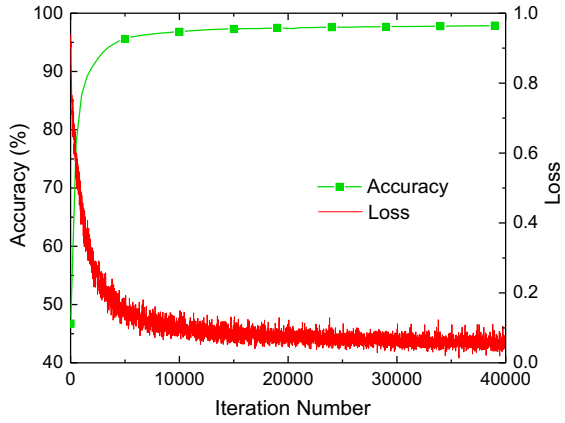

Fig. 8 Variation of accuracy and loss in the training process with 200 k images

Typical variations of the accuracy and loss (in the validation) are shown in Fig. 8. With NVIDIA K620 GPU, it took 14 h for GeoNet to achieve an accuracy of 97.94%, which scores the output as the accuracy of output with respect to the target. The loss in Figure, L, was calculated as

$$L(w) = \frac{1}{N}\sum_{i=1}^{N} f_w\left(X^i\right) + \lambda r(w) \tag{4}$$

where $f_w(X^i)$ is the loss at X^i and $r(w)$ is a regularization term avoiding overfitting. Thus, it has been shown that an accuracy exceeding the traditional geotechnical ANNs' can be achieved with an entry-level CNN and a small amount of computational effort while dealing with data whose size (e.g., Gigabytes), however, exceeds the size of the data processed by traditional ANNs (e.g., bytes) by orders (e.g., 9 orders).

4.6 Prediction and Testing

After training, the trained model can be applied to predict/analyze image data that are independent of the training data. Ideally, it is hoped the trained model can be used to predict data that are out of the range of the training data–achieving a good generalization ability. This generalization ability has been tested to some extent in the testing process. In fact, when the validation dataset is independent of the training dataset, validation serves a role similar to testing. However, the validation may not be totally independent of the training, because the training and validation data, though exclusive of each other, can still be related depending on the sampling technique adopted. In this study, the training and validation datasets were generated by randomly splitting the training (and validation) datasets before training and remained unchanged during training. However, to provide a totally independent evaluation of the trained CNN, the trained model was used to analyze the 20,000 independent retaining wall images generated in the same way as the training and validation data. Such data were also labeled using the result from the mechanistic analysis. As a result, the labeled data can be used to validate the predictions, which provides us with a second validation of GeoNet.

The predictions were carried out with packages from the Python interface of Caffe. The images were first pre-processed in the same way as that used for the training and validation data. The same trained model, GeoNet, was used for predictions; therefore, the net architecture remains the same except for changing the final output layers from loss and accuracy layers to a Softmax layer for classifications, which is needed for predictions.

In binary classification problems, it is common to use a confusion matrix consisting of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) to measure the outcome of the. The prediction results for 20,000 images are listed in Table 3. The accuracy is defined as

$$A = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

The accuracy of the classification using the trained GeoNet is 97.90%. As can be seen, the precision is very close to the accuracy obtained in the testing process, e.g., 97.94%. These results confirmed the high accuracy of the trained CNN in analyzing the stability of the geosystems via image data.

Table 3 Confusion matrix for the prediction results with the trained model

Predicted	Actual	
	Safe	Failed
Safe	TP = 9797	FP = 218
Failed	FN = 203	TN = 9782

Sometimes, a trained model can exhibit significantly different performance in different aspects of predictions, i.e., in the selections of safe instances and failed instances. To further evaluate the performance of the trained CNN, the precision and recall were calculated. Precision P, also called positive predictive value, is the fraction of relevant instances (accurately predicted as safe) among the retrieved instances (predicted as safe); while recall R, also known as sensitivity, is the fraction of relevant instances (accurately predicted as safe) that have been retrieved over the total amount of relevant instances (safe):

