[Array 11 (2021) 100068](https://doi.org/10.1016/j.array.2021.100068)

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Contents lists available at ScienceDirect |  |
| Array |
| journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal |
|  | | |

Deep learning regularization techniques to genomics data

Harouna Soumarea,b,\*, Alia Benkahlab,1, Nabil Gmatic,1

a The Laboratory of Mathematical Modelling and Numeric in Engineering Sciences, National Engineering School of Tunis, University of Tunis El Manar, Rue B�echir Salem Belkhiria Campus Universitaire, B.P. 37, 1002, Tunis Belv�ed�ere, Tunisia   
b Laboratory of BioInformatics, BioMathematics, and BioStatistics, Institut Pasteur de Tunis, 13 Place Pasteur, B.P. 74 1002, Tunis, Belv�ed�ere, Tunisia c College of Sciences & Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, 31441, Dammam, Saudi Arabia

|  |  |
| --- | --- |
| A R T I C L E I N F O | A B S T R A C T |
| Keywords:  Deep learning  Overfitting  Regularization techniques Dropout  Genomics | Deep Learning algorithms have achieved a great success in many domains where large scale datasets are used. However, training these algorithms on high dimensional data requires the adjustment of many parameters. Avoiding overfitting problem is difficult. Regularization techniques such as L1and L2are used to prevent the parameters of training model from being large. Another commonly used regularization method called Dropout randomly removes some hidden units during the training phase. In this work, we describe some architectures of Deep Learning algorithms, we explain optimization process for training them and attempt to establish a theo- |

retical relationship between L2-regularization and Dropout. We experimentally compare the effect of these techniques on the learning model using genomics datasets.

|  |  |
| --- | --- |
| 1. Introduction | is that the training model fits well the training dataset but looses its |

prediction capacity on unseen datasets.

In the last decade, Deep Learning (DL) algorithms have achieved tremendous success in many domains where large scale datasets are used such as Bioinformatics [2,13,50,62,88,94], Natural Language Processing [5,15,28,47,71], Computer Vision and Speech Recognition [1,4,29,34, 37,56,65].

In this work, we review a class of DL algorithms called Feedforward Neural Network (FNN) [54,68,91], in which information moves in one direction, from input to output through sequential operations called“layers”. These models are the generalization of logistic regression models, both (FNN and logistic regression) are widely used in Bioinfor-matics and Biomedical science to perform classification and diagnosis tasks [8,20,21,24,27,44,48,70,73]. In most cases, we look for a non linear mapping y ¼ fðxÞ between a variable y and a vector of variables x. The form of f depends on the complexity of the studied problem.

Logistic regression defines a low complexity model using a simple non linear mapping from inputs to outputs. Whereas FNN defines a more complex mapping between inputs and their corresponding outputs, thus resulting models have high complexity and flexibility and better pre-diction capacity. However, increasing the complexity of predictive models increases also the risk of overfitting problem, which repercussion

Preventing overfitting problem is one major challenge in training these algorithms. However, there are many techniques that deal with the problem of overfitting called “regularization techniques”. The most used regularization techniques in Machine Learning (ML) community are L1 and L2regularization's [53]. The idea is to prevent the weights of the model from being large by adding a supplementary term to the loss function. The effect of this penalization is to make it so the learning al-gorithm prefers to learn small weights. This method makes models less complex and avoid the risk of overfitting. Another commonly used reg-ularization technique so-called “Dropout”, developed by Hinton et al. [33] consists to randomly remove some neurons (in hidden layers) dur-ing the training phase. This forces the hidden units to extract useful in-formation's from the input data and reduce co-adaptation between hidden units, thus making the model less sensitive to the specific weights of neurons. The Dropout technique allows to train an exponential number of (thinned) Networks in a reasonable time [33]. During the test phase, taking the mean prediction of the different (thinned) Networks is equivalent to test on a single Network with all the hidden neurons [6] (without dropping out any unit). To compensate the fact that the weights are learned under Dropout, the outcome weights of neurons of each

\* Corresponding author. The Laboratory of Mathematical Modelling and Numeric in Engineering Sciences, National Engineering School of Tunis, University of Tunis El Manar, Rue B�echir Salem Belkhiria Campus Universitaire, B.P. 37, 1002, Tunis Belv�ed�ere, Tunisia.

E-mail addresses: [soumare.harouna@enit.utm.tn](mailto:soumare.harouna@enit.utm.tn) (H. Soumare), [Alia.Benkahla@pasteur.tn](mailto:Alia.Benkahla@pasteur.tn) (A. Benkahla), [nmgmati@iau.edu.sa](mailto:nmgmati@iau.edu.sa) (N. Gmati). 1 These autors con[tributed equally, the order of](mailto:soumare.harouna@enit.utm.tn) their names is a[lphabetical.](mailto:Alia.Benkahla@pasteur.tn)

<https://doi.org/10.1016/j.array.2021.100068>  
Received 28 November 2020; Received in revised form 26 March 2021; Accepted 10 May 2021   
Available online 24 May 2021   
2590-0056/© 2021 Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

H. Soumare et al. Array 11 (2021) 100068

hidden layer are multiplied by the Dropout rate of that layer, which is a gain in terms of computation time. However, the quality of this approximation remains little known.

Many theoretical Dropout analyses have been explored [6,23,31,49, 55,58,75,79,81]. Baldi et al. [6] showed how the technique acts as adaptative stochastic Gradient Descent. Wager et al. [79] analyzed Dropout as an adaptive regularizer for Generalized Linear Models (GLMs). Ma et al. [46] attempted to explicitly quantify the gap between Dropout's training and inference phases and showed that the gap can be used to regularize the standard Dropout training loss function.

This paper explains the mathematics behind training DL algorithms and attempts to further establish the theoretical relationship that exists between Dropout and other regularizations, mainly L2norm. We compare experimentally the effects of regularization techniques on training models using two different genomic classification datasets.

The human DNA is a long chain of 3 billion base pairs, the function of a large part of it, is unknown. Some fragments of DNA called genes code for proteins that play important roles in chemical processes essential to life. Some changes in the genes cause a dysfunction in the production of the corresponding proteins, which could cause genetic diseases. The most common genetic changes are called Single Nucleotide Polymorphisms (SNPS) and are caused by a change of a base pair by another one at a given position in the genome. It has been shown that some SNPS are involved in several human diseases and can be used to predict human reponse to certain drugs [27].

In our experiments, we started by using Logistic Regression on cancer datasets, obtained from the Expression Project for Oncology (EXPO) [60]. Then we trained FNN on one 1000 Genomes Project dataset for individual ancestry prediction according to their genetic profile [59]). All in-dividuals are represented in both datasets by their SNPS profile [14].

This work is organized as follows:Section 2 describes the FNN archi-tectures and the mathematics behind them; in Section 3 Gradient descent algorithm is presented; Section 4 describes the traditional regularization techniques and Dropout, Section 5 describes the materials and methods and in Section 6, we present the experimental results, where different regularization techniques are used.

