How Artificial Intelligence is Supporting Neuroscience Research: A Discussion About Foundations, Methods and Applications

Rafael T. Gonzalez(✉), Jaime A. Riascos, and Dante A.C. Barone

Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil
rthomazigonzalez@gmail.com, jandresrsalas@gmail.com, barone@inf.ufrgs.br

Abstract. The Artificial Intelligence (AI) research field has presented a considerable growth in the last decades, helping researchers to explore new possibilities into their works. Neuroscience's studies are characterized for recording high dimensional and complex brain data, making the data analysis computationally expensive and time consuming. Neuroscience takes advantage of AI techniques and the increasing processing power in modern computers, which helped improving the understanding of brain behavior. This paper presents some AI techniques, focusing mainly in Deep Learning (DL), as a powerful tool for data analysis. The foundations and basic concepts of some DL models are presented in order to offer a brief understanding to scientists. Likewise, applications of these models on Neuroscience researches are also presented.

Keywords: Neuroscience · Neural Networks · Deep Learning

1 Introduction

The quick grow of Artificial Intelligence (AI) and its approaches have had an important incidence into several research fields, such as economics, engineering, medicine, so on. This has helped researchers to overcome the limitations raised when analyzing great amounts of data, making possible to explore new horizons in their areas. Modern experimental methods in neuroscience areas such as brain imaging generate vast amount of high dimensional and complex data whose analysis represents a challenge [1]. Machine Learning (ML) models, a sub-set of models of AI that iteratively learn from data without being explicitly programmed where to look [2], are becoming ever more important for extracting reliable and meaningful relationships and for making accurate predictions. Over the past decade, several ML models has been applied to analysis of neuropsychological data such as Magnetic Resonance Imaging (fMRI), Near-Infrared Spectroscopy (NIRS), Electroencephalography (EEG), brain imaging, electromyography (EMG) [3–5] as well as in a high-level, AI can be used to modelling several brain functions [6, 7]. The most popular amongst these methods is Support Vector Machine (SVM) [8]. Despite its popularity, SVM has been criticized for not performing well on raw data and requiring the expert use of design techniques to

D.A.C. Barone et al. (Eds.): LAWCN 2017, CCIS 720, pp. 63–77, 2017.
https://doi.org/10.1007/978-3-319-71011-2_6

extract the less redundant and more informative features (a step known as "feature selection") [9]. These features, rather than the original data, are then used for classification. Most common AI models are considered to be shallow, i.e. they do not create multiple layers of adaptive features and so they are of limited interest to neuroscientists trying to understand perceptual pathways.

More recently, however, it was discovered unsupervised methods for creating multiple layers of features, one layer at a time, without requiring any labels. These methods has proven to be significantly better at creating useful high-level features. These alternative family of ML methods known as Deep Learning (DL) [10] is gaining considerable attention in the wider scientific community [9, 11, 12]. DL methods are a type of representation-learning methods, which means that they can automatically identify the optimal representation from the raw data without requiring prior feature selection. This is achieved through the use of a hierarchical structure with different levels of complexity, which involves the application of consecutive nonlinear transformations to the raw data. These transformations result in increasingly higher levels of abstraction [9]. Inspired by how the human brain processes information, the building blocks of DL neural networks − known as "artificial neurons" − are loosely modelled after biological neurons. Learning is achieved through an iterative process of adjustment of the interconnections between the artificial neurons within the network, much like in the human brain [10]. An essential aspect of DL that differentiates it from other machine learning methods is that the features are not manually engineered; instead, they are learned from the data, resulting in a more objective and less bias-prone process. Besides, the ability to achieve higher orders of abstraction and complexity relative to other ML methods such as SVM makes DL better suited for detecting complex, scattered and subtle patterns in the data [13]. Since high-level features can be more robust against noise in the input data, deep architectures may be more suitable for studying this kind of data than conventional ML methods.

Given the increasingly interest in DL within the field of neuroscience, this review aims to give a brief overview of the foundation of some DL methods and some applications that had been carried out with them in Neuroscience area. In the first part of this review, it is presented the underlying concepts of DL, i.e. Neural Networks. This will be followed by a description of Deep Learning foundations. To achieve this, it will be first presented the concept of Unsupervised Learning, including the common used method called Autoencoder. Afterwards, three Supervised Learning models are presented: Stacked Autoencoder, Convolutional Neural Networks and Recurrent Neural Networks. For each of these models it is presented some of its applications in neuroscience. Finally, the paper is concluded with some future directions.

