ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო

უნივერსიტეტი

ჯაბა გედენიძე

ნეირონული ქსელების გამოყენება კრიპტოგრაფიულად მედეგი ფსევდოშემთხვევითი გენერატორის ასაგებად

სამაგისტრო პროგრამა: ინფორმაციული სისტემები

ხელმძღვანელი: **ზურაბ ქოჩლაძე**

ასისტენტ პროფესორი

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი

ანოტაცია

ნაშრომი წარმოადგენს კრიპტოგრაფიაში არსებული პრობლემის გადაწყვეტის ერთ-ერთ მცდელობას. კერძოდ, ნაკადური შიფრებისთვის კრიპტოგრაფიულად საიმედო გამის დაგენერირება, დაკავშირებულია გარკვეულ პრობლემებთან, რომელიც განხილულია მოცემულ ნაშრომში. ჩვენი ნაშრომი შეეხება ნეირონული ქსელის საშუალებით ისეთი გენერატორის შექმნის მცდელობას რომელიც დაფუძნებულია ცალმხრივ ფუნქციაზე. ცალმხრივ ფუნქციაზე დაფუძნებულ გენერატორებს წარმოადგენენ RSA და BBC გენერატორები, რომლებიც წარმოქმნიან საკმაოდ საიმედო გამას, მაგრამ მათი ნაკლი არის ის, რომ ეს გენერატორები არის ძალიან ნელი. სწორედ ამ პრობლემის გადასაჭრელად გადავწყვიტეთ ნეირონული ქსელის საშუალებით ავაგოთ გენერატორი რომელიც დაფუძნებული იქნება RSA გენერატორზე, რომელიც დააგენერირებს საიმედო გამას და ამავე დროს იქნება RSA გენერატორზე სწრაფი.

Annotation

In the work, " Construct the pseudo-random number generator using neural networks", represents one attempt to solve the problems in cryptography. Our work is by means of neural network an attempt to create a generator based on a one-way function. A one-way function based on RSA and BBC generators, these generators that produce extremely reliable gamma, but the drawback is that these generators is very slow. To solve this problem, we decided to build a neural network based on the RSA generator to the generator, which will generate reliable gamma and this generator is a fast than RSA generator.

სარჩევი

შესავალი

ჩვენი ნაშრომი არის კრიპტოგრაფიაში აქტუალური პრობლემის გადაწყვეტის ერთ-ერთი მცდელობა. კრიპტოგრაფიაში ხშირი გამოყენება აქვს ნაკადურ შიფრებს, რადგან ნაკადური შიფრების გამოყენება უფრო მარტივიცაა და უფრო სწრაფიც, ასევე მათ არ გააჩნიათ შეცდომების გამრავლების თვისება ანუ ერთი ბიტის ცვლილება სინქრონულ ნაკადურ შიფრებში გამოიწვევს მხოლოდ ერთ შეცდომას, ხოლო ასინქრონულ ნაკადურ შიფრებში შეცდომა გავრცელდება მხოლოდ n ბიტზე.

შიფროტექსტის კრიპტომედეგობა დამოკიდებულია გასაღებების გამის ფსევდოშემთხვევითობაზე. ფსევდოშემთხვევითი მიმდევრობის კრიპტომედეგობის გამოსაკვლევად საჭიროა სტატისტიკური ტესტების გავლა. მიუხედავად იმისა, რომ წრფივი უკუკავშირის გენერატორებს აქვთ კარგი სტატისტიკური მაჩვენებლები ისინი მაინც არ წარმოადგენენ საიმედო გენერატორებს.

წრფივი უკუკავშირის რეგისტრებიდან (LFSR)- დან გამომავალი მიმდევრობა არ იძლევა საიმედო გამას და არ შეიძლება მათი პირდაპირი გამოყენება ნაკადურ შიფრებში. ბერლეკამპ - მესის ალგორითმი გამოიყენება s ბინარული მიმდევრობის წრფივი სირთულის გამოსათვლელად. ამ ალგორითმის მნიშვნელოვან თვისებას წარმოადგენს, რომ ნებისმიერი $2L$ სიგრძის მიმდევრობა, რომელიც მიიღება LFSR გამოსასვლელზე, საკმარისია იმისთვის, რომ ალგორითმა მთლიანად განსაზღვროს მიმდევრობა, რომელსაც გააჩნია L წრფივი სირთულე, რაც ნიშნავს იმას, რომ ამ ალგორითმისთვის საკმარისია $2L$ სიგრძის ნებისმიერი მიმდევრობა ბიტებისა, რათა დაადგინოს ის მინიმალური LFSR, რომელიც გამოიმუშავებს ამ მიმდევრობას. ბერლეკამპ - მესის ალგორითმის ეს თვისება პირდაპრი მიუთითებს, რომ LFSR-ის გამოყენება ნაკადური შიფრების შესაქმნელად არ შეიძლება, მიუხედავად იმისა, რომ მათ მიერ გამომუშავებულ მიმდევრობებს გააჩნიათ მაქსიმალური პერიოდი და კარგი სტატისტიკური თვისებები, რადგან ღია ტექსტით შეტევის შემთხვევაში გამის გამოთვლა ძალიან მარტივია (დაშიფრული და ღია ტექსტის XOR-ით შეკრება გვაძლევს გამას), სულ მცირე რაოდენობის ღია ტექსტია საკმარისი იმისათვის, რომ გასაღები „გატყდეს“.

LFSR-ზე დაფუძნებული გენერატორების აგების დროს ჯერ ხდება გენერატორის კონსტრუქციის აგება და შემდეგ ხდება უკვე გენერატორის კრიპტოანალიზი. შესაძლებელია უფრო საიმედო გენერატორის აგება თუ გამოვიყენებთ სირთულის თეორიას. ასეთ

შემთხვევაში ჯერ შეირჩევა სირთულის ამოცანა და შემდეგ ხდება გენერატორის აგება. ამ შემთხვევაში გენერატორის გასატეხად საჭირო იქნება შესაბამისი რთული ამოცანის ამოხსნა.

ამ მხრივ საიმედოა ცალმხრივ ფუნქციებზე აგებული გენერატორები, ისეთი, როგორიცაა RSA გენერატორი, რომელიც წარმოადგენს RSA ალგორითმის მოდიფიცირებას. ესეთი გენერატორები არის კრიპტომედეგი, მაგრამ არიან ძალიან ნელი. სწორედ ამ პრობლემის გადასაჭრელად ჩვენ გადავწყვიტეთ აგვეგო RSA ტიპის გენერატორი ნეირონული ქსელის საშუალებით. ჩვენ აგებული გვაქვს RSA გენერატორი რომელიც რეალიზებულია Matlab-ში და მრავალშრიანი ნეირონული ქსელი (MLP) რომელსაც ვასწავლეთ RSA გენერატორის გამომუშავებული მიმდევრობის მსგავსი მიმდევრობის გამომუშავება. ნეირონული ქსელის სწავლება განხორციელდა წარმატებით და ქსელმა გამოსასვლელზე მოგვცა საკმაოდ კარგი მიმდევრობა, რომელიც ახლოსაა RSA გენერატორის მიერ გამომუშავებულ მიმდევრობასთან.

თავი 1. ნაკადური შიფრები

1.1 ნაკადური შიფრების განმარტება. ნაკადური შიფრები წარმოადგენენ სიმეტრიული შიფრების ერთ-ერთ მნიშვნელოვან ნაწილს. ნაკადური შიფრებში გასაღებისა და სპეციალური ალგორითმის საშუალებით ხდება ფსევდოშემთხვევითი მიმდევრობების გამომუშავება, რომელსაც უწოდებენ გასაღებების ნაკადს (**გამას**). გასაღებების ნაკადი იკრიბება ღია ტექსტთან და მიიღება შიფროტექსტი.

ნაკადური შიფრები აპარატურული შესრულებით ზოგადად უფრო სწრაფია ვიდრე ბლოკური შიფრი. ამიტომ ეს ალგორითმები ხშირად გამოიყენება სატელეკომუნიკაციო არხებში ინფორმაციის ონლაინ რეჟიმში გადასაცემად, როდესაც გადამცემი სისტემის მეხსიერება შეზღუდულია და როდესაც გადასაცემი ტექსტის ასო-ნიშნები უნდა დამუშავდეს იმ მიმდევრობით რა მიმდევრობითაც ისინი შემოდიან დამშიფრავ ალგორითმში. რადგან მათ არა აქვთ შეცდომების გამრავლების თვისება ან შეცდომებს ამრავლებენ შეზღუდულად, მათი გამოყენება მოსახერხებელია იმ სიტუაციებში, როდესაც მაღალია შეცდომის დაშვების ალბათობა.

ნაკადური შიფრები შეგვიძლია აღვწეროთ (M, C, K, L, F, E, D) ობიექტების ერთობლიობით, რომლებისთვისაც დაკმაყოფილებულია შემდეგი პირობები:

1. M არის შესაძლო ღია ტექსტის სასრული სიმრავლე;
2. C არის შესაძლო შიფროტექსტების სასრული სიმრავლე;
3. K არის შესაძლო გასაღებების სასრული სიმრავლე;
4. L არის გასაღებების ნაკადის (გამის) ალფაბეტი;
5. $F = (f_1, f_2, ...)$ არის გასაღების ნაკადის გენერატორი, რომელიც ყოველი $i \geq 1$-თვის გამოიმუშავებს ნაკადის (გამის) შემდეგ ელემენტებს $f_i : K \times M^{i-1} \to L$
6. ყველა $z \in L$- თვის არსებობს დაშიფვრის წესი $e_z \in E$ და შესაბამისი დეშიფრაციის წესი $d_z \in D$ ისეთი, რომ $e_z : M \to C$ და $d_z : C \to M$ არიან ფუნქციები, რომლებიც აკმაყოფილებენ პირობას $d_z(e_z(m)) = m$ ნებისმიერი $m \in M$ -თვის.

<u>განსაზღვრება</u> გასაღებების ნაკადის (გამის) თვისებას $z_{i+d} = z_i$ ყველა $i \geq 1$-თვის, სადაც i ნებისმიერი დადებითი რიცხვია ეწოდება გასაღების პერიოდულობა, d -ს კი გასაღებების პერიოდი.

რაც უფრო დიდი იქნება გასაღების პერიოდი, მით უფრო გამძლე იქნება შიფრი კრიპტო შემოტევების მიმართ. ამას ადასტურებს ვერნამის შიფრიც, რომლის გასაღების პერიოდი უსასრულოა. რადგანაც ვერნამის შიფრის გამოყენება დაკავშირეულია გარკვეულ სიძნელეებთან, ხოლო მარტო საწყისი გასაღების გამოყენება ფსევდოშემთხვევით მიმდევრობად არაა საკმარისი ღია ტექსტის სტრუქტურის დასამალად, თანამედროვე ნაკადური შიფრების შექმნის დროს ფართოდ გამოიყენება ფსევდოშემთხვევითი მიმდევრობების გენერატორები, რომლებიც საწყისი, შედარებით პატარა გასაღებებიდან ქმნიან გარკვეული პერიოდის მქონე გასაღებების მიმდევრობას (გამას).

განვიხილოთ ნაკადური შიფრების ტიპები:

1.2 სინქრონული ნაკადური შიფრები

შიფრს ეწოდება სინქრონული ნაკადური შიფრი თუ მისი გასაღების მიმდევრობა გამომუშავდება ღია ან შიფროტექსტისგან დამოუკიდებლად. ასეთ შიფრებში დაშიფვრის პროცესი შეიძლება აღიწეროს შემდეგნაირად:

$$\sigma_{i+1} = f(\sigma_i, k)$$

$$z_i = g(\sigma_i, k)$$
$$c_i = h(z_i, m_i)$$

სადაც f და g ფუნქციებით და k საწყისი გასაღებით ხდება დამშიფრავი გამის გამომუშავება და h ფუნქციით კი ღია ტექსტის დაშიფვრა.

1.3 თვითსინქრონიზებადი ანუ ასინქრონული ნაკადური შიფრები

ასინქრონული ნაკადური შიფრი ეწოდება ისეთ ნაკადურ შიფრს, რომელშიც გამის გენერირებაში მონაწილეობენ როგორც გასაღები ასევე შიფროტექსტის გარკვეული, მუდმივი, n რაოდენობის ბიტები. მათემატიკურად ეს პროცესი შეიძლება აღვწეროთ შემდეგი სახით:

$$\sigma_i = (c_{i-n}, c_{i-n+1}, ..., c_{i-1})$$

$$z_i = g(\sigma_i, k)$$

$$c_i = h(z_i, m_i)$$

სადაც $\sigma_0 = (c_{-n}, c_{-n+1}, ..., c_{-1})$ არის ინიციალიზაციის ვექტორი, k არის გასაღები, g არის ფუნქცია, რომლის საშუალებითაც გამომუშავდება დამშიფრავი ნაკადი (გამა) და h არის გამოსასვლელის ფუნქცია, რომელიც ღია ტექსტს შიფრავს მიღებული გამის საშუალებით.

1.4 ბინარულ ადიტიური ნაკადური შიფრი.

ბინარულ ადიტიური ნაკადური შიფრი წარმოადგენს ნაკადური შიფრების კერძო კლასს, რომელშიც როგორც ღია ტექსტის, ასევე გასაღებების და შიფროგრამების ანბანი შედგება ორი ელემენტისაგან (0, 1) და დამშიფრავი h ფუნქცია წარმოადგენს **XOR** ოპერაციას.