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

The precision and recall are 97.82% and 97.97% respectively. Despite the slight difference, both of these two values are very close to the accuracy. Therefore, the trained CNN has a good performance in different aspects of the predictions. It is worthwhile to mention that an accuracy value of 97.90% does not mean that the deep learning method is inferior to the classic mechanistic methods, which can obtain a 100% accuracy here. In this study, the classic mechanistic stability analysis method was just used as a reference. As we know, no mechanistic stability method is accurate due to the high uncertainty in geotechnical system analysis with typical safety factors ranging from 1.5 to 3. The goal of this study is to show the possibility of analyzing the stability of geosystems with image data directly, which has never been attempted. The mechanistic method was just utilized as a benchmark. This study could be a groundbreaking one in that it provides a big-data solution for geotechnical engineering and other civil engineering areas, especially for traditional stability analysis. Future studies with more categories, i.e., safety factors, instead of a binary classification of failed and safe, and more complicated and realistic images will advance the technique toward real-world applications.

The data size and computing power are two major concerns in the application of deep learning to the stability analysis of geosystems. Though big data is an inevitable trend in the long run, great numbers of high-quality labeled data are still not easy to obtain. Therefore, it is desired that deep learning can be satisfactorily implemented with a relatively small amount of data. This concern was studied in this study via benchmark training with different dataset sizes. For the purpose, datasets with different sizes, i.e., 500, 1000, 2000, 5000, 10,000, 20,000, 40,000, 80,000, 120,000, 160,000, and 200,000, were generated and split into training (and validation) datasets using the same ratio, i.e., 5 training to 1 validation. The data were preprocessed in the way as explained in the previous sections. Shown in Fig. 9 are the variations of accuracy and loss with 5000 and 200,000 images. It seems that, for the stability analysis problem targeted in this study, at least 5000 images are needed. The results of the training with 5000 images exhibits overfitting: accuracy from validation does not improve while loss keeps decreasing. Training with a small dataset such as 500 is implementable, however, the training result with an accuracy value of 80% may not be acceptable. Therefore, in the application of deep learning to stability analysis, the data size is still a major factor for ensuring good training results.

To assess the influence of computing power, training with the above-mentioned datasets was carried out with two different GPU resources: 1 NVIDIA K620 (1 physical core, 384 CUDA cores, 400 MHz, 2 GB memory, 29 GB/s bandwidth) and 2 NVIDIA K80 (2 X 13 physical cores, 2 X 2496 CUDA cores, 562 MHz, 2 X 12 GB memory, 2 X 240 GB/s bandwidth). Significant differences were observed by comparing the time and results for the training with the two GPU resources. For time, training with different GPUs consumed significantly different amounts of

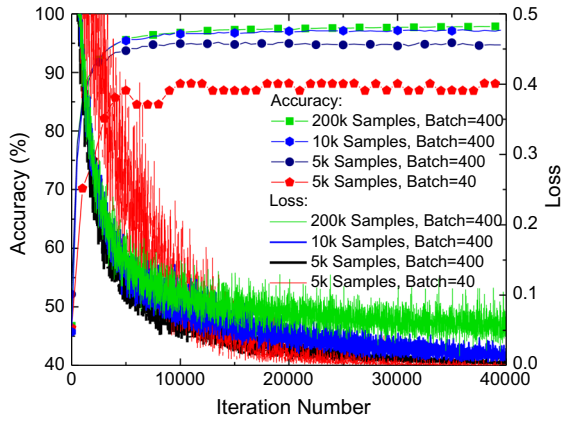

Fig. 9 Training with different data sizes and computing power

time. For example, for 200,000 samples and a batch size of 40, training with K620 used over 630 min while that with 2 K80 used only 64 min. For results, the major differences were observed between training processes with different batch sizes. K620 only allows a small batch size such as 40, while K80 can easily handle a batch size of 400 due to their different memory sizes. In Fig. 9, the training results for 5000 images but different batch sizes, i.e., 40 (K620) and 400 (K80), are illustrated. As can be seen, training with a small batch size exhibits a much lower accuracy and much more oscillations in the accuracy and loss variations. Therefore, higher computing power with better GPUs is another critical factor in gaining better deep learning results in the stability analysis of geosystems.

