Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**Отчёт по лабораторной работе №16**

“Согласование криптографических ключей на основе технологий искусственных нейронных сетей”

**Выполнил:** студент 3 курса

4 группы специальности ПОИТ

Мозолевский Александр Дмитриевич

**Проверил:** преподаватель

Сазонова Дарья Владимировна

Минск 2023

**Разработать приложение, реализующее модель ТСР, – эмулятор процесса синхронизации весовых коэффициентов двух ИНС. Алгоритм случайного блуждания.**

Реализация приложения:

|  |
| --- |
| package main  import (  "fmt"  "math"  "math/rand"  )  type NeuralNetwork struct {  Weights [][]float64  }  func InitializeWeights(layers, neurons int) [][]float64 {  weights := make([][]float64, layers)  for i := range weights {  weights[i] = make([]float64, neurons)  for j := range weights[i] {  weights[i][j] = rand.Float64()  }  }  return weights  }  func CalculateDifference(nn1, nn2 NeuralNetwork) float64 {  diff := 0.0  for i := range nn1.Weights {  for j := range nn1.Weights[i] {  diff += math.Abs(nn1.Weights[i][j] - nn2.Weights[i][j])  }  }  return diff  }  func UpdateWeights(nn1, nn2 \*NeuralNetwork, diff float64) {  for i := range nn1.Weights {  for j := range nn1.Weights[i] {  delta := (rand.Float64() \* 2 \* diff) - diff  nn1.Weights[i][j] += delta  nn2.Weights[i][j] -= delta  }  }  }  func PrintWeights(nn NeuralNetwork) {  for i := range nn.Weights {  for j := range nn.Weights[i] {  fmt.Printf("%.4f ", nn.Weights[i][j])  }  fmt.Println()  }  fmt.Println()  }  func main() {  rand.Seed(42)  layers := 2  neurons := 3  iterations := 50  threshold := 0.01  steps := 0  nn1 := NeuralNetwork{Weights: InitializeWeights(layers, neurons)}  nn2 := NeuralNetwork{Weights: InitializeWeights(layers, neurons)}  for i := 0; i < iterations; i++ {  diff := CalculateDifference(nn1, nn2)  if diff < threshold {  fmt.Println("Convergence achieved. Stopping synchronization.")  steps = iterations  break  }  UpdateWeights(&nn1, &nn2, diff)  }  fmt.Println("Neural Network 1:")  PrintWeights(nn1)  fmt.Println("Neural Network 2:")  PrintWeights(nn2)  if steps == 0 {  steps = iterations  }  fmt.Println("Steps: ", steps)  } |

Результат выполнения:

