Лучшим решением (private 0.44384) оказалось усреднение предсказаний 20 моделей. Я не стал присылать все 20. Добавил один ноутбук с одной из лучших соло-моделей (private 0.45519). А также несколько ноутбуков с различными идеями, которые пробовал, но они давали либо слабое улучшение, либо не давали его вообще.

Основная модель

Признаки features.ipynb

Для каждого игрока в каждой игре вычислялись среднее/медиана/макс/мин значения рейтинга в предыдущий/следующий/текущий моменты времени(X21). А также расстояние до следующей/предыдущей игры. Если следующей/предыдущей игры не было, то подставлялось текущее значение.

Только эти признаки в логреге без тюнинга давали ~0.55.

Модель the_best.ipynb

Модель представляет собой две полносвязные нейронные сети, обрабатывающие признаки из features.ipynb в одной части и фигуры без учета их порядка в другой, после чего результаты объединяются и подаются на вход еще одной полносвязной сети. Все это обучается end-to-end. Такой подход давал результат лучше, чем если бы на первых слоях были бы связи у признаков, связанных с фигурами и с рейтингом.

Дополнительные подходы

Учет того, что порядок игроков не важен. Тут было два подхода.

Первый просто увеличить датасет в два раза переставив игроков местами и поменяв результат на обратный, прироста это не давало, ноутбук приводить не стал т.к. преобразование тривиальное.

Второй подход состоял в том, чтобы результат самой модели минимально зависел от порядка игроков. Для этого были созданы фрагменты сети, которые имели общие веса для обработки фигур/фичей из features.ipynb у обоих игроков. После чего их результаты объединялись операцией, которая бы сохраняла результат с точность до изменения знака на противоположный при перестановке и после направлялось в полносвязную сеть см. same_weights.ipynb. (private 0.47057)

Учет порядка фигур.

В архитектуру аналогичную the_best.ipynb был добавлен эмбеддинг слой (общий для обоих игроков), после которого результаты также подавались в полносвязный слой, но уже с учетом порядка (см. add_order.ipynb)

Учет опыта игрока и героев

Т.к. в задаче было сказано, что игроки прокачивают юнитов, то я решил попробовать это учесть. Я для каждого игрока добавил его опыт (количество игр до) и опыт юнитов 1+log(количество игр до этим юнитом у игрока). Это давало совсем минимальное преимущество, которое все скорее было случайностью. (см. same_weights_exp.ipynb) (private 0.46976)

Другой подсчет рейтинга следующей игры

Пробовал считать, что следующая игра — это либо с тем же значением X21, либо со следующим, плюс добавлять флаги что таких не нашлось или нашлась только одна, но такой подходи не дал заметного прироста и я не стал добавлять ноутбуки по нему