

Лабораторный практикум по дисциплине «Организация вычислительных систем»

Лабораторная работа №1

«Проектирование функциональной модели нейронной сети»

Задачи

1. Разработать функциональную модель работы нейронной сети, решающей задачу распознавания геометрических фигур на изображениях, представленных на рисунке 1.
2. Рассмотреть несколько структур сетей и выбрать подходящую для решения задачи, путем обучения сети методом обратного распространения ошибки. Выбирается сеть, имеющая наилучшие обобщающие свойства.
3. Для выбранной структуры сети оценить:
 - 3.1. Время обучения.
 - 3.2. Количество эпох обучения.
 - 3.3. Время вычисления выхода сети.
 - 3.4. Количество требуемой памяти в процессе вычисления выхода и для процесса обучения.
4. Сформировать отчет по работе, в котором описать:
 - 4.1. Структуру выбранной сети.
 - 4.2. Свойства сети: время обучения, количество эпох обучения, время вычисления выхода, количество требуемой памяти.
 - 4.3. Тестовую выборку, используемую для обучения.
 - 4.4. Пример работы сети на выборке изображений, которые распознаются корректно и выборке, на которой сеть дает сбой.

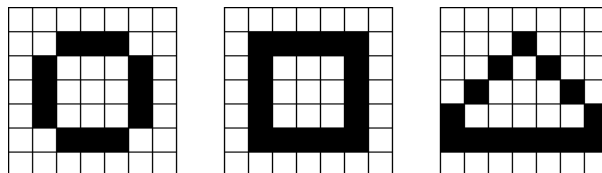


Рисунок 1 – Изображения с фигурами для распознавания

Требования к работе

1. В качестве типа нейронной сети рекомендуется использовать многослойную полносвязную сеть прямого распространения с сигмоидальной функцией активации:

$$f(x) = \frac{1}{1 + e^{-x}}$$

2. Реализуемая нейронная сеть должна иметь три выхода – по количеству распознаваемых фигур. Выход изменяется от 0 до 1, в соответствии с диапазоном выходных значений сигмоидальной функции активации, и показывает вероятность того, что на входе изображения круга, квадрата или треугольника. При однозначном распознавании фигуры значение 1 формируется только на одном выходе, отвечающего за распознавания фигуры данного типа.

3. Реализуемая сеть должна иметь не менее двух слоев. Она обязательно должна содержать один скрытый и выходной слой.
4. Для функциональной модели должно быть возможным задание в виде параметров следующих характеристик:
 - 4.1. Количество входов.
 - 4.2. Количество слоев.
 - 4.3. Количество нейронов в слоях.
5. Функциональная модель может быть написана на любом языке программирования (C/C++, python и др.). Рекомендуется использовать C++.
6. На вход сети должны подаваться черно-белые изображения, представленные в виде битовой карты 7x7 пикселей (см. рисунок 1). Значение каждого пикселя входного изображения равно либо 0, либо 1.
7. Сеть должна правильно распознавать искаженные изображения, пример которых представлен на рисунке 2.

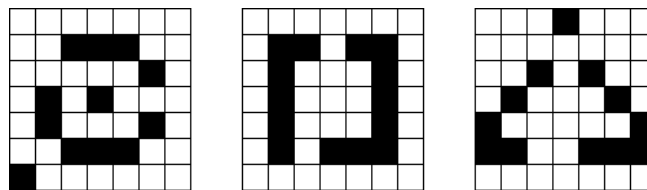


Рисунок 2 – Искаженные изображения с фигурами

Методические указания к работе

В рамках работы требуется реализовать многослойную нейронную сеть, способную распознавать геометрические фигуры на изображениях. Пример работы сети представлен на рисунках 3 и 4.

