The Power of Extra Analog Neuron

Jiří Šíma*

Institute of Computer Science, Academy of Sciences of the Czech Republic
P. O. Box 5, 18207 Prague 8, Czech Republic, sima@cs.cas.cz

Abstract. In the effort to refine the analysis of computational power of neural nets between integer and rational weights we study a hybrid binary-state network with an extra analog unit. We introduce a finite automaton with a register which is shown to be computationally equivalent to such a network. The main result is a sufficient condition for a language accepted by this automaton to be regular which is based on the new concept of a quasi-periodic power series. These preliminary results suggest an interesting connection with the active research field on the expansions of numbers in non-integer bases which seems to be a fruitful area for further research including many important open problems.

Keywords: Neural computing, analog state, beta-expansion

1 Introduction

The computational power of neural networks with the saturated-linear activation function[1] depends on the descriptive complexity of their weight parameters [25, 30]. Neural nets with *integer* weights corresponding to binary-state networks coincide with finite automata [3, 9, 11, 16, 29, 31]. *Rational* weights make the analog-state networks computationally equivalent to Turing machines [11, 27], and thus (by a real-time simulation [27]) polynomial-time computations of such networks are characterized by the complexity class P. Moreover, neural nets with arbitrary *real* weights can even derive "super-Turing" computational capabilities [25, 26]. For example, their polynomial-time computations correspond to the nonuniform complexity class P/poly while any I/O mapping (including undecidable problems) can be computed within exponential time. In addition, a proper hierarchy of nonuniform complexity classes between P and P/poly has been established for polynomial-time computations of neural nets with increasing Kolmogorov complexity of real weights [4].

It follows that our understanding of the computational power of neural networks is satisfactorily fine-grained when changing from rational to arbitrary real weights. In contrast, there is still a gap between integer and rational weights which results in a jump from regular to recursive languages in the Chomsky hierarchy. It appears that Turing machines can be simulated by the neural networks

* Research was supported by the projects GA ČR P202/10/1333 and RVO: 67985807.

[1] The results are valid for more general classes of activation functions [14, 24, 28, 32] including the logistic function [13].

that, apart from binary-state neurons interconnected via integer weights, include only *two* analog-state units with rational weights implementing two stacks of pushdown automata, a model equivalent to Turing machines [27]. A natural question arises: what is the computational power of hybrid binary-state networks with one extra analog unit having rational weights? Our investigation which was originally motivated by the quest of refining the analysis along this direction, has revealed interesting connections with other active research fields such as representations of numbers in non-integer bases [1, 2, 5, 7, 8, 15, 19, 20, 22, 23] and automata with multiplication [6, 10, 12, 17, 21]. In addition, our analysis leads to interesting open problems and even to new concepts which are worth investigating on their own.

The present paper which initiates our preliminary study, is organized as follows. In Section 2, we give a brief review of basic definitions concerning the language acceptors based on a hybrid model of binary-state neural networks with an extra analog unit. In Section 3, we introduce a new notion of a finite automaton with a register whose domain is partitioned into a finite number of intervals, each associated with a local state-transition function. This automaton is shown to be computationally equivalent by mutual simulations to a neural network with an analog unit. Our main technical result in Section 4 provides a sufficient condition when a finite automaton with a register accepts a regular language, which is based on the new concept of a quasi-periodic power series. In section 5, related results on so-called β-expansions of numbers in non-integer bases are surveyed and emerging directions for ongoing research are discussed.

2 Neural Language Acceptors with Extra Analog Unit

We will specify a hybrid model of a *binary-state neural network with an analog unit (NN1A)* N which will be used as a formal language acceptor. The network consists of s *units (neurons)*, indexed as $V = \{1, \ldots, s\}$, where s is called the network *size*. All the units in N are assumed to be binary-state *perceptrons (i.e. threshold gates)* except for the last sth neuron which is an *analog unit*. The neurons are connected into a directed graph representing the *architecture* of N, in which each edge (i, j) leading from unit i to j is labeled with a rational *weight* $w(i, j) = w_{ji} \in \mathbb{Q}$. The absence of a connection within the architecture corresponds to a zero weight between the respective neurons, and vice versa.

The *computational dynamics* of N determines for each unit $j \in V$ its *state (output)* $y_j^{(t)}$ at discrete time instants $t = 0, 1, 2, \ldots$. The states $y_j^{(t)}$ of the first $s - 1$ perceptrons $j \in V \setminus \{s\}$ are binary values from $\{0, 1\}$, whereas $y_s^{(t)}$ of analog unit $s \in V$ is a rational number from the unit interval $\mathbb{I} = [0, 1] \cap \mathbb{Q}$. This establishes the *network state* $\mathbf{y}^{(t)} = (y_1^{(t)}, \ldots, y_s^{(t)}) \in \{0, 1\}^{s-1} \times \mathbb{I}$ at each discrete time instant $t \geq 0$. At the beginning of a computation, the neural network N is placed in an *initial state* $\mathbf{y}^{(0)}$ which may also include an external input. At discrete time instant $t \geq 0$, an *excitation* of any neuron $j \in V$ is defined as $\xi_j^{(t)} = \sum_{i=0}^{s} w_{ji} y_i^{(t)}$, including a rational *bias* value $w_{j0} \in \mathbb{Q}$ which can be viewed as the weight $w(0,j)$ from a formal constant unit input $y_0^{(t)} \equiv 1$. At the next instant $t+1$, the neurons $j \in \alpha_{t+1}$ from a selected subset $\alpha_{t+1} \subseteq V$ compute their new outputs $y_j^{(t+1)} = \sigma_j(\xi_j^{(t)})$ in parallel by applying an *activation function* $\sigma_j : \mathbb{R} \longrightarrow \mathbb{R}$ to $\xi_j^{(t)}$, whereas $y_j^{(t+1)} = y_j^{(t)}$ for the remaining units $j \in V \setminus \alpha_{t+1}$. For perceptrons $j \in V \setminus \{s\}$ with binary states $y_j \in \{0,1\}$ the *Heaviside* activation function $\sigma_j(\xi) = \sigma_H(\xi)$ is used where $\sigma_H(\xi) = 1$ for $\xi \geq 0$ and $\sigma_H(\xi) = 0$ for $\xi < 0$, while the analog-state unit $s \in V$ employs the *saturated-linear* function $\sigma_s(\xi) = \sigma_L(\xi)$ where

$$\sigma_L(\xi) = \begin{cases} 1 & \text{for } \xi \geq 1 \\ \xi & \text{for } 0 < \xi < 1 \\ 0 & \text{for } \xi \leq 0\,. \end{cases} \tag{1}$$

In this way, the new network state $\mathbf{y}^{(t+1)}$ at time $t+1$ is determined.

