ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет об исследовательском проекте на тему:

Исследование парадигмы лотерейных билетов в глубинном обучении

Выполнил:

студент группы БПМИ216
Панфилов Борис Сергеевич _____ (подпись) _____ (дата)

Принял руководитель проекта:

Шабалин Александ Михайлович
Внештатный преподаватель
Факультета компьютерных наук НИУ ВШЭ

_____ (подпись) _____ (дата)

Москва 2023

Содержание

Аннотация

В данном проекте проводится исследование парадигмы Lottery Ticket Hypothesis, предполагающую существование в исходной инициализации модели "подмодели" при обучении которой достигается качество полной модели. В ходе работы над проектом будет необходимо провести ряд экспериментов для обнаружения зависимостей между архитектурой, размером, способом обучения модели и возникновением описанного феномена.

Ключевые слова

Глубинное обучение, полносвязные сети, сверточные сети, гипотеза лотерейных билетов, прунинг, ландшафт потерь, интерполяция, гиперпараметры

1 Введение

В основном, модели глубинного обучения обладают большим числом параметров, что требует значительных вычислительных ресурсов и памяти. Поэтому необходимо придумывать различные способы уменьшения размера сети, желательно, без потери качества. Человечество придумало для решения этой проблемы следующие методы: прунниг [2], дестилляция [3], квантизация [5]. В этом проекте мы подробнее рассмотрим прунниг, так же известный как "гипотеза лотерейных билетов". Этот метод впервые был описан в статье [2] и утверждает, что внутри большой нейронной сети существуют подсети, которые можно идентифицировать и обучить в отдельности с такой же или даже лучшей производительностью, чем у полной сети.

В данной работе мы исследуем эту гипотезу, проводя серию экспериментов на различных архитектурах нейронных сетей и задачах классификации изображений. Мы обнаруживаем, что прунинг (удаление весов) большой нейронной сети и последующее дообучение оставшихся весов позволяют найти эти "лотерейные билеты"или подсети. Удивительно, что эти подсети могут достичь такой же или близкой к ней производительности, как полная сеть, даже если они значительно меньше по размеру.

В наших экспериментах мы показываем, что лотерейные билеты можно найти в разных архитектурах, включая полносвязные сети и сверточные нейронные сети, и на разных наборах данных. Однако бывают ситуации, когда лотерейные билеты не находяться, если пользоваться алгоритмом, описанным в статье [2], тем не менее, модифицировав различные параметры обучения, найти подходящие подсети все же можно. В следствие чего одной из целей работы стало исследование этих изменений с точки зрения пейзажа ошибок.

2 Постановка задачи

Рассмотрим плотную нейронную сеть $f(x;\theta)$ с начальными параметрами $\theta = \theta_0 \sim D_\theta$. При оптимизации с помощью стохастического градиентного спуска (SGD) на обучающей выборке, f достигает минимальной валидационной потери l на итерации j с точностью на тестовой выборке a. Кроме того, рассмотрим обучение модели $f(x; m \odot \theta)$ с маской $m \in \{0,1\}^{|\theta|}$ и начальной инициализацией модели $m \odot \theta_0$. При оптимизации с SGD на том же обучающем множестве (с фиксированным m), f достигает минимальной валидационной потери l' на итерации j' с точностью на тестовой выборке a'. Гипотеза лотерейных билетов утверждает, что $\exists m$, для которых $j' \leq j$ (соизмеримо время обучения), $a' \geq a$ (соизмеримая точность), и

$\|m\|_0 \ll |\theta|$ (меньше параметров).

Мы обнаружили, что стандартная техника обрезки в большинстве случаев выявляет такие обучаемые подсети из полносвязных и сверточных сетей. Мы обозначаем эти обучаемые подсети, $f(x; m \odot \theta_0)$, выигрышными билетами, поскольку они выиграли в лотерею инициализации с комбинацией весов и связей, способных к обучению.

Задача данной работы заключается в том, чтобы определить, при каких настройках сети выигрышные билеты находятся, и объяснить это с точки зрения ландшафта ошибок.

3 Изучение основ глубинного обучения

Поскольку в данной работе мы исследуем нейронные сети, нам нужно было изучить то, как происходит обучение сетей, какие существуют методы опитмизации обучения, какие бывают слои в моделях глубинного обучения. А также изучить как все это реализовано в библиотеке PyTorch, чтобы в дальнейшем реализовать для проведения экспериментов.

