**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего профессионального образования**

**Московский технический университет связи и информатики**

Факультет повышения квалификации

**Лабораторная работа №2**

Выполнил:

магистрант гр. 2МИБ1501

Иванов И.И.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись дата

Проверил:

к.т.н. Махров С.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись дата

**Москва 2016**

**Обучение однослойного персептрона методом стохастического градиентного спуска**

**Цель работы**. Изучить алгоритм обучения однослойного персептрона методом стохастического градиентного спуска.

**Задание.** Построить и обучить нейронную сеть для распознавания цифровых рукописных символов из базы данных MNIST (Mixed National Institute of Standards and Technology database). Нейронная сеть должна корректно распознавать образы из тестовой выборки в большинстве случаев. Общий процент ошибки распознавания образов не должен быть выше 20%.

**Теоретические сведения.**

***Алгоритм обучения методом стохастического градиентного спуска***

Шаг 1. Подготовить обучающую выборку, каждый элемент которой будет состоять из пар *(X, D)m (m=1,…q)* – обучающего вектора *X = (x1,…,xn)* (*i=1,…,n)* с вектором желаемых значений *D = (d1,…,dk) (j=1,…,k*) выходов персептрона.

Шаг 2. Генератором случайных чисел всем синаптическим весам *wi,j* и нейронным смещениям *w0,j* (*i=1,…,n; j=1,…,k*) присваиваются некоторые малые случайные значения.

Шаг 3. Из обучающей выборки *(X, D)1,…,(X, D)q*, случайным образом взять пару *(X, D)m*.

Шаг 4. Из этой выбранной пары *(X, D)m* взять вектор *Xт = (x1,…,xn)* и подать его на входы персептрона *x1,…,xn*. Сигналам нейронных входов смещения *x0*присваиваются единичные значения: *x0 = 1*.

Шаг 5. Для каждого *j*-го нейрона вычислить взвешенную сумму входных сигналов *netj* и выходной сигнал *yj* на основании функции активации *f*:

|  |  |
| --- | --- |
|  |  |
|  |  |

Шаг 6. Вычислить ошибку *ε* для текущего обучающего вектора *ε*:

|  |  |
| --- | --- |
|  |  |

где *dj* – желаемое, а *yj* - фактическое значение выхода *j*-го нейрона в соответствии с поданным *m*-ым входным вектором *Xm* из пары *(X, D)m*, *k* – количество выходов (классов) персептрона.

Шаг 7. Проверить критерий останова обучения, если он выполняется, то завершить обучение. В противном случае переход к следующему шагу.

Шаг 8. Произвести коррекцию синаптических весов *j*-го нейрона и нейронных смещений:

|  |  |
| --- | --- |
|  |  |
|  |  |

где t – номер итерации, *η* – коэффициент скорости обучения.

|  |  |
| --- | --- |
|  |  |

Шаги 5-8 выполняются последовательно для каждого входного образа, на котором обучается персептрон.

Шаги 5 и 8 повторяются для всех нейронов персептронного слоя при подаче конкретного образа.

Критерии останова алгоритма обучения могут быть следующими:

1. Значение ошибки *ε* для текущего обучающего вектора, вычисляемое на текущем случайно взятом обучающем векторе не превышает заданного заранее установленного порогового значения *εпорог*, близкого к нулю: .
2. Превышен установленный лимит количества эпох.
3. Значение общей ошибки всей сети меняется незначительно на протяжении нескольких эпох.

Указанные критерии останова обучения используются как по отдельности, так и вместе. Важным замечанием будет то, что использование только 2-го и 3-го критерий могут привести к недообучению сети, поэтому их использование без комбинации с 1-ым критерием, не рекомендуется.

Скорость обучения *η* выбирают в пределах [0.05…1.0] в зависимости от функции активации, но если скорость обучения константа, то её следует выбирать в пределах [0.0…0.01], так как при больших значениях метод градиентного спуска может расходиться и «проскакивать» области минимума, которые вычисляет антиградиент. Метод может расходиться и при таком коэффициенте скорости обучения, в зависимости от функции активации.

**Ход выполнения работы**

(Ход выполнения работы должен быть описан достаточно подробно с нарисованной структурной схемой нейронной сети. Указать параметры, при которых обучалась сеть (скорость обучения, какой критерий останова использовался, диапазон, из которого случайным образом назначались значения синаптических весов при старте сети).

**Вывод.**

(Сделать вывод об эффективности алгоритма стохастического градиентного спуска для обучения персептрона по сравнению с методом коррекции по ошибке. Указать, везде ли были корректные ответы сети? Если нет, указать почему? Если ответы некорректные, то можно попробовать задать более низкую скорость обучения, генерировать при старте нейронной сети синаптические веса не из дипазона [-0.03, 0.03], а из [-0.003, 0.003] или [-0.0003, 0.0003] и т.д. Указать количество эпох, за которое сеть обучилась)

**Исходный код программы**

(Привести исходный код каждого java-класса)