**Задача: единичный персептрон**

**Описание**

Реализация данной задачи показывает работу единичного персептрона и его обучение с помощью метода обратного распространения ошибки. В качестве примера представим, что персептрон принимает на вход 2 числа, на выходе получается их сумма. Необходимо построить и обучить такую нейронную сеть.

**Решение**

Нейросеть будет состоять из одного слоя с линейной функцией активации и коэффициентом слоя .

* 0 слой = 2 входа
* 1 слой = 1 нейрон

Функция активации:

Для обучения используется 20 примеров, ещё 10 будет использоваться для валидации результатов.

Для обучения нейросети использовались следующие параметры:

* Метод инициализации – Нгуен-Видроу (InitG)
* Вид целевой функции – Разность квадратов (MSE)
* Метод обучения – Метод обратного распространения ошибки
* Использование регуляризации (уровень = 0.00008)
* Использование инерции (уровень = 0.025)
* Количество эпох обучения = 10000
* Минимальная ошибка обучения = 6.0e-06
* Использование случайного перемешивания обучающих примеров

Нейросеть обучилась за 179 эпох. В ходе обучения была получена следующая матрицы весов:

|  |  |  |
| --- | --- | --- |
| **w1** | **w2** | **bias** |
| 1.000002 | 1.000001 | -0.000042 |

Идеальные весовые коэффициенты для этой задачи при этом следующие:

|  |  |  |
| --- | --- | --- |
| **w1** | **w2** | **bias** |
| 1.0 | 1.0 | 0.0 |

Коэффициенты подобрались почти точно. На выходе нейросети получаем сумму двух чисел. На последней итерации обучения была достигнута следующая энергия:

* Суммовая = 1.22851e-08
* Средняя = 5.5419e-06
* Максимальная = 3.11899e-05
* Регуляризация = 1.0

После обучения на тренировочном множестве получаем следующую картину:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вход | | Выход | | |  |
| x | y | output | outrun | outpostrun | energy |
| 0 | 1 | 1 | 0.99996 | 0.99996 | 8.405E-10 |
| 1 | 0 | 1 | 0.99996 | 0.99996 | 8E-10 |
| 8 | 12 | 20 | 19.99999 | 19.99999 | 1.125E-10 |
| 12 | 8 | 20 | 19.99999 | 19.99999 | 8.45E-11 |
| -6 | 6 | 0 | -0.00004 | -0.00004 | 9.68E-10 |
| 6 | -6 | 0 | -0.00004 | -0.00004 | 7.605E-10 |
| -2 | 5 | 3 | 2.99996 | 2.99996 | 7.605E-10 |
| 5 | -2 | 3 | 2.99996 | 2.99996 | 6.48E-10 |
| 7 | -5 | 2 | 1.99996 | 1.99996 | 6.845E-10 |
| -5 | 7 | 2 | 1.99996 | 1.99996 | 8.405E-10 |
| 2.5 | 5 | 7.5 | 7.49997 | 7.49997 | 5.12E-10 |
| 5 | 2.5 | 7.5 | 7.49997 | 7.49997 | 4.805E-10 |
| 3 | 2 | 5 | 4.99997 | 4.99997 | 6.125E-10 |
| 2 | 3 | 5 | 4.99997 | 4.99997 | 6.125E-10 |
| 4 | 2 | 6 | 5.99997 | 5.99997 | 5.445E-10 |
| 2 | 4 | 6 | 5.99997 | 5.99997 | 5.78E-10 |
| 0 | 8 | 8 | 7.99997 | 7.99997 | 5.12E-10 |
| 8 | 0 | 8 | 7.99997 | 7.99997 | 4.5E-10 |
| -1 | 4 | 3 | 2.99996 | 2.99996 | 7.22E-10 |
| 4 | -1 | 3 | 2.99996 | 2.99996 | 6.845E-10 |

Теперь посмотрим работу нейросети на валидационном множестве:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вход | | Выход | | |  |
| x | y | output | outrun | outpostrun | energy |
| 9 | 11 | 20 | 19.99999 | 19.99999 | 1.125E-10 |
| 11 | 9 | 20 | 19.99999 | 19.99999 | 8.45E-11 |
| 7 | 7.5 | 14.5 | 14.49998 | 14.49998 | 2.42E-10 |
| 7.5 | 7 | 14.5 | 14.49998 | 14.49998 | 2.42E-10 |
| 2 | 9 | 11 | 10.99997 | 10.99997 | 4.205E-10 |
| 9 | 2 | 11 | 10.99998 | 10.99998 | 3.125E-10 |
| 8 | -5 | 3 | 2.99997 | 2.99997 | 6.125E-10 |
| -5 | 8 | 3 | 2.99996 | 2.99996 | 8E-10 |
| 6 | 7 | 13 | 12.99998 | 12.99998 | 2.88E-10 |
| 7 | 6 | 13 | 12.99998 | 12.99998 | 2.88E-10 |

После обучения нейросеть справляется с заданной задачей нахождения суммы 2 чисел с заданной погрешностью.