Innovative Deep Learning Approach for Biomedical Data Instantiation and Visualization

Ryad Zemouri, Daniel Racoceanu

Innovative Deep Learning Approach for Biomedical Data Instantiation and Visualization

Ryad Zemouri and Daniel Racoceanu

Abstract One of the main problems that most of biomedical applications face, is represented by the massive amount of unlabeled data. Manually analyzing and classifying massive database by human expert is mostly unfeasible, being - in certain limited conditions (still, extremely time-consuming) - partially been done, only for simple signatures, easily recognizable by an expert. Concerning this aspect, medical experts face two challenging problems: how to select the most significant data for labeling, and what is the minimum size of the data set - but sufficient to define each pathology - to perform the training of the classifier. In this chapter, we propose a new method, based on a visual data analysis, to build an efficient classifier with a minimum of labeled data. An encoder, part of a *Convolutional Variational Autoencoder* (CVAE), is used as a data projection for a 2D-visualization. The input vectors are encoded into a 2D-latent space, which helps the expert to visually analyze the spatial distribution of the training data set.

Key words: Data Visualisation, Deep Learning, Variational autoencoders, Breast Cancer diagnosis.

1 Introduction

Artificial Neural Networks (ANNs) and *Deep Learning* (DL) are actually the leading *Machine Learning* (ML) tools in biomedical fields, as reported by recent survey publications [46, 48, 29, 7, 34, 22, 2, 30]. With the technological and scientific

Ryad Zemouri
CEDRIC Laboratory of the Conservatoire National des Arts et Metiers (CNAM), HESAM University, Paris, France, e-mail: ryad.zemouri@cnam.fr

Daniel Racoceanu
Sorbonne University and Paris Brain Institute (CNRS UMR 7225, Inserm U 1127), Paris, France
e-mail: daniel.racoceanu@sorbonne-universite.fr

advances, the biomedical data used by the medical practitioners are very heterogeneous, such as a wide range of clinical analyses, biological parameters and medical imaging modalities. By the multitude of these data as well as the completeness of certain atypical diseases, biomedical data are usually imbalanced [20, 42] and non-stationary [11], being characterized by a high complexity [20]. In this context, ML represents a tremendous opportunity: (1) to support physicians, biologists and medical authorities to exploit and significantly improve big medical data analysis; (2) to reduce the risk of medical errors; and (3) to generate a better harmonization of the diagnosis and prognosis protocols.

The applications of the DL in the biomedical fields cover all the medical levels, i.e. from the genomic applications to the public medical health management, and are structured according to three main orientations [48]:

- *Computer-Aided Diagnosis*: to help the physicians for an efficient and early diagnosis, with a better harmonization and less contradictory diagnosis;
- *Patients Medical care*: to enhance the medical care of patients with better personalized therapies; and
- *Human wellbeing*: to improve the human wellbeing, for example by analyzing the spread of disease and social behaviors in relation with environmental factors, or to implement a brain–machine interface for controlling a wheelchair [1].

One of the main problems that most of biomedical applications face, is represented by the massive amount of unlabeled data. Manually analyzing and classifying massive database by a human expert is mostly unfeasible, being - in certain limited condition (still, extremely time-consuming) - partially been done, only for simple signatures, easily recognizable by the expert. Concerning this aspect, medical experts face two challenging problems: how to select the most significant data for labeling, and what is the minimum size of the data set - but sufficient to define each pathology - to perform the training of the classifier. In this chapter, we propose a new method, based on a visual data analysis, to build an efficient classifier with a minimum of labeled data. An encoder, part of a *Convolutional Variational Autoencoder* (CVAE) [49] [25] [50], is used as a data projection for a 2D-visualization. The input vectors are encoded into a 2D-latent space, which helps the expert to visually analyze the spatial distribution of the training data set.

The rest of the chapter is organized as follows : In section 2, we give a brief introduction to the ANNs with a particular focus to some of the weaknesses usually encountered. Then, section 3 presents some of the emerging architectures that have recently find a great success in the biomedical applications. Section 4 is dedicated to the *Variational Autoencoder* (VAE) [24], [23], [49] applied to data visualization and data analysis. An innovative DL approach for biomedical data instantiation and visualization is then presented. In section 5, we give some results around a practical case study which is the *Breast Cancer Wisconsin dataset*, available by anonymous ftp from ice.uci.edu [12]. Finally, critical discussion and open challenges are given in section 6.

2 Deep Neural Networks: A brief introduction

In this section, we first introduce a brief history of ANNs and present the main concepts of *Deep Neural Networks* (DNNs). Then, we develop the two major weaknesses of ANNs. The first one is the difficulty to find the best neural structure while the second is the lack of interpretability of the obtained results.

2.1 From Shallow to Deep Neural Networks

ANNs were inspired - in the 1960s - by biological neural networks in the brain. The feed forward ANNs are composed by layers of interconnected units (*neurons*). The mathematical point of view of ANNs consists of a non-linear transformation $y = F(x)$ of the input x (Fig1.A). Compared to shallow architectures, ANNs with more hidden layers, called DNNs [37], offer much higher capacity to learn fitting and feature extracting from high complexity input data (Fig1.B). The starting point of DL was in 2006, with the greedy layer-wise unsupervised learning algorithm used for *Deep Belief Networks* (DBNs) ([21, 5]).

The interconnection between two units or neurons, has an associated connection weight w_{ji}, which is fitted during the learning phase. The input data are propagated from the input layer, neuron after neuron, until the output layer. This propagation will transform these data from a given space to another one, by the neurons of the layers, in a nonlinear way. Each neuron computes a weighted sum of its inputs and applies a nonlinear activation function to calculate its output $f(x)$ (Fig1.C). The most used activation functions are the *sigmoid function* [37] and its variant the *hyperbolic tangent function* for the *shallow architectures*, the *Rectified Linear Unit function* (ReLU) and its variant the *softplus function* for the deep architectures, and the *softmax function* commonly used for the final layer in classification tasks.

The two main applications of the ANNs are $classification$ and *regression*. The objective of the classification is to organize the input data space into several classes by supervised or unsupervised learning techniques. In the regression applications or function approximation, the objective is to predict an unknown output parameter, usually by supervised learning.

In supervised learning, the predicted label is compared with the true label, for the current set of model weights θ_w, to compute the output error (also called *loss function* $L(\theta_w)$) (Fig2.A). The loss function is high-dimensional and non-convex, with many local optimums. The learning phase consists of tuning the connection weights at each learning step, in order to minimize $L(\theta_w)$, by backward propagating the gradient of the loss function, through the network. This back propagation gradient was the renew of the ANNs in the mid of the 80s, when the *Back Propagation* (BP) algorithm was used for classification [36]. During the learning procedure, two sets of data are usually used: training test and test set (sometime a third set is used for validation). The training set is used for the learning while the test set is used for the ANNs performances evaluation. An efficient learning algorithm is to converge

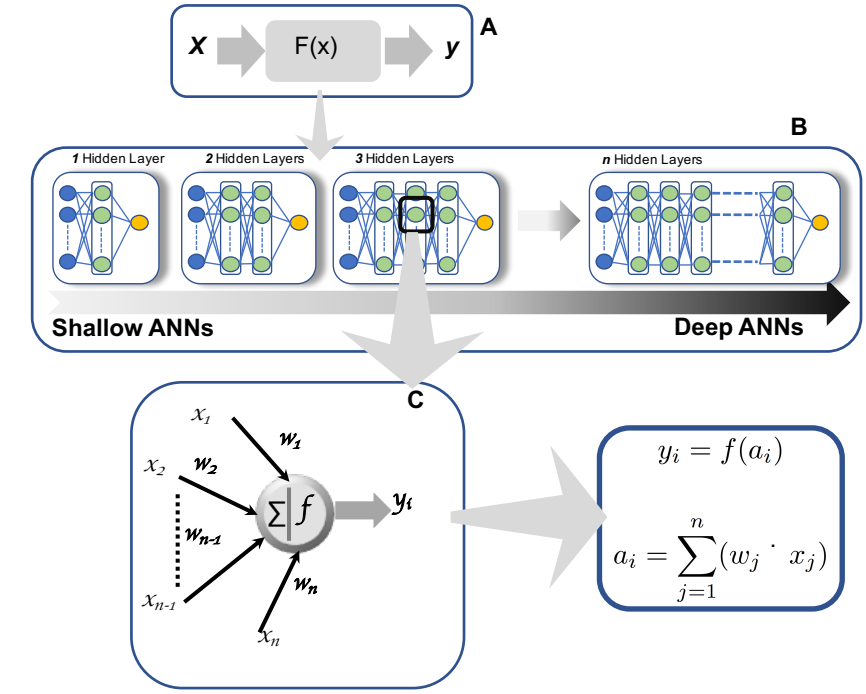

Fig. 1 ANNs [48].

towards a global optimum while avoiding all the local optimums of the loss function, which looks like a landscape, with many hills and valleys. A learning rate η is used to jump over valleys at the beginning of the training and fine-tune the weights in later stages of the learning process. If the learning rate is too low (little jump), it may take forever to converge with a high risk of jamming in a local optimum. Conversely, a too high value (big jump) can cause a non-convergence of the learning algorithm (Fig2.B). Varying and adapting the learning rate during the training process, produces better template update.