3.1 Процесс обучения

Обучение нейронных сетей происходит путем подачи обучающих примеров (наборов данных) на вход сети и корректировки весов сети на основе полученных предсказаний и целевых значений. Глобально процесс обучения нейронной сети включает следующие шаги:

1. Инициализация весов: веса нейронной сети инициализируются случайными значениями перед началом обучения. Хорошая инициализация весов может помочь ускорить процесс обучения и достижение лучших результатов.

2. Прямое распространение: на вход нейронной сети подаются входные данные, после чего последовательно в каждом слое вычисляется функция активации каждого нейрона из этого слоя, что позволяет получить предсказания модели.

3. Вычисление функции потерь: функция потерь определяет разницу между предсказаниями модели и целевыми значениями. Чем меньше значение функции потерь, тем лучше модель выполняет поставленную задачу.

4. Обратное распространение: на данном этапе используется метод градиентного спуска. Подсчитываются градиенты функции потерь по отношению к весам сети. Градиенты передаются назад через сеть, позволяя определить, какие веса нужно корректировать и на сколько.

5. Обновление весов: Веса сети обновляются на основе вычисленных градиентов. Обычно используются оптимизации алгоритма градиентного спуска, такие как стохастический градиентный спуск или его улучшения (например, Adam или AdamW), чтобы корректировать веса в направлении уменьшения функции потерь.

6. Повторение шагов 2-5: Процессы прямого и обратного распространения повторяются для каждой серии обучающих данных до тех пор, пока не будет достигнуто условие остановки, такое как достижение определенного количества эпох обучения или сходимости функции потерь.

3.2 Виды слоев

Далее, чтобы строить собственные нейронные сети, нам необходимо было узнать их составные части – слои. Как правило, в моделях, которые используются в задачах классификации изображений, используются полносвязные и сверточные слои.

Полносвязные слои представляют собой наиболее простой тип слоев в нейронных сетях. В полносвязном слое каждый нейрон связан с каждым нейроном предыдущего и следующего слоя. Входные данные подаются на вход полносвязного слоя, и каждый нейрон вычисляет взвешенную сумму входов, применяет функцию активации и передает результаты на выход. В полносвязных слоях используется большое количество параметров, поскольку каждый нейрон имеет свои собственные веса.

Сверточные слои являются ключевым элементом сверточных нейронных сетей, которые широко используются для анализа изображений. Они основаны на операции свертки, которая позволяет эффективно выделять локальные шаблоны и признаки входных данных.

Эти слои состоят из набора фильтров (или ядер), которые скользят по входным данным и производят свертку с ними. Каждый фильтр представляет собой набор весов, которые используются для вычисления свертки. При свертке фильтр перемещается с некоторым шагом (шаг свертки) по входу, умножая значения в окне свертки на соответствующие веса и вычисляя сумму произведений. Результаты свертки передаются через функцию активации на выход.

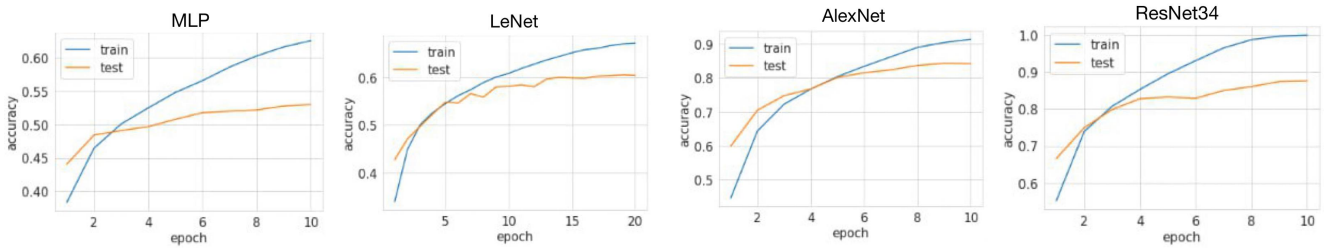

Рис. 3.1: Результаты обучения различных архитектур на датасете CIFAR10.

3.3 Известные архитектуры

Теперь мы были готовы к написанию первых моделей. Чтобы понять как устроен этот мир, было принято решение добиться качества около 90% на датасете CIFAR10, который состоит из 10 классов изображений.

