



Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра суперкомпьютеров и квантовой информатики

Практическое использование нейронных сетей для аппроксимации функции

ОТЧЕТ ПО ПРАКТИКУМУ

Студент дневного отделения бакалавариата

Лю Хайлинь

Преподаватель:

канд. физ.-мат. наук, Буряк Д.Ю.

Москва, 2024

I. ПОДСТАНОВКА ЗАДАЧИ

Аппроксимировать функции $f: R \rightarrow R$ с помощью полносвязанной сети, выборка составляет из 63 пары.

II. ОСНОВНЫЕ ЭЛЕМЕНТЫ

Самый распространенная проблема, возникающая при обучении глубоких нейронных сетей переобучение представляет собой переобучение, в этой задаче у нас есть только 63 пары данных для обучения. Мы считаем, что причиной переобучения будет слишком малый обучающий набор. Чтобы избежать переобучения, попробуем реализовать некоторые технологии, регуляризацию, кросс-валидацию, и обучение несколько моделей. Кроме того, использование многократной кросс-валидации позволяет лучше качество валидации.

2.1. Нормализация

Для точного анализа данных значения признаков необходимо нормализовать в определенном диапазоне. Этот метод позволяет избежать ошибок, связанных с размером и смещением значения. Процесс нормализации показан в уравнении (1), где X - значение, подлежащее нормализации, μ - математическое ожидание данного распределения, а σ - стандартное отклонение данного распределения. После применения нормализации все данные в наборе данных удовлетворяют распределению со средним значением 0 и стандартным отклонением 1.

$$Z = \frac{X - \mu}{\sigma} \quad (1)$$

2.2. Многократная кросс-валидация (K Flow валидация)

По многократной кросс-валидации, обучающий набор делится на k поднаборов данных одинаковой длины, используется $k - 1$ Flow для обучения, и проверяется на остальных данных (т.е. используется в качестве тестового набора для расчета показателей производительности, таких как точность).

Показатель производительности, полученный с помощью k -кратной кросс-валидации, представляет собой среднее значение, рассчитанное в цикле. Этот подход может

быть дорогостоящим в вычислительном отношении, но не тратит слишком много данных, что является основным преимуществом в таких задачах, как обратный вывод, когда количество выборок невелико.

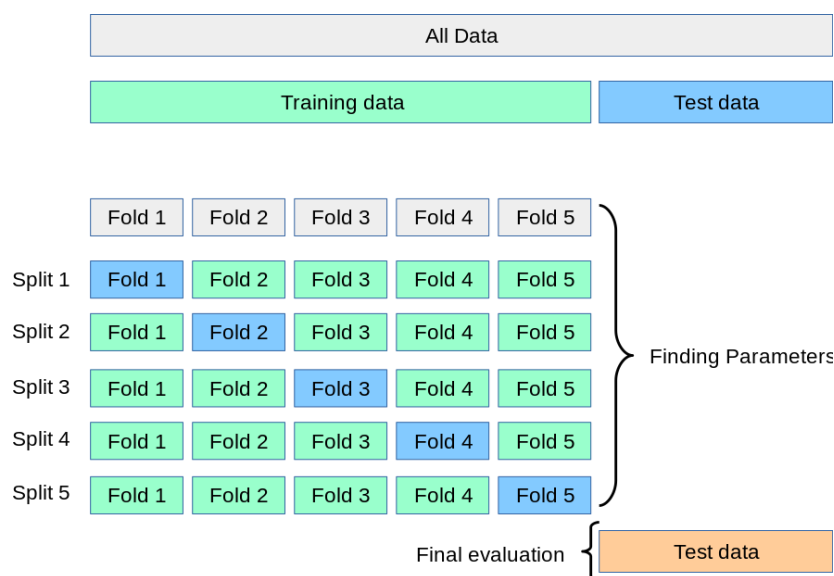


Рис.1 K-Flow валидация

III. ЭКСПЕРИМЕНТАЛЬНАЯ УСТАНОВКА

3.1. Аппаратное обеспечение

Критерием оценки эффективности нейронных сетей является время выполнения, которое зависит от производительности используемого компьютера. В связи с этим технические характеристики компьютера, использованного в настоящей работе, следующие:

Центральный процессор: 12-е поколение Intel(R) Core(TM) i7-12700H 2.30 ГГц

Оперативная память: 16 ГБ (используемая 15.7 ГБ)

