УДК 004.89 DOI: 10.31114/2078-7707-2021-2-23-28

Модели нейросетей на основе тензорно-матричной теории

В.И. Слюсар

Полтавская государственная аграрная академия, г. Полтава, swadim@ukr.net

***Аннотация* — Рассмотрены варианты математической формализации описания структуры нейронных гиперсетей на основе предложенного автором семейства проникающих торцевых произведений матриц и тензоров, развёрнутых в блочные матрицы. Впервые введена операция блочного проникающего кронекеровского произведения матриц, позволяющая, например, формализовать модель входного слоя нейронной гиперсети, оперирующей множеством видеопотоков от нескольких видеокамер в различных спектральных диапазонах и обрабатываемых параллельно совокупностью нескольких нейросетей.**

***Ключевые слова* — матрица, тензор, тензорно-матричная теория, нейронная сеть, гиперсеть, функция активации, проникающее торцевое произведение, обобщённое торцевое произведение, блочное обобщённое произведение, блочное проникающее кронекеровское произведение.**

I. Введение

Важным направлением в развитии микроэлектроники является совершенствование элементной базы для реализации технологий искусственного интеллекта. Обработка в нейросетях видеопотоков в реальном масштабе времени неэффективна на конечном оборудовании пользователей с ограниченными вычислительными ресурсами, в том числе FPGA [1, 2]. Поэтому многие производители пошли путём создания специализированной элементной базы. Вместе с тем, традиционный матричный аппарат, являющийся основой существующих технологий реализации нейронных сетей, становится одним из сдерживающих факторов на пути перехода к внедрению нейронных гиперсетей [3]. Для уменьшения времени реакции ансамблей нейронных сетей может быть предложена реализация развитой в [4, 5] тензорно-матричной теории на основе проникающего торцевого произведения матриц.

Целью доклада является рассмотрение особенностей применения модифицированной тензорно-матричной теории для формализации моделей типовых нейросетей и их ансамблей.

II. Модели нейросетей

A. Проникающее торцевое произведение

Согласно определению, предложенному в [4], проникающим торцевым произведением p×g-матрицы **A** и n-мерного тензора **B**, развёрнутого в блочную матрицу, содержащую p×g-блоки (**B**=[B_n], n>1), является матрица вида:

$$\mathbf{A} \blacksquare \mathbf{B}=[\mathbf{A}\circ B_n], \quad (1)$$

где ■ - символ проникающего торцевого произведения, $\mathbf{A}\circ B_n$ представляет собой произведение Адамара.

Если тензор **B** записан как блок-строка, получим:

$$\mathbf{A} \blacksquare \mathbf{B}=[\mathbf{A}\circ B_r]=[\mathbf{A}\circ B_1 \ \ \mathbf{A}\circ B_2 \,\vdots\, \ldots \,\vdots\, \mathbf{A}\circ B_r \,\vdots\, \ldots]. \quad (2)$$

Например:

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}, \ \mathbf{B}=\left[\begin{array}{cc:cc:cc} b_{111} & b_{121} & b_{112} & b_{122} & b_{113} & b_{123} \\ b_{211} & b_{221} & b_{212} & b_{222} & b_{213} & b_{223} \\ b_{311} & b_{321} & b_{312} & b_{322} & b_{313} & b_{323} \end{array}\right],$$

$$\mathbf{A} \blacksquare \mathbf{B} = \left[\begin{array}{cc:cc:cc} a_{11}\cdot b_{111} & a_{12}\cdot b_{121} & a_{11}\cdot b_{112} & a_{12}\cdot b_{122} & a_{11}\cdot b_{113} & a_{12}\cdot b_{123} \\ a_{21}\cdot b_{211} & a_{22}\cdot b_{221} & a_{21}\cdot b_{212} & a_{22}\cdot b_{222} & a_{21}\cdot b_{213} & a_{22}\cdot b_{223} \\ a_{31}\cdot b_{311} & a_{32}\cdot b_{321} & a_{31}\cdot b_{312} & a_{32}\cdot b_{322} & a_{31}\cdot b_{313} & a_{32}\cdot b_{323} \end{array}\right].$$

В данном случае матрицу **A** можно рассматривать как исходную матрицу пикселей изображения на входе нейросети. При этом каждый блок матрицы **B** будет соответствовать блоку весовых коэффициентов нескольких нейронов в одном слое нейронной сети. С другой стороны, если рассматривать матрицу **A** как матрицу весовых коэффициентов нейросети, то в этом случае матрицу **B** можно также трактовать как набор отдельных кадров входного видеопотока.