2 Neural Networks

It is estimated that human brain has about 100 billion neurons (nerve cells), connected by an estimated 100 trillion synapses [14]. Neurons share many characteristics with the other cells in the body, but they have unique capabilities for receiving, processing, and transmitting electrochemical signals over the neural pathways that make up the brain's communication system [15]. Given this amazing number of neurons and synapses, it is

considered that the human brain operates as a complex, non-linear and parallel computer [16]. An Artificial Neural Network (ANN) is a computational model biologically inspired in the information-processing structures of the human brain. ANNs consist of processing elements called neurons or perceptron and connections between them. These connections are bounded to coefficients (weights) which represent the "memory" of the system. Given the main role of the connections in Neural Networks, they are called connectionist models. Even though ANNs have similarities to the human brain, they are not meant to model it. They are meant to be used for problem-solving and knowledge-engineering in a "humanlike" way [17]. Neural Networks have been applied to many problems of interest to computer science and engineering, such as, pattern classification (the task of assigning an input pattern to one of many prespecified classes), function approximation (finding an estimated value of an unknown function), forecasting (given a set of labeled training patterns in a time sequence, predict the value of a sample at future time) and optimization (find a solution satisfying a set of constraints such that an objective function is maximized or minimized) [18].

Neural Networks are organized, as shown in Fig. 1, in a layer-wise structure where each layer stores increasingly more abstract representations of the data. The first layer (L_1) is the input layer where the data is entered into the model. In neuroimaging, the data can be represented as a one-dimensional vector with each value corresponding to the intensity of one voxel. The last layer (L_3) is the output layer which, in the context of classification, yields the probability of a given subject belonging to one group or the other. The layers between the input and output layers are called hidden layers, with the number of hidden layers representing the depth of the network. Each layer comprises a set of artificial neurons or "nodes" in which each neuron is fully connected to all neurons in the previous layer. Each connection is associated with a weight value (a_j^i), which reflects the strength of each neuron input, much like a synapse between two biological neurons. The structure of neural networks itself allows the transformation of the input space. The consecutive layers perform a cascade of nonlinear transformations that distort the input space allowing the data to become more easily separable.

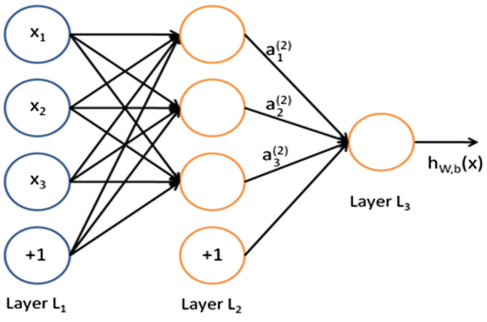

Fig. 1. A neural network organized into 3 layers. L_1 represents the input layer, L_2 is a hidden layer, and L_3 is the output layer [19].

Traditionally, neural networks can learn through a gradient descent-based algorithm. The gradient descent algorithm aims to find the values of the network weights that best minimize the error (difference) between the estimated and true outputs. Since Multi-Layer Perceptron (MLP) can have several layers, in order to adjust all the weights along the hidden layers, it is necessary to propagate this error backward (from the output to the input layer). This propagation procedure is called *Backpropagation* [20], and allows the network to estimate how much the weights from the lower layers need to be changed by the gradient descent algorithm. Initially, when a neural network is trained, the weights are set at random. When the training set is presented to the network, this forward propagates the data through the nonlinear transformation along the layers. The estimated output is then compared to the true output, and the error is propagated from the output towards the input, allowing the gradient descent algorithm to adjust the weights as required. The process continues iteratively until the error has reached its minimum value.

Neural networks, which are commonly used in classification tasks, have been applied into a great number of Brain-Computer Interfaces (BCI) studies. In their reviews, Lotte et al. [21], highlighted how NN have been applied to recognize mental states using brain data such as electroencephalogram signals (EEG). It is shown that NN have being used in different problems such as binary or multiclass classification and synchronous or asynchronous BCI. Another application of EEG data and NN classifier is shown in [22]. Bi et al. use NN to translate features extracted from pre-processed and digitized EEG signals into output commands to a robotic system. However, the accuracy of NN classifiers may not be satisfactory in these applications because these models are sensitive to overtraining especially when dealing with such noisy and non-stationary data as EEG [23]. Therefore, careful architecture selection and regularization is required [24].

3 Deep Learning

When tasked with a problem to solve, humans often decompose the problem into smaller, easier-to-solve subproblems at different levels of representation. Humans are able to inadvertently exploit intuition and describe concepts in hierarchical ways, based on multiple levels of abstraction. For example, an individual seeks to identity an image. Taking the entire image into account, the individual looks specifically at the important features of the image. The individual sees a human form in the image, notices facial hair, body structure, and clothing and determines that the image is a man. That is, the individual has identified the image by breaking it down into smaller features, such as “has beard”, “has broad shoulders”, “is wearing a suit” and then determined a classification for the image. The problem is broken down on many levels. Without much conscious thought, humans look at much smaller features of images, such as lines, curves, and edges to determine the higher-level features. These numerous highly-varying, nonlinear features organized into layers are what constitute a deep network [25].

