

ну. В декодере 8 происходит обратное преобразование системы остаточных классов: восстанавливается значение, выполняется проверка на наличие ошибок и их коррекция.

Поскольку в канал связи передаются не значения чисел, а их остатки от деления на соответствующие модули, и выбор несущей частоты осуществляется по псевдослучайному закону, данные дополнительно защищены от несанкционированного доступа. Воздействие на сигнал спланированных помех на одной из несущих частот приведет к искажению только остатка по соответствующему модулю, что можно легко обнаружить и исправить введением дополнительных модулей.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1104 с.
2. Столлингс В. Беспроводные линии связи и сети.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 640 с.
3. Петрович И.Т., Размахнин М.К. Системы связи с шумоподобными сигналами. – М.: Сов. радио, 1969. – 232 с.
4. Акушкин И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. - М.: Сов. радио. – 1968. – 460 с.
5. Торгашев В.А. Система остаточных классов и надёжность ЦВМ. – М.: Сов. радио, 1973. – 274 с.

Материал поступил в редакцию 24.01.08

JACKIV N.G., JACKIV V.V., KREPICH R.V. Method of coding given in wireless computer networks on the basis of transformation of system of residual classes

The offered method of coding and transfer given on the basis of expansion of a spectrum of a signal and transformation of system of residual classes allows to reduce influence of handicapes and to raise a level of protection from the non-authorized access.

УДК 621.38

Боровый А., Константакос В., Кочан В., Турченко В., Саченко А., Лаопулос Т.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОЦЕНКИ ЭНЕРГИИ ВЫПОЛНЕНИЯ КОМАНД МИКРОПРОЦЕССОРОМ

Введение. При использовании систем с автономным питанием возникает вопрос о снижении их энергопотребления. В связи с тем, что цена на источники питания большой емкости относительно высока, все более актуальной становится задача их эффективного использования. Оценка затрат энергии вычислительных устройств показывает, что традиционные подходы, используемые при их построении, не всегда оптимальны с точки зрения энергопотребления. Увеличить эффективность использования источников питания можно разными путями:

- за счет выполнения аппаратного обеспечения по технологии с низким энергопотреблением;
- за счет оптимизации структуры аппаратного обеспечения, которая подразумевает снижение энергопотребления (сюда относятся методы уменьшения энергопотребления за счет динамической реконфигурации структуры, например, отключение временно неиспользуемых узлов);
- за счет оптимизации алгоритмов работы устройств системы и их программного обеспечения.

Все эти методы используются на практике, но они отличаются перспективой и сферами применения.

Традиционно, для уменьшения энергопотребления, автономные системы строятся на базе КМОП технологии. Особенностью этой технологии является очень низкий ток потребления элементов системы, когда они находятся в статическом состоянии. В момент переключения элементов (смены состояния выхода) потребление увеличивается в десятки раз (за счет сквозного тока и перезаряда паразитных емкостей), таким образом, энергопотребление системы зависит от количества переключаемых элементов. Очевидно, что в таком случае энергопотребление зависит также и от частоты переключения. На сегодня потенциал КМОП-технологии в значительной мере исчерпан, поэтому этот путь уменьшения энергопотребления требует значительных капиталовложений для фундаментальных исследований с целью разработки новых версий технологических процессов.

Поэтому все чаще используется второй метод. Но даже в системах на отдельных узлах этот метод имеет серьезные ограничения, например, время включения узла. В процессорах и системах на одном кристалле этот метод могут использовать только разработчики

новых устройств. Поэтому самым серьезным недостатком этого метода является невозможность его применения для существующих микропроцессоров.

Наиболее доступным (с точки зрения разработчиков систем, использующих автономное питание), является третий метод, предполагающий оптимизацию программного обеспечения, исходя из минимального энергопотребления. Главным преимуществом данного метода является возможность оптимизации программного обеспечения для всех типов микропроцессоров на всех стадиях разработки и эксплуатации систем (а не только для новых моделей). Этот метод был впервые применен в 1994 г. В. Тивари при анализе программного обеспечения, созданного для процессора Intel 486DX. В результате проведенных работ энергопотребление уменьшилось на 40%.