2. Deep Learning: Feedforward Neural Network(FNN)

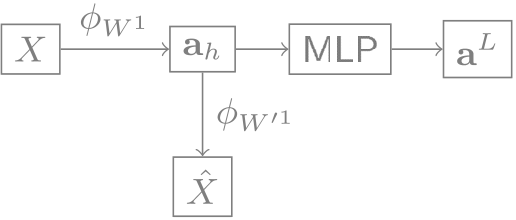
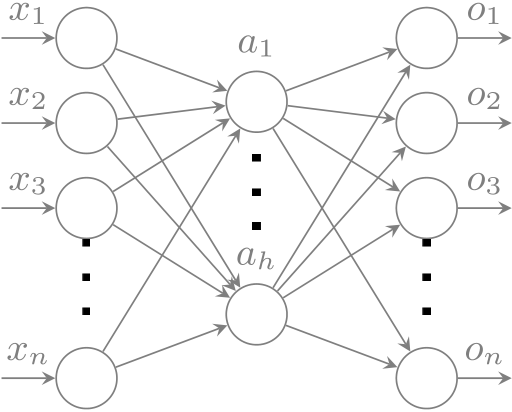
In this work, we discuss Feedforward Neural Network (FNN) [42,54, 57,69,76,91] or Multi-Layer Perceptron (MLP). In such Networks, the in-formation moves only from the input to the output (see Fig. 2), without any loop. This type of model is mostly used for supervised ML tasks such as regression or classification tasks, where the target function is known. The basic supervised learning algorithm is linear regression [12,51,82], in this task the algorithm learns to map an input data x 2 Rdto some real value y, by a linear transformation

f : Rd→ R   
 x → z ¼ x � w þ b:

Where w and b are respectively the weight vector and the bias term. The symbol “�” is the dot product between two vectors. Another simple su-pervised learning algorithm called logistic regression is used for the classification problems where the target function takes discrete values. Given an input data x, the logistic regression [18,19,39,74,86] applies a non linear function to its corresponding linear regression output z, to produce classes membership probabilities. For example, in a binary classification task, given x and it corresponding class C1, logistic regression algorithm outputs the conditional probability PðC1jxÞ of x given C1. This probability is given by sigmoid function σðzÞ ¼ 1þe�z. In the case where there are more than two classes, the conditional probability

|  |
| --- |
| ezi PðCijxÞ is given by the softmax function softmaxðzÞi ¼  ðz1 z2 …zncÞ and zi ¼ x � wi þ bi. wi and bi are respectively the weight vector and bias term of the ith class Ci. nc is the number of classes and the Pnc k¼1ezk. Where z ¼ |

2



H. Soumare et al. Array 11 (2021) 100068

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | |  | |  |  |  | |  |  |  |  | |  | |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  | |  |  |  |



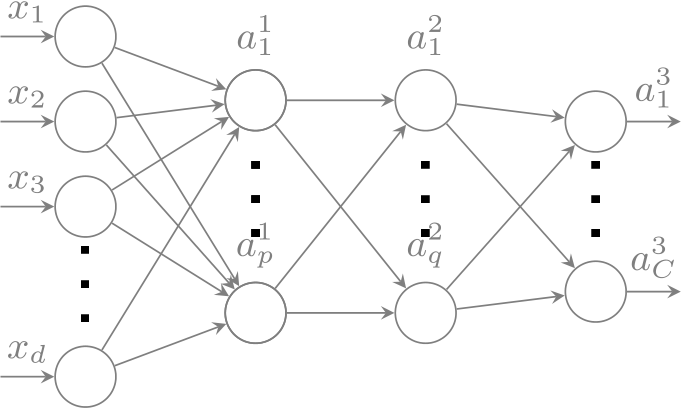


Fig. 2. Classification network.

hidden representation and decoder tries to reconstruct inputs from an encoder, so it contains at least one hidden layer. In an Autoencoder Network, the target yðxÞ of each input sample x is the input it self, i.e. yðxÞ ¼ x, 8x 2 Rd. At the end the output has the size of the input. The main objective of an Autoencoder is to automatically capture the most relevant features from input data. It is also used as a nonlinear dimensionality reduction technique [32,66,80] to transform a high d dimensional data to a lower dimensional data. Mathematically it is defined by the following application:

o : Rd→ Rd

xi → φ W01 ∘ φW1ðxiÞ; 8xi 2 Rd

Where φW1 and φ W01 are the encoding and decoding functions parame-trized by W12 Rd�hand W 01 2 Rh�d respectively and defined as follow:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  | φW1 : | Rd→ Rh  xi → ahðxiÞ; | | φ 0  W01 : | Rh→ Rd | : |
|  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| ahðxiÞ → oðxiÞ |
| Where ah and o are, respectively, the hidden and output layers output | | | | | |
| vectors. The parameters ðW1; W reconstruction error between the input and the output of Network 01Þ are learned by minimizing the | | | | | |
| L ¼1 2n | X | kxi � φ W01 ∘ φW1ðxiÞk2 2: | | | |

After Autoencoder training, the decoding layers are removed and the   
encoding layers are retained and the learned matrix W1is then used as

|  |  |  |
| --- | --- | --- |
| Fig. 3. Autencoder | parameters of the first layer(s) of the supervised Network (see Fig. 3). | |
| Alternatively, the couple (W1;W | 0 1) can be learned jointly(see Fig. 4) with |

Supposing that the couples ðxi; yiÞ, i 2 f0; …; 1g are independent, the likelihood function is given

P yjXÞ ¼ �Y Pðy ¼ yijX ! (4)

¼ �Y aLðxiÞyi�1 � aLðxiÞ1�yi� : (5)

Training the NN consists to maximize the likelihood function which is

equivalent to minimize the crossentropy loss function defined by

C ¼ �1 X yilog aLðxiÞ þ ð1 � yiÞlogð1 � aLðxiÞÞ: (6)

Consider now, a multi-class classification problem, where the labels

are mutually exclusive. In this case, (6) takes the form

C ¼ �1 X X ykðxiÞlog aL kðxiÞ: (7)

aL

1. In the rest of this work, C denotes the loss function defined by (3). kðxiÞ is the softmax function satisfying 0 � aL kðxiÞ � 1 and Pnc k¼1aL kðxiÞ ¼

2.2. Unsupervised Neural Network(Autoencoder)

So far, we have described FNN in the supervised learning case. Here,

we suppose that input samples X ¼ fx1; x2; …; xng are unlabeled, where

xi 2 Rd. Autoencoder is one the most used unsupervised learning algo-rithms [41,52,77,83,93]. An Autoencoder is a NN designed to learn an

identity function in a way that the original input can be reconstruct from

a compressed version. Such a network will allow the discovery of a more

efficient and compressed representation of the input data. It consists of

two parts, an encoder and a decoder. Encoder maps input samples to a

3

H. Soumare et al. Array 11 (2021) 100068

Descent (SGD) method.

3.1. Stochastic gradient descent

The idea of stochastic gradient descent [9,10,40,89] is to estimate at each iteration partial derivatives for only a small randomly chosen sample Xm ¼ fx1; x2; …; xmg called mini-batch and train with it.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | wl | → wl | α | | X | ∂Cx | : | (9) |
| |  |  |  | | --- | --- | --- | | ij | ij� | m | | X | ∂wl ij |  |  |

We then take another randomly chosen mini-batch and the weight pa-rameters are updated on it, until the training inputs are exhausted, which is called an epoch of training. At this point, we start again with a new epoch. To compute the partial derivatives,∂Cx∂wl ijat each layer, we apply the chain rule:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ∂Cx | δl x ∂zl | δl xal�1 | ðxÞ: | |
| ∂wl ij | |  |  |  | | --- | --- | --- | | ¼ | jðÞ | ij | | ¼ jðÞ |
| Where δl j¼ ∂Cx jrepresent the error function of j neuron in the lth layer, for | | | | |
| an input x. For the sake of simplicity, we just write δl jand al�1 | | | | instead of |

δl jðxÞ and al�1 ðxÞ. This expression tells us how a little change in the weighted input to the jth neuron in layer l changes the overall behavior of the loss function. The backpropagation algorithm is used to compute δl j for each layer.

means that errors are computed backwards, hence the name back-propagation. By writing partial derivatives,∂Cx∂wl ijwith respect to δl j, the

gradient descent updating rule is rewritten

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| wl ij→ wl ij� α m | X | X | δl jðxÞal�1 | ðxÞ: | (12) |