თავი 2. ფსევდოშემთხვევითი მიმდევრობები

2.1 ფსევდოშემთხვევითი ბიტების გენერატორი

რადგანაც ნაკადურ შიფრებში გამოიყენება გარდაქმნის ძალიან მარტივი ოპერაციები (ძირითადად XOR-ით შეკრება), ამიტომ მედეგობა თითქმის მთლიანად დამოკიდებული იქნება იმ ფსევდოშემთხვევით მიმდევრობაზე, რომელიც გამომუშავდება გენერატორის მიერ. ფსევდოშემთხვევითი ბიტების გენერაორი (PRBG) წარმოადგენს დეტერმინირებულ ალგორითმს, რომელიც რაიმე მოცემული k სიგრძის ნამდვილად შემთხვევით მიმდევრობიდან გამოიმუშავებს ბინარულ მიმდევრობას, რომლის სიგრძე l გაცილებით მეტია k-ზე და „წააგავს“ შემთხვევით მიმდევრობას ანუ ფსევდოშემთხვევით მიმდევრობას ვიღებთ. PRBG-ს შესასვლელ მიმდევრობას ეწოდება საწყისი მიმდევრობა, ანუ გასაღები, ხოლო გამოსასვლელზე მიღებულ მიმდევრობას კი ფსევდოშემთხვევითი ბიტური მიმდევრობა.

PRBG გამოსასვლელი მიმდევრობა არ არის შემთხვევითი მიმდევრობა. ფაქტობრივად შესაძლო სხვადასხვა მიმდევრობების რაოდენობა, რომლებიც შეიძლება გამოიმუშაოს გენერატორმა k სიგრძის საწყისი მიმდევრობიდან წამროადგენს მცირე, კერძოდ $2^k/2^l$ ნაწილს ყველა l სიგრძის მიმდევრობებისა. ასეთი გენერატორების იდეა სწორედ იმაში მდგომარეობს, რომ პატარა შემთხვევითი მიდევრობა გავჭიმოთ ისე, რომ მოწინააღმდეგემ ვერ შეძლოს გაარჩიოს მიღებული მიმდებრობა შემთხვევითია თუ არა. იმისათვის რომ დავრწმუნდეთ, რომ მოწინააღმდეგე ვერ შეძლებს ამის გაკეთებას, საჭიროა PRBG წინასწარ გაიაროს შესაბამისი ტესტები.

1. PRBG რომ დააკმაყოფილოს კრიპტოგრაფიული საიმედოობის მოთხოვნები, საჭიროა მისი x_0 საწყისი მნიშვნელობის სიგრძე k იყოს საკმარისად დიდი, რათა მოწინააღმდეგემ ვერ შეძლოს ძალისმიერი შეტევით გამოთვალოს ის;
2. აუცილებელია, რომ გამოსასვლელზე მიღებული მიმდევრობა სტატისტიკურად არ განსხვავდებოდეს შემთხვევითი მიმდევრობისგან;

3. შეზღუდული გამოთვლითი საშუალებების მქონე მოწინააღმდეგემ ვერ უნდა შეძლოს მიმდევრობის უკვე გამომუშავებული ბიტების საშუალებით გამოთვალოს შემდეგი გამოსასვლელი ბიტი.

ამ მოთხოვნებიდან გამომდინარე გენერატორები გადიან სტატისტიკურ ტესტებს. სტატისტიკური თვისებები, რომლებსაც უნდა აკმაყოფილებდეს PRBG, სამ ძირითად პოსტულატად ჩამოაყალიბა ს.გოლომბომ და ისინი ცნობილია, როგორც გოლომბოს პოსტულატები. ესენია:

1. „1“- ის და „0“- ის რაოდენობა მიმდევრობის ყოველ პერიოდში ერთმანეთისგან შეიძლება განსხვავდებოდეს მხოლოდ ერთით;
2. ყოველ პერიოდში ერთნაირი სიმბოლოებისაგან შედგენილი სერიების რაოდენობის ნახევარს უნდა ქონდეს ერთის ტოლი სიგრძე, სერიების ერთ მეოთხედს - ორი, ერთ მერვედს - სამი და ა.შ. უფრო მეტიც, ყოველი ამ სიგრძეებისთვის უნდა იყოს ერთნაირი რაოდენობის სერიები, შედგენილი ერთებისა და ნოლებისგან;
3. დავუშვათ გვაქვს ერთი და იგივე მიმდევრობის ორი ასლი პერიოდით p, რომლებიც წაძრულები არიან ერთმანეთის მიმართ რაიმე d მანძილით. მაშინ თითოეული d - თვის $(0 \le d \le p-1)$ შეგვიძლია დავთვალოთ შეთანხმებულობის (A_d) და შეუთანხმებლობის (D_d) რაოდენობები ამ მიმდევრობებისათვის. ავტოკორელაციის კოეფიციენტი თითოეული d - სთვის გამოითვლება ფორმულით:

$$(A_d - D_d)/p$$

ზოგადად ავტოკორელაციის ფუნქცია ღებულობს სხვადასხვა მნიშვნელობებს, როდესაც d გარბის ყველა დასაშვებ მნიშვნელობას. ნებისმიერი ფუნქციისათვის, რომელიც აკმაყოფილებს პირველ და მეორე პირობას, ავტოკორელაციის ფუნქცია უნდა ღებულობდეს მხოლოდ ორ მნიშვნელობას.

ცხადია, მარტო ეს პოსტულატები არაა საკმარისი ფსევდოშემთხვევითი მიმდევრობის კრიპტომედეგობის გამოსაკვლევად. საჭიროა კიდევ მრავალი სტატისტიკური ტესტების გავლა. ჩამოვთალოთ რამოდენიმე სტატისტიკური ტესტი, რომელთაც შემდეგ ქვმოთ განვიხილავთ. ესენია: სიხშირული ტესტი, მიმდევრობითი ტესტი, სერიების ტესტი, ავტოკორელაციური ტესტი, გამეორებათა ტესტის და ა.შ.

2.2 ძვრის წრფივი უკუკავშირიანი რეგისტრი (Linear Feedback Shift Register (LFSR))

ნაკადურ შიფრებში ყველაზე ხშირად გამოიყენებოდა ძვრის უკუკავშირიანი რეგისტრები და კერძოდ, ძვრის წრფივი უკუკავშირიანი რეგისტრი **(LFSR).** ამას განაპირობებს ის, რომ ასეთი რეგისტრების ტექნიკური შესრულება ადვილია, მათ შეუძლიათ საკმარისად დიდი პერიოდის მქონე მიმდევრობების გენერირება, შეუძლიათ გამოიმუშაონ მიმდევრობები საუკეთესო სტატისტიკური მახასიათებლებით და მათი სტრუქტურიდან გამომდინარე ადვილია მათი ანალიზი ალგებრული სტრუქტურების გამოყენებით.

t	regiსtri 2	registri 1	registri 0
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	1	0	1
6	0	1	0
7	0	0	1

ცხრილი 2.1

L სიგრძის ძვრის წრფივი უკუკავშირის რეგისტრი შეიცავს L რეგისტრს გადანომრილს $0,1,2,\ldots,L-1$ თითოეულ მათგანს შეუძლია ერთი ბიტის მიღება, შენახვა და გადაცემა. რეგისტრში გვაქვს მთვლელი, რომელიც აკონტროლებს ბიტების მოძრაობას.

სურ. 2.2.1 L სიგრძის ძვრის წრფივი უკუკავშირის რეგისტრი s_i

დროის ფიქსირებულ მომენტში შესაძლებელია შემდეგი ოპერაციების ჩატარება:

1. ნულოვან რეგისტრში მოთავსებული ბიტი გამოდის და ხდება გამოსასვლელი მიმდევრობის ერთი ბიტი;
2. i -ური რეგისტრის ბიტი გადაეცემა $i-1$ რეგისტრს ყველა i - თვის $1 \le i \le L-1$;
3. ახალ შესასვლელს $L-1$ რეგისტრისათვის წარმოადგენს უკუკავშირის ბიტი s_i, რომელიც გამოითვლება ფიქსირებული რეგისტრების მნიშვნელობების XOR-ით შეკრების გზით; (სურ. 2.2.1)

სურათზე მითითებული c_i სიდიდეები ღებულობენ მნიშვნელობას {0, 1} სიმრავლიდან. ნახევარწრეები კი წარმოადგენენ ლოგიკურ ფუნქციას „და“, ამიტომ შეიკრიბება ის ბიტები, რომელთა შესაბამისი c_i - მნიშვნელობები იქნება ერთის ტოლი. თუ რეგისტრებს შევუსაბამებთ X ცვლადს ნოლიდან L ხარისხის ჩათვლით, ხოლო c_i - ები იქნება კოეფიციენტები შესაბამის ხარისხებთან, მაშინ ასეთი LFSR სტრუქტურა შეიძლება აღვწეროთ $C(X) = 1 + c_1 X + c_2 X^2 + ... + c_L X^L \in Z_2[X]$ პოლინომის საშუალებით, რომელსაც უწოდებენ LFSR- ის მახასიათებელ პოლინომს და აღნიშნავენ $\langle L, C(X) \rangle$ სიმბოლოთი. LFSR- ს უწოდებენ არასინგულარულს, თუ $C(X)$ პოლინომი არის L ხარისხის (ანუ $c_L = 1$). თუ საწყის მომენტში თითოეული i -ურ რეგისტრში ($1 \le i \le L-1$) მოთავსებულია $s_i \in \{0,1\}$ ბიტი, მაშინ $[s_{L-1}, ..., s_1, s_0]$ მიმდევრობას ეწოდება საწყისი მდგომარეობა. LFSR გამოსავალი მიმდევრობა $s = s_0, s_1, ...$ ცალსახად განისაზღვრება შემდეგი რეკურსიით:

$s_j = (c_1 s_{j-1} + c_2 s_{j-2} + ... + c_L s_{j-L}) \bmod 2,$ ყველა $j \ge L$.

თეორემა 2.1 L ხარისხის LFSR -ის მიერ წარმოქმნილი მიმდევრობის მაქსიმალური პერიოდი ტოლია $2^L - 1$.

მოვიყვანოთ რამდენიმე დებულება, რომლებიც განსაზღვრავენ LFSR მიერ გამომუშავებული მიმდევრობის თვისაებების დამოკიდებულებას მახასიათებელ პოლინომთან:

- თუ $C_L = 0$, LFSR მიერ გენერირებული მიმდევრობების პერიოდულობა შეიძლება თავიდანვე არ გამჟღავნდეს.

- თუ $C_L = 1$, მაშინ LFSR უწოდებენ არასინგულარულს. ასეთი პოლინომის მქონე LFSR- ის გამოსასვლელი მიმდევრობა თავიდანვე იქნება პერიოდული, ანუ

t	X_3	X_2	X_1	X_0	t	X_3	X_2	X_1	X_0

ნ ე გ ი ს მ ი ე რ

ი i - სთვის შესრულდება ტოლობა $s_{N+i} = s_i$.

- თუ $C(X)$ იქნება დაუყვანადი პოლინომი მაშინ ნებისმიერი არანულოვანი საწყისი მდგომარეობისათვის გენერირებული მიდევრობის პერიოდი ტოლია იმ უმცირესი N რიცხვისა, რომლისთვისაც $C(X)$ ყოფს $(1+X^N)$ პოლინომს და შესაბამისად მიმდევრობის პერიოდი გაყოფს $2^L - 1$ რიცხვს.
- თუ $C(X)$ იქნება დაუყვანადი პოლინომი, მაშინ ნებისმიერი არანულოვანი საწყისი მდგომარეობისათვის LFSR-ის მიერ გამომუშავებულ მიმდევრობას ექნება შესაძლო მაქსიმალური პერიოდი $2^L - 1$. ასეთი პერიოდის მქონე გამოსასვლელ მიმდევრობას, უწოდებენ $m-$ მიმდევრობას.

მაგალითად მეოთხე ხარისხის LFSR-ის მახასიათებელი პოლინომია პრიმიტიული პოლინომი $X^4 + X + 1$ და საწყისი მდგომარეობაა $[0,1,1,0]$. ცხრილში 2.2 მოყვანილია LFSR-ს რეგისტრების მნიშვნელობები ყოველი ტაქტის ბოლოს. გამოსასვლელ მიმდევრობას კი, რომელსაც გამოიმუშავებს LFSR ექნება შემდეგი სახე:

$s = 0,1,1,0,0,1,0,0,0,1,1,1,1,0,1,\ldots$ მაშინ ამ LFSR-ის პერიოდი იქნება 15.

0	0	1	1	0	8	1	1	1	0
1	0	0	1	1	9	1	1	1	1
2	1	0	0	1	10	0	1	1	1
3	0	1	0	0	11	1	0	1	1
4	0	0	1	0	12	0	1	0	1
5	0	0	0	1	13	1	0	1	0
6	1	0	0	0	14	1	1	0	1
7	1	1	0	0	15	0	1	1	0

ცხრილი2.2

2.3 წრფივი სირთულე

იმისთვის, რომ შევაფასოთ თუ რამდენად საიმედოა LFSR- ის მიერ გამომუშავებული ფსევდოშემთხვევითი მიმდევრობა შემოდის წრფივი სირთულის ცნება.

s მიმდევრობის წრფივი სირთულე ($L(s)$) განისაზღვრება შემდეგნაირად:

- თუ გვაქვს ნულოვანი მიმდევრობა ანუ $s = 0,0,0,\ldots$, მაშინ $L(s) = 0$;
- თუ არსებობს ისეთი LFSR, რომელიც წარმოქმნის s უსასრულო მიმდევრობას, მაშინ $L(s) = \infty$;
- დანარჩენ შემთხვევებში $L(s)$ უდრის იმ უმოკლესი LFSR სიგრძეს, რომელიც წარმოქმნის s -ს.