Как видим, последний метод является одним из самых перспективных, оставаясь при этом очень гибким, т.е. его можно успешно комбинировать с другими методами. Однако для эффективного применения данного метода требуется:

- провести детальное исследование самого процесса потребления энергии микропроцессором, как во время выполнения отдельных команд, так и во время межкомандных переходов;
- разработать адекватные модели энергопотребления командами микропроцессора и межкомандными переходами;
- создать средства оценки энергии выполнения заданной программы микропроцессором.

При этом каждый следующий этап полностью базируется на результатах предыдущего, и его точность определяется как собственными погрешностями, так и погрешностями предыдущих этапов. Среди последних исследований процесса энергопотребления микропроцессорами следует выделить работы [3, 4, 5]. В них предложена методика измерения энергии команд и межкомандных переходов с помощью специализированной системы измерительной системы, а также разработаны модели энергопотребления, учитывающие ряд особенностей работы микропроцессора ARM7TDMI. Однако предложенные модели не обеспечивают высокой точности оценки энергии команд и межкомандных переходов в исследуемых программах. Это вызвано тем, что предложенный подход к построению моделей энергопотребления требует исследования потребления энергии всех

Боровый Андрей, аспирант кафедры информационно-вычислительных систем и управления факультета компьютерных информационных технологий, член группы прецизионных измерительных систем НИИ Интеллектуальных компьютерных систем Тернопольского национального экономического университета, г. Тернополь, Украина.

Константакос Василеос, аспирант по радиозлектронике на факультете физики в университете им. Аристотеля г. Салоники, Греция.

Владимир Кочан, кандидат технических наук, доцент кафедры информационно-вычислительных систем и управления, факультета компьютерных информационных технологий и директор НИИ Интеллектуальных компьютерных систем Тернопольского национального экономического университета, г. Тернополь, Украина.

Лаопулос Теодор, профессор и директор лаборатории электроники, факультет физики, университет им. Аристотеля, г. Салоники, Греция.

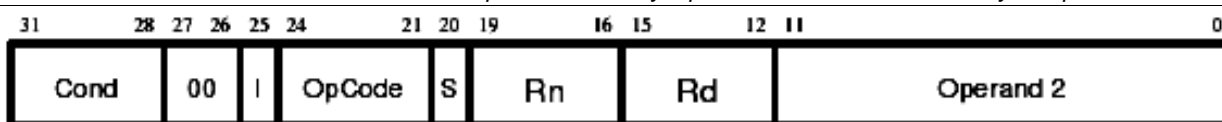


Рис. 1. Синтаксис набора команд обработки данных

команд во всех режимах работы и при всех возможных вариантах межкомандных переходов [4]. Оценка погрешностей существующей измерительной установки, а также варианты устранения недостатков предложены в [6].

Целью данной статьи является исследование возможности использования обобщающих свойств нейронных сетей (НС) для уменьшения трудоемкости исследований при построении математических моделей, которые могут быть использованы для прогноза (оценки) энергии команд и межкомандных переходов в исследуемых и оптимизируемых программах.

Проблемы использования нейронных сетей для прогноза энергии команд. Процессорное ядро ARM7TDMI работает в одном из следующих режимов: режим команд ARM – ядро работает с 32-х разрядными командами, и режим THUMB – ядро работает с 16-и разрядными командами. Все исследования в рамках данной статьи проводились для команд в режиме ARM.

В общем случае задача построения матмоделей процесса, характеризующего многими параметрами, может быть решена на базе корреляционного анализа. Однако большинство процессов выполнения команд происходят на аппаратном уровне в процессорном ядре. Для пользователя и программиста эти процессы невидимы, поэтому использование корреляционного анализа в этом случае затруднительно.

Более перспективным представляется использование для построения матмоделей потребления энергии команд и межкомандных переходов обобщающих свойств НС. В этом случае можно прогнозировать энергопотребление микропроцессора для выполнения программы не углубляясь во внутреннюю структуру микропроцессора. В процессе обучения НС должна обобщить информацию, содержащуюся в ограниченной выборке результатов измерения энергии команд и межкомандных переходов, для того, чтобы в дальнейшем прогнозировать потребление энергии всех команд во всех режимах работы и при всех возможных вариантах межкомандных переходов с достаточной точностью.

