MoMLeT&DS-2024

SYNTHESIS OF MULTITHRESHOLD NEURAL NETWORK CLASSIFIER

VLADYSLAV KOTSOVSKY

GOALS OF THE RESEARCH

- The study of the model of 2-layer neural network whose hidden layer consists of binary-valued multithreshold neurons.
- The development of the synthesis algorithm for such network and the investigation of its performance.

MODEL OF LABELED *K*-THRESHOLD NEURAL UNIT

The performance of labeled k-threshold *binary-valued* neuron with the *weight vector* $\mathbf{w} \in \mathbf{R}^n$ and (ordered) *threshold vector* $\mathbf{t} \in \mathbf{R}^k$ can be described as follows:

$$y = \begin{cases} \lambda, & \text{if } t_{2j-1} \le \mathbf{w} \cdot \mathbf{x} < t_{2j},\ j \in \{1, \ldots [k/2]\}, \\ \bar{\lambda}, & \text{if } t_{2j} \le \mathbf{w} \cdot \mathbf{x} < t_{2j+1},\ j \in \{0, 1, \ldots [k/2]\}, \end{cases}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is an input vector, $\mathbf{w} \cdot \mathbf{x} = w_1 x_1 + \ldots + w_n x_n$, $\lambda \in \{0,1\}$ is a binary label of the neuron, $\bar{\lambda}$ is its negation, $\lfloor a \rfloor$ is the floor function (the integer part of the number a), and y is the output value of the neuron.

