

Рисунок 3.3 – Обучающая кривая за 20 эпох

Примечание – Источник: собственная разработка.

Тут может возникнуть резонный вопрос: почему с 8 по 10ю эпохи модель ухудшает свои результаты, хотя алгоритм обратного распространения ошибки однозначно всегда должен быть направлен в сторону наискорейшего спада целевой функции? Это сыграла свою роль техника обучения модели с использованием разбиения на батчи – целевую функцию мы вычисляем на полных данных, но при использовании некоторых батчей для определения градиента алгоритм «не видел» локального минимума возле которого находился в момент 9ой эпохи, в результате успешно пропустил этот минимум продолжил оптимизацию. По завершению оптимизации были снят AUC модели, он представлен в таблице М.1 в строке соответствующей 20 эпохам.

Как видно, по рисунку оптимизация могла быть успешно продолжена – последние эпохи наблюдалось стремительное улучшение целевой функции. Потому, попробуем доучить модель еще на 30 эпох. Для воплощения этого в жизнь достаточно лишь еще раз вызвать метод «fit» ранее созданного «day1_trainer», указав в аргументе «epochs» требуемое число эпох

Кривые обучения за 50 полученных таким образом эпох представлены на рисунке 3.4.

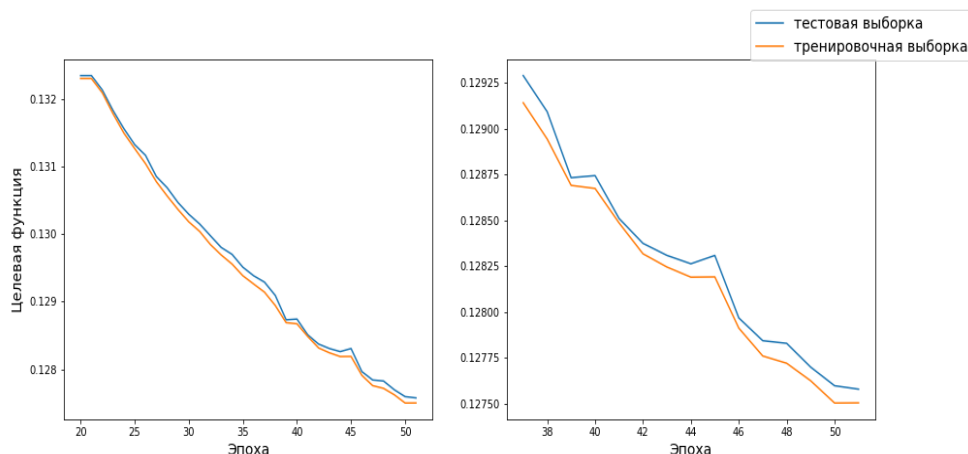


Рисунок 3.4 – Обучающая кривая за 50 эпох

Примечание – Источник: собственная разработка.

Метрика AUC модели после этого шага представлена в таблице М.1 в строке соответствующей 50-ти эпохам.

Как и ожидалось, оптимизация была продолжена, но обратим внимание, что на более поздних этапах целевая функция не слишком плавно начинает убывать, появляются флуктуации. Это следствие того, что иногда делаются слишком большие шаги не гарантирующие уменьшение ошибки. Для решения этой проблемы можно понизить параметр `learning rate`. Переопределив оптимизатор с параметром «lr» равным 0,0001, проведем еще 50 эпох обучения модели. Данный шаг находит отражение в рисунке 3.4. А характеристика AUC указана с строке соответствующей сотне эпох таблицы М.1.

Как видно, это помогло добиться более плавной оптимизации, но вместе с тем после 50-й эпохи скорость оптимизации заметно падает. Вне описания этой работы еще лежит необходимость экономии вычислений – важная часть работы инженера по машинному обучению связанная с экономией ресурсов затрачиваемых на вычисления. К сожалению, современная вычислительная техника не всегда позволяет просчитать все варианты, потребуется недопустимо много времени, электроэнергии или денег на аренду вычислительных мощностей на стороне. Сейчас мы столкнулись с одним из примеров, действительно используя меньший `learning rate` у нас больше шансов подобраться к минимуму более точно, но нельзя забывать, что в каждый шаг вложена достаточно сложная вычислительная процедура.

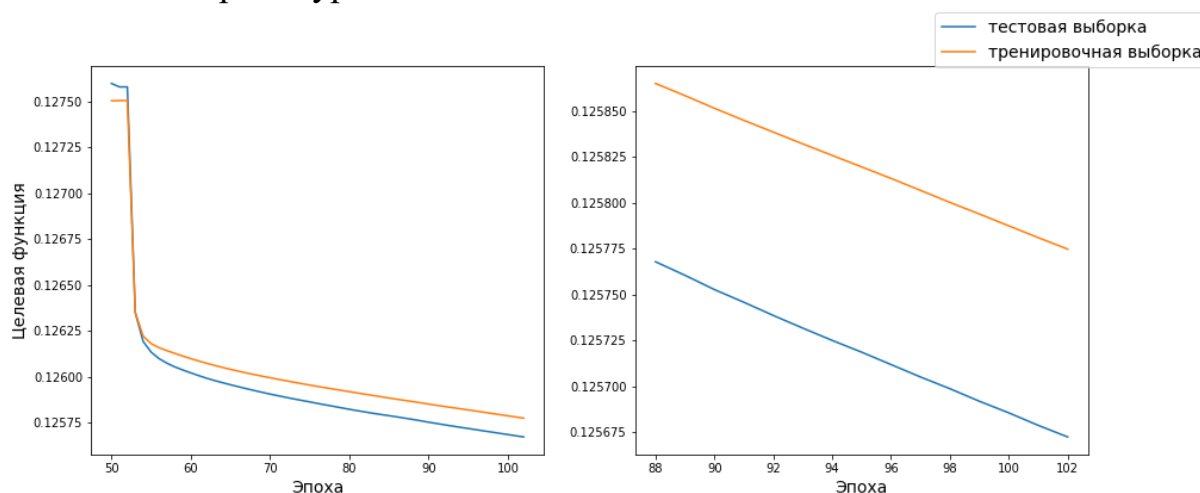


Рисунок 3.5 – Понижение `learning rate`
Примечание – Источник: собственная разработка.

Используя только описанные выше техники было очень сложно добиться пущего улучшения модели, потому следующий пункт мы начнем с описания тех улучшений которые нам потребовались дополнительно.

Этот пункт дает представление о том как начать обучение модели от технической подготовки данных и до первых заметных результатов, но тут не указано множество экспериментов, которые пришлось провести прежде чем мы пришли к такому состоянию, а указан лишь самый успешный случай. Тем не