В частном случае обработки отдельно взятого изображения **A** ансамблем нейросетей соответствующая модель нейронной гиперсети сведется к проникающему произведению вида:

$$\mathbf{A} \blacksquare \mathbf{B} = \mathrm{A} \blacksquare \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} =$$

$$= \begin{bmatrix} \mathrm{A}\circ B_{11} & \mathrm{A}\circ B_{12} & \cdots & \mathrm{A}\circ B_{1G} \\ \mathrm{A}\circ B_{21} & \mathrm{A}\circ B_{22} & \cdots & \mathrm{A}\circ B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{A}\circ B_{P1} & \mathrm{A}\circ B_{P2} & \cdots & \mathrm{A}\circ B_{PG} \end{bmatrix}. \quad (3)$$

Кстати, данный вариант проникающего произведения был предложен в 2006 г. в [6] под названием блочного произведения Адамара. Однако такое же название было использовано в [7] в отношении предложенной в [8] операции поблочного произведения блочных матриц.

Дальнейшие шаги по обработке цифровых данных в рассмотренных моделях нейросетей зависят от структуры и типа слоев. Например, в случае свёрточной нейросети, модель которой может быть представлена блок-строкой (2), результат проникающего произведения $\mathbf{A} \boxdot \mathbf{B}$ следует умножить на единичный вектор. При этом можно получить:

- вектор-строку $\mathbf{1}^{\mathrm{T}}(\mathbf{A} \boxdot \mathbf{B})$;
- вектор $(\mathbf{A} \boxdot \mathbf{B})\times\mathbf{1}$, где × – символ обычного матричного умножения, $\mathbf{1}$ – вектор единиц;
- матрицу $(\mathbf{A} \boxdot \mathbf{B})\ [\times]\mathbf{1}$, где [×] – символ блочного обычного произведения матриц, $\mathbf{1}$ – блок-вектор единиц;
- скаляр $\mathbf{1}^{\mathrm{T}}(\mathrm{A} \boxdot \mathrm{B})\mathbf{1}$.

Результаты такого умножения необходимо далее использовать в качестве аргумента функции активации нейрона, например:

$$\tanh[(\mathbf{A} \boxdot \mathbf{B})\times\mathbf{1}+\mathbf{d}] \text{ или } \mathrm{SReLU}[\mathbf{1}^{\mathrm{T}}(\mathbf{A} \boxdot \mathbf{B})\mathbf{1}+\mathbf{d}],$$

где $\mathbf{d}$ является вектором (скаляром).

В числе свойств проникающего произведения заслуживает внимания его связь с торцевым произведением матриц:

$$\mathbf{A} \square \mathbf{A} = \mathbf{A} \boxdot (\mathbf{A} \otimes \mathbf{1}^{\mathrm{T}}),\ \mathbf{c} \square \mathbf{A} = \mathbf{A} \square \mathbf{c} = \mathbf{c} \boxdot \mathbf{A} = \mathbf{A} \boxdot \mathbf{c},$$

где □ – символ торцевого произведения [4, 9], ⊗ - произведение Кронекера, $\mathbf{1}^{\mathrm{T}}$ – вектор-строка единиц, c – вектор.

Эти свойства позволяют применить для выполнения проникающего торцевого произведения оператор «tf.multiply», встроенный в библиотеку машинного обучения TensorFlow [10], поскольку данный оператор обеспечивает торцевое умножение вектора и матрицы [11]. Аналогичный подход был рассмотрен также в работе [3], где описана модель нейронной гиперсети на основе операции поэлементного произведения вектора на матрицу, что является частным случаем проникающего торцевого произведения. Однако вариант обработки данных, используемый в операторе «tf.multiply» и [3], не применим в отношении произведения матриц и требует предварительной векторизации матрицы меньшей размерности $\mathbf{A}$ в сочетании с векторизацией блоков [5] согласованной с ней блочной матрицы $\mathbf{B}$. Необходимая для решения этой задачи процедура блочной векторизации, предложенная в [5], также реализуется в рамках TensorFlow.

Если результатом первого слоя нейросети по-прежнему является матрица, хотя и меньшей размерности, то во втором слое операция проникающего торцевого произведения может быть применена повторно. Продемонстрируем это следующим образом. Предположим, что на выходе первого слоя нейросети имеет место матрица

$$\mathbf{A}_2=\mathrm{Softmax}[(\mathbf{A}_1 \boxdot \mathbf{B}_1)\ [\times]\ \mathbf{1}+\mathbf{d}_1],$$

где $\mathbf{A}_1$ – исходная, анализируемая нейросетью, матрица пикселей изображения, $\mathbf{B}_1$ – блок-матрица коэффициентов нейронов первого слоя, матрица постоянного смещения.

В данном случае применение функции активации Softmax в отношении матрицы означает, что эта функция выполняется для каждого элемента матрицы-аргумента.

