Optimal Classification Tiling Algorithm

Wei-Chen Cheng
Polylogarithmic technology Ltd
Department of Oncology, University of Oxford
United Kingdom
wei-chen.cheng@polylogarithmic.com

Cheng-Yuan Liou
Dept. of Computer Science and Information Engineering
National Taiwan University
Taiwan, R.O.C.

***Abstract*— This work presents a max-margin tiling algorithm to construct multi-layer perceptrons with perfect performance for all training patterns. Each perceptron distinguishes a portion of a single class patterns from all the rest patterns, as large as possible. The process is a divide and conquer approach therefore the size of the portion gradually reduced and the algorithm accelerate accordingly. Each layer of such perceptrons encodes different class patterns with different codes, maintaining faithful representations. The max-margin separation boundary is optimal to against noise therefore this construction provides resolution of challenging classification of noisy patterns such as seriously corrupted images. To our knowledge, it gives the best performance for noisy testing patterns among all existing methods. We applied the algorithm on highly sparse data of genetic mutations and identified not only gene NF1, which had been previously found by monogenic analysis, but also a list of genes whose deletions may polygenically associate with a group of breast cancer patients.**

Keywords—pattern recognition; classification; faithful representation; tiling algorithm; sparse binary pattern; gene mutation

I. INTRODUCTION

The tiling algorithm [5] used to train the feedforward neural network tend to generate an excessively large number of neurons and layers in order to achieve zero learning error, 100% training accuracy, for all binary patterns in the training dataset. They inserted a hyperplane for each pattern as near as a small distance ε and place the pattern at the positive side, the stable side, of the hyperplane. The small distance ε causes this kind of hyperplane unable to tolerate any noise. However, the assumption of infinitely small distance, ε, is necessary in their proof of the Theorem [5] under two special conditions, where the first is the pattern spaces span a large number of dimensions and the second is patterns are randomly distributed. In real applications, the number of pattern dimensions is normally limited to a relatively small number, for example twenty thousand genes in bioinformatics. Besides that, patterns are all fixed in its space instead of a random distribution. For these reasons, the limitation of extremely small distance ε can be eased. The relaxation of the requirement on small distance allows the separation boundaries to have wide margins between different classes of patterns thus tolerate noise in the task of pattern classifications. This work devises various techniques to construct neurons and layers of neural networks taking advantages of large margins.

In one hand, the backpropagation algorithm used to train a fixed structure of neural network cannot accomplish zero learning error [7][8] as the training was performed on a pre-determined structure. In the other hand, Support Vector Machines (SVM) [1] uses only a single perceptron finding a decision boundary with maximal margin between two classes. Due to the limitation of the structure, it relies on kernel function to nonlinearly transform the data into a different space, normally to a higher dimensional space to gain more flexibility and nonlinearity, to accomplish the training of the single perceptron. In other words, what SVM adjusts is mainly the feature space of the data. Although the kernel function can be powerfully expressive, SVM does not have a mean to train the transformation, therefore it is an arbitrarily designed encoder. To determine the right space for the kernel functions, one has to use some other statistical techniques such as cross-validation [9], which had been widely used to decide the parameters of statistical models, by repeatedly learning and testing [10]. Giving certain degree of penalty to the loss function of SVM and creating a so-called soft margin hyperplane, a better classification outcome can be achieved. This work further devises effective tiling algorithms to construct multilayer neural network using linear SVM with its property of wide margin.

The Bi-perceptron algorithm [2][3] tried tiling algorithms with wide-margin properties to separating different classes of patterns. Perceptrons with large margins between patterns performed better on noisy data and greatly reduce the total number of layers and neurons. It also preserved the faithful representations [5] in each layer hence held the key to improve the tiling algorithm and augment the classification power of neural network. The Bi-perceptron algorithm achieved zero error for all training patterns by constructing three hidden layers if patterns are analog and two layers if binary. As shown in the work published in [2], the two parallel separation planes of the perceptron separate one portion of patterns in a single class as large as possible from the rest of classes with wide margins, see Fig.1. Portion by portion, all patterns in that single class can be separated and isolated from the rest in a divide and conquer manner. Such strategy not only drastically reduced the total number of faithful representations [4][5] and layers but also be able to deal with imperfect patterns such as gene expression.

