Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

Звіт

про виконання лабораторної роботи №3

з курсу

“Нейронні мережі”

Виконала:

Студентка групи ПМ-23

Мінзюк Наталія

Перевірив:

Мастикаш О.В.

Львів – 2021 р.

**Тема:** НАВЧАННЯ МЕРЕЖ. АПРОКСИМАЦІЯ ФУНКЦІЙ

**Мета:** Навчитися використовувати нейронні мережі для апроксимації наборів

точок, які репрезентують собою функції (у свою чергу ці функції можуть

описувати певний процес)

**Завдання**

Власноруч розробити програму для апроксимації функції вказаної у вашому варіанті на зручній для вас мові програмування.

Варіант №5



**Хід роботи**

const int M = 10;

const int P = 100;

const double lr = 0.001;

const double eps = 0.5;

int epochs = 1000;

vector<double> X(P);

vector<double> Y(P);

vector<double> error(P);

vector<double> center(M);

vector<double> delta(M);

vector<double> Weight(M);

inline double Func(double x) {

return x\*pow(sin(x),2);

}

//заповнення векторів для тренування

void Input(double floor, double ceil) {

for (int i = 0; i < P; ++i) {

X[i] = floor + 1.0 \* rand() / RAND\_MAX \* (ceil - floor);

Y[i] = Func(X[i]) ;

}

}

//початкова ініціалізація

void Init(vector<double>& vec, double floor, double ceil) {

for (int i = 0; i < vec.size(); ++i)

vec[i] = floor + 1.0 \* rand() / RAND\_MAX \* (ceil - floor);

}

//результат роботи мережі

double Res(double x) {

double y = 0.0;

for (int i = 0; i < M; ++i)

y += Weight[i] \* exp(-1.0 \* (x - center[i]) \* (x - center[i]) / (2 \* delta[i] \* delta[i]));

return y;

}

//обчислення помилки

double Error() {

double res = 0.0;

for (int i = 0; i < P; ++i) {

error[i] = Y[i]- Res(X[i]);

res += error[i] \* error[i];

}

return res / 2;

}

// навчання

void Train() {

for (int j = 0; j < M; j++) {

double sumC = 0.0, sumD = 0.0, sumW = 0.0;

for (int i = 0; i < P; ++i) {

double tmp = X[i] - center[j];

double err\_ex = error[i] \* exp(-1.0 \* pow(tmp, 2) / (2 \* pow(delta[j],2)));

sumW += err\_ex;

sumC += err\_ex \* tmp;

sumD += err\_ex \* pow(tmp, 2);

}

center[j] += (lr \* Weight[j] / (delta[j] \* delta[j]) \* sumC);

delta[j] += (lr \* Weight[j] / pow(delta[j], 3) \* sumD);

Weight[j] += (lr \* sumW);

}

}

int main(int argc, char\* argv[]) {

srand(time(0));

Init(Weight, -0.1, 0.1);

Init(center, -4.0, 4.0);

Init(delta, 0.1, 0.3);

Input(-4.0,4.0);

while (epochs-- > 0) {

if(Error() < eps)

break;

Train();

}

for (int x = -4; x < 5; ++x)

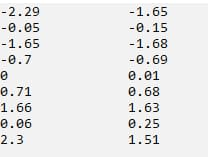
cout << ceilf(Func(x) \* 100) / 100<<"\t\t"<< ceilf(Res(x) \* 100) / 100 << endl;

return 0;

}

Результат роботи програми

еталон результат



Висновок

Навчитилася використовувати нейронні мережі для апроксимації наборів точок, які репрезентують собою функції.