Операционная система: Ubuntu 22.04.4 LTS

Графический процессор: NVIDIA GeForce RTX 3060 Laptop GPU

3.2. Способ реализации

Мы используем фреймворк Pytorch для создания нескольких многослойных перцептронов, и провести исследование структуры сети, размера пакетов и параметров регуляризации, используем `relu()` в качестве функции активации на скрытом слое и не используем функцию активации на выходном слое. Набор данных случайным образом

делится на обучающий набор и тестовый набор в соотношении 70%-30%, используя традиционный метод 5-Fold кросс-валидациями на обучающем наборе, и анализируем среднее значение 5 кривых потерь.

В качестве функции потерь выбираем `MSELoss()`, а оптимизатор `Adam()`.

IV. РЕЗУЛЬТАТ И ОБСУЖДЕНИЕ

4.1. Рассмотрение результатов обучения

Мы отдельно обучаем сети с 1 или 2 скрытыми слоями на основе многослойных перцептронов, сравниваем и анализируем результаты подгонки с 6, 8 или 10 нейронами в скрытом слое. Записываем взаимосвязь между изменением потерь и количеством итераций в обучающем наборе и наборе проверки, генерируем кривую потерь.

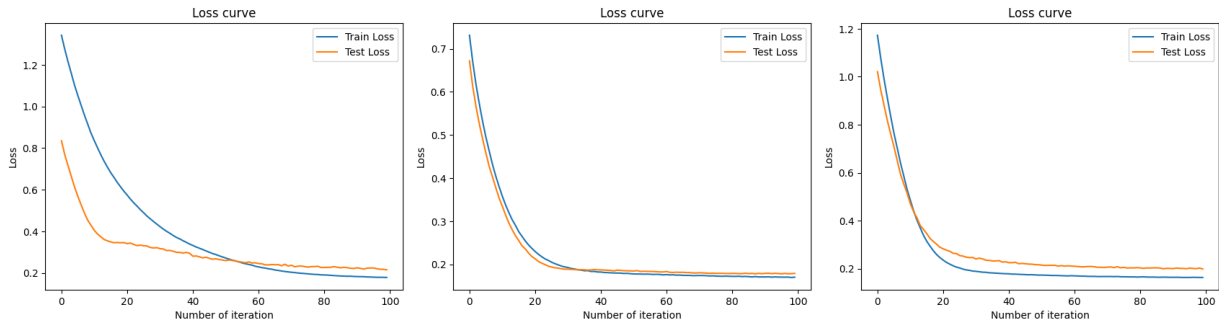


Рис.2 Loss curve 1 скрытого слоя (6, 8, 10 нейронов)

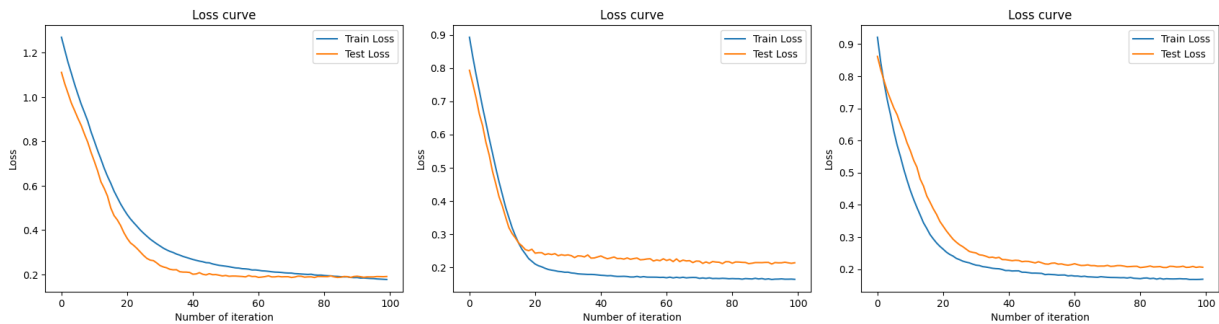


Рис.3 Loss curve 2 скрытых слоев (6, 8, 10 нейронов на каждом)