979-8-3503-6463-7/24/$31.00 ©2024 IEEE

Due to the nature of pattern distribution, a reasonable attempt to select the portion to be isolated is to choose the patterns in the class from the outermost portion, farthest from the center, in the space by using a perceptron. Then take that portion of patterns out and repeat the process for the remaining patterns until none of them left. This isolation strategy is different from those proposed in [2][5] thus will be discussed and developed in the rest of this work.

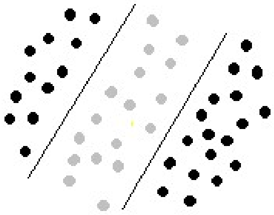

Fig. 1: Illustration of an isolated portion of single class patterns from all the other classes by two parallel perceptrons in the Bi-perceptron algorithm [2].

In terms of binary data, all patterns are located at the corners of a hypercube and are equally away from the center. Fig. 2 sketches a simplified 3D pattern space and a perceptron representing a wide-margin hyperplane that sperate a pattern from the other three of different classes. The wide-margin hyperplane is an optimal separation boundary to resist the noise because both side of the hyperplane have the space to against data fluctuation.

During the isolation process of building a neural network, a perceptron that cover the largest portion of training patterns in same class is added to current layer. Those patterns being covered will be excluded therefore the size of training set reduces gradually after each isolation. This isolation process guarantees the output of the layer are faithful [4][5] for all patterns, which means different classes of patterns will have different output. In the following sections, we show how to operate the perceptrons to isolate binary patterns in the first layer.

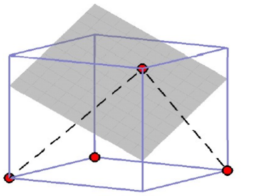

Fig. 2: Schematic diagram of a perceptron obtained from linear SVM, whose hyperplane $W_1^{(1)}$, separating four binary patterns. One pattern, the red point at top, 'p_m' is on positive side of the plane, $|\{s_m\}|^1=|\{s_M\}|^1=1$. Three patterns, the red points at bottom, are on negative side $|\{d_m\}|^1=3$ of the plane; two of the three patterns, leftmost and rightmost points, are the support vectors of SVM in $\{d_m\}$.

II. TILING ALGORITHM FOR BINARY DATA

Considering binary input of cancer mutation data of currently known genes for breast, prostate, and lung cancer patients, we have M samples and N genes. The collection of all samples is represented as a set in $\{p_m\ , m=1 \sim M\}$ and the size of the set $|\{p_m\}|$ is M. All genes are either having damaged mutation or being normal therefore in one of the binary states. Each type of cancer is a class of sample and three classes, breast, prostate, and lung, are used as an example, $C \equiv \{C_0=$ 'breast', $C_1=$ 'prostate', $C_3=$ 'lung'$\}$. We plan to constructed multiple perceptrons in the first layer of a network to separate breast cancer samples C_0 from other samples $(C_0)'$. There are five steps of them

Step 1: Construct ancillary hyperplane.

In order to determine an ancillary hyperplane W_m for each breast cancer sample p_m in C_0, we find the set $\{d_m\}$ which is the collection of samples of other classes $(C_0)'$ nearest to p_m. Therefore, $\{d_m\}$ may contain samples from either prostate or lung cancer, or both. The variable m will iterate through each breast cancer samples from one to $|C_0|$. The size of $\{d_m\}$ has to be at least of N because that is the minimal number required to exactly determine a hyperplane of input with N dimensions. In this example, the hyperplane has N variables corresponding to N genes. These N samples in set $\{d_m\} \equiv \{p_q;\ p_q \in \{$smallest values $|p_m - p_q|$; $q=1 \sim N$ and $p_q \notin C_0\}\}$ are those samples p_q having the most similar mutation patterns but different cancer types to p_m. With this size of sample set $|\{d_m\}|=N$, it's enough to construct an ancillary hyperplane W_m, which has N dimensions, that passes through these N samples in the set $\{d_m\}$. The sample p_m is placed at the positive side of this ancillary hyperplane and the set of samples $\{d_m\}$ at negative side for simpler notations. This is not the final hyperplane of the perceptron therefore is ancillary.