Deep learning generally refers to learning models which use feature hierarchies with many layers. The hidden layers are composed of units that can be used to describe underlying features of the data. In a common facial recognition task, the input layer

represents the pixels of the image while the output is the corresponding identity of the face, while the hidden layers can represent low-level features, such as edges and shapes, to high-level features, such as "big eyes" or "short hair". Learning the structure of a deep architecture aims to automatically discover these abstractions, from the lowest to highest levels. Favorable learning algorithms would depend on minimal human effort, while allowing the network to discover these latent variables on its own, rather than requiring a predefined set of all possible abstractions. The ability to achieve this task while requiring little human input is particularly important for higher-level abstractions as humans are often unable to explicitly identify potential underlying factors of the raw input [10]. Thus, the power to automatically learn important underlying features made deep architectures so popular.

At the beginning, training these deep networks using Backpropagation hasn't show good results. The problem stemmed from the fact that as a layer eventually learned a task reasonably well, the learned features were not successfully propagated to successive layers in the network. In these models, the information of the error becomes increasingly smaller as it propagates backward from the output to the input layer, to a point where initial layers do not get useful feedback on how to adjust their weights. This issue was called "the vanishing gradient problem" [26]. In 1992, Hochreiter's mentor, Jürgen Schmidhuber, attempted to solve this problem by organizing a multi-level deep hierarchy which could be effectively pre-trained one level at a time via random initialization and unsupervised learning, followed by a supervised Backpropagation pass for fine-tuning [27]. This method allows each level of the hierarchy to learn a compressed representation of the input observation which is in turn fed into the next level as the successive input. However, while deep architectures were promising, the issue remained that many poor results were suggesting that gradient-based training of randomly initialized supervised deep neural networks easily got stuck in local minima or plateaus [25] and that it becomes increasingly difficult to find a good generalization as the architecture got deeper [28]. In 2006, however, Hinton and colleagues revolutionized the DL field by presenting the idea of "greedy layerwise training" algorithm for construction deep architectures [29]. This method consists of two steps: (1) an unsupervised step, where each layer is trained individually and (2) a supervised step, where the previously trained layers are stacked, one additional layer is added to perform the classification (the output layer), and the whole network parameters are fine-tuned. This breakthrough led to the fast-growing interest in Deep Learning and enabled the development of models that yielded state-of-the-art results in tasks such as handwritten digits classification [29].

3.1 Unsupervised Learning

Deep Learning techniques became practically feasible to some extent through the help of Unsupervised Learning (UL). In this context, UL studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. These methods work only with the observed input data, thus there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output [30]. The

advantage of learning features from unlabeled data is that, utilizing the plentiful unlabeled data, potentially better features than hand-crafted features can be learned. Both these advantages reduce the need for expertise of the data.

For example, incoming data such as video or speech streams can be encoded in a form that is more convenient for subsequent goal-directed learning. In particular, codes that describe the original data in a less redundant or more compact way can be fed into learning models, whose search spaces may thus become smaller than those necessary for dealing with the raw data. Many methods of UL have been proposed for regularizing NNs, that is, searching for solution computing but simple, low-complexity, which yields higher generalization performance, without overfitting the training data [31].

In [29], Hinton, et al. it is provided an algorithm to pre-train each layer of a deep network using an unsupervised approach. This greedy layer-wise unsupervised learning algorithm first involves training the lower layer of the model with an unsupervised learning algorithm which yields some initial set of parameters for that first layer of the network. That output from the first layer is a reduced representation of the input. This output then acts as the input for the following layer which is similarly trained, resulting in initial parameters for that layer. Again, the output from this second layer is used as the input for the next layer until the parameters for each layer are initialized. The overall output of the network is delivered as the final activation vector. Following this unsupervised pre-training phase of stacked layers, the entire network can then be fine-tuned in the opposite direction using backpropagation in this supervised learning phase.