5 Conclusions

This paper presents a study for exploring the concept of image-data analysis of the stability of geosystems with deep learning. A deep CNN, GeoNet, which consists of multiple convolutional, pooling, ReLU, drop-out, and full connection layers was employed. The deep net functions like a brain. This brain, which was like a newborn baby's, was trained with images that were labeled as "cat (safe)" or "dog (failed)" in datasets of sizes from 500 to 2,000,000 images. With datasets including 10,000 or more images, this training enabled the "baby's brain" to soon gain an ability to classify retaining walls far superseding that of any human's, that is, with a success rate of over 97%. Independent validations also show the values of accuracy, precision, and recall are close to the accuracy obtained in training and validation. Technical details for all steps of deep learning with CNNs were provided. Further investigations into the dataset sizes and computational power yielded quantitative insights into the influence of data and computing resources on the application of deep learning in the stability analysis of geosystems. This study is a ground-breaking one for applying the recent advances in computer vision, including convolutional neural network, to the stability analysis of geosystems and also likely to be among the first of such kind in the mechanical analysis.

Funding The author would like to acknowledge the financial support from the United States National Science Foundation (NSF) via Award 1742656 from the Geotechnical Engineering and Materials Program (now part of CMMI ECI). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abdel-Rahman AH (2008) Predicting compaction of cohesionless soils using ANN. Proc Inst Civil Eng-Ground Improve 161(1):3–8

Ahmad I, El Naggar MH, Khan AN (2007) Artificial neural network application to estimate kinematic soil pile interaction response parameters. Soil Dyn Earthq Eng 27(9):892–905

Alimoradi A, Moradzadeh A, Naderi R, Salehi MZ, Etemadi A (2008) Prediction of geological hazardous zones in front of a tunnel face using TSP-203 and artificial neural networks. Tunn Undergr Space Technol 23(6):711–717

Ardalan H, Eslami A, Nariman-Zadeh N (2009) Piles shaft capacity from CPT and CPTu data by polynomial neural networks and genetic algorithms. Comput Geotech 36(4):616–625

Ashayeri I, Yasrebi S (2009) Free-swell and swelling pressure of unsaturated compacted clays; experiments and neural networks modeling. Geotech Geol Eng 27(1):137

Banimahd M, Yasrobi S, Woodward P (2005) Artificial neural network for stress–strain behavior of sandy soils: Knowledge based verification. Comput Geotech 32(5):377–386

Basheer I (2002) Stress-strain behavior of geomaterials in loading reversal simulated by time-delay neural networks. J Mater Civ Eng 14(3):270–273

Baykasoğlu A, Güllü H, Çanakçı H, Özbakır L (2008) Prediction of compressive and tensile strength of limestone via genetic programming. Expert Syst Appl 35(1):111–123

Benardos A, Kaliampakos D (2004) Modelling TBM performance with artificial neural networks. Tunn Undergr Space Technol 19(6):597–605

Bhattacharya B, Solomatine DP (2006) Machine learning in soil classification. Neural Netw 19(2):186–195

Byeon WY, Lee SR, Kim Y (2006) Application of flat DMT and ANN to Korean soft clay deposits for reliable estimation of undrained shear strength. Int J Offshore Polar Eng 16(1):73–80

Caglar N, Arman H (2007) The applicability of neural networks in the determination of soil profiles. Bull Eng Geol Env 66(3):295–301

Çelik S, Tan Ö (2005) Determination of preconsolidation pressure with artificial neural network. Civ Eng Environ Syst 22(4):217–231

Chen X-W, Lin X (2014) Big data deep learning: challenges and perspectives. IEEE Access 2:514–525