The hyperplane W_m can be obtained using linear SVM [1] as a binary classifier by placing the single sample p_m on the positive side and the set of samples $\{d_m\}$ on the other side of the hyperplane. The linear SVM algorithm compute a max-margin hyperplane to split the two groups of samples, breast cancer and other types of tissue. It doesn't require N samples for N dimensional data to determine the hyperplane hence $|\{d_m\}|$ is not necessary to be N. The plane can serve as the ancillary hyperplane W_m that we are constructing. After the W_m being determined, we can shift it by changing the bias term, gradually move the hyperplane without altering its orientation toward $\{d_m\}$ and away from p_m until the hyperplane finally meet one of the nearest support vector or data in set $\{d_m\}$. It is highly recommended to use this SVM perceptron to obtain the ancillary hyperplane W_m.

Step 2: Construct a second set.

For each m, the next step is to find a set $\{s_m\}$ containing all samples that of the same type of cancer and being classified the same with p_m, which means those breast cancer samples on the positive side of the hyperplane W_m. The size of $\{s_m\}$, denoted as $|\{s_m\}|$, can vary for each m, depends on how many samples of breast cancer in C_0 are correctly placed on the right side of the hyperplane. Another way is using all samples in C_0 to be the set of $\{s_m\}$ for the sample p_m in C_0. In this way, the set size $|\{s_m\}|$ will be equal to $|C_0|$.

Step 3: Picking the set of $\{s_m\}$ with largest size.

Among different samples of breast cancer p_m, this step is to pick the set $\{s_M\}^1$ containing the largest number of C_0 samples, which means the set having the most samples on the correct side of hyperplane W_M. The $\{s_M\}^1$ can be expressed as $max_m\{|\{s_m\}|; p_m \in C_0\}$. We now have the required samples, which are in $\{s_M\}^1$, to construct the first isolation perceptron, $W_1^{(1)}$, for the class C_0.

Step 4: Construction of isolation perceptron.

Select the sample, p_u, which is the nearest to hyperplane W_M from $\{s_M\}^1$, so $p_u \in \{s_M\}^1$. A normal vector perpendicular to the ancillary hyperplane W_M and passing p_u will have a projection p_v on the hyperplane. Construct an isolation hyperplane that passes the middle point between p_u and p_v, and is parallel to the ancillary hyperplane W_m. The isolation hyperplane is denoted as $W_1^{(1)}$ where the superscript is the layer number and the subscript is the neuron number. The $W_1^{(1)}$, intersects at the middle point with the normal vector. The synaptic weights of the first neuron in the first layer can be obtained from the coefficients of the hyperplane $W_1^{(1)}$.

Another way is that $W_1^{(1)}$ can be obtained from linear SVM trained by the setting of sample p_u on the positive side and the $\{d_m\}$ on the negative side. This SVM-version of $W_1^{(1)}$ has the wide margin between p_u and its support vectors of $\{d_m\}$. The weights of the first neuron in the hidden layer, Fig. 3, can be obtained using the coefficients of the SVM-version of $W_1^{(1)}$.

Step 5: Reduce the set of C_0 and construct next perceptron.

Once we have a perceptron covering a group of breast cancer samples $\{s_M\}^1$ and producing consistent output for them, all the samples in set $\{s_M\}^1$ can be removed from the training of current layer. The remaining breast cancer samples not yet to be classified correctly is collected as a set C_0^1. The C_0^1 is the intersect between C_0 and $(\{s_M\}^1)'$, $C_0^1 \equiv C_0 \cap (\{s_M\}^1)'$, if $(\{s_M\}^1)'$ is the complement set of $\{s_M\}^1$.