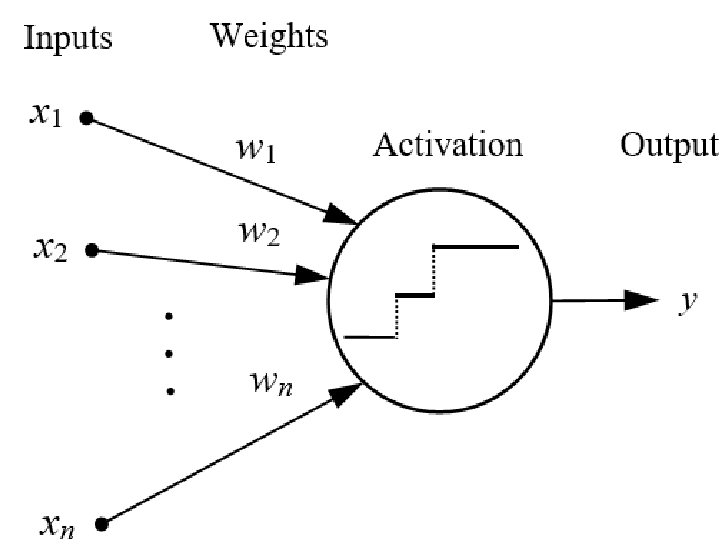
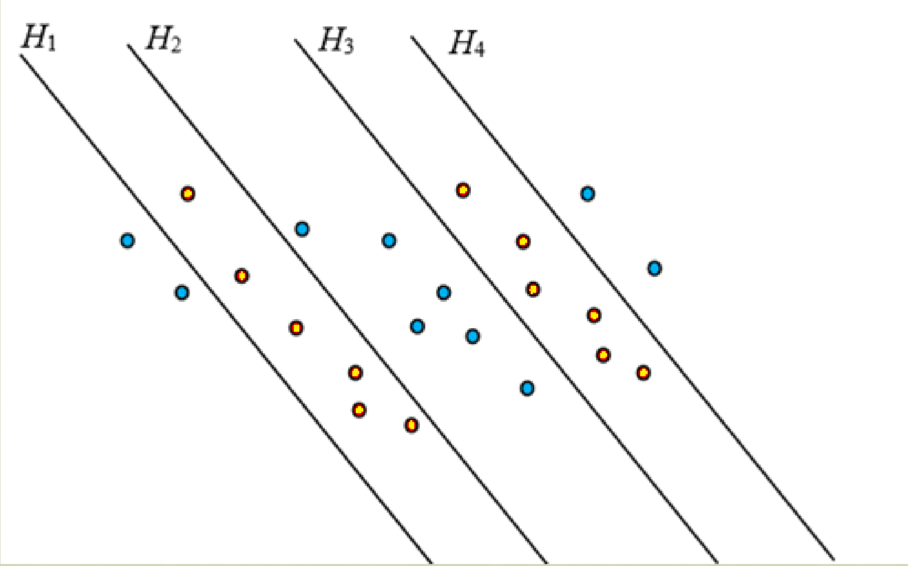
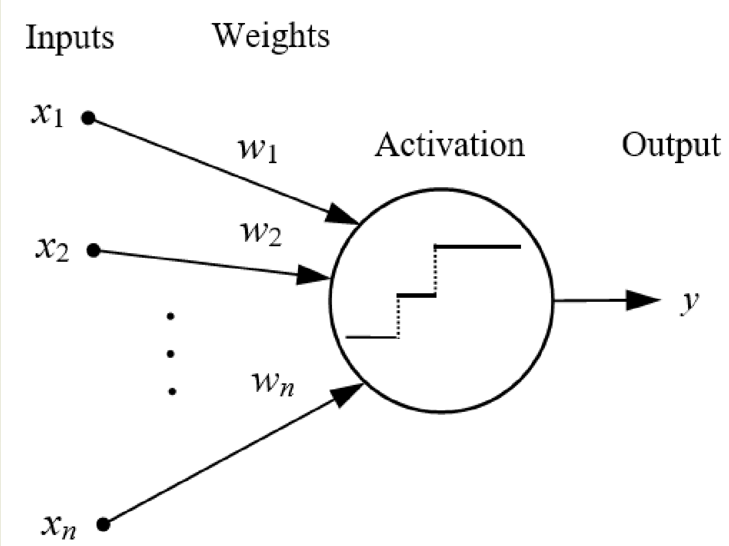
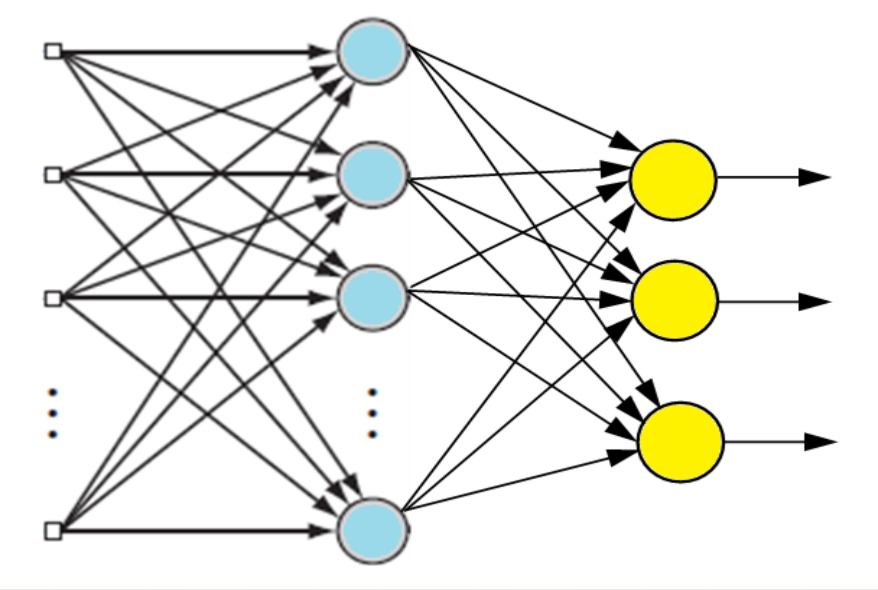