Если использовать для второго слоя нейросети в качестве весовых коэффициентов блочную матрицу $\mathbf{B}_2$, размерность блоков которой согласована с размерностью $\mathbf{A}_2$, то можно получить, к примеру, следующую модель двухслойной нейронной сети с функцией активации ReLU и выходным результатом в виде вектора:

$$\mathrm{ReLU}[(\mathbf{A}_2 \boxdot \mathbf{B}_2)\times\mathbf{1}_2+\mathbf{d}_2]=$$

$$= \mathrm{ReLU}[(\{\mathrm{Softmax}[(\mathbf{A}_1 \boxdot \mathbf{B}_1)[\times]\mathbf{1}_1+\mathbf{d}_1]\} \boxdot \mathbf{B}_2)\times\mathbf{1}_2+\mathbf{d}_2].$$

Аналогичным образом формируется иерархическая модель произвольной многослойной нейросети (гиперсети) с различной комбинацией функций активации.

В интересах обработки данных в более сложных, многоуровневых иерархиях кластеров нейронных сетей предлагается использовать обобщённое проникающее произведение или его транспонированную версию [5, 12]. Данные операции умножения предназначены исключительно для блочных матриц, имеющих блоки равной размерности.

B. Обобщенное торцевое произведение

В частности **обобщённое торцевое произведение (ОТП)** блочных матриц $\mathrm{A}=\left[\mathrm{A}_{\mathrm{in}}\right]$ и $\mathrm{B}=\left[\mathrm{B}_{\mathrm{ig}}\right]$ с согласованным разбиением на блоки равной размерности и одинаковым количеством блок-строк порождает матрицу $\mathbf{A}\ [\square]\ \mathbf{B}$, в которой каждая i-я блок-строка представляет собой совокупность проникающих торцевых произведений всех блоков A_{in} i-ой блок-строки левой матрицы на соответствующую ей по номеру блок-строку $\mathrm{B}_{\mathrm{i}}=\left[\mathrm{B}_{\mathrm{i1}}\quad \mathrm{B}_{\mathrm{i2}}\quad \cdots\quad \mathrm{B}_{\mathrm{iG}}\right]$ правой матрицы $\mathbf{B}$:

$$\mathbf{A}\ [\square]\ \mathbf{B}=\left[A_{in} \boxdot \left[B_{i1}\quad B_{i2}\quad \cdots\quad B_{iG}\right]\right]. \qquad (4)$$

Пример:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1T} \\ A_{21} & A_{22} & \cdots & A_{2T} \\ \vdots & \vdots & \cdots & \vdots \\ A_{P1} & A_{P2} & \cdots & A_{PT} \end{bmatrix} [\square] \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} =$$

$$= \begin{bmatrix} A_{11} \boxdot [B_{11}\ B_{12}\ \cdots\ B_{1G}] & \cdots & A_{1T} \boxdot [B_{11}\ B_{12}\ \cdots\ B_{1G}] \\ A_{21} \boxdot [B_{21}\ B_{22}\ \cdots\ B_{2G}] & \cdots & A_{2T} \boxdot [B_{21}\ B_{22}\ \cdots\ B_{2G}] \\ \vdots & \ddots & \vdots \\ A_{P1} \boxdot [B_{P1}\ B_{P2}\ \cdots\ B_{PG}] & \cdots & A_{PT} \boxdot [B_{P1}\ B_{P2}\ \cdots\ B_{PG}] \end{bmatrix}.$$

Проводя сопоставление торцевого произведения [4] с матричной операцией (4) нетрудно заметить, что ОТП по сути является его аналогом, только на более высоком уровне обобщения, при котором в роли элементов матриц, фигурировавших прежде в торцевом умножении, теперь выступают матричные блоки, а вместо обычного произведения в ОТП фактически используется произведение Адамара (см. определение проникающего торцевого умножения (1)).

Применительно к нейросетям операция ОТП позволяет формализовать модель билинейного пулинга (объединения) в отношении нескольких параллельных потоков обработки видеоизображений. В этом случае, например, в качестве блок-строк матрицы

$$\mathbf{A}=\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1T} \\ A_{21} & A_{22} & \cdots & A_{2T} \\ \vdots & \vdots & \cdots & \vdots \\ A_{P1} & A_{P2} & \cdots & A_{PT} \end{bmatrix}$$

