Artificial Intelligence Applications in Stroke

10

Arlindo L. Oliveira

10.1 Introduction

Artificial intelligence (AI) techniques, namely those that are based in machine learning (ML), can be used effectively in order to improve the diagnostic and treatment of stroke. Machine learning techniques can leverage the significant amounts of data obtained by clinicians in order to provide insights on the seriousness of stroke incidents, improve the quality of the diagnosis, and predict patient outcomes. Machine learning techniques make it possible to use data to create models that can be used to support decisions in the different phases of stroke diagnostic and treatment. The application of machine learning is widely viewed as a key research priority, as academic institutions, companies, and clinicians strive to improve the treatment of stroke.

Until recently, the data that could be used to support decisions in this area was mainly the clinical data obtained during admission and stay of patients at the hospital. This data can be very informative and, duly analyzed, can be used to effectively make diagnosis and predictions about the future evolution of stroke patients. However, in the last few years, the emergence of deep learning techniques that can effectively process image data enabled machine learning-based systems to use other sources of information, in particular imaging data obtained using computed tomography (CT), with or without contrast, and magnetic resonance imaging (MRI). Furthermore, the increased availability of patient genetic data creates the possibility of integrating clinical, imaging, and genetic data in order to improve the prediction ability of the models. The number of applications of artificial intelligence and machine learning in the area of stroke is, therefore, increasing rapidly, from early detection of ischemia on CT images to the determination of key metrics on

A. L. Oliveira (✉)
INESC-ID/Instituto Superior Técnico, Lisbon, Portugal
e-mail: arlindo.oliveira@tecnico.ulisboa.pt

A. C. Fonseca, J. M. Ferro (eds.), *Precision Medicine in Stroke*,
https://doi.org/10.1007/978-3-030-70761-3_10

perfusion or blood flow in specific areas of the brain. Improving decision support systems with the use of machine learning techniques will be an important objective in the years to come.

10.2 Using Machine Learning to Infer Predictive Models

The key technology behind the most recent advances in artificial intelligence is machine learning, a designation that stands for a vast set of techniques that aim at inferring general rules from specific examples.

In its simpler form, inductive learning infers a general rule from a set of labeled instances, called the training set. Each instance is defined by a set of attributes (or variables) and their respective values. To provide a concrete example, one may build a table with patient information at the time of admission, which includes the age and gender of the patient, the existence of past stroke events, the delay between stroke onset and hospital arrival, and the NIH Stroke Score (NIHSS). In this example, one wishes to infer, from the available data, the expected final outcome, 3 months after the stroke incident, in the modified Rankin Scale (mRS). The relevant data could be organized into something that would resemble Table 10.1, where the specific values for each attribute, for each patient, are listed in the corresponding column, one row per patient.

The organization of data in tabular form is familiar to many readers and is extensively used to keep track of many types of patient information. It turns out that the existence of data in tabular format makes possible the direct application of machine learning techniques to the problem of inferring the outcome (the target label, listed in the last column of Table 10.1) from the independent patient attributes, listed in columns 2–6 of that table. The patient ID is simply an identifier of the patient and has no predictive value. In many cases, it is used to preserve the anonymity of the patients, by making sure that the correspondence between the patient ID and the real identification of the patients is kept reserved.

Over many decades machine learning researchers have developed hundreds, if not thousands, of methods and algorithms that can be used to infer the target label from the independent attributes, a problem known as supervised learning. If the label is discrete, as is the case in this table, the problem is called a classification problem. If the label is continuous (typically a real number) the problem is called a

Table 10.1 Example of table with instances of stroke incidents

Patient ID	Age	Gender	Past events	Delay (h)	NIHSS	mRS
001	57	Male	No	3.5	12	2
002	65	Male	Yes	4.0	27	5
003	71	Female	No	1.5	14	3
004	58	Female	No	3.5	20	2
005	66	Male	Yes	6.0	27	5
006	68	Male	No	1.0	25	4

regression problem. In this chapter, we focus mainly on classification problems, but many of the techniques developed for classification problems can also be used in regression problems, although sometimes they require some modifications.