Для проверки этой гипотезы, по результатам проведенных экспериментальных исследований, была создана выборка для обучения НС, в которой на вход НС подавались закодированные команды обработки данных и режим адресации. НС на своем выходе должна прогнозировать потребляемую энергию этих команд. При этом 90% результатов измерения использовались в составе обучающей выборки, а остальные 10% – для оценки погрешности прогноза. В этом исследовании используется традиционный трехуровневый перцептрон. Первый уровень – распределительный, второй использует сигмоидную функцию активации, а третий – линейный выходной нейрон, который отображает количество потребленной энергии при использовании микропроцессором данной команды. Было установлено, что процесс обучения НС протекал неустойчиво, а погрешность прогноза колебалась от 16 до более 100%.

Причиной такого большого разброса результатов прогноза, по нашему мнению, являются разные характеры энергопотребления команд и междукомандных переходов и недостаточный объем обучающей выборки. На основании полученных результатов решено отказаться от данного метода формирования обучающей выборки и провести анализ известных данных на предмет поиска информации, которая более адекватно отображала бы процесс энергопотребления.

Анализ структуры команд для формирования обучающей выборки. Как было отмечено выше, процессорное ядро работает либо с 32-х либо с 16-и разрядными командами. Поэтому для формирования обучающей выборки было решено использовать двоичное представление ассемблерных команд. Согласно [7] все команды можно разбить на следующие отдельные группы:

1) команды ветвления;

- 2) команды обработки данных;
- 3) команды передачи статуса регистра;
- 4) команды загрузки и сохранения;
- 5) команды сопроцессора;
- 6) команды создания исключений.

Каждая из этих групп имеет собственную методику кодирования, вследствие чего предлагается использовать для каждой из них отдельную НС. Таким образом, в результате получим каскад (набор) НС для прогнозирования энергопотребления микропроцессором при выполнении всех команд и всех межкомандных переходов.

В связи с тем, что НС хорошо обучается на обучающей выборке большого объема, исследования проводились для инструкций обработки данных, так как эта группа включает 16 инструкций в 11 режимах адресации. Таким образом, для первого варианта получим обучающую выборку объемом 176 векторов.

Задачей первого варианта является исследование возможности прогноза энергии выполнения всех команд с заданной точностью. Сначала необходимо преобразовать команды в двоичные коды [8] входа НС. На рис. 1 представлен общий синтаксис и размещение полей для команд обработки данных. Описание этих полей представлено в табл. 1. Если на вход НС команда подается целиком – для её кодирования надо 32 нейрона входного уровня, на которые поступают значения 0 или 1. Это количество нейронов соответствует количеству бит, которые передаются микропроцессору для выполнения команды.

Таблица 1. Перечень битов, соответствующих командам обработки данных

Номера битов	Значение битов
31:28	Код условия
27:26	Постоянные биты
25	Операнд переменной
24:21	Код операции
20	Код установки условия
19:16	Первый операнд (всегда регистр)
15:12	Регистр конечного значения
11:0	Операнд 2

Из данных [4] видно, что биты [31:26] и [20:12] являются постоянными. Предварительные исследования показали, что наличие неизменных значений на некоторых входах НС значительно снижают качество обучения. Поэтому эти биты исключены из обучающей выборки. Анализ поля Operand 2 показал, что биты [3:2] также являются постоянными, поэтому из обучающей выборки они также были исключены.

Структура нейронной сети. На основе проведенного анализа, можно сделать следующее заключение о структуре нейронной сети. Входной слой содержит нейроны, количество которых равно количеству известных битов. В данном случае под известными битами подразумеваем биты, меняющие свое значение в зависимости от команды и известных параметров. Таким образом, на входе НС находятся 13 распределительных нейронов, входы которых соответствуют битам: [25:21], [11:4 – 1:0]. Количество нейронов скрытого слоя равно 6, что соответствует известным методам определения количества нейронов. Выводной слой содержит 1 нейрон, который возвращает прогнозируемое значение энергии потребленной микропроцессором при исполнении команды. В качестве модели нейронной сети выбран обычный многоуровневый перцептрон с алгоритмом обратного распространения ошибки с адаптивным шагом обучения [9]. Данная модель нейронной сети является достаточной простой и более детально рассматривается в [10-12].

Выходное значение трехуровневого перцептрона (рис. 1) можно представить в виде:

$$y = F_3 \left(\sum_{i=1}^N w_{i3} h_i - T \right), \quad (1)$$

где N является числом нейронов скрытого слоя, w_{i3} весовой коэффициент, идущий от i нейрона скрытого слоя к выходному нейрону, h_i выходное значение i нейрона, T порог выходного нейрона, а F_3 функция активации выходного нейрона.