ანალოგურად განისაზღვრება s^n მიმდევროსათვის წრფივი სირთულე. ამ სიდიდეს გააჩნია შემდეგი თვისებები:

- s^n ქვემიმდევრობის წრფივი სირთულე აკმაყოფილებს უტოლობას $0 \le L(s^n) \le n$ ნებისმიერი $n \ge 1$.
- $L(s^n) = 0$ მაშინ და მხოლოდ მაშინ, როდესაც s^n არის n სიგრძის ნულოვანი მიმდევრობა;
- $L(s^n) = n$ მაშინ და მხოლოდ მაშინ, როდესაც $s^n = 0,0,\ldots 0,1.$;
- თუ s არის პერიოდული მიმდევრობა პერიოდით N , მაშინ $L(s) \le N$;
- $L(s \oplus t) \le L(s) \oplus L(t)$;

კრიპტოგრაფიულად საიმედო გამას უნდა გააჩნდეს ძალიან მაღალი წრფივი სირთულე. წრფივი სირთულის თვისებებიდან კი გამოდის, რომ n სიგრძის მიმდევრბისათვის მაქსიმალური მნიშვნელობა არის n და ეს მიიღწევა მაშინ, როდესაც s^n მიმდევროზას აქვს $s^n = 0,0,...0,1$ სახე. მიუხედავად ამისა $s^n = 0,0,...0,1$ მიმდევრობა არ ტოვებს შთაბეჭდილებას, რომ ის იქნება კრიპტოგმედეგი გამის წარმომქმნელი, რადგან საწყის მდგომარეობა ძალიან ახლოსაა ნულოვან მდგომარეობასთან ჰემინგის მანძილით. ამიტომ წრფივი სირთულე არ არის საკმარისი გამის შესაფასებლად.

LFSR- ზე დაფუძნებულოი გენერატორების აგება კონსტრუირების თვალსაზრისით ჰგავს ბლოკური შიფრების აგების პროცესს. ჯერ ხდება გენერატორის კონსტრუქციის აგება და შემდეგ ხდება უკვე გენერატორის კრიპტოანალიზი. შესაძლებელია უფრო საიმედო გენერატორების კრიპტოანალიზი. შესაძლებელია უფრო საიმედო გენერატორების აგება, თუ გამოვიყენებთ სირთულის თეორიას. ასეთ შემთხვევაში ჯერ შეირჩევა, რომელიმე სირთულის ამოცანა და შემდეგ მის საფუძველზე ხდება გენერატორის აგება მაშინ გენერატორის გასატეხად საჭირო იქნება შესაბამისო რთული ამოცანის ამოხსნა. ასეთ გენერატორებს წარმოადგენენ RSA და BBS გენერატორები, რომლებიც დაფუძნებულია ცალმხრივ მიმართულ ფუნქციაზე და ეს გენერატორები გამოიმუშავებენ კრიპტომედეგ გამას.

2.4 RSA გენერატორი

ეს გენერატორი წარმოადგენს RSA ალგორითმის მოდიფიკაციას. საწყისი პარამეტრებია მოდულის ფუძე $n = p \cdot q$, სადაც p და q ორი დიდი მარტივი რიცხვია და e რიცხვი, რომელიც აკმაყოფილებს პირობას:

$$(e, \phi(n)) = 1$$

სადაც $\varphi(n) = (p-1)(q-1)$

ამის შემდეგ ალგორითმს მიეწოდება ნებისმიერი ტექსტი M, რომელსაც დაშიფრავს, $C = M^e \bmod n$ ჩვენ ვიღებთ დაშიფრული ტექსტის ბოლო ბიტს და ვინახავთ ფაილში. ამ პროცესის გამეორებით ვახდენთ მიმდევრობების დაგროვებას.

შემდეგი ეტეპია ნეირონული ქსელის აგება და დასწავლა, ამისათვის განვიხილოთ ნეირონული ქსელების აგების და დასწავლის მეთოდები.

თავი 3.ნეირონული ქსელები

3.1 ნეირონული ქსელების განმარტება

ნეირონული ქსელი (Neural Network) წარმოადგენს ადამიანის ტვინის გამარტივებულ მოდელს. ხელოვნური ნეირონული ქსელები ახდენენ ტვინის მუშაობის იმიტაციას. ინფორმაცია გადაეცემა ნეირონებს შორის, ხოლო ქსელის სტრუქტურა და ნეირონების დაბოლოებების წონები განსაზღვრავენ ქსელის ქცევას.

ნეირონული ქსელების გამოყენება

ხელოვნური ნეირონული ქსელების გამოყენების სფეროებია: კლასიფიკაციის და პროგნოზირების ამოცანები, გენეტიკურ ალგორითმებთან ერთობლიობაში გამოიყენება ნეირონული ქსელები.

მათემატიკური მოდელი

ერთმანეთში გაერთიანებული ნეირონები ქმნიან ქსელს რომელიც ტვინის მუშაობის მოდელირებას ახდენს. ასეთ მოდელს შეუძლია მიიღოს ინფორმსცია გარედან და მას გააჩნია თვითსწავლის უნარი.

მათემატიკური თვალსაზრისით ნეირონის მოდელი ასრულებს X(1), X(2), .. X(N) შემავალი სიგნალის არაწრფივ გარდაქმნას Y - გამოსავალ სიგნალში.

თითოეული ნეირონი, როგორც ქსელის ელემენტი, აღიწერება:

- **შესასვლელი სიგნალები** x_i - ესაა მონაცემები, რომლებიც მიეწოდება ნეირონს შესასვლელზე გარე სამყაროდან ან სხვა ნეირონებიდან. შესასვლელი ნეირონების დიაპაზონი სხვადასხვა მოდელებისათვის შეიძლება იყოს განსხვავებული. როგორც წესი ეს სიგნალები არიან დისკრეტული და ღებულობენ მნიშვნელობებს {0,1}, {-1,1} სიმრავლეებიდან ან **R**-დან.
- **წონითი კოეფიციენტები** w_i. წონითი კოეფიციენტები აიღება ნამდვილ რიცხვთა სიმრავლიდან და განსაზღვრავენ კავშირის ძალას ნეირონებს შორის.
- ნეირონის აქტივაციის დონე. ესაა სიდიდე, რომელიც განსაზღვრება ნეირონში შემავალი $\sum w_i x_i$ სიგნალების შეწონილი ჯამით (ვექტორულად **WX**).
- ზღურბლოვანი ფუნქცია f. ამ ფუნქციის დანიშნულებაა აქტივაციის დონის რაიმე ზღურბლთან შედარების გზით გამოთვალოს გამოსასვლელი სიგნალის მნიშვნელობა. ზღურბლოვანი ფუნქცია განსაზღვრავს აქტიურია თუ არა ნეირონი.

3.2 ქსელის სწავლება

სწავლების პროცესი შეიძლება განხორციელდეს ორი გზით: სწავლება მასწავლებლით (supervised learning), სწავლება მიმდინარეობს და სწავლების პროცესი იმართება ცნობილი ამოხსნების მქონე მაგალითების გამოყენებით და სწავლება მასავლებლის გარეშე (unsupervised).

სწავლება მასწავლებლით: სასწავლი სისტემის შესავალზე მიეწოდება გარემოდან სიგნალი. ასეთი სიგნალის ასლი მიეწოდება მასწავლებლის შესავალზე, რომელიც გამოიმუშავებს სწორ პასუხს. მასწავლებლის პასუხი შეედარება მოსწავლის გამომავალ სიგნალს. სწორ ამონახსნსა და ქსელის შედეგს შორის სხვაობა არის ის შეცდომა, რომლის შემცირებაც არის მიზანი ქსელის თავისუფალი პარამეტრების მორგების საშუალებით. ამისთვის აიგება შეცდომის კვადრატების ჯამი, რომელიც არის ქსელის თავისუფალი პარამეტრების ფუნქცია. ამ ფუნქციის მინიმუმის მოძებნაში მდგომარეობს სწავლების პროცესი.

სწავლება მასწავლებლის გარეშე მიმდინარეობს ქსელის თვითორგანიზაციის პრინციპის საფუძველზე.

აქტივაციის ფუნქცია f - შეიძლება იყოს შემდეგი სახის:

1.ზღრუბლური ფუნქცია (threshold function),

f (u)=1, თუ u ≥0

f (u)=0, თუ u<1

2. სიგმოიდური ფუნქცია (Sigmoid function)

f u)=1/(1+e^{-bu}) , სადაც b>0

3. უბან-უბან წრფივი ფუნქცია (pice -wise -function)

f (u)=1,თუ u>0,5

f (u)=|u|,თუ |u|<0,5

f (u)=0, თუu<0,5

4. ფუნქცია ნიშანი (signum)

f (u)= -1, თუ u <0

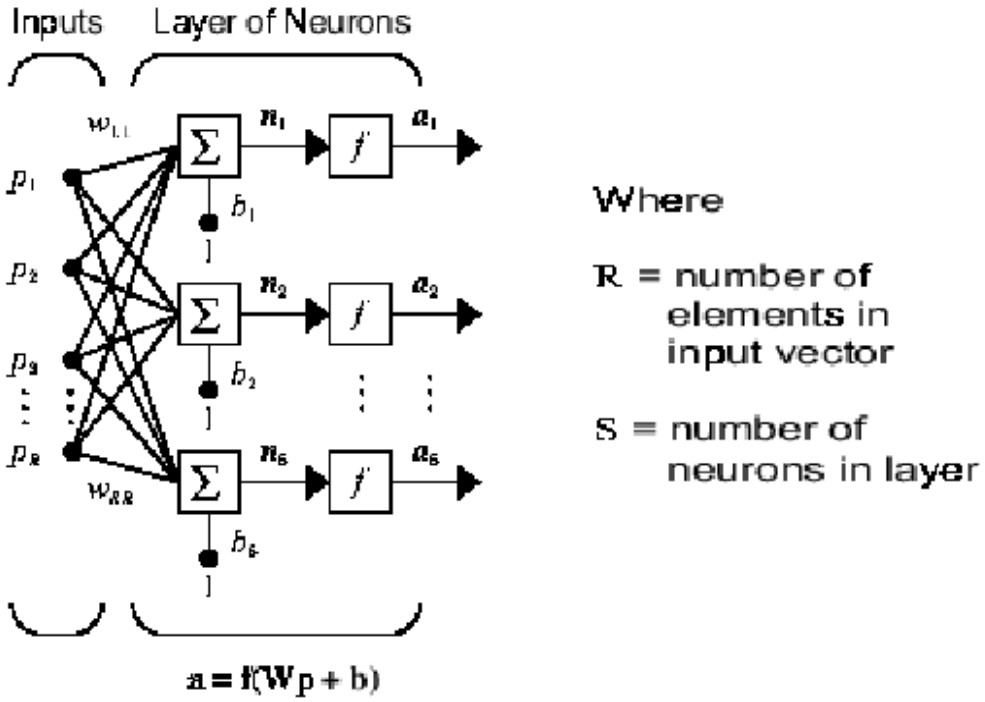
f (u)= 1, თუ u ≥0

3.3 მრავალშრიანი ნეირონული ქსელი (MLP)

მრავალშრიანი პერცეპტრონი შედგება სამი ან მეტი შრისგან (შემავალი და გამომავალი შრეები და ერთი ან მეტი ფარული შრე) და არაწრფივი აქტივაციის კვანძებისგან. ერთი შრიდან თითოეული კვანძს აკავშირებს გარკვეული წონა w_{ij} შემდეგი ფენის ყველა კვანძთან.

მრავალშრიანი პერცეპტრონის ქსელის არქიტექტურა

ნეირონების შრე: ერთ შრიანი S ნეირონის ქსელი ნაჩვენებია სურათ 4.1 ზე სადაც თითოეული R შესასვლელი დაკავშირებულია თითოეულ ნეირონთან და წონების მატრიცას აქვს S რიგი

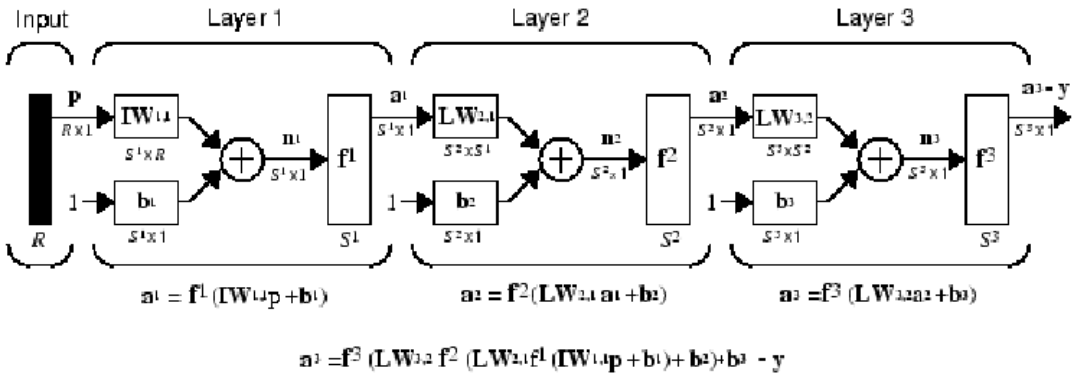


სურათი 3.3.1

შრე მოიცავს წონების მატრიცას, ჯამებს, კონპენსაციის (მიკერძოების) ვექტორი b, გაცვლის ფუნქციას და გამომავალი ვექტორ a-ს. თითოეული ელემენტის შეტანის ვექტორი p არის დაკავშირებული თითოეულ ნეირონთან წონების მატრიცის W საშუალებით. თითოეულ ნეირონს აქვს b_i , ჯამი, გადაცემის ფუნქცია f და გამოსასვლელი a_i . ერთად აღებული გამომავალი სიგნალები ქმნიან გამომავალ ვექტორს a. წონების მატრიცას აქვს შემდეგი სახე:

$$W = \begin{bmatrix} W_{1,1} & \cdots & W_{1,R} \\ \vdots & \ddots & \vdots \\ W_{s,1} & \cdots & W_{S,R} \end{bmatrix}$$

ნეირონების მრავალი შრე: ამ ქსელის თითოეულ ფენას აქვს საკუთარი წონების მატრიცა W, აქვს საკუთარი კომპენსაციის ვექტორი b ქსელის შემავალი ვექტორი n და გამომავალი ვექტორი a. გარკვეული ჩანაწერებით უნდა განსხვავებდეთ ამ ფენებს. ამისთვის გამოიყენება საბსკრიპტი. მაგალითად მეორე შრის წონების მატრიცა ჩაიწერება W^2. სურათზე 4.2 მოცემულია სამ შრიანი ნეირონული ქსელი:



სურათი 3.3.2

როგორც სურათზეა ნაჩვენები არის R შესასვლელი, S^1 ნეირონი პირველ შრეში, S^2 ნეირონი მეორე შრეში და ა.შ. პირველ შრეს ეწოდება შესასვლელი შრე, მეორე შრე შეიძლება ჩაითვალოს როგორც ერთშრიანი ნეირონული ქსელი რომლისთვისაც $R=S^1$ არის შემავალი სიგნალი, $S = S^2$ ნეირონით და $S^{1*} S^2$ წონების მატრიცით. შრეს რომლის გამოსასვლელი არის ქსელის გამოსასვლელი ეწოდება გამომავალი შრე. სხვა ფენებს ეწოდება ფარული შრეები. სურათზე ნაჩვენებ ქსელს აქვს გამომავალი შრე(მესამე შრე) და ორი ფარული შრე(პირველი და მეორე შრე).