This problem is known as supervised learning because a supervisor (or teacher) provides the value of the labels in the training set, the dataset that can be used to train the machine learning system. If labels are not available, the problem is known as unsupervised learning. In that case, the objective is usually to aggregate the instances into classes that share a significant similarity, a problem known as clustering, but other objectives are also possible, such as the identification of outliers or the cleaning of data. In this chapter, we focus on the problem of supervised learning, since the availability of labels for a subset of the instances (the training set) usually enables the algorithms to perform more interesting predictions.

Supervised machine learning algorithms infer a model from the data available in the training set. Such a model can take many different forms. It can be a mathematical formula, a set of rules, the parameters of a network, and several other forms. In practice, the model enables the user to apply a fixed set of computations to the attributes of an instance in order to obtain the value of the target label. One significant issue in machine learning is overfitting. Overfitting occurs when the model performs well in the available data (training set) but does not perform equally well on future data (test set). Since the objective of creating the model is to use it to predict future instances, avoiding overfitting is essential when machine learning techniques are used.

Machine learning algorithms can be classified into one of the four categories: symbolic, statistical, similarity based, and connectionist. These categories are not mutually exclusive, and one method may share features of more than one category. It is also possible to partition the methods in a different and more or less numerous set of categories. Pedro Domingos, for instance, considered an additional category, genetically inspired machine learning algorithms, obtaining a five-category taxonomy [1]. In the sequence, I will provide a brief introduction to each of these families of machine learning algorithms.

10.2.1 Symbolic Methods

Symbolic methods derive symbolic rules that can be used to derive the target labels of unseen instances. One may imagine, for instance, a symbolic rule of the type "*if age > 60 and NIHSS > 24 then mRS = 5*," which would enable us to predict that persons older than 60 that obtain an NIHSS larger than 24 will be expected to have a final modified Rankin Scale of 5. Such a rule is just an example, of course. In a realistic case, a set of such rules would enable clinicians to derive the predicted Rankin Scale for any incoming patient. Symbolic methods have a significant advantage: the classification they perform can be, in many cases, intelligible to a human, and the inferred rules can be analyzed and checked for consistency.

Although there are many methods that use a symbolic approach, one of them is particularly popular and effective: decision trees. A decision tree inference algorithm, such as ID3 [2], C4.5 [3], or CART [4], processes the tabular data and derives a decision tree that closely matches the target labels in the training set. Decision tree inference algorithms aim at deriving compact trees, which are more likely to have high predictive values for unseen instances. Such trees not only match closely the target labels in the training set, but can also be used to effectively predict these labels in new instances, never seen before. For instance, the decision tree in Fig. 10.1 correctly labels all the instances in Table 10.1, and can be used to infer the outcomes of patients who are not in that table.

By inspection, it is fairly easy to understand the set of rules that correspond to the tree in Fig. 10.1. For instance, the tree predicts that patients with an NIHSS larger than 24 who have been admitted to the hospital more than 3 h after the stroke incident are expected to have a final mRS of 5. This tree should not be viewed as an accurate classifier for this particular problem, as it is being used solely to illustrate the approach. In practice, trees are inferred from much larger datasets and are usually significantly more complex [5]. Other symbolic models, such as decision lists [6], have also been proposed since they exhibit, over other approaches, the advantage of being more easily understandable. One popular and effective approach, random forests, is based on the creation of a population of decision trees that "vote" on the predicted outcome. The most popular outcome (the one that gets more votes from the trees) is selected as the predicted outcome. In reality, decision forests can also be viewed as a statistical method, since it is sampling from a distribution of trees that models the phenomenon under study.