можно рассматривать отдельные кадры единичного видеопотока, которым соответствуют блок-строки коэффициентов отдельно взятого слоя нейросетей, образующие матрицу **B**. Возможен также другой вариант, когда в качестве блок-строк матрицы **A** удобно рассматривать фрагменты с типичным для нейросетей форматом (например, 128x128 пикселей) одного большого изображения, имеющего формат Full HD, 4K или 8K. В качестве эквивалента описанного подхода следует рассматривать использование **транспонированного обобщённого торцевого произведения (ТОТП)** матриц. Согласно определению [5, 12], для блочных матриц $\mathrm{A}=\left[A_{ij}\right]$ и $\mathrm{B}=\left[B_{gj}\right]$ с согласованным разбиением на блоки равной размерности и одинаковым количеством блок–столбцов транспонированное обобщённое торцевое произведение представляет собой матрицу **A** [■] **B**, в которой каждый j-й блок–столбец формируется как совокупность проникающих торцевых произведений всех блоков A_{ij} j-го блок–столбца левой матрицы на соответствующий ему по номеру блок–столбец $B_j=\begin{bmatrix} B_{1j} \\ \vdots \\ B_{Gj} \end{bmatrix}$ правой матрицы **B**:

$$\mathbf{A}\ [\blacksquare]\ \mathbf{B} = \begin{bmatrix} A_{ij} \blacksquare \begin{bmatrix} B_{1j} \\ \vdots \\ B_{Gj} \end{bmatrix} \end{bmatrix}. \qquad (5)$$

Пример [5, 12]:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1T} \\ A_{21} & A_{22} & \cdots & A_{2T} \\ \vdots & \vdots & \cdots & \vdots \\ A_{P1} & A_{P2} & \cdots & A_{PT} \end{bmatrix} [\blacksquare] \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} =$$

$$= \begin{bmatrix} A_{11} \blacksquare \begin{bmatrix} B_{11} \\ B_{21} \\ \vdots \\ B_{P1} \end{bmatrix} & A_{12} \blacksquare \begin{bmatrix} B_{12} \\ B_{22} \\ \vdots \\ B_{P2} \end{bmatrix} & \cdots & A_{1T} \blacksquare \begin{bmatrix} B_{1G} \\ B_{2G} \\ \vdots \\ B_{PG} \end{bmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ A_{P1} \blacksquare \begin{bmatrix} B_{11} \\ B_{21} \\ \vdots \\ B_{P1} \end{bmatrix} & A_{P2} \blacksquare \begin{bmatrix} B_{12} \\ B_{22} \\ \vdots \\ B_{P2} \end{bmatrix} & \cdots & A_{PT} \blacksquare \begin{bmatrix} B_{1G} \\ B_{2G} \\ \vdots \\ B_{PG} \end{bmatrix} \end{bmatrix}.$$

Примечательно, что в теоретическом плане можно сколь угодно долго повышать уровень обобщения в ОТП, рассматривая последовательно совокупность блоков блочных матриц как элементы некой новой "суперматрицы" (тензора), переходя тем самым от описания одних иерархических уровней сложных (фрактальных) систем искусственного интеллекта к другим. Подробнее о возможном варианте конкретного механизма такого обобщения речь пойдет далее при рассмотрении блочной модификации ОТП. О том же, насколько мощным оказывается указанный матричный аппарат, позволяет судить напрашивающаяся аналогия его абстрагированной процедуры с единством иерархических обобщений, присущих известным формам существования материи. Исходным пунктом таковых можно считать, к примеру, субэлементарные и элементарные частицы. Их производными, как известно, являются атомы, образующие молекулы, агрегатированные в физические тела и т. д. Таким образом, как справедливо отмечено в [5, 12] методологическая значимость ОТП выходит далеко за пределы рассматриваемых здесь вопросов, и заключает в себе большие потенциальные возможности как перспективного средства общесистемного анализа и синтеза.

C. Блочные обобщённые торцевые произведения

Как уже отмечалось, ОТП примечательно возможностью осуществления различных по уровням обобщений, при которых совокупность блоков блочных матриц может трактоваться как новый блок матрицы более высокой размерности. В отличие же от ОТП, идея его блочной модификации состоит в наложении ограничений на операцию умножения, которые сводятся к тому, что процедура ОТП выполняется поблочно, между блоками одного иерархического уровня.

Определение. **Блочным обобщённым торцевым произведением** тензора, представленного в виде dbp×ngs–матрицы $\mathbf{A}=\left[\mathrm{A}_{\mathrm{bg}}\right]_{dn}$, и тензора в виде dbp×nks–матрицы $\mathbf{B}=\left[B_{\mathrm{bk}}\right]_{dn}$, состоящих из одинакового количества (d×n) суперблоков, размерностью b×g и b×k соответственно, образованных

b блок-строками каждый, в составе g (матрица **A**) и k (матрица B) p×s–блоков, называется dbp×ngks–матрица **A** [[□]] **B**, каждый dn-й суперблок которой представляет собой обобщённое торцевое произведение соответствующих суперблоков исходных матриц, то есть

$$\mathbf{A}\ [[\square]]\ \mathbf{B} = \left[\mathrm{A}_{bg}\ [\square]\ \mathrm{B}_{bk}\right]_{dn}. \qquad (6)$$