3.4 მრავალშრიანი ნეირონული ქსელი (MLP) როგორც ფსევდო შემთხვევითი რიცხვების გენერატორი.

ჩვენ ვიყენებთ ნეირონულ ქსელებს, როგორც შავი ყუთი, რომელსაც შეუძლია გადაჭრას კონკრეტული პრობლემები. გარკვეული მონაცემების შეყვანა წარმოშობს სასურველ გამომავალ შედეგს. ამ თვალსაზრისით ნეირონული ქსელები არიან პროგნოზირებადი სისტემები.

მაგრამ თუ ნეირონული ქსელის სწავლების პროცესში მეტი fitting-ის პრობლემა მოხდება, ნეირონული ქსელი მოგვცემს არაპროგნოზირებად გამომავალ სიგნალს. ეს მახასიათებელი ხშირად გამოიყენება ნეირონულ ქსელებზე დაფუძნებულ შემთხვევითი რიცხვების გენერატორებში. მრავალშრიანი პერცეპტონი არის ერთ-ერთი ყველაზე ცნობილი ნეირონული ქსელი, რომელსაც აქვს ეს თვისება. MLP სწავლობს ყველა მონაცემით,

რომელიც არის წარმოდგენილი და ასევე შეიძლება განზოგადება იმ მონაცემებზე, რომლებიც არ იყო წარმოდგენილი სწავლების დროს. MLP შეესაბამება სასურველ ზედაპირს, რომელიც წარმოდგენილია შეტანილი მონაცემებით. თუ მეტი fitting მოხდება სწავლების დროს შემდეგ ცდა MLP მოერგოს სასურველ ზედაპირს მაღალი ხარისხისაა. ასე რომ ჩვენ არ შეგვიძლია პროგნოზირება გამოსასვლელის , როცა ქსელი იღებს ახალ სიგნალს. ამის მიზეზი არის, რომ ჩვენ არ ვიცით რა ზედაპირს შეესაბამება MLP. ეს შემთხვევითი რიცხვების გენერატორები წარმოადგენენ მძლავრ შესაძლებლობას შემთხვევითი რიცხვების გენერირებისთვის.

3.5 დელტა წესი

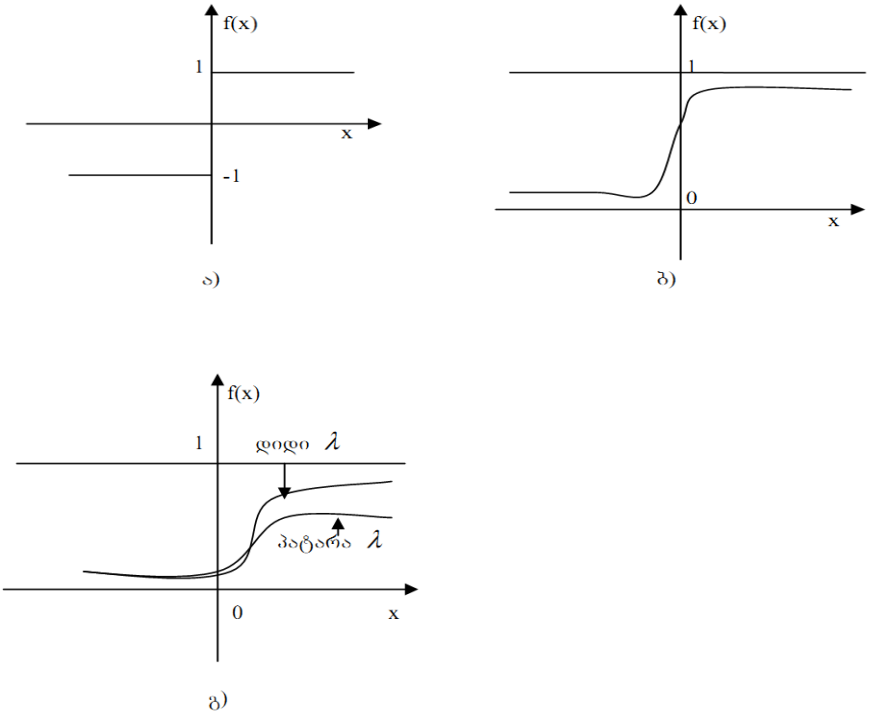
იმისათვის, რომ განვაზოგადოთ პერცეპტრონის იდეა და ავაგოთ დასწავლის უფრო დახვეწილი ალგორითმები, საჭიროა შევცვალოთ მისი ზღურბლოვანი ფუნქცია და გავხადოთ ის უფრო მოქნილი და უწყვეტი ფუნქცია, რომელიც საშუალებას მოგვცემს უფრო ზუსტად გამოვთვალოთ წონები. ერთ-ერთ ასეთ ფუნქციას წარმოადგენს სიგმაიდური ფუნქცია.

ტიპური სიგმოიდური აქტივაციური ფუნქცია, ანუ ლოგისტიკული ფუნქცია მოიცემა განტოლებით:

$$f(NET) = 1/(1+e^{-\lambda net})$$

სადაც $NET = \sum w_i x_i$. აქ λ არის ამოზნექილობის პარამეტრი, რომელიც გამოიყენება სიგმოიდური წირის ფორმის აგებისათვის. λ-ს დიდი მნიშვნელობებისთვის სიგმოიდას ფორმა უახლოვდება ჩვეულებრივ წყვეტილ-წრფივ ზღურბლურ ფუნქციას, რომლის მნიშვნელობები მოთავსებულია {0,1} დიაპაზონში, ხოლო სიმრუდის პარამეტრის იმ მნიშვნელობებისათვის, რომლებიც ახლოს არიან 1-თან, სიგმოიდური წირი ჰგავს სწორ ხაზს. ზღურბლური ფუნქციის არგუმენტს წარმოადგენს ნეირონის აქტივაციის დონე, ხოლო მისი მნიშვნელობაა ნეირონის გამოსასვლელი. სიგმოიდური აქტივაციური ფუნქცია არის უწყვეტი რაც, იძლევა ქსელის გამოსასვლელზე შეცდომების უფრო ზუსტად შეფასების საშუალებას. ჩვეულებრივი ზღურბლური ფუნქციის მსგავსად, სიგმოიდური აქტივაციური ფუნქცია ასახავს განსაზღვრის არის წერტილებს (0,1) ინტერვალში მოთავსებულ მნიშვნელობებში. თუმცა უნდა ითქვას ისიც, რომ ჩვეულებრივი ზღურბლური ფუნქციისგან

განსხვავებით სიგმოიდას შეუძლია მიიღოს მნიშვნელობები მთელი ინტერვალიდან. ე.ი. ის უზრუნველყოფს კლასიკური ზღურბლური ფუნქციის უწყვეტ აპროქსიმაციას. λ პარამეტრი განსაზღვრავს გადასვლის სიმრუდეს.

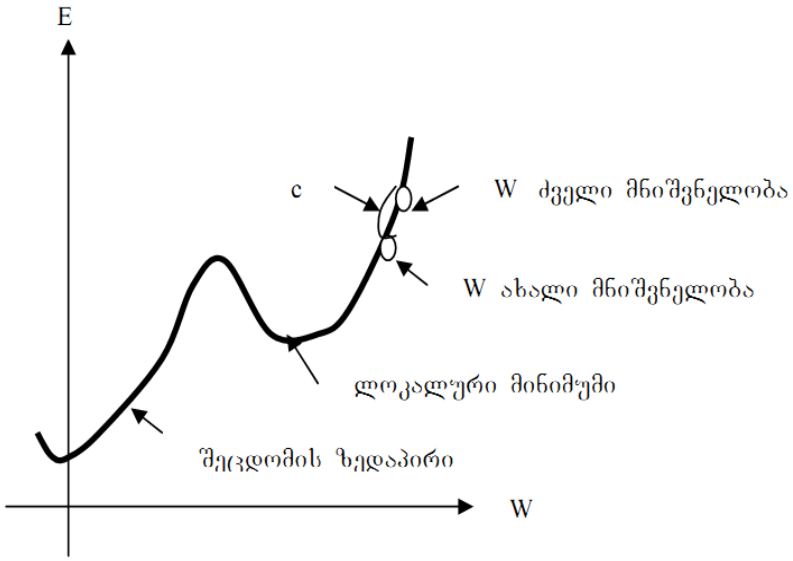


სურათი 3.5.1

ა) წყვეტილ- წრფივი ბიპოლარული ფუნქცია;

ბ) სიგმოიდური უნიპოლარული ზღურბლოვანი ფუნქცია;

გ) წაძრული სიგმოიდური ფუნქცია სხვადასხვა სიმრუდით.



სურათი 3.5.2 შეცდომის ორგანზომილებიანი სიბრტყე. c კონსტანტა განსაზღვრავს დასწავლის ბიჯის ზომას.

უწყვეტი აქტივაციური ფუნქციების მქონე ქსელებისათვის სწავლების ყველაზე მნიშვნელოვან წესს წარმოადგენს დელტა-წესი. ინტიუციურად დელტა-წესი ეფუძნება ისეთ ცნებას, როგორიცაა შეცდომის ზედაპირი (სურ. 3.5.2), რომელიც მონაცემთა მთელი ნაკადისათვის ჯამურ შეცდომას განსაზღვრავს როგორც ფუნქციას დამოკიდებულს ქსელის წონებზე. წონების ყოველი შესაძლო კონფიგურაცია განსაზღვრავს შეცდომის ზედაპირის წერტილს. გვაქვს რა წონების გარკვეული კონფიგურაცია, სწავლების ალგორითმის საშუალებით შეიძლება ვიპოვოთ ამ ზედაპირზე მიმართულება, რომლის გასწვრივაც ყველაზე სწრაფად ხდება შეცდომის ფუნქციის შემცირება. ამ მიდგომას ეწოდება სწავლება გრადიენტული დაშვების მეთოდით, რადგანაც გრადიენტი განსაზღვრავს ზედაპირის დახრილობას მის ყოველ წერტილში.

დელტა-წესი გვთავაზობს უწყვეტი და დიფერენცირებადი აქტივაციური ფუნქციის გამოყენებას. ეს თვისებები გააჩნია ზემოდ განხილულ ლოგისტიკურ ფუნქციას. დელტა-წესს ქსელის i-ური კვანძის j-ური წონითი კოეფიციენტის რეგულირებისათვის აქვს შემდეგი სახე:

$$\Delta w_k = c(d_i - O_i) f'(net_i) x_i$$

სადაც c არის სწავლების სიჩქარის მუდმივი კოეფიციენტი, d_i-ური და O_i-ური i-ური ნეირონის მოსალოდნელი და რეალური გამოსასვლელები, f' i-ური კვანძის

აქტივაციური ფუნქციის წარმოებული, ხოლო x_j i-ური კვანძის j-ური შესასვლელის მნიშვნელობა.

3.6 შეცდომის უკუგავრცელების მეთოდი

უკუგავრცელების მეთოდი წარმოადგენს დელტა-წესის განზოგადოებას. აქაც გამოიყენება გრადიენტული დაშვების მეთოდი. დაფარული შრის კვანძებისათვის შეცდომა გამოითვლება გამომავალი შრის შეცდომის საფუძველზე. შეცდომების უკუგავრცელების მეთოდის გამოყენების შემთხვევაში k და i ნეირონებს შორის კავშირის w_{ki} წონითი კოეფიციენტის ცვლილება შესასვლელი შრის კვანძებისათვის გამოითვლება ფორმულით:

$$\Delta w_{ki} = -\lambda c(d_i - O_i)O_i(1-O_i)x_k$$

და დაფარული შრეების კვანძებისათვის კი -

$$\Delta w_{ki} = -\lambda c O_i(1-O_i)\sum_j(-delta_j w_{ij})x_k$$

მეორე ფორმულაში j არის შემდეგი შრის კვანძის ინდექსი, სადამდისაც ვრცელდება სიგნალი i - ური ნეირონიდან და

$$delta_j = -\frac{\partial E}{\partial net_j} = (d_i - O_i)O_i(1-O_i)$$

ქსელებში, რომლებიც შეიცავენ რამდენიმე დაფარულ შრეს, შეცდომის გავრცელების ეს პროცედურა გამოიყენება რეკურსიულად n-ური დაფარულ შრიდან $(n-1)$ დაფარულ შრისაკენ. შეცდომის უკუგავრცელების მეთოდი იძლევა მრავალშრიანი ქსელების სწავლების პრობლემის გადაწყვეტის საშუალებას.

