УДК 004.8.032.26

АДАПТАЦИЯ ТОПОЛОГИЙ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПРИМЕНЕНИЯ CUDA-ТЕХНОЛОГИИ

О.С. ХИЛЬКО, С.П. КУНДАС, В.И. КОВАЛЕНКО

Международный государственный экологический университет им. А.Д. Сахарова
ул. Долгобродская, 23, Минск, 220009, Беларусь

Поступила в редакцию 11 сентября 2012

В работе рассматривается применение CUDA-технологии в программной реализации прямого и обратного проходов искусственной нейронной сети (ИНС) на основе алгоритма обратного распространения ошибки. Показана правомерность добавления в слои ИНС «мнимых» нейронов, позволяющих адаптировать топологию нейронной сети для использования CUDA-технологии. При этом доказано, что введение «мнимых» нейронов не влияет на ход вычислений при выполнении алгоритма обратного распространения ошибки.

Ключевые слова: параллельные вычисления, искусственные нейронные сети, CUDA-технология, «мнимый» нейрон, вычисления на GPU.

Введение

В настоящее время *CUDA*-технология (*Compute Unified Device Architecture*), используемая для выполнения расчетов на графических процессорах (*GPU*) фирмы *Nvidia*, применяется в искусственных нейронных сетях (ИНС) для решения широкого спектра задач из разных предметных областей [1], в частности, при реализации алгоритма обратного распространения ошибки. К примеру, в разработанном нами программном комплексе *SPS* (*Simulation Processes in Soil*) многослойный персептрон на основе алгоритма обратного распространения применен для прогнозирования миграции загрязняющих веществ в почве и на ее поверхности [2]. *CUDA*-технология использована в нем для ускорения обучения ИНС и ее функционирования в режиме расчета (прямого прохода).

В вычислительной архитектуре *CUDA* имеется шесть видов памяти: регистровая, локальная, глобальная, разделяемая, константная, текстурная. Каждый из этих типов памяти имеет определенное назначение, которое обуславливается техническими параметрами памяти (скорость работы, уровень доступа на чтение и запись) [3]. При решении прикладных задач тип (или сочетание типов) памяти выбирается в каждом конкретном случае и зависит от множества факторов: размерности обрабатываемых данных, их внутренней структуры и типа, необходимости проведения промежуточных вычислений и др.

Для повышения производительности и минимизации обращений к глобальной памяти при работе с массивами данных большой размерности, а также для хранения локальных переменных внутренних методов используется «быстрая» разделяемая память. Адресация разделяемой памяти между нитями потока одинакова в пределах одного блока, что может быть использовано для обмена данными между потоками в пределах одного блока.

Следует подчеркнуть важность выбора оптимального размера блоков разделяемой памяти, что связано с аппаратными ограничениями видеокарты (в видеокартах *Nvidia GeForce* серий 8*x*/9*x* максимальный размер разделяемой памяти для блока составляет 16384 байта, количество потоков в блоке не может превышать 512). Соответственно, при работе с матрицами и векторами из элементов с плавающей точкой одинарной точности (*float*) целесообразно использовать двумерные квадратные блоки не выше размерности 16×16 [1].

Искусственные нейронные сети требуют проведения однотипных операций над большими объемами данных. Поэтому использование *CUDA*-технологии для реализации вычислительных алгоритмов ИНС способно значительно ускорить их выполнение.

Однако применение *CUDA*-технологии требует приведения количества нейронов в слоях к размерностям 2n, где с учетом применения двумерных квадратных блоков $n = \{1,\ldots,5\}$ [1,4]. Таким образом, для каждого слоя ИНС матрица весовых коэффициентов нейронов, векторы входных, выходных параметров и порогов должны быть выровнены.

Авторами статьи предложено при проведении вычислений ввести в слои ИНС «мнимые» нейроны, которые позволяют адаптировать топологию искусственной нейронной сети для использования *CUDA*-технологии. Добавление «мнимых» нейронов предполагает заполнение нулями недостающих параметров и установление желаемых значений для «мнимых» нейронов последнего выходного слоя равными значению функции активации от нуля $F_{act}(0)$.