Выходное значение нейрона j , который находится в скрытом слое, рассчитывается следующим образом:

$$h_j = F_2 \left(\sum_{i=1}^M w_{ij} x_i - T_j \right), \quad (2)$$

где w_{ij} весовой коэффициент связи входного нейрона и j нейрона скрытого слоя, x_i входные значения а T_j порог j нейрона. Для скрытого уровня функцией активации выступает сигмоида, для выходного уровня - линейная функция, с коэффициентом K [29].

Результаты экспериментальных исследований. Для проведения экспериментальных исследований сначала созданы две выборки: одну, размером 100 векторов (обучающую), и другую, размером 76 векторов (контрольную, для контроля качества прогноза). Выборки сформированы путем последовательного перебора двоичного кода соответствующих полей набора команд обработки данных (рис. 1).

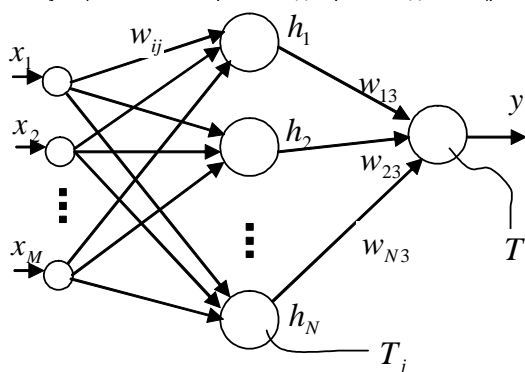


Рис. 1. Структура нейронной сети

Первая группа экспериментов проведена с целью проверки того, насколько результат прогноза зависит от распределения данных, подаваемых на вход НС. Для этого проведено 10 экспериментов, организованных следующим образом. 100 векторов созданной обучающей выборки были пронумерованы:

1. Для эксперимента № 1 – как обычная последовательность векторов;
2. Для эксперимента № 2 – как 1, 2, 1, 2, ... 1, 2;
3. Для эксперимента № 3 – как 1, 2, 3, 1, 2, 3, ... 1, 2, 3;
4. По аналогичному принципу до эксперимента № 10.

В дальнейшем в обучающих выборках для каждого эксперимента векторы соответствующих номеров были перетасованы по правилу – 1, 1, 1, ... 2, 2, 2, ... 3, 3, 3 ... и подавались на входы соответствующих НС. Результаты проведенных экспериментов представлены в табл. 2.

Как видно из таблицы 2, НС во всех случаях учится довольно хорошо – средняя ошибка обучения не превышает 2,0%, что вполне приемлемо для оценки энергии выполнения команд. Однако погрешность прогноза для 76 векторов контрольной выборки достаточно велика. А главное – средняя погрешность прогноза $\delta X_{\text{прогн. ср.}}$

достаточно сильно зависит от номера эксперимента, то есть от того, каким образом перетасованы векторы обучающей выборки. Наилучший результат достигнут при проведении эксперимента №6, где средняя погрешность прогноза не превышает приемлемого значения (2,1%). Однако даже этот результат не является удовлетворительным, поскольку максимальная погрешность является достаточно высокой (более 16%).

Вторая группа экспериментов проведена с целью проверки целесообразности деления команд обработки данных на группы в соответствии с рекомендациями, приведенными в [4]. Согласно [4] все команды обработки данных были разделены на 3 группы, в соответствии с характером энергопотребления. В табл. 3. предоставлены результаты экспериментов для первой группы, которая состоит из 8 команд обработки данных.

Общее количество векторов обучающей выборки составляло 88, из которых 84 вектора были использованы для обучения НС и 4 вектора для прогноза. Выбор такого соотношения предопределен необходимостью проверки характера энергопотребления команд этой группы. В результате анализа проведения эксперимента № 1 было выявлено, что из всего набора команд операции работы с числами составляют 9% (каждая команда имеет 10 операций работы с регистрами и одну операцию работы с числами). Поэтому при проведении экспериментов 2...4 операции работы с числами были исключены из обучающей выборки. В результате погрешность прогноза резко снизилась, что указывает на существенно разный характер потребления операций работы с регистрами и с числами. При проведении эксперимента № 2 прогнозировалась команда TEQ в 7,8, 10 и 11 режимах адресации. При проведении эксперимента № 3 выборка обучения пересортирована на четные и нечетные векторы. Из этого набора 76 векторов использовались для обучения и 4 вектора для прогноза. При проведении эксперимента № 4 сначала на вход НС подавались четные, а потом нечетные вектора.