The whole procedure, step 1~5, repeats again using the samples in reduced breast cancer set C_0^1 to generate another isolation perceptron $W_2^{(1)}$ obtained from $\{s_M\}^2$. The intersection of $\{s_M\}^a$ and $\{s_M\}^b$ is empty, $\{s_M\}^a \cap \{s_M\}^b = \emptyset$ for a ≠ b, and the union of all reduced sets containing all samples of breast cancer C_0, $\{\{s_M\}^1 \cup \{s_M\}^2 \cup \{s_M\}^3 \cup \ldots \cup \{s_M\}^L\} \equiv C_0$.

Set all the weights in the second hidden layer to be one, $W_{kj}^{(2)}=1$ for k=1 and all j=1, .., L, see Fig. 3. The mutation patterns of breast cancer will trigger the network and produce more than one activated perceptron at the hidden layer because the sample is on the positive side of the hyperplane as the way it was constructed. At the hidden layer, some neurons are therefore positive if input is from breast cancer while all neurons are negative if from the other cancers. The output layer can simply detect if any hidden neuron is activated. In this figure, we set N_1=L and $W_{ji}^{(1)}$ is the synaptic weight of the j^th neuron to the i^th input gene. The genes as shown in Fig. 3 can be from different biological pathways. We expect the multilayer network that utilizes the max-margin ability to associate the output and gene mutation patterns to provide us useful insight about biological pathways, for example links between homologous recombination deficiency and breast cancer [11], used to achieve perfect classification.

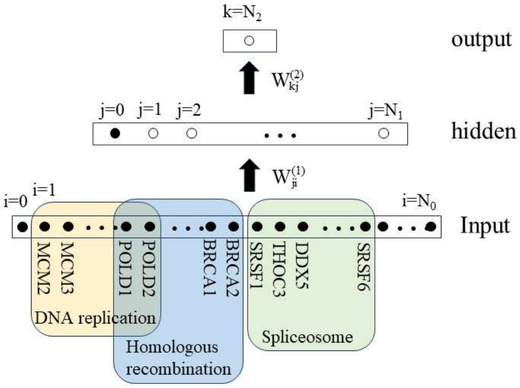

Fig. 3: Illustration of the neural network for gene mutation analysis. The number of neurons at input layer is the number of genes $N_0=N$, the hidden layer is $N_1=L$, and the output layer is $N_2=1$. The threshold value of input layer is $W_{k0}^{(1)}=1$ and the hidden layer is $W_{k0}^{(2)}=-0.5$.

III. ALTERNATIVE CONSTRUCTION FOR BINARY PATTERNS

In the above sections, one may pick either $p_m \in C_0$ or $p_m \in (C_0)'$ using the learning method of the five steps. Due to the hypercubic structure of binary input in space, we can pick any pattern $p_m \in C$ that will be on the outer edge of the pattern space. We can construct a separation hyperplane which isolates, as large as possible, a portion of the same class patterns as that of p_m. Then pick another sample from the remaining C. Iteratively operate these five steps and keep reducing the patterns in set C, to obtain a series of isolation perceptrons, $\{W_1^{(1)} W_2^{(1)} W_3^{(1)},.., W_L^{(1)}\}$, where $W_L^{(1)}$ is the last isolation perceptron, namely the last neuron in the hidden layer, see Fig. 3.

These neurons comprised of the neurons in the first hidden layer as that described in [2][3]. The following layer can be constructed in a similar way by using the outputs of its previous layer. In his way, the construction maintains the faithfulness of each layer. The construction of layers finishes when the classification is accomplished as in [4][5]. This construction method gives better performance in many cases.

IV. ANALOG PATTERNS

Since the hypercubic structure for binary pattern is not available for continuous data features, we slightly modify the way of construction for analog patterns. The schema of the construction procedures is depicted in Fig. 4. The isolation boundary was placed to separate a portion of same patterns that are on the outer edge of the data cluster. The construction of neurons in each layer will be finished when the output of the layer reaches faithfulness, in other words patterns in different classes produces different output. The construction of layers will be stopped whenever the classification task is accomplished [1][4].

