Программное обеспечение процессора NM6403

Описание реализации программы

Оценка производительности эмуляции нейронной сети

Автор: Михаил Забалуев

Оглавление

Введение

В документе описывается программная реализация работы нейронной сети на процессоре NM6403 [1]. Программа носит характер примера. Предназначение примера – указание способов эффективной реализации нейросетевых вычислений на данном процессоре и оценка производительности процессора на задачах данного класса. Нейронная сеть [4], над которой производятся вычисления по схеме прямого распространения, обладает следующими особенностями:

- сеть имеет 12 входов, первый промежуточный слой из некоторого большого (порядка нескольких сотен) количества нейронов и выходной слой из 12 нейронов;
- входные и выходные значения нейронов представлены 64-разрядными целыми числами со знаком;
- весовые коэффициенты нейронов представлены 16-разрядными целыми числами со знаком, упакованными по четыре в 64-разрядных словах;
- выходная характеристика нейрона (зависимость выходного значения от взвешенной суммы входов) определяется целочисленной функцией с нелинейным (сигмоидальным) вещественным прототипом, область значений которой лежит в диапазоне от -32767 до 32767. При значениях аргумента меньших -32768 или больших 32767 функция принимает крайние значения, -32767 или 32767 соответственно.

Оценивается быстродействие работы программы, вычисляющей выходные значения сети. Основные процедуры написаны на языке ассемблера и оптимизированы вручную для достижения наибольшей производительности процессора.

Многослойная нейронная сеть типа перцептрона представляет из себя несколько слоев однотипных *нейронов* (Рис. 1). Нейрон можно представить в виде численной (скалярной) функции от вектора численных входных значений. Выходное значение нейрона вычисляется следующим образом: входные значения умножаются на соответствующие входам *весовые коэффициенты*, полученные значения суммируются. По сумме вычисляется выходное значение по некоторой функции, называемой *пороговой функцией* или *функцией активации*. Как правило, функция выбирается монотонно неубывающей и ограниченной в некотором диапазоне. При этом она сходится к крайним значениям при удалении значения аргумента от некоторой центральной точки. В целях лучшей обучаемости градиентными методами функцию (или, при вычислении дискретных значений, ее вещественный прообраз) делают гладкой и нелинейной. Нейроны одного *слоя*, например, персептрона, получают один и тот же вектор входов. Изменение выходных характеристик сети достигается за счет настройки весовых коэффициентов, называемой обучением.

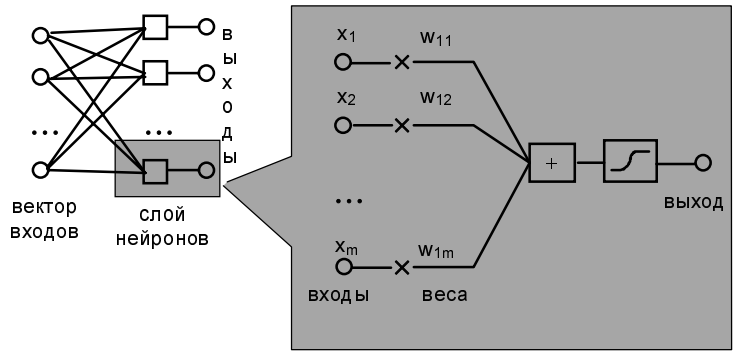

Рис. 1. Схема нейронного слоя.

Вычисление взвешенных сумм

Матрично-векторное устройство процессора NM6403 [1] обладает способностью производить операции взвешенного суммирования, характерные для матричных вычислений, над данными с изменяемой разрядностью. В нашем примере в матрицу множителей векторного устройства загружаются на всю "ширину" 64-разрядные входные значения, представляющие собой сегменты входного вектора по четыре значения:

```
rep 4 wfifo = [ ar4++ ], ftw;
...
wtw;
```

На эти значения умножаются с учетом знаков соответствующие сегменты матрицы весов из четырех 16-разрядных значений, упакованных в 64-разрядные слова. Четыре произведения суммируются в 64-разрядное значение, которое добавляется к уже вычисленной частичной сумме (Рис. 2). Эти четыре умножения и сложение пяти значений происходят в течение одного шага векторной операции, которая, будучи многошаговой, вычисляет несколько (до 32) таких частичных сумм для нескольких нейронов благодаря тому, что вектор входных значений общий у всего слоя:

```
rep 32 data = [ ar0 += gr0 ] with vsum ,data, 0;
```

(для первых четырех входных значений),

```
rep 32 data = [ ar0 += gr0 ] with vsum ,data, afifo;
```

(для последующих входных значений, прибавляются уже вычисленные суммы из очереди результатов).