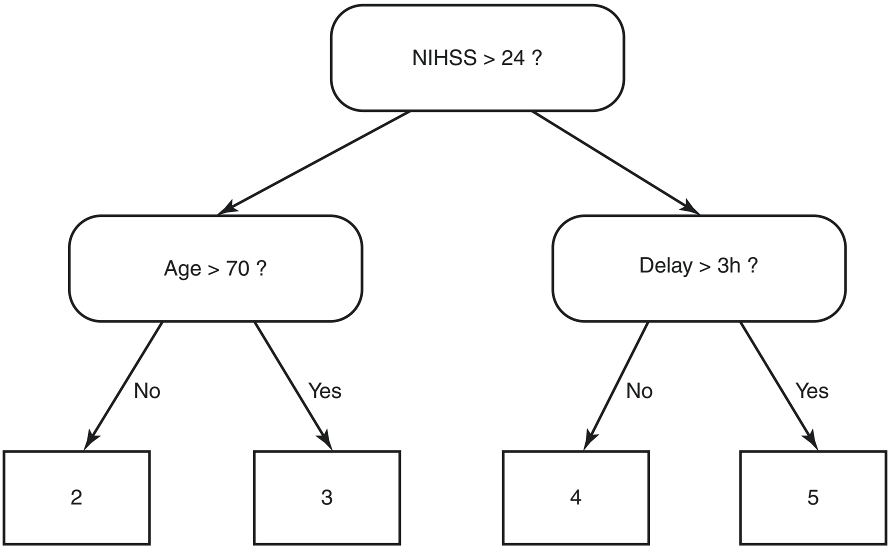

Fig. 10.1 Decision tree for the instances in Table 10.1

10.2.2 Statistical Methods

Statistical methods aim at inferring relevant probabilistic relations between the independent attributes and the target label, in order to predict the most likely value of the target label from the values of the attributes. More formally, statistical methods estimate the joint probability distributions (either implicitly or explicitly) of the input/output attributes and use this joint distribution to infer the most likely value of the target output. One of the most relevant and well-known statistical methods is regression, a simple technique that is, however, very relevant in many practical settings. In its simplest form, single linear regression finds a linear relation between an independent continuous variable and a dependent continuous variable, by looking for a set of parameters that minimize the sum of the squares of the differences between the model and the observed values of the independent variable. Multiple linear regression finds a relation between a set of continuous variables and a single independent variable. In some cases, one expects a nonlinear relationship between the independent variables and the dependent variable. In these cases, polynomial regression can be used, a technique that derives the coefficients of a polynomial relating the input and output variables.

Single or multiple, linear or polynomial, regression is not applicable in classification problems, where the independent variable is discrete, as is the example in Table 10.1. In these cases, logistic regression should be used. Logistic regression is directly applicable when the independent variable, the target class label, is binary. As in linear regression, the coefficients derived by the mathematical optimization algorithm provide an indication of the relevance of a given input variable in the determination of the class label. Multinomial logistic regression is used when there are more than two possible values for the target label.

Regularization can be used to improve the performance of regression methods. When there are many input variables, standard regression (linear or logistic) tends to weight many, or even all, input variables, leading in many cases to the phenomenon of overfitting, introduced above. Overfitting can be controlled in regression if one forces a significant fraction of the coefficients to become zero, forcing the regression function to depend only on a small subset of the input variables. The mathematical formulation of regularized regression adds a term to the sum of the squares of the errors that penalizes the existence of many coefficients different from zero. Depending on the form of the penalty, this formulation may lead to ridge regression, Lasso regression, or other possibilities. Regression-based methods have been extensively used in the prediction of stroke outcomes [7, 8], with good results.

Another important statistical method is based on the direct application of Bayes' theorem, which relates the probability of a given value of the target label with the probability of observing a specific value in the independent attributes, for each value of the target. Although a direct application of Bayes' theorem is usually not doable, since it involves the estimation of the joint probability distribution of the input variables for each possible class, a simplified method, called naive Bayes, can be applied and can be very effective, in practice [9].