Применение такого решения требует проведения точного математического анализа, так как в ИНС используются нелинейные функции активации. Также необходимо доказать, что добавление «мнимых» нейронов не влияет на результаты расчетов в ходе проведения вычислений алгоритма обратного распространения ошибки.

Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки [5,6] является эффективным средством обучения нейронных сетей. В частности, для прогнозирования миграции загрязняющих веществ в почве и на ее поверхности указанный алгоритм представляет собой следующую последовательность шагов [5–7]:

Шаг 1. Задается скорость обучения r ($0 < r < 1$) момент η ($0 < \eta < 1$), желаемая ошибка нейронной сети E_m и максимальное количество итераций обучения t_{max}.

Шаг 2. Случайным образом инициализируются весовые коэффициенты и пороговые значения нейронной сети в интервале [$(-0.05;0)$ U $(0;0.05)$].

Шаг 3. Последовательно подаются образы из обучающей выборки на вход нейронной сети. При этом для каждого входного образа выполняются следующие действия.

1. Производится расчет прямого распространения входного образа по нейронной сети, для чего вычисляется выходная активность всех нейронных элементов сети:

$$y_j = F_{act}(\sum_i w_{ij} y_i - T_j) = F_{act}(\sum_i w_{ji}{}^T y_i - T_j) = F_{act}(S_j), \quad (1)$$

где индекс j характеризует нейроны следующего слоя по отношению к слою i.

Для сигмоидальной функции активации выходная активность вычисляется по формуле (2), для гиперболического тангенса и биполярной сигмоидальной функции – по формулам (3) и (4) соответственно:

$$F_{act}(S_j) = \frac{1}{1+e^{-\alpha S_j}}, \quad (2)$$

$$F_{act}(S_j) = \beta \frac{e^{\alpha S_j} - e^{-\alpha S_j}}{e^{\alpha S_j} + e^{-\alpha S_j}}, \quad (3)$$

$$F_{act}(S_j) = \frac{2}{1+e^{-\alpha S_j}} - 1. \quad (4)$$

2. Производится расчет обратного распространения сигнала, в результате которого вычисляется ошибка сети и определяется локальный градиент δ_j нейронных элементов всех слоев.

Вычисляется суммарная среднеквадратичная ошибка нейронной сети (5), либо средняя относительная ошибка (6):

$$E = \frac{1}{2}\sum_s \sum_j (d_j{}^s - y_j{}^s)^2, \quad (5)$$

$$\sigma = \frac{1}{SN_{out}} \sum_s \sum_j \left(\frac{\left| d_j^{\,s} - y_j^{\,s} \right|}{\left| d_j^{\,s} \right|} \right) \cdot 100\%, \tag{6}$$

где S – размерность обучающей выборки, N_{out} – количество нейронов в выходном слое, $j = \{1, \ldots N_{out}\}$, $s = \{1, \ldots S\}$, d_j – желаемое значение выходного сигнала j-го нейрона последнего слоя.

Если $E > E_m$ или $\sigma > E_m$, то алгоритм выполняется далее. В противном случае алгоритм обратного распространения ошибки заканчивается.

Локальный градиент для выходного и скрытого слоев рассчитывается по формулам (7) и (8) соответственно:

$$\delta_j = (y_j - d_j) F'_{act}(S_j), \tag{7}$$

$$\delta_j = F'_{act}(S_j) \sum_k w_{jk} \delta_k, \tag{8}$$

В последнем выражении индекс k характеризует нейронные элементы следующего слоя по отношению к слою j.

