

не сверхточные HC. Все легко реализуемых встра. В том же классе, реализуемые веса отсутствуют. Чтобы устранить этот недостаток много был найден класс, позволяющий реализовать СНС произвольной архитектуры и применять их к различным задачам. Качать класс можно по ссылке [2].

Сам класс был написан так, чтобы тому, кто им пользуется была максимальная видна структура сети. Все очень обильно прокомментировано, на названии переменных не экономил. Скорость симуляции сети неплоха и составляет доли секунды. Скорость обучения пока не велика (>10 ч). Так же существуют реализации СНС на C++, могут найти ее по ссылке [3] <http://cbscat.sourceforge.net/> и [4].

Библиографический список:

1. YamleSun, J. S. Denker, S. Solla, R. E. Howard and L. D. Jackel: Optimal Brain Damage. in Touretzky, David (Eds). Advances in Neural Information Processing Systems 2 (NIPS*89). Morgan Kaufman, Denver, CO, 1990
2. Y. LeCun and Y. Bengio: Convolutional Networks for Images, Speech, and Time-Series. in Arbib, M. A. (Eds). The Handbook of Brain Theory and Neural Networks, MIT Press, 1995
3. Y. LeCun, L. Bottou, G. Orr and K. Muller: Efficient BackProp. in Orr, G. and Muller K. (Eds). Neural Networks: Tricks of the trade. Springer, 1998
4. RamziouMare'Aurelio, Christopher Roulhney, Sumit Chopra and YamleSun: Efficient Learning of Sparse Representations with an Energy-Based Model. in J. Platt et al. (Eds). Advances in Neural Information Processing Systems (NIPS 2006), MIT Press, 2006
5. Способы распознавания изображений с помощью нейросетей [http://www.habr.ru/post/74326/](http://habrahabr.ru/post/74326/)
6. <http://www.habr.ru/post/74326/>
7. <http://cbscat.sourceforge.net/>
8. <http://www.coderproject.com/KB/Library/NeuralNetRecognition.aspx>

АНАЛИЗ И РАЗРАБОТКА СПЕЦИАЛИЗИРОВАННОЙ МАТЕМАТИЧЕСКОЙ БИБЛИОТЕКИ ДЛЯ АРХИТЕКТУРЫ ARM.

Мартычев Т.М.
Начинный руководитель: д.т.н., проф. Маринин Г. М.
Кафедра «Компьютерных систем управления» ФГБОУ ВПО МГТУ «СТАНКИН»

На сегодняшний день стали широко использовать процессоры ARM ввиду их низкого энергопотребления. Устройства на базе архитектуры ARM, могут быть значительно малых размеров, по сравнению с устройствами на базе архитектуры x86 либо x64. Следует заметить, что в архитектуре ARM быстроедействие увеличивается за счет упрощения отдельных операций процессора ARM процессоры, также можно увидеть в некоторых ПЛИК, например ПЛИК фирмы ОВЕН. Вычислительным ресурсом для ПЛИК63/ПЛИК73 фирмы ОВЕН является процессор архитектуры ARM7. В потоне за снижением энергопотребления и увеличением быстродействия, производители процессоров опускают такие возможности процессора как вычисления с использованием плавающей запятой, что в свою очередь создает некоторые проблемы программисту, при написании программ для процессоров с ARM архитектурой. Кроме того у разработчика теряется возможность работать с рациональными числами. Возникает потребность в создании кроссплатформенной библиотеки на языке Си++. Данная библиотека выполняет вычисления с точностью до 7 знаков после запятой. За основу данной библиотеки берется целые числа, представленные в виде числа с фиксированной запятой. В статье продемонстрированы ARM архитектуры, в которых отсутствует FPU

реализация проблемы, их реализация в сравнении со стандартной библиотекой math.h. **Анализ ARM процессоров.**

Версии архитектур созданные до разработки семейства ядер Cortex.

ВFP: Ресурсы Архитектуры в семействе ядер Cortex.

Таблица 1. Названия FPU/VFP в архитектурах из семейства ядер Cortex (ARM)

| | | | | | |
|-------------|---------|-------|-------|-------|-----|
| Версия | ARMv7-A | ARMv7 | ARMv7 | ARMv7 | ARM |
| Архитектура | | E | | | M |
| FPU | + | + | + | - | - |
| VFP | + | + | + | - | - |

IEEE 754 - стандарт двойной арифметики с плавающей точкой.