Without loss of efficiency [18] we assume synchronous computations for which the sets α_t, defining the computational dynamics of N, are predestined deterministically. Usually, sets α_t correspond to layers in the architecture of N which are updated one by one (e.g., a feedforward subnetwork). In particular, we use a systematic periodic choice of α_t so that $\alpha_{t+d} = \alpha_t$ for any $t \geq 0$ where an integer parameter $d \geq 1$ represents the number of updates within one *macroscopic time* step (e.g., d is the number of layers). We assume that the analog unit $s \in V$ is updated exactly once in every macroscopic time step, say $s \in \alpha_{d\tau}$ for every $\tau \geq 1$.

The computational power of neural networks has been studied analogously to the traditional models of computations so that the networks are exploited as acceptors of formal languages $L \subseteq \{0,1\}^*$ over the binary alphabet. For the finite networks the following I/O protocol has been used [3, 4, 9, 11, 24–27, 30, 31]. A binary input word (string) $\mathbf{x} = x_1 \ldots x_n \in \{0,1\}^n$ of arbitrary length $n \geq 0$ is sequentially presented to the network bit by bit via the first, so-called *input neuron* $1 \in V$. The state of this unit is externally set (and clamped) to the respective input bits at microscopic time instants, regardless of any influence from the remaining neurons in the network, that is, $y_1^{(d(\tau-1)+k)} = x_\tau$ for $\tau = 1, \ldots, n$ and every $k = 0 \ldots, d-1$ where an integer $d \geq 1$ is the *time overhead* for processing a single input bit which coincides with the microscopic time step. Then, the second, so-called *output neuron* $2 \in V$ signals at microscopic time instant n whether the input word belongs to underlying language L, that is, $y_2^{(dn)} = 1$ for $\mathbf{x} \in L$ whereas $y_2^{(dn)} = 0$ for $\mathbf{x} \notin L$. Thus, a language $L \subseteq \{0,1\}^*$ is *accepted by NN1A* N, which is denoted by $L = L(N)$, if for any input word $\mathbf{x} \in \{0,1\}^*$, $\mathbf{x}$ is accepted by N iff $\mathbf{x} \in L$.

3 Finite Automata with a Register

We introduce a (deterministic) *finite automaton with a register (FAR)* which is formally a nine-tuple $A = (Q, \Sigma, \{I_1, \ldots, I_p\}, a, (\Delta_1, \ldots, \Delta_p), \delta, q_0, z_0, F)$ where, as usual, Q is a finite set of automaton *states* including a *start (initial) state* $q_0 \in Q$ and a subset $F \subseteq Q$ of *accept (final) states*. We assume $\Sigma = \{0,1\}$ to be a binary *input alphabet*. In addition, the automaton is augmented with a register which stores a rational number $z \in \mathbb{I} = [0,1] \cap \mathbb{Q}$. Domain $\mathbb{I}$ is partitioned into a finite number of intervals $I_1, \ldots, I_p$, possibly of different types: open, closed, half-closed, or degenerate (containing a single point) bounded intervals with rational endpoints. Each such an interval I_r is associated with a usual local state-transition function $\delta_r : Q \times \Sigma \longrightarrow Q$ which is employed if the current register value z falls into this interval I_r.

Moreover, we have a rational *shift* function $\Delta_r : Q \times \Sigma \longrightarrow \mathbb{Q}$ for each interval I_r, $r = 1, \ldots, p$. The register is initialized to a *start (initial)* value $z_0 \in \mathbb{I}$, and during each state transition, its value $z \in \mathbb{I}$ is updated to $\sigma_L(az + \Delta_r(q,x)) \in \mathbb{I}$ by applying a linear mapping with saturation (1) having a fixed slope $a \in \mathbb{Q}$ called *multiplier* and an y-intercept $\Delta_r(q,x) \in \mathbb{Q}$ given by the shift function Δ_r for $z \in I_r$ which depends on current state $q \in Q$ and input bit $x \in \Sigma$. In summary, for current state $q \in Q$, register value $z \in \mathbb{I}$, and input bit $x \in \Sigma$, the global *state-transition function* $\delta : Q \times \mathbb{I} \times \Sigma \longrightarrow Q \times \mathbb{I}$ produces the new state and the new register value of automaton A as follows:

$$\delta(q,z,x) = (\delta_r(q,x), \sigma_L(az + \Delta_r(q,x))) \quad \text{if } z \in I_r\,. \tag{2}$$

A binary input word $\mathbf{x} \in \Sigma^*$ is accepted by A if automaton A, starting at initial state q_0 with start register value z_0, reaches a final state $q \in F$ by a sequence of state transitions according to (2) while reading the input $\mathbf{x}$ from left to right. A language $L \subseteq \{0,1\}^*$ is *accepted by FAR A*, which is denoted by $L = L(A)$, if for any input word $\mathbf{x} \in \Sigma^*$, $\mathbf{x}$ is accepted by A iff $\mathbf{x} \in L$. The concept of FAR is reminiscent of today's already classical definition of finite automaton with multiplication [10]. In the following theorems, we will show by mutual simulations that the binary-state neural networks with analog unit introduced in Section 2 are computationally equivalent to the finite automata with register.

Theorem 1. *For any binary-state neural network with an analog unit, there is a finite automaton with a register such that both accept the same language.*

Proof. Let $L \subseteq \{0,1\}^*$ be a language accepted by NN1A N, that is, $L = L(N)$. We will construct a FAR A such that $L(A) = L$. Let $Q = \{0,1\}^{s-2}$ be a finite set of automaton states corresponding to all possible binary states $(y_2^{(d\tau)}, \ldots, y_{s-1}^{(d\tau)})$ of neurons in $V \setminus \{1,s\}$ at macroscopic time $\tau \geq 0$, excluding the input and analog unit. The start state $q_0 = (y_2^{(0)}, \ldots, y_{s-1}^{(0)}) \in Q$ of A is defined using the initial state of N and $F = \{1\} \times \{0,1\}^{s-3}$ represents the set of accept states.