Мы решили начать с простейших основ, и первой нашей моделью стал многослойный персептрон (рис. 3.1 - 1 график). Он состоял из 2 полносвязных слоев по 64 нейрона в каждом. Результат получился довольно неплохой - порядка 53%. С одной стороны, мы были впечатлены, но с другой, этого было явно мало.

Далее мы решили начать использовать сверточные слои, так как они широко используются для анализа изображений и способны эффективно выделять локальные шаблоны и признаки входных данных. В нашей работе первой моделью со сверточными слоями стала нейронная сеть LeNet5 [6]. Процесс ее обучения изображен на 2 графике рис. 3.1, она показала себя на 7% лучше, чем обычная полносвязная сеть. Тем не менее, результат был всего 60%, потому что модель все еще содержала слишком мало весов, чего нам явно было недостаточно.

Следующей моделью стала AlexNet [1]. Если LeNet содержит порядка 420 тысяч параметров, то AlexNet - уже порядка 60 миллионов (что в 142 раза больше). График ее обучения, третий на рис. 3.1, и она показала весьма хороший результат - 83%.

До 90% не дотягивали, поэтому мы решили обучить заключительную модель в этом списке - ResNet34 [4]. Основная идея ResNet заключается в использовании блоков со скип-соединениями или остаточными соединениями, которые позволяют сети эффективно обучаться вопреки проблемам с затуханием градиентов. В итоге нам удалось получить качество около 89%, и мы перешли к эксперементам по пруннингу моделей.

4 Воспроизведение алгоритма поиска лотерейных билетов

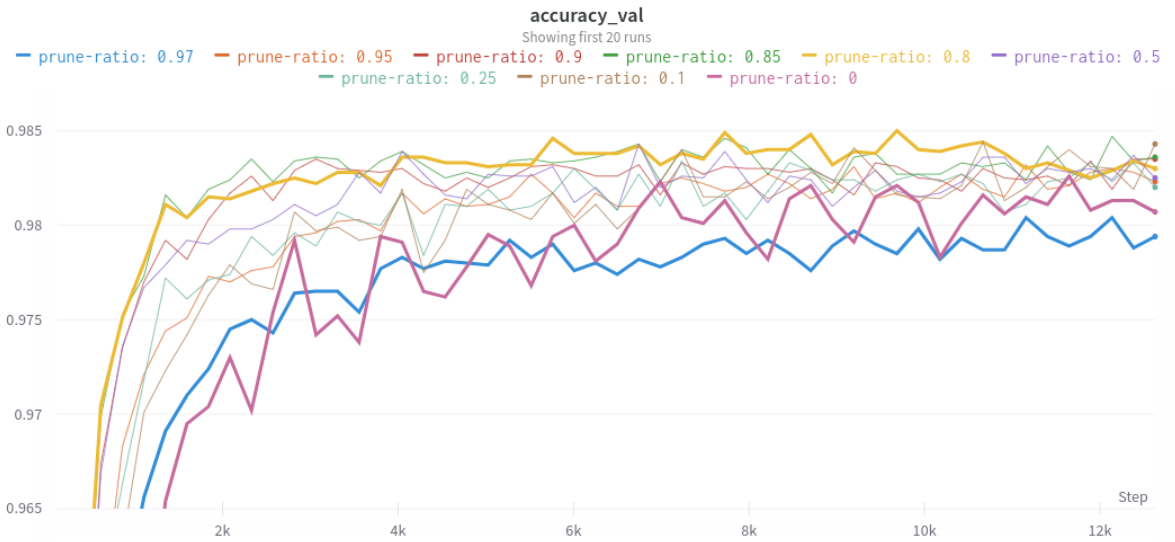

Рис. 4.1: Процесс обучения моделей. Результаты сгруппированы по доле обрезанных весов в моделях.

В данной части работы мы обучили 216 многослойных персептронов(рис. 4.1) в разных конфигурациях на датасете MNIST, чтобы воспроизвести результаты статьи [2]. А именно убедиться в том, что теория лотерейных билетов работает и понять какой из предложенных в статье алгоритмов лучше ищет выигрышные билеты. Основной алгоритм формулируется следующим образом:

1. Случайно инициализируем нейронную сеть $f(x;\theta_0)$ (где $\theta_0 \sim D_0$).
2. Обучить сеть в течение j итераций, получив параметры θ_j .
3. Обрезать $p\%$ параметров в θ_j, создав маску m.
4. Сбросить оставшиеся параметры до их значений в θ_0, создавая выигрышный билет $f(x; m \odot \theta_0)$.