Данный стандарт разработан ассоциацией IEEE (Institute of Electrical and Electronics Engineers) и используется для представления действительных чисел (чисел с плавающей точкой) в двоичном коде. Наиболее используемый стандарт для вычислений с плавающей точкой, используется многими микропроцессорами и логическими устройствами, а также программными средствами.

Стандарт определяет 4 формата представления числа с плавающей точкой.

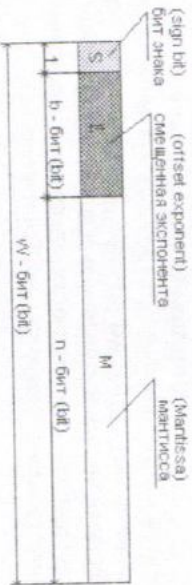


Рис. 1. Представление числа в формате IEEE 754

- Повешение к рисунку 1:
- S - бит знака, если S=0 - положительное число; S=1 - отрицательное число
- E - смещенная экспонента двоичного числа.
- $e \times 2 = E - (2(b - 1) - 1)$ - экспонента двоичного нормализованного числа с плавающей точкой.
- $(2(b - 1) - 1)$ - заданное смещение экспоненты.
- M - остаток мантиссы двоичного нормализованного числа с плавающей точкой.
- Максимальное значение для однарной точности приблизительно равно $3.40282347 \times e+38$, минимальное $-3.40282347 \times e+38$ [1]

Число с фиксированной запятой. Формат представления вещественного числа в виде целого числа. При этом само число x и его целочисленное представление x' связаны формулой: $x = x' \cdot z$, где z -цена младшего разряда. Реализация операций для числе с фиксированной запятой будет выглядеть вот так:

$$(x * y)' = [x' * y' * z] = \left[\frac{x' * y'}{z} \right] \quad (1) \quad \left(\frac{x}{z} \right)' = \left(\frac{x'}{z} \right) = \frac{x' * z}{z^2} \quad (2)$$

Достоинства фиксированной запятой.

Операции [] - означает округление до целого [2]. Числа с фиксированной точкой используются в случае, если диапазон значений и точность заранее определена. Для задач, решаемых на производстве заранее ясно эти два параметра: точность и в каких пределах задача будет решаться. Приведем пример с обычным токарным станком на производстве. Масштабы работы могут достигать, грубо говоря, до ~1-2 м, а точности в 7 знаков после запятой будет достаточно. Причины использования фиксированной точки и её достоинства:

1) Полный контроль за поведением кода. Фиксированная точка исключает появление «неожиданностей», связанных с особенностями реализации плавающей запятой на используемой платформе.

2) Автоматическая «фильтрация» пренебрежимо малых значений. В плавающей запятой ошибки вычислений могут накапливаться, в фиксированной точке этого не происходит (за счет отбрасывания малых значений) или процесс накопления ошибок можно контролировать алгоритмически.

3) Алгоритмически контролируемый диапазон значений переменных. Плавающая запятая дает больше свободы в вычислениях, но результат может выскочить за пределы допустимых, что приводит к необходимости его контролировать отдельно. В фиксированной точке эта проблема решается автоматически на этапе разработки и отладки алгоритма [3].

Наряду с плюсами есть и некоторые минусы, которые заключаются в перенесении базовых математических функций, например, тригонометрических, логарифмических и т.д., что и является темой статьи.

В сравнении с плавающей запятой хочется отметить, что стандарт IEEE 754 дает возможность работать с большим диапазоном значений, но отсутствие возможности работать с ним на некоторых процессорах, проблемы, встречающиеся при программировании с использованием чисел с плавающей запятой (например простые условия), делают фиксированную точку более заманчивой.

Создание базовых функций.

Для того, чтобы создать базовые функции для математической библиотеки нужно рассмотреть возможные варианты алгоритмов функций. Для примера мы возьмем тригонометрию. В статье продемонстрировано решение для функции тангенса.

Исследования показали, что решить задачу программно можно 3-мя методами.

1) С помощью разложения тангенса в ряд Тейлора

$$\operatorname{tg}(x) = x + \frac{x^3}{3!} + \frac{2x^5}{15!} + \dots$$

$$= \sum_{n=1}^{\infty} \frac{(-1)^{n-1} x^{2n-1}}{(2n-1)!} \quad x \in (-\frac{\pi}{2}, \frac{\pi}{2}) \quad B_{2n-1} - \text{числа Бернулли}, |x| < \frac{\pi}{2} \quad (3) \quad [4]$$

2) С помощью разложения в целую дробь

$$\left[a_0, a_1, a_2, a_3, \dots \right] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}} \quad (+)$$

где $a_1, a_2, a_3, \dots, a_n$ – натуральные числа, a_0 – натуральное число или ноль, называется целой дробью [4].