3.2 Autoencoder

Autoencoders are a special case of feedforward networks which comprise of two main components. The first component, i.e. the "encoder", learns to generate a latent representation of the input data, whereas the second component, i.e. the "decoder", learns to use these learned latent representations to reconstruct the input data as close as possible to the original [32]. In its shallow structure, as shown in Fig. 2, an autoencoder

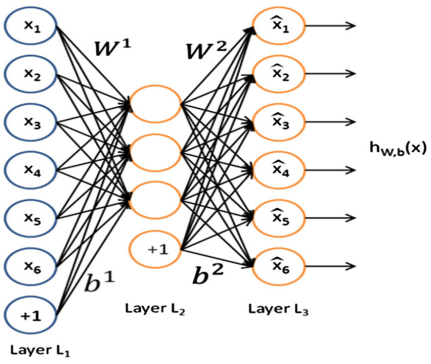

Fig. 2. A single layer autoencoder. Layer L_1 is the original input data, L_2 is the encoded representation of the input data, and L_3 is the reconstruction of the input [33].

is comprised of three layers: an input layer (L_1), one hidden layer (L_2) and an output layer (L_3). Moreover, it has a set of parameters (W, b), where (W^1, b^1) represents the weights and biases of the encoder network, and (W^2, b^2) represents the weights and biases of the decoder network.

In previous neural networks, labelled data were required to act as training examples essential to the backpropagation fine-tuning pass as those labels were used to readjust the connection weights. However, since an autoencoder does not make use of labels, its training is an unsupervised learning process. An autoencoder neural network performs backpropagation by setting the target output values equal to the input values, and thus it is trained to minimize the discrepancy between the data and its reconstruction. In other words, it is trying to learn an approximation to the identity function, so as to output $\hat{x}$ that is similar to x [33]. While this may seem like a trivial learning task, placing constraints on the network can reveal interesting structure of the data. An example of a constraint is a limitation to the number of hidden units in the hidden layer, thus forcing the network to learn a compressed representation of the input. This method allows for the discovery of internal representations of the data that rely on fewer intermediate features. Another constraint that can be applied to the network could be the sparsity of hidden units that are activated. Sparsity is a useful constraint when the number of hidden units is large (even larger than the number of input values) that can allow for the discovery of interesting structure of the data [32]. A sparse autoencoder has very few neurons that are active. A neuron in an artificial neural network is informally considered "active" if its output value is close to 1, while it is considered "inactive" if its output value is close to 0. The concept of creating a sparse autoencoder involves constraining the most of the neurons to be inactive [32]. As a result, even with many hidden units, the data is constrained, forcing the network to learn the important features of the data in order to reconstruct it.

Hinton et al. defined an autoencoder as a nonlinear generalization of Principal Components Analysis (PCA), which is restricted to linear mapping [34]. If no non-linear function is used in the encoder network of the AE and the number of neurons in the hidden layer is of smaller dimension than that of the input then PCA and AE can yield similar results. On the other hand, in the case of the neurons in the hidden layer is greater than the input size, the AE is transforming the input data from one feature space to another wherein the data in the new feature space disentangles factors of variation. PCA has the advantage that it can work with very little data, while Autoencoders can overfit if not enough data is available.

Autoencoders have being widely applied in studies that use a range of neuroimaging modalities including structural Magnetic Resonance Imaging (sMRI), resting-state functional MRI (rsfMRI) and positron emission tomography (PET) [35, 36]. In [37], Payan and Montana used Sparse Autoencoders and Convolutional Neural Networks to predict the Alzheimer's Disease (AD) status of a patient based on 3D MRI scan of the brain. Their proposed method outperforms several other classifiers reported in the literature and produce state-of-art results. Suk et al. [38] developed an approach which classifies people with Mild Cognitive Impairment (MCI) and healthy controls using a deep autoencoder to extract hierarchical nonlinear relations among brain regions, whilst modelling the inherent functional dynamics of rsfMRI data. This was also one of the few

studies in which the same DL model was tested against and surpassed other competing models in two independent datasets, thus providing evidence of replicability, a crucial feature for diagnostic tools.

3.3 Stacked Autoencoder

Based on the fact that Autoencoders are automatic features extractors, they can also be stacked to create a deep structure to increase the level of abstraction of learned features. Thus, a Stacked Autoencoder (SAE) is a neural network consisting of multiple layers of Autoencoders [10]. In this case, the network is pre-trained, i.e. each layer is treated as a shallow autoencoder, generating latent representations of the input data. These latent representations are then used as input for the subsequent layers before the full network is fine-tuned using standard supervised learning algorithm [25]. With this deep architecture, learning-feature hierarchies are formed by using lower-level learned features to compose higher levels of the hierarchy. The first layer of a SAE tends to learn first-order features in the raw input (such as edges in an image), the second layer tends to learn second-order features corresponding to patterns in the appearance of first-order features (for example, contour or corner detectors) and, following this logic, higher layers to learn even higher-order features [25].