Пример:

$$\mathrm{A} = \left[\mathrm{A}_1 \vdots \mathrm{A}_2\right] = \left[\begin{array}{cc:cc} A_{111} & A_{121} & A_{112} & A_{122} \\ A_{211} & A_{221} & A_{212} & A_{222} \end{array}\right],$$

$$\mathrm{B} = \left[\mathrm{B}_1 \vdots \mathrm{B}_2\right] = \left[\begin{array}{cc:cc} B_{111} & B_{121} & B_{112} & B_{122} \\ B_{211} & B_{221} & B_{212} & B_{222} \end{array}\right],$$

$$\mathbf{A}\ [[\square]]\ \mathbf{B} =$$

$$= \left[\begin{bmatrix} A_{111} & A_{121} \\ A_{211} & A_{221} \end{bmatrix} [\square] \begin{bmatrix} B_{111} & B_{121} \\ B_{211} & B_{221} \end{bmatrix} \vdots \begin{bmatrix} A_{112} & A_{122} \\ A_{212} & A_{222} \end{bmatrix} [\square] \begin{bmatrix} B_{112} & B_{122} \\ B_{212} & B_{222} \end{bmatrix}\right] =$$

$$= \left[\begin{array}{cc:cc} A_{111} \blacksquare [B_{111}\ B_{121}] & A_{121} \blacksquare [B_{111}\ B_{121}] & A_{112} \blacksquare [B_{112}\ B_{122}] & A_{122} \blacksquare [B_{112}\ B_{122}] \\ A_{211} \blacksquare [B_{211}\ B_{221}] & A_{221} \blacksquare [B_{211}\ B_{221}] & A_{212} \blacksquare [B_{212}\ B_{222}] & A_{222} \blacksquare [B_{212}\ B_{222}] \end{array}\right].$$

Как и в случае с обобщёнными торцевыми произведениями, целесообразность введения транспонированных альтернатив обобщённым торцевым операциям логически обусловлена уже самим фактом существования принципа симметрии. Поэтому, по аналогии с ранее рассмотренными понятиями, дадим указанным операциям матричного умножения следующие определения.

Определение. Транспонированным блочным обобщённым торцевым произведением (ТБОТП) тензора, развернутого в dbp×ngs–матрицу $\mathbf{A} = \left[\mathrm{A}_{bg}\right]_{dn}$ и тензора в виде dkp×ngs–матрицы $\mathbf{B} = \left[B_{kg}\right]_{dn}$, состоящих из одинакового количества (d×n) суперблоков, размерностью b×g и k×g соответственно, образованных g блок–столбцами каждый, в составе b (матрица **A**) и k (матрица **B**) p×s–блоков, называется dbkp×ngs матрица **A** [[■]] **B**, каждый dn-й суперблок которой представляет собой транспонированное обобщённое торцевое произведение соответствующих суперблоков исходных матриц, то есть

$$\mathbf{A}\ [[\blacksquare]]\ \mathbf{B} = \left[\mathrm{A}_{bg}\ [\blacksquare]\ \mathrm{B}_{kg}\right]_{dn}.$$

Пример [3, 10]:

$$\mathrm{A} = \left[\mathrm{A}_1 \vdots \mathrm{A}_2\right] = \left[\begin{array}{cc:cc} A_{111} & A_{121} & A_{112} & A_{122} \\ A_{211} & A_{221} & A_{212} & A_{222} \end{array}\right],$$

$$\mathrm{B} = \left[\mathrm{B}_1 \vdots \mathrm{B}_2\right] = \left[\begin{array}{cc:cc} B_{111} & B_{121} & B_{112} & B_{122} \\ B_{211} & B_{221} & B_{212} & B_{222} \end{array}\right],$$

$$\mathbf{A}\ [[\blacksquare]]\ \mathbf{B} =$$

$$= \left[\begin{bmatrix} A_{111} & A_{121} \\ A_{211} & A_{221} \end{bmatrix} [\blacksquare] \begin{bmatrix} B_{111} & B_{121} \\ B_{211} & B_{221} \end{bmatrix} \vdots \begin{bmatrix} A_{112} & A_{122} \\ A_{212} & A_{222} \end{bmatrix} [\blacksquare] \begin{bmatrix} B_{112} & B_{122} \\ B_{212} & B_{222} \end{bmatrix}\right] =$$