Chen Y-L, Azzam R, Zhang F-B (2006) The displacement computation and construction pre-control of a foundation pit in Shanghai utilizing FEM and intelligent methods. Geotech Geol Eng 24(6):1781–1801

Chen Y-L, Azzam R, Fernandez-Steeger TM, Li L (2009) Studies on construction pre-control of a connection aisle between two neighbouring tunnels in Shanghai by means of 3D FEM, neural networks and fuzzy logic. Geotech Geol Eng 27(1):155

Cho SE (2009) Probabilistic stability analyses of slopes using the ANN-based response surface. Comput Geotech 36(5):787–797

Chua C, Goh AT (2005) Estimating wall deflections in deep excavations using Bayesian neural networks. Tunn Undergr Space Technol 20(4):400–409

Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. Proc., The 25th international conference on Machine learning, ACM, 160–167

Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans Audio Speech Lang Process 20(1):30–42

Das BM (2015) Principles of foundation engineering. Cengage learning, Boston

Das SK, Basudhar PK (2006) Undrained lateral load capacity of piles in clay using artificial neural network. Comput Geotech 33(8):454–459

Das SK, Samui P, Sabat AK (2011) Prediction of field hydraulic conductivity of clay liners using an artificial neural network and support vector machine. Int J Geomech 12(5):606–611

Deng J, Lee C (2001) Displacement back analysis for a steep slope at the three gorges project site. Int J Rock Mech Min Sci 38(2):259–268

Ding X, Zhang Y, Liu T, Duan J (2015) Deep learning for event-driven stock prediction. Proc., IJCAI, 2327–2333

Erzin Y (2007) Artificial neural networks approach for swell pressure versus soil suction behaviour. Can Geotech J 44(10):1215–1223

Erzin Y, Cetin T (2013) The prediction of the critical factor of safety of homogeneous finite slopes using neural networks and multiple regressions. Comput Geosci 51:305–313

Fakoor R, Ladhak F, Nazi A, Huber M (2013) Using deep learning to enhance cancer diagnosis and classification. Proc., the 30th international conference on machine learning, Atlanta, Georgia, USA

Farrokhzad F, Barari A, Choobbasti A, Ibsen LB (2011) Neural network-based model for landslide susceptibility and soil longitudinal profile analyses: two case studies. J Afr Earth Sc 61(5):349–357

Ferentinou M, Sakellariou M (2007) Computational intelligence tools for the prediction of slope performance. Comput Geotech 34(5):362–384

Fu Q, Hashash YM, Jung S, Ghaboussi J (2007) Integration of laboratory testing and constitutive modeling of soils. Comput Geotech 34(5):330–345

Gangopadhyay S, Gautam TR, Gupta AD (1999) Subsurface characterization using artificial neural network and GIS. J Comput Civ Eng 13(3):153–161

Gao W, Feng X, Zheng Y (2004) Identification of a constitutive model for geo-materials using a new intelligent bionics algorithm. Int J Rock Mech Min Sci 41:454–459

Garcıa S, Romo M, Figueroa-Nazuno J (2006) Soil dynamic properties determination: a neurofuzzy system approach. Control Intell Syst 34(1):1–11

Glauner PO (2015) Deep convolutional neural networks for smile recognition. IEEE/ACM Trans Audio Speech Lang Proc 22(10):1533–1545

Goh AT, Kulhawy FH (2003) Neural network approach to model the limit state surface for reliability analysis. Can Geotech J 40(6):1235–1244

Goh A, Kulhawy F (2005) Reliability assessment of serviceability performance of braced retaining walls using a neural network approach. Int J Numer Anal Meth Geomech 29(6):627–642

Goh AT, Wong K, Broms B (1995) Estimation of lateral wall movements in braced excavations using neural networks. Can Geotech J 32(6):1059–1064

Goh AT, Kulhawy FH, Chua C (2005) Bayesian neural network analysis of undrained side resistance of drilled shafts. J Geotech Geoenviron Eng 131(1):84–93

Gokceoglu C, Yesilnacar E, Sonmez H, Kayabasi A (2004) A neuro-fuzzy model for modulus of deformation of jointed rock masses. Comput Geotech 31(5):375–383

