Artificial intelligence in a rugged design based on multi-bit rules

T A Zolotareva
Lipetsk State Pedagogical P.Semenov-Tyan-Shansky University, Lipetsk, 398020, Russia
E-mail: zolotarevatatyana2016@yandex.ru

Abstract. In this paper, the technologies for training large artificial neural networks are considered: the first technology is based on the use of multilayer “deep” neural networks; the second technology involves the use of a “wide” single-layer network of neurons giving 256 private binary solutions. A list of attacks aimed at the simplest one-bit neural network decision rule is given: knowledge extraction attacks and software data modification attacks; their content is considered. All single-bit decision rules are unsafe for applying. It is necessary to use other decision rules. The security of applying neural network decision rules in relation to deliberate hacker attacks is significantly reduced if you use a decision rule of a large number of output bits. The most important property of neural network transducers is that when it is trained using 20 examples of the “Friend” image, the “Friend” output code of 256 bits long is correctly reproduced with a confidence level of 0.95. This means that the entropy of the “Friend” output codes is close to zero. A well-trained neural network virtually eliminates the ambiguity of the “Friend” image data. On the contrary, for the “Foe” images, their initial natural entropy is enhanced by the neural network. The considered works made it possible to create a draft of the second national standard for automatic training of networks of quadratic neurons with multilevel quantizers.

1. Methods

1.1. Learning technologies for large artificial neural networks
Nowadays, two industrial technologies for training large artificial neural networks have been created. The first technology is based on the use of multilayer “deep” neural networks [1-2] and requires millions of examples, recognizable patterns. Training of “deep” neural networks is slow and computationally intensive. Usually hundreds of outputs from intermediate neurons in a “deep” network are summarized by Euclid's last quadratic neuron.

The second technology is standardized in Russia [3] and involves the use of a “wide” single-layer network of neurons giving 256 private binary solutions. Nevertheless, using a standardized algorithm [3], it is possible to automatically and very quickly (in a few seconds) train one neuron with a large number of inputs.

Both of these technologies (the last neuron of “deep” networks) and one neuron trained in a standard way [3] largely coincide in their essence and are illustrated in figure 1.
Trust in artificial intelligence is largely determined by how difficult it is for an attacker to investigate its decision rule and distort this decision rule in their favor or in favor of third parties. Table 1 shows a list of attacks aimed at the simplest one-bit neural network decision rule.

**Table 1.** List of attacks aimed at the simplest one-bit neural network decision rule.

| Content of the attack                                                                 | Aim                        |
|--------------------------------------------------------------------------------------|----------------------------|
| Knowledge extraction attacks from software                                           |                            |
| Copying decision rule software                                                     | Continued attack on AI     |
| Extracting knowledge about the code “Friend”                                        | Continued attack on AI     |
| Interception of the vector parameters of the “Friend” image                        | Continued attack on AI     |
| Extracting knowledge of the location of the last bit of the integrity verification mechanism | Continued attack on AI     |
| Extracting knowledge of the location of the last bit of the decision rule           | Continued attack on AI     |
| Location of neuron connection tables data                                           | Continued attack on AI     |
| Location of data from tables of neuron weighting coefficients                      | Continued attack on AI     |

| Software data modification attacks                                                  | Aim                        |
|--------------------------------------------------------------------------------------|----------------------------|
| Substitution of the last bit of the integrity check mechanism                       | Disabling the AI integrity control mechanism |
| Substitution of the last bit of the decision rule                                   | AI reverse function         |
| Partial or complete substitution of the link table                                  | Deteriorating AI quality    |
| Partial or complete substitution of the weighting coefficients table                | Deteriorating AI quality    |
| Complete substitution of weighting coefficients for another image “Foe-k” in place of the image “Friend” | Another AI function |

The bottom line is that all single-bit decision rules are dangerous to use. It is necessary to use other decision rules.

The security of applying neural network decision rules in relation to deliberate hacker attacks is significantly reduced if we use a decision rule from a large number of output bits (256 bits). The first standard in world practice that regulates such decision rules is the Russian State Standard GOST R 52633.5-2011. The structure of this technical solution is shown in figure 2. The data in the figure reproduce the extremely efficient state of the “BioNeuroAutograph” modeling environment [3]. The neurons of a single-layer network have 16 inputs randomly connected to a vector of 416 observed biometric parameters.
The choice of the output code length of 256 bits is caused by the fact that Windows, Linux, Android operating systems are capable of accepting passwords of such long length (32 random characters in 8-bit encoding). If these operating systems allowed the use of longer passwords and domestic cryptography standards were focused on longer keys, then the length of the output code of neural network structures could be increased.

![Neural network GOST R 52633.5](image)

**Figure 2.** Neural network decision rule that transforms the observed biometric data vector of 416 parameters into 256 bits of the output code.

The most important property of neural network transducers is that when it is trained using 20 examples of the “Friend” image, the “Friend” output code of 256 bits long is correctly reproduced with a confidence level of 0.95. This means that the entropy of the “Friend” output codes is close to zero.

\[
H(\ll c_1, c_2, \ldots, c_{256} \gg) \approx 0.05
\]  

(1)

A well-trained neural network virtually eliminates the ambiguity of the “Friend” image data. On the contrary, for the “Foe” images, their initial natural entropy is enhanced by the neural network:

\[
H(\ll x_1, x_2, \ldots, x_{256} \gg) \approx 25.05
\]  

(2)

This effect is reproduced due to the structural solutions underlying the GOST R 52633.5 algorithm. As can be seen from figure 3, all hyperplanes dividing the 416-dimensional space do not intersect the ellipse of the data distribution of the “Friend” image. On the contrary, all the “Foe” images are repeatedly intersected by the hyperplanes of artificial neurons. For this reason, minor changes in the examples of the “Foe” image lead to a strong change in some of the bits of the output code of the neural network.