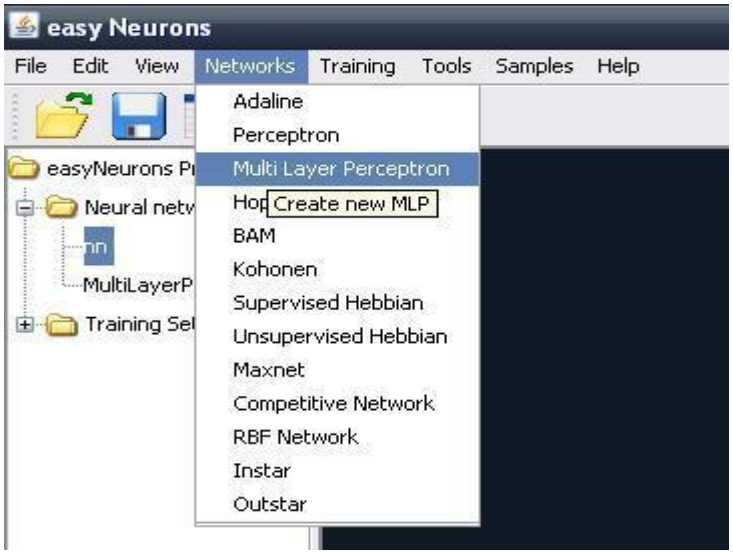
თავი 4. ამოცანის დასმა

ზემოთ განხილული თავებიდან ჩანს, რომ ნაკადური შიფრი ფართოდ გამოიყენება კრიპტოგრაფიაში, ხოლო საიმედო გამის გამომუშავება დაკავშირებულია გარკვეულ სიძნელეებთან. არსებობს ნაკადური შიფრებისთვის გამის მისაღები ფსევდო შემთხვევითი რიცხვების მრავალი გენერატორი, მაგრამ როგორც უკვე ვთქვით ზემოთ, მათ შორის ყველაზე კარგია ცალმხრივმიმართულ ფუნქციაზე დაფუძნებული გენერატორები, როგორებიცაა RSA და Blum Blum Shub. ეს გენერატორები გვაძლევენ საიმედო და კრიპტომედეგ გამას, მაგრამ ამ გენერატორების ნაკლი არის ის, რომ ისინი არიან საკმაოდ ნელი გენერატორები.

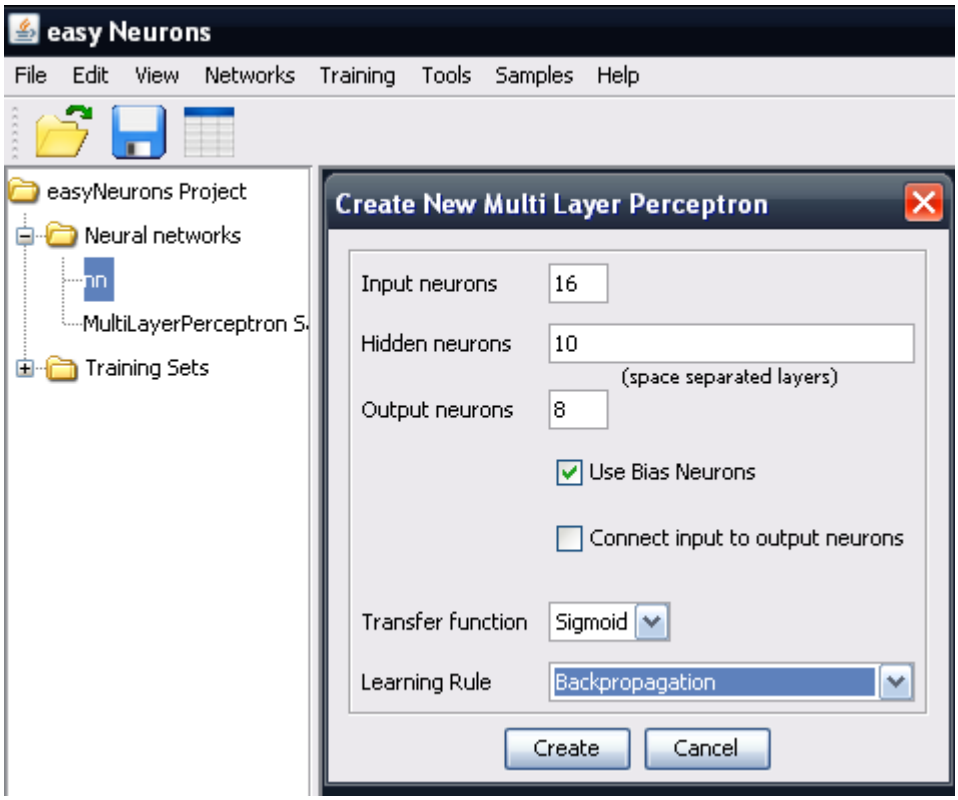
სწორედ ამიტომ, ჩვენ გადავწყვიტეთ აგვეგო მრავალშრიანი ნეირონული ქსელი (MLP), რომელსაც ვასწავლიდით RSA გენერატორის გამომუშავებული მიმდევრობის მსგავსი მიმდევრობის გამომუშავებას. ამისათვის ავაგეთ RSA გენერატორი, რომელიც წარმოადგენს RSA ალგორითმის მოდიფიკაციას. RSA გენერატორი რეალიზებულია მათლაბში და გენერატორი გამოსასვლელზე გვაძლევს რვა ბიტიან შემთხვევით მიმდევრობას. ვაგროვებთ ამ მიმდევრობებს და ამ მიმდევრობების გამოყენებით ვასწავლით შემდეგ ნეირონულ ქსელს მსგავსი მიმდევრობების გამომუშავებას.

მრავალშრიანი ნეირონული ქსელის ასაგებად გამოვიყენეთ სპეციალური პროგრამა neuroph studio (neuroph_2.4u1), რომელსაც საშუალება აქვს ავაგოთ ჩვენთვის სასურველი ქსელი და დასწავლის ფუნქცია რომელსაც გადავსცემთ ჩვენთვის სასურველ მონაცემებს.

ქვემოთ განხილული და მოყვანილია მაგალითი მრავალშრიანი ნეირონული ქსელის აგებისა და ნაჩვენებია ქსელის მიერ გამომუშავებული შედეგი.



სურათი 4.1 MLP აგება



სურათი4.2

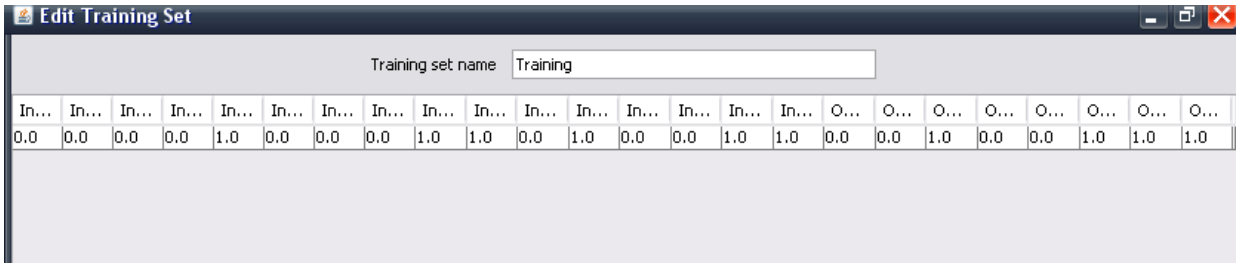
სურათზე 4.1 და 4.2 ნაჩვენებია მრავალშრიანი ნეირონული ქსელის აგების მაგალითი, ქსელი რომელსაც ჩვენ ვაგებთ იქნება სამ შრიანი ერთი შესასვლელი შრე, ერთი ფარული შრე და ერთიც გამომავალი შრე. შესასვლელი შრე შედგება 16 ნეირონისგან, ფარული შრე- 10 ნეირონისგან, ხოლო გამომავალი შრე 8 ნეირონისგან გამომავალი ნეირონების რიცხვი ემთხვევა RSA გენერატორის მიერ გამომუშავებული მიმდევრობის სიგრძეს. ქსელი იყენებს სიგმოიდურ ფუნქციას და ქსელის სწავლების წესი არის შეცდომის უკუგავრცელების მეთოდი.

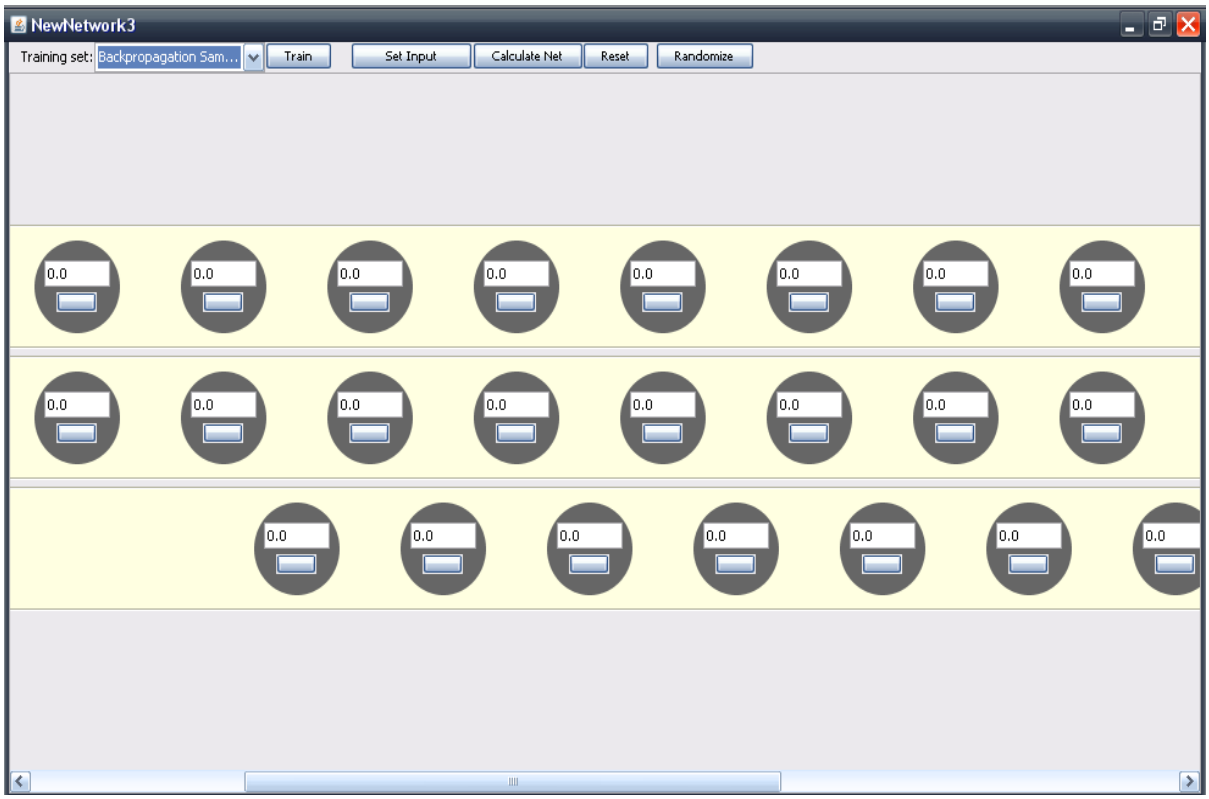
სურათზე 4.3 ნაჩვენებია MLP ქსელის ბლოკ სქემა 16 შესასვლელით და 8 გამოსასვლელით და ერთი ფარული შრით.

სურათზე 4.4 ნაჩვენებია სასწავლო კომლლექტის შექმნა რომლის მიხედვითაც ხდება ქსელის სწავლება. სწავლების ფუნქციას ექნება 16 შემავალი მონაცემი და 8 გამომავალი მონაცემი

სურათ 4.5- ზე ნაჩვენებია სასწვალი მონაცემების შევსება ფაილიდან, ფაილში შენახული გვაქვს RSA - ს შემავალი თექვსმეტ ბიტიანი და გამომუშავებული რვა ბიტიანი მიმდევრობა. ფაილში პირველი 16 ბიტი წარმოადგენს შემავალ მონაცემებს ხოლო ბოლო 8 გამომავალ

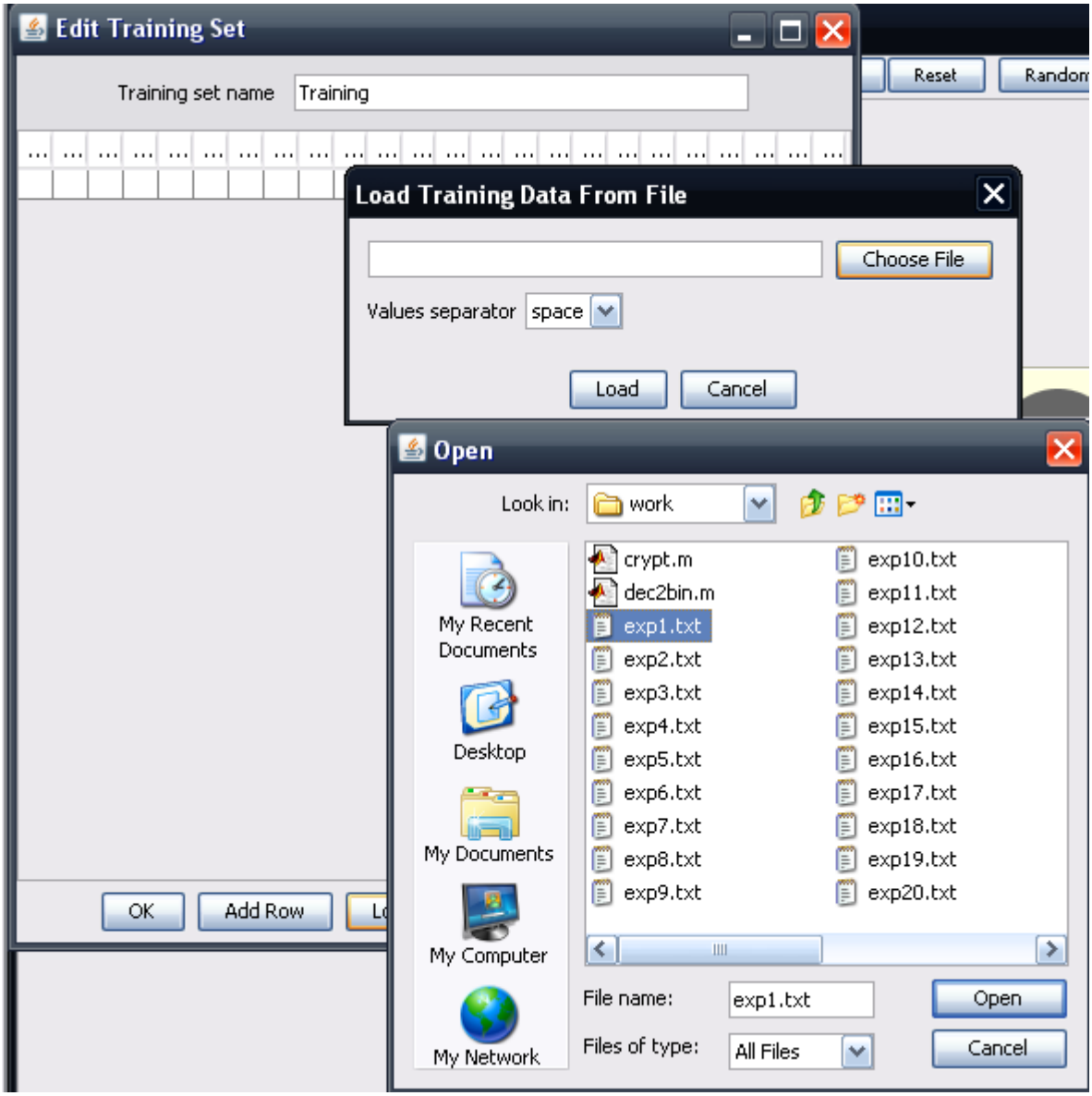
მონაცემებს.