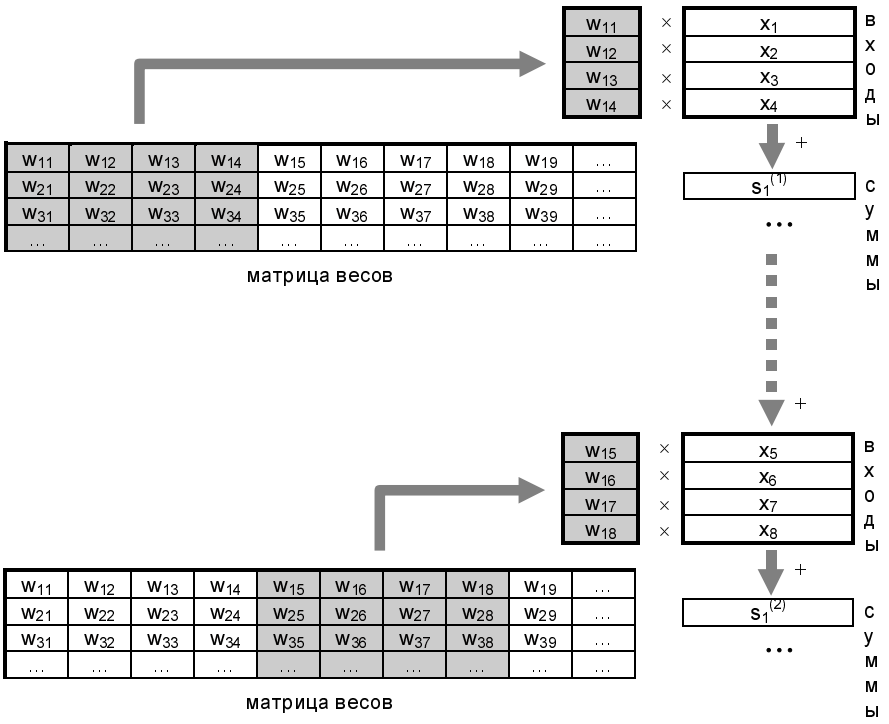

Рис. 2. Схема вычисления взвешенных сумм нейронного слоя.

Чтобы избежать потерь на сохранение и чтение частичных сумм из памяти, суммы всех обрабатываемых за одну такую операцию нейронов используются как слагаемые в следующей итерации и не покидают векторного устройства.

Таким образом, поочередно сменяя четверки входных значений в матрице множителей, вычисляются полные взвешенные суммы для, положим для определенности, 32 нейронов. Загрузка матрицы – дорогостоящая операция, но в большинстве случаев удается загружать новые коэффициенты на фоне основных вычислений. Для этого в процессоре существует “теневая” матрица, которая за один такт может быть скопирована в рабочую.

В следующий проход по входам вычисляются суммы еще 32 нейронов, и т.д., пока не окажутся вычисленными суммы нейронов всего слоя.

Вычисление пороговой функции

В силу того, что пороговая функция при больших по модулю значениях аргумента практически превращается в постоянную, мы можем вычислять ее значения в некотором диапазоне по таблице, а на значения аргумента, выходящие за границы диапазона, распространить значения функции в соответствующих крайних точках диапазона. Для этого удобно воспользоваться аппаратной *функцией насыщения* векторного устройства:

```
rep 32 with activate afifo + 0;
```

Управляющие регистры настраиваются таким образом, что при применении этой функции к взвешенным суммам значения в диапазоне от -32768 до 32767 остаются без изменения, значения, меньшие -32768 или большие 32767, заменяются на -32768 и 32767 соответственно. Значения функции для каждого нейрона затем вычисляются на скалярном процессоре по таблице для 65536 возможных значений аргумента. В нашем примере числа в таблице представляют собой преобразованные к целому значения сигмоидальной функции:

$$f = 32767 \cdot \frac{1 - e^t}{1 + e^t}, t = -\frac{x}{2048}$$

, где x пробегает целые значения от -32768 до 32767.

Вычисление значений нейронной сети производят три процедуры, написанные на языке ассемблера [2, 3]. Они вызываются из основной C++-программы. Ниже перечислены прототипы вызовов для языка C++ и описание работы этих процедур.

```
void SumLayer ( size_t nInput, size_t nNeurons,
                const long *input,
                const unsigned long *weights,
                long *output );
```

