УДК 004.932.2

Н.О. Іванова

# ПОРІВНЯННЯ ЕФЕКТИВНОСТІ ПРОГРАМНИХ РЕАЛІЗАЦІЙ АЛГОРИТМУ СТИСКАННЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МЕРЕЖІ КОХОНЕНА В СЕРЕДОВИЩІ MATLAB І МОВОЮ С

Одним з програмних засобів, що дозволяють моделювати роботу штучних нейронних мереж, є пакет MATLAB і його розширення Neural Networks Toolbox, який містить засоби для проектування, моделювання, навчання й використання великої кількості штучних нейронних мереж: від базових моделей персептрона до найсучасніших асоціативних мереж і мереж, що самоорганізуються.

Отже, ідея векторного квантування полягає в наступному [1,2]. Зображення розбивається на квадратні блоки, наприклад 2×2, 4×4 або 8×8. Кожен блок розглядається як вектор в 4-мірному, 16-мірному або 64-мірному просторі. Із цього простору вибирається обмежена кількість векторів, які утворюють кодову книгу, але так, щоб з найбільшою точністю апроксимувати вектори, які видаляються з вихідного зображення. У канал зв'язку або файл записуються номери векторів з кодової книги, які мають найменшу відстань від векторів, які вилучаються із вхідного зображення, і сама кодова книга. Оскільки векторів у кодовій книзі значно менше загальної кількості векторів у вихідному зображенні, то для представлення номера вектора витрачається менше біт, ніж для початкового вектора. За рахунок цього й досягається стискання. Ідеальним варіантом для вирішення цієї задачі є нейронні мережі, запропоновані Кохоненом.

Мережа Кохонена дозволяє виділяти схожі фрагменти даних у класи. Номер класу звичайно займає набагато менше місця в пам'яті, ніж ядро класу. Якщо передати одержувачеві всі ядра класів і номера класів, що кодують кожен фрагмент даних, то дані можуть бути відновлені. При цьому якщо число класів менше числа різних фрагментів даних, тоді неминучі втрати. Тому мережу Кохонена можна застосувати для стискання зображень із втратами.



 ISSN 1562-9945

Вихідне зображення ділиться на невеликі квадратні фрагменти, розміром декілька пікселів. Створюється шар Кохонена. Кількість входів нейронної мережі повинна збігатися з кількістю пікселів в одному квадратному фрагменті. Якщо нейронів у шарі Кохонена буде стільки, скільки різновидів фрагментів може зустрітися в зображенні, то реалізувати стискання буде не можливо, тому що отримана матриця класів буде займати стільки ж місця, скільки й вихідне зображення. Щоб застосування шару Кохонена забезпечило гарні результати стискання зображення, кількість нейронів (а значить і число класів) у ньому повинна бути менше числа можливих різновидів фрагментів.

Мережа вчиться активувати один і той самий нейрон для подібних фрагментів. Активується той нейрон, що відповідає класу, до якого віднесений даний фрагмент зображення. При навчанні мережа сама формує ядра класів, тобто набір фрагментів, з яких будується зображення. Чим більше розмір фрагментів, на які розбивається зображення, і чим менше нейронів у шарі Кохонена, тим вище ступінь стискання, але також і більші втрати при декодуванні, тобто гірше якість відновленого зображення.

Щоб порівняти ефективність програмних реалізацій розробленого алгоритму стискання зображень з використанням нейронних мереж в середовищі MATLAB із застосуванням засобів пакета Neural Networks Toolbox і мовою C, в якості тестового береться монохромне зображення розміром 100×100 пікселів. Це зображення розбивається на квадратні блоки розміром 2×2 пікселя, число нейронів при тестуванні змінюється від 2 до 1024. При оцінці ефективності різних реалізацій варто враховувати час стискання (тобто тривалість процесу кодування) і якість відновленого зображення, кількісною мірою якого є величина PSNR – пікове відношення сигналу до шуму.

У таблиці 1 представлені результати роботи алгоритму стискання зображень, реалізованого за допомогою засобів пакета Neural Networks Toolbox.

Таблиця 1

Результати тестування алгоритму стискання, реалізованого в середовищі MATLAB

| Розмір блоку | Число нейронів | Час стискання, с | Час декодування, с | PSNR | RMSE | Максимальне відхилення |
|---|---|---|---|---|---|---|
| 2x2 | 2 | 580,19 | 0,063 | 16,008 | 40,375 | 161 |
| 2x2 | 4 | 581,06 | 0,062 | 18,634 | 29,842 | 148 |
| 2x2 | 8 | 589,7 | 0,063 | 20,194 | 24,936 | 150 |
| 2x2 | 16 | 606,17 | 0,063 | 21,764 | 20,814 | 121 |
| 2x2 | 32 | 635,81 | 0,063 | 23,317 | 17,406 | 104 |
| 2x2 | 64 | 691,03 | 0,063 | 24,852 | 14,587 | 97 |
| 2x2 | 128 | 788,52 | 0,063 | 26,5 | 12,066 | 87 |
| 2x2 | 256 | 1038,4 | 0,062 | 28,468 | 9,6195 | 60 |
| 2x2 | 512 | 1356 | 0,062 | 30,002 | 8,0617 | 68 |
| 2x2 | 1024 | 2011,5 | 0,047 | 29,115 | 8,9292 | 71 |

У таблиці 2 представлені результати роботи алгоритму стискання зображень, реалізованого за допомогою функції, що моделює роботу шару Кохонена, яка написана мовою С.