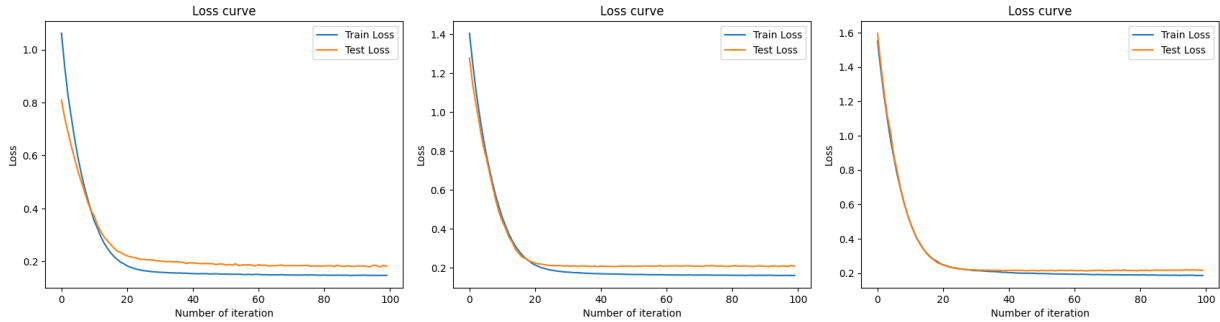
Number of neurons	1 hidden layer		2 hidden layers	
	Training loss	Validation loss	Training loss	Validation loss
6	0.1794	0.2192	0.1792	0.1902
8	0.1700	0.1784	0.1641	0.2134
10	0.1632	0.2006	0.1682	0.2067

Таб.1 Потери на обучающем и валиционном наборе при разных количествах нейронов

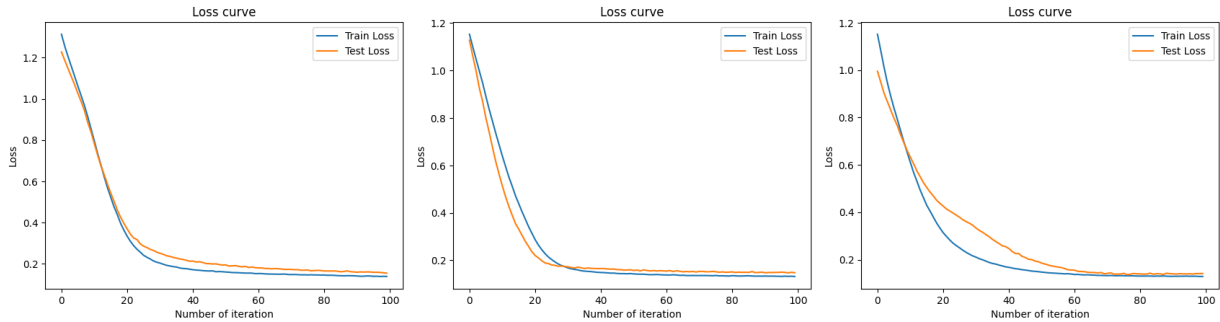
Мы выполнили 100 итераций и зафиксировали средние потери за последние 5 итераций. Как видно по результатам, модель с большим количеством нейронов также имела меньшие потери на обучающем наборе, но это тоже приводило к чрезмерным проблемам. по мере увеличения сложности модели потери в наборе проверки увеличиваются и происходит переобучение.

4.2. Применение L2 регулязации

Мы строим полносвязные сети с 1 и 2 скрытыми слоями соответственно, где сеть с 1 скрытым слоем имеет 16 нейронов, а сеть с 2 скрытыми слоями имеет 6 нейронов на каждом скрытом слое. С помощью трех различных параметров `weight_decay` анализируем влияние коэффициента L2 регулязации на эффект подгонки модели.



*Рис.4 Loss curve перцептрона с 16 нейронами
(weight decay=[1e-6, 1e-5, 1e-4])*



*Рис.5 Loss curve перцептрона с 6 нейронами на обоих слоях
(weight decay=[1e-6, 1e-5, 1e-4])*

weight decay	16 нейронов		6 нейронов на обоих слоях	
	Training loss	Validation loss	Training loss	Validation loss
1e-6	0.1467	0.1825	0.1390	0.1572
1e-5	0.1596	0.2079	0.1325	0.1486
1e-4	0.1868	0.2174	0.1295	0.1407

Таб.2 Потери при разных параметрах L2 регулязации

Мы обнаружили, что применение регулязации в трехслойных и четырехслойных сетях дает совершенно разные результаты.