Gowda B, Chethan V, Rao Sri Rama T (2012) Optimization of counterfort retaining wall using artificial neural network. Proc., A proceedings of national conference on contemporary civil engineering research and practices, M.I.T, Manipal

Habibagahi G, Bamdad A (2003) A neural network framework for mechanical behavior of unsaturated soils. Can Geotech J 40(3):684–693

Hanna AM, Morcous G, Helmy M (2004) Efficiency of pile groups installed in cohesionless soil using artificial neural networks. Can Geotech J 41(6):1241–1249

Hashash Y, Jung S, Ghaboussi J (2004) Numerical implementation of a neural network based material model in finite element analysis. Int J Numer Meth Eng 59(7):989–1005

Heidari A (2011) Calculation of frequency of retaining wall by back propagation neural network. Asian J Civil Eng 12(3):267–278

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507

Hinton G, Deng L, Yu D, Dahl GE, Mohamed A-R, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process Mag 29(6):82–97

Hopmans JW, Šimunek J, Bristow KL (2002) Indirect estimation of soil thermal properties and water flux using heat pulse probe measurements: geometry and dispersion effects. Water Resour Res 38(1):7–1–7–14

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. Proc., proceedings of the 22nd ACM international conference on multimedia, ACM, 675–678

Kaisler S, Armour F, Espinosa JA, Money W (2013) Big data: Issues and challenges moving forward. Proc., System