When deep architectures are used, the magnitude of the back propagated error derivative decreases rapidly along the layers, resulting in slight update of the weights in the first layers (Fig2.C).

This drawback was partially solved by using the ReLU and the *softplus* activation function which allow faster learning and superior performances compared to the conventional activation function (e.g. *sigmoid* or *hyperbolic tangent*). Other solution is to consider the learning rate as a hyper parameter where different learning rates are used for different layers. However, few works in the literature use this concept (see [8] for a review). The most popular method used to create deep architectures

and solve the problem of the random initialization of the weight parameters, consists in an unsupervised pre-training phase used before the supervised fined-tuned learning phase. *Auto-Encoders* (AEs) and *Restricted Boltzmann Machine* (RBM) are stacking in a layer-wise as the basic building blocks [37].

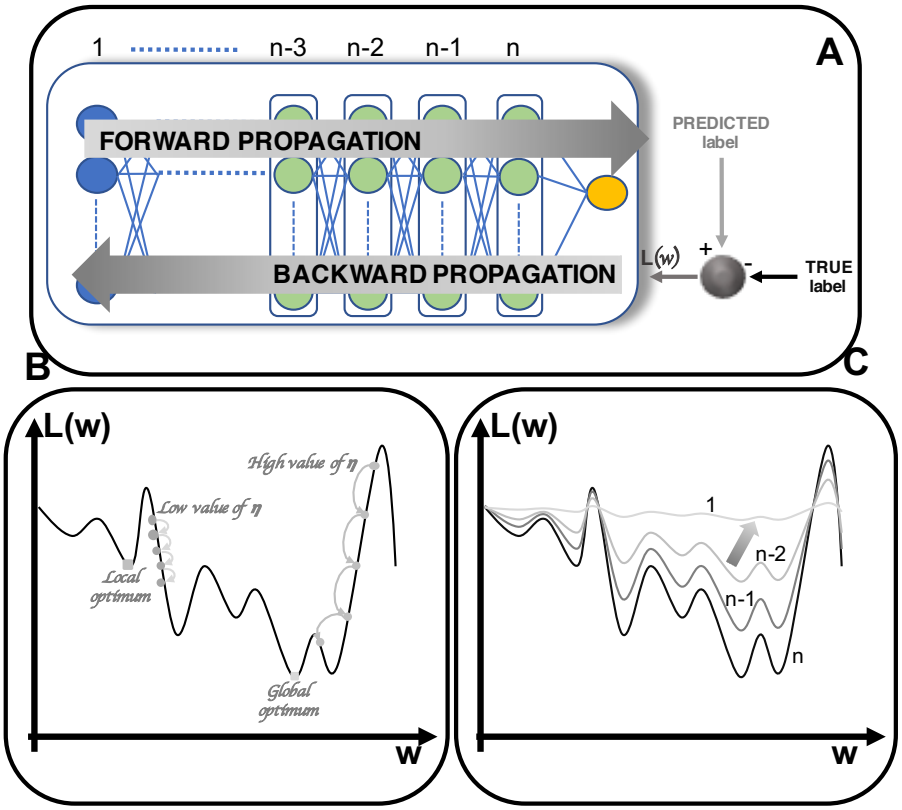

Fig. 2 The Loss function and the backward propagation of its gradient. A- The predicted label is compared with the true label for the current set of model weights θ_w to compute the output error (also called *loss function*) $L(\theta_w)$. B- The learning rate η is used to jump over valleys. If the learning rate is too low (little jump), it may take forever to converge with a high risk of jamming in a local optimum. Conversely, a too high value (big jump) can cause a non-convergence of the learning algorithm. C- For deep architectures, the magnitude of the back propagated error derivative decreases rapidly along the layers, resulting in slight update of the weights in the first layers [48].

2.2 Some Weaknesses

The different steps to follow when using an ANN are: 1) model choosing, 2) model building, 3) model learning, 4) model checking. In the first step, we must choose one neural architecture (CNN, AE, DBN, ...). In the second step, we have to define the size of the ANN: how many layers, how many units per layer, how many convolution filters and what is their size. In the third step, the ANN will be trained by unsupervised or supervised techniques while avoiding over-fitting and under-fitting. During the last step, we have to check the quality of the ANN.

2.2.1 Finding the best neural structure

The main difficulty with the ANNs is the model building. What are the criteria that define the number of hidden layers and neurons per layer? The user often proceeds by trying several ANN topologies to find the best structure and try to avoid the oversized and undersized structure.

Finding the best ANN architecture for a given problem is still challenging, especially for the DNN learning, which remains an active research area.

This is a real and computationally expensive drawback, especially when deep architectures are used. There is no guarantee that the selected number of hidden layers/units is optimal. To prevent the network from over-training (usually caused by the oversized design), some regularization techniques are used, as the *dropout* [39], *Maxout* [17] or the *weight decay* [32], which is a penalty added to the error function.

Evolutionary learning procedures give also interesting solutions where the ANN evolved gradually during the training procedure until an optimum structure that satisfy some evolutionary criteria. These adaptive ANNs are divided into three categories: *constructive* or *growing* algorithms [47], *pruning* algorithms [14] and *hybrid* methods [18]. A good alternative to this drawback is to consider the neural architecture as a hyper-parameter evolving during the learning process. The ANN is built step by step during the learning process, until convergence. To avoid an oversized architecture, some of the parameters as non-significant units or connections between neurons can be removed. Recently, several promising studies about the constructive and pruning algorithms were published (see [47], [33] for a complete survey).

2.2.2 The interpretability of the obtained results

The other difficulty is the interpretability of the obtained results. It is very hard to understand what happen in the hidden layers and why a trained ANN gives a positive diagnosis for a certain pathology.

ANNs learn to associate an output according to a given input, but they do not learn to give any reason or interpretation associated to this response.

This black-box aspect is very restrictive, specially in medicine, where a decision interpretability is very important and can have serious legal consequences [27]. When the convolution networks are used for image processing, several methods have been developed to visualize what happen in the intermediate layers. Some of these algorithms are, for example, visual explanations from CNN networks via gradient-based localization [38]; a visualization technique of the input stimuli that excite individual feature maps at any layer in a CNN model [44] or; a *deep Taylor decomposition method* [31] for interpreting generic multilayer ANNs by decomposing the network classification decision into contributions of its input elements [31]. When the input data are not images, the interpretability of the hidden layers activities is less obvious. Some visualization techniques, as the *t-Distributed stochastic Neighbor Embedding projection* (t-SNE) [28], converts a high-dimensional data set into a 2D-matrix of pairwise similarities. Feature maps of the model are then obtained; all the difficulty is to explain the classification decisions, according to these maps.

3 Emergent architectures: Generative Networks

One of the most emerging architectures used in the biomedical applications are the *Generative Networks* (GNs). The GNs provide a way of data augmentation to enlarge the deep representations without extensively annotated training data [9]. Two kinds of GNs exist:

- *Generative Adversarial Networks* (GANs) [16], [45]
- *Variational Auto-Encoders* (VAEs) [24], [23], [49].

