

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 $h \times w$ (рекомендуемый размер – не менее 256×256 (или картинка с таким же числом пикселей)) разбивается на **L** прямоугольников размерностью $n \times m$ ($n * m = N/|S|$) ($n \leq h$, $m \leq w$), 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 пиксела 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 пиксела 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 $X_0(q)_{i+|S|(j+k*n)} = c(q)_i(jk)$, где целые неотрицательные $j < n$, $k < m$. Т. е. число элементов $X_0(q)$ будет равно **N**, где $X_0(q)$ – это вектор размера $1 \times N$, а его значения после преобразований уже будут $|X_0(q)_k| \leq 1$.

Пример 1:

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

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 \mathbf{W} для первого слоя сети размерностью $\mathbf{N} \times \mathbf{p}$, где \mathbf{p} – это заданное пользователем число нейронов второго слоя (как правило такое, что $\mathbf{Z}(\mathbf{p}, \mathbf{N}, \mathbf{L}) > 1$). Значения весов должны быть значениями равномерно распределённой случайной величины $|\mathbf{W}_{ij}| \leq 1$. Матрица весов на втором слое \mathbf{W}' .

Производится обучение последовательно на выборке из \mathbf{L} эталонных образов длиной \mathbf{N} . Принимается $\mathbf{X}(\mathbf{i}) = \mathbf{X}_0(\mathbf{i})$. Следует произвести следующие преобразования:

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

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

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

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

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

где $0 < \alpha' \leq 0.01$ – коэффициент обучения, указываемый пользователем либо вычисляемый согласно выражению для расчета адаптивного шага:

$$\alpha' = 1/(\mathbf{Y}(\mathbf{i}) * \mathbf{Y}(\mathbf{i})^T),$$

Для обучения нейронов первого слоя следует вычислить:

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

где $0 < \alpha \leq 0.01$ – коэффициент обучения, указываемый пользователем ($\alpha = \alpha'$) либо вычисляемый согласно выражению для расчета адаптивного шага:

$$\alpha = 1/(\mathbf{X}(\mathbf{i}) * \mathbf{X}(\mathbf{i})^T)^2,$$

Нормализация может быть осуществлена согласно выражениям:

$$\mathbf{W}_{ij} = \mathbf{W}_{ij} / \|\mathbf{W}\|^T j|,$$

$$\mathbf{W}'_{ji} = \mathbf{W}'_{ji} / \|\mathbf{W}'\|^T i|.$$

Максимальную допустимую ошибку \mathbf{e} задаёт пользователь, где $0 < \mathbf{e} \leq \mathbf{k} * \mathbf{p}$ ($\alpha \leq \mathbf{e}$). Суммарная среднеквадратическая ошибка для прямоугольника \mathbf{q} в процессе обучения высчитывается по формуле:

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

Суммарная ошибка для обучающей выборки равна:

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

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

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

Далее следует производить вычисления (сжатие и восстановление) для каждого эталонного прямоугольника по формулам (2) и (3), где $\mathbf{X}(\mathbf{i}) = \mathbf{X}_0(\mathbf{i})$, а \mathbf{W} и \mathbf{W}' – веса, полученные в результате обучения. После восстанавливаются значения пикселей по формуле обратной формуле (1):

$$\mathbf{u}_k = \mathbf{C}_{\max} * (\mathbf{X}'(\mathbf{i})_k + 1) / 2, \quad (9)$$

где \mathbf{u}_k – значение пикселя для восстановленного изображения.

После восстановления пикселей собирается изображение из \mathbf{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 $Z(p, N, L)$, при условии представления с одинаковой точностью всех числовых параметров:

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

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