Этот подход к обрезке является одноразовым: сеть обучается один раз, p% весов обрезаются, а оставшиеся веса инициализируются изначальными значениями. Однако в данной работе мы сосредоточимся на итеративной обрезке, которая повторяет пункты 2-4 в течение n раундов; в каждом раунде обрезается $p^{\frac{1}{n}}\%$ от весов, которые выжили в предыдущем раунде. Наши результаты показывают, что итеративная обрезка находит выигрышные билеты

которые соответствуют точности исходной сети при меньших размерах, чем при одноразовой обрезке.

Всего в статье было описано 2 различные конфигурации этого алгоритма, которые можно менять:

1. Реинициализация весов после каждой итерации пруннинга. С одной стороны можно инициализировать выжившие веса изначальными значениями, а с другой - случайными.

2. Способ выбора весов для обрезки. С одной стороны можно обрезать случайные веса модели, а с другой те, у которых абсолютное значение минимально.

Это можно представить в виде следующей таблички. За выигрышные билеты я обозначил место, где по результатам авторов статьи находится наилучшая конфигурация.

Метод обрезки\Реинициализация весов	Да	Нет
Случайный выбор	?	?
L1	?	Выигрышные билеты

Без реинициализации и веса для обрезки выбираем случайно

При обучении используем слудующий алгоритм:

1. обрезаем p% весов

2. присваиваем всем значениям изначальные и дообучаем модель

3. повторяем 1-2 n раз

4. присваиваем модели изначальные веса

5. обучаем модель

То есть смысл этих моделей показать важность обрезки именно меньших по модулю весов, а не случайных.