Так как $y_j = F_{act}(S_j)$ то для различных функций активации справедливы следующие соотношения. Для сигмоидальной функции формулы (7) и (8) можно переписать в виде:

$$\delta_j = (y_j - d_j) F'_{act}(S_j) = \alpha (y_j - d_j) y_j (1 - y_j),$$

$$\delta_j = F'_{act}(S_j) \sum_k w_{jk} \delta_k = \alpha y_j (1 - y_j) \sum_k w_{jk} \delta_k;$$

для гиперболического тангенса как:

$$\delta_j = (y_j - d_j) F'_{act}(S_j) = \frac{\alpha}{\beta} (y_j - d_j)(\beta + y_j)(\beta - y_j),$$

$$\delta_j = F'_{act}(S_j) \sum_k w_{jk} \delta_k = \frac{\alpha}{\beta} (\beta + y_j)(\beta - y_j) \sum_k w_{jk} \delta_k;$$

для биполярной сигмоидальной функции активации в виде:

$$\delta_j = (y_j - d_j) F'_{act}(S_j) = \frac{\alpha}{2} (y_j - d_j)(1 + y_j)(1 - y_j),$$

$$\delta_j = F'_{act}(S_j) \sum_k w_{jk} \delta_k = \frac{\alpha}{2} (1 + y_j)(1 - y_j) \sum_k w_{jk} \delta_k.$$

3. Для каждого слоя нейронной сети находятся корректирующие значения весовых коэффициентов $\Delta w_{ij}(t)$ и порогов $\Delta T_j(t)$ на итерации t с учетом их значений на предыдущей итерации t–1:

$$\Delta w_{ij}(t) = r(\eta \Delta w_{ij}(t-1) + (1 - \eta) \delta_j y_i), \tag{9}$$

$$\Delta T_j(t) = r(\eta \Delta T_j(t-1) + (1 - \eta) \delta_j), \tag{10}$$

после чего происходит изменение весовых коэффициентов и порогов нейронных элементов:

$$w_{ij}(t) = w_{ij}(t-1) - \Delta w_{ij}(t), \tag{11}$$

$$T_j(t) = T_j(t-1) + \Delta T_j(t). \tag{12}$$

Шаг 4. Если алгоритм выполнен на всей обучающей выборке заданное количество итераций tmax, его выполнение останавливается. В противном случае происходит возврат к *шагу 3*.

Таким образом, алгоритм обратного распространения ошибки функционирует до тех пор, пока ошибка сети не станет меньше заданной, т.е. $E > E_m$ или $\sigma > E_m$, либо обучение не будет выполнено на всей выборке $t_{\max}$ раз.

Добавление «мнимых» нейронов при выравнивании размерностей слоев ИНС

Для выравнивания размерности слоев ИНС нами предложено добавление «мнимых» нейронов, у которых нулю равны:

– весовые коэффициенты $w_{ij} = 0 \; \forall i$ и их корректирующие значения $\Delta w_{ij} = 0 \; \forall i$,

– порог $T_j = 0$ и его корректирующее значение $\Delta T_j = 0$,

– весовые коэффициенты нейронов $w_{jk} = 0 \; \forall k$ последующего слоя, связанных с рассматриваемым «мнимым» нейроном, и соответствующие корректирующие значения $\Delta w_{jk} = 0 \; \forall k$.

Покажем, что при соблюдении определенных условий на любой итерации указанные величины для «мнимого» нейрона не изменяются в ходе выполнения алгоритма для всей обучающей выборки, а также докажем, что наличие «мнимого» нейрона в любом слое (входном, скрытом или выходном) не влияет на выполнение алгоритма.

Вычисление выходной активности.

Рассмотрим нейроны двух смежных слоев ИНС (см. рис. 1).

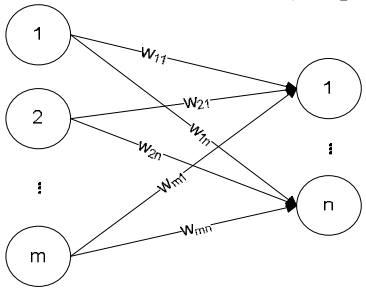

Рис. 1. Два смежных слоя нейронной сети

Для последующего слоя выход любого нейрона $y_j = F_{act}(S_j)$ согласно формуле (1) будет рассчитываться как

$$S_j = \sum_{i=1}^{m} w_{ij} y_i - T_j = w_{1j} x_1 + w_{2j} x_2 + \ldots + w_{mj} x_m - T_j, \qquad (13)$$

учитывая, что выход предыдущего слоя y_i является входом для последующего, т.е. $y_i = x_i$.