At any time instant $t \geq 0$, the computational dynamics of N ensures $y_j^{(t+1)} = 1$ iff $\sum_{i=0}^{s-1} w_{ji} y_i^{(t)} + w_{js} y_s^{(t)} \geq 0$ for a non-input binary-state neuron $j \in \{2 \ldots, s-1\} \cap \alpha_{t+1}$. For $w_{js} \neq 0$, this condition can be rewritten as

$$y_j^{(t+1)} = 1 \text{ iff } \left(w_{js} > 0 \;\&\; y_s^{(t)} \geq c_j\left(\tilde{\mathbf{y}}^{(t)}\right)\right) \vee \left(w_{js} < 0 \;\&\; y_s^{(t)} \leq c_j\left(\tilde{\mathbf{y}}^{(t)}\right)\right) \tag{3}$$

where $\tilde{\mathbf{y}}^{(t)} = (y_1^{(t)}, \ldots, y_{s-1}^{(t)})$ and $c_j(\mathbf{y}) = (-\sum_{i=0}^{s-1} w_{ji} y_i)/w_{js} \in \mathbb{Q}$ for $\mathbf{y} \in \{0,1\}^{s-1}$. Let $C = \{(c_j(\mathbf{y}), 1-\sigma_H(w_{js})) \in \mathbb{I} \times \{0,1\} \mid 1 < j < s$ such that $w_{js} \neq 0$,

$\mathbf{y} \in \{0,1\}^{s-1}\} \cup \{(0,0),(1,1)\}$ be a finite set of all possible values $c_j(\mathbf{y}) \in \mathbb{I}$ associated with the opposite signs of corresponding weights w_{js}, which is extended with the endpoints $0,1$ of $\mathbb{I}$. We sort the elements of C lexicographically as $(0,0) = (c_1,s_1) < (c_2,s_2) < \ldots < (c_{p+1},s_{p+1}) = (1,1)$, which defines the partition of $\mathbb{I}$ to rational intervals $I_1,\ldots,I_p$ as $I_r = [c_r,c_{r+1})$ if $s_r = 0 \,\&\, s_{r+1} = 0$, $I_r = [c_r,c_{r+1}]$ if $s_r = 0 \,\&\, s_{r+1} = 1$, $I_r = (c_r,c_{r+1})$ if $s_r = 1 \,\&\, s_{r+1} = 0$, and $I_r = (c_r,c_{r+1}]$ if $s_r = 1 \,\&\, s_{r+1} = 1$, for $r = 1,\ldots,p$. It follows from (3) that for any interval I_r $(1 \le r \le p)$ of this partition, for every neuron $j = 2,\ldots,s-1$ and for any $\tilde{\mathbf{y}}^{(t)} \in \{0,1\}^{s-1}$, the inequality $\sum_{i=0}^{s-1} w_{ji} y_i^{(t)} + w_{js} y_s^{(t)} \ge 0$ either holds for all $y_s^{(t)} \in I_r$ or it is not satisfied for all $y_s^{(t)} \in I_r$. This means that $y_s^{(t)} \in \mathbb{I}$ influences the state $y_j^{(t+1)}$ only by its membership to particular interval I_r and not by its exact analog value.

We can define local state-transition functions $\delta_r : Q \times \Sigma \longrightarrow Q$ of A for each interval I_r, $r = 1,\ldots,p$. Given an automaton state $q = (y_2^{(d(\tau-1))},\ldots,y_{s-1}^{(d(\tau-1))}) \in Q$ corresponding to the network state $\mathbf{y}^{(d(\tau-1))}$ at microscopic time $\tau - 1$ when a current input bit $y_1^{(d(\tau-1))} = x_\tau \in \Sigma$ is read, and let $y_s^{(d(\tau-1))} \in I_r$, we know that $y_1^{(d(\tau-1)+k)} = y_1^{(d(\tau-1))}$ and $y_s^{(d(\tau-1)+k)} = y_s^{(d(\tau-1))}$ for every $k = 1,\ldots,d-1$, as the input and analog units are updated only at microscopic time instants. Hence, for this interval I_r, the neuron states $(y_2^{(d\tau))},\ldots,y_{s-1}^{(d\tau))}) = q' \in Q$ depend only on state $q \in Q$ and input bit $x_\tau \in \Sigma$ using the computational dynamics of N, which define $\delta_r(q,x_\tau) = q'$.

Finally, the register of A is initialized as $z_0 = y_s^{(0)} \in \mathbb{I}$. We define the multiplier $a = w_{ss} \in \mathbb{Q}$ and the shift functions $\Delta_r : Q \times \Sigma \longrightarrow \mathbb{Q}$ for $r = 1,\ldots,p$ as $\Delta_r(q,x) = \sum_{i=0}^{s-1} w_{si} y_i^{(d\tau-1)} \in \mathbb{Q}$ for $q = (y_2^{(d(\tau-1))},\ldots,y_{s-1}^{(d(\tau-1))}) \in Q$ and $x = y_1^{(d(\tau-1))} = y_1^{(d\tau-1)} \in \Sigma$, and $y_s^{(d(\tau-1))} = y_s^{(d\tau-1)} \in I_r$, which is a correct definition since the network state $\mathbf{y}^{(d\tau-1)}$ is uniquely determined by the state $\mathbf{y}^{(d(\tau-1))}$ at the last microscopic time instant $\tau - 1$ using the computational dynamics of N. By induction on microscopic time τ, the register of A stores the current state $y_s^{(d\tau)}$ of analog unit $s \in V$, as its value $z = y_s^{(d(\tau-1))} = y_s^{(d\tau-1)} \in I_r$ is updated to $\sigma_L(az + \Delta_r(q,x)) = \sigma_s\left(w_{ss} y_s^{(d\tau-1)} + \sum_{i=0}^{s-1} w_{si} y_i^{(d\tau-1)}\right) = y_s^{(d\tau)}$ according to (2) and (1). This completes the definition of global state-transition function δ which ensures that A simulates N. $\square$