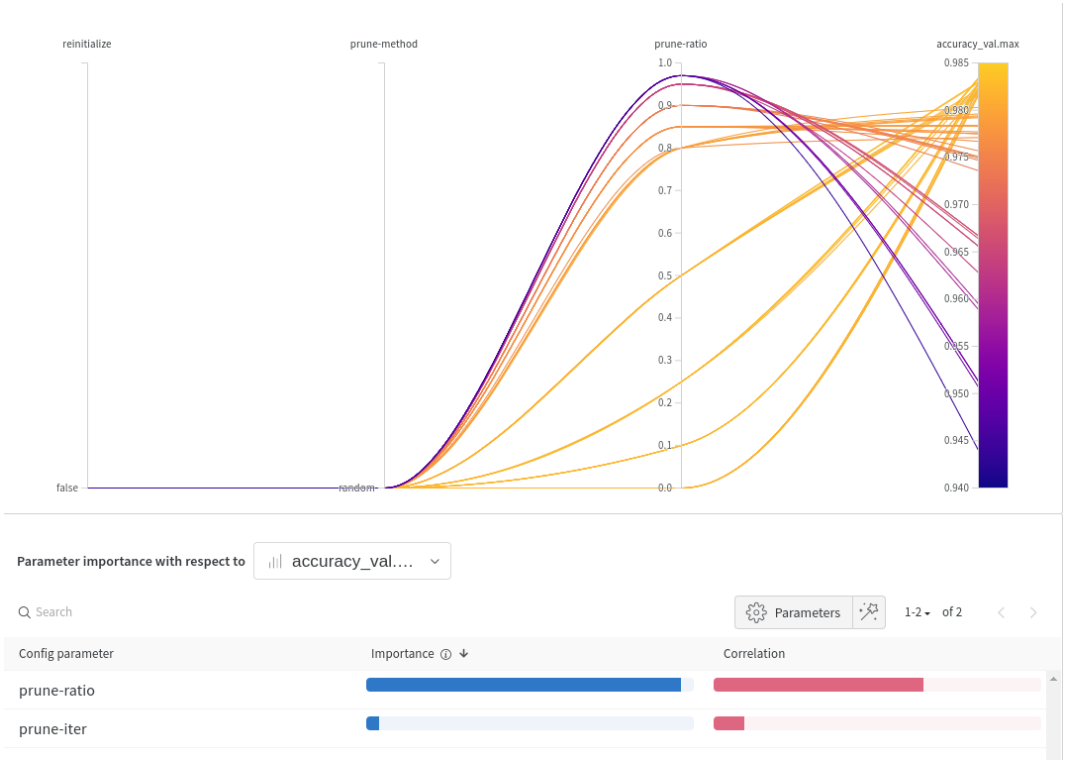

Рис. 4.2: Первый столбец - реинициализируем мы значения после каждой итерации или нет (в данном случае нет). Второй столбец - то как мы выбираем веса для обрезки(в данном случае случайно). Третий столбец - часть весов которую мы обрезаем. Четвертый столбец - результат обученной модели на тестовой выборке. Под графиком в первой строчке важность и корреляция доли обрезонных весов с результатом, во второй - количества итераций.

Не трудно заметить отрицательную корреляцию(-0.639) и если учитывать весь диапазон prune-ratio, и если выбрать некритические уровни обрезки моделей - (0% - 85%). А значит можем сделать вывод, что при таком методе обучения уменьшение весов модели влечет ухудшение качества.

Случайно реинициализируем веса и обрезаем наименьшие по модулю веса

При обучении используем слудующий алгоритм:

1. Обрезаем p% весов
2. Присваиваем всем значениям изначальные и дообучаем модель
3. Повторяем 1-2 n раз
4. Присваиваем начальным весам случайные веса
5. Обучаем модель

То есть смысл этих моделей показать важность инициализации модели изначальными весами в конце.

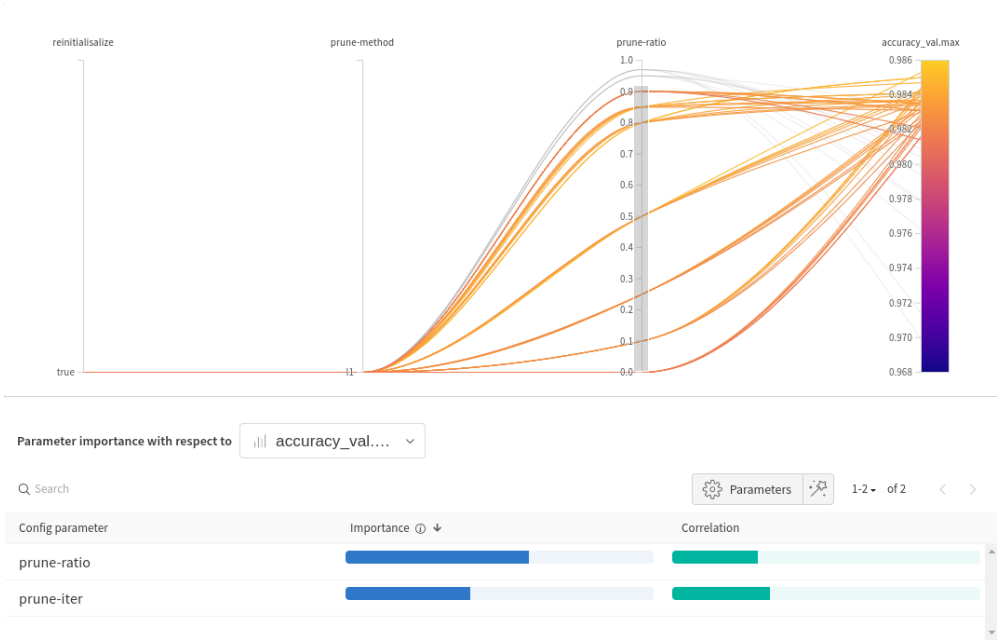

Рис. 4.3: Первый столбец - реинициализируем мы значения после каждой итерации или нет(в данном случае да). Второй столбец - то как мы выбираем веса для обрезки(в данном случае по абсолютному значению). Третий столбец - часть весов которую мы обрезаем. Четвертый столбец - результат обученной модели на тестовой выборке. Под графиком в первой строчке важность и корреляция доли обрезонных весов с результатом, во второй - количества итераций.

Если рассматривать все модели, то корреляция по прежнему будет отрицательной. Тем не менее, если оставить лишь некритические уровни моделей, то мы получим положительную корреляцию порядка 0.418. Это уже довольно круто, но в статье говорится, что это еще не лучший алгоритм, так что рассмотрим наилучшую комбинацию далее.