В более простой трехслойной сети по мере увеличения параметров регуляризации также увеличиваются конечные потери модели на обучающем наборе, что может указывать на то, что эта более простая модель не может хорошо соответствовать нашему набору данных модели.

В более сложной сети из четырех слоев произошло явление переобучения. По мере увеличения параметров регуляризации потеря модели на проверочном наборе постепенно приближается к потере на обучающем наборе, что показывает, что регуляризация может облегчить проблему переобучения, вызванной сложными моделями.

4.3. Исследование влияния размера пакета

Изучаем размер пакета в двух сетях, упомянутых в предыдущем разделе. Параметр регуляризации трехуровневой сети был установлен на $1e-6$, а параметр регуляризации четырехуровневой сети был установлен на $1e-4$. В нашем наборе данных размер обучающего набора составляет 36 выборок, а размер проверочного набора 9 выборок. Мы изучаем случай, когда размер пакета в обучающем наборе и проверочном наборе равен 1, 3 или 9. В дополнение к потере мы также записываем количество итераций обработки в секунду на платформе.

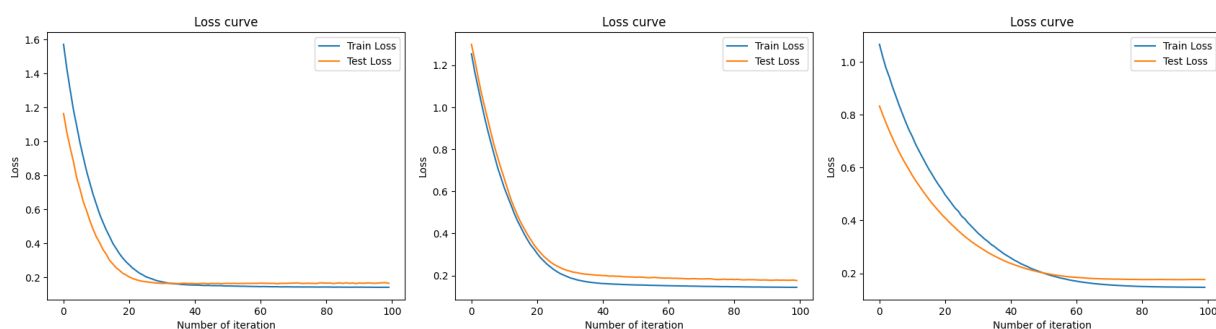


Рис.6 Loss curve перцептрона с 16 нейронами (batch size=[1, 3, 9])

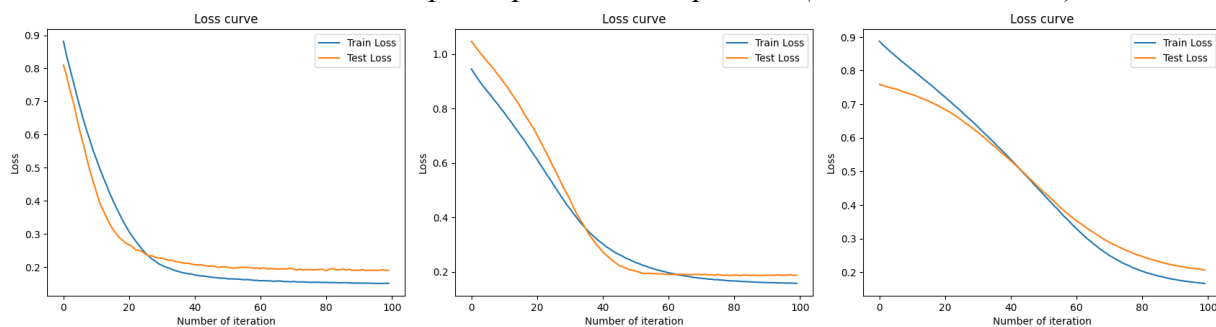


Рис.7 Loss curve перцептрона с 6 нейронами на обеих слоях (batch size=[1, 3, 9])

batch_size	16 нейронов			6 нейронов на обоих слоях		
	Train loss	Val loss	Speed(itps)	Train loss	Val loss	Speed(itps)
1	0.1413	0.1666	21.08	0.1500	0.1899	19.41
3	0.1440	0.1776	51.90	0.1584	0.1880	46.93
9	0.1470	0.1764	95.64	0.1682	0.2089	91.41