Theorem 2. *For any finite automaton with a register, there is a binary-state neural network with an analog unit accepting the same language.*

Proof. Let $L \subseteq \{0,1\}^*$ be a language accepted by FAR $A = (Q, \Sigma, \{I_1,\ldots,I_p\}, a, (\Delta_1,\ldots,\Delta_p), \delta, q_0, z_0, F)$, that is, $L = L(A)$. We will construct a NN1A N such that $L(N) = L$. Apart from the input, output, and analog neurons $\{1,2,s\}$, the set of neurons V contains four types of units corresponding to the automaton states from Q, to the given partition $I_1,\ldots,I_p$ of domain $\mathbb{I}$, to all triples from $Q \times \Sigma \times \{I_1,\ldots,I_p\}$, and to the endpoints $0 \le c_2 \le \cdots \le c_p \le 1$ of rational intervals from the partition (excluding the left endpoint $c_1 = 0$ of I_1 and the right endpoint $c_{p+1} = 1$ of I_p), respectively. For simplicity, we will identify the names of neurons with these objects, e.g. c_r has two different meanings, once denoting neuron $c_r \in V$ and other times standing for rational number $c_r \in \mathbb{I}$. The initial network state $\mathbf{y}^{(0)} \in \{0,1\}^{s-1} \times \mathbb{I}$ is defined as an almost null vector except for the input unit receiving the first input bit $y_1^{(0)} = x_1 \in \Sigma = \{0,1\}$, the output neuron whose state $y_2^{(0)} = 1$ iff $q_0 \in F$, the neuron corresponding to the initial automaton state $q_0 \in Q$ with output $y_{q_0}^{(0)} = 1$, and the analog unit implementing the register initialized with its start value $y_s^{(0)} = z_0$.

Each microscopic time step of N is composed of $d = 4$ updates. At the first time instant $4(\tau - 1) + 1$ within the microscopic step $\tau \geq 1$, each neuron c_r $(1 < r \leq p)$ corresponding to the left endpoit of I_r fires, i.e. $y_{c_r}^{(4(\tau-1)+1)} = 1$ iff either $y_s^{(4(\tau-1))} \geq c_r$ for left-closed interval I_r or $y_s^{(4(\tau-1))} \leq c_r$ for left-open interval I_r, which is implemented by weights $w(s, c_r) = 1$ and biases $w(0, c_r) = -c_r$ for left-closed I_r, and $w(s, c_r) = -1$ and $w(0, c_r) = c_r$ for right-closed I_r, for every $r = 2, \ldots, p$. Thus, $\alpha_{4(\tau-1)+1} = \{c_2, \ldots, c_r\} \subseteq V$. At the second time instant $4(\tau - 1) + 2$, neuron I_r $(1 \leq r \leq p)$ representing the interval I_r from the partition of $\mathbb{I}$ fires, i.e. $y_{I_r}^{(4(\tau-1)+2)} = 1$ iff the current register value falls in I_r, that is, iff $y_s^{(4(\tau-1))} = y_s^{(4(\tau-1)+1)} \in I_r$. This is implemented by the following weights: $w(c_r, I_r) = 1$ if I_r is left-closed whereas $w(c_r, I_r) = -1$ if I_r is left-open, for $r = 2, \ldots, p$; $w(c_{r+1}, I_r) = 1$ if I_r is right-closed whereas $w(c_{r+1}, I_r) = -1$ if I_r is right-open, for $r = 1, \ldots, p-1$; $w(0, I_r) = -2$ if I_r is closed, $w(0, I_r) = 0$ if I_r is open, and $w(0, I_r) = -1$ otherwise, for $r = 2, \ldots, p-1$, while the biases of units I_1 and I_p having only one incoming edge are by 1 greater than those defined for $I_2, \ldots, I_{p-1}$. Thus, $\alpha_{4(\tau-1)+2} = \{I_1, \ldots, I_r\} \subseteq V$.

At the third time instant $4(\tau - 1) + 3$, units in $\alpha_{4(\tau-1)+3} = Q \times \Sigma \times \{I_1, \ldots, I_p\} \subseteq V$ are updated so that the only firing neuron $(q, x, I_r) \in V$ among $\alpha_{4(\tau-1)+3}$ indicates the current triple of state $q \in V$, input bit $x \in \Sigma = \{0,1\}$, and the interval I_r such that $y_s^{(4(\tau-1))} \in I_r$. For any $q \in Q$ and every $r = 1, \ldots, p$, this is simply implemented by weights $w(q, (q, x, I_r)) = w(I_r, (q, x, I_r)) = 1$ for any $x \in \Sigma$, $w(1, (q, 1, I_r)) = 1$, $w(1, (q, 0, I_r)) = -1$, and biases $w(0, (q, 1, I_r)) = -3$, $w(0, (q, 0, I_r)) = -2$. At the next time instant 4τ when $\alpha_{4\tau} = Q \cup \{2, s\} \subseteq V$, the new automaton state is computed while the output neuron signals whether this state is accepting. For any $q, q' \in Q$, $x \in \Sigma$, and $r = 1, \ldots, p$, we define the weight $w((q, x, I_r), q') = 1$ iff $\delta_r(q, x) = q'$, and the bias $w(0, q') = -1$, while $w((q, x, I_r), 2) = 1$ iff $q \in F$, and $w(0, 2) = -1$. Finally, the register value is properly updated according to (2) using the weights $w_{ss} = a$ and $w((q, x, I_r), s) = \Delta_r(q, x)$ for any $q \in Q$, $x \in \Sigma$, and every $r = 1, \ldots, p$. This completes the construction of network N simulating FAR A. $\square$

