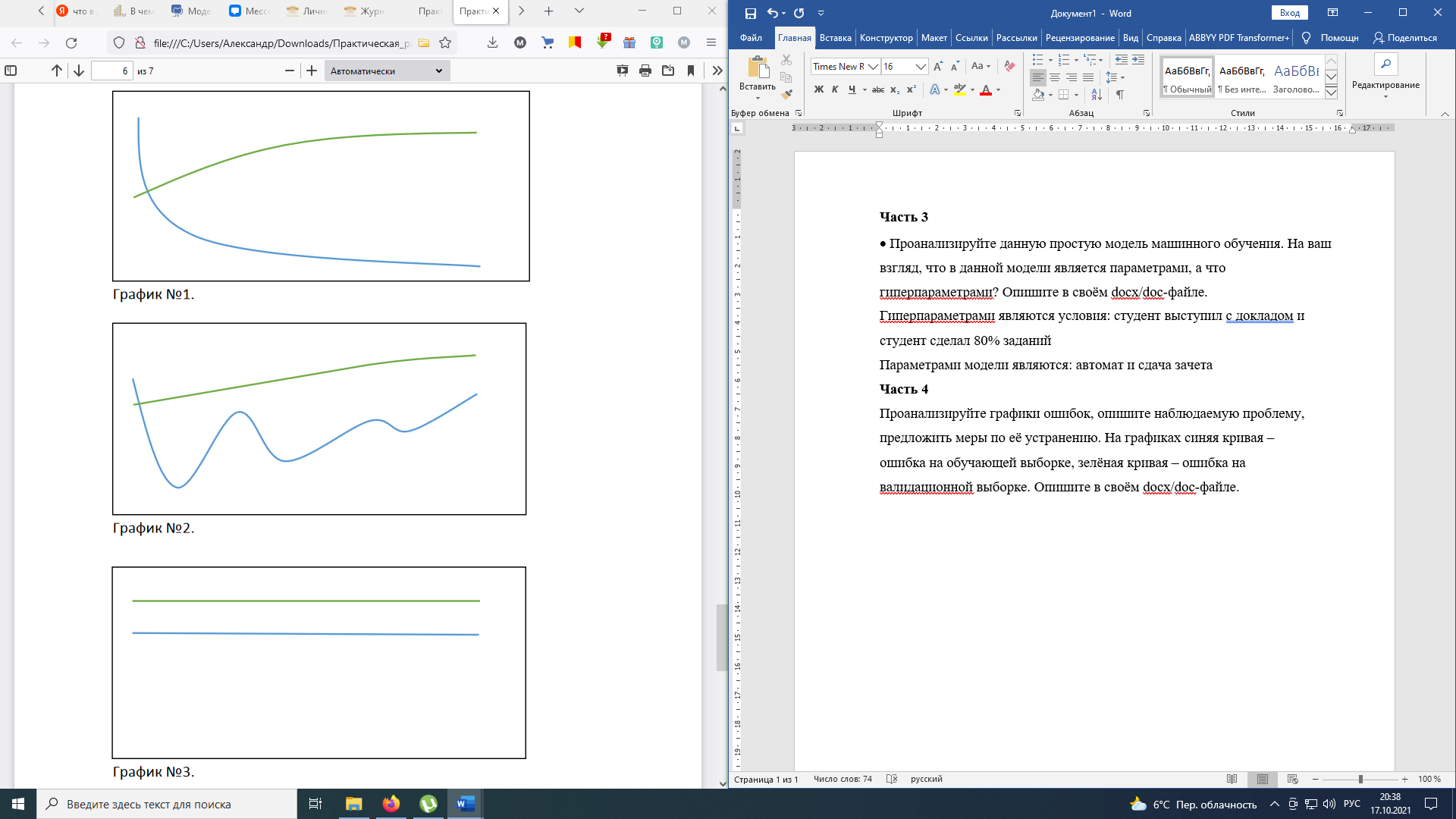
**Часть 3**  
• Проанализируйте данную простую модель машинного обучения. На ваш   
взгляд, что в данной модели является параметрами, а что   
гиперпараметрами? Опишите в своём docx/doc-файле.

Гиперпараметрами являются условия: студент выступил с докладом и студент сделал 80% заданий

Параметрами модели являются: автомат и сдача зачета

**Часть 4**

Проанализируйте графики ошибок, опишите наблюдаемую проблему,   
предложить меры по её устранению. На графиках синяя кривая –   
ошибка на обучающей выборке, зелёная кривая – ошибка на   
валидационной выборке. Опишите в своём docx/doc-файле.