Where nh is the number of neurons in the lth layer. Typically, in DL al-gorithms, the SGD algorithm is combined with backpropagation, where we have to compute the gradient of a loss function, to be minimized for a large set of data. The implementation of this algorithm is done in a few steps:

1. Provide a set of training examples

|  |
| --- |
| 2. For each example x:give a1ðxÞ, and perform the following steps: ● Do a Feedforward: For l ¼ 2; 3; …; L compute zlðxÞ ¼ Wlal�1ðxÞ þ blwith alðxÞ ¼ φðzlðxÞÞ.  ● Output error function δL:Compute δLðxÞ ¼ rCx � φ 0ðzLðxÞÞ.  ● Backpropagate the error: For l ¼ L � 1; L � 2; …; 2 compute δlðxÞ ¼ ððWlþ1ÞTδlþ1ðxÞÞ � φ 0ðzlðxÞÞ  3. Gradient descent:For l ¼ L; L � 1; …; 2 update the weights according to the formula Wl→ Wl�α  with small computations that the update formula for the vector bl m P x2XmδlðxÞðal�1ðxÞÞT. We can also show  containing the bias terms in any l layer is written:bl→ bl�α  m P x2XmδlðxÞ  To implement stochastic gradient descent in practice, an external loop |

|  |  |
| --- | --- |
| 3.2. Backpropagation   Backpropagation [30,84,85] is a widely used algorithm in minimizing | generating mini training example runs, and an external loop running through several training epochs are required. However, these were omitted for simplicity. |

Feedforward Neural Network loss functions. It uses the chain rule to

compute iteratively the error of each neuron in a Network, from the

output to the input layer.

Errors at the output layer: Let's begin by computing δL j; i 2 f1;…;cg,

errors of neurons in the last layer L. By using the chain rule, we have

|  |  |  |
| --- | --- | --- |
| δL j¼ ∂Cx j | ∂aL j  ∂zL j | (10) |
| ¼�yj � aL�φ 0�zL j�  Because the loss function depends on zL j, through aL jonly. | |

Errors at any hidden layer: error δl jof any hidden neuron j at any

layer l. The weighted input zl jof a hidden layer l is linked to the loss

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| function through all weighted inputs ðzlþ1 | | | | | | Þk to the next layer. |
| δl j¼ | X∂zlþ1 k  ∂Cx | | | ∂zlþ1 k | |
| ∂zl j | |  |
| ¼ | X δlþ1 k | | ∂zlþ1 k | | : |
| ∂zl j | |  |
| Using the chain rule, we have | | | | | |
| ∂zlþ1 k | | ¼∂zlþ1 j | | ∂alþ1 k | |
| ∂zl j | | ∂zl j | |  |

4. Regularization techniques

One of the most serious problems in training ML models, particularly for NN, is overfitting. This problem occurs when a training model is too complex.

4.1. L1and L2regularization techniques

A widely used technique to reduce a model complexity is to add a regularization term [26] to the loss function C. The new model loss function Cλ is defined as follows:

Cλ ¼ C þ λΩðWÞ

These, update the general cost function by adding another term known as the regularization term, where Ω is L1or L2norm and w is the NN weight parameters.

4.1.1. L2regularization   
 The L2regularization term, commonly known as weight decay. The idea of this technique also known as ridge regression or Tikhonov regularization [78], is to add a L2term to the function to be minimized, in this case ΩðWÞ ¼1 2kWk2 2. This added term in L2 norm imposes the weights to live in a sphere of radius inversely proportional to the regu-larization parameter [ [26], p. 249] λ. In this context, the updating rule,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ¼ wlþ1 jkφ 0�zl | | �: | | using gradient descent strategy becomes | | | | | | |
| wl ij→ | � | |  |  |  | | --- | --- | --- | | 1αλ | wl | α | | X | ∂Cxi | : | (13) |
| δl j¼ φ 0�zl j | �X wlþ1 jkδlþ1 k: | | (11) | � | |  |  |  | | --- | --- | --- | | � | � | ij� | | X | ∂wl ij |  |
| this means that, after each iteration, the weights are multiplied by a factor slightly smaller 1. It tends to force the model to prefer small weights. | | | | | | |
| The above expression tells us that error functions at any hidden layer are given by the weighted sum of the error functions at the next layer. Which | | | |

4

H. Soumare et al. Array 11 (2021) 100068

4.1.2. L1regularization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| L1regularization modifies the loss function by adding a L1term, i.e. ΩðWÞ ¼P loss function by removing the irrelevant features from the training w2Wjwj. The idea behind this technique is to regularize the  model. In this situation, the updating rule is written | | | | | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | wl | → wl | αλ | sgn | wl | | � | α | X | ∂Cxi | : | (14) |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | ij | ij� | n | � | ij | | � | � | X | ∂wl ij |  |  |

Where sgnðwl ijÞ is the sign of wl ij. Both types of regularization try to penalize the big weights when it's necessary by shrinking them after each updating step, but the way of shrinkage is different [26]. When L2reg-ularization is used, the weights are shrunk by an amount proportional to wl ij, whereas in L1 regularization, the weights are shrunk by constant quantity toward to zero. As shown in Fig. 5 (graph on the left), in a two dimensional space, the L1norm defines a parameter space bounded by a parallelogram at the origin. In this case, the loss function is likely to hit the vertices of the parallelogram rather than its edges. L1regularization removes some of the parameters, thus L1technique can be used as a feature selection technique. On the other hand, the L2regularization defines a circle whose radius size is inversely proportional to the regu-larization parameter (see Fig. 6).

In this work, without any assumption of input data, we quantify explicitly the gap and then show how it related to L2regularization.

4.2.1. Dropout application to linear networks   
 To see more clearly the relationship between L2-regularization, we start by studying the problem in a very simple case, where all activation functions in the model are linear. Consider a NN, where all units are linear (i.e. al¼ al�1Wl, where aland Wlare the output vectors and weight matrix of layer l 2 f1; …; Lg respectively). The Dropout NN loss function is

|  |
| --- |
| n X���yðxÞ � aL�1ðxÞ ~W L���2þ1 � pL�1���ΣL�1 ~W L���2: (16) 1 2  yðxÞ is the output vector given an input vector x. ΣL�1¼  �ndiagðaL�1ðXÞðaL�1ðXÞÞTðXÞÞ�1 ,~W L ¼ pL�1WL. Given a matrix A, we  denote by diagðAÞ, a diagonal matrix with the same size and diagonal elements as A. |

At each layer l, we define a matrix alðXÞ whose columns correspond to the values taken by the vector of the activation function alacross input data: alðXÞ ¼ ðal iðxjÞÞ, 1 � i � m, 1 � j � n, where al iðxjÞ is the ith output neuron in (l)th layer of the jth input and m is the number of neurons in the layer.

|  |  |  |
| --- | --- | --- |
| 4.2. Dropout technique | Proof. | Training a standard nn without dropping neurons is done by |

In training a NN, Dropout technique regularizes learning by dropping out some hidden units with certain probability. This is equivalent to modifying [72] the NN by setting some hidden activation functions to zero. Using Dropout, we can formally define the NN as follows:

~al j¼ δl jal j; (15)