4 A Sufficient Condition for Accepting Regular Languages

In this section, we prove a sufficient condition when a finite automaton with a register accepts a regular language. For this purpose, we introduce a new concept of a quasi-periodic power series. We say that a power series $\sum_{k=0}^{\infty} b_k a^k$

is eventually *quasi-periodic* with *maximum period* $M \geq 1$ and *period sum* P if there is an increasing infinite sequence of its term indices $0 \leq k_1 < k_2 < k_3 < \cdots$ such that $0 < m_i = k_{i+1} - k_i \leq M$ and for every $i \geq 1$, $P_i = (\sum_{k=0}^{m_i-1} b_{k_i+k} a^k)/(1 - a^{m_i}) = P$ where k_1 is the length of preperiodic part, that is, for any $0 \leq k_0 < k_1$, $P_0 \neq P$. For example, $\sum_{k=1}^{\infty} b_k a^k$ is eventually quasi-periodic with maximum period $m \geq 1$ if associated sequence $(b_k)_{k=1}^{\infty}$ is eventually periodic, that is, there exists $k_1 \geq 0$ such that $b_k = b_{k+m}$ for every $k \geq k_1$. For $|a| < 1$, one can calculate the sum of any eventually quasi-periodic power series as $\sum_{k=1}^{\infty} b_k a^k = \sum_{k=0}^{k_1-1} b_k a^k + \sum_{k=k_1}^{\infty} b_k a^k$ where $\sum_{k=k_1}^{\infty} b_k a^k = \sum_{i=1}^{\infty} a^{k_i} \sum_{k=0}^{m_i-1} b_{k_i+k} a^k = P \cdot \sum_{i=1}^{\infty} a^{k_i}\,(1 - a^{m_i})$, which gives

$$\sum_{k=1}^{\infty} b_k a^k = \sum_{k=0}^{k_1-1} b_k a^k + a^{k_1} P \tag{4}$$

since the absolutely convergent series $\sum_{i=1}^{\infty} a^{k_i}(1 - a^{m_i}) = \sum_{i=1}^{\infty}(a^{k_i} - a^{k_{i+1}})$ sums up to a^{k_1}. It follows that the sum (4) of eventually quasi-periodic power series does not change if any quasi-repeating block $b_{k_i}, b_{k_i+1}, \ldots, b_{k_{i+1}-1}$ satisfying $P_i = P$ is removed from associated sequence $(b_k)_{k=1}^{\infty}$ or if it is inserted in between two other quasi-repeating blocks, which means that these quasi-repeating blocks can also be permuted arbitrarily.

Theorem 3. *Let $A = (Q, \Sigma, \{I_1, \ldots, I_p\}, a, (\Delta_1, \ldots, \Delta_p), \delta, q_0, z_0, F)$ be a finite automaton with a register satisfying $|a| \leq 1$. Denote by $C \subseteq \mathbb{I}$ the finite set of all endpoints of rational intervals $I_1, \ldots, I_p$ and let $B = \bigcup_{r=1}^{p} \Delta_r(Q \times \Sigma) \cup \{0, 1, z_0\} \subseteq \mathbb{Q}$ be the finite set of all possible shifts including $0, 1$, and the initial register value. If every series $\sum_{k=0}^{\infty} b_k a^k \in C$ with all $b_k \in B$ is eventually quasi-periodic, then $L = L(A)$ is a regular language.*

Proof. We will construct a conventional finite automaton $A' = (Q', \Sigma, \delta', q_0', F')$ with binary input alphabet $\Sigma = \{0, 1\}$ simulating FAR A so that $L(A') = L$, which shows that L is regular. According to (2) and (1), a current register value

$$z = \sum_{k=0}^{h} b_k a^k \in \mathbb{I} \tag{5}$$

is uniquely determined by the complete history of shifts $b_0, b_1, \ldots, b_h \in B$ since the last time instant when either the register was initialized with start value $b_h = z_0$ or its value saturated at $b_h = 0$ or $b_h = 1$. For $a = 0$ or $|a| = 1$, the set of all possible register values proves to be finite, and henceforth assume $0 < |a| < 1$.

Let $C' = C \cap \{\sum_{k=0}^{\infty} b_k a^k \,|\, \text{all } b_k \in B\} = \{c_1, \ldots, c_\gamma\}$ be a subset of the interval endpoints from C that are reached by eventually quasi-periodic series according to the assumption of the theorem, where $\gamma = |C'|$. We choose an integer $\kappa' \geq 0$ so that each such series $\sum_{k=0}^{\infty} b_k a^k \in C'$ meets $k_1 + 2M \leq \kappa' + 1$ where k_1 is the length of its preperiodic part and M is its maximum period, while we set $\kappa' = 0$ if $\gamma = 0$. It follows that one can decide whether $\sum_{k=0}^{\infty} b_k a^k \notin C$ with all $b_k \in B$, based only on the first $\kappa' + 1$ terms $b_0, b_1, \ldots, b_{\kappa'}$. We observe that there exists an integer $\kappa \geq \kappa'$ such that for every series $\sum_{k=0}^{\infty} b_k a^k \notin C$ with all $b_k \in B$, the interval $I(b_0, b_1, \ldots, b_\kappa) = [z_\kappa + \sum_{k=\kappa+1}^{\infty} \min_{b\in B}\left(ba^k\right), z_\kappa + \sum_{k=\kappa+1}^{\infty} \max_{b\in B}\left(ba^k\right)]$ where $z_\kappa = \sum_{k=0}^{\kappa} b_k a^k$, does not contain any $c \in C$, since the opposite would force $c = \sum_{k=0}^{\infty} b_k a^k$ by Cantor's intersection theorem.

A finite set $Q' = Q \times B^\kappa \times \{<, =, >\}^\gamma$ is now composed of the states of A which are extended with a *limited* history of register shifts $b_0, b_1, \ldots, b_\kappa \in B$ up to the last κ state transitions. If $h < \kappa$, then $b_k = 0$ for every $k = h+1, \ldots, \kappa$. Moreover, a critical information $\varrho_j \in \{<, =, >\}$ is recorded from the "prehistory" when $h > \kappa$, which is specific to each $c_j \in C'$ for $j = 1, \ldots, \gamma$. In addition, let $q_0' = (q_0, z_0, 0, \ldots, 0, =, \ldots, =) \in Q'$ be an initial state of A', while $F' = F \times B^\kappa \times \{<, =, >\}^\gamma \subseteq Q'$ represents the set of final states.