Таблиця 2

Результати тестування алгоритму стискання, реалізованого мовою С

| Розмір блоку | Число нейронів | Час стискання, с | Час декодування, с | PSNR | RMSE | Максимальне відхилення |
|---|---|---|---|---|---|---|
| 2x2 | 2 | 1,484 | 0,016 | 16,029 | 40,278 | 166 |
| 2x2 | 4 | 2,609 | 0,015 | 18,654 | 29,775 | 144 |
| 2x2 | 8 | 5,141 | 0,016 | 20,255 | 24,763 | 149 |
| 2x2 | 16 | 10,469 | 0,015 | 21,764 | 20,812 | 113 |
| 2x2 | 32 | 21,187 | 0,016 | 23,292 | 17,456 | 93 |
| 2x2 | 64 | 42,078 | 0,016 | 24,802 | 14,67 | 87 |
| 2x2 | 128 | 83,032 | 0,015 | 26,522 | 12,034 | 79 |
| 2x2 | 256 | 164,97 | 0,015 | 28,354 | 9,7459 | 70 |
| 2x2 | 512 | 329,17 | 0,016 | 30,351 | 7,7442 | 63 |
| 2x2 | 1024 | 659,19 | 0,016 | 32,352 | 6,151 | 54 |

Графік залежності часу стискання від числа нейронів для різних реалізацій алгоритму представлений на рис. 1. Як видно із

 ISSN 1562-9945

графіка на рис. 1, реалізація алгоритму стискання зображень мовою С є більш ефективною в плані швидкодії.

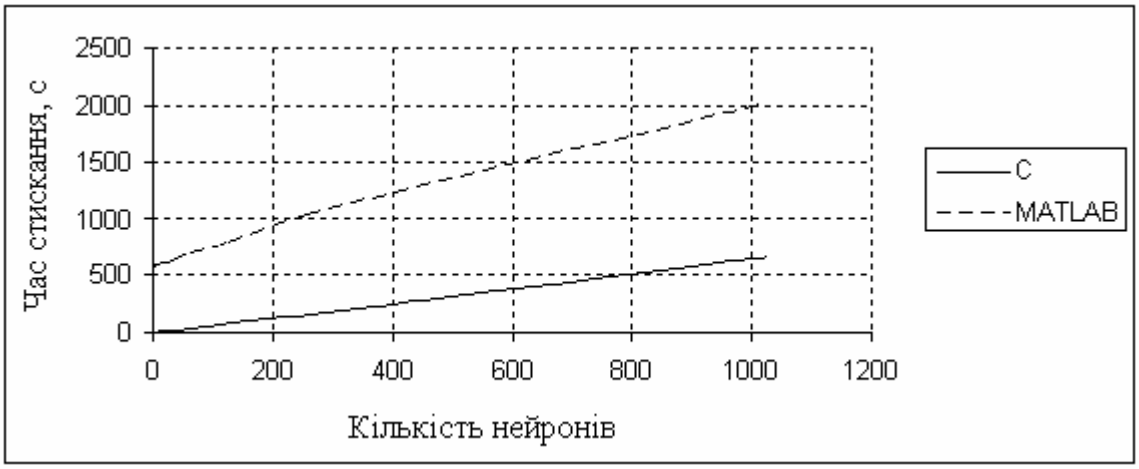


Рисунок 1 - Залежність часу стискання від числа нейронів

При однакових параметрах стискання (розмір блоку й число нейронів) швидкість процесу кодування в цьому випадку в 3 - 300 разів більше аналогічної характеристики при використанні готових функцій MATLAB для роботи з нейронними мережами. При цьому, чим менше число нейронів використовується, тим більше різниця у швидкодії. На рисунку 2 представлений графік залежності якості відновленого зображення (PSNR) від числа нейронів.


Рисунок 2 - Залежність якості зображення від числа нейронів

Як видно із графіка на рисунку 2, при використанні до 256 нейронів включно обидві реалізації алгоритму демонструють приблизно однакові результати якості відновленого зображення.

Проте для 512 нейронів і більше реалізація мовою С показує кращі результати.

Таким чином, на підставі отриманих результатів можна зробити висновок про те, що реалізація алгоритму стискання зображень за допомогою нейронних мереж мовою С є більш ефективною, оскільки в цьому випадку забезпечується більш швидке кодування (в 4 - 10 разів для найбільш оптимальної кількості нейронів), а якість відновленого зображення при певних параметрах стискання виявляється більш високою. Реалізація в середовищі MATLAB з використанням готових функцій для роботи із нейронними мережами з пакета Neural Networks Toolbox вимагає значно більшого часу на виконання процесу ущільнення, тому для проведення досліджень використовувалася реалізація мовою С.

## ЛІТЕРАТУРА

1. Сэломон Д. Сжатие данных, изображений и звука. – М: Техносфера, 2004. – 368с.
2. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с.

Одержано 15.11.2008р.

 ISSN 1562-9945