10.2.3 Similarity-Based Methods

A third category of methods used to infer a classification method from a set of labeled instances is similarity-based methods. The fundamental idea behind similarity-based methods is to classify a new instance in the same class as the most similar instance or instances in the training set. For example, given the instances in Table 10.1, if one is asked to determine the most likely outcome for a new male patient, with 57 years and no prior incidents, admitted 3 hours after the stroke incident and classified with an NIHSS of 12, it would be reasonable to estimate that the mRS outcome would be 2, since this case is very similar to the patient that corresponds to the first instance in the table. This classification method is called the nearest-neighbor algorithm, and is reasonably effective in some specific conditions. When the algorithm considers a fixed number of nearest neighbors (*k*) and decides the class by looking at the most common class among these neighbors, it is called the *k*-nearest neighbor algorithm.

In general, the application of similarity-based methods requires the definition of a metric of similarity (a distance) between two instances that takes into account the nature of the different input attributes. For instance, in our example, a difference of 1 year in age is probably less significant than a difference of 1 h in the admission delay. Furthermore, both categorical and continuous attributes have to contribute to this metric in such a way that the distances between instances make sense. Defining such a similarity metric can be challenging, but there are algorithms that perform this task automatically, adjusting the scales of the different dimensions.

Although the basic idea behind similarity-based methods is very simple, very sophisticated approaches based on this idea have been developed, and some of them involve very sophisticated mathematical machinery. One method that can be classified in the family of similarity-based methods (although it can also be viewed as a modified form of regression) is support vector (SVM) classification. Support vector classification (or support vector machines) can be used in a wide range of circumstances, although it is based on the simple idea of finding a hyperplane in input space that separates the two classes under analysis. Modified versions of the algorithm can work with more than two classes. Since a hyperplane can only separate the instances of two classes in very particular cases, when the classes are linearly separable, one would think that SVMs have a very narrow range of applicability. However, that is not so, because SVM algorithms perform a transformation on the original space into a high-dimensional space using what is known as the kernel trick [10, 11], enabling SVM classifiers to work even in cases where the stroke data is not linearly separable [12].

10.2.4 Connectionist Methods

The original idea behind connectionist methods was inspired in the behavior of the human brain, which performs very sophisticated computations using millions of simple computational units (biological neurons) interconnected in complex, and mostly unknown, patterns. Biological neurons are complex cells, which perform computations by integrating incoming information, generated by upstream neurons and received in the dendritic trees. When the neuron excitation is sufficient to generate a signal at the output, the neuron generates a pulse or a train of electric pulses that are transmitted through the axon to other neurons downstream. The actual behavior of individual neurons is very complex, and can only be modeled by considering the actual physical characteristics of the neuron and the electrical parameters of the neuron membrane, among many other factors. However, the idea that supports connectionist methods, also known as artificial neural networks (ANN), or simply neural networks (NN), is to use a very simplified model of the behavior of neurons, illustrated in Fig. 10.2. In this simplified model, an artificial neuron simply computes a weighted sum of the inputs it receives and generates, at the output, a nonlinear function of this weighted sum. In this example, the nonlinear function is simply the Heaviside step function (equal to 1 if the input is positive, 0 otherwise), but other functions are extensively used, such as the sigmoid/logistic, the hyperbolic tangent, or the rectified linear unit.

A simple artificial neuron, like the one depicted in Fig. 10.2, does not perform a very useful computation. The true computational power of artificial neural networks derives from the fact that artificial neurons can be connected, in networks, in order to perform complex tasks, mimicking, in this way, the behavior of brains. Networks of artificial neurons, called multilayer perceptrons (MLPs), depicted in Fig. 10.3 can indeed perform very complex tasks, if the connection weights between the artificial neurons are set to the appropriate values.