SAE have been successfully used in studies of brain psychosis diagnostics. In [39], it is used a SAE to extract latent features from neuroimaging data (sMRI, PET and CSF), which were then used to predict clinical data and class labels. The resulting learned features were combined with original low-level features to build a robust model for AD/MCI classification that achieved high diagnostic accuracy. Another research area of high clinical interest is prediction of response to treatment. In several psychiatric and neurological disorders, a better understanding of why some patients benefit from a certain treatment whereas others do not, could help clinicians make more-effective treatment decisions and improve long-term clinical outcomes [40]. In [41], it presented an algorithm that distinguished between patients with temporal lobe epilepsy (TLE) who did and did not benefit from surgical treatment. The proposed uses a SAE to extract meaningful features from diffusion-weighted images (DWI) while a SVM is chosen as the classifier.

3.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are very similar to ordinary Neural Networks previously explained. They are a special type of feedforward neural networks that were biologically-inspired by the visual cortex [42]. The visual cortex contains a complex arrangement of cells that are sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field. These cells act as local filters over the input space and are well-suited to exploit the strong spatially local correlation present in natural images [43].

In addition to the input and output layers, CNN can mainly comprise of three types of layers: a convolutional layer, a pooling layer, and a fully-connected layer [44]. The first one acts as feature identifiers, i.e. they are filters that extract characteristics from input data (such as edges and curves). As the number of convolutional layers increase,

more complex features can be represented (such as hands or ears). A common operator used together with convolution is pooling, which combines nearby values in input or feature space through a sample-based discretization process. The objective is to down-sample an input representation, reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. This process helps over-fitting by providing an abstracted form of the representation. As well, it reduces the computational cost by reducing the number of parameters to learn. Finally, the fully-connected layers are similar to the hidden layers from the conventional MLP where the neurons are connected to all neurons from the previous layer. The CNN arranges its neurons in three dimensions (width, height, depth), these values are proportional to the size and channels of the input. Every layer in the CNN transforms the 3D input volume to a 3D output volume of neuron activations. Figure 3 exposes a normal Neural Network and a CNN.

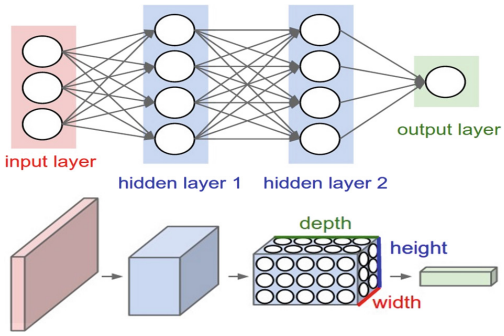

Fig. 3. Top: common Neural Network. Bottom: CNN [44]

The CNN's architecture is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal) and to encode certain properties into the architecture. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully-connected layers networks with the same number of hidden units. Many neurally-inspired models can be found in the literature, such as the NeoCognitron [45], HMAX [46] and LeNet-5 [43]. Many present competition-winning models are based on CNNs. A BP-trained CNN set a new MNIST (handwritten digit recognition dataset) record of 0.39% [47], using training pattern deformations but no unsupervised pre-training.

As was exposed above, CNN was mainly designed for images as input; therefore, the processing, analysis and classification of neuroimaging may turn out to be among the most important applications of Deep Learning, because it could not only save lots of money, but also make expert diagnostics more accessible. For example, an important application of CNNs on cancer diagnosis was presented in [48], the authors successfully use deep max-pooling CNNs to detect mitosis in breast histology images. Likewise, several authors have worked with CNN using either MRI and functional MRI (fMRI) as data input. Sarraf and Tofighi [49] used fMRI data for classification of

Alzheimer's Disease (AD). This work suggests that the shift and scale invariant features extracted by CNN and a deep learning classification are the most powerful method to distinguish clinical data from healthy data in fMRI. Another study performed by Liu et al. [50] use fusing multi-modal neuroimaging features to aid the diagnosis of AD. This framework presented the potential to require less labelled data and therefore a performance gain was achieved in both binary classification and multi-class classification of AD. Moreover, van der Burgh et al. [51] used clinical characteristics in combination with MRI data to predict survival of Amyotrophic lateral sclerosis (ALS) patients using deep learning. This approach reached an accuracy of 84.4%.