sciences (HICSS), 2013 46th Hawaii International Conference on System Sciences, IEEE, 995–1004
Kanungo D, Arora M, Sarkar S, Gupta R (2006) A comparative study of conventional, ANN black box, fuzzy and combined neural and fuzzy weighting procedures for landslide susceptibility zonation in Darjeeling Himalayas. Eng Geol 85(3):347–366
Kasa A, Chik Z, Mohd Raihan T (2011) "Prediction of internal stability for geogrid-reinforced segmental walls." Proc Adv Mater Res 462:1319–1324
Kaya A (2009) Residual and fully softened strength evaluation of soils using artificial neural networks. Geotech Geol Eng 27(2):281–288
Kaya Z (2016) Predicting liquefaction-induced lateral spreading by using neural network and neuro-fuzzy techniques. Int J Geomech 16(4):04015095
Kelnar D (2016) The fourth industrial revolution: a primer on artificial intelligence (AI). Medium.com
Kim Y-S, Kim B-T (2008) Prediction of relative crest settlement of concrete-faced rockfill dams analyzed using an artificial neural network model. Comput Geotech 35(3):313–322
Kitchin R (2014) Big data, new epistemologies and paradigm shifts. Big Data Soc 1(1):2053951714528481
Kostić S, Vasović N, Todorović K, Samčović A (2016) Application of artificial neural networks for slope stability analysis in geotechnical practice. Proc., 13th Symposium on Neural Networks and Applications (NEUREL), IEEE, 1–6
Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Proc., Advances in neural information processing systems, 1097–1105
Kung GT, Hsiao EC, Schuster M, Juang CH (2007) A neural network approach to estimating deflection of diaphragm walls caused by excavation in clays. Comput Geotech 34(5):385–396
Kuo Y, Jaksa M, Lyamin A, Kaggwa W (2009) ANN-based model for predicting the bearing capacity of strip footing on multi-layered cohesive soil. Comput Geotech 36(3):503–516
Kurup PU, Dudani NK (2002) Neural networks for profiling stress history of clays from PCPT data. J Geotech Geoenviron Eng 128(7):569–579
Kurup PU, Griffin EP (2006) Prediction of soil composition from CPT data using general regression neural network. J Comput Civ Eng 20(4):281–289
Lee S, Lee S, Kim Y (2003) An approach to estimate unsaturated shear strength using artificial neural network and hyperbolic formulation. Comput Geotech 30(6):489–503
Lefik M, Schrefler B (2003) Artificial neural network as an incremental non-linear constitutive model for a finite element code. Comput Methods Appl Mech Eng 192(28):3265–3283
Li R, Zhang W, Suk HI, Wang L, Li J, Shen D, Ji S (2014) "Deep learning based imaging data completion for improved brain disease diagnosis." *Proc., International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 305–312
Lohr S (2012) "The age of big data." New York Times, 11
Lu Y (2005) Underground blast induced ground shock and its modelling using artificial neural network. Comput Geotech 32(3):164–178
Ma S, Cao L, Li H (2006) The improved neutral network and its application for valuing rock mass mechanical parameter. J Coal Sci Eng 12(1):21–24
Maji VB, Sitharam T (2008) Prediction of elastic modulus of jointed rock mass using artificial neural networks. Geotech Geol Eng 26(4):443–452
Manning CD (2016) Computational linguistics and deep learning. Comput Ling 41(4):701–707
Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers AH (2011) Big data: the next frontier for innovation, competition, and productivity, Mckinsey Global Institute, Washington DC
Mayoraz F, Vulliet L (2002) Neural networks for slope movement prediction. The Int J Geomech 2(2):153–173
Metz C (2009) Inside Google, data drives 'everything'. The Register, UK
Millar D, Hudson J (1994) Performance monitoring of rock engineering systems using neural networks. Trans Inst Min Metall Sect A Min Ind 103: A3–A16
Moayedi H, Hayati S (2018) Applicability of a CPT-based neural network solution in predicting load-settlement responses of bored pile. Int J Geomech 18(6):06018009
Moyer C (2016) How Google's AlphaGo beat a Go world champion. The Atlantic, March, 28
Najjar YM, Huang C (2007) Simulating the stress–strain behavior of Georgia kaolin via recurrent neuronet approach. Comput Geotech 34(5):346–361
Narendra B, Sivapullaiah P, Suresh S, Omkar S (2006) Prediction of unconfined compressive strength of soft grounds using computational intelligence techniques: a comparative study. Comput Geotech 33(3):196–208
Neaupane K, Achet S (2004a) Some applications of a backpropagation neural network in geo-engineering. Environ Geol 45(4):567–575
Neaupane KM, Achet SH (2004b) Use of backpropagation neural network for landslide monitoring: a case study in the higher Himalaya. Eng Geol 74(3):213–226
Neaupane KM, Adhikari N (2006) Prediction of tunneling-induced ground movement with the multi-layer perceptron. Tunn Undergr Space Technol 21(2):151–159
Nguyen A, Yosinski J, Clune J (2015) Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Proc., The IEEE conference on computer vision and pattern recognition, 427–436
Ohlsson S (2011) Deep learning: How the mind overrides experience. Cambridge University Press
Ozturk TE (2014) Artificial neural networks approach for earthquake deformation determination of geosynthetic reinforced retaining walls. Int J Intell Sys App Eng 2(1):1–9
Padmini D, Ilamparuthi K, Sudheer K (2008) Ultimate bearing capacity prediction of shallow foundations on cohesionless soils using neurofuzzy models. Comput Geotech 35(1):33–46
Park J, Li D, Murphey YL, Kristinsson J, McGee R, Kuang M, Phillips T (2011) Real time vehicle speed prediction using a neural network traffic model. Proc., The 2011