The idea of using artificial neurons to process information is more than 60 years old [13] but this approach is only useful if appropriate algorithms to set the weights are available. Indeed, one of the winters of artificial intelligence was caused mainly by the realization that a single artificial neuron is not particularly useful and that

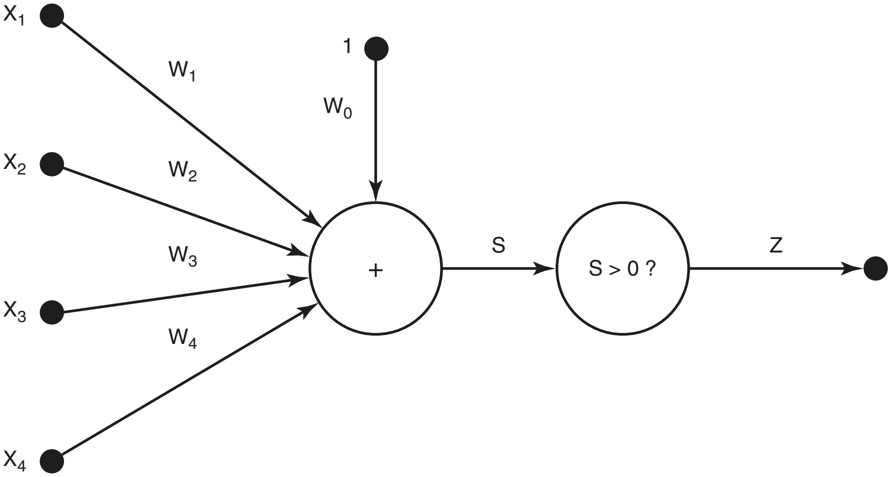

Fig. 10.2 Simplified mathematical model of a neuron

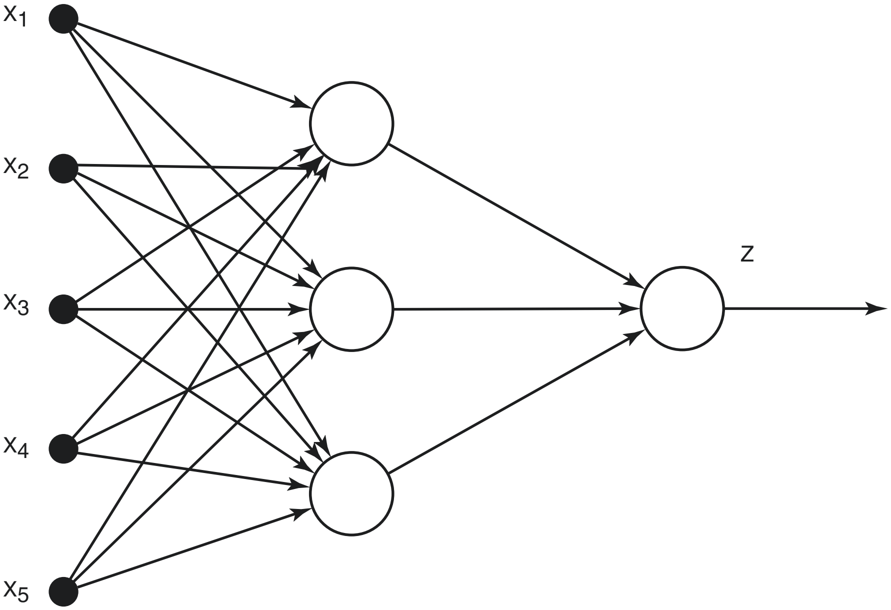

Fig. 10.3 Multilayer perceptron

finding the right value for the interconnection weights in a network is a very difficult computational problem [14]. However, the (re)discovery of the backpropagation [15] enabled researchers to apply artificial neural networks to a wide variety of problems. Backpropagation is a mathematical optimization method that derives the weights on an artificial neural network by computing the derivative of the error function with respect to each of the weights in the network. The error function can take several forms, but in general has a minimum when the network output is equal to the desired output, for all instances. One popular error function is the sum of the squares of the difference between the target output and the network output.