Если нейрон *m* предыдущего слоя является «мнимым», то $w_{mj} = 0 \; \forall j = \overline{1,n}$ (см. рис. 2).

Формула (13) для каждого из *n* нейронов последующего слоя примет вид:

$$S_j = \sum_{i=1}^{m} w_{ij} y_i - T_j = w_{1j} x_1 + w_{2j} x_2 + \ldots + 0 - T_j. \qquad (14)$$

Исходя из формулы (14) можно сформулировать следующее утверждение.

Утверждение 1. *«Мнимые» нейроны предыдущего слоя не влияют на вычисление индуцированного поля S_j и выхода (выходной активности) $y_j = F_{act}(S_j)$ последующего слоя.*

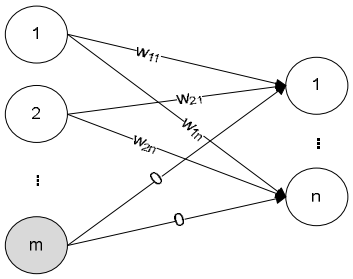

Рис. 2. Часть топологии ИНС с добавленным «мнимым» нейроном в предыдущем слое

Предположим, что нейрон *n* последующего слоя является «мнимым» (см. рис. 3).

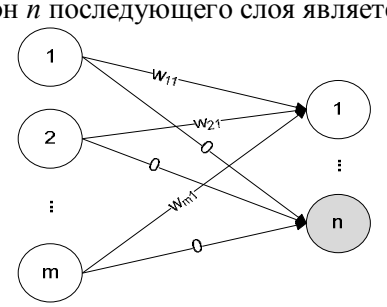

Рис. 3. Часть топологии ИНС с добавленным «мнимым» нейроном в последующем слое

Принимая во внимание $w_{in} = 0 \; \forall i = \overline{1,m}$ и $T_n = 0$, величина индуцированного поля S_n согласно формуле (13) будет равняться 0:

$$S_n = w_{1n}x_1 + w_{2n}x_2 + ... + w_{mn}x_m - T_n = 0 + 0 + ... + 0 - 0 = 0. \quad (15)$$

Тогда, подставляя (15) в (1), получаем, что значение выхода будет постоянно:

$$y_n = F_{act}(S_n) = F_{act}(0) = const. \quad (16)$$

На основании формулы (16) можно сформулировать утверждение 2.

Утверждение 2. *Расчетное значение выхода «мнимого» нейрона всегда постоянно и равно значению активационной функции от нуля.*

Для сигмоидальной функции активации выходная активность в соответствии с формулой (2) равна:

$$F_{act}(0) = \frac{1}{1+e^{-\alpha S_j}} = \frac{1}{1+e^0} = \frac{1}{2};$$

для гиперболического тангенса исходя из формулы (3)

$$F_{act}(0) = \beta\frac{e^{\alpha S_j} - e^{-\alpha S_j}}{e^{\alpha S_j} + e^{-\alpha S_j}} = \beta\frac{e^0 - e^0}{e^0 + e^0} = \beta\frac{1-1}{1+1} = 0;$$

для биполярной сигмоидальной функции активации, согласно (4), получаем:

$$F_{act}(0) = \frac{2}{1+e^{-\alpha S_j}} - 1 = \frac{2}{1+e^0} - 1 = \frac{2}{1+1} - 1 = 1 - 1 = 0.$$

Для биполярной сигмоидальной активационной функции и гиперболического тангенса утверждение 2 будет сформулировано следующим образом.