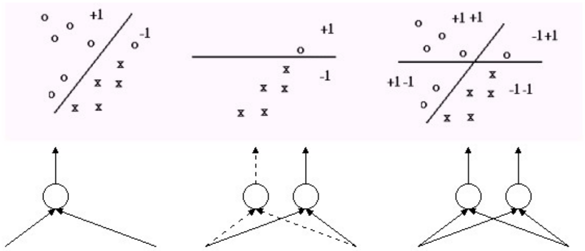

Fig. 4: Illustration of hyperplane constructions for analog patterns in the first hidden layer. The first neuron (left) split the majority of the circles from the crosses with one error. The second neuron (center) separate the remaining circle from the crosses and both neurons (right) together successfully generate faithful representations.

V. ALTERNATIVE ISOLATION TECHNIQUE

For more complex data, one can isolate different classes by sampling patterns near the separation border right between neighboring classes, see Fig. 5. Then group a small number of neighboring patterns along the border, see the large circles in Fig. 5. Construct one linear SVM hyperplane for the border patterns in each group. These SVM hyperplanes comprise the neurons in the first hidden layer. We stop increasing the number of neurons when the faithful representation of the first hidden layer is obtained. Then proceed to the construction of the next hidden layer. The layer construction will be stopped when the correct classification of all input is reached. This border construction solves the local minima problem suffered by back propagation algorithm [7][8].

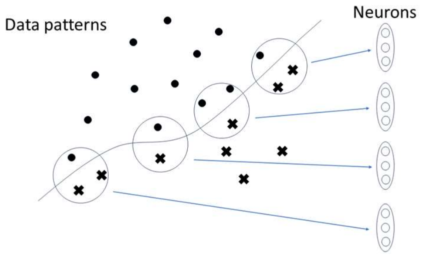

Fig. 5: The focus on the neighbors between two different classes, black circles and crosses, offer the power to approximate the non-linear separation boundary by multiple linear hyperplanes, which represented by the neurons plotted at the right-hand side.

VI. SIMULATION RESULT

The mutation profiles of breast, prostate, and lung cancers, were downloaded from The Cancer Genome Atlas Program (TCGA) [13]. These three types of cancers are with the highest incident rates globally. The sequence protocol of breast and prostate cancer were WXS and lung cancer, or lung squamous cell carcinoma specifically, was capture sequencing. This means only mutations on exon regions are to be detected, the data didn't contain mutations in intergenic regions and intron regions, etc. Only two types of mutations were considered in our analysis. The first type is the damaging mutations causing malfunctioned protein product, such as changes at start codons. The second type is the non-conserving mutation causing the properties of proteins changed, such as insertion or deletion at stop codons. The sample size of each cancer type in the dataset were $|C_0|=992$, $|C_1|=331$, $|C_2|=178$, overall $M=1501$ in total. The number of genes detected with single nucleotide mutation was around $N=18676$. During processing the breast cancer samples in C_0, the size distribution of each $\{s_m\}$ for first neuron is listed in Table 1. The size of $\{s_M\}^1$, largest set among $\{s_m\}$, was $|\{s_M\}^1|=672$, accounting for 67.74% of breast cancer samples in the dataset. The sparsity of the matrix, the number of non-zero elements divided by the size of the matrix, was around 0.44%.

Table 1: The distribution of the size $|\{s_m\}|$ for the m iterating through one to $|C_0|=992$.

Number of $|\{s_m\}|$	Frequency
<5	854
5~10	116
>10	22

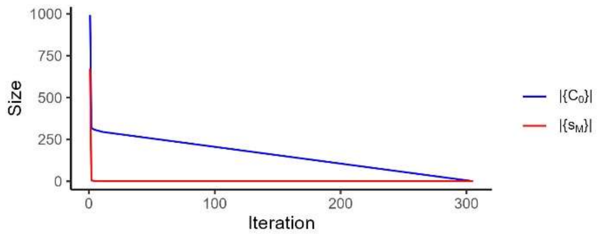

Figure 6: The training procedure were recorded. The size of C_0 (blue line) was gradually decreased as the iteration increased. The size of $\{s_M\}^1$ (red line) was 672, $\{s_M\}^2$ was 6, and kept reducing as the number of neurons increased. The training process reached 100% accuracy to classify all breast cancer samples correctly in C_0 after 305 iterations.