$$= \left[\begin{array}{cc:cc} A_{111} \blacksquare \begin{bmatrix} B_{111} \\ B_{211} \end{bmatrix} & A_{121} \blacksquare \begin{bmatrix} B_{121} \\ B_{221} \end{bmatrix} & A_{112} \blacksquare \begin{bmatrix} B_{112} \\ B_{212} \end{bmatrix} & A_{122} \blacksquare \begin{bmatrix} B_{122} \\ B_{222} \end{bmatrix} \\ A_{211} \blacksquare \begin{bmatrix} B_{111} \\ B_{211} \end{bmatrix} & A_{221} \blacksquare \begin{bmatrix} B_{121} \\ B_{221} \end{bmatrix} & A_{212} \blacksquare \begin{bmatrix} B_{112} \\ B_{212} \end{bmatrix} & A_{222} \blacksquare \begin{bmatrix} B_{122} \\ B_{222} \end{bmatrix} \end{array}\right].$$

D. Проникающее кронекеровское произведение и его блочная версия

В тех случаях, когда одно и то же изображение или видеопоток анализируются параллельно несколькими нейросетями, для формализации структуры их ансамбля удобно применить *проникающее прямое (кронекеровское) произведение*. Суть его сводится к обобщению операции (1) на случай, когда тензоры **A** и **B** развёрнуты в блочные матрицы с блоками одинаковой размерности:

$$\mathbf{A}\ [\blacksquare]\ \mathbf{B} = [\mathrm{A}_{ij} \blacksquare \mathbf{B}] = [\mathrm{A}_{ij} \circ \mathrm{B}_{mr}]. \qquad (7)$$

Применительно к рассматриваемой задаче обработки видеопотоков такой вариант матричного умножения позволяет получить поэлементное произведение каждого блока матрицы пикселей **A** на все блоки матрицы коэффициентов нейросети **B**:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1T} \\ A_{21} & A_{22} & \cdots & A_{2T} \\ \vdots & \vdots & \cdots & \vdots \\ A_{P1} & A_{P2} & \cdots & A_{PT} \end{bmatrix} [\blacksquare] \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} =$$

$$= \begin{bmatrix} A_{11} \blacksquare \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} & \cdots & A_{1T} \blacksquare \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ A_{P1} \blacksquare \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} & \cdots & A_{PT} \blacksquare \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} \end{bmatrix}.$$

В случае обработки параллельно несколькими нейросетями последовательности кадров одного видеопотока **A** можно записать:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1T} \end{bmatrix} [\blacksquare] \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PG} \end{bmatrix} =$$

$$= \left[A_{11} \boxdot \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{\mathrm{P}1} & B_{\mathrm{P}2} & \cdots & B_{PG} \end{bmatrix} \cdots A_{1T} \boxdot \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1G} \\ B_{21} & B_{22} & \cdots & B_{2G} \\ \vdots & \vdots & \cdots & \vdots \\ B_{\mathrm{P}1} & B_{\mathrm{P}2} & \cdots & B_{PG} \end{bmatrix} \right].$$

В данном выражении блок-строки матрицы **B** соответствуют коэффициентам одной нейросети.

В более общем случае для построения модели нейронной гиперсети следует воспользоваться *блочной версией проникающего кронекеровского произведения.* Суть ее сводится к тому, что для двух матриц с одинаковым количеством блоков первого уровня, содержащих произвольное количество блоков второго уровня размерностью p×g каждый, результат соответствующего умножения имеет вид:

$$\mathbf{A}\ [[\boxdot]]\ \mathbf{B}=[\mathbf{A}_{ij}\ [\boxdot]\ \mathbf{B}_{ij}]= [[\mathrm{A}_{bc} \circ \mathrm{B}_{mr}]_{ij}], \qquad (8)$$

где i, j – индексы нумерации блоков второго уровня; b, c и m, r – индексы нумерации блоков первого уровня внутри ij-го блока второго уровня матрицы **A** и **B** соответственно.

Пример.

$$\mathrm{A} = \left[\begin{array}{cc:cc} A_{111} & A_{121} & A_{112} & A_{122} \\ A_{211} & A_{221} & A_{212} & A_{222} \end{array}\right],\ \mathrm{B} = \left[\begin{array}{cc:cc} B_{111} & B_{121} & B_{112} & B_{122} \\ B_{211} & B_{221} & B_{212} & B_{222} \\ \hdashline B_{311} & B_{321} & B_{312} & B_{322} \end{array}\right],$$