3.1 The generative adversarial networks

Proposed in 2014 by Goodfellow [16], a GAN includes two DNNs: a generator and a discriminator. The first network is seen in a common analogy, as an art forger that creates forgeries with the aim of making realistic images. The discriminator represents the ”art expert” which should distinguish between the synthetic and the authentic images. The training of a GAN requires both finding the parameters of a discriminator and a generator, the discriminator having to maximize its classification accuracy and the generator having to confuse the discriminator as much as possible [45], [9]. During the training process, only one of the two networks is concerned by the parameters updating. The second one keeps its own parameters, frozen.

The GANs were recently applied in all the biomedical fields such as in Omics for a protein modeling [26], where the loop modeling is seen as the image in-painting problem and the generative network has to capture the context of the loop region with a prediction of the missing area.

In the *Brain and Body Machine Interface* (BBMI) applications [48] such as cardiac *ElectroCardioGram* (ECG) [41], a novel concept that embeds a generative VAE into the objective function of Bayesian optimization is applied to estimating tissue excitability in a cardiac electrophysiological model by [10]. In [15], a deep generative model is trained to generate the spatiotemporal dynamics of *TransMembrane Potential* (TMP).

In bioImaging histology applications, in order to classify the newly given prostate datasets into low and high *Gleason grade*, an adversarial training is used to minimize the distribution discrepancy at the feature space, with the loss function adopted from the GAN [35]. In [19], a cascaded of refinement GANs for phase contrast microscopy image super-resolution is proposed.

Most of the publications using the GANs, concern the medical imaging application for image quality enhancement, image reconstruction, crafted images generation or image registration and segmentation (see [48] for a extensive survey).

3.2 Variational auto-encoders

An AE is an unsupervised ANN trained to recreate or reproduce the input vector $\mathbf{x}$ [43], [4], [13] . The AE is composed by two main structures: an encoder and a decoder (Fig. 3) which are multilayered ANNs parameterized by ϕ and θ, respectively. The first one encodes the input data $\mathbf{x}$ into a latent representation $\mathbf{z}$ by the encoder function $\mathbf{z} = f_\phi(\mathbf{x})$, whereas the second one decodes this latent representation onto $\hat{\mathbf{x}} = h_\theta(\mathbf{z})$ which is an approximation or reconstruction of the original data. In an AE, an equal number of units are used in the input/output layers while less units are used in the latent layer (Fig. 3). The AEs are usually used for data compression (i.e., feature extraction/reduction), noise removal and pre-trained parameters for a complex network.

A VAE has the same functions as the AE; in the sense that it is composed by an encoder and a decoder (Fig. 3). VAE becomes a popular generative model by combining *Bayesian inference* [24], [23] and the efficiency of the ANNs to obtain a nonlinear low-dimensional latent space. The Bayesian inference is obtained by an additional layer used for sampling the latent vector $\mathbf{z}$ with a prior specified distribution $p(\mathbf{z})$, usually assumed to be a *standard Gaussian* $\mathcal{N}(0,\mathbf{I})$ [24], [23], where $\mathbf{I}$ is the identity matrix. Each element z_i of the latent layer is obtained as follow:

$$z_i = \mu_i + \sigma_i.\varepsilon \tag{1}$$

where μ_i and σ_i are the i^{th} components of the mean and standard deviation vectors, ε is a random variable following a *standard normal distribution* [24], [23]

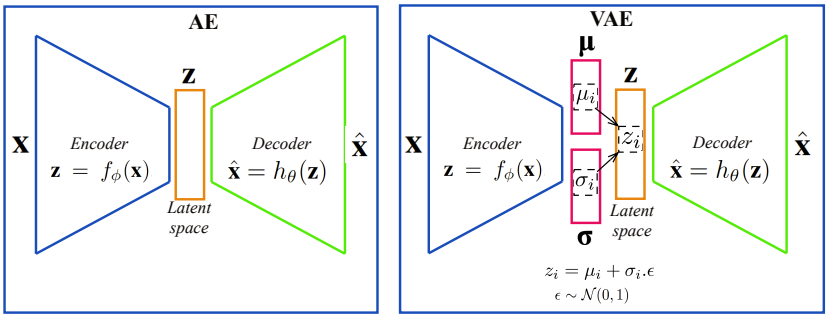

Fig. 3 Schematic architecture of a standard deep autoencoder and a variational deep autoencoder. Both architectures have two parts: an encoder and a decoder.

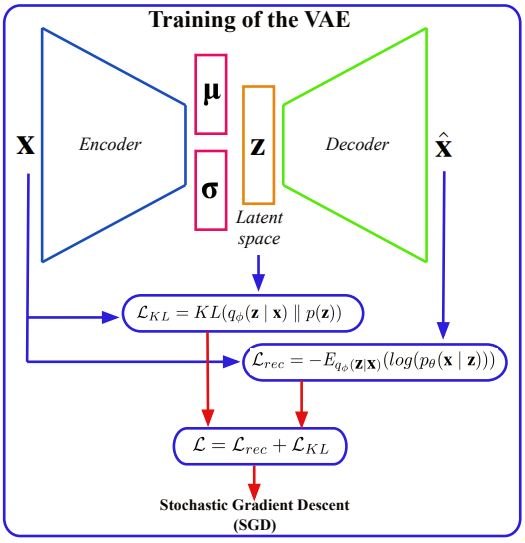

Fig. 4 The VAE loss function. The first term $\mathscr{L}_{rec}$ is the reconstruction loss function. The second term $\mathscr{L}_{KL}$ corresponds to the Kullback-Liebler divergence loss term that forces the generation of a latent vector with the specified Normal distribution. When the VAE is trained, the two functions encoder/decoder can be used separately even to reduce the space dimension by encoding the input data or to generate synthetic samples by decoding new variables from the latent space.

($\varepsilon \sim \mathscr{N}(0,1)$). Unlike the AE, which generates the latent vector $\mathbf{z}$, the VAE generates vector of means μ_i and standard deviations σ_i. This allows to have more continuity in the latent space than the original AE. The VAE loss function given by the equation 2 has two terms. The first term $\mathscr{L}_{rec}$ is the reconstruction loss function (equ. 3). Usually, the negative expected log-likelihood (e.g., the cross-entropy function) is used but also the mean squared error. The second term $\mathscr{L}_{KL}$ (equ. 4) corresponds to the *Kullback-Liebler* (KL) [24], [23] divergence loss term that forces the generation of a latent vector with the specified *normal distribution* [24, 23]. The

KL divergence is an information theoretic measure of proximity between two densities $q(x)$ and $p(x)$. It is asymmetric ($KL(q \parallel p) \neq KL(p \parallel q)$) and non-negative. It is minimized when $q(x) = p(x)$ [6]. Thus, the KL divergence term measures how closely the conditional distribution density $q_\phi(\mathbf{z} \mid \mathbf{x})$ of the encoded latent vectors is from the desired Normal distribution $p(\mathbf{z})$. The value of KL is zero when the two probability distributions are the same, which forces the encoder of VAE $q_\phi(\mathbf{z} \mid \mathbf{x})$ to learn the latent variables that follow a multivariate normal distribution over a k-dimensional latent space.

$$\mathscr{L} = \mathscr{L}_{rec} + \mathscr{L}_{KL} \tag{2}$$

$$\mathscr{L}_{rec} = -E_{q_\phi(\mathbf{z}|\mathbf{x})}(log(p_\theta(\mathbf{x} \mid \mathbf{z}))) \tag{3}$$

$$\mathscr{L}_{KL} = KL(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})) \tag{4}$$

When the VAE is trained, the two functions (encoder and decoder) can be used separately, even to reduce the space dimension by encoding the input data, or to generate synthetic samples by decoding new variables from the latent space (Fig. 4).

4 Variational autoencoders for data visualization and analysis

In this section, we present an innovative DL approach for biomedical data instantiation and visualization. This approach is based on the use of the VAE to solve the problem of the massive high dimensionality data that is usually encountered in biomedical applications. This section is an extension of our previous study [49]. We begin by a brief introduction on the importance of the spatial distribution of the training data. Next, we present the framework of the visual classification methodology as a support for data labeling.

4.1 Massive amount of unlabeled data

One of the main issue faced by most of the biomedical applications, concerns the massive amount of unlabeled data. Manually analyzing and classifying huge database by a human expert is mostly unfeasible, being partially been done, only for simple signatures, easily recognizable by the expert. Concerning this matter, the medical experts face two challenging problems: how to select the most significant data for labeling, and what is the minimum size of the data set - sufficient to define each pathology - in order to perform a proper training of the classifier.