In each iteration, the algorithm visited through all m, and found the largest set of $\{s_m\}$ containing the correctly classified samples. The set size of samples, $|\{s_M\}^i|$, covered by each neuron gradually reduced to one, $|\{s_M\}^{i>10}|=1$, and the output of the network eventually reached 100% classification accuracy, see Fig. 6. The small set size after certain iteration, $|\{s_M\}^{i>10}|=1$, could be due to the limited sample size and we expect the capacity of each neuron can cover will increase if the collection of samples expands. The algorithm halted as it reached faithful representation for all samples in C_0 after 305 iterations of repeating the five steps described in Section 2 and reached zero training error.

There were 13801 non-zero elements in the first neuron $W_1^{(1)}$, corresponding to the 13801 genes contributed to shape of decision boundary. Six of the genes were Mitochondrial. The $W_1^{(1)}$ covered samples from as little as one detected protein-affected mutation up to as high as 1060 mutations (988 genes). The threshold value of the first neuron at the hidden layer was $W_{10}^{(1)}$=-0.11 therefore the higher the weight was, the more the mutation contributes to reach the threshold of being classified as this group of breast cancer samples. Table 2 lists the highest and lowest ten weights of them.

Table 2: The top ten positive and negative weights of genes contributed to the recognition of the breast cancer group classified by the first neuron $W_1^{(1)}$.

Top 10		Bottom 10	
Weight	Gene	Weight	Gene
1.68	NF1	-0.12	HERC5
0.02	CDH10	-0.07	ASB5
0.02	ZFHX4	-0.07	ERBB3
0.02	COL22A1	-0.07	APAF1
0.02	ZNF208	-0.07	ADGRL3
0.02	SRRM2	-0.06	DMBX1
0.02	SPHKAP	-0.06	IRX1
0.02	RELN	-0.06	TP53
0.02	KIAA1549	-0.06	NOP58
0.02	UNC5D	-0.06	OSBPL10

The largest weight of the first neuron corresponds to Neurofibromin 1 (NF1). At least fourteen isoforms are currently known for this gene and its deletion was reported to be linked with the risk of breast cancer [14][15][16]. In $\{s_M\}^1$, 17 patients were detected with damaging mutations on NF1 and 10 patients with non-conserving mutations to it. The weight of HERC5 is the negatively largest however none of patients in this group having a damaging and non-conserving mutation to this gene. The low weight of HERC5 doesn't mean the damage of the HERC5 function lower the chance of being classified as breast cancer because the other neurons might cover the patients with this mutation. Therefore, the negative weight means the classification of this type of samples tends to be not in this group.

We visualized [12] the mutation profiles of neuron $W_1^{(1)}$ along with chromosome, in Fig 7, and can see the determinant mutations are not uniformly scattered on the DNA. Most of them contribute some but a few of them contribute significantly more, such as those genes listed in Table 2, for the final classification of this specific group of breast cancer samples. As expected, none of the mutations should happen in chromosome Y as the majority of the breast cancer cases are female.