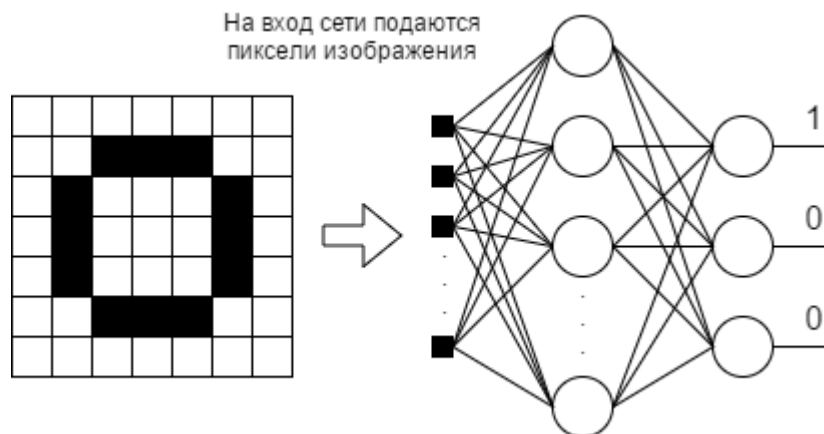


Рисунок 3 – Пример работы сети при подаче на вход четкого образа фигуры

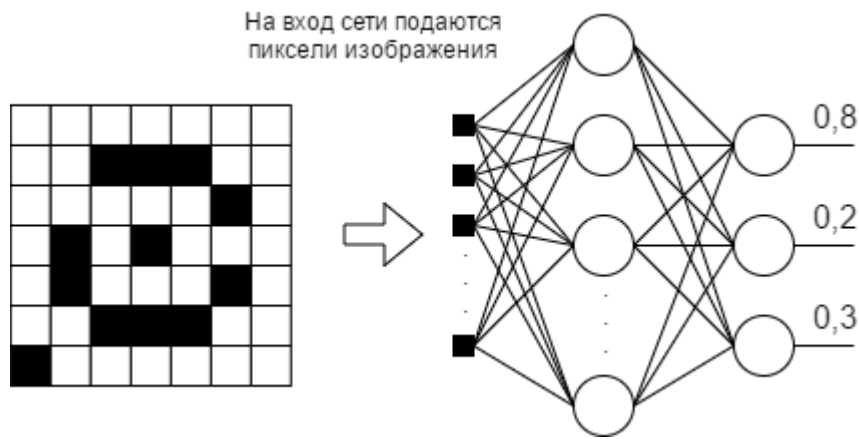


Рисунок 4 – Пример работы сети при подаче на вход искаженного образа фигуры

Каждый вход сети имеет вес w_{ij}^k , где k – номер слоя, i – номер нейрона в слое, j – номер входа нейрона. В процессе работы выход сети выход каждого нейрона вычисляется по формуле:

$$y_i^k = f\left(\sum_{j=1}^{n(k-1)} y_j^{k-1} w_{ij}^k\right)$$

В формуле:

- y_i^k – выход нейрона i слоя k ;
- $n(k-1)$ – количество нейронов в слое с номером $(k-1)$;
- $f(\dots)$ – сигмоидальная функция активации.

При этом значения y_i^0 равны входам сети.

Процесс обучения сети включает следующие шаги:

1. Вычисление выхода сети.
2. Вычисление ошибки функционирования по формуле

$$E = \frac{1}{2} \sum_i^{n(N)} (t_i - y_i^N)$$

В формуле:

- t_i – ожидаемое значение выхода i сети;
 - N – номер последнего слоя сети.
3. Если ошибка больше заданного значения, например 0.1, то происходит подстройка весов по алгоритму обратного распространения ошибки. Алгоритм вычисления значения новых весов представлен ниже.
 4. Пункты 1 – 3 повторяются на всех тестовых наборах входных данных. Проход по всем тестовым наборам называется эпохой обучения.
 5. Обучение останавливается, когда на всех тестовых наборах ошибка на выходе не превышает заданного значения.

Подстройка весов производится по формуле:

$$w_{new\,ij}^k = w_{ij}^k + \Delta w_{ij}^k$$

$$\Delta w_{ij}^k = \alpha \delta_i^k y_j^{k-1}$$

Значение α – выбирается эмпирически в диапазоне от 0 до 1 и влияет на скорость минимизации ошибки обучения. Для предотвращения переобучения сети рекомендуется использовать значения от 0.1 до 0.3. Для улучшения качества обучения возможно динамически подстраивать значение α , выбирая на начальных этапах большое значение, а затем его минимизируя при уменьшении ошибки на выходе. Тем самым возможно организовать более «точную» подстройку весов сети.

Значение δ_i^k вычисляется по-разному для выходного слоя и внутренних слоев сети.

Для выходного слоя:

$$\delta_i^N = y_i^N (1 - y_i^N) (t_i - y_i^N)$$

Для внутренних слоев сети:

$$\delta_i^k = y_i^k (1 - y_i^k) \sum_{j=1}^{n(k+1)} \delta_j^{k+1} w_{ji}^{k+1}$$