We define the transition function $\delta' : Q' \times \Sigma \longrightarrow Q'$ of A' by using the local state transition and shift functions of A as follows:

$$\begin{aligned}\delta'((q, b_0, \ldots, b_\kappa, \varrho_1, \ldots, \varrho_\gamma), x) \\ = \begin{cases} (\delta_r(q,x), \Delta_r(q,x), b_0, \ldots, b_{\kappa-1}, \varrho_1', \ldots, \varrho_\gamma') & \text{if } 0 < z_\kappa < 1 \\ (\delta_r(q,x), \sigma_L(z_\kappa), 0, \ldots, 0, =, \ldots, =) & \text{otherwise,} \end{cases}\end{aligned} \tag{6}$$

for $q \in Q$, $b_0, \ldots, b_\kappa \in B$, $\varrho_1, \ldots, \varrho_\gamma \in \{<, =, >\}$, and $x \in \Sigma$. In the following we will describe two cases of choosing the parameter r in definition (6) which depend on whether or not the arguments $b_0, \ldots, b_\kappa$ coincide with the first $\kappa+1$ coefficients of a series from C'. We first consider the case when for any series $\sum_{k=0}^{\infty} b_k' a^k \in C'$ with all $b_k' \in B$,

$$b_k \neq b_k' \quad \text{for some } 0 \leq k \leq \kappa\,. \tag{7}$$

In this case, parameter r is chosen so that $z_\kappa \in I_r$. Obviously, the actual register value (5) is approximated with z_κ in (6), which gives a correct simulation of A by A' according to (2), as long as $h \leq \kappa$ implying $z = z_\kappa$. Note that the register saturates properly at value $\sigma_L(z_\kappa) \in \{0, 1\}$ if $z_\kappa \leq 0$ or $z_\kappa \geq 1$. Nevertheless, the correctness of the simulation must still be proven for $h > \kappa$, and henceforth assume $h > \kappa$. Condition (7) implies $\{z_\kappa, z\} \cap C = \emptyset$, and $\{z_\kappa, z\} \subseteq I(b_0, \ldots, b_\kappa)$. It follows from the definition of κ that there is only one $r \in \{1, \ldots, p\}$ such that $I(b_0, \ldots, b_\kappa) \subset I_r$ while $I(b_0, \ldots, b_\kappa) \cap I_{r'} = \emptyset$ for the remaining $r' \neq r$, which gives $z_\kappa \in I_r$ iff $z \in I_r$ in this case.

Now consider the case when the arguments $b_0, \ldots, b_\kappa \in B$ do not satisfy condition (7), which means there exists a quasi-periodic series $\sum_{k=0}^{\infty} b_k' a^k = c_j \in C'$ with all $b_k' \in B$, maximum period $M \geq 1$, and period sum P such that

$$b_k = b_k' \quad \text{for every } k = 0, \ldots, \kappa\,. \tag{8}$$

Let $0 \leq k_1 < k_2 < k_3 < \cdots$ be the increasing infinite sequence of its term indices, which delimit the quasi-periods $m_i = k_{i+1} - k_i \leq M$ with $P_i = P$ for $i \geq 1$, so that the shifts $b_0, \ldots, b_h \in B$ defining the register value (5) coincide with the coefficients of the series $\sum_{k=0}^{\infty} b_k' a^k = c_j \in C'$ up to the first d quasi-repeating blocks for the maximum possible $d \geq 1$ over the permutations of these blocks, that is,

$$b_k = b_k' \quad \text{for every } k = 0, \ldots, k_{d+1} - 1\,, \tag{9}$$

where $\kappa \leq k_{d+1} - 1 \leq h$ according to (8). Recall that $c_j \in C'$ may serve as an endpoint of possibly three neighbor intervals I_r including a degenerate one. According to (2), parameter r in (6) can thus be chosen uniquely based on whether $z \,\varrho\, c_j$ for $\varrho \in \{<,=,>\}$. In particular, $z \,\varrho\, c_j$ rewrites to $z = (\sum_{k=0}^{k_1-1} b_k a^k + \sum_{i=1}^{d-1} a^{k_i} \sum_{k=0}^{m_i-1} b_{k_i+k} a^k + a^{k_d} \sum_{k=k_d}^{h} b_k a^{k-k_d}) \,\varrho\, c_j$ which reduces to $(a^{k_d} \sum_{k=k_d}^{h} b_k a^{k-k_d}) \,\varrho\, (a^{k_d} P)$ according to (9) and (4). Furthermore, we divide this inequality by $a^{k_d-k_1} \neq 0$ and add $\sum_{k=0}^{k_1-1} b_k a^k$ to both its sides, which yields

$$z' = \left(\sum_{k=0}^{k_1-1} b_k a^k + a^{k_1} \sum_{k=k_d}^{h} b_k a^{k-k_d} \right) \varrho' \; c_j \tag{10}$$

where $\varrho' \in \{<,=,>\}$ differs from $\varrho \in \{<,>\}$ iff

$$a < 0 \quad \& \quad k_d - k_1 = \sum_{i=1}^{d-1} m_i \text{ is odd.} \tag{11}$$

It follows that $z \,\varrho\, c_j$ can be replaced by $z' \,\varrho'\, c_j$ where z' is determined by the history of shifts $b_0, \ldots, b_{k_1-1}, b_{k_d}, \ldots, b_{k_{d+1}}, \ldots, b_h$ according to (5) in which the terms $b_{k_1}, \ldots, b_{k_d-1}$ corresponding to the first $d-1$ quasi-repeating blocks of $\sum_{k=0}^{\infty} b'_k a^k = c_j$, are excluded.

By the definition of κ', we know that $\kappa \geq \kappa' \geq k_1 + 2M - 1 \geq k_1 + m_d + m_{d+1} - 1 \geq k_1 + k_{d+2} - k_d - 1$ which gives $k_{d+2} - 1 \leq k_d + \kappa - k_1$. In addition, suppose that the history for z' in (10) exceeds $\kappa+1$ shifts (c.f. assumption $h > \kappa$ for z), that is, $k_1 + h - k_d > \kappa$ implying $h > k_d + \kappa - k_1$. This yields $z \neq c_j$ since otherwise c_j could be expressed as a finite sum (5) with $h = k_{d+1} - 1$ producing a contradiction $\kappa' + 1 \leq \kappa + 1 < k_1 + k_{d+1} - k_d = k_1 + m_d \leq k_1 + M$. Hence, there is an index $k_{d+1} \leq k \leq k_{d+2} - 1$ such that $b_k \neq b'_k$ due to the maximality of d. By the definition of κ, condition $z \,\varrho\, c_j$ can further be reduced to

$$z'_\kappa = \left(\sum_{k=0}^{k_1-1} b_k a^k + a^{k_1} \sum_{k=k_d}^{k_d+\kappa-k_1} b_k a^{k-k_d} \right) \varrho' \; c_j \tag{12}$$

which only includes the history of $\kappa + 1$ shifts from (10).