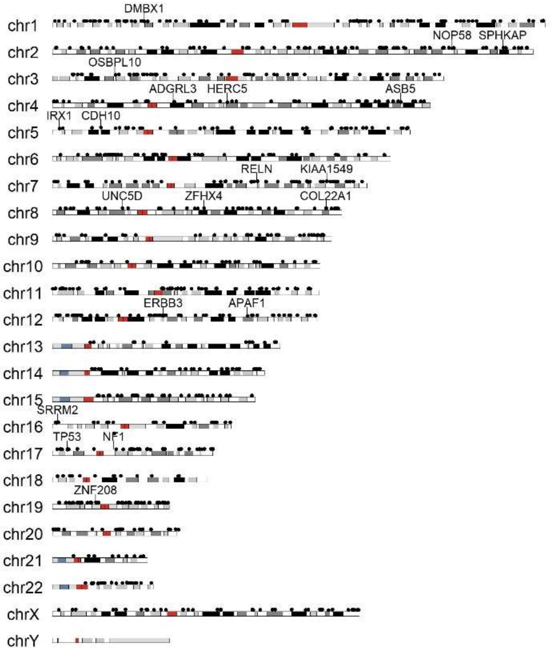

Figure 7: The weight of mutation profiles on the ideogram. The chromosomes are listed from chromosomes one to 22, X and Y. Mitochondrial DNA was used but is not plotted here. The height (y-axis) of the black dots above each chromosome denotes the synaptic weights corresponding to the genes. Top ten genes with largest and smallest weights are annotated with their gene names accordingly.

The advantageous multivariate analysis by proposed model considered the interaction and covariance among genes. For this reason, the weights and effects were not seen as standalone variables; genes are correlated. Figure 8 illustrates a view about how the decision were shaped. For example, the missense mutation of the tumor suppressor gene TP53 brought the initial weight closer to the threshold and a sequence of cumulative weights over mutated genes led to the final classification outcome of a patient sample. The classification was not based on single gene.

The patient on the left of Fig. 8 who had 15 damaging mutations and 33 non-conserving mutations were classified into the breast cancer group. Among the 15 damaging mutations, 11 of them were causing frame-shifting, three were nonsense mutations and one happened at splice site. For the 33 non-conserving mutations, only two of them were caused by in-frame deletion and the rest were missense mutations. The patient on the right of Fig. 8 had 13 damaging mutations and 46 non-conserving mutations. Five of the damaging mutations were deletion causing frame-shifting, four were nonsense mutation, and four were mutations at splice sites. For the 46 non-conserving mutations, only a mutation was in-frame deletion and the rest were missense mutations. Note that the mutations only happened to one of the alleles, therefore the protein functions might be decreased but not ceased.

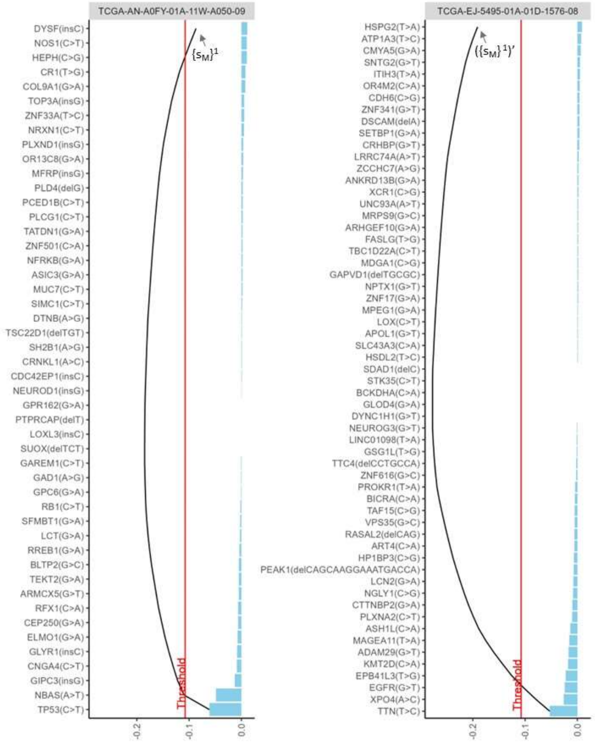

Figure 8: The cumulative weights against the threshold $W_{10}^{(1)}$. The blue barplot indicates the synaptic weights for each individual genes shown in y-axis and the black curve is the cumulative weights from the bottom of the figure to that point. The classification of each sample, whose barcodes are shown in subtitle at top, can be seen as the comparison of the final cumulated weight at the top with the threshold (red line). The sample is classified in the set $\{s_M\}^1$ if the sum of weight is larger than the threshold and not in the set if smaller.