Usually, one consider that classification improves with the number of labeled data, but this is usually, not enough. In fact, the quality of the classification depends on the spatial distribution of the training data set, which is a very important parameter to be considered before processing the training.

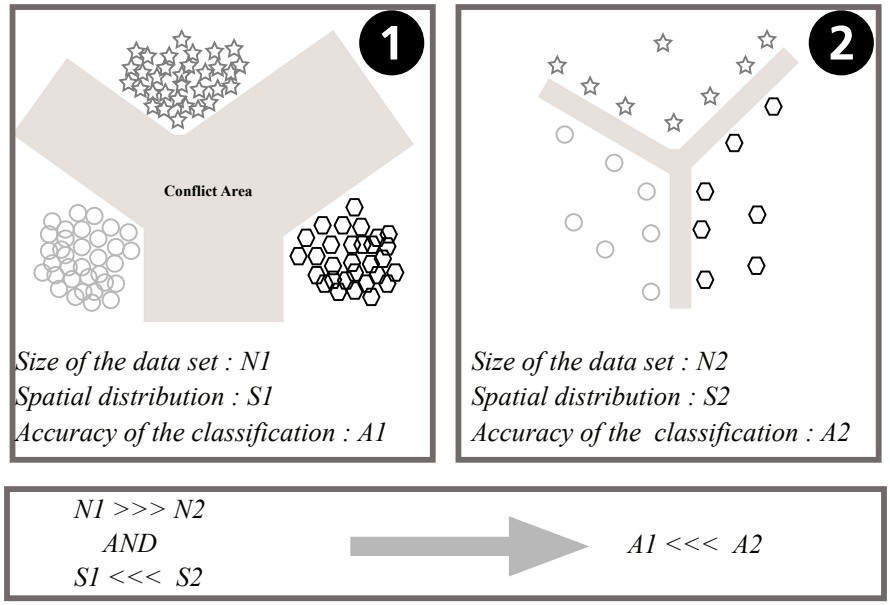

Fig. 5 Basic illustration of an arbitrary 2D-representation of two different training data sets with two different spatial distributions. The data set #1 has more samples than the data set #2 but the accuracy of the classification is better for the data set #2. The spatial distribution of the data is more important than the size of the dataset [49].

Fig. 5 gives a basic illustration of two different training data sets. The first set has more samples than the second one, but the spatial distribution of the second data set is better than the first one. The grey zone represents the conflict area, where most of the false positive predictions are produced by the classifier. Better the spatial distribution of the training data set, better the accuracy of the prediction. To reduce this ”dead zone”, the expert must choose some new points belonging or being near to this conflict area, for labeling.

Many research have been recently focusing on improving learning from imbalanced data, but none of these methods rely on a visual analysis of the learning data - one of the most intuitive approach. We consider that the visualization of the learning data to understanding its nature and to extracting more information from the 2D-space distribution, is an essential step during the design of the diagnosis model.

One of the main limitations of ANNs, is the lack of interpretability of the obtained results. It is very hard to understand what happens in the hidden layers and why a trained ANN gives a positive diagnosis for a given input sample. This "black-box" aspect is very restrictive in many medical application fields, where a decision interpretability can lead to serious legal consequences [27].

When the CNNs are used for medical image processing, several methods have been developed to visualize what happen in the intermediate layers. Some of these algorithms are for example visual explanations from CNNs via gradient-based localization [38]; a visualization technique of the input stimuli that excites individual feature maps at any layer in a CNN model [44] or; a deep Taylor decomposition method for interpreting generic multilayer ANNs by decomposing the network classification decision into contributions of its input elements [31]. When the input data are not images, as for the industrial measurements, the interpretability of the hidden layers activities is less obvious.

4.2 Visual Support for data labeling

Fig. 6 shows the framework of the visual classification methodology as a support for data labeling. An encoder, part of a VAE, is used as a data projection for a 2D-visualization. The input features vectors are encoded into a 2D-latent space, which helps the expert to visually analyze the space distribution of the training data set. At the beginning, few data points from the big unlabeled database are selected from the 2D-latent space. These points are labeled by the expert and used to train a neural classifier. The obtained classifier is then tested over all the unlabeled data set.

To identify the conflict area, i.e., the gray zones illustrated in the Fig. 5, several classifiers are trained on the same initial labeled data set, and tested over all the unlabeled data set. The conflict zones are identified on the 2D-space by analyzing the conflicts results of all the trained classifiers.

As we can see on the Fig. 5, if the conflict zone is too large, the boundary between classes, fixed by the classifier, is too uncertain. Usually, most of the false predictions occur near this conflict zone. To reduce the proportion of these false predictions, new points – near or belonging to these conflict zones – are then selected and labeled by the expert. These new labeled samples are added to the initial training set for a new training iterations. Therefore, several interactions between the diagnostic expert and the DL expert are necessary, in order to refine and reduce the boundary zone.

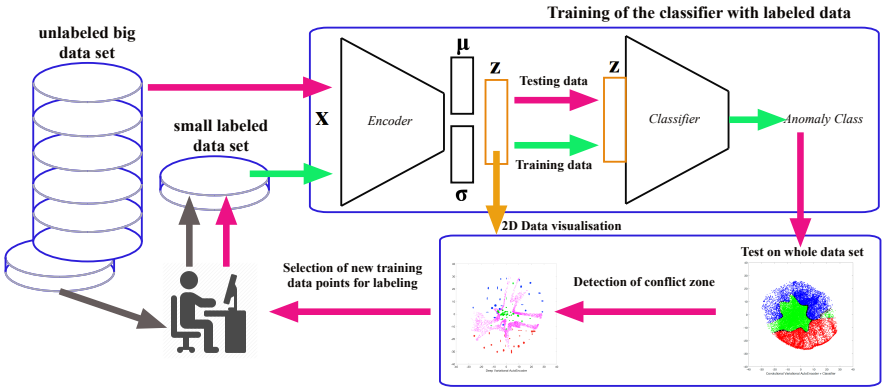

Fig. 6 The framework of the visual classification methodology using an encoder function of a VAE as a data projection for a 2D-visualization. The input features vectors are encoded into a 2D-latent space. If the conflict zone, identified on the 2D-space, is too large, new points near these conflict zones are selected and labeled by the expert. These new labeled points are then added to the initial training set for new training/testing iterations [49].

4.3 Identification and reduction of the conflict area

As described in the previous section, the conflict zone is the area in the data space definition where most of the false predictions occur. This is due to the poor coverage of the learning data space caused by the selection of only clear examples when designing the classifier. Because the human brain can only perceive two or three-dimensional space, it is impossible for humans to process the n-features dimensional space used to define the input vector of a biomedical data. It is even worse when trying to compare data between them. Therefore, projecting data in the 2D-latent space will help the expert to easily identify clusters of similar data and locate boundary or conflict zone on which he needs to work. When the conflict zone is too large, the boundary between classes, fixed by the classifier, leaves a large number of data with uncertain classification. This means that two different classifiers C_i and C_j with $i \neq j$ will definitely have two opposite responses for the same input data k:

$$\Psi_i(k) \neq \Psi_j(k) \tag{5}$$

where $\Psi_i(k)$ and $\Psi_j(k)$ are respectively the output class obtained by the classifiers C_i and C_j for the input sample k. To identify these conflict zones, several classifiers C_i were trained with the same 2D training data set Ω^{2D}_{train}. Subsequently, all the trained classifiers have been tested over the entire data set. For each input sample k of the data set, if two classifiers have two opposite responses, the input sample k is then considered as a conflict sample.

To reduce the proportion of false predictions, the size of the conflict zone must be reduced.

Therefore, the expert will choose new learning points in these poorly covered areas, in order to adjust the learning of the classifier. These new samples are then labeled by the expert and added to the previous training set. The entire procedure is then repeated until the conflict zone is considered as acceptable by the medical expert, without having to consider all the data but just a few additional points located in the conflict zone.