At each neuron j, in a hidden layer l, the output activation al jis multiplied by a sampled variable δl j, to produce thinned output activations ~al j. These thinned functions are then used as inputs to the next layer and the same process is applied at each layer. This application is equivalent to sampling a sub Neural Networks from a larger network. Where δl   
jis a Bernoulli random variable (δl j↪ Bernoulli(pl)) of parameter pl, i.e. a neuron in the lth layer is kept with a probability of pland removed with a probability

minimizing the following loss function:

|  |
| --- |
| n X��yðxÞ � aL�1ðxÞWL��2 (17)  1  Dropout modifies the training process and the loss function in (17) |

becomes

|  |
| --- |
| n X EδL�1  ��yðxÞ ��δL�1ðxÞ � aL�1ðxÞ�WL��2 2: (18)  1  Where δL�1is a random vector of the layer L � 1 with δL�1↪  BernoulliðpL�1Þ and � denotes the Hadamard product. Using the formula  EðX2Þ ¼ ðEðXÞÞ2þ VarðXÞ for a random variable X, we show that (18) is equal to |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | yx E | δL�1x aL�1x | WL2 | 1 | Var | δL�1x aL�1x | | WL |
| n | |  |  | | --- | --- | | X��ðÞ �δL�1��1 ðÞ �ðÞ�  n X��yðxÞ � pL�1aL�1ðxÞWL��2 2þ1 | ���2þ X �� ðÞ �ðÞ�  X WLVar  �δL�1ðxÞ � aL�1ðxÞ�ðWLÞ T: | | ¼ |

1 � pl. Srivastava et al. [72] suggested that, applying Dropout to a NN with n units can be seen as sampling 2nsub Networks with weight sharing. In the test phase, as it is not always practical to take the mean of 2nmodels, an approximate averaging method is used. The idea is to approximate the exponentially many Networks by a single NN without Dropout. To correct the fact that training outgoing weights of a layer are obtained under condition that neurons were retained with a probability p, the weights are simply multiplied by p. This approximation has been proved for lo-gistic and linear regression models [72,79]. But, for Deep Neural Net-worksDNNs, there is an unknown gap between the expected output of exponential sub Networks and the output of a single deterministic model. Ma et al. [46] showed that under some assumptions on input data, the gap is controlled and it can be used to regularize the single NN.

5

H. Soumare et al. Array 11 (2021) 100068

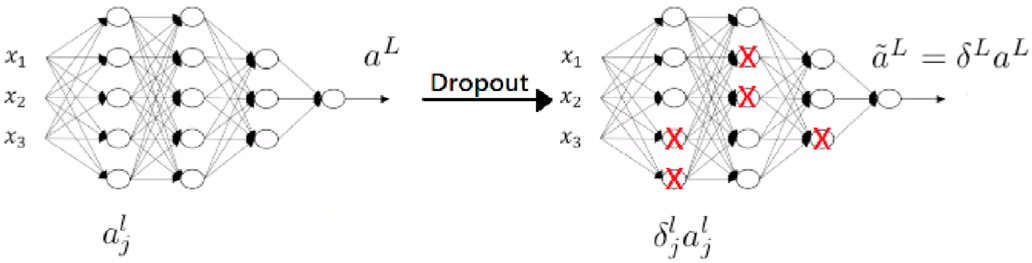


Fig. 6. Dropout.

|  |
| --- |
| As VarðδL�1ðxÞ �aL�1ðxÞÞ ¼ �1�pL�1 pL�1�aL�1ðxÞaL�1ðxÞT, we obtain the  desired result. Under the assumption that input layers follow a |

Gaussian distribution with standard deviation σ, Dropout is equivalent in expectation to L2-regularization. The regularization parameter λ is a

|  |
| --- |
| function of1�pL�1 pL�1 σ2 which increases (resp. decreases) with the variance of  input layers σ2(resp. with pL�1). Thus, Dropout regularization consists of |

detecting the inputs with more variance and shrink their weights.

4.2.2. Dropout application to non linear networks   
 Here, we try to generalize the relationship between Dropout and L2-regularization to Networks with nonlinear units. Consider a NN with a non linear activation function, i.e., al¼ φðal�1WLÞ. Dropout training expected loss function is given by

which the weights are scaled by pL�1to compensate the fact that they are learned under conditions in which 1 � pL�1of hidden units where dropped out. In this case, Dropout training model can be seen as an L2-regularization where, the regularizer λ depends on: Dropout rate; the variance of each input and output layer.

4.2.3. Dropout with others regularization techniques   
 Dropout is known to improve training model performance when it is combined with other regularization techniques. Batch normalization, introduced by Ref. [35], is a regularization technique used to speed up the training and improve performance of Deep NNs. In the training of a DNN, the distribution of each layer's inputs change, as the parameters of all layers that come before it, variate. This can slow down by requiring small learning rates and careful parameter initialization. Given a batch of sample used to update parameters, batch normalization normalizes the inputs of each layer by recentering and rescaling (subtracting the mean and dividing by the batch standard deviation). Thus, batch normalization prevents layers inputs to have large standard deviations [35]. show experimentally that batch normalization with large learning rate, speed up significantly training as it can eliminate the need for Dropout. In fact, as discussed in Section ??, Dropout look for layer's inputs with more varitions and shrink their weights, this function of shrinking is then largely reduced by batch normalization application. Combining dropout

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | �yx φ�~W | LT | |  |  | | --- | --- | | aL�1x | �� |   2 | 1 | |  |  |  | | --- | --- | --- | | �1 � p� | φ00 | aL�1x~ WL | | ΣL�1 | WL2 : | (19) |
| n | X���ðÞ � � | � | ðÞ ���2 | þ 2n | X k� ðÞ | � | k 2 |  |

Where ΣL�1 ¼�aL�1ðxÞðaL�1ðxÞÞT�1 2 and ~W L ¼ pL�1WL.

Proof. We know that a non linear Dropout Network training loss is

defined as

|  |  |
| --- | --- |
| n X EδL�1  ��yðxÞ � φ��δL�1ðxÞ � aL�1ðxÞ�WL���2 2:  1  Using triangle inequality, (20) is bounded by | (20) |

with batch normalization [25,35,43] can improve DNNs prediction accu-racy. Dropout regularization is known to give a significant improvement when it is combined with others regularization methods such as max-norm [72] and weight normalization [67]. Rather than constraining whole weight matrix of each layer as in L2regularization, these constrain each column of the weight matrix to prevent separately any hidden neuron from having very large weights. Max-norm regularization con-sists to constrain the incoming weight vector of each hidden neuron to live in a ball of radius c, where c is a hyper-parameter. Weight normal-

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | |  |  |  | | --- | --- | --- | | yx φ | pL�1aL�1xWL2 | 1 | | E | |  |  |  | | --- | --- | --- | | φ | aL�1x δL�1x | WL | | φ pL�1aL�1xWL2 |
| n | |  |  |  | | --- | --- | --- | | X��ðÞ � | � | ðÞ���2þ | | X��δL�1ðxÞ | |  |  |  |  | | --- | --- | --- | --- | | � | �� | ðÞ �ðÞ� | �� | | |  |  |  | | --- | --- | --- | | � | � | ðÞ� ��  2 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ��Now, by applying a second order Taylor expansion of φ around EδL�1 δL�1ðxÞÞWL� pL�1aL�1ðxÞWL, we have φððaL�1ðxÞ �δL�1ðxÞÞWLÞ ¼  φðpL�1aL�1ðxÞWLÞ þ φ  ðpL�1aL�1ðxÞWLÞVarðZZTÞ. Because Z is centered i.e., EδL�1ðxÞðZÞ ¼ 0. Then EδL�1ðxÞ�φððaL�1ðxÞ �δL�1ðxÞÞWLÞ   0ðpL�1aL�1ðxÞWLÞZ þ 1  � 2φ00ðpL�1aL�1ðxÞWLÞZZT.  � φðpL�1aL�1ðxÞWLÞ ¼ 1 2φ00  Thus, an upper bound of (20) is given by | | | | | |
| 1 | yx φ aL�1x~ WL | 1 | �1 � pL�1 | |  |  |  | | --- | --- | --- | | � | φ00 | aL�1x~ WL | | �Σx~ WLk2 2: |
| n | |  |  |  | | --- | --- | --- | | X���ðÞ � | � | ðÞ | | ����2þ 2n |  | X k� ðÞ |
| Here again, Dropout can be seen as regularizer, where the regularizer represents the gap between EδL�1 output of exponential thinned Networks produced by applying Dropout�φððaL�1�δL�1ÞWLÞ� the expected  and φðpL�1aL�1WLÞ, the output of a single deterministic Network, in | | | | | |

6

|  |  |  |
| --- | --- | --- |
| H. Soumare et al. | Table 2 | Array 11 (2021) 100068 |
| the problem complexity. Stochastic gradient descent is adopted in all |

experiments as an optimization strategy. Two types of datasets are included in our experiments, Expression Project for Oncology (expO) cancer datasets and 1000 Genomes Project ethnicity datasets respectively used for training logistic regression and FFN models.