3.5 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are obtained from the feedforward network by connecting the neurons' output to their inputs [52]. They are called recurrent due that perform the same task for every element of a sequence; thus, the output of the network is depended on the previous computations. The short-term time-dependency is modelled by the hidden-to-hidden connections without using any time delay-taps. They are usually trained iteratively via a procedure known as backpropagation-through-time (BPTT). RNNs can be seen as very deep networks with shared parameters at each layer when unfolded in time. This results in the problem of vanishing gradients [53]. Long Short-Term Memory (LSTM) [54] was proposed to resolve this problem for Recurrent Neural Networks. A LSTM memory cell is composed of four main elements: an input gate, a neuron with a self-recurrent connection, a forget gate and an output gate. The input gate controls the impact of the input value on the state of the memory cell and the output gate controls the impact of the state of the memory cell on the output. The self-recurrent connection controls the evolution of the state of the memory cell and the forget gate determines how much of prior memory value should be passed into the next time step. Depending on the states of these gates, LSTM can represent long-term or short-term dependency of sequential data.

Recently, LSTM RNNs won several international competitions and set numerous benchmark records. A stack of bidirectional LSTM RNNs broke a famous TIMIT speech (phoneme) recognition record [55]. For optical character recognition (OCR), LSTM RNNs outperformed commercial recognizers of historical data [56]. LSTM-based systems also set benchmark records in language identification [57], medium-vocabulary speech recognition [58], and text-to-speech synthesis [59].

RNN has being widely used for modelling various neurobiological phenomena, considering anatomical, electrophysiological and computational constraints. The computational power of RNNs comes from the fact that its neuron's activity is affected not only by the current stimulus (input) to the network but also by the current state of the network, it means that into the network will keep on traces of past inputs [61]. Thus, the RNNs are ideally suited for computations that unfold over time such as holding items in working memory or accumulating evidence for decision-making. Nevertheless, it is this feature that difficult to train RNNs (as was explained above). For example, Barak [60] presents RNNs as a versatile tool to explain such neural phenomena including several constraints. He exposes how combining trained RNNs with

reverse engineering can represent an alternative framework for neuroscience modelling, potentially serving as a powerful hypothesis generation tool. Moreover, Rajan et al. [61] show RNN models of neural sequences of memory based on decision-making tasks generated by minimally structured networks. It suggests that neural sequences activation may provide a dynamic mechanism for short-term memory, which comes from largely unstructured network architectures. In the same way, Güçlü and van Gerven [62] show how RNNs are a well-suited tool for modelling the dynamics of human brain activity. In their approach, they investigated how the internal memories of RNNs can be used in the prediction of feature-evoked response sequence, which are commonly measured by fMRI. Likewise, Susillo et al. [63] use RNNs to generate muscle activity signals (electromyography, EMG) to explain how the neural responses in motor cortex. They started with the hypotheses that motor cortex reflects a dynamic, which is used for generating temporal commands. Thus, the RNNs are used to transform simple inputs into temporal and spatial complex patterns of muscle activity. Finally, Viera et al. [64] made a deep review and discussion of how deep learning has been used to investigate the neuroimaging correlates of psychiatric and neurological disorders, explaining and comparing different methods and applications.

4 Conclusion

This paper offers a discussion about the significant support that AI offers to Neuroscience research. The capacity of DL models to learn complex and abstract representations through nonlinear transformations provides preliminary evidences supporting its potential role in the future development of Neuroscience. Some learning methods and its applications were presented here, mainly focusing on Neural Network and Deep Learning. The models were explained since their foundations until their current state. In summary, Neural Networks have being used in classification, regression and clustering problems, playing an important role in BCI studies. DL models offer alternative and powerful tools for modelling neural activity (RNN), processing neuroimaging (CNN and AE), predicting clinical data and class labels (SAE) and performing dimensionality reduction (AE).

Nevertheless, several improvements will be required before the full potential of DL in Neuroscience can be achieved. Given the complexity of DL models, we need to favor studies that use large data samples. A possible way of achieving this is through multi-center collaborations, in which data is collected using similar criteria and scanning protocols across sites. Moreover, the integration of CNN and RNN is likely to lead to significant advances in DL in the next few years [65]. In neuroimaging, for example, this integration could be particularly useful for analyzing fMRI data, as it would allow the detection of intricate spatial patterns while simultaneously modelling the temporal component of some signals. Finally, it is so important to highlight the necessity of a strong collaboration between Neuroscience and DL. Neuroscientists are interested in DL models as an important tool for understanding the brain behavior and, likewise, computer scientists are taking into account brain theories when creating new models and techniques.