Таб.3 Потери при разных размерах пакета

Мы наблюдаем, что чем больше размер пакета, тем больше итераций сеть может выполнить за определенный период времени, но в то же время, чтобы обеспечить эффект подгонки, необходимо больше итераций при подгонке целевой кривой. В трехуровневой сети, когда размеры пакетов равны 1, 3 и 9 соответственно, количество требуемых итераций составляет соответственно 25, 35 и 60. В четырехуровневой сети необходимое количество итераций составляет 35, 60. и 100 соответственно. Для этого набора данных лучше подходят небольшие размеры пакетов.

4.4. Окончательная оценка

Наконец, мы обучаем модель на всем обучающем наборе и проверяем потери на тестовом наборе. Мы продолжаем использовать параметры модели из предыдущего раздела и устанавливаем пакетный размер равным 1.

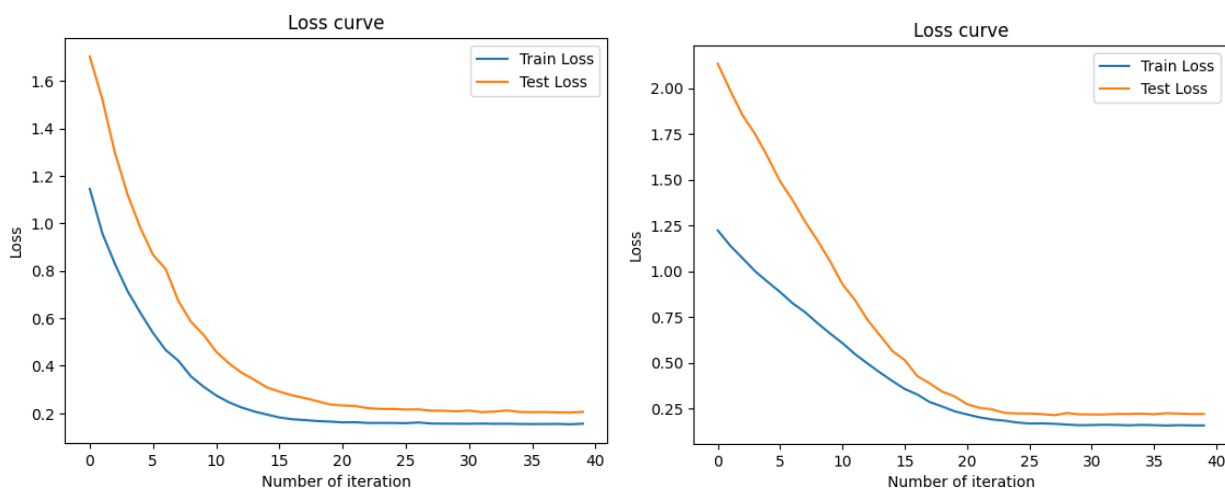


Рис.8 Loss curve 3 и 4-слойной нейронной сети

- Потеря 3-слойной нейронной сети:

Training loss: 0.1546 Validation loss: 0.2043

- Потеря 4-слойной нейронной сети:

Training loss: 0.1576 Validation loss: 0.2207

V. ЗАКЛЮЧЕНИЕ

В настоящей статье мы обучали сети с 1 или 2 скрытыми слоями на основе многослойных перцептронов, сравнивали результаты подгонки с разным количеством нейронов в скрытом слое, обнаружили, что модель с большим количеством нейронов имела меньшие потери на обучающем наборе, но это также приводило к переобучению на наборе проверки.

Мы применили L2 регуляризацию, в более простой трехслойной сети увеличение параметров регуляризации приводило к увеличению конечных потерь на обучающем наборе, а в более сложной сети из четырех слоев регуляризация помогла снизить переобучение.

Мы также исследовали влияние размера пакета, наблюдали, что больший размер пакета позволяет сети выполнять больше итераций за определенное время, однако для эффективной подгонки требуется больше итераций при маленьких размерах пакетов.

Наконец, мы обучили модель на всем обучающем наборе и проверили потери на тестовом наборе, зафиксировали потери для 3-слойной и 4-слойной нейронных сетей.

БИБЛИОГРАФИЯ

- [1] Cross-validation: evaluating estimator performance [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html [Accessed: 31-MAR-2024].