

Лабораторная работа №1

Тема: Реализация линейной рециркуляционной сети как модели автодекодера для задачи понижения размерности данных.

Цель: Ознакомиться, проанализировать и получить навыки реализации модели линейной рециркуляционной сети для задачи сжатия графической информации (или другой информации).

Для лабораторной работы в качестве данных исходного растрового изображения можно использовать файлы следующих форматов:

- BMP;
- TIFF;
- PNG;
- JPG.

Можно использовать как цветное, так и чёрно-белое изображение, где значения интенсивности цвет пиксела задаются R, G, B на множестве шкал S (например $S = \{R, G, B\}$). Исходное изображение hw (рекомендуемый размер – не менее 256256 (или картинка с таким же числом пикселей)) разбивается на L прямоугольников размерностью nm ($n*m = N/|S|$) ($n \leq h, m \leq w$), где значения n, m может указать пользователь. Все прямоугольники должны полностью покрывать исходное изображение и не выходить за его границы, может допускаться перекрытие прямоугольников, которое может задать пользователь.

Пусть C_i обозначает одно из значений R, G, B ($i \in S$) и принимает значения в диапазоне $0 \leq C_i \leq 255 = C_{\max}$. Для того, чтобы использовать значения цвета пикселя исходного изображения для дальнейшей обработки, следует выполнить преобразование:

$$c_i(jk) = (2 * C_i(jk) / C_{\max}) - 1, \quad (1)$$

где $c_i(jk)$ – это значение цвета пикселя для дальнейших преобразований (jk – координаты пикселя).

Так преобразуются все значения пикселей для всех прямоугольников. Значения цвета пикселей из прямоугольника q собираются в свой (эталонный) вектор $X_0(q)$, компоненты которого вычисляются в соответствии с выражением $X_0(q)_{i+|S|*(j+k*n)} = c_i(q)(jk)$, где $j \leq n, k \leq m$. Т. е. число элементов $X_0(q)$ будет равно N , где $X_0(q)$ – это вектор размера $1N$, а его значения после преобразований уже будут $|X_0(q)| \leq 1$.

Пример 1:

$$c(q) = \begin{bmatrix} 0.1 & 0.5 & 0.2 \\ 0.3 & -0.2 & 0.6 \\ 0.7 & -0.7 & 0.9 \end{bmatrix},$$

тогда $X_0(q) = [0.1, 0.5, 0.2, 0.3, -0.2, 0.6, 0.7, -0.7, 0.9]$.

Далее следует использовать матрицу весов W для первого слоя сети размерностью Np , где p – это заданное пользователем число нейронов второго слоя (как правило $Z(p, N, L) < 1$). Значения весов должны быть $|W_{ij}|$

1. Матрица весов на втором слое $W' = [W]^T$.

Производится обучение последовательно на выборке из L эталонных образов длиной N . Принимается $X(i) = X_0(i)$. Следует произвести следующие преобразования:

$$Y(i) = X(i) * W \quad (2)$$

$$X'(i) = Y(i) * W' \quad (3)$$

$$\Delta X(i) = X'(i) - X(i) \quad (4)$$

Для корректировки весов на втором слое следует вычислить значения согласно формуле:

$$W'(t+1) = W'(t) - \eta * [Y(i)]^T * \Delta X(i), \quad (5)$$

где η – коэффициент обучения, указываемый пользователем либо вычисляемый согласно выражению для расчета адаптивного шага:

$$\eta = 1 / (Y(i) * Y(i)^T),$$

Для обучения нейронов первого слоя следует вычислить:

$$W(t+1) = W(t) - \eta * [X(i)]^T * \Delta X(i) * [W'(t)]^T, \quad (6)$$

где η – коэффициент обучения, указываемый пользователем ($\eta = \eta'$) либо вычисляемый согласно выражению для расчета адаптивного шага:

$$\eta' = 1 / (X(i) * X(i)^T)^2,$$

Нормализация может быть осуществлена согласно выражениям:

$$W_{ij} = W_{ij} / |[W]|^T_j,$$

$$W'_{ji} = W'_{ji} / |[W']^T_i|.$$

Максимальную допустимую ошибку ϵ задаёт пользователь, где $0 < \epsilon < 0.1 * p$ (p – количество элементов). Суммарная среднеквадратическая ошибка для прямоугольника q в процессе обучения высчитывается по формуле:

$$E(q) = \sum \Delta X(q)_i * \Delta X(q)_i, \text{ где } 1 \leq i \leq N \quad (7)$$

Суммарная ошибка для обучающей выборки равна:

$$E = \sum E(q), \text{ где } 1 \leq q \leq L \quad (8)$$

$E(i+1)$ не вычисляется до тех пор, пока не произойдёт вычисление $E(i)$. Т.е. вычисление по формулам (5) – (7) происходит L раз.

Если $E > \epsilon$, тогда повторять действия, описанные в формулах (2) – (8), до тех пор, пока $E \leq \epsilon$.

Далее следует производить вычисления (сжатие и восстановление) для каждого эталонного прямоугольника по формулам (2) и (3), где $X(i) = X_0(i)$, а W и W' – веса, полученные в результате обучения. После восстанавливаются значения пикселей по формуле обратной формуле (1):

$$u_k = C_{\max} * (X'(i)_k + 1) / 2, \quad (9)$$

где u_k – значение пикселя для восстановленного изображения.

После восстановления пикселей собирается изображение из L элементов. Для этого производятся преобразования обратные тем, которые приведены в Примере 1. Полученное изображение выводится на экран. Так же требуется вывести результаты следующих вычислений:

- достигнутая ошибка при обучении;
- матрица весов на первом слое (для отладки);
- матрица весов на втором слое (для отладки);
- количество пройденных обучаемых шагов.

Коэффициент сжатия Z , при условии представления с одинаковой точностью всех числовых параметров:

$$Z = (N * L) / ((N + L) * p + 2), \quad (10)$$

В отчёте привести графики и таблицы следующих зависимостей:

- числа итераций обучения от коэффициента сжатия Z (для фиксированного изображения и параметров);
- числа итераций обучения для разных изображений (для фиксированных параметров и Z);
- числа итераций от ϵ (остальные параметры фиксированы);
- числа итераций от η (остальные параметры фиксированы).

Задания:

1. Реализовать модель линейной рециркуляционной сети.
2. Реализовать модель линейной рециркуляционной сети с адаптивным шагом обучения.
3. Реализовать модель линейной рециркуляционной сети с нормированными весами.
4. Реализовать модель линейной рециркуляционной сети с адаптивным шагом обучения с нормированными весами.

Литература:

1. Головкин В.А. Нейроинтеллект: теория и применение. Книга 1. Организация и обучение нейронных сетей с прямыми и обратными связями. Брест, - БПИ; 1999.
2. Головкин В.А. Нейроинтеллект: теория и применение. Книга 2. Самоорганизация, отказоустойчивость и применение нейронных сетей. Брест, - БПИ; 1999.
3. Осоский С. Нейронные сети для обработки информации. М., - “Финансы и статистика”; 2002.