The set of derivatives of the error function with respect to each network weight is called, in mathematical terms, the gradient, and can be computed using standard mathematical operations, in particular the chain rule. Once computed, the gradient can be used to minimize the error at the output of the network, by performing a (typically very long) series of small changes in the weights, a process known as gradient descent. Gradient descent methods, which include backpropagation and conjugate gradient algorithms, among others, find the combination of weights that minimize the output error of the network. Such a configuration of weights can then be used to compute the network output for any combination of variables in the inputs.

For ANNs to be useful, the weights in the network have to be set in such a way as to make the network map the input variables to the desired output value. Using our running example, variables X_1–X_5 could be the attributes in columns 2–6 of Table 10.1 and the output Z would be the target label, the mRS value. When the weights are set to the right values, the network will compute the correct mRS value from the value of the input variables, the age, gender, past events, elapsed time, and NIHSS value of the admitted patient. Variables that are categorical have to be encoded as an integer, in order to be processed by the network units.

In this particularly straightforward problem of inferring the mRS from these five clinical and personal attributes, ANNs have a performance that is, probably, comparable to other methods we already discussed [8]. However, the power of artificial neural networks is that they can be applied to perform computations using high-dimensional data, such as images or sounds, which no other known machine learning method can cope with. Deep learning methods use artificial neural networks with many levels (in some cases, hundreds of levels) and hundreds of thousands of artificial neurons to perform image classification tasks that we were unable to perform, until only a few years ago.

10.3 Deep Learning for Image Processing

The problem of obtaining diagnosis from medical images is not different, in its essence, from the basic problem we have been discussing, that of inferring the target labels from a set of labeled instances. An image could in principle be described by a set of independent attributes, leading to a formulation not very different from the one that is exemplified in Table 10.1. Consider, for example, a grayscale, one-megapixel image obtained using any standard imaging technique. Such an image could be described by 1 million attributes, one attribute for each pixel in the image. The equivalent of Table 10.1 for such a task would consist of a table with 1 million columns, each column corresponding to the value of a pixel. Each entry in the table would be a number between 0 (for a black pixel) and 1 (for a white pixel), and intermediate values corresponding to intermediate levels of gray. For a color image, of the same dimension, the table would have 3 million attributes, since each pixel is described by three continuous values, for each of the three-color channels.

In principle, one could learn a classification rule from a table with 1 million columns (or 3 million columns, for color images) by using any of the techniques described in the previous section: decision trees, logistic regression, naive Bayes, nearest-neighbor, or support vector machines. In practice, such an approach does not work, because no single pixel provides any significant information about the target class. The relevant information is hidden in complex patterns that involve thousands or tens of thousands of pixels, patterns that are easily recognized by human experts but that are opaque to these basic machine learning methods. For that reason, the application of machine learning techniques to the classification of medical images was neither practical nor relevant, until a few years ago. Image processing algorithms were used to enhance the images, by manipulating contrast and by applying other image transformations, but stand-alone systems that could process and classify medical images were not common, if they existed at all. However, in the last decade, a significant number of developments took place, which enabled machine learning techniques to process and classify high-dimensional data, such as medical images or 3D medical images, obtained using magnetic resonance imaging or computed tomography.

The key development was the ability to apply convolutional neural networks to the processing of high-dimensional image data. Convolutional neural networks are multilayer perceptrons with a particular architecture where the first layers perform convolution operations on the image. Figure 10.4 shows an example of a convolutional neural network, with two convolutional layers and one fully connected layer at the output. Each convolutional layer applies a filter to the image, specified by a kernel, which is used to "scan" the image, applying the same filter over and over again as it is swept over the rows and columns of the image. In the simplified network in Fig. 10.4, only one kernel is used, leading to a single channel as an output of the first convolutional layer. In practice, several kernels are used in each layer, leading to a number of channels in each convolutional layer. The last layer (or layers) is usually a fully connected layer, enabling the output units (there is only one in this simplified case) to compute the desired output by combining the outputs of the neurons in the last convolutional layer. The kernel parameters (which are the input weights of the neurons in the convolutional layers) as well as all the other network weights are computed using gradient descent, as in standard neural networks. Other types of operations, such as pooling (selecting the largest value in a range), are also commonly used in convolutional neural networks.