Без реинициализации и обрезаем наименьшие по модулю веса

При обучении используем слудующий алгоритм:

1. Обрезаем p% весов
2. Присваиваем всем значениям изначальные и дообучаем модель
3. Повторяем 1-2 n раз
4. Присваиваем начальным весам случайные веса
5. Обучаем модель

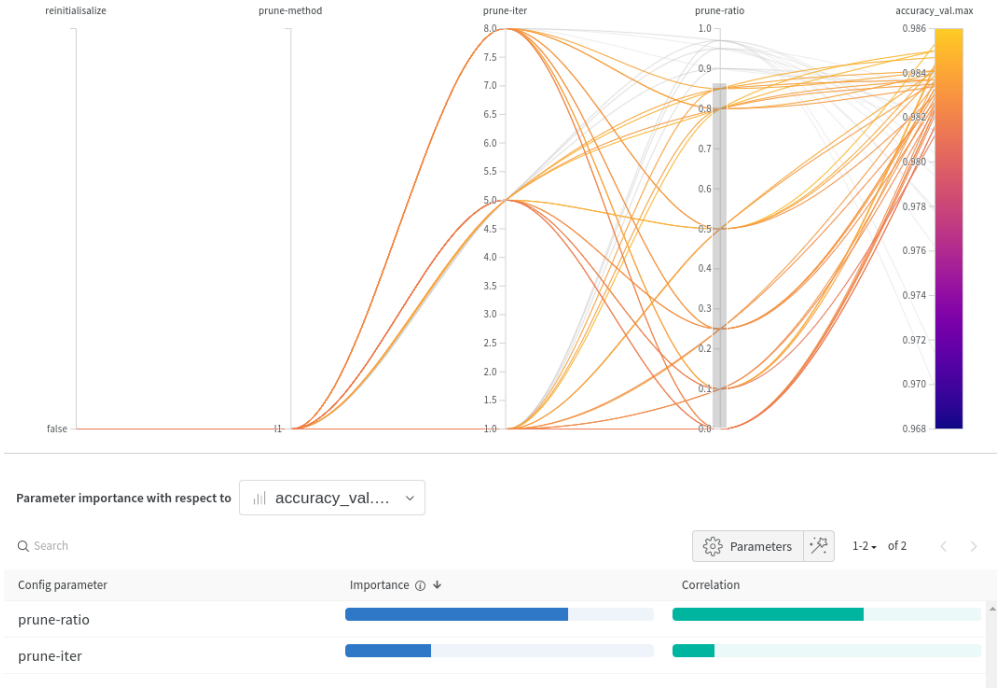

Рис. 4.4: Первый столбец - реинициализируем мы значения после каждой итерации или нет (в данном случае нет). Второй столбец - то как мы выбираем веса для обрезки (в данном случае по абсолютному значению). Третий столбец - часть весов которую мы обрезаем. Четвертый столбец - результат обученной модели на тестовой выборке. Под графиком в первой строчке важность и корреляция доли обрезонных весов с результатом, во второй - количества итераций.

Исходя из статьи именно тут мы должны получить наилучший результат.

Если рассматривать все модели, то корреляция получается по прежнему отрицательной. Но оно и логично, потому что вряд ли у модели с 5% весов удастся работать так же хорошо, как и со 100%. Тем не менее, если мы как обычно рассмотрим только модели с уровнем обрезки не более 85%, то мы получим наибольшую корреляцию - 0.619. То есть мы можем сделать вывод, что мы нашли алгоритм определяющий выигрышные билеты (наиболее важные веса), такие что при обучении только с ними мы не теряем качество.

5 Исследование ландшафта ошибок при поиске лотерейных билетов

В данной части работы исследуется причина, по которой слишком большой уровень пруннинга мешает получить такое же хорошее качество, как и на наеобрезанной модели с точки зрения ландшафта ошибок, способы позволяющие преодолеть эти преграды и исследование того, почему они помогают. В данном разделе исследование проводится на датасете

CIFAR100 и архитектуре ResNet18 , так как она более приближена к современным нейросетям по размеру и используемым технологиям.

Кроме того, замечу, что теперь в обучении используется только конфигурация, в которой мы не реинициализируем и обрезаем наименьшие по модулю веса, так как этот алгоритм показал себя лучше всего в предыдущей части.

