

**Что происходит с тензорами, если при применении к ним какой-либо операции размеры не совпадают?**

При работе с тензорами numpy сравнивает их формы. Два измерения совместимы, когда они равны или один из них тензор 1 ранга, при чем количество его элементов должно соответствовать количеству элементов в строке (или столбце, если используем метод reshape). Пример сложения матрицы размера (3,3) с вектором с 3 элементами и матрицей (3,2) представлен на рис. 1.

```
X
[[ 1  2  3]
 [ 5  6  7]
 [ 8  9 10]]
```

```
Y
[-1 -2 -3]
```

```
Z
[[-1 -2]
 [-5 -6]
 [ 9 10]]
```

```
X+Y
[[0 0 0]
 [4 4 4]
 [7 7 7]]
```

```
X+Z
```

```
Traceback (most recent call last):
```

```
File "C:/lr3/main.py", line 20, in <module>
```

```
    print(x+z)
```

```
ValueError: operands could not be broadcast together with shapes (3,3) (3,2)
```

Рисунок 1 – Пример сложения тензоров разных размерностей.

**В чем разница методов оптимизации второго порядка от методов оптимизации первого?**

Оптимизаторы второго порядка используют гессиан – матрицу вторых производных, а методы первого порядка используют градиент.

**Для чего используются проверку по k-блокам?**

Этот метод используется, когда качество модели слишком сильно зависит от деления данных на тренировочный и контрольный наборы. Например, если у нас небольшой набор данных, то проверочный набор может

получиться очень маленьким, как следствие, оценки при проверке могут сильно меняться в зависимости от того какие данные попадут в проверочный и обучающий наборы.

**Чем обосновано использование  $\text{mae}$  в качестве метрики в данной л.р.?**

В данной лабораторной работе мы решаем задачу регрессии. Понятие точности неприменимо для регрессии, поэтому для оценки качества часто применяется MAE – абсолютное значение разности между предсказанными и целевыми значениями.

**Для чего нужен участок кода, находящийся на 22-29 строчках?**

Если передать в нейронную сеть значения, имеющие самые разные диапазоны, то она, конечно, сможет автоматически адаптироваться к таким разнородным данным, но это усложнит обучение. На практике к таким данным принято применять нормализацию, в результате признак центрируется по нулевому значению и имеет стандартное отклонение, равное единице. В строчках 22-29 происходит нормализация данных.

**На основании каких выборок идет проверка по  $k$ -блокам?**

Если я правильно понял вопрос, то в перекрестной проверке по  $K$  блокам данные делятся на  $K$  блоков, создается  $K$  идентичных моделей, который обучаются на  $K-1$  блоках с оценкой по оставшемуся блоку. По полученным  $K$  оценкам вычисляется среднее значение, которое принимается как оценка модели.



Рисунок 2 – Перекрестная проверка по 3 блокам

**Почему значение параметра `batch_size` равно 1 в функции `fit` на 45 строке?**

Вообще мы можем увеличить `batch_size`, но у нас небольшой набор данных, поэтому прогон с таким размером несильно увеличит время работы, при этом с маленьким `batch_size` хорошо работает оптимизатор `rmsprop`. Так же можно сказать, что для маленьких пакетов приводит к более быстрой сходимости.