international joint conference on neural networks (IJCNN), IEEE, 2991–2996
Provenzano P, Ferlisi S, Musso A (2004) Interpretation of a model footing response through an adaptive neural fuzzy inference system. Comput Geotech 31(3):251–266
Romo M, Garcıa S (2003) Neurofuzzy mapping of CPT values into soil dynamic properties. Soil Dyn Earthq Eng 23(6):473–482
Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252
Sakellariou M, Ferentinou M (2005) A study of slope stability prediction using neural networks. Geotech Geol Eng 23(4):419
Samui P, Sitharam T (2010) Site characterization model using artificial neural network and kriging. Int J Geomech 10(5):171–180
Shahin MA (2010) Intelligent computing for modeling axial capacity of pile foundations. Can Geotech J 47(2):230–243
Shahin MA, Indraratna B (2006) Modeling the mechanical behavior of railway ballast using artificial neural networks. Can Geotech J 43(11):1144–1152
Shahin M, Jaksa M (2005) Neural network prediction of pullout capacity of marquee ground anchors. Comput Geotech 32(3):153–163
Shahin MA, Jaksa MB (2006) Pullout capacity of small ground anchors by direct cone penetration test methods and neural networks. Can Geotech J 43(6):626–637
Shahin MA, Jaksa MB (2009) "Intelligent computing for predicting axial capacity of drilled shafts." Contemporary Topics in In Situ Testing, Analysis, and Reliability of Foundations, 26–33
Shahin MA, Jaksa MB, Maier HR (2001) Artificial neural network applications in geotechnical engineering. Australian Geomech 36(1):49–62
Shahin MA, Jaksa MB, Maier HR (2002) Artificial neural network based settlement prediction formula for shallow foundations on granular soils. Australian Geomech: J News The Australian Geomech Soci 37(4):45
Shahin M, Maier H, Jaksa M (2003) Settlement prediction of shallow foundations on granular soils using B-spline neurofuzzy models. Comput Geotech 30(8):637–647
Shahin M, Jaksa M, Maier H (2005) Neural network based stochastic design charts for settlement prediction. Can Geotech J 42(1):110–120
Shahin MA, Jaksa MB, Maier HR (2008) State of the art of artificial neural networks in geotechnical engineering. Electron J Geotech Eng 8:1–26
Shang J, Ding W, Rowe R, Josic L (2004) Detecting heavy metal contamination in soil using complex permittivity and artificial neural networks. Can Geotech J 41(6):1054–1067
Shangguan Z, Li S, Luan M (2009) Intelligent forecasting method for slope stability estimation by using probabilistic neural networks. Electr J Geotech Eng, 13, Bund. C: 1–10
Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529(7587):484–489
Singh T, Singh V (2005) An intelligent approach to prediction and control ground vibration in mines. Geotech Geol Eng 23(3):249–262
Singh T, Verma A, Singh V, Sahu A (2005) Slake durability study of shaly rock and its predictions. Environ Geol 47(2):246–253
Singh T, Verma A, Sharma P (2007) A neuro-genetic approach for prediction of time dependent deformational characteristic of rock and its sensitivity analysis. Geotech Geol Eng 25(4):395–407
Sinha SK, Wang MC (2008) Artificial neural network prediction models for soil compaction and permeability. Geotech Geol Eng 26(1):47–64
Sitharam T, Samui P, Anbazhagan P (2008) Spatial variability of rock depth in Bangalore using geostatistical, neural network and support vector machine models. Geotech Geol Eng 26(5):503–517
Wang HB, Sassa K (2006) Rainfall-induced landslide hazard assessment using artificial neural networks. Earth Surf Proc Land 31(2):235–247
Wang CY, Zhang F, Han WD (2014) "A Study on the Application of RBF Neural Network in Slope Stability of Bayan Obo East Mine." Proc., Advanced Materials Research, Trans Tech Publ, 1507–1510
Wu X, Zhu X, Wu G-Q, Ding W (2014) Data mining with big data. IEEE Trans Knowl Data Eng 26(1):97–107
Yang Y, Rosenbaum M (2002) The artificial neural network as a tool for assessing geotechnical properties. Geotech Geol Eng 20(2):149–168
Yildiz E, Ozyazicioglu M, Özkan MY (2010) Lateral pressures on rigid retaining walls: A neural network approach. Gazi Univ J Sci 23(2):201–210
Yoo C, Kim J-M (2007) Tunneling performance prediction using an integrated GIS and neural network. Comput Geotech 34(1):19–30
Yu Y, Zhang B, Yuan H (2007) An intelligent displacement back-analysis method for earth-rockfill dams. Comput Geotech 34(6):423–434
Zeiler MD (2012) ADADELTA: An Adaptive Learning Rate Method. https://arxiv.org/abs/1212.5701