Individuals in selected datasets are humans that are represented by

Unregularized logistic reg.

|  |  |
| --- | --- |
| Dataset | Accuracy (in %) |
| Breast-Kidney  Colon-Kidney  Breast-Colon | 96.53  97.82  94.13 |

|  |  |  |
| --- | --- | --- |
| the list of their SNPS. Each SNP is represented by its genotype (i.e. genetic | Colon-Prostate | 97.46 |

information) at a specific locus. In a diploid organism at each locus, there are two copies of alleles, one comes from the father and other from the

|  |  |
| --- | --- |
| mother. Consequently, a genotype takes one of three values for a diploid organism: 0 (homozygous reference), 1 (heterozygous) and 2 (homozy- | Table 3  Logistic reg. with L1norm. |

gous alternate). The homozygous reference refers to the base that is found in the reference genome, an homozygous alternate refers to any base, other than the reference, that is found at that locus and genotype is said heterozygous at a given position, when the two alleles are different.

of individuals included in the study and d correspond to the number of The input of a model is a matrix X of size n � d, where n is the number

features (SNPS). The output y takes discrete value(s) between 0 and 1.

5.1. Expression Project for Oncology(expO) cancer datasets

The different cancer samples included in this study (see Table 1), are downloaded from Ref. [11]. The original datasets can be obtained from the Expression Project for Oncology (expO) that was deposited at Gene Expression Omnibus (GEO) repository [7], with accession number GSE2109. The objective of expO is to obtain and perform gene expression analysis on cancer tissue samples and assemble the patient's long term clinical results.

5.2. 1000 Genomes Project dataset

The 1000 Genomes Project [16] took advantage of developments in

|  |  |  |
| --- | --- | --- |
| Dataset | Regularization L1 | Accuracy(in %) |
| Breast-Kidney | λ ¼ 10�2  λ ¼ 10�3  λ ¼ 10�2  λ ¼ 10�3  λ ¼ 10�2  10�3 | 97.36 |
| Colon-Kidney | 99.01 |
| 95.82 |
| Breast-Colon | 97.45 |
| 94.44 |
| 93.17 |
| Colon-Prostate  Table 4  Logistic reg. with L2no | λ ¼ 10�2  λ ¼ 10�3  rm. | 98.59 |
| 98.03 |

|  |  |  |
| --- | --- | --- |
| Dataset | Regularization L2 | Accuracy(in %) |
| Breast-Kidney | λ ¼ 10�2  λ ¼ 10�3  λ ¼ 10�2  λ ¼ 10�3  λ ¼ 10�2  λ ¼ 10�3  λ ¼ 10�2  λ ¼ 10�3 | 98.18 |
| Colon-Kidney | 98.02 |
| 98.18 |
| Breast-Colon | 98.91 |
| 92.86 |
| Colon-Prostate | 99.44 |
| 97.18 |
| 96.62 |

|  |  |
| --- | --- |
| Next-generation sequencing (NGS), which allows to sequence DNA and RNA much more quickly and cheaply. It's the first project to sequence the | Table 5  MLP accuracy vs its size. |

genomes of a large number of people in populations from different re-

gions and countries. In this study, n ¼ 3450 is the number of individuals sampled worldwide from 26 populations and d ¼ 315345 is the number of SNPS. The desired output of the model is a vector Y 2 Rc, whose components correspond to the 26 classes of populations (i.e. c ¼ 26). The model consists of an input layer, an output layer and two hidden layers of

equal size. Given the input matrix X, the model output is a vector a32 Rc. A reluaction function is used in the two hidden layers followed by a

sotmax layer to perform ancestry prediction.

|  |  |
| --- | --- |
| # of units by hidden layer | Accuracy(in %) |
| [50]  [50-50]  [100]  [100�100]  [100-100-100]  [500-500-500] | 81.33  81.68  90.68  92.70  90.49  90.46 |

6.2. Ancestry prediction using a multilayer perceptron (MLP)

|  |  |
| --- | --- |
| 6. Experiments | In this subsection, FNN is used on 1000 Genome Project ethnicity |

In this section, we present the effects of regularization techniques on training models for different datasets (see Table 1).

6.1. Cancer dataset classification using logistic regression

Un-regularized logistic regression results are reported in Table 2, despite its simplicity, logistic regression model gives good classification accuracy on these cancer datasets. To improve prediction capacities of the present model, a penalty term is added and obtained results are presented in Table 3 and Table 4 for L1and L2regularization added term, respectively. We can observe from theses table that penalization with the appropriate regularization parameter improves the classification accu-racy. For example, when L1regularization is used with regularizer λ ¼ 10�3, the classification accuracy on Breast-Kidney dataset goes from 96:53 to 99:01. Similarly, when L2penalization is applied(λ ¼ 10�3), the prediction accuracy on Breast-Colon dataset increases from 94.44 in unregularized case to 99:44.

7

H. Soumare et al. Array 11 (2021) 100068

Table 6   
Prediction accuracy for Dropout, batch normalization and dropout combination with batch normalization.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # of uni. by hid. layer | Drop.(p¼0.2) | Drop.(p¼0.5) | Bat.norm | Drop.(p¼0.2)þBat. norm | Drop.(p¼0.5)þBat. norm |
|  | (accuracy in%) | (accuracy in%) | (accuracy in%) | (accuracy in%) | (accuracy in%) |
| [50] | 90.32 | 87.54 | 90.58 | 91.48 | 92.61 |
| [50-50] | 89.94 | 40.96 | 91.01 | 92.58 | 92.93 |
| [100] | 88.81 | 92.26 | 90.75 | 91.65 | 93.01 |
| [100�100] | 90.32 | 64.12 | 89.19 | 92.46 | 93.00 |

Table 7   
Prediction accuracy for L1, L2, Dropout regularization techniques and Dropout combination with others regularization techniques.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # of uni. by hid. layer | L1(λ ¼ 10�4) reg. | L2(λ ¼ 10�3) reg. | Drop.(p¼0.5)þBat. norm | Drop.(p¼0.5)þMa.norm | Drop.(p¼0.2)þUn.norm |
| (accuracy in%) | (accuracy in%) | (accuracy in%) | (accuracy in%) | (accuracy in%) |
| [50] | 92.13 | 92.61 | 92.61 | 92.35 | 93.25 |
| [50-50] | 91.77 | 90.75 | 92.93 | 83.68 | 90.32 |
| [100] | 92.70 | 92.70 | 93.01 | 93.83 | 94.43 |
| [100�100]  [100-100-100] | 92.29 | 93.39 | 93.00 | 90.17 | 91.94 |
| 90.87 | 91.01 | 92.42 | 89.68 | 92.19 |