References

1. Helmstaedter, M.: The mutual inspirations of machine learning and neuroscience. Neuron **86** (1), 25–28 (2015)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2007). ISBN-10: 0387310738, ISBN-13: 978-0387310732
3. Patel, M.J., Khalaf, A., Aizenstein, H.J.: Studying depression using imaging and machine learning methods. NeuroImage: Clin. **10**, 115–123 (2016)
4. Khachab, M., Mokbel, C., Kaakour, S., Saliba, N., Chollet, G.: Brain imaging and machine learning for brain-computer interface. In: Biomedical Imaging, InTech (2010)
5. Lemm, S., Blankertz, B., Dickhaus, T., Müller, K.-R.: Introduction to machine learning for brain imaging. NeuroImage **56**(2), 387–399 (2011)
6. Yamins, D.L.K., DiCarlo, J.J.: Using goal-driven deep learning models to understand sensory cortex. Nat. Neurosci. **19**, 356–365 (2016)
7. Kasabov, N.K.: NeuCube: a spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. Neural Netw. **52**, 62–76 (2014)
8. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
9. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
10. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**, 1–127 (2009)
11. Arbabshirani, M.R., Plis, S., Sui, J., Calhoun, V.D.: Single subject prediction of brain disorders in neuroimaging: promises and pitfalls. Neuroimage **145**, 137–165 (2016)
12. Calhoun, V.D., Sui, J.: Multimodal fusion of brain imaging data: a key to finding the missing link(s) in complex mental illness. Biol. Psychiatry: Cogn. Neurosci. Neuroimaging **1**, 230–244 (2016)
13. Plis, S.M., Hjelm, D.R., Salakhutdinov, R., Allen, E.A., Bockholt, H.J., Long, J.D., Johnson, H.J., Paulsen, J.S., Turner, J., Calhoun, V.D.: Deep learning for neuroimaging: a validation study. Front. Neurosci. **8**, 1–11 (2014)
14. Herculano-Houzel, S.: The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. In: Proceedings of the National Academy of Sciences, USA, vol. 109 (Supp 1), pp. 10661–10668 (2012)
15. Herculano-Houzel, S.: The human brain in numbers: a linearly scaled-up primate brain. Front. Hum. Neurosci. **3**, 31 (2009). https://doi.org/10.3389/neuro.09.031.2009
16. Nygren, K.: Stock prediction - a neural network approach. Master thesis, Royal Institute of Technology, KTH (April 2004)
17. Kasabov, N.K.: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. MIT Press, Cambridge (1996)
18. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: a tutorial. Computer **29**(3), 31–44 (1996)
19. Ng, A., Ngiam, J., Foo, C., Mai, Y., Suen, C.: UFLDL Tutorial (2013) Retrieved from Stanford Deep Learning: http://uflDl.stanford.edu/wiki/index.php/Neural_Networks
20. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
21. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A review of classification algorithms for EEG-based brain–computer interfaces. J. Neural Eng. **4**(2), R1–R13 (2007)
22. Bi, L., Fan, X.A., Liu, Y.: EEG-based brain-controlled mobile robots: a survey. IEEE Trans. Hum. Mach. Syst. **43**(2), 161–176 (2013)

23. Balakrishnan, D., Puthusserypady, S.: Multilayer perceptrons for the classification of brain computer interface data. In: Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference (2005)
24. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000)
25. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, p. 153 (2007)
26. Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis. Institut f. Informatik, Technische Univ. Munich (1991)
27. Schmidhuber, J.: Learning complex, extended sequences using the principle of history compression. Neural Comput. **4**(2), 234–242 (1992)
28. Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P.: Exploring strategies for training deep neural networks. J. Mach. Learn. Res. **10**, 1–40 (2009)
29. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
30. Barlow, H.B.: Unsupervised learning. Neural Comput. **1**, 295–311 (1989)
31. Baum, E.B., Haussler, D.: What size net gives valid generalization? Neural Comput. **1**(1), 151–160 (1989)
32. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
33. Ng, A., Ngiam, J., Foo, C., Mai, Y., Suen, C.: UFLDL Tutorial (2013). Retrieved from Stanford Deep Learning: http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity
34. Calhoun, V.D., Silva, R.F., Adali, T., Rachakonda, S.: Comparison of PCA approaches for very large group ICA. Neuroimage **118**, 662–666 (2015). https://doi.org/10.1016/j.neuroimage.2015.05.047
35. Liu, S., Liu, S., Cai, W., Che, H., Pujol, S., Kikinis, R., Feng, D., Fulham, M.J.: Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease IEEE Trans. Biomed. Eng. **62**, 1132–1140 (2015)
36. Han, X., Zhong, Y., He, L., Philip, S.Y., Zhang, L.: The unsupervised hierarchical convolutional sparse auto-encoder for neuroimaging data classification. In: Guo, Y., Friston, K., Aldo, F., Hill, S., Peng, H. (eds.) BIH 2015. LNCS, vol. 9250, pp. 156–166. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23344-4_16
37. Payan, A., Montana, G.: Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. arXiv preprint arXiv:1502.02506 (2015)
38. Suk, H.I., Wee, C.Y., Lee, S.W., Shen, D.: State-space model with deep learning for functional dynamics estimation in resting-state fMRI. Neuroimage **129**, 292–307 (2016)
39. Suk, H.I., Shen, D.: Deep learning-based feature representation for AD/MCI classification. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 583–590. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40763-5_72
40. Mechelli, A., Prata, D., Kefford, C., Kapur, S.: Predicting clinical response in people at ultra-high risk of psychosis: a systematic and quantitative review. Drug Discov. Today **20**, 924–927 (2015)
41. Munsell, B.C., Wee, C.Y., Keller, S.S., Weber, B., Elger, C., da Silva, L.A.T., Nesland, T., Styner, M., Shen, D., Bonilha, L.: Evaluation of machine learning algorithms for treatment outcome prediction in patients with epilepsy based on structural connectome data. Neuroimage **118**, 219–230 (2015)
42. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324 (1988)