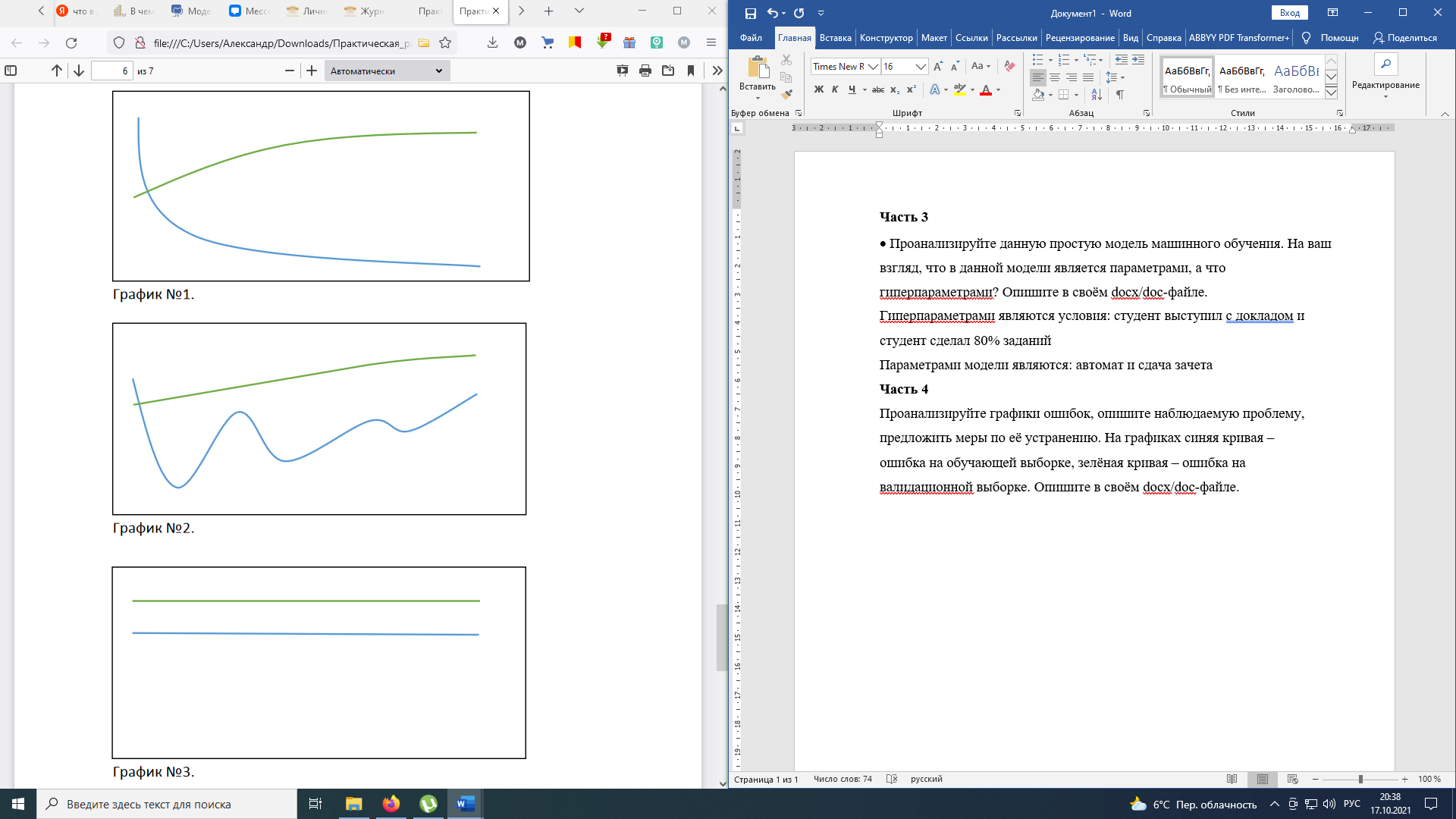
Если сеть обучать слишком долго или с использованием слишком большого числа параметров (весов), то произойдёт переобучение. Это связано с тем, что с определённого момента сеть начинает «подстраиваться» не под общие зависимости в данных, а под особенности отдельных примеров, которые могут содержать [аномальные значения](https://wiki.loginom.ru/articles/outlier.html), ошибки и т.д.