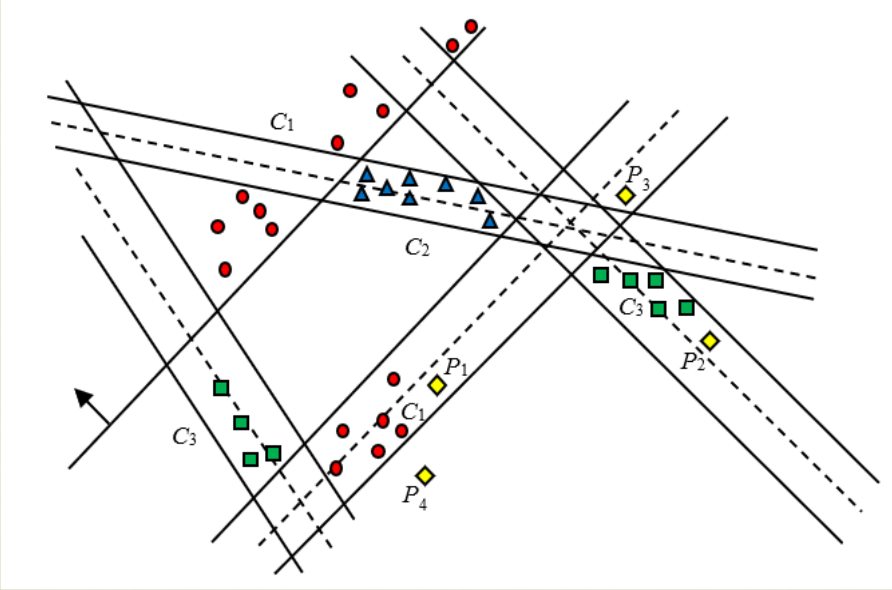
ILLUSTRATION OF THE PERFORMANCE OF MULTITHRESHOLD UNIT





MODEL OF 2-LAYER MULTITHRESHOLD NEURAL NETWORK





NETWORK SYNTHESIS ALGORITHM

MultithresholdSynthesis(X, $\mathbf{y}$, α, p)

Stage 1. Synthesis of the hidden layer

1.1 $Z \leftarrow \{X_1, \ldots, X_l\}$

1.2 $h \leftarrow 0$

1.3 `while` $Z \neq \varnothing$:

1.4 select a random set X_i from Z

1.5 $h \leftarrow h + 1$

1.6 $c[h] \leftarrow i$

1.7 $r \leftarrow \min\{n, |X_i|\}$

1.8 Move r randomly chosen patterns from X_i into the matrix A

1.9 Solve linear system $\mathbf{w} \cdot A^T = \mathbf{1}$

1.10 $\varepsilon \leftarrow \alpha \min\{\, |\mathbf{w} \cdot \mathbf{x} - 1| \mid \mathbf{x} \in X \setminus X_i \}$

1.11 $X_i \leftarrow X_i \setminus \{\, \mathbf{x} \in X_i \mid |\mathbf{w} \cdot \mathbf{x} - 1| < \varepsilon \}$

1.12 $\mathbf{t} \leftarrow (1 - \varepsilon, 1 + \varepsilon)$

1.13 $\lambda \leftarrow 1$

NETWORK SYNTHESIS ALGORITHM

1.14 foreach $\mathbf{a} \in X_i$:

1.15 $\quad s \leftarrow \mathbf{w} \cdot \mathbf{a}$

1.16 $\quad \varepsilon_1 \leftarrow \alpha \min\{\, s - \mathbf{w} \cdot \mathbf{x} \mid \mathbf{x} \in X \setminus X_i,\, \mathbf{w} \cdot \mathbf{x} < s \}$

1.17 $\quad \varepsilon_2 \leftarrow \alpha \min\{\, \mathbf{w} \cdot \mathbf{x} - s \mid \mathbf{x} \in X \setminus X_i,\, \mathbf{w} \cdot \mathbf{x} \geq s \}$

1.18 $\quad B \leftarrow \{\, \mathbf{x} \mid \mathbf{x} \in X_i,\, \varepsilon_1 < \mathbf{w} \cdot \mathbf{x} - s < \varepsilon_2 \}$

1.19 $\quad$ if $|B| \geq p$:

1.20 $\qquad X_i \leftarrow X_i \setminus B$

1.21 $\qquad$ if ε_1 is well-defined: insert $s - \varepsilon_1$ in the ordered vector $\mathbf{t}$