Based on the preceding analysis, we can now specify $\varrho'_1, \ldots, \varrho'_\gamma \in \{<,=,>\}$ in definition (6) of δ' which make the correct choice of parameter r possible in the case of (8). According to (6), $\varrho_1, \ldots, \varrho_\gamma$ are set to default $=$ whenever the register value z saturates at 0 or 1, including the initial state q'_0. The value of ϱ_j for $1 \leq j \leq \gamma$ is then updated only if the arguments $b_0, \ldots, b_\kappa \in B$ of δ' start with any quasi-repeating block $b'_{k_i}, \ldots, b'_{k_{i+1}-1}$ of a quasi-periodic series $\sum_{k=0}^{\infty} b'_k a^k = c_j \in C'$, which means $b_k = b'_{k_i+k}$ for every $k = 0, \ldots, m_i - 1$ where $m_i = k_{i+1} - k_i$. Otherwise set $\varrho'_j = \varrho_j$. Moreover, the update of ϱ_j depends on whether or not the block is followed by another quasi-repeating block of the series. If it is not the case, the value of ϱ'_j is chosen to satisfy the inequality $(\sum_{k=0}^{k_1-1} b'_k a^k + a^{k_1} \sum_{k=0}^{\kappa} b_k a^k) \,\varrho'_j\, c_j$ anticipating (12) with $k_d = k_i$. If, on the other hand,

$b_0, \ldots, b_\kappa$ start with at least two quasi-repeating blocks of the series (i.e. $i < d$), then ϱ'_j differs from $\varrho_j \in \{<, >\}$ iff $a > 0$ and m_i is odd, complying with (11). It follows from (12) and (11) that inequality $z\, \varrho'_j\, c_j$ holds when the arguments $b_0, \ldots, b_\kappa$ meet (8), which automaton A' exploits for deciding whether $z \in I_r$, particularly at the endpoint $c_j \in C'$ of interval I_r. This determines parameter r in definition (6) for the case of (8) and completes the proof of the theorem. $\square$

5 Directions for Ongoing Research

In the effort to fill the gap in the analysis of computational power of neural nets between integer a rational weights we have investigated a hybrid model of a binary-state network with an extra analog unit. We have shown this model to be computationally equivalent to a finite automaton with a register. Our main result in Theorem 3 formulates a sufficient condition for a language accepted by this automaton to be regular. Our preliminary study leads to natural open problems for further research such as completing the statement in Theorem 3 for $|a| > 1$, finding a corresponding necessary condition for accepting regular languages, analyzing the algebraic properties of quasi-periodic power series, characterizing the full power of finite automata with register, e.g. by comparing them to finite automata with multiplication [10, 12] etc.

Even more important, our analysis of computational power of neural nets has revealed interesting connections with an active research on representations of numbers in non-integer bases (see [1, 2, 5, 7, 8, 15, 19, 20, 22, 23] including references there). In particular, a power series $\sum_{k=0}^{\infty} b_k a^k$ can be interpreted as a representation of a number from $[0, 1]$ in base $\beta = 1/a$ using the digits from a finite set B, which is called a *β-expansion* when $\beta > 1$ and $B = \{0, 1, \ldots, \lceil\beta\rceil - 1\}$ (usually starting from $k = 1$). Any number from $\left[0, \frac{\lceil\beta\rceil-1}{\beta-1}\right]$ has a β-expansion which need not be unique. Obviously, for any integer bases $\beta \geq 2$ when multiplier a has the from $1/\beta$, the β-expansion of $c \in [0, 1]$ is eventually periodic iff c is a rational number, which satisfies the assumption of Theorem 3. For simplicity, we further assume a binary set of digits $B = \{0, 1\}$ corresponding to $1 < \beta < 2$, that is, $\frac{1}{2} < a < 1$, although the analysis has partially been extended to sets of integer digits that can even be greater than $\lceil\beta\rceil - 1$ [15].

It has been shown [23] that for $\beta \in (1, \varphi)$ where $\varphi = (1 + \sqrt{5})/2$ is the golden ratio, which means for $0.618033\ldots \leq a < 1$, any number from $[0, 1]$ has a continuum of distinct β-expansions including those not quasi-periodic, which breaks the assumption of Theorem 3. For $\beta \in (\varphi, q_c)$ where q_c is the (transcendental) *Komornik-Loreti constant* (i.e. the unique solution of equation $\sum_{k=1}^{\infty} t_k q_c^{-k} = 1$ where $(t_k)_{k=1}^{\infty}$ is the *Thue-Morse sequence* in which $t_k \in \{0, 1\}$ is the parity of the number of 1's in the binary representation of k), that is, for $0.559524\ldots < a < 0.618033\ldots$, there are countably many numbers in $[0, 1]$ having eventually periodic unique β-expansions, which are candidate elements to C in Theorem 3, while for $\beta \in (q_c, 2)$ corresponding to $\frac{1}{2} < a \leq 0.559524\ldots$, the set of numbers from $[0, 1]$ having unique β-expansions has the cardinality of continuum and a positive Hausdorff dimension (although its Lebesgue measure remains zero) [7]. In addition, for $0 < a < \frac{1}{2}$ (i.e. $\beta > 2$ whereas $B = \{0, 1\}$), not every number from $[0, 1]$ has a β-expansion (in fact, the β-expansions create a Cantor-like set in this case), which can fulfill the assumption of Theorem 3 if the elements of C do not have β-expansions.