5 Case Study: The Breast Cancer Wisconsin dataset

In this section, we present a step-by-step practical case study which concerns the diagnosis of the breast cancer [3]. We show how the VAE can be practically used to solve the problem of labeling the minimum of data to quickly improve the learning process. The proposed DL approach has been developed using Keras[1] DL framework on a PC machine with 4.0-GHz Intel Xeon CPU and 32-GB memory.

5.1 Description of the dataset

The *Breast Cancer Wisconsin dataset* used as a case study is available by anonymous ftp from ice.uci.edu [12]. This dataset consists of 569 breast cancer patterns from the university of Wisconsin. Each pattern has 30 attributes and the dataset is divided into two classes, 212 are malignant and 357 are benign. The original dataset was proposed by Nick Street, Wolberg and Mangasarian [40].

5.2 Evaluation Metrics

To be able to evaluate the performance of various classification methods, there is a need to introduce quantitative criteria. A confusion matrix is commonly used to calculate these performance parameters (fig. 7). This matrix contains information about the actual and the predicted classifications:

- *True Positive* (TP) values are the number of Positive classification correctly classified as Positive,
- *True Negative* (TN) values represent the number of Negative classifications correctly classified as Negative,

[1] https://keras.io/

- *False Positive* (FP) values are the number of Negative classifications incorrectly classified as Positive,
- *False Negative* (FN) values represent the number of Positive classification incorrectly classified as Negative,
- *Not Classified* (NC) values represent the number of samples belonging to the conflict area.

Based on these values, the following metrics are thus calculated:

- *Accuracy* (Acc) = $\frac{TP+TN}{TP+TN+FP+FN+NC}$
- *Negative Predictive Value* (NPV) = $\frac{TN}{FN+TN}$
- *Positive Predictive Value* (PPV) = $\frac{TP}{FP+TP}$
- *True Negative Rate* (TNR) = $\frac{TN}{FP+TN}$
- *True Positive Rate* (TPR) = $\frac{TP}{TP+FN}$

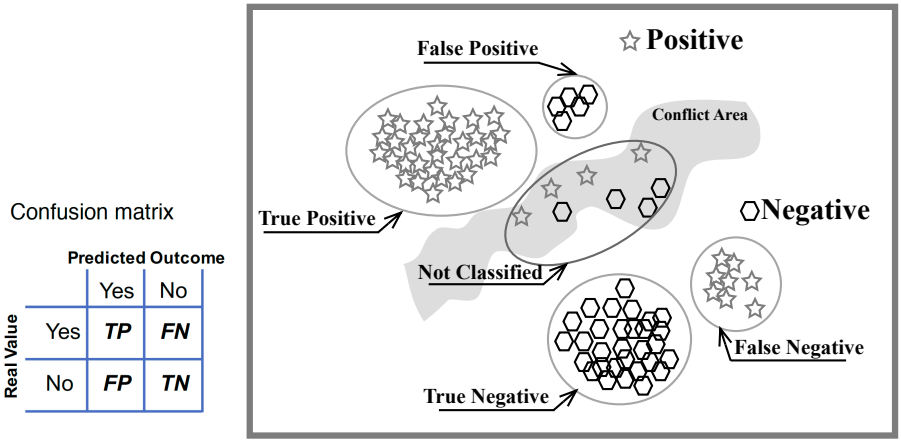

Fig. 7 Confusion matrix

5.3 The variational auto-encoder used as visual support

The VAE architecture used is illustrated by the Fig. 8. This architecture includes two parts, an encoder and a decoder, which are two symmetrical and reversed structures. Each one is composed by three fully connected layers. The latent two-dimensional

space is represented by two layers for the encoder: the mean and the standard deviation layers (i.e., μ and σ), and one sampling layer ($\mathbf{Z}$) for the decoder.

Training the VAE does not need the label information of the input data. However, for an efficient data encoding, we used an indirect labeling, since all training samples belong to one of the classes described above.

The first step was to train the whole VAE architecture for the reconstruction of the feature vector ($\hat{\mathbf{S}} = \mathscr{F}(\mathbf{S})$). When the training process of the VAE is done, the encoder part is then used jointly with a neural classifier, as presented in Fig. 8. The mean layer μ of the variational encoder is considered as the input 2D-vector of the classifier. The second step is then to train the classifier for the diagnosis. The encoder parameters obtained by the previous step are then frozen during the classifier training step.

All the details of DNNs architectures used, i.e. the VAE and the classifier, are presented in the Table 1.

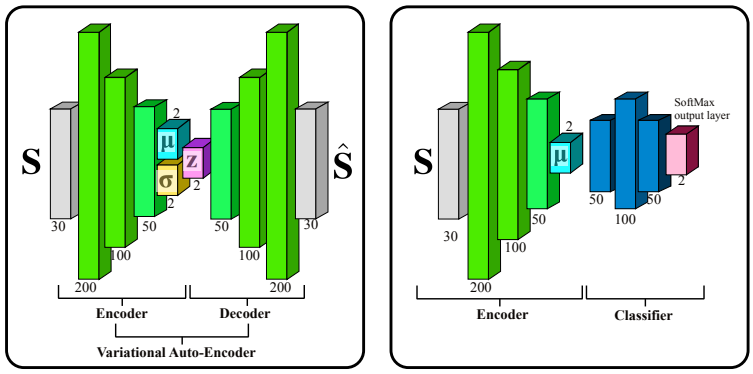

Fig. 8 The VAE architecture used. The encoder used jointly with a neural classifier. The mean layer of the convolutional encoder is considered as the input 2D-vector of the classifier.

5.4 Visualization of the latent space

Considering that the latent vectors are the encoding representation of the input feature vectors, it is interesting to visualize the 2-dimensional representation of the original feature and to evaluate the similarities within each class. Figure 9 shows the 2D-latent representation at different iterations of the training process of the variational auto-encoder.

Table 1 Proposed DNNs architectures.

#	Layer Type	Neurons	#	Layer Type	Neurons
Encoder			Classifier		
0	Input feature vector	30	0	Input (mean layer)	2
1	Fully connected	200	1	Fully connected	50
2	Fully connected	100	2	Fully connected	100
3	Fully connected	50	3	Fully connected	50
4	Mean layer	2	4	SoftMax output layer	2
4	Standard deviation layer	2			
Decoder					
0	Sampling layer	2			
1	Fully connected	50			
2	Fully connected	100			
3	Fully connected	200			
4	Output reconstructed vector	30			

The 2D-latent representation is the output of the sampling layer Z of the encoder illustrated in the figure 8.

At the beginning of the training process (i.e. iteration 1), the 2-D representation looks like a compact cluster. As the learning progresses, the cluster extends into the 2-D latent space until covering the entire space (iteration 1000), with almost a uniform Normal distribution – forced by the Kullback-Liebler divergence loss term (see section 3.2). Two clusters are then formed, the green for the benign class and the red for the malignant class.

5.5 Selection of the training samples from the 2D-latent space

Considering that all the used dataset is unlabeled, we will see how the 2D-latent space used as a visualization support can help the expert to select the most significant data for labeling. The figure 10.A shows the 2D-projection of all the dataset obtained by the variational encoder. Since all the dataset is unlabeled, all the samples belong to the same cluster.

5.5.1 Step1: Selection of the first training set

At the beginning, the expert must choose some samples to label for the training process. Instead of choosing randomly the samples, we can use the 2D-projection as a visualization support to choose the samples with respect to a certain spatial distribution. For example, as shown by the figure 10.A, a first set of six samples (Set

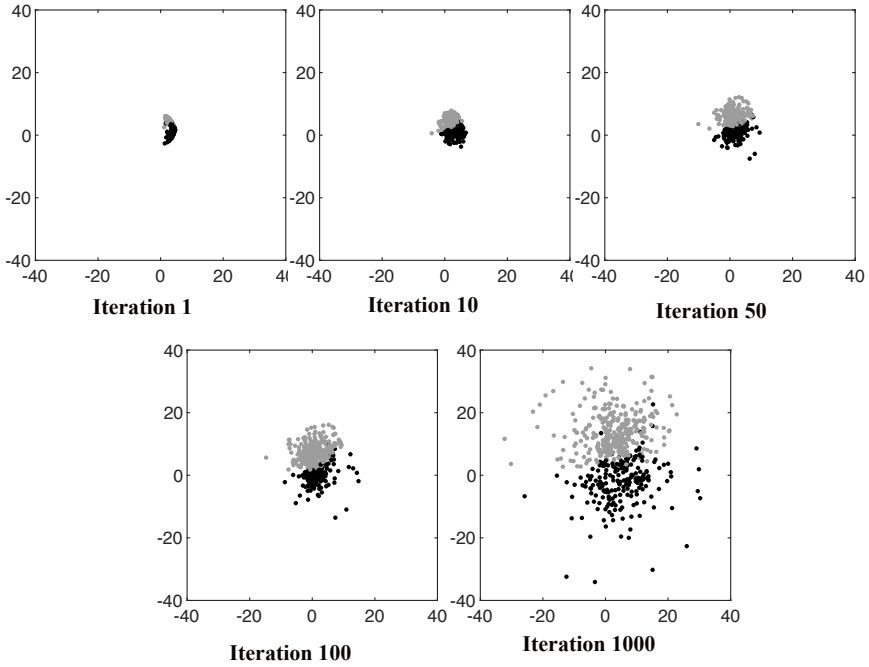

Fig. 9 The progression of the latent space throughout the learning process of the VAE.