43. Hubel, D., Wiesel, T.: Receptive fields and functional architecture of monkey striate cortex. J. Physiol. (London) **195**, 215–243 (1968)
44. CS231n Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/convolutional-networks/. Accessed 14 Sept 2017
45. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**, 193–202 (1980)
46. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M.: Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 411–426 (2007)
47. Ranzato, M., Poultney, C., Chopra, S., LeCun, Y.: Efficient learning of sparse representations with an energy-based model. In: Platt, J. et al. (eds.), Advances in neural information processing systems (NIPS 2006). MIT Press (2006)
48. Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 411–418. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40763-5_51
49. Sarraf, S., Tofighi, G.: Classification of Alzheimer's Disease using fMRI Data and Deep Learning Convolutional Neural Networks. arXiv preprint arXiv:1603.08631 (2016)
50. Liu, S., Liu, S., Cai, W., Che, H., Pujol, S., Kikinis, R., Adni, M.J.: Multi-modal neuroimaging feature learning for multi-class diagnosis of Alzheimer's disease. IEEE Trans. Biomed. Eng. **62**(4), 1132–1140 (2015). https://doi.org/10.1109/TBME.2014.2372011
51. van der Burgh, H.K., Schmidt, R., Westeneng, H.J., de Reus, M.A., van den Berg, L.H., van den Heuvel, M.P.: Deep learning predictions of survival based on MRI in amyotrophic lateral sclerosis. NeuroImage: Clinic. **13**, 361–369 (2017). ISSN 2213-1582. http://dx.doi.org/10.1016/j.nicl.2016.10.008
52. Hüsken, M., Stagge, P.: Recurrent neural networks for time series classification. Neurocomputing **50**, 223–235 (2003)
53. Pascanu, R., Mikolov, T., Bengio, Y.: Understanding the exploding gradient problem. Computing Research Repository (CoRR) abs/1211.5063 (2012)
54. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
55. Graves, A., Mohamed, A.-R., Hinton, G.E.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649. IEEE (2013)
56. Breuel, T.M., Ul-Hasan, A., Al-Azawi, M.A., Shafait, F.: High-performance OCR for printed English and Fraktur using LSTM networks. In: 12th International Conference on Document Analysis and Recognition, pp. 683–687. IEEE (2013)
57. Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., Moreno, P.J.: Automatic language identification using long short-term memory recurrent neural networks. In: Proceedings of Interspeech (2014)
58. Geiger, J.T., Zhang, Z., Weninger, F., Schuller, B., Rigoll, G.: Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling. In: Proceedings of Interspeech (2014)
59. Fan, Y., Qian, Y., Xie, F., Soong, F.K.: TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Proceedings of Interspeech (2014)
60. Barak, O.: Recurrent neural networks as versatile tool of neuroscience research. Curr. Opin. Neurobiol. **46**, 1–6 (2017)
61. Rajan, K., Harvey, C.D., Tank, D.W.: Recurrent network models of sequence generation and memory. Neuron **90**(1), 128–142 (2016). https://doi.org/10.1016/j.neuron.2016.02.009

62. Güçlü, U., van Gerven, M.A.J.: Modeling the dynamics of human brain activity with recurrent neural networks. Front. Comput. Neurosci. **11**, 7 (2017). https://doi.org/10.3389/fncom.2017.00007
63. Sussillo, D., Churchland, M.M., Kaufman, M.T., Shenoy, K.V.: A neural network that finds a naturalistic solution for the production of muscle activity. Nat. Neurosci. **18**, 1025–1033 (2015)
64. Vieira, S., Pinaya, W.H.L., Mechelli, A.: Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: methods and applications. Neurosci. Biobehav. Rev. **74**, 58–75 (2017). https://doi.org/10.1016/j.neubiorev.2017.01.002
65. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)