It should also be noted that not only the data of the distribution of examples of the “Friend” image, but also the data with the inverse distribution of the parameters of the “Friend” image, are stable.
2. Results

Unfortunately, quadratic neurons of Mahalanobis or quadratic neurons of Euclid, by analogy with GOST R 52633.5, may well change the output states of 256 neurons, under the “Friend” key code. However, such biometrics-to-code converters cannot be used. The reason for this is the lack of hashing effect in quadratic neurons for these “Foe” images.

It is enough to send any “Foe” image to the input of the network of quadratic neurons, and we get a very stable output code inverse to the “Friend” code with very low entropy:

\[ H(\ll x_1, x_2, \ldots, x_{256} \gg) \approx H(\ll c_1, c_2, \ldots, c_{256} \gg) \approx 0.05 \]  

It is because of the lack of hashing in quadratic neuron networks (3) that they have not been previously used in secure biometrics. Previously, preference was given to networks of neurons with linear data accumulation [4] only because they initially have hashing for “Foe” images and, accordingly, prevent attackers from extracting knowledge from software implementations of decision rules.

Since the middle of the last century, during the next 60 years of research [1-2], it was customary to use artificial neurons with monotonic excitation functions or binary quantizers. The use of such simplified models of artificial neurons was understandable at the initial stages of research, but three generations of researchers of artificial neurons got used to them and stopped noticing the progress of 60 years of research by physiologists [5]. According to the real experimental data of physiologists, human and animal brains do not have neurons with binary states. The transmission of data from one neuron to another is transmitted by several types of neurotransmitters. It is reasonable to assume that the presence of one neurotransmitter of data transmission should give two states (“no mediator”, “with mediator”). The presence of two neurotransmitters can (should) lead to the appearance of neurons with data quantization for three or four output states. Nowadays, physiologists know dozens of neurotransmitters [5], which is equivalent to the operation of natural neurons in the mode of using quantizers with dozens of states. It is possible that the number system used by our brains has a base (module of operations) that is higher than the decimal base we are used to. Most likely, natural neural network figurative computers perform operations in those number systems (with the same number of quantum states) in which they previously performed their self-learning. When there is too much data, there is a natural increase in the number of neurons and the number of their output quantum states.

Increasing the number of output states of neuron quantizers is one of the simplest ways to increase the entropy of output code states. In this case, it does not really matter in what space before this the neuron performed the accumulation of data (in linear or quadratic). It is important that quadratic neurons with quantizers having three or more output states acquire the ability to hash the data of the “Foe” images. This was first shown for quantizers with three output states located at the output of the adders of Cramer-von Mises neurons. It was Cramer-von Mises probabilistic neurons that made it possible to prove the possibility of implementing the hashing mode in quadratic neurons. That is, a real opportunity
has opened up to take advantage of quadratic neurons in terms of the separability of biometric images in comparison with the already used linear neurons.

All of the above works made it possible to create a draft of the second national standard for automatic training of networks of quadratic neurons with multilevel quantizers. The training method proposed in the draft of the new standard is illustrated in figure 4.

![Figure 4](image)

**Figure 4.** Data at the output of a quadratic neuron with an 8-level output quantizer (only 5 quantizer levels actually work).

At the first stage of neuron tuning, training is performed. Next, several equally spaced thresholds of the multilevel quantizer are selected. In figure 4, these thresholds form a series k = {0, 2, 4, 6, 8, 10, 12, 14}. At the next stage of training, the data of the “Friend” image is transferred to one of the randomly selected intervals of the multilevel quantizer.

It should be noted that the multilevel quantizer in figure 4 is not monotonic. The absence of monotonicity of the levels of the quantizer’s output states is technically extremely advantageous, since it significantly increases the hashing properties of quadratic neurons.

As a result, it becomes possible to significantly enhance the protective properties of neural network converters of biometrics into code due to the realization of several topological advantages of quadratic neurons in comparison with already well-studied networks of linear neurons. All this can be technically implemented if we find ways to reliably distinguish between dependent and independent data using 20 examples of a small training sample of the “Friend” image.

References

[1] Junwei Sun, Xiao Xiao and Yanfeng Wang 2021 Memristor-based Hopfield network circuit for recognition and sequencing application *AEU - International Journal of Electronics and Communications* 295

[2] Alberto Prieto, Beatriz Prieto and Ignacio Rojas 2016 Neural networks: An overview of early research, current frameworks and new challenges *Neurocomputing* 632

[3] Soheila Nazari, Masoud Amiri and Mahmood Amiri 2015 Multiplier-less digital implementation of neuron–astrocyte signalling on FPGA *Neurocomputing* 1125

[4] Soheila Nazari and Karim Faez 2018 Spiking pattern recognition using informative signal of image and unsupervised biologically plausible learning *Neurocomputing* 458

[5] Arun A, Rameshkumar K and Sumesh A 2018 Arun Tool Condition Monitoring Of Cylindrical Grinding Process Using Acoustic Emission Sensor *Materials Today: Proceedings* 785

[6] Bosque G, del Campo I and Echanobe J 2014 Fuzzy systems, neural networks and neuro-fuzzy systems: A vision on their hardware implementation and platforms over two decades *Engineering Applications of Artificial Intelligence* 358

[7] Bosque G, del Campo I and Echanobe J 2014 Fuzzy systems, neural networks and neuro-fuzzy systems: A vision on their hardware implementation and platforms over two decades *Engineering Applications of Artificial Intelligence* 568

[8] Misbah Shafi, Rakesh Kumar Jha and Manish Sabraj 2020 A survey on security issues of 5G NR: Perspective of artificial dust and artificial rain *Journal of Network and Computer Applications* 566