Таблица 2. Результаты экспериментов

Номер эксперимента	1	2	3	4	5	6	7	8	9	10
Среднеквадратичная погрешность *10 ⁻⁴ %	4,1	2,5	1,9	0,9	2,0	2,0	1,3	9,3	4,9	2,7
К-во эпох обучения x*10 ⁴	21,5	3,9	17,6	58,9	21,9	16,3	103,5	5,2	21,3	14,2
Средняя погрешность обучения $\delta X_{\text{обуч. ср.}}$ %	1,3	1,0	0,9	0,6	0,9	0,9	0,8	2,0	1,4	1,2
Максимальная погрешность обучения $\delta X_{\text{обуч. макс.}}$ %	4,3	2,9	4,2	3,2	4,8	3,6	3,4	9,6	5,3	4,6
Средняя погрешность прогноза $\delta X_{\text{прогн. ср.}}$ %	14,0	5,2	2,5	5,1	2,2	2,1	2,5	3,5	2,8	2,6
Максимальная погрешность прогноза $\delta X_{\text{прогн. макс.}}$ %	38,3	17,4	23,0	22,5	10,3	16,0	14,3	14,8	10,7	15,5

Таблица 3. Результаты верификации нейронной сети для команд обработки данных

Номер эксперимента	1	2	3	4
Среднеквадратичная погрешность *10 ⁻⁴ %	0,8	2,0	1,6	1,3
К-во эпох обучения x*10 ⁴	0,9	16,1	28,0	0,1
Средняя погрешность обучения $\delta X_{\text{обуч. ср.}}$ %	2,4%	0,7%	0,6%	2,8%
Максимальная погрешность обучения $\delta X_{\text{обуч. макс.}}$ %	9,8%	3,7%	3,3%	12,0%
Средняя погрешность прогноза $\delta X_{\text{прогн. ср.}}$ %	2,4%	2,1%	1,2%	3,3%
Максимальная погрешность прогноза $\delta X_{\text{прогн. макс.}}$ %	3,5%	4,0%	2,5%	5,8%

Таблица 4. Результаты опытов

Номер эксперимента	1	2	3
Кол-во векторов обучения	30	20	10
Кол-во векторов прогноза	10	20	30
Среднеквадратичная погрешность $\cdot 10^{-4}$ %	10	3,6	1
К-во эпох обучения $x \cdot 10^4$	53,8	16,3	17,5
Средняя погрешность обучения $\delta X_{\text{обуч. ср.}}$ %	0,5	0,3	0,3
Максимальная погрешность обучения $\delta X_{\text{обуч. макс.}}$ %	1,3	0,8	0,8
Средняя погрешность прогноза $\delta X_{\text{прогн. ср.}}$ %	1,1	1,3	3,5
Максимальная погрешность прогноза $\delta X_{\text{прогн. макс.}}$ %	2,8	5,5	11,5

Как видно из табл. 3, данные обучающей выборки, несмотря на деление на группы, не являются однородными. Это указывает на то, что команды, объединенные в одну группу, имеют все же разный характер энергопотребления. Поэтому принято решение изменить принцип распределения команд по группам согласно характеру энергопотребления. Анализ результатов измерения показал, что команды по характеру энергопотребления целесообразно разделить на 9 групп. Дальнейшие исследования проводились для группы, которая объединяла наибольшее количество команд (CMP, ORR, SBC, SUB). При проведении эксперимента № 1 (табл. 4) обучающая выборка содержала 30 векторов, а 10 векторов служили для оценки погрешности прогноза. При проведении эксперимента № 2 это соотношение было изменено – обучающая выборка содержала 20 векторов, а 20 векторов служили для оценки погрешности прогноза, при проведении эксперимента № 3 обучающая выборка содержала 10 векторов, а 30 векторов служили для оценки погрешности прогноза.

Как видно из табл. 4, средняя погрешность прогноза при обучающей выборке 10 векторов, то есть 25% результатов измерения, средняя погрешность прогноза 75% результатов измерения составляет 3,5%, что является приемлемым с точки зрения компромисса между необходимой трудоемкостью экспериментальных исследований и достижимой точностью оценки энергии, необходимой для выполнения программы микропроцессором.