6.2.2. Classification with regularization   
 When Dropout is used alone, the choice of its rate p is very important as in Table 6, we have to be more careful about it. Combining Dropout with batch normalization improves the performance training model and makes the choice of p less important. L1and L2regularization, give good prediction accuracy and outperform Dropout as observed in Table 7. However, when Dropout is combined with batch normalization, max-norm and unit-norm outperforms the traditional regularization tech-niques. We have obtained our best prediction accuracy 94.43%, when Dropout is combined with unit norm constraint.

that Dropout is less effective than L1and L2regularization techniques [58] when the training model is not complex (as for our model). Combining Dropout with techniques such as Batch normation, Unit norm constraint or Max norm enabled the training model to achieve its best accuracy. The obtained results are compared to the results in Table 9 obtained by Ref. [64] on the same dataset. In Ref. [64], the authors proposed auxiliary NNs to predict the parameters of the first hidden layer of the classification NNs and different features embedding techniques such as Random projection(RP), Per class histogram, and SNPtoVec have also been proposed. Achieving such prediction accuracy obtained with SNP data, these regularization techniques will allow us to face more

7. Discussion complicated problems in many domains such as preventive medicine.

Regularized Logistic Regression has achieved good results compared 8. Conclusion

to previous machine learning approaches tested on the same cancer samples [70,73]. For instance, Stiglic et al. [73] combined different feature selection methods such as Vector Machines Recursive Feature Elimination (SVM-RFE) and ReliefF followed by SVM or k-nearest neighbors. To the best of our knowledge, the best results on these data-sets were reported in Ref. [70], where the authors used Stacked Sparse Autoencoders (SSAE) to select most relevant features followed by a classification Neural Network to categorize the samples. Despite its

In this work, we have explained stochastic gradient descent optimi-zation technique with back-propagation in training DL algorithms. To prevent overfitting problem, regularization techniques are studied and, theoretical relationship between Dropout and L2regularization is established. Experimental results have shown that Dropout, when it is combined with techniques such as batch normalization, max-norm or unit-norm gives better performance than L1and L2regularization

simplicity, the proposed approach outperforms SSAE on datasets such as techniques.

Breast-Kidney and Breast-Colon (see Table 8). In Stacked Sparse Autoencoders [36,45] many Autoencoder layers are stacked together to form an unsupervised learning algorithm, where the encoder layer computed by an Autoencoder will be used as the input to another Autoencoder layer. In practice, a logistic regression model may be better than a NN for relatively small data sets and simple classification tasks, where the classes are more or less linearly separable. Indeed, the latter are more difficult to train, require more training samples and are more prone to overfitting than logistic regression.

Logistic regression application to anscentry prediction dataset lead to

For future work, we expect to further study these regularization techniques in DNN and use them to analyze gene expression profile data with the aim of predicting rare diseases.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

poor prediction accuracy, which is due to the large dimension of input Acknowledgments

features and high nonlinear correlation between them, and to the genetic

similarity between some ethnic of groups of populations.

Training a NN with an Autoencoder reconstruction path improved the

This project was partly funded by H3ABioNet, which is supported by the National Institutes of Health Common Fund under grant number

results. However, training an Autoencoder in conjunction with the clas- U41HG006941.

sification Network makes the high dimensional optimization problem   
more difficult to solve than simply training the classification Network, Appendix yielding in a higher classification error. To improve the results, regula-

rization techniques are used. One can notice that traditional regulariza-tion technique's application has more improved the prediction accuracy of the model compared to Dropout. This could be attributed to the fact

8

H. Soumare et al. Array 11 (2021) 100068

Table 8   
Logistic reg.vs SSAE.

|  |  |  |
| --- | --- | --- |
| Dataset | Stacked Sparse Autoencoders | Logistic regression |
| Breast-Kidney | 98.4 | 99.01 |
| Colon-Kidney | 99.5 | 98.91 |
| Breast-Colon | 97.3 | 99.44 |
| Colon-Prostate | 99.7 | 98.59 |

Table 9   
Reported results in Ref. [64].

|  |  |  |
| --- | --- | --- |
| Model & Embedding | Mean Misclassif. Error. | # of free |
| (%) | param. |
| Basic | 8:31 � 1:83  8:88 � 1:41  9:03 � 1:20  7:60 � 1:28  7:88 � 1:40  7:76 � 1:38  8:28 � 1:92  8:03 � 1:0:3 | 31.5 M |
| Raw end2end | 21.27K |
| Random Projection | 10.1K |
| SNP2Vec | 10.1K |
| Per class histograms | 7.9K |
| Basic with reconstruction | 63 M |
| Raw end2end with reconstruction | 227.3K |
| Random Projection with | 20.2K |
| reconstruction |
| SNP2Vec with reconstruction | 7:88 � 0:72  7:44 � 0:45 | 20.2K |
| Per class histograms with | 15.8K |
| reconstruction |

[19] [Dreiseitl S, Ohno-Machado L. Logistic regression and artif](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref19)i[cial neural network](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref19)

[20] Fakoor R, Ladhak F, Nazi Z, Huber M. Using deep learning to enhance cancer [classif](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref19)i[cation models: a methodology review. J Biomed Inf 2002;35:352](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref19)–[9](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref19).

diagnosis and classification. In: Proceed. of the inter. conf. on ML. New York, USA: ACM; 2013. <https://doi.org/10.1109/ICSCAN.2018.8541142>.

[21] Fort G, Lambert-Lacroix S. Classification using partial least squares with penalized logistic regression. Bioinformatics 2005;21:1104–11. [https://doi.org/10.1093/ bioinformatics/bti114](https://doi.org/10.1093/bioinformatics/bti114).

[22] [Fu X, Wei Y, Xu F, Wang T, Lu Y, Li J, Huang JZ. Semi-supervised aspect-level](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref22)  [sentiment classif](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref22)i[cation model based on variational autoencoder. Knowl Base Syst](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref22)

[23] [Gal Y, Ghahramani Z. Dropout as a bayesian approximation: representing model](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref23) [2019;171:81](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref22)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref23)[92](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref22).

[uncertainty in deep learning. In: International conference on machine learning.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref23)

[24] Ganesan N, Venkatesh K, Rama MA, Palani AM. Application of neural networks in [PMLR; 2016. p. 1050](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref23)–[9](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref23).

diagnosing cancer disease using demographic data. Int J Chem Appl 2010;1:76–85. <https://doi.org/10.5120/476-783>.

[25] Garbin C, Zhu X, Marques O. Dropout vs. batch normalization: an empirical study of their impact to deep learning. Multimed Tool Appl 2020:1–39. [https://doi.org/ 10.1007/s11042-019-08453-9](https://doi.org/10.1007/s11042-019-08453-9).

[26] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, vol. 1. MIT press Cambridge; 2016. <https://doi.org/10.1007/s10710-017-9314-z>.

[27] Group, I.S.M.W., et al.. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. Nature 2001;409:928. [https://doi.org/ 10.1038/35057149](https://doi.org/10.1038/35057149).

[28] Guo J, He H, He T, Lausen L, Li M, Lin H, Shi X, Wang C, Xie J, Zha S, et al. Gluoncv and gluonnlp: deep learning in computer vision and natural language processing.

[29] Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, Prenger R, Satheesh S, J Mach Learn Res 2020;21:1–7. 1907.04433.

Sengupta S, Coates A, et al. Deep speech: scaling up end-to-end speech recognition. arXiv preprint arXiv, 1412.5567; 2014. <https://arxiv.org/abs/1412.5567>.

