	Класифікатор	Нейронна мережа
Undersample	12	8
SMOTE	18	16

Рис. 1. Результати оцінки помилок другого роду

На рисунку представлені абсолютні значення кількості пропущених шахрайських запитів, тоді як загальна кількість шахрайських запитів, що мали бути виявлені, становить 98.

Висновок. Розроблені моделі класифікації на основі штучно збалансованих наборів даних показали ефективність використання обидвох методів для тренування класифікаторів або класифікаційних нейронних мереж.

О. Тимошенко
Науковий керівник – д.т.н., проф. Л. В. Мороз

МЕТОД ШВИДКОГО ОБЧИСЛЕННЯ ІНВЕРСНОГО $\sqrt{x}$ ДЛЯ ЗНАЧЕНЬ ТИПУ FLOAT

Обчислення нормалізованих векторів для комп’ютерної графіки є досить складною задачею навіть для потужних сучасних комп’ютерів. Оскільки програма з 3D графікою використовує їх для визначення освітлення і відображення, мільйони цих обчислень повинні виконуватися за секунду. Нормалізація вектора це домноження кожної його координати на величину $1/\sqrt{x^2+y^2+z^2}$.

На відміну від додавання та піднесення до степеня розрахунок квадратного кореня та ділення – складні для процесора операції. Саме тому **актуальною є задача** розробки алгоритмів швидкого обчислення оберненого квадратного кореня для чисел з рухомою комою.

На початку 1990-х років співробітники компанії Silicon Graphics винайшли алгоритм InvSqrt1, на основі якого в 2018 році було розроблено алгоритм InvSqrt2, який описано в статті Improving the accuracy of the fast inverse square root algorithm (2018). Вищезгадані алгоритми наведено в першому рядочку Таблиці 2.

В ході роботи нами було розроблено алгоритми InvSqrt3 та InvSqrt4, які продемонстровано в другому рядочку Таблиці 2. В даних алгоритмах використовуються уточнені константи для обчислення початкового наближення та для першої ітерації Ньютона. В кожному із запропонованих алгоритмів нами було модифіковано другу ітерацію, що дало змогу зменшити похибку обчислень. При цьому алгоритм InvSqrt4 використовує метод Хаусхолдера другого порядку.

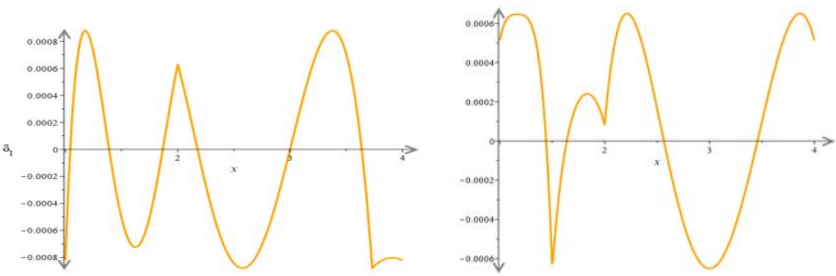
Алгоритм обчислення оберненого квадратного кореня приймає на вхід x типу float та повертає наближене до числа $1/\sqrt{x}$ значення. Максимальні відносні похибки алгоритмів наведені в Таблиці 1.

Таблиця 1

Результати порівняння алгоритмів

Результати роботи алгоритмів			
Алгоритм	Операції множення	Максимальна відносна похибка	Правильні біти результату
Sqrt1	7	4.86?10^{-6}	17.65
Sqrt2	7	7.37?10^{-7}	20.37
Sqrt3	8	4.09?10^{-7}	21.22
Sqrt4	9	8.96?10^{-8}	23.41

Графіки відносних похибок алгоритмів Sqrt2 та Sqrt3 після першої ітерації наведено на Рисунку 1(а, б) відповідно.



а) б)

Рис. 1. Графіки відносних похибок алгоритмів Sqrt2 -а та Sqrt3 – б, де значення аргументу x ∈ [1,4].

Таблиця 2

Реалізація алгоритмів мовою С++

Відомі алгоритми	float **InvSqrt1**(float x){
	float halfnumber = 0.5f*x;
	int i = *(int*)&x;
	i = 0x5f3759df – (i >> 1);
	x = *(float*)&i;
	x = x*(1.5f – halfnumber*x*x);
	x = x*(1.5f – halfnumber*x*x);
	return x;}
		float **InvSqrt2**(float x){
		float halfnumber = 0.5f*x;
		int i = *(int*)&x;
		i = 0x5f376908 – (i >> 1);
		x = *(float*)&i;
		x = x*(1.5008789f – halfnumber*x*x);
		x = x*(1.5000006f – halfnumber*x*x);
		return x;}

Розроблені алгоритми	float **InvSqrt3**(float x){
	int i = *(int*)&x;
	i = 0x5f5ffff8 – (i >> 1);
	float y = *(float*)&i;
	y = 0.248884737f*y*(4.778488636f – x*y*y);
	float c = x*y;
	c = fmaf(y, -c, 1.00000065f);
	y = fmaf(y, 0.5f*c, y);
	return y;}
		float **InvSqrt4**(float x){
		int i = *(int*)&x;
		i = 0x5f5ffff8 – (i >> 1);
		float y = *(float*)&i;
		y = 0.248884737f*y*(4.778488636f – x*y*y);
		float c = x*y;
		float r = fmaf(y, -c, 1.0f);
		c = fmaf(0.375f, r, 0.5f);
		r = r*c;
		y = fmaf(y, r, y);
		return y;}

Висновок: Вище розглянуті модифікації алгоритмів InvSqrt1 та InvSqrt2 дають можливість на порядок зменшити похибку обчислень при відносно малій кількості додаткових операцій множення, а отже при незначній втраті швидкодії. Алгоритми InvSqrt3 та InvSqrt4 працюють вдвічі швидше ніж стандартна функція pow і при апаратній реалізації функції fmaf спостерігається значне підвищення швидкодії.

О. Ворончук

Науковий керівник – д.т.н., професор Г. І. Клим

СПЕЦІАЛІЗОВАНА СИСТЕМА ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ТІЛА ЛЮДИНИ У ВІРТУАЛЬНІЙ РЕАЛЬНОСТІ

Перші прототипи шоломів віртуальної реальності появилися ще в 60-х роках минулого сторіччя, але назвати їх повноцінним комерційним рішенням досить важко. З розвитком технологій реалізація таких пристроїв стає все легшою, а самі шоломи користуються попитом та набувають масовості. Протягом останнього десятиріччя у віртуальній