1.22 $\qquad$ else: $\lambda \leftarrow 0$

1.23 $\qquad$ if ε_2 is well-defined: insert $s + \varepsilon_2$ in the ordered vector $\mathbf{t}$

1.24 Add multithreshold neuron $(\mathbf{w}, \mathbf{t}, \lambda)$ in the hidden layer

1.25 if $|X_i| = \varnothing$:

1.26 $\quad$ remove X_i from S

Stage 2. Synthesis of the output layer

2.1 for $i \leftarrow 1$ to l:

2.2 $\quad$ for $j \leftarrow 1$ to h:

2.3 $\qquad$ if $c[h] = i$: $v_{ij} \leftarrow 2$

2.4 $\qquad$ else: $v_{ij} \leftarrow 0$

2.5 $\quad$ Add threshold neuron $(\mathbf{v}_i, 1)$ in the output layer

ESTIMATION OF THE ALGORITHM PERFORMANCE

Proposition. An arbitrary partition of m-set X into l disjoint nonempty sets X_1, ..., X_l of n-dimensional vectors can be performed by a 2-layer multithreshold NN with at most $\lceil m/n \rceil + l$ nodes in the hidden layer and l neurons in the output layer, whereas the network can be synthesized using MultithresholdSynthesis(X, $\mathbf{y}$, α, p) in $O\left(m^3\right)$ time.

EXPERIMENT RESULTS

Accuracy metric

Classifier	Cross-validation		Test set
	Train score	Test score	
7-Nearest Neighbor	65.45	54.43	56.01
Decision tree	100	57.68	60.92
Random forest	100	65.62	67.87
Multilayer perceptron	91.71	63.11	62.13
Support vector machine	67.55	56.77	58.15
2-layer multithreshold NN1	100	49.02	54.28
2-layer multithreshold NN2	93.02	56.24	60.95

Top-2 metric

Classifier	Cross-validation		Test set
	Train score	Test score	
7-Nearest Neighbor	92.2	84.31	85.01
Decision tree	100	61.17	63.88
Random forest	100	91.38	92.05
Multilayer perceptron	98.46	85.6	87.97
Support vector machine	88.84	77.14	77.8
2-layer multithreshold NN1	100	57.11	56.29
2-layer multithreshold NN2	98.02	83.92	84.36

EXPERIMENT RESULTS

- The basic version of multithreshold network NN1 was good memorizer and performed perfectly on the training set.
- The generalization ability of NN1 was very poor. Its performance on the test set was worst for both considered metrics.
- The optimized version NN2 lost a bit in the accuracy on the training set, but performed considerable better on new patterns.
- The proper choice of the α allow us to avoid the great overfitting.
- The test error during cross-validation varies significantly.
- The use of value of α lying inside the half-interval (0, 1] results in the degradation of the generalization capability of the classifier.
- The most typical number of thresholds of the neuron in the hidden layer of NN2 was 3.
- The growth of the value of hyperparameter p results in almost bithreshold hidden layer.
- Using the modified software implementation and additional memory it is possible to decrease the time complexity of MultithresholdSynthesis(X, $\mathbf{y}$, α, p) in Proposition from $O\left(m^3\right)$ to $O\left(m^2\left(n+\log m\right)\right)$.

CONCLUSIONS

- The ways of the use of multithreshold paradigm in neural computation were treated in the paper.
- The new modification of the model of binary-valued multithreshold neural unit was proposed, which allows often to reduce number of thresholds.
- The 2-layer neural network architecture was studied whose hidden layer consists of multithreshold neurons with different number of thresholds.
- The synthesis algorithm was developed for such multithreshold NN. The multiclass classifier was designed on the base of such architecture.

- Experiment results of the performance of multithreshold classifier on “Wine Quality” real-world dataset were presented.
- The impact of algorithm hyperparameters on the classifier ability to make a precise prediction, as well as and the ways to improve it were discussed.
- The main conclusion is that there exists a need in the development of localized modifications of the multithreshold activation function and additional hyperparameters in synthesis algorithm in order to regulate the network performance more precisely.

THANK YOU FOR YOUR ATTENTION