Утверждение 3. *Значение выхода «мнимого» нейрона для биполярной сигмоидальной активационной функции и гиперболического тангенса равно нулю.*

Оценка влияния «мнимого» нейрона в выходном слое на величину ошибки сети.

В формулы (5) и (6) входит величина $\Delta y_j^{\,s} = d_j^{\,s} - y_j^{\,s}$, характеризующая отклонение рассчитанного значения выхода от желаемого для каждого обучающего набора данных *s*. При $d_{im}^{\,s} = y_{im}^{\,s}$ величина погрешности нейрона $\Delta y_{im}^{\,s} = 0$. Для «мнимого» нейрона $y_{im}^{\,s}$, принимая во

внимание утверждение 2, справедливо $y_{im}{}^{s} = F_{act}(0)$. Таким образом, можно сформулировать утверждение 4.

Утверждение 4. *Для каждого набора данных s при величине желаемого значения «мнимого» нейрона выходного слоя равной* $d_{im}{}^{s} = F_{act}(0)$ *его наличие не влияет на ошибку сети.*

Рассмотрим значение локального градиента «мнимого» нейрона. Для выходного слоя согласно формуле (7) при учете $d_{im}{}^{s} = F_{act}(0)$ (см. утверждение 4) и $y_{im}{}^{s} = F_{act}(0)$ (см. утверждение 2) справедливо:

$$\delta_{im} = (y_{im} - d_{im})F'_{act}(0) = (F_{act}(0) - F_{act}(0)) \cdot F'_{act}(0) = 0. \quad (17)$$

Для скрытого слоя согласно (8) локальный градиент δ_{im} «мнимого» нейрона m предыдущего слоя с учетом $w_{mj} = 0 \; \forall j = \overline{1,n}$ (см. рис. 2) будет вычислен как:

$$\delta_{im} = \delta_m = F'_{act}(0)[w_{m1}\delta_1 + ... + w_{mn}\delta_n] = F'_{act}(0)[0 + .. + 0] = 0. \quad (18)$$

Рассмотрим влияние «мнимого» нейрона *n* в последующем слое на расчет локального градиента одного из действительных нейронов *i* (см. рис. 3). Принимая во внимание $w_{in} = 0 \; \forall i = \overline{1,m}$ справедливо соотношение:

$$\delta_i = F'_{act}(y_i)[w_{i1}\delta_1 + ... + w_{in}\delta_n] = F'_{act}(y_i)[w_{i1}\delta_1 + ... + 0] = F'_{act}(y_i)[w_{i1}\delta_1 + ... + w_{in-1}\delta_{n-1}]. \quad (19)$$

Исходя из формул (17)(17)–(19) можно сформулировать утверждение 5.

Утверждение 5. *Величина локального градиента «мнимого» нейрона выходного (при обеспечении равенства желаемого значения значению функции активации от нуля, т.е.* $d_{im}{}^{s} = F_{act}(0)$*) и скрытого слоев равна 0 (*$\delta_{im} = 0$*) и не влияет на расчет локального градиента нейрона предыдущего слоя.*

Рассмотрим влияние локального градиента «мнимого» нейрона на вычисление корректирующих значений его весовых коэффициентов и порога (см. рис. 3). Согласно формулам (9) и (10) и учитывая δ_{im}=0 (см. утверждение 5), можно записать:

$$\Delta w_{i\,im}(t) = r(\eta \Delta w_{i\,im}(t-1) + (1-\eta)\delta_{im} y_i) = r\eta \Delta w_{i\,im}(t-1), \quad (20)$$

$$\Delta T_{im}(t) = r(\eta \Delta T_{im}(t-1) + (1-\eta)\delta_{im}) = r\eta \Delta T_{im}(t-1). \quad (21)$$

В случае начальной инициализации нулевыми значениями $\Delta w_{i\,im}(0) = 0$ и $\Delta T_{im}(0) = 0$ все последующие корректирующие значения будут равняться нулю: $\Delta w_{i\,im}(t) = 0$ и $\Delta T_{im}(t) = 0$.