#1) is selected from the 2D-latent space: 3 benign samples and 3 malignant samples. These samples are used for the training process of the classifiers.

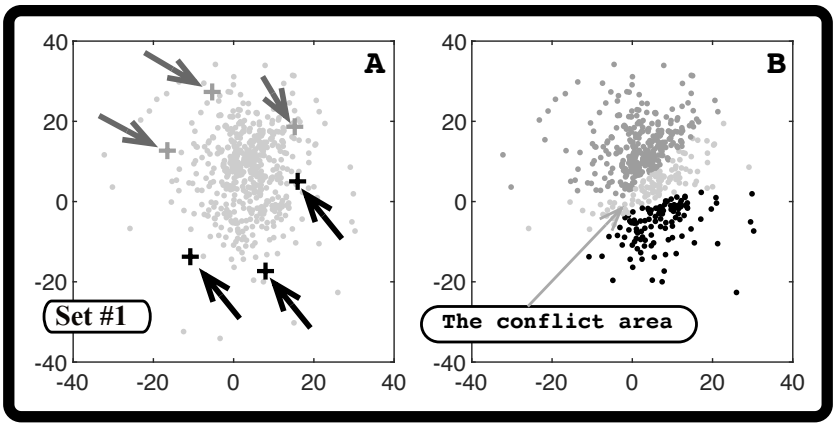

Fig. 10 A. First training set selection from 2D-projection, B. Identification of the conflict area.

5.5.2 Step2: Identification of the conflict area

As described by the section 4.3, to identify the conflict area between the two classes (i.e. benign and malignant), we train different classifiers with the same training samples. The conflict area, in the figure 10.B, represents the cases where the classifiers are in disagreement.

5.5.3 Step3: Reduction of the conflict area

To reduce the conflict area, new samples are selected nearby or belonging to the conflict area. As shown by the figure 11.A, a second set of six new samples (Set #2) is selected from the 2D-latent space and added to the initial training set. After a new training process of the classifiers, the new obtained conflict area is then reduced, as we can see on the figure 11.B

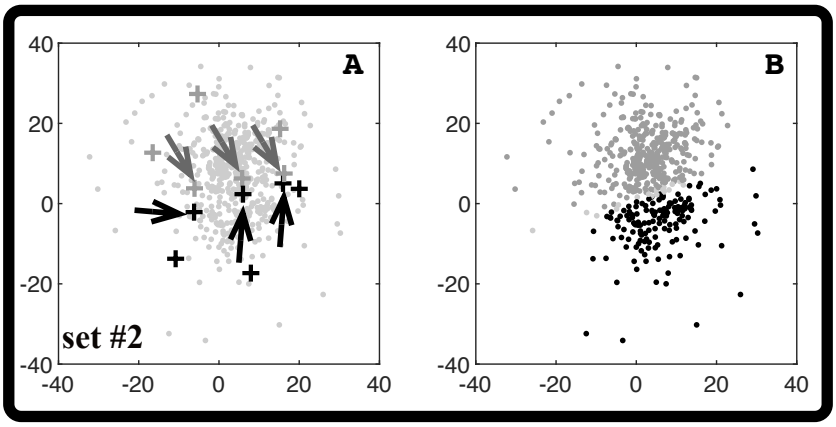

Fig. 11 A. Selection of new training samples nearby or belonging to the conflict area from the 2D-Projection, B. Reduction of the new conflict area.

5.5.4 Results Analysis

The table 2 shows the entire classification performances obtained on the whole dataset for each of the training set#1 and #2. The metrics used to analyze the obtained results are those presented by the section 5.2. We suppose that the malignant set is the Positive class and the benign is the Negative class. We can observe that the conflict area (i.e. the not classified patterns) has been scaled down between the

training set#1 and #2. Indeed, this disagreement zone decreases from 112 samples to 21, which represents respectively 19.68 and 3.69% compared to the whole dataset.

By reducing the proportion of the uncertainty area, the size of the true and false predictions increases. In fact, the number of the True Positive and the True Negative samples has been raised, which gives a better accuracy for the training set#2 (88.22% instead of 77.50% for the set#1). On the other hand, the proportion of the false predictions have been also raised by reducing the conflict area, which proves that most of the false predictions are near or within the conflict area. For this reason, the NPV, PPV, TNR and the TPR values are better for the training set #1.

Table 2 Results of the classification performances obtained on the whole dataset for each one of the training set#1 and #2. The malignant set is the Positive class and the benign is the Negative one.

Training Set	TP	TN	FP	FN	Not Classified
Set #1	116 (20.39%)	325 (57.12%)	0 (0%)	16 (2.81%)	112 (19.68%)
Set #2	148 (26.01%)	354 (62.21%)	1 (0.18%)	45 (7.91%)	21 (3.69%)
	Acc	NPV	PPV	TNR	TPR
Set #1	77.50%	95.31%	100%	100%	87.88%
Set #2	88.22%	88.72%	99.33%	99.72%	76.68%

5.6 Random selection of training samples

This section provides comparison results between the proposed 2D-visual selection method and a random selection of the training samples. The figure 12 gives the 2D-representation of four training sets (set #3, 4, 5 and 6) randomly selected from the entire dataset. These sets are respectively shown in the figures A, C, E and G. For each example, the conflict areas obtained by the confrontation tests of the classifiers are illustrated in the figures B, D, F and H.

The table 3 summarizes the results of the classification performances obtained for each training group on the whole dataset using the performances metrics listed in the previous section. The analysis of these comparative results reveals several strong points. First, it is evident that the performances are very disparate for the random selection, as we can see for the TP values (47 instances for the sets #3 against 120 for the set#6). For each random selection, thanks to the 2D-representation, we can visually evaluate the spatial distribution of the chosen training instances (fig. 12). Qualitatively, the distributions of the sets #1 and #2 (shown by the figures 10 and 11) are better than those obtained by the sets #3, 4, 5 and 6. Indeed, the data space is better covered by the learning sets #1 and #2. This better coverage of the data space necessarily leads to better classification results. It is clearly observed that the performances obtained by these two groups outperform the others by a large margin with the highest accuracy. The comparison results between our 2D-visual

selection method and the conventional random selection method demonstrate that our framework benefits of a better data visibility and a better results interpretation.