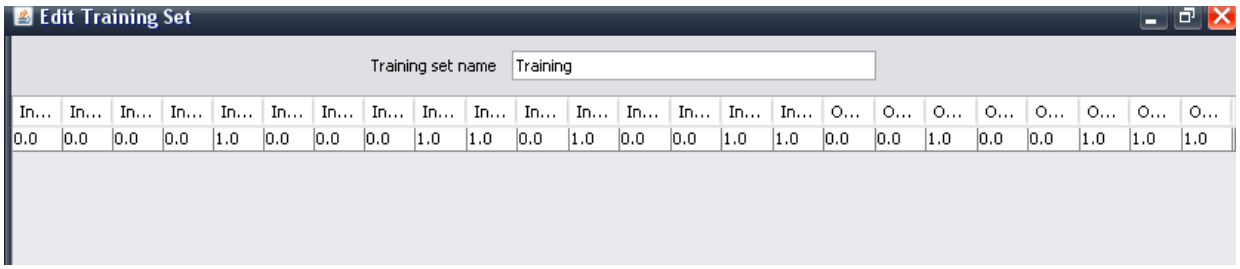


სურათი 4.3 MLP ქსელი

სურათი 4.4 სასწავლო კომპლექტის შექმნა

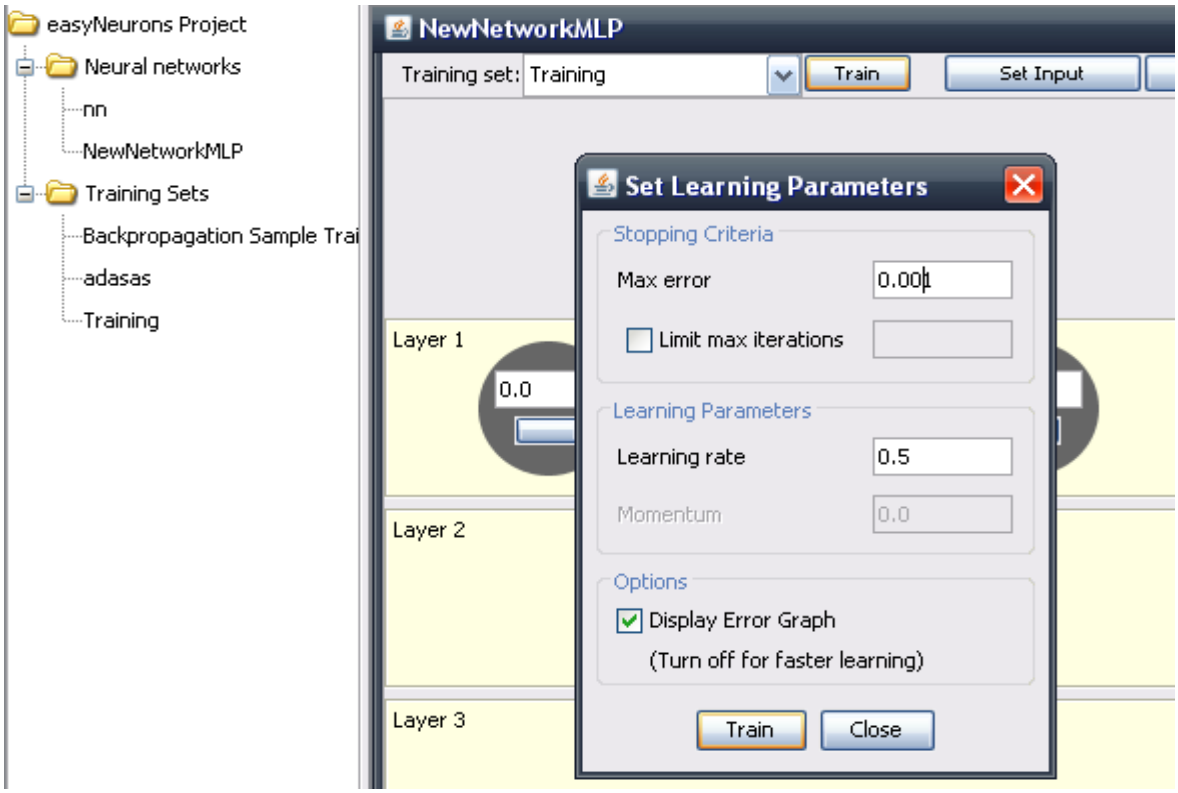


1

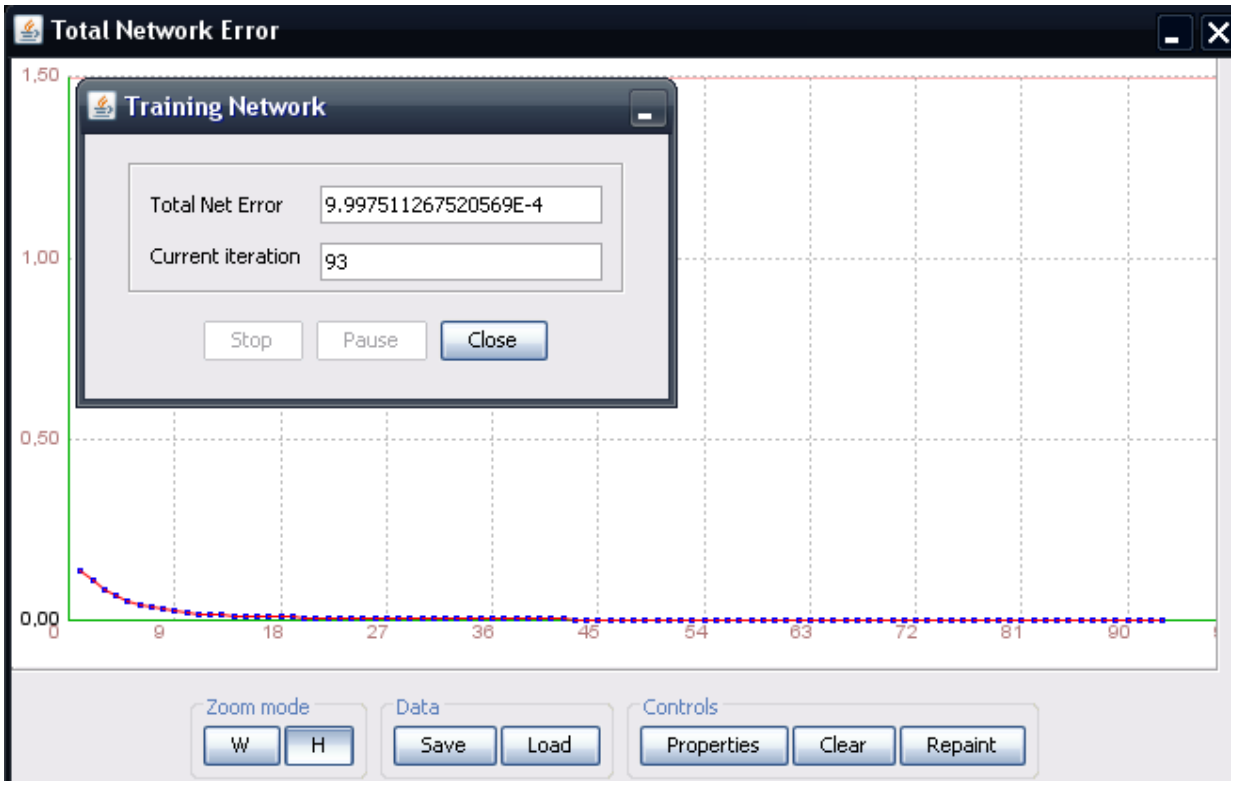


2

სურათი 4.5 1)სასწავლი მონაცემების შევსება ფაილიდან. 2) შევსებული მონაცემებით



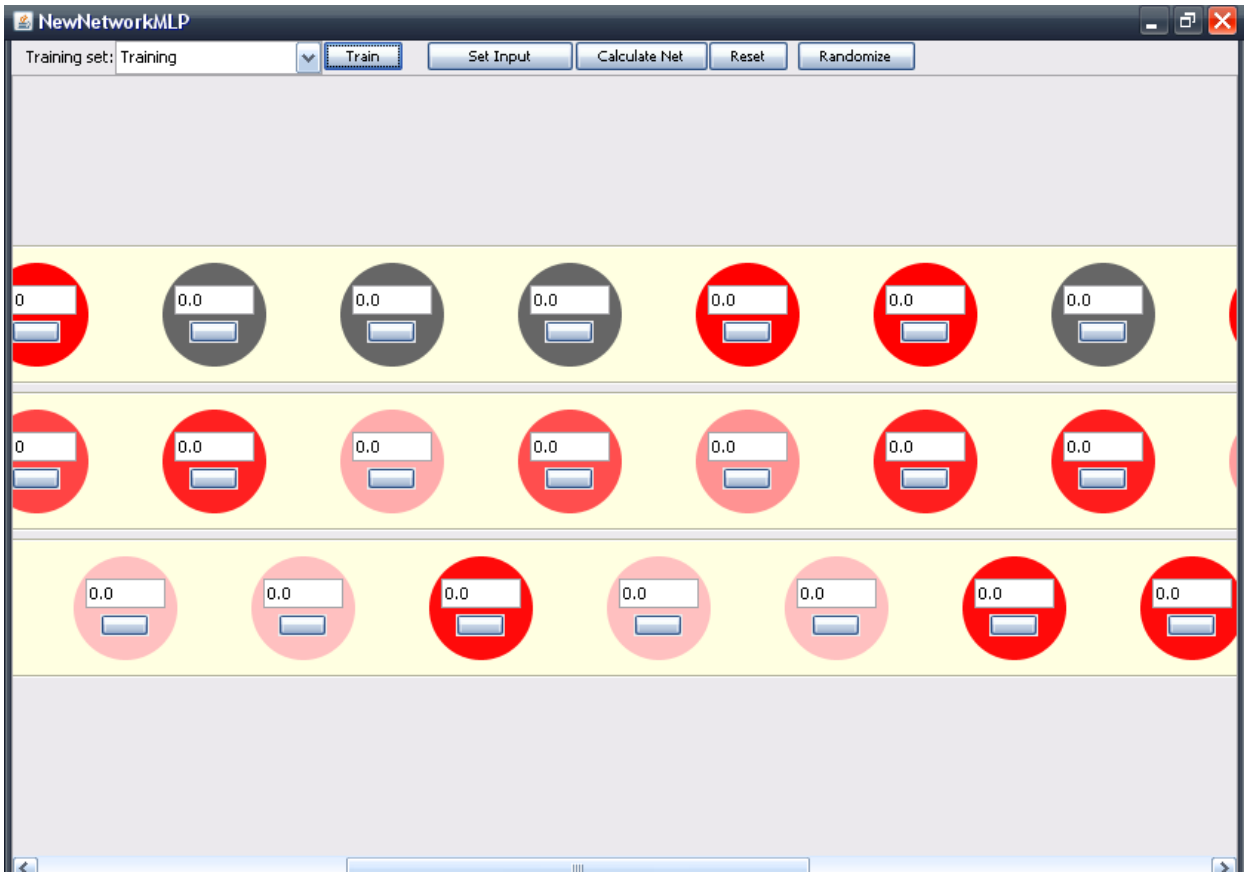
1.



2.