VII. Discussion

Neural networks have been successfully used to accomplish complex tasks in many areas including biology and large language modeling. We have described a training method for multilayer neural networks with flexible structure by taking the advantage of the property of large margin to tolerate noisy data in the task of pattern recognition and guarantee a hundred percent of training accuracy. This neural network can generate dynamic hypothesis space therefore high VC dimensions [17]. The optimal classification tiling algorithm is shown to be a powerful model to analyze complex and sparse data of genetic mutation from large database. This multivariate analysis not only found the gene previously identified but also provided new candidates for further analysis.

References

[1] C. Cortes, V.N. Vapnik, "Support vector networks," Machine Learning, 20, 1995, pp. 273-297

[2] D.-R Liou, Y.-E. Chen, C.-Y. Liou, "A parallel bi-perceptron approach and its application to data classification," International Conference on Computational Science and Computational Intelligence, 2016, pp.1152~1157, December 15~17, Las Vegas

[3] C.-Y. Liou, W-J. Yu, "Initializing the weights in multilayer network with quadratic sigmoid function," International Conference on Neural Information Processing, ICONIP, 1994, pp.1387-1392, October 17-20, Seoul

[4] C.-Y. Liou, W.-J. Yu, "Ambiguous binary representation in multilayer neural network," Proceedings of International Conference on Neural Networks, ICNN, 1995, pp. 379-384, November 27 - December 1, Perth, Australia

[5] M. Mézard, J.-P. Nadal, "Learning in feed-forward layered networks: The tiling algorithm," Journal of Physics A, 22, 1989, pp. 2191-2203

[6] W. Pedrycz, J. Waletzky, "Neural-network front ends in unsupervised learning," *IEEE Trans. on* Neural Networks 8, 1997, pp. 390-401

[7] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning representations by back-propagating errors," Nature, 323, 1986, pp. 533-536

[8] T. J. Sejnowski, C. R. Rosenberg, "NETalk: a parallel network that learns to read aloud," The Johns Hopkins University Electrical Engineering and Computer Science Tech. Report, JHU/EECS-86/01, 198

[9] G. Wahba and S. Wold, "Periodic splines for spectral density estimation: The use of cross validation for determining the degree of smoothing." Communications in Statistics, 4, 1975, pp. 125–141

[10] P. Burman, "A comparative study of ordinary cross-validation, r-fold cross-validation and the repeated learning-testing methods." Biometrika, 76, 1989, pp. 503-514

[11] M.D. Stewart, D. M. Vega, R. C. Arend, etc., "Homologous Recombination Deficiency: Concepts, Definitions, and Assays." The Oncologist, 27, 2022, pp. 167–174

[12] B. Gel and E. Serra. "karyoploteR: an R/Bioconductor package to plot customizable genomes displaying arbitrary data," Bioinformatics, 33, 2017, pp. 3088-3090

[13] The Cancer Genome Atlas Research Network, J. Weinstein, E. Collisson, et al., "The Cancer Genome Atlas Pan-Cancer analysis project," Nature Genetics, 45, 2013, pp. 1113–1120

[14] L. Pacot, J. Masliah-Planchon, A. Petcu, B., et al., "Breast cancer risk in NF1-deleted patients," Journal of Medical Genetics, 61, 2024, pp. 428-429

[15] M.D. Wallace, A.D. Pfefferle, L.S. Shen, A.J. McNairn, E.G. Cerami, B.L. Fallon, V.D. Rinaldi, T.L. Southard, C.M. Perou, J.C. Schimenti, "Comparative Oncogenomics Implicates the Neurofibromin 1 Gene (NF1) as a Breast Cancer Driver," Genetics, 192, 2012, pp. 385–396

[16] S.J. Howell, K. Hockenhull, Z. Salih, D.G. Evans. "Increased risk of breast cancer in neurofibromatosis type 1: current insights." Breast Cancer: Targets and Therapy, 9, 2017, 531–536

[17] V. N. Vapnik "Statistical Learning Theory." 1998