$$\mathbf{A}\ [[\boxdot]]\ \mathbf{B}=$$

$$= \left[\begin{array}{cc:cc} A_{111} \boxdot \begin{bmatrix} B_{111} & B_{121} \\ B_{211} & B_{221} \\ B_{311} & B_{321} \end{bmatrix} & A_{121} \boxdot \begin{bmatrix} B_{111} & B_{121} \\ B_{211} & B_{221} \\ B_{311} & B_{321} \end{bmatrix} & & \\ A_{211} \boxdot \begin{bmatrix} B_{111} & B_{121} \\ B_{211} & B_{221} \\ B_{311} & B_{321} \end{bmatrix} & A_{221} \boxdot \begin{bmatrix} B_{111} & B_{121} \\ B_{211} & B_{221} \\ B_{311} & B_{321} \end{bmatrix} & & \\ \hdashline & & A_{112} \boxdot \begin{bmatrix} B_{112} & B_{122} \\ B_{212} & B_{222} \\ B_{312} & B_{322} \end{bmatrix} & A_{122} \boxdot \begin{bmatrix} B_{112} & B_{122} \\ B_{212} & B_{222} \\ B_{312} & B_{322} \end{bmatrix} \\ & & A_{212} \boxdot \begin{bmatrix} B_{112} & B_{122} \\ B_{212} & B_{222} \\ B_{312} & B_{322} \end{bmatrix} & A_{222} \boxdot \begin{bmatrix} B_{112} & B_{122} \\ B_{212} & B_{222} \\ B_{312} & B_{322} \end{bmatrix} \end{array}\right].$$

Произведение (8) позволяет формализовать модель входного слоя нейронной гиперсети, оперирующей например, множеством видеопотоков от нескольких видеокамер в различных спектральных диапазонах и обрабатываемых параллельно несколькими различными нейросетями. При реализации свёрточного слоя размерность блок-вектора единиц и его структура должны быть согласованы с результатом рассмотренного матричного умножения.

III. Заключение

В завершение следует отметить, что изложенный подход позволяет унифицировать формализацию описания нейросетей различной структуры и сложности. Предложенный вариант операции умножения скалярных матриц на блочные матрицы большей размерности является обобщением уже реализованного в различных средствах разработки нейросетей частного случая проникающего произведения, сводящегося к поэлементному умножению вектора и матрицы. Такое обобщение расширяет набор средств, применяемых при решении задач глубокого обучения нейросетей, и обеспечивает переход к эффективному обучению ансамблей нейросетей и нейронных гиперсетей. Кроме того, за счет отказа от промежуточной операции векторизации данных, предложенный тензорно-матричный аппарат создаёт предпосылки для обработки данных в периферийных нейросетевых устройствах в масштабе времени, близком к реальному.

Литература

[1] Соловьев Р.А., Кустов А.Г., Рухлов В.С. Методика реализации нейронной сети для распознания рукописных цифр в FPGA на основе вычислений с фиксированной точкой. // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2018. Вып. 3. С. 126-131. doi:10.31114/2078-7707-2018-3-126-131.

[2] R.A. Solovyev, D.V. Telpukhov, A.G. Kustov, V.S. Rukhlov, T.Y. Isaeva. Real-Time Recognition of Handwritten Digits in FPGA Based on Neural Network with Fixed Point Calculations / Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2019. Вып. 4. С. 38-43.

[3] Ha D., Dai A.M., Le Q.V. HyperNetworks. // The International Conference on Learning Representations (ICLR) 2017. – Toulon, 2017. - https://arxiv.org/abs/1609.09106.

[4] Слюсар В.И. Семейство торцевых произведений матриц и его свойства// Кибернетика и системный анализ. – 1999.- Том 35; № 3.- С. 379-384.- DOI: 10.1007/BF02733426.

[5] Слюсар В.И. Обобщенные торцевые произведения матриц в моделях цифровых антенных решеток с неидентичными каналами.//Известия высших учебных заведений. Радиоэлектроника.- 2003. - Том 46, № 10. - С. 15 – 26.

[6] Zeyad Al Zhour, Adem Kiliçman. Some new connections between matrix products for partitioned and non-partitioned matrices.// Computers and Mathematics with Applications, 54 (2007). Pp. 763 – 784.

[7] M. Günther, L. Klotz. Schur's theorem for a block Hadamard product. // Linear Algebra and its Applications, 437 (2012). – Pp. 948 – 956.

[8] R.A. Horn, R. Mathias, Y. Nakamura, Inequalities for unitarily invariant norms and bilinear matrix products.// Linear and Multilinear Algebra, 30 (1991). – Pp. 303 – 314.

[9] Слюсар В.И. Торцевые произведения матриц в радиолокационных приложениях// Известия высших учебных заведений. Радиоэлектроника.- 1998. - Том 41, № 3.- С. 71 - 75.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, at all. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR, abs/1603.04467, 2016. - http://arxiv.org/abs/1603.04467.