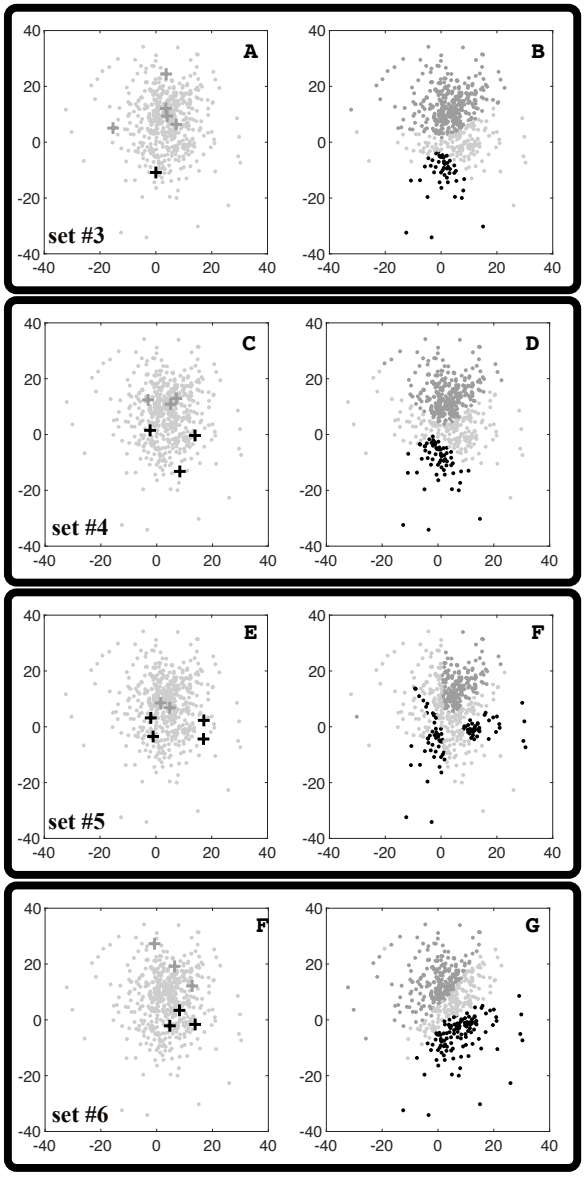

Fig. 12 Four examples of a random selection. The figures A, C, E and G gives four examples of four training sets (set #3, 4, 5 and 6) randomly selected from the entire dataset. For each training set, the obtained classifiers are tested on the whole dataset where different conflict areas are obtained, as shown by the figures B, D, F and H.

Table 3 Comparison results between the proposed 2D-visual selection method (Sets #1 and 2) and a random selection of the training samples (Sets #3, 4, 5 and 6). For each set, we give between brackets the number of instances per class (# of benign, # of malignant).

Training Set	TP	TN	FP	FN	Not Classified
Set #1 (3, 3)	116 (20.39%)	325 (57.12%)	0 (0%)	16 (2.81%)	112 (19.68%)
Set #2 (6, 6)	148 (26.01%)	354 (62.21%)	1 (0.18%)	45 (7.91%)	21 (3.69%)
Set #3 (5, 1)	47 (8.26%)	344 (60.46%)	0 (0%)	37 (6.50%)	141 (24.78%)
Set #4 (3, 3)	63 (11.07%)	317 (55.71%)	0 (0%)	11 (1.93%)	178 (31.28%)
Set #5 (2, 4)	114 (20.04%)	253 (44.46%)	50 (8.79%)	16 (2.81%)	136 (23.90%)
Set #6 (3, 3)	120 (21.09%)	274 (48.15%)	1 (0.18%)	9 (1.58%)	165 (29%)
	Acc	NPV	PPV	TNR	TPR
Set #1 (3, 3)	77.50%	95.31%	100%	100%	87.88%
Set #2 (6, 6)	88.22%	88.72%	99.33%	99.72%	76.68%
Set #3 (5, 1)	68.71%	90.29%	100%	100%	55.95%
Set #4 (3, 3)	66.78%	96.64%	100%	100%	85.13%
Set #5 (2, 4)	64.49%	94.05%	69.51%	83.49%	87.69%
Set #6 (3, 3)	69.24%	96.81%	99.17%	99.63%	93.02%

6 Conclusion

In this chapter, we investigated the use of VAE as a support for data visualization and diagnosis interpretation. However, after training of the VAE, the encoder part is used as a data projection from an input features mapping to a latent 2D-mapping. The VAE is trained without using class labels to learn properties in the data. The power of VAE falls in capturing the complicated data features from the multi-dimensional input space and compressing them into a smaller 2D-latent space, more easy for a visualization by a human expert. The VAE reveals a promising tool to produce a 2-dimensional embedding of high dimensional data with the goal of simplifying the identification of clusters when used jointly with a classifier.

As articulated in the results section, it is quite easy to understand the diagnosis given by the neural classifier by visually analyzing the spatial distribution of the classified samples. Therefore, the knowledge area of the ANN and the boundaries between the classes are perceived by the expert. The conflict regions which are poorly covered by the training samples are then easily identified. The ANN is then less perceived as a ”black-box” in the sense that its knowledge area is visible, the interpretation of the false predictions and their understanding become realizable.

The ending condition of the proposed algorithm is the diagnosis model evaluation through the testing of a performance criterion. The criterion used is the reduction of the conflict area, which is visually perceived by the expert, at each round of the algorithm. To improve the performances of the proposed method, other metrics than the reduction of the conflict area must be defined and used to evaluate the performances of a such model during the building process. These metrics should give some answering elements to these non-exhaustive questions:

- How to evaluate the quality of the spatial distribution of such an input training dataset?
- How to evaluate the features used which are closely related to the quality of the input vector?
- How to evaluate the performances of the VAE, especially the encoder function?
- A more fundamental question concerns the minimum size of the conflict area tolerated by the medical expert - a real dilemma for biomedical applications. For some borderline cases, an ambiguity response from the ANN can be more acceptable than a false prediction. For these critical cases, It is then preferable that the ANN lets the medical expert decide, rather than providing a catastrophic false prediction.

References

[1] Alvarado-Díaz W, Lima P, Meneses-Claudio B, Roman-Gonzalez A (2017) Implementation of a brain-machine interface for controlling a wheelchair. In: 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), pp 1–6, DOI 10.1109/CHILECON.2017.8229668

[2] Angermueller C, Pärnamaa T, Parts L, Stegle O (2016) Deep learning for computational biology. Molecular Systems Biology 12(7), DOI 10.15252/msb.20156651, URL http://msb.embopress.org/content/12/7/878, http://msb.embopress.org/content/12/7/878.full.pdf

[3] Baltres A, Zeina AM, et al RZ (2020) Prediction of oncotype dx recurrence score using deep multi layer perceptrons in estrogen receptor-positive, her2 negative breast cancer. Breast Cancer 27(5):1007–1016, DOI https://doi.org/10.1007/s12282-020-01100-4

[4] Bengio Y (2014) How auto-encoders could provide credit assignment in deep networks via target propagation. CoRR abs/1407.7906, URL http://arxiv.org/abs/1407.7906, 1407.7906

[5] Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. In: Schölkopf B, Platt J, Hoffman T (eds) Advances in Neural Information Processing Systems (NIPS 06), MIT Press, pp 153–160, DOI http://www.iro.umontreal.ca/ lisa/pointeurs/BengioNips2006All.pdf

[6] Blei DM, Kucukelbir A, McAuliffe JD (2016) Variational inference: A review for statisticians. arXiv e-prints arXiv:1601.00670, 1601.00670

[7] Cao C, Liu F, Tan H, Song D, Shu W, Li W, Zhou Y, Bo X, Xie Z (2018) Deep learning and its applications in biomedicine. Genomics, Proteomics & Bioinformatics 16(1):17 – 32, DOI https://doi.org/10.1016/j.gpb.2017.07.003, URL http://www.sciencedirect.com/science/article/pii/S1672022918300020

[8] Chandra B, Sharma RK (2016) Deep learning with adaptive learning rate using laplacian score. Expert Systems with Applications 63:1 – 7, DOI http://doi.org/10.1016/j.eswa.2016.05.022

[9] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: An overview. IEEE Signal Processing Magazine 35(1):53–65, DOI 10.1109/MSP.2017.2765202

[10] Dhamala J, Ghimire S, Sapp JL, Horáček BM, Wang L (2018) High-dimensional bayesian optimization of personalized cardiac model parameters via an embedded generative model. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Springer International Publishing, Cham, pp 499–507

[11] Ditzler G, Roveri M, Alippi C, Polikar R (2015) Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine 10(4):12–25, DOI 10.1109/MCI.2015.2471196

[12] Dua D, Graff C (2017) UCI machine learning repository. URL http://archive.ics.uci.edu/ml

[13] Fan YJ (2019) Autoencoder node saliency: Selecting relevant latent representations. Pattern Recognition 88:643 – 653, DOI https://doi.org/10.1016/j.patcog.2018.12.015, URL http://www.sciencedirect.com/science/article/pii/S0031320318304369