In the last decade convolutional neural networks became the architecture best suited for many kinds of image processing tasks, among many other applications. Although convolutional neural networks have been proposed decades ago [16, 17] they became the solution of choice only more recently, with the availability of large datasets that can be used to train the networks, and the appearance of faster computers that speed up the computation of the solution using gradient descent methods, in many cases using graphic processing units (GPUs) developed for gaming

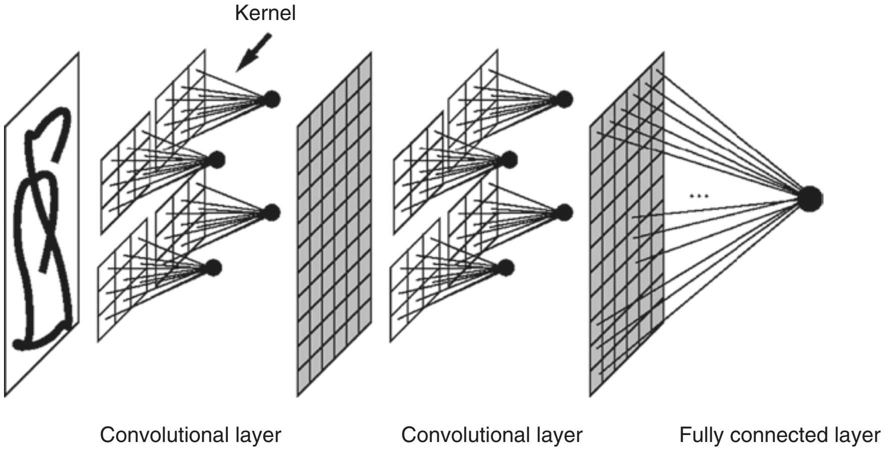

Fig. 10.4 Convolutional neural network

applications. After the first evidence that convolutional neural networks outperform any competing approaches in image recognition and labeling problems [18] many architectures with ever-increasing complexity [19–21] have been proposed and used in many different problems. These architectures have in common the fact that they included hundreds of thousands or even millions of artificial neurons, organized in very deep networks that exhibit many layers (in some cases, hundreds of layers).

10.4 Applications of Machine Learning in Stroke

Given the inherent flexibility of machine learning methods, and its ability to process many different types of data, it is no surprise that the number of applications in the area of stroke has exploded in the last few years. Applications include outcome prediction from clinical and/or imaging data, segmentation of lesions, and assessment of treatment effectiveness, among others.

The basic idea of outcome prediction is to infer the most likely outcome from clinical data available. Although a number of scores have been proposed to that effect [22], machine learning methods hold the promise of increasing the accuracy by being able to use more effectively all the available data. A number of authors developed methods to predict outcomes from clinical information, using support vector machines [12]; random forests, support vector machines, logistic regression, and decision trees [7]; decision trees [5]; neural networks, random forests, and logistic regression [8]; and support vector machines, random forests, and neural networks [23].

A number of high-profile public competitions to develop the most accurate methods to process MRI stroke data were held between 2015 and 2017. The 2015 edition of the ISLES (Ischemic Stroke Lesion Segmentation) challenge [24] proposed two tasks to the participants, subacute ischemic stroke lesion segmentation and acute stroke outcome/penumbra estimation. The 2016 edition consisted of two tasks: lesion outcome prediction and clinical outcome prediction, while the 2017 edition asked the participants to predict the lesion outcome [25]. In all cases, the available training data was multispectral MRI scans of acute stroke patients. Many other authors proposed to apply deep neural networks to process different types of stroke imaging data [26–29].

Given the increased availability of clinical and imaging data, and the ever-increasing effectiveness of machine learning methods, one should expect significant developments in the application of machine learning technologies to the task of outcome prediction from available information.