Согласно формулам (11), (12), (20), (21), а также учитывая $w_{i\,im} = 0 \; \forall i = \overline{1,m}$ и $T_{im} = 0$, значения весовых коэффициентов и порога «мнимого» нейрона будут вычислены как

$$w_{i\,im}(t) = w_{i\,im}(t-1) - \Delta w_{i\,im}(t) = w_{i\,im}(t-1) = 0,$$

$$T_{im}(t) = T_{im}(t-1) + \Delta T_{im}(t) = T_{im}(t-1) = 0.$$

Исходя из вышеизложенного можно сформулировать утверждение 6.

Утверждение 6. *При инициализации корректирующих значений весовых коэффициентов и порога «мнимого» нейрона нулевыми значениями (*$\Delta w_{i\,im}(0)=0$ *и* $\Delta T_{im}(0)=0$*) весовые коэффициенты и порог «мнимого» нейрона на любой итерации будут также принимать нулевые значения (*$w_{i\,im}(t) = 0$ *и* $T_i(m) = 0$*) при условии их нулевой инициализации.*

Заключение

На основании утверждений 1–6 можно сделать вывод о том, что наличие «мнимого» нейрона в слое ИНС не влияет на ход алгоритма обратного распространения ошибки с соблюдением ряда условий при инициализации параметров ИНС:

– весовые коэффициенты и порог «мнимого» нейрона равны нулю ($\Delta w_{i\,im}(0)=0$ для любых i и $T_{im}(0)=0$);

– корректирующие значения весовых коэффициентов и порога «мнимого» нейрона равны нулю ($w_{i\,im}(0)=0$ для любых i и $T_{im}(0)=0$);

– весовые коэффициенты нейронов $w_{jk}(0)=0$ последующего слоя, связанных с рассматриваемым «мнимым» нейроном, и соответствующие корректирующие значения $w_{jk}(0)=0$ для любых k равны нулю;

– желаемое значение «мнимого» нейрона выходного слоя для каждого набора обучающих данных s равно значению активационной функции от нуля $d_{im}^{\,s}=F_{act}(0)$ (для гиперболического тангенса и биполярного сигмоида равняется нулю, для сигмоидальной функции – 0,5).

Добавление «мнимых» нейронов позволяет адаптировать топологии искусственных нейронных сетей для использования *CUDA*-технологии, что дает возможность значительно (от двух до пяти раз, как получено нами в [2]) ускорить обучение ИНС.

ADAPTATION OF ARTIFICIAL NEURAL NETWORKS FOR CUDA-TECHNOLOGY APPLICATION

O.S. HILKO, S.P. KUNDAS, V.I. KOVALENKO

Abstract

The paper deals with the application of CUDA-technology on software implementation of direct and reverse passes of artificial neural network (ANN) based on back-propagation algorithm. It is shown that introduction of «imaginary» neurons helps to adapt the topology of the neural network to be trained and calculated with CUDA-technology. It is proved that «imaginary» neurons do not affect on the calculations in the back propagation algorithm.

Список литературы

1. *Хилько О.С., Коваленко В.И., Кундас С.П.* // Докл. БГУИР. 2010, № 7 (53). С. 83–88.
2. *Кундас С.П., Гишкелюк И.А., Коваленко В.И. и др.* Компьютерное моделирование миграции загрязняющих веществ в природных дисперсных средах. Мн.: МГЭУ им. А.Д. Сахарова, 2011.
3. *Антонов И.* // Хакер. 2009, № 127. С. 28–31.
4. *Хилько О.С.* // Доклады БГУИР. 2011, № 4 (58). С. 55–62.
5. *Хайкин С.* Нейронные сети. Полный курс. СПб : Вильямс, 2006.
6. *Головко В.А.* Нейронные сети: обучение, организация и применение. М., 2001.
7. *Кундас С.П., Коваленко В.И., Хилько О.С.* // Вестник ПГУ. 2009. № 9. С. 32–38.