[14] Fnaiech N, Fnaiech F, Jervis BW (2011) Feedforward NeuralNetworks Pruning Algorithms, Industrial Electronics Handbook, vol 5, j.d. irwin, 2nd edn, chap 15, pp 15–1 to 15–15

[15] Ghimire S, Dhamala J, Gyawali PK, Sapp JL, Horacek M, Wang L (2018) Generative modeling and inverse imaging of cardiac transmembrane potential. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Springer International Publishing, Cham, pp 508–516

[16] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in Neural Information Processing Systems 27, Curran Associates, Inc., pp 2672–2680, URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[17] Goodfellow IJ, Warde-farley D, Mirza M, Courville A, Bengio Y (2013) Maxout networks. In: In ICML

[18] Han HG, Qiao JF (2013) A structure optimisation algorithm for feedforward neural network construction. Neurocomputing 99:347 – 357, DOI http://dx.doi.org/10.1016/j.neucom.2012.07.023

[19] Han L, Yin Z (2018) A cascaded refinement gan for phase contrast microscopy image super resolution. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Springer International Publishing, Cham, pp 347–355

[20] He H, Garcia EA (2009) Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9):1263–1284, DOI 10.1109/TKDE.2008.239

[21] Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554, DOI 10.1162/neco.2006.18.7.1527

[22] Jones W, Alasoo K, Fishman D, Parts L (2017) Computational biology: deep learning. Emerging Topics in Life Sciences 1(3):257–274, DOI 10.1042/ETLS20160025, URL http://www.emergtoplifesci.org/content/1/3/257, http://www.emergtoplifesci.org/content/1/3/257.full.pdf

[23] Kingma D (2017) Variational inference & deep learning: A new synthesis. PhD thesis, Faculty of Science (FNWI), Informatics Institute (IVI), University of Amsterdam, URL https://hdl.handle.net/11245.1/8e55e07f-e4be-458f-a929-2f9bc2d169e8

[24] Kingma DP, Welling M (2013) Auto-Encoding Variational Bayes. arXiv e-prints arXiv:1312.6114, 1312.6114

[25] Lee S, Kwak M, Tsui KL, Kim SB (2019) Process monitoring using variational autoencoder for high-dimensional nonlinear processes. Engineering Applications of Artificial Intelligence 83:13 – 27, DOI https://doi.org/10.1016/j.engappai.2019.04.013, URL http://www.sciencedirect.com/science/article/pii/S0952197619300983

[26] Li Z, Nguyen SP, Xu D, Shang Y (2017) Protein loop modeling using deep generative adversarial network. In: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), pp 1085–1091, DOI 10.1109/ICTAI.2017.00166

[27] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JA, van Ginneken B, Sánchez CI (2017) A survey on deep learning in medical image analysis. Medical Image Analysis 42:60 – 88, DOI https://doi.org/10.1016/j.media.2017.07.005, URL http://www.sciencedirect.com/science/article/pii/S1361841517301135

[28] van der Maaten L, Hinton G (2008) Visualizing data using t-sne. Journal of Machine Learning Research 9(11):2579–2605

[29] Mahmud M, Kaiser M, Hussain A, Vassanelli S (2018) Applications of deep learning and reinforcement learning to biological data. IEEE Trans Neural Netw Learn Syst DOI 10.1109/TNNLS.2018.2790388.

[30] Min S, Lee B, Yoon S (2016) Deep learning in bioinformatics. CoRR abs/1603.06430, URL http://arxiv.org/abs/1603.06430, 1603.06430

[31] Montavon G, Lapuschkin S, Binder A, Samek W, Müller KR (2017) Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition 65:211 – 222, DOI https://doi.org/10.1016/j.patcog.2016.11.008, URL http://www.sciencedirect.com/science/article/pii/S0031320316303582

[32] Nakamura K, Hong B (2019) Adaptive weight decay for deep neural networks. CoRR abs/1907.08931, URL http://arxiv.org/abs/1907.08931, 1907.08931

[33] Pérez-Sánchez B, Fontenla-Romero O, Guijarro-Berdiñas B (2016) A review of adaptive online learning for artificial neural networks. Artificial Intelligence Review DOI 10.1007/s10462-016-9526-2

[34] Ravì D, Wong C, Deligianni F, Berthelot M, Andreu-Perez J, Lo B, Yang GZ (2017) Deep learning for health informatics. IEEE Journal of Biomedical and Health Informatics 21(1):4–21, DOI 10.1109/JBHI.2016.2636665

[35] Ren J, Hacihaliloglu I, Singer EA, Foran DJ, Qi X (2018) Adversarial domain adaptation for classification of prostate histopathology whole-slide images. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Springer International Publishing, Cham, pp 201–209

[36] Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533 EP –, URL http://dx.doi.org/10.1038/323533a0

[37] Schmidhuber J (2015) Deep learning in neural networks: An overview. Neural Networks 61:85 – 117, DOI http://doi.org/10.1016/j.neunet.2014.09.003

[38] Selvaraju RR, Das A, Vedantam R, Cogswell M, Parikh D, Batra D (2016) Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. CoRR abs/1610.02391, URL http://arxiv.org/abs/1610.02391, 1610.02391

[39] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15:1929–1958, DOI http://jmlr.org/papers/v15/srivastava14a.html

[40] Street WN, Wolberg WH, Mangasarian OL (1993) Nuclear feature extraction for breast tumor diagnosis. International Symposium on Electronic Imaging Science and Technology 1905, DOI 10.1117/12.148698, URL https://doi.org/10.1117/12.148698

[41] Tan JH, Hagiwara Y, Pang W, Lim I, Oh SL, Adam M, Tan RS, Chen M, Acharya UR (2018) Application of stacked convolutional and long short-term memory network for accurate identification of cad ecg signals. Computers in Biology and Medicine 94:19 – 26, DOI https://doi.org/10.1016/j.compbiomed.2017.12.023, URL http://www.sciencedirect.com/science/article/pii/S0010482517304201

[42] Yu H, Yang X, Zheng S, Sun C (2018) Active learning from imbalanced data: A solution of online weighted extreme learning machine. IEEE Transactions on Neural Networks and Learning Systems pp 1–16, DOI 10.1109/TNNLS.2018.2855446

[43] Yu S, Príncipe JC (2019) Understanding autoencoders with information theoretic concepts. Neural Networks 117:104 – 123, DOI https://doi.org/10.1016/j.neunet.2019.05.003, URL http://www.sciencedirect.com/science/article/pii/S0893608019301352

[44] Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision – ECCV 2014, Springer International Publishing, Cham, pp 818–833

[45] Zemouri R (2020) Semi-supervised adversarial variational autoencoder. Machine Learning and Knowledge Extraction (MAKE) 2(3):361–378, DOI https://doi.org/10.3390/make2030020

[46] Zemouri R, Devalland C, Valmary-Degano S, Zerhouni N (2019) Intelligence artificielle : quel avenir en anatomie pathologique ? Annales de Pathologie 39(2):119 – 129, DOI https://doi.org/10.1016/j.annpat.2019.01.004, URL http://www.sciencedirect.com/science/article/pii/S0242649819300203, l'anatomopathologie augmentée

[47] Zemouri R, Omri N, Fnaiech F, Zerhouni N, Fnaiech N (2019) A new growing pruning deep learning neural network algorithm (gp-dlnn). Neural Computing and Applications DOI 10.1007/s00521-019-04196-8, URL https://doi.org/10.1007/s00521-019-04196-8

[48] Zemouri R, Zerhouni N, Racoceanu D (2019) Deep learning in the biomedical applications: Recent and future status. Applied Sciences 9(8), DOI 10.3390/app9081526, URL https://www.mdpi.com/2076-3417/9/8/1526

[49] Zemouri R, Lévesque M, Amyot N, Hudon C, Kokoko O, Tahan SA (2020) Deep convolutional variational autoencoder as a 2d-visualization tool for partial discharge source classification in hydrogenerators. IEEE Access 8:5438–5454, DOI 10.1109/ACCESS.2019.2962775

[50] Zhang Z, Jiang T, Zhan C, Yang Y (2019) Gaussian feature learning based on variational autoencoder for improving nonlinear process monitoring. Journal of Process Control 75:136 – 155, DOI https://doi.org/10.1016/j.jprocont.2019.01.008, URL http://www.sciencedirect.com/science/article/pii/S095915241930037X