[11] Tensorflow, how to multiply a 2D tensor (matrix) by corresponding elements in a 1D vector. – 2017. - https://stackoverflow.com/questions/47817135/tensorflow-how-to-multiply-a-2d-tensor-matrix-by-corresponding-elements-in-a.

[12] Основы военно-технических исследований. Теория и приложения. Том. 2. Синтез средств информационного обеспечения вооружения и военной техники. / А.И. Миночкин, В.И. Рудаков, В.И. Слюсар. – Киев:«Гранмна, 2012. – С. 7 – 98; 354 – 521.

Neural Networks Models based on the tensor-matrix theory

V.I. Slyusar

Poltava State Agricultural Academy, Poltava, swadim@ukr.net

***Abstract* — The versions of the mathematical formalization of neural hypernetworks based on the family of penetrating face products of matrices and tensors expanded to the block matrices are considered. As an example, the matrix A in the penetrating face product of matrix A and block matrix B can be considered as a picture pixels matrix on the input of a neural network. In this case, every block of matrix B corresponds to a block of weight coefficients for a few neurons in one layer of the neural network. Further steps of data processing in the considered neural network can be varied depending on the structure and type of layers of neural network. In the case of convolutional neural networks the result of penetrating face products of matrices A and B has to be multiplied by a vector of one's 1. This multiplication can produce a scalar, a vector-row, a vector, or a matrix. The result of such multiplication can be used as argument of an activation function. For the data processing in hierarchies of neural hypernetworks clusters, the generalized face-splitting products of matrices and block version of these multiplications can be used. The operation of block penetrating Kronecker product of matrices has been introduced to simulate the input layer of a neural hypernetwork which processes multiple video streams from several video cameras in different spectral ranges in parallel by a set of several neural networks.**

***Keywords* — matrix, tensor, tensor-matrix theory, neural network, hypernetwork, activation function, penetrating face multiplication, generalized face-splitting product, block generalized face-splitting product, block penetrating Kronecker product.**

REFERENCES

[1] Solovyev R.A., Kustov A.G., Rukhlov V.S. The Technique for Implementing a Neural Network for Recognizing Handwritten Digits in FPGAs Based on Fixed Point Calculations // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2018. Issue 3. P. 126-131. doi:10.31114/2078-7707-2018-3-126-131

[2] R.A. Solovyev, D.V. Telpukhov, A.G. Kustov, V.S. Rukhlov, T.Y. Isaeva. Real-Time Recognition of Handwritten Digits in FPGA Based on Neural Network with Fixed Point Calculations // Problems of Perspective Micro- and Nanoelectronic Systems Developmen. 2019. Issue 4. Pp. 38 - 43. DOI: 10.31114/2078-7707-2019-4-38-43

[3] Ha D., Dai A.M., Le Q.V. HyperNetworks. // The International Conference on Learning Representations (ICLR) 2017. – Toulon, 2017. - https://arxiv.org/abs/1609.09106.

[4] Slyusar V.I. A family of face products of matrices and its properties. Cybernetics and Systems Analysis 35, 379–384 (1999). doi: 10.1007/BF02733426.

[5] Slyusar V.I. Generalized face-products of matrices in models of digital antenna arrays with nonidentical channels// Radioelectronics and Communications Systems. – 2003, Vol. 46; Part 10, pages 9-17.

[6] Zeyad Al Zhour, Adem Kiliçman. Some new connections between matrix products for partitioned and non-partitioned matrices.// Computers and Mathematics with Applications, 54 (2007). Pp. 763 – 784.

[7] M. Günther, L. Klotz. Schur's theorem for a block Hadamard product. // Linear Algebra and its Applications, 437 (2012). – Pp. 948 – 956.

[8] R.A. Horn, R. Mathias, Y. Nakamura, Inequalities for unitarily invariant norms and bilinear matrix products.// Linear and Multilinear Algebra, 30 (1991). – Pp. 303 – 314.

[9] Slyusar V.I. End products in matrices in radar applications// Radioelectronics and Communications Systems.– 1998, Vol. 41; Number 3, pages 50-53.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR, abs/1603.04467, 2016. - http://arxiv.org/abs/1603.04467.

[11] Tensorflow, how to multiply a 2D tensor (matrix) by corresponding elements in a 1D vector. – 2017. - https://stackoverflow.com/questions/47817135/tensorflow-how-to-multiply-a-2d-tensor-matrix-by-corresponding-elements-in-a.

[12] Minochkin, A.I., Rudakov, V.I. & Slyusar, V.I. "Osnovy voyenno-tekhnicheskikh issledovanii. Teoriia i prilozheniia. T. 2. Sintez sredstv informatsionnogo obespecheniia vooruzheniia i voennoi tekhniki" [Theoretical bases of military-technical researches]. Vol. 2. K. 2012. Pp. 7–98; 354–521.