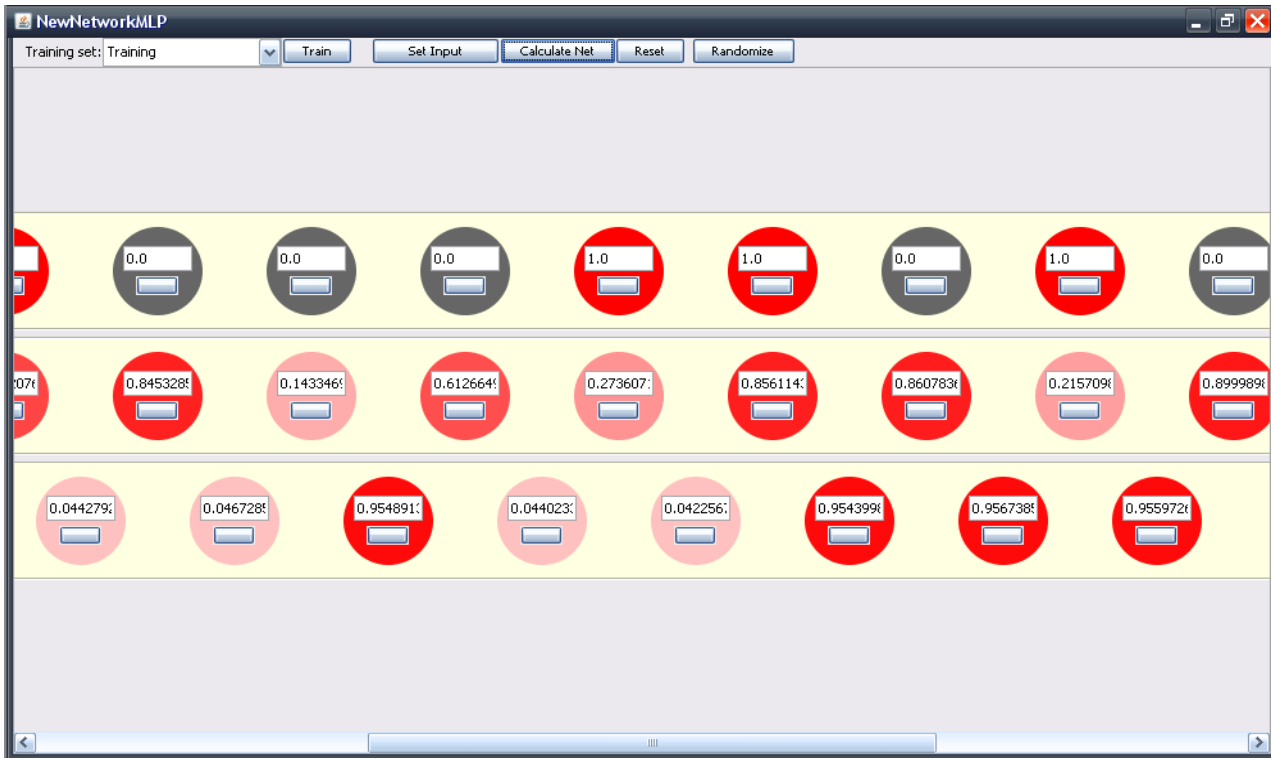
სურათი 4.6 ქსელის სწავლება

სურათზე 4.6 ნაჩვენებია ქსელის სწავლების პროცესი როგორც სურათიდან ჩანს training sets ჩამონათვალში გამოჩნდა ჩვენი შექმნილი training. ნეირონული ქსელის training set ველში მოვნიშნავთ ჩვენ შექმნილ training -ს და დავაჭერთ train ღილაკს რის შედეგადაც მოხდება ქსელის სწავლება. რის შემდეგაც გამოდის შეცდომების გრაფიკი, როგორც სურათზე ჩანს კონკრეტული მონაცემებისთვის ქსელს დასჭირდა 93 იტერაცია რათა შეცდომები მინიმუმამდე დაეყვანა.

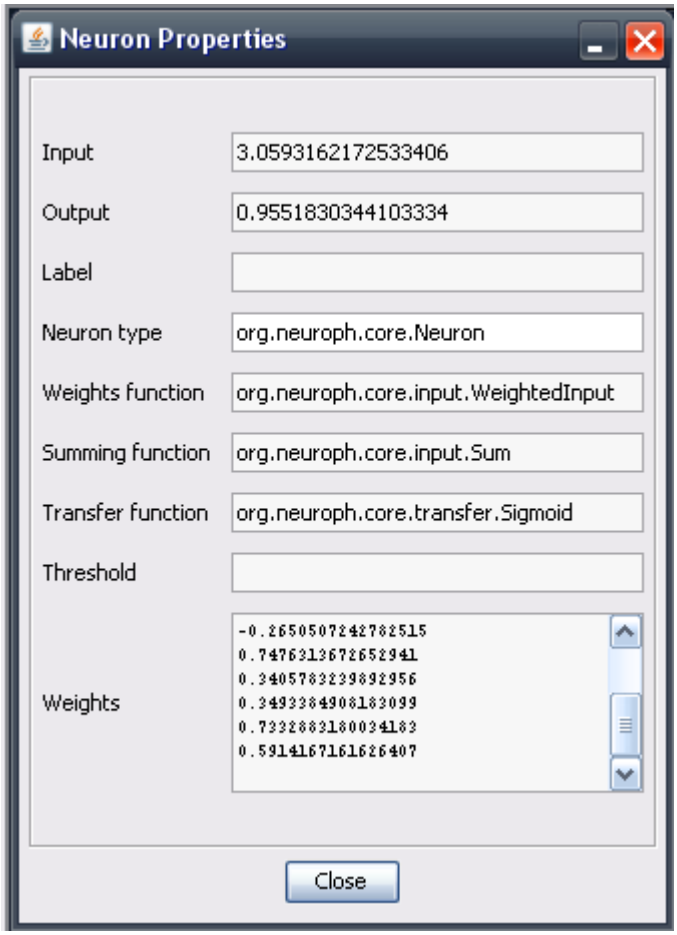


სურათი 4.7 ქსელი სწავლების შემდეგ

სურათზე ნაჩვენებია ქსელის სწავლების შედეგი. სურათზე წითლად აღნიშნულია ის ნეირონები რომლებიც აქტიურია ანუ შემავალ შრეში ეს ნეირონები იღებს მონაცემებს მნიშვნელობით 1, ხოლო გამომავალ შრეში გამოსასვლეზე გვაძლევს მნიშვნელობას რომელიც ახლოსაა ერთთან. შავი ფერით აღნიშნული ნეირონები რომლებიც იღებენ მნიშვნელობას 0-ს, ხოლო ვარდისფრად აღნიშნულია ნეირონები რომლებიც გამოსავალზე გვაძლევენ ნოლთან მიახლოებულ მნიშვნელობებს. ღილაკ calculate net -ზე დაჭერით გამოითვლება ქსელის მნიშვნელობები. შემდეგ სურათზე მოცემული უკვე გამოთვლილი ქსელის მნიშვნელობები.



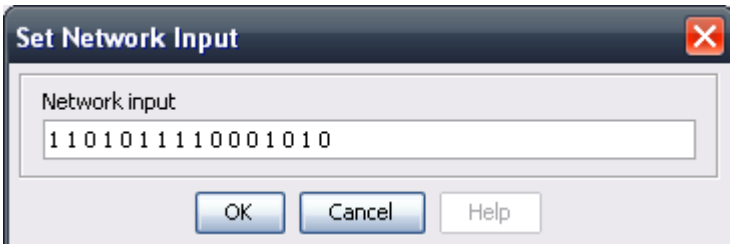
სურათი 4.8 ნეირონული ქსელი გამოთვლილი მნიშვნელობებით.



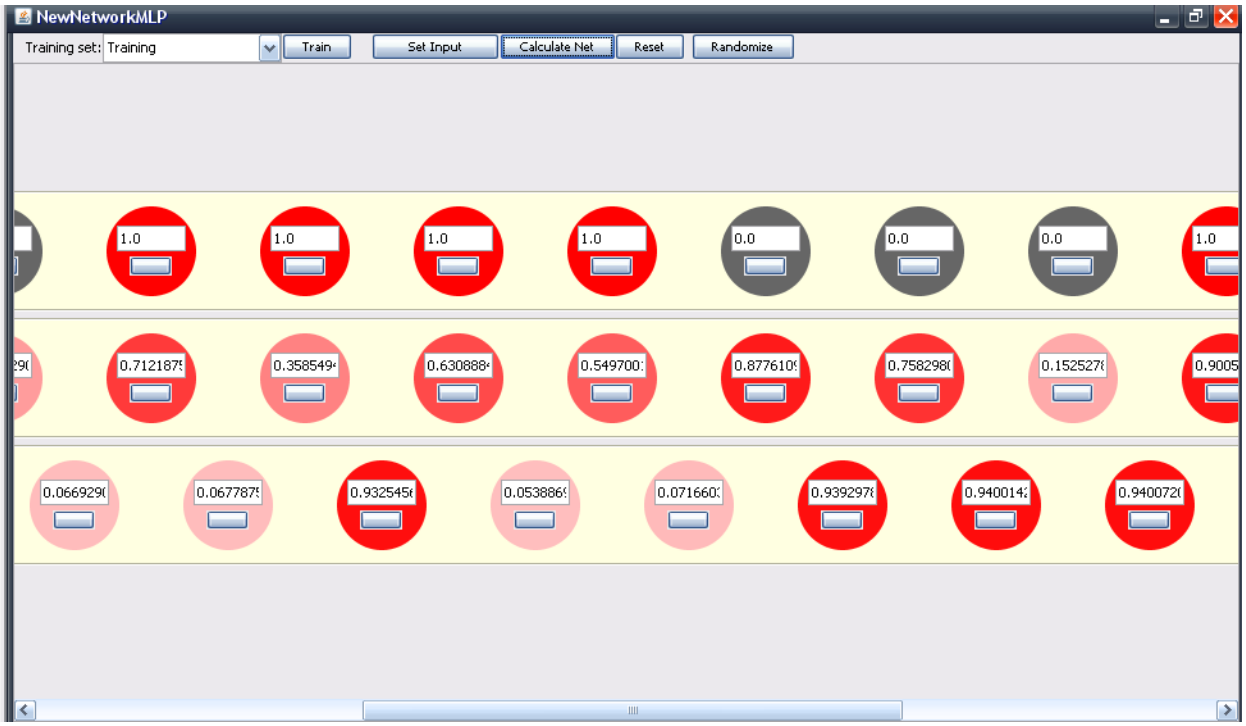
სურათი 4.9

ქსელის სწავლება განხორციელდა წარმატებით ჩვენს მიერ გადაცემული გამომავალი მნიშვნელობები სასწავლო კომლექტში (learning set) იყო 0 0 1 0 0 1 1 1 და როგორც ვხედავთ ქსელმაც გამოსასვლელზე მოგვცა მიმდევობა რომელიც საკმაოდ ახლოსაა აღნიშნულ მიმდევრობასთან. ასევე შეგვიძლია ვნახოთ თითოეული ნეირონის მახასიათებლები მაგალითად გამოსასვლელი შრის მესამე ნეირონის მახასიათებლები ნაჩვენებია სურათზე 4.9

ქსელს რომელმაც დასწავლის პროცესი წარმატებით გაიარა ასევე შეგვიძლია გადავცეთ შემავალი სხვა მნიშვნელობები და ქსელი მოგვცემს იგივე გამომავალ მნიშვნელობას, რაც სწავლების დროს გადაეცა. ეს თვისება წარმოადგენს MLP ქსელების თვისებას მოახდინოს სწავლების განზოგადება ისეთ მონაცემებზე, რომლებიც სწავლების პროცესში არ იყო წარმოდგენილი. ახლა ვნახოთ შესასვლელი სხვა მონაცემებისთვის რას მოგვცემს ქსელი:

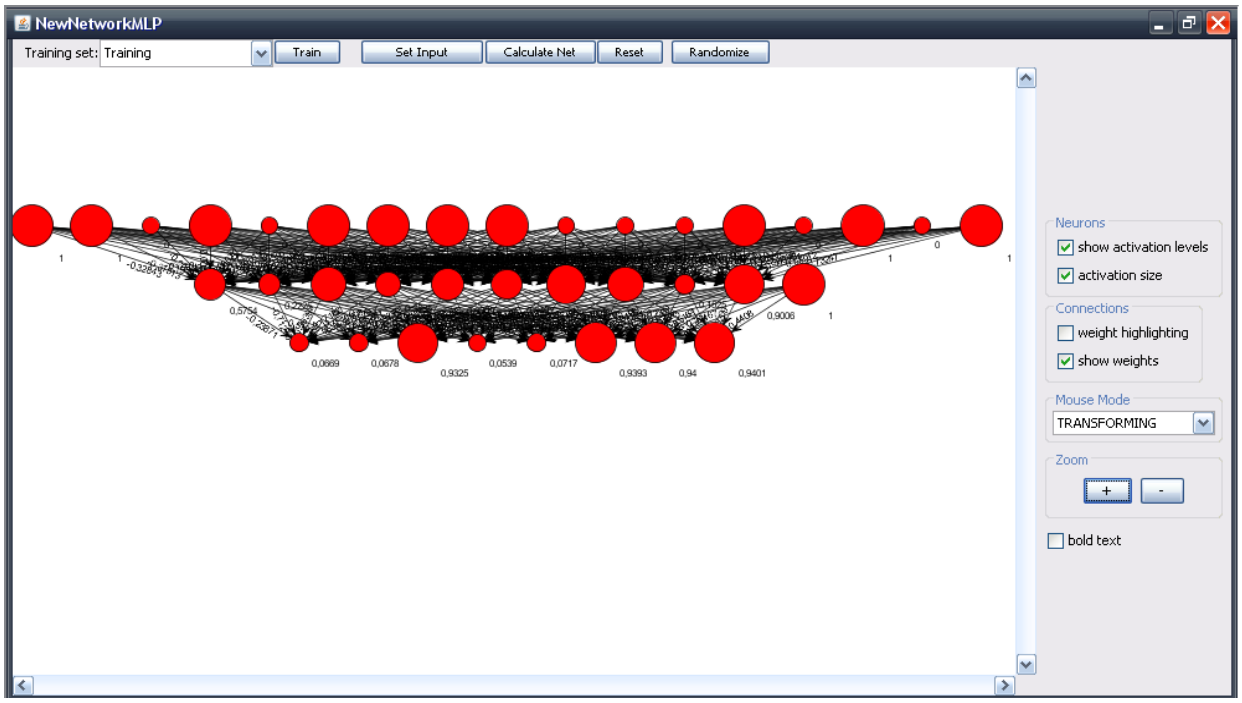


სურათი 4.10 ქსელის შესასვლელი მონაცემები



სურათი 4.11 ნეირონული ქსელის გამოსასვლელი მნიშვნელობები სხვა მონაცემებისთვის

როგორც სურათებიდან ჩანს ქსელმა გამოსასვლელზე ისევე მოგვცა თავდაპირველად გადაცემულ მიმდევრობის (0 0 1 0 0 1 1 1) მსგავსი მიმდევრობა. სურათი 4.12 წარმოადგენს ჩვენი ნეირონული ქსელის გრაფიკულ მოდელს.