[30] Hecht-Nielsen R. Theory of the backpropagation neural network. In: N. netw. for

|  |  |
| --- | --- |
| References | percep. Elsevier; 1992. p. 65–93. <https://doi.org/10.1016/B978-0-12-741252-8.50010-8>.  [31] [Helmbold DP, Long PM. Surprising properties of dropout in deep networks. In:](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref31) |

[1] Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access 2018;6:14410–30. [https://doi.org/10.1109/ ACCESS.2018.2807385](https://doi.org/10.1109/ACCESS.2018.2807385).

[2] Almagro Armenteros JJ, Sønderby CK, Sønderby SK, Nielsen H, Winther O. Deeploc: prediction of protein subcellular localization using deep learning. Bioinformatics

[3] Amari S. Backpropagation and stochastic gradient descent method. 2017;33:3387–95. <https://doi.org/10.1093/bioinformatics/btx431>.

Neurocomputing 1993;5:185–96. <https://doi.org/10.1016/0925-2312(93)90006-O>.

[4] Amodei D, Ananthanarayanan S, Anubhai R, Bai J, Battenberg E, Case C, Casper J, Catanzaro B, Cheng Q, Chen G, et al. Deep speech 2: end-to-end speech recognition in English and Mandarin. In: Inter. conf. on ML; 2016. p. 173–82. [https://arxiv .org/abs/1512.02595](https://arxiv.org/abs/1512.02595).

[5] Bacchi S, Oakden-Rayner L, Zerner T, Kleinig T, Patel S, Jannes J. Deep learning natural language processing successfully predicts the cerebrovascular cause of transient ischemic attack-like presentations. Stroke 2019;50:758–60. [https:// doi.org/10.1161/STROKEAHA.118.024124](https://doi.org/10.1161/STROKEAHA.118.024124).

[6] Baldi P, Sadowski PJ. Understanding dropout. In: NIPS; 2013. p. 2814–22. [https:// doi.org/10.1016/j.artint.2014.02.004](https://doi.org/10.1016/j.artint.2014.02.004).

[7] [Barrett T, Edgar R. [19] gene expression omnibus: microarray data storage,](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref7)  [submission, retrieval, and analysis. Meths. in enzy. 2006;411:352](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref7)–[69. doi: S0076687906110198](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref7).

[8] Bilen M, Is¸ik AH, Yi�git T. A hybrid artificial neural network-genetic algorithm approach for classification of microarray data. In: 2015 23nd SPCA conf. (SIU),

[9] [Bottou L. Stochastic gradient learning in neural networks. Proceedings of Neuro-](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref9)IEEE; 2015. p. 339[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref9)42. <https://doi.org/10.1109/SIU.2015.7129828>.

[Nımes 1991;91:12](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref9).

[10] [Bottou L. Stochastic gradient descent tricks. In: Neural networks: tricks of the trade.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref10)  [Springer; 2012. p. 421](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref10)–[36](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref10).

[11] Cancer D. Openml; 2015. [https://www.openml.org/search?type](https://www.openml.org/search?type&equals;data)¼[data](https://www.openml.org/search?type&equals;data). [Accessed 7 September 2020].

[12] [Chatterjee S, Hadi AS. Sensitivity analysis in linear regression, vol. 327. John Wiley](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref12)  & [Sons; 2009](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref12).

[13] Chen Y, Li Y, Narayan R, Subramanian A, Xie X. Gene expression inference with deep learning. Bioinformatics 2016;32:1832–9. [https://doi.org/10.1093/ bioinformatics/btw074](https://doi.org/10.1093/bioinformatics/btw074).

[14] Collins FS, Brooks LD, Chakravarti A. A dna polymorphism discovery resource for research on human genetic variation. Geno. research 1998;8:1229–31. [https:// doi.org/10.1101/gr.8.12.1229](https://doi.org/10.1101/gr.8.12.1229).

[15] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceed. of the 25th inter. conf. on ML;

[16] Consortium GP, et al. A map of human genome variation from population-scale 2008. p. 160–7. <https://doi.org/10.1145/1390156.1390177>.

sequencing. Nature 2010;467:1061. <https://doi.org/10.1038/nature09534>.

[17] Curry HB. The method of steepest descent for non-linear minimization problems. Quart. of App. Maths. 1944;2:258[–](https://www.ams.org/journals/qam/1944-02-03/S0033-569X-1944-10667-3/S0033-569X-1944-10667-3.pdf)61. [https://www.ams.org/journals/qam/1944-0 2-03/S0033-569X-1944-10667-3/S0033-569X-1944-10667-3.pdf](https://www.ams.org/journals/qam/1944-02-03/S0033-569X-1944-10667-3/S0033-569X-1944-10667-3.pdf).

[18] [Denoeux T. Logistic regression, neural networks and dempster](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref18)–[shafer theory: a new](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref18)  [perspective. Knowl Base Syst 2019;176:54](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref18)–[67](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref18).

9

H. Soumare et al. Array 11 (2021) 100068

[51] [Montgomery DC, Peck EA, Vining GG. Introduction to linear regression analysis.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref51)  [John Wiley](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref51) & [Sons; 2021](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref51).

[52] Ng A, et al. Sparse autoencoder. CS294A Lect. notes 72. 2011. p. 1[–](http://ailab.chonbuk.ac.kr/seminar_board/pds1_files/sparseAutoencoder.pdf)19. [http ://ailab.chonbuk.ac.kr/seminar\_board/pds1\_f](http://ailab.chonbuk.ac.kr/seminar_board/pds1_files/sparseAutoencoder.pdf)i[les/sparseAutoencoder.pdf](http://ailab.chonbuk.ac.kr/seminar_board/pds1_files/sparseAutoencoder.pdf).

[53] Owen AB. A robust hybrid of lasso and ridge regression. Contemp Math 2007;443: 59[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref54)72. <https://doi.org/10.1090/conm/443/08555>.

[54] [Ozanich E, Gerstoft P, Niu H. A feedforward neural network for direction-of-arrival](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref54)

[55] Pal A, Lane C, Vidal R, Haeffele BD. On the regularization properties of structured [estimation. J Acoust Soc Am 2020;147:2035](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref54)–[48](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref54).

dropout. In: Proceedings of the IEEE/CVF conference on computer vision and

[56] Patel P, Thakkar A. The upsurge of deep learning for computer vision applications. pattern recognition; 2020. p. 7671–9. <https://arxiv.org/abs/1910.14186>.

Int J Electr Comput Eng 2020;10:538. [https://doi.org/10.11591/ ijece.v10i1.pp538-548](https://doi.org/10.11591/ijece.v10i1.pp538-548).

[57] [Pei J, Wang W, Osman MK, Gan X. Multiparameter optimization for the nonlinear performance improvement of centrifugal pumps using a multilayer neural network.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref57)

[58] [Phaisangittisagul E. An analysis of the regularization between l2 and dropout in](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref58) [J Mech Sci Technol 2019;33:2681](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref57)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref58)[91](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref57).

[single hidden layer neural network. In: 2016 7th international conference on](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref58)

[59] Project, G., . 1000 Genome project datasets. [intelligent systems, modelling and simulation (ISMS). IEEE; 2016. p. 174](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref58)–9.

[60] project OE. Expression project for oncology. 2005. [https://www.ncbi.nlm.nih](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc&equals;GSE2109)

[61] [Qi GJ, Zhang L, Lin F, Wang X. Learning generalized transformation equivariant](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref61) [.gov/geo/query/acc.cgi?acc](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc&equals;GSE2109)[¼](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref61)[GSE2109](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc&equals;GSE2109). [Accessed 7 September 2020].

[representations via autoencoding transformations. IEEE Transactions on Pattern Analysis and Machine Intelligence; 2020](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref61).

