1. **Что такое L2-регуляризация, какой геометрический смысл регуляризации?**

Чтобы бороться с переобучением модели, стремятся избегать больших весов (по сравнению с остальными). Поэтому для весов вводится т.н. штраф, добавляемый к функции потерь, за счет чего экстремально большие веса уменьшаются в процессе обучения. Суть L2-регуляризации заключается в добавлении к функции потерь величины, пропорциональной сумме квадратов весов. Экстремально большие веса вносят основной вклад в увеличение значения функции потерь, поэтому они уменьшаются в процессе обучения. Геометрический смысл состоит в том, что искомая функция (для задачи регрессии) сглаживается, избавляется от перегибов, связанных с излишней сложностью, вызванной переобучением.

2. **Как происходит скалярное произведение тензоров разных рангов?**

Скалярное произведение векторов разных рангов может быть получено только если последняя размерность левого тензора совпадает с первой размерностью правого, то есть если их формы соответственно (s_{11}, …, s_{1m}, n) и (n, s_{21}, …, s_{2k}). Форма результирующего тензора будет (s_{11}, …, s_{1m}, s_{21}, …, $s_{2k)}$. m и k могут не совпадать, то есть тензоры могут быть разных рангов.

3. **Почему анализируется mae,а не mse?**

Во-первых, mae нагляднее, потому что это средний модуль ошибки, и он показывает среднее отклонение стоимости от ее прогноза. Во-вторых, так как квадратичная функция возрастает быстрее линейной, средний квадрат (mse) более чувствителен к скачкам, чем mae. Таким образом, если использовать mse мы получим график, похожий на mae, но отличающийся более резкими скачками и более острыми перегибами, что тоже делает mae нагляднее.

4. **Сколько весов в Вашей модели?**

```
model.add(Dense(64, activation='relu', input_shape=shape))
model.add(Dense(64, activation='relu'))
model.add(Dense(1))
# input_shape = (13)
```

13 * 64 весов между входным и первым внутренним слоем,

64 * 64 весов между внутренними слоями,

64 * 1 весов между вторым внутренним и выходным слоями --

Всего получается 4992.

5. Почему не происходит начального перемешивания данных?

Начальное перемешивание не решает проблему выявления сетью закономерностей в повторяющемся порядке образцов, потому что на первой эпохе порядок не может быть таким же, как на предыдущей, так как это первая эпоха. Можно перемешивать данные между эпохами. K-fold частично решает проблему повторяющегося порядка, так как K моделей обучаются на данных, которые идут в разном порядке, и оценка ошибки по всем моделям усредняется. Но лучше всего было бы для каждого блока между эпохами перемешивать данные.