4.11 ნეირონული ქსელის გრაფიკული მოდელი

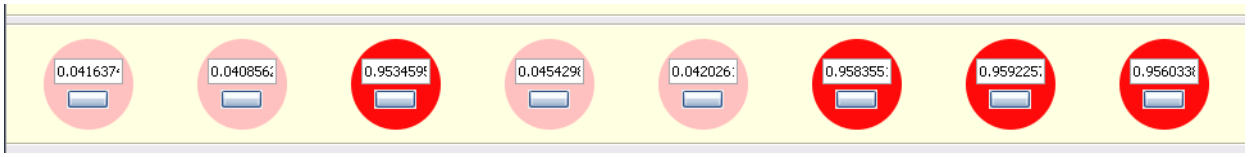
ზემოთ განხილული მაგალითის მსგავსად ჩავატარეთ ცდები:

1. ნეირონულ ქსელს ვასწავლეთ ჩვენთვის სასურველი მნიშვნელობების გამომუშავება. გავიმეორეთ ეს პროცესი დაახლოებით 25 სხვადასხვა მნიშვნელობისათვის.
2. მრავალშრიან ნეირონულ ქსელს ვასწავლეთ ერთი ტიპის მონაცემებით და შემდეგ შესასვლელ შრეზე გადავეცით სხვა მონაცემები.
3. ქსელს ვასწავლეთ ჩვენი მონაცემებით და სწავლების პროცესი ერთიდაიგივე მონაცემებისთვის გავიმეორეთ რამდენჯერმე და ვამოწმებდით გამოსასვლელ მონაცემებს.

თითოეული ტიპის ცდა ჩავატარეთ დაახლოებით 25 სხვადასხვა მონაცემისთვის, რომლებიც მიღებული გვქონდა RSA გენერატორისგან. ამ ცდებიდან უმეტესობამ მოგვცა კარგი შედეგი.

პირველი ტიპის ცდის ჩატარებისას ნეირონული ქსელის სწავლება განხორციელდა წარმატებით. ნეირონული ქსელი, ყოველი ახალი მონაცემებით სწავლებისას გამოსასვლელზე ყოველთვის გვაძლევს სასურველ მიმდევრობას, იმ მიმდევრობას, რომელიც გადავეცით დასწავლის პროცესში ქსელს.

მეორე ტიპის ცდის ჩატარების დროს ჩვენ ერთხელ მოვახდინეთ ქსელის სწავლება და შემდეგ ვცვლიდით მარტო შესასვლელ მონაცემებს, ეს არის MLP ქსელის თვისება სწავლება განაზოგადოს სხვა მონაცემებზეც, რომლებიც არ იყო წარმოდგენილი სწავლების დროს. ქსელის დასასწავლი მონაცემები იყო: სასწავლო კომპლექტის შესასვლელი მონაცემები - 0 0 0 0 1 0 0 0 1 1 0 1 0 0 1 1 , სასწავლო კომპლექტის გამოსასვლელი მონაცემები - 0 0 1 0 0 1 1 1 ქსელმა რომელმაც წარმატებით გაიარა დასწავლის პროცესი უნდა მოგვცეს 0 0 1 0 0 1 1 1 მიმდევრობასთან მაქსიმალურად ახლოს მყოფი მიმდევრობა (უნდა აღინიშნოს, რომ ნეირონები გამოსასვლელზე გვაძლევს მიიახლოებულ მნიშვნელობებს რომლების დამრგვალებითაც მიციღებთ სასურველ მიმდევრობას ნოლებისა და ერთების, მთავარია ნეირონმა გამოსასვლელზე მოგვცეს მაქსიმალურად ახლოს მყოფი მნიშვნელობა ნოლთან ან ერთთან) ქსელმა სწავლების შემდეგ გამომავალ მონაცემებად მოგვცა შემდეგი მიმდევრობა (სურათი (0.0416, 0.0408, 0.953, 0.045, 0.042, 0.958, 0.959, 0.956), როგორც ჩანს ეს მიმდევრობა წარმოადგენს საკმაოდ კარგ მიმდევრობას რადგან ახლოს დგას იმ მიმდევრობასთან რომელიც გვინდა რომ მივიღოთ. ცდის შედეგებმა აჩვენეს, რომ ჩატარებული ცდებიდან უმრავლესობა გვაძლევს კარგ შედეგს კერძოდ ოცმა ცდამ აჩვენა რომ ნეირონული ქსელის გამოსასვლელზე საშუალოდ არც ისე კარგ შედეგს იძლევა შესასვლელი 3 მიმდევრობა დანარჩენი 17 ცდა გვაძლევს კარგ მიმდევრობას.



სურათი 4.12 ქსელის გამოსასვლელი მიმდევრობა.

მესამე ტიპის ცდის ჩატარებამ აჩვენა, რომ (მაგ: შესასვლელი მონაცემებით 0 0 1 0 0 0 0 1 1 1 0 1 1 0 1 0 და გამოსასვლელი მონაცემებით 0 1 1 1 1 0 0 0) გამოსასვლელი მიმდევრობები თითქმის არ იცვლება, ანუ იცვლება მაგრამ უმნიშვნელოდ, რადგან ჩვენ გამოსასვლელზე ვიღებთ არა ნოლების და ერთების მიმდევრობას არამედ ათწილად რიცხვებს ცვლილება

სხავადასხვა გამოსასვლელებზე მაინც არის მაგრამ ძალიან უმნიშვნელო ამ ცდებმაც აჩვენა, რომ ნეირონული ქსელი გვაძლეს კარგ მიმდევრობას ნაკადურ შიფრებში გამოსაყენებლად.

ამასთან ნეირონული ქსელის სწავლებაზე და ქსელით მიმდევრობების მიღებაზე იხარჯება გაცილებით ნაკლები დრო ვიდრე RSA გენერატორით მიმდევრობების გამომუშავებაზე. მრავალშრიან ნეირონულ ქსელს რომელიც აგენერირებს მიმდევრობებს, სწავლებისთვის მხოლოდ პირველჯერზე ჭირდება გარკვეული დრო, მაგალითად როგორც ზევით მაგალითშია მოცემული ქსელმა სწავლებისთვის გამოიყენა 93 იტერაცია (სურათი 4.6) ერთხელ დასწავლილ ქსელს კი, როგორც ცდებმა აჩვენა, შეუძლია მიიღოს საკმაოდ კარგი მიმდევრობები მაშინაც კი, როცა შესასვლელი მონაცემები იცვლება. ხოლო RSA გენერატორით მიმდევრობების მისაღებად საჭიროა უფრო მეტი დრო, რადგან მან ყოველ ჯერზე უნდა მოახდინოს ორი მარტივი რიცხვის ამორჩევა, შემოწმება, შემავალი მონაცემების გარდაქმნა, დაშიფვრა და შემდეგ მოგვცემს მიმდევრობას, რასაც საკმაოდ დიდი დრო ჭირდება. ხოლო ნეირონული ქსელი ერთხელ დასწავლილი მონაცემებით გვაძლევს იგივე კრიპტოგრაფიულად მდგრად მიმდევრობას.

თავი 5. პროგრამული უზრუნველყოფა

5.1 RSA გენერატორი

RSA გენერატორი რეალიზებულია matlab-ში.

პარამეტრების ინიციალიზაციის ფუნქცია

function [Pk,Phi,d,e] = intialize(p,q) // პარამეტრების გამოსათვლელად ფუნქციას გადაეცემა p და q მარტივი რიცხვები.

```
clc;

disp('Intaializing:');

Pk=p*q;                    // მოდულის ფუძე n = p · q

Phi=(p-1)*(q-1);           // φ(n) = (p −1)(q −1)

        %Calculate the value of e    // e- ღია გასაღების მნიშვნელობის გამოთვლა

x=2; e=1;

while x > 1

          e=e+1;     x=gcd(Phi,e);

end

        %Calculate the value of d          // d- საიდუმლო გასაღების გამოთვლა

i=1; r=1;

while r > 0

              k=(Phi*i)+1;     r=rem(k,e);     i=i+1;

end

d=k/e;

clc;

// პარამეტრების გამოთვლილი  მნიშვნელობების დაბეჭდვა

disp(['The value of (N) is: ' num2str(Pk)]);

disp(['The public key (e) is: ' num2str(e)]);

disp(['The value of (Phi) is: ' num2str(Phi)]);
```

```
disp(['The private key (d)is: ' num2str(d)]);
```

ფუნქცია რომლითაც ხდება დაშიფვრა

```
function mc = crypt(M,N,e)            //M- ტექსტი, N- მოდულის ფუძე, e- ღია გასაღები
e=dec2bin(e);
k = 655;   c  = M;  cf = 1;
cf=mod(c*cf,N);
for i=k-1:-1:1
                c = mod(c*c,N);
                j=k-i+1;
        if e(j)==1
                        cf=mod(c*cf,N);
        end
end
mc=cf;
```

RSA ალგორითმის რეალიზება

```
clc;
disp('Implementation of RSA Algorithm');
clear all;   close all;
```

p და q დიდი მარტივი რიცხვების აღება შემთხვევითად.

```
range=3:3000;
m=isprime(range);
prime_mat=range(find(m));
p=randsample(prime_mat,1);
q=randsample(prime_mat,1);
```

```
[Pk,Phi,d,e] = intialize(p,q);                    //პარამეტრების ინიციალიზება


M = input('\nEnter the message: ','s');    // ტექსტის შეტანა
x=length(M);
c=0;
სიმბოლოების გადაყვანა შესაბამის ათობით რიცხვებში და შედეგის დაბეჭდვა
for j= 1:x
   for i=0:122
      if strcmp(M(j),char(i))
               c(j)=i;
      end
   end
end
disp('ASCII Code of the entered Message:');
disp(c);
დაშიფვრა
% % %Encryption
for j= 1:x
  cipher(j)= crypt(c(j),Pk,e);
  last_number=cipher(j);
  sms=c(j);
end
```

// ბეჭდვა დაშიფრული შიფროტექსტის და p და q.

disp('Cipher Text of the entered Message:'); disp(cipher);

disp('p='); disp (p);

disp('q='); disp (q);

// შიფრო ტექსტიდან ვიღებთ ბოლო ბაიტს (შემთხვევითი მიმდევრობა, რომელიც გამოიყენება ნაკადური შიფრის გამად) და ვინახავთ ფაილში.

bin_number=dec2bin(last_number); last_bait=bin_number(1:8);

disp('last bait:'); disp(last_bait);

bin_text=dec2bin(sms); text=bin_text(1:16);

disp('bin_text'); disp(text);

fileID = fopen(' example.txt','w'); // ფაილის გახსნა

fprintf(fileID,'%i ',text, last_bait); // ფაილში მონაცემების ჩაწერა 16 ბიტი შეტანილი ტექსტი ორობით რიცხვებში , მომდევნო 8 ბიტი დაშიფრული ტექსტის ბოლო ბაიტი (შემთხვევითი მიმდევრობა)

fclose(fileID);

დეშიფრაცია

% % %Decryption

for j= 1:x

message(j)= crypt(cipher(j),Pk,d);

end

disp('Decrypted ASCII of Message:'); disp(message); // ტექსტი ათობით რიცხვებში

disp(['Decrypted Message is: ' message]); // ტექსტი

დასკვნა

სამგისტრო ნაშრომის თემატიკა ნეირონული ქსელის საშუალებით ფსევდოშემთხვევითი რიცხვების გენერატორის შექმნაა. ჩვენი ამოცანა იყო აგვეგო კრიპროგრაფიულად საიმედო ფსევდოემთხვევითი მიმდევრობათა გენერატორი ნეირონული ქსელის საშუალებით. ამ მიზნით შევსწავლე RSA გენერატორი და matlab- ში ავაგე RSA გენერატორი, რომელიც წარმოადგენს RSA ალგორითმის მოდიფიცირებას და ავაგე ამ გენერატორის საშუალებით კრიპტოგრაფიულად საიმედო ფსევდოშემთხვევითი მიმდევრობები, რომლის საფუძველზეც შემდეგში უნდა მოხდეს ნეირონული ქსელის დასწავლა. შევისწავლე ნეირონული ქსელის აგების და დასწავლის მეთოდები, კერძოდ მრავალშრიანი ნეირონული ქსელი (MLP) . შემდეგ ავაგეთ MLP ქსელი და ჩავატარე ბევრი ცდა რომელთაგან უმრავლესობამ აჩვენა, რომ ნეიროული ქსელი გვაძლევს იგივე კრიპტომედეგ მიმდევრობას როგორსაც RSA გენერატორი. ჩვენი მიზანი იყო გვეჩვენებინა რომ მრავალშრიან ნეირონულ ქსელს შეუძლია გამოიმუშაოს ისეთივე კრიპტომედეგი მიმდევრობა როგორიც RSA-ამ, რაც ვაჩვენეთ კიდეც ჩვენს მიერ ჩატარებული ცდებით. ეს ნამუშევარი წარმოადგენს კვლევის ხასიათის ნაშრომს, ამიტომ ჩვენ თავდაპირველად ვმუშაობდით პატარა რიცხვებზე და პატარა მიმდევრობაზე, რადგან ჩვენ ვაჩვენეთ, რომ ნეირონული ქსელი შეიძლება იყოს კარგი გენერატორი ამის განზოგადება დიდ რიცხვებზეც შეიძლება რაც შესაძლოა ცალკე კვლევის ობიექტიც გახდეს.

გამოყენებული ლიტერატურა

1. Robshaw M.J., <<Security Estimates for 512-bit RSA>>, RSA Laboratories, June 29, 1995.
2. Williams H.C. A modification of the RSA public-key cryptosystem // IEEE Trans. Inform. Theory. 1980. V.26, No. 6. -.726-729.
3. Karam M. Z. Othman, Mohammed H. AL Jammas Implemetation of neural-cryptographic system using FPGA ournal of Engineering Science and Technology Vol. 6, No. 4 (2011) 411 - 428 .
4. http://en.wikipedia.org/wiki/Pseudorandom_number_generator
5. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction to neural networks
6. Neural Network Toolbox R2013a - Mark Hudson Beale, Mar tin T. Hag an, Howard B. Demuth
7. Developing pseudo random number generator based on neural networks and neurofuzzy systems. - Kayvan Tirdad - Ryerson University 1-1-2010