[62] Ravì D, Wong C, Deligianni F, Berthelot M, Andreu-Perez J, Lo B, Yang G. Deep learning for health informatics. IEEE JBHI 2016;21:4–21. [https://doi.org/10.1109/ JBHI.2016.2636665](https://doi.org/10.1109/JBHI.2016.2636665).

[63] [Reed R, MarksII RJ. Neural smithing: supervised learning in feedforward artif](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref63)i[cial](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref63)  [neural networks. Mit Press; 1999](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref63).

[64] Romero A, Carrier PL, Erraqabi A, Sylvain T, Auvolat A, Dejoie E, Legault MA, Dub�e MP, Hussin JG, Bengio Y. Diet networks: thin parameters for fat genomics. 2016. arXiv preprint arXiv, 1611.09340.

[65] Rong D, Xie L, Ying Y. Computer vision detection of foreign objects in walnuts using deep learning. Comput Electron Agric 2019;162:1001–10. [https://doi.org/ 10.1016/j.compag.2019.05.019](https://doi.org/10.1016/j.compag.2019.05.019).

[66] Sakurada M, Yairi T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceed. Of the MLSDA 2014 2nd workshop MLS data

[67] Salimans T, Kingma DP. Weight normalization: a simple reparameterization to analysis; 2014. p. 4–11. <https://doi.org/10.1145/2689746.2689747>.

accelerate training of deep neural networks. In: NIPS; 2016. p. 901–9. [https://arxiv. org/pdf/1602.07868.pdf](https://arxiv.org/pdf/1602.07868.pdf).

[68] Schmidhuber J. Deep learning in neural networks: an overview. Neur. networ.

[69] [Sharkawy AN. Principle of neural network and its main types. J. Adv. Appl.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref69) 2015;61:85[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref69)117. <https://doi.org/10.1016/j.neunet.2014.09.003>.

[70] [Singh V, Baranwal N, Sevakula RK, Verma NK, Cui Y. Layerwise feature selection in](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref70) [Comput. Math. 2020;7:8](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref69)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref70)[19](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref69).

[stacked sparse auto-encoder for tumor type prediction. In: 2016 IEEE international](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref70)

[71] Sit MA, Koylu C, Demir I. Identifying disaster-related tweets and their semantic, [conference on Bioinformatics and biomedicine (BIBM). IEEE; 2016. p. 1542](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref70)–[8](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref70).

[73] Stiglic G, Kokol P. Stability of ranked gene lists in large microarray analysis studies. BioMed Res Int 2010;2010. <https://doi.org/10.1155/2010/616358>.

[74] [Sutradhar R, Barbera L. Comparing an artif](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref74)i[cial neural network to logistic regression for predicting ed visit risk among patients with cancer: a population-based cohort study. J Pain Symptom Manag 2020;60:1](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref74)–[9](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref74).

[75] Suzuki T. Generalization bound of globally optimal non-convex neural network training: transportation map estimation by infinite dimensional Langevin dynamics. 2020. arXiv preprint arXiv, 2007.05824.

[76] [Svozil D, Kvasnicka V, Pospichal J. Introduction to multi-layer feed-forward neural](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref76)

[77] [Tian Y, Lu C, Zhang X, Tan KC, Jin Y. Solving large-scale multiobjective](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref77) [networks. Chemometr Intell Lab Syst 1997;39:43](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref76)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref77)[62](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref76).

[optimization problems with sparse optimal solutions via unsupervised neural networks. IEEE transactions on cybernetics 2020](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref77).

[78] Tikhonov AN. On the stability of inverse problems. In: Dokl. Akad. Nauk SSSR;

[79] Wager S, Wang S, Liang SP. Dropout training as adaptive regularization. In: NIPS; 1943. p. 195–8. <https://doi.org/10.1007/978-3-642-81472-3_5>.

[80] Wang Y, Yao H, Zhao S. Auto-encoder based dimensionality reduction. 2013. p. 351–9. <https://arxiv.org/pdf/1307.1493.pdf>.

Neurocomputing 2016;184:232–42. [https://doi.org/10.1016/ j.neucom.2015.08.104](https://doi.org/10.1016/j.neucom.2015.08.104).

[81] Wei C, Kakade S, Ma T. The implicit and explicit regularization effects of dropout. In: International conference on machine learning. PMLR; 2020. p. 10181–92. [htt ps://arxiv.org/abs/2002.12915](https://arxiv.org/abs/2002.12915).

[82] [Weisberg S. Applied linear regression, ume 528. John Wiley](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref82) & [Sons; 2005](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref82).

[83] [Wen T, Zhang Z. Deep convolution neural network and autoencoders-based](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref83)

[84] Werbos PJ. Generalization of backpropagation with application to a recurrent gas [unsupervised feature learning of eeg signals. IEEE Access 2018;6:25399](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref83)–[410](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref83).

market model. N. networ. 1988;1:339–56. [https://doi.org/10.1016/0893-6080(88) 90007-X](https://doi.org/10.1016/0893-6080(88)90007-X).

[85] Werbos PJ. Backpropagation through time: what it does and how to do it. Proc of

[86] [Wright RE. Logistic regression. 1995](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref86). the IEEE 1990;78:1550[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref86)60. <https://doi.org/10.1109/5.58337>.

[87] [Xia Y, Qin T, Chen W, Bian J, Yu N, Liu TY. Dual supervised learning. In:](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref87)

[88] Xie F, Zhang J, Wang J, Reuben A, Xu W, Yi X, Varn FS, Ye Y, Cheng J, Yu M, et al. [International conference on machine learning. PMLR; 2017. p. 3789](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref87)–[98](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref87).

Multifactorial deep learning reveals pan-cancer genomic tumor clusters with distinct immunogenomic landscape and response to immunotherapy. Clin Canc Res

[89] [Xie Y, Wu X, Ward R. Linear convergence of adaptive stochastic gradient descent.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref89) 2020a;26:2908[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref89)20. <https://doi.org/10.1158/1078-0432.CCR-19-1744>.

[In: International conference on artif](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref89)i[cial intelligence and statistics. PMLR; 2020b.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref89)

[90] [Xin J, Embrechts MJ. Supervised learning with spiking neural networks. In:](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref90) [p. 1475](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref89)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref90)[85](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref89).

[IJCNN’01. International joint conference on neural networks. Proceedings (cat.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref90)

[91] [Yang J, Ma J. Feed-forward neural network training using sparse representation.](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref91) [Noh01CH37222), IEEE; 2001. p. 1772](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref90)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref91)[7](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref90).

[92] [Yang X, Deng C, Zheng F, Yan J, Liu W. Deep spectral clustering using dual](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref92) [Expert Syst Appl 2019;116:255](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref91)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref92)[64](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref91).

[autoencoder network. In: Proceedings of the IEEE/CVF conference on computer](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref92)

[93] [Zheng S, Zhao J. A new unsupervised data mining method based on the stacked](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref93) [vision and pattern recognition; 2019. p. 4066](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref92)[–](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref93)[75](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref92).

[autoencoder for chemical process fault diagnosis. Comput Chem Eng 2020;135:](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref93)

spatial and temporal context using deep learning, natural language processing and [106755](http://refhub.elsevier.com/S2590-0056(21)00016-3/sref93).

spatial analysis: a case study of hurricane irma. International Journal of Digital Earth 2019. <https://doi.org/10.1080/17538947.2018.1563219>.

[72] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. JMLR 2014;15:1929–58. <https://dl.acm.org/doi/abs/10.5555/2627435.2670313>.

10