Furthermore, for every $m \geq 2$, there exists $\beta_m \in [\varphi, 2)$ corresponding to $\frac{1}{2} < a_m < 0.618033\ldots$ such that there exists a number from $[0, 1]$ that has a periodic unique β-expansion of period m if $a < a_m$, while there is no such a number for $a \geq a_m$ [2]. In addition, a so-called *greedy* (resp. *lazy*) β-expansion has been considered which is lexicographically maximal (resp. minimal) for a given number. Denote by $\mathrm{Per}(\beta)$ a set of numbers having a quasi-periodic greedy β-expansions. If $\mathbb{I} \subseteq \mathrm{Per}(\beta)$, then β is either a Pisot or a Salem number [22] where a *Pisot* (resp. *Salem*) number is a real algebraic integer (a root of some monic polynomial with integer coefficients) greater than 1 such that all its Galois conjugates (other roots of such a unique monic polynomial with minimal degree) are in absolute value less than 1 (resp. less or equal to 1 and at least one equals 1). For any Pisot number β, it holds $\mathbb{I} \subseteq \mathrm{Per}(\beta)$, while for Salem numbers this implication is still open [8, 22]. It follows that for any non-integer rational β (which is not a Pisot nor Salem number by the integral root theorem) corresponding to irreducible fraction $a = a_1/a_2$ where $a_1 \geq 2$ and a_2 are integers, there always exists a number from $\mathbb{I}$ whose (greedy) β-expansion is not quasi-periodic.

It appears that the computational power of neural nets with extra analog unit is strongly related to the results on β-expansions which still need to be elaborated and generalized, e.g. to arbitrary sets of digits B. This opens a wide field of interesting research problems which undoubtedly deserves a deeper study.

References

1. Adamczewski, B., Frougny, C., Siegel, A., Steiner, W.: Rational numbers with purely periodic β-expansion. Bulletin of The London Mathematical Society 42(3), 538–552 (2010)
2. Allouche, J.P., Clarke, M., Sidorov, N.: Periodic unique beta-expansions: The Sharkovskiĭ ordering. Ergodic Theory and Dynamical Systems 29(4), 1055–1074 (2009)
3. Alon, N., Dewdney, A.K., Ott, T.J.: Efficient simulation of finite automata by neural nets. Journal of the ACM 38(2), 495–514 (1991)
4. Balcázar, J.L., Gavaldà, R., Siegelmann, H.T.: Computational power of neural networks: A characterization in terms of Kolmogorov complexity. IEEE Transactions on Information Theory 43(4), 1175–1183 (1997)
5. Chunarom, D., Laohakosol, V.: Expansions of real numbers in non-integer bases. Journal of the Korean Mathematical Society 47(4), 861–877 (2010)
6. Dassow, J., Mitrana, V.: Finite automata over free groups. International Journal of Algebra and Computation 10(6), 725–738 (2000)
7. Glendinning, P., Sidorov, N.: Unique representations of real numbers in non-integer bases. Mathematical Research Letters 8(4), 535–543 (2001)
8. Hare, K.G.: Beta-expansions of Pisot and Salem numbers. In: Proceedings of the Waterloo Workshop in Computer Algebra 2006: Latest Advances in Symbolic Algorithms. pp. 67–84. World Scientific (2007)

9. Horne, B.G., Hush, D.R.: Bounds on the complexity of recurrent neural network implementations of finite state machines. Neural Networks 9(2), 243–252 (1996)
10. Ibarra, O.H., Sahni, S., Kim, C.E.: Finite automata with multiplication. Theoretical Computer Science 2(3), 271–294 (1976)
11. Indyk, P.: Optimal simulation of automata by neural nets. In: Proceedings of the STACS 1995 Twelfth Annual Symposium on Theoretical Aspects of Computer Science. LNCS, vol. 900, pp. 337–348 (1995)
12. Kambites, M.E.: Formal languages and groups as memory. Communications in Algebra 37(1), 193–208 (2009)
13. Kilian, J., Siegelmann, H.T.: The dynamic universality of sigmoidal neural networks. Information and Computation 128(1), 48–56 (1996)
14. Koiran, P.: A family of universal recurrent networks. Theoretical Computer Science 168(2), 473–480 (1996)
15. Komornik, V., Loreti, P.: Subexpansions, superexpansions and uniqueness properties in non-integer bases. Periodica Mathematica Hungarica 44(2), 197–218 (2002)
16. Minsky, M.: Computations: Finite and Infinite Machines. Prentice-Hall, Englewood Cliffs (1967)
17. Mitrana, V., Stiebe, R.: Extended finite automata over groups. Discrete Applied Mathematics 108(3), 287–300 (2001)
18. Orponen, P.: Computing with truly asynchronous threshold logic networks. Theoretical Computer Science 174(1-2), 123–136 (1997)
19. Parry, W.: On the β-expansions of real numbers. Acta Mathematica Hungarica 11(3), 401–416 (1960)
20. Rényi, A.: Representations for real numbers and their ergodic properties. Acta Mathematica Academiae Scientiarum Hungaricae 8(3-4), 477–493 (1957)
21. Salehi, Ö., Yakaryilmaz, A., Say, A.C.C.: Real-time vector automata. In: Proceedings of the FCT 2013 Nineteenth International Symposium on Fundamentals of Computation Theory. LNCS, vol. 8070, pp. 293–304 (2013)
22. Schmidt, K.: On periodic expansions of Pisot numbers and Salem numbers. Bulletin of the London Mathematical Society 12(4), 269–278 (1980)
23. Sidorov, N.: Expansions in non-integer bases: Lower, middle and top orders. Journal of Number Theory 129(4), 741–754 (2009)
24. Siegelmann, H.T.: Recurrent neural networks and finite automata. Journal of Computational Intelligence 12(4), 567–574 (1996)
25. Siegelmann, H.T.: Neural Networks and Analog Computation: Beyond the Turing Limit. Birkhäuser, Boston (1999)
26. Siegelmann, H.T., Sontag, E.D.: Analog computation via neural networks. Theoretical Computer Science 131(2), 331–360 (1994)
27. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. Journal of Computer System Science 50(1), 132–150 (1995)
28. Šíma, J.: Analog stable simulation of discrete neural networks. Neural Network World 7(6), 679–686 (1997)
29. Šíma, J.: Energy complexity of recurrent neural networks. Neural Computation 26(5), 953–973 (2014)
30. Šíma, J., Orponen, P.: General-purpose computation with neural networks: A survey of complexity theoretic results. Neural Computation 15(12), 2727–2778 (2003)
31. Šíma, J., Wiedermann, J.: Theory of neuromata. Journal of the ACM 45(1), 155–178 (1998)
32. Šorel, M., Šíma, J.: Robust RBF finite automata. Neurocomputing 62, 93–110 (2004)