Тангенс в результате разложения выглядит так:

$$\operatorname{tg} x = \frac{x}{1 - \frac{x^2}{3!} - \frac{x^4}{5!} - \frac{x^6}{7!} - \dots} \quad \dots, k = 1, 2, \dots, |x| < \frac{\pi}{2} \quad (5) \quad [5]$$

3) С помощью разложения в ряд Тейлора, мы значительно проиhrываем во времени, по сравнению с библиотечной math.h.

Результат вычисления.

При выполнении разложения в ряд Тейлора, мы значительно проиhrываем во времени, по сравнению с библиотечной math.h.

Пример для $\pi/2$.

$$\frac{\pi}{2} \approx 1.57080000. \text{ Что в свою очередь является } 15700000 \text{ итераций при точности в } 0,0000001.$$

При выполнении разложения в целую дробь мы получаем результат, несомненно отличающийся от результата math.h.

Результат вычисления.

При выполнении разложения в целую дробь мы получаем результат, несомненно отличающийся от результата math.h.

Результат вычисления.

При выполнении разложения в целую дробь мы получаем результат, несомненно отличающийся от результата math.h.

Количество времени/Итераций

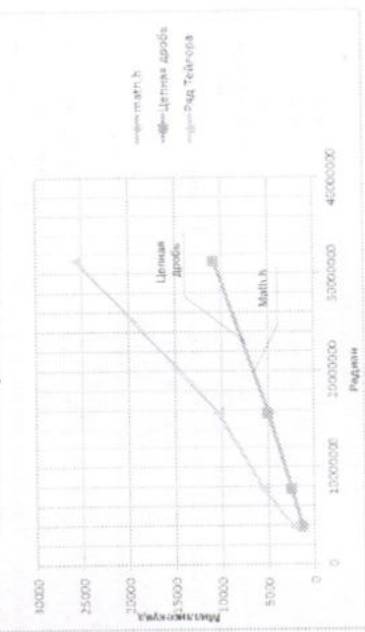


Рис. 2. График

Вывод: Решения базовых математических функций на основе целых чисел, дают нам возможность работать в прикладных областях. В некоторых случаях происходит улучшение в вопросах скорости выполнения, а результат получается с лучшей точностью.

Библиографический список:

1. IEEE Standard for Binary Floating-Point Arithmetic. Copyright 1985 by The Institute of Electrical and Electronics Engineers, Inc 345 East 47th Street, New York, NY 10017, USA
2. http://ru.wikipedia.org/wiki/Число_с_фиксированной_запятой (11.03.2013)
3. <http://habrahabr.ru/post/131171/> (11.03.2013)
4. Целые дроби Н.М. Бескин
5. Вестник Ставропольского государственного университета. Статья «О разложении функции $\sin(x)$ в вставившиеся целые дроби». 70/2010

ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ ГРАФИЧЕСКИХ КОМПОНЕНТОВ С ЯДРОМ СИСТЕМЫ ЧПУ

Никашевичи П.А. – аспирант 2го года обучения.
 Научный руководитель: д.т.н., проф. Маршинов Г.М.
 Кафедра «Компьютерные Системы Управления» ФГБОУ ВПО МГТУ «СТАНКИ».

Современные промышленные технологии требуют высокого уровня автоматизации и гибкости систем управления, что сегодня вполне успешно может быть реализовано с помощью принципов открытости и модульности. [1,2] Модульная организация аппаратно-программного обеспечения системы управления формирует такие ее характеристики, как масштабируемость, конфигурируемость и многофункциональность. Система ЧПУ построена по типу двухкомпьютерной архитектуры, и относится к классу PCNC-2. Система делится на терминальный компьютер, и компьютер реального времени, в котором функционирует ядро системы ЧПУ. Одним из главных свойств разрабатываемой системы управления является ее открытая модульная архитектура. Подобная архитектура различных типов технологического оборудования и различных технологических задач, и во-вторых, расширять ее функциональные возможности за счет простой интеграции новых программно-аппаратных решений. [3]