Процедура вычисляет вектор взвешенных сумм слоя нейронов произвольного размера. К вычисленным суммам применяется функция насыщения, приводящая их к диапазону от -32768 до 32767. Количество нейронов в слое задается параметром `nNeurons`, количество входов – параметром `nInput`. Указатель `input` ссылается на массив 64-разрядных входных значений, `weights` указывает на начало матрицы 16-разрядных весов, упакованных по четыре в 64-разрядные слова, `output` указывает на массив 64-разрядных взвешенных сумм, вычисляемых в ходе выполнения процедуры. В силу многошагового характера векторных операций на размеры массивов в памяти налагаются требования кратности, достигаемой при необходимости дополнением нулевыми элементами. Размер массива входов должен быть не меньше `nInput` и кратен четырем 64-разрядным словам. Каждая строка из подряд идущих 16-разрядных значений матрицы весов (такая строка соответствует весам одного нейрона) занимает целое число 64-разрядных слов. Количество этих строк и количество элементов выходного массива должно быть не меньше `nNeurons` и кратно 32.

```
void SumLayer12 ( size_t nInput,
                  const long *input,
                  const unsigned long *weights,
                  long *output );
```

Процедура аналогична `SumLayer` за тем исключением, что количество нейронов в слое задано равным 12, соответственно опущен параметр `nNeurons`. В такой специализированной для конкретной сети версии процедуры отсутствует внешний цикл по группам из 32 нейронов, и операции взвешенного суммирования работают по 12 шагов. Соответственно, число выходных значений и строк матрицы весов равно 12.

```
void ApplyNeuroFunc ( size_t nInput,
                      const long *input,
                      long *output );
```

Процедура вычисляет значение пороговой функции для вектора частичных сумм слоя нейронов. Значения частичных сумм, как было

сказано выше, приведены к диапазону от -2^{15} до $2^{15}-1$. Значение функции определяется по таблице из 65536 значений, соответствующих возможным значениям аргумента.

Ниже приведено тело основной процедуры тестовой программы и некоторые объявления относящихся к ней данных.

```
    // number of neurons in the hidden layer
static const size_t nHidden = 1024;


extern "C" {
    long input[];
    long hidden[];
    long output[];
    unsigned long weights1[];
    unsigned long weights2[];
}


int main ()
{

        // initialize data

    Init();

        // get the starting clock value

    clock_t t1 = clock();

        // evaluate the sums of the first layer

    SumLayer( 12, nHidden, input, weights1, hidden );

        // evaluate the outputs of the first layer

    ApplyNeuroFunc( nHidden, hidden, hidden );

        // evaluate the sums of the second layer

    SumLayer12( nHidden, hidden, weights2, output );

        // evaluate the outputs of the second layer

    ApplyNeuroFunc( 12, output, output );

        // get the final clock value

    clock_t t2 = clock();

        // return elapsed processor clocks

    return t2 - t1;
}
```

Сеть состоит из двух слоев, первый, т.н. скрытый, имеет 12 входов, количество нейронов этого слоя определяется константой `nHidden`.

Выходы этих нейронов поступают на вход второго слоя из 12 нейронов. Время, затраченное на вычисление выходных значений сети, оценивается с помощью вызовов `clock()`, возвращающих значения с дискретностью тактов процессора.

Основная процедура программы описана в файле main.cpp, тела вычислительных процедур находятся в файле layers.asm, их прототипы на C++ – в заголовочном файле layers.h. В файле nfunctab.asm описана таблица значений пороговой функции нейрона. Файл data.asm содержит описатели массивов, используемых при работе программы.

По значению, возвращаемому тестовой программой, делается оценка времени, затрачиваемого процессором на выполнение вычислительных процедур. В таблице даны результаты выполнения программы при двух различных значениях константы `nHidden` (зависимость времени выполнения от числа нейронов слоя имеет линейный характер). Программа запускалась на процессоре NM6403 платы МЦ4.01 с тактовой частотой 40 МГц.

Значение nHidden	Число эквивалентных операций умножения элементов	Время выполнения в тактах процессора	Время выполнения в миллисекундах
512	12288	13150	0,33 мсек
1024	24576	25800	0,65 мсек

1. НТЦ Модуль. "Процессор NeuroMatrix® NM6403. Введение в архитектуру". http://www.module.ru/files/archover.pdf
2. НТЦ Модуль. "ПО процессора NeuroMatrix® NM6403. Справочное руководство".
3. НТЦ Модуль. "ПО процессора NeuroMatrix® NM6403. Описание языка ассемблера (предварительная версия)".
4. Fausett, L. Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Englewood Cliffs, NJ: Prentice Hall, 1994, ISBN 0-13-334186-0.

АКЦИОНЕРНОЕ ОБЩЕСТВО
НАУЧНО-ТЕХНИЧЕСКИЙ ЦЕНТР

Научно-технический центр Модуль
АЯ 166, Москва, 125190, Россия
Тел: +7 (095) 152-9335
Факс: +7 (095) 152-4661
E-Mail: postmast@module.ru
WWW: http://www.module.ru

Напечатано в России. Дата издания: 08.04.99