Заключение. Как видно из проведенных исследований, нейронные сети могут успешно использоваться для оценки энергопотребления микропроцессора при выполнении команд. Однако при формировании обучающих выборок большое внимание надо уделить однородности характера данных. При условиях правильного формирования выборок, нейронная сеть может достаточно точно прогнозировать потребляемую энергию даже при очень небольшом объеме результатов экспериментальных исследований (результатов измерения энергии выполнения отдельных команд). Однако для обеспечения высокой точности прогноза необходимо:

- удалить одинаковые для всех элементов обучающих выборок данные («зашумляющих») нейронную сеть);
- правильно определить критерии сортировки результатов измерения при создании выборок обучения.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Power Analysis of Embedded software: A First Step Towards Software Power Minimization / Vivek Tiwari, Sharad Malik and

- Andrew Wolfe // IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Том. 2, № 4, December 1994, С. 437 - 445.
2. Instruction Level Power Analysis and Optimization of Software / Vivek Tiwari, Sharad Malik, Andrew Wolfe, Mike Tien-Chien Lee // Journal of VLSI Signal Processing, Том. 13, № 2-3, Август 1996, С. 223 - 238.
3. IS/2000-30093/EASY Project: "Instruction-level power measurement methodology" / S. Nikolaidis, N. Kavvadias, Th. Laopoulos; Инв. №: EASY/WP2/AUTH/DL/P/D8/B1, Март 2002 - 24 с.
4. IS/2000-30093/EASY Project: Instruction level power measurements and analysis / S. Nikolaidis, N. Kavvadias, Th. Laopoulos; Инв. №: EASY/WP2/AUTH/DL/P/D15/B1, Октябрь 2002 - 44 с.
5. Instruction level energy modeling for pipelined processors / S. Nikolaidis, N. Kavvadias, Th. Laopoulos, L. Bisdounis, S. Blonias // Int. Workshop Power and Timing Modeling, Optimization and Simulation, Турин, Италия, Октябрь 2003.
6. Analysis of Circuits for Measurement of Energy of Central Processor Unit's Instruction Executing / A. Borovyi, V. Kochan, A. Sachenko, V. Konstandakos, V. Yaskilka // Proceedings of the Forth IEEE Workshop on "Intelligent Data Acquisition and Advanced Computing Systems: Technology and Application", IDAACS'2007. - Дортмунд, Германия. - 2007, Сентябрь 6-8. С. 42 - 46
7. Advanced RISC Machines Ltd. ARM Architecture Reference Manual - ARM DDI 0100E, Copyright 1966-2000 ARM Limited.
8. Peter Knaggs, Stephen Welsh. ARM Assembly Language Programming, School of Design, Engineering & Computing, Bournemouth University, 2004. - 142 с.
9. Нейронные сети: обучение, организация и применение: Учеб. пособие по направлению "Прикл. математика и физика" / В.А. Головкин, А.И. Галушкин; Под общ. ред. А.И. Галушкина. - М.: Изд. предприятие ред. журн. "Радиотехника", 2001. - 256 с.
10. Multilayer Feedforward Networks are Universal Approximators / K. Hornik, M. Stinchcombe, H. White // Neural Networks 2 (1989). С. 359-366.
11. A Neural Controller Based on Backpropagation Algorithm / M. Saerens, A. Soquet // Proceedings of "First IEE International Conference on Artificial Neural Networks", Лондон, Великобритания, 1989. - С. 211-215.
12. A Non-linear Regulator Design in the Presence of System Uncertainties Using Multi-layered Neural Networks / Y. Iguni, H. Sakai, H. Tokumaru //, IEEE Transactions on Neural Networks 2 (1991). С. 410-417.

Материал поступил в редакцию 24.01.08

BOROVYJ A., KONSTANTAKOS V., KOCHAN V., TURCHENKO V., SACHENKO A., LAOPOULOS T. Use nejronnyh of networks for an estimation of energy of performance of commands by the microprocessor

In the given job the analysis of use нейронных of networks for forecasting energy of consumption by the microprocessor is carried out at performance of commands. But the basis of the received data can make a conclusion, that with the help of this method the put task can be decided successfully.