Меры по устранению**:**

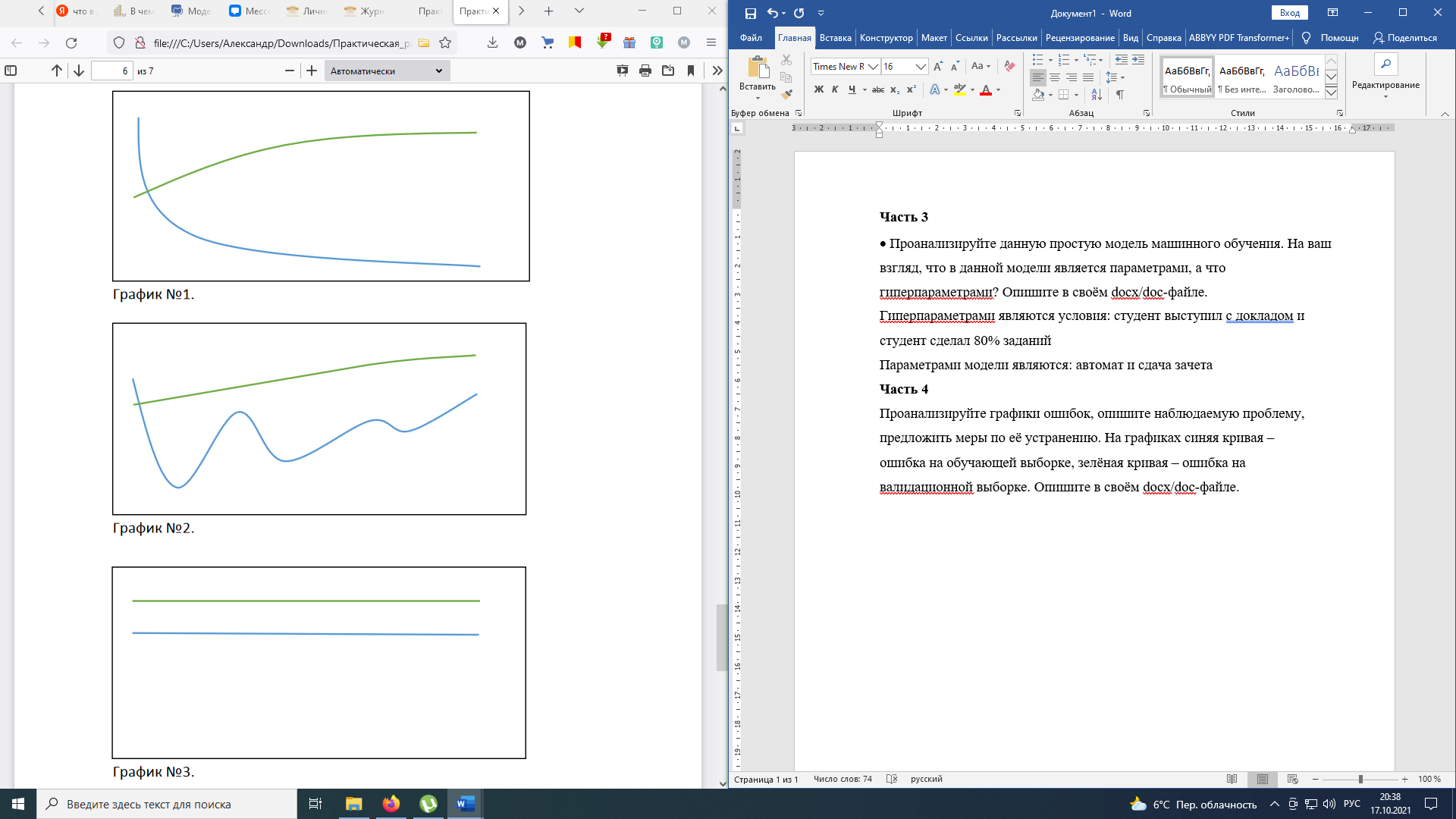
**Применение тестового множества**. [Тестовое множество](https://wiki.loginom.ru/articles/test-set.html) формируется из обучающего набора данных случайным образом и его примеры подаются на вход сети между обучающими примерами, но корректировку весов сети на вызывают.

**Использование** [**перекрёстной проверки**](https://wiki.loginom.ru/articles/cross-validation.html). Все данные, на которых строится модель, разделяются на k блоков равного размера. При этом обучение производится на k-1 блоках, а тестирование — на k-м. Процедура повторяется k раз, при этом для тестирования каждый раз выбирается другой блок. В результате все блоки оказываются используемыми и как обучающие, и как тестирующие.

**Выбор конфигурации нейронной сети**. Нужно выбирать конфигурацию сети (число слоёв и нейронов), чтобы количество параметров модели (весов) была в 2-3 раза меньше числа примеров обучающего множества. Если число параметров модели и обучающих примеров окажется соизмеримым, то сеть просто «запомнит» все комбинации вход-выход в примерах обучающего множества, и будет воспроизводить только их, а на новых данных допускать ошибки.



Недообучение - ситуация, когда в параметрическом семействе функций не удается найти функцию, которая хорошо описывает данные. Самая частая причина недообучения - когда сложность устройства данных выше, чем сложность устройства модели, которую придумал исследователь. Решением для такой проблемы будет усложнение модели и поиск лучшего описания эффектов, которые есть в данных.



Недообучение - ситуация, когда в параметрическом семействе функций не удается найти функцию, которая хорошо описывает данные. Самая частая причина недообучения - когда сложность устройства данных выше, чем сложность устройства модели, которую придумал исследователь. Решением для такой проблемы будет усложнение модели и поиск лучшего описания эффектов, которые есть в данных.