References

1. Domingos P. The master algorithm. London: Allen Lane; 2015.
2. Quinlan JR. Induction of decision trees. Mach Learn. 1986;1(1):81–106.
3. Quinlan JR. C4.5—programs for machine learning. San Mateo: Morgan Kaufmann; 1993.

4. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. Boca Raton, FL: CRC Press; 1984.
5. Saraee MH, Keane J. Using T3, an improved decision tree classifier, for mining stroke-related medical data. Methods Inf Med. 2007;46(5):523–9.
6. Letham B, Rudin C, McCormick TH, Madigan D. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. Ann Appl Stat. 2015;9(3):1350–71.
7. Monteiro M, Fonseca AC, Freitas AT, Pinho E, Melo T, Francisco AP, Ferro JM, Oliveira AL. Using machine learning to improve the prediction of functional outcome in ischemic stroke patients. IEEE/ACM Trans Comput Biol Bioinform. 2018;15(6):1953–9.
8. Heo JN, Yoon JG, Park H, Kim YD, Nam HS, Heo JH. Machine learning-based model for prediction of outcomes in acute stroke. Stroke. 2019;50(5):1263–5.
9. Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn. 1997;29(2–3):103–30.
10. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–97.
11. Scholkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond, Adaptive computation and machine learning series. Cambridge, MA: MIT Press; 2018.
12. Jeena RS, Kumar S. Stroke prediction using SVM. In: 2016 International conference on control, instrumentation, communication and computational technologies (ICCICCT 2016). Piscataway, NJ: IEEE; 2017. p. 600–2.
13. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev. 1958;65(6):386–408.
14. Minsky M, Papert S. Perceptrons. Cambridge: MIT Press; 1969.
15. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;5(6088):533–6.
16. Fukushima K, Miyake S. Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and cooperation in neural nets. Berlin: Springer; 1982. p. 267–85.
17. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten ZIP code recognition. Neural Comput. 1989;1(4):541–51.
18. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, 2012. p. 1097–105.
19. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Dec 2016. p. 770–8.
20. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision, 2015
21. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the 30th IEEE conference on computer vision and pattern recognition (CVPR 2017), Jan 2017. p. 2261–9.
22. Drozdowska BA, Singh S, Quinn TJ. Thinking about the future: a review of prognostic scales used in acute stroke. Front Neurol. 2019;10:274.
23. Lin C-H, Hsu K-C, Johnson KR, Fann YC, Tsai C-H, Sun Y, et al. Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry. Comput Methods Programs Biomed. 2020;190:105381.
24. Maier O, Menze BH, von der Gablentz J, Häni L, Heinrich MP, Liebrand M, et al. ISLES 2015—a public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI. Med Image Anal. 2017;35:250–69.
25. Winzeck S, Hakim A, McKinley R, Pinto JA, Alves V, Silva C, et al. ISLES 2016 and 2017-benchmarking ischemic stroke lesion outcome prediction based on multispectral MRI. Front Neurol. 2018;9:679.

26. Kamal H, Lopez V, Sheth SA. Machine learning in acute ischemic stroke neuroimaging. Front Neurol. 2018;9:7–12.
27. Sheth SA, Lopez-Rivera V, Barman A, Grotta JC, Yoo AJ, Lee S, et al. Machine learning-enabled automated determination of acute ischemic core from computed tomography angiography. Stroke. 2019;50(11):3093–100.
28. Nielsen A, Hansen MB, Tietze A, Mouridsen K. Prediction of tissue outcome and assessment of treatment effect in acute ischemic stroke using deep learning. Stroke. 2018;49(6):1394–401.
29. Chauhan S, Vig L, De Grazia MDF, Corbetta M, Ahmad S, Zorzi M. A comparison of shallow and deep learning methods for predicting cognitive performance of stroke patients from MRI lesion images. Front Neuroinform. 2019;13:53.