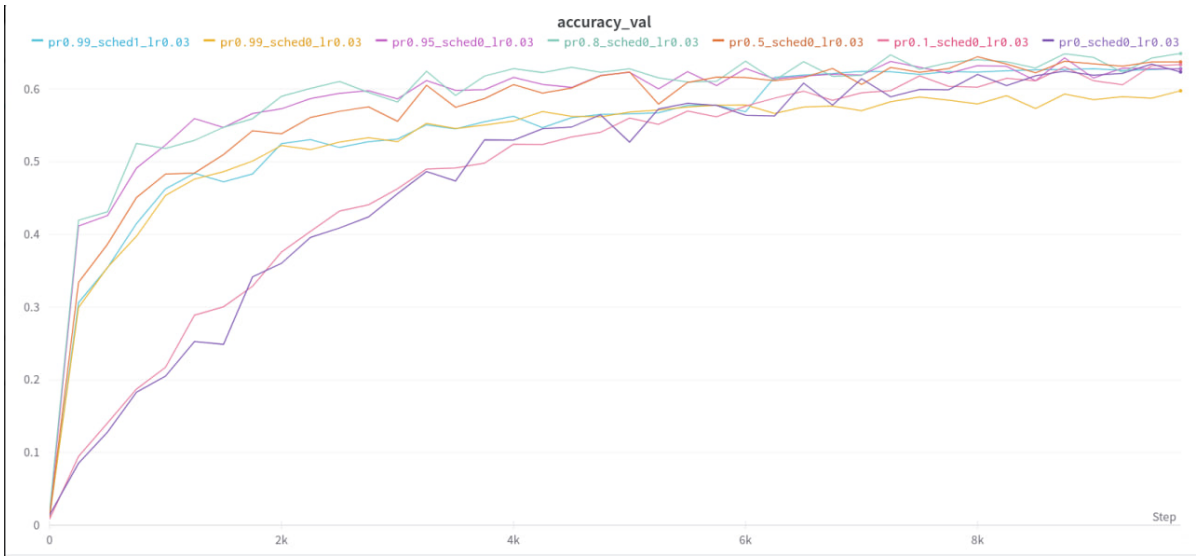

Рис. 5.1: Процесс обучения моделей ResNet18 с разными степенями обрезки.

На графике 5.1 можно заметить, что 5 из 6 обрезанных моделей, показали качестество даже лучшее, чем не обрезанная модель. В целях нашего исследования требовалось понять, почему желтой модели(с долей обрезки 0.99) не удалось получить такое же хорошее качество, как и изначальная модель. Для этого мы построили интерполяцию, то есть нашли значения в промежуточных точках.

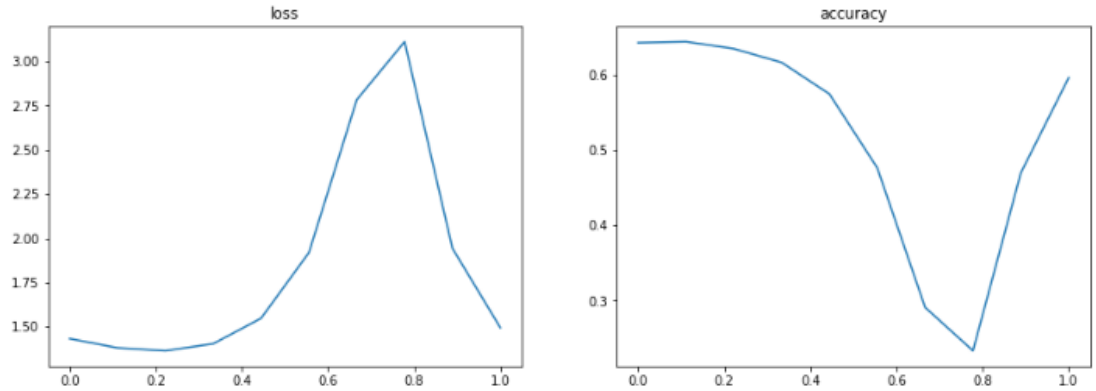

Рис. 5.2: Интерполяция между необрезанной моделью и моделью с уровнем обрезки 0.99 .

Из графиков 5.2 видно что модели попали в разные базины. Именно это и помешало модели с уровнем обрезки 0.99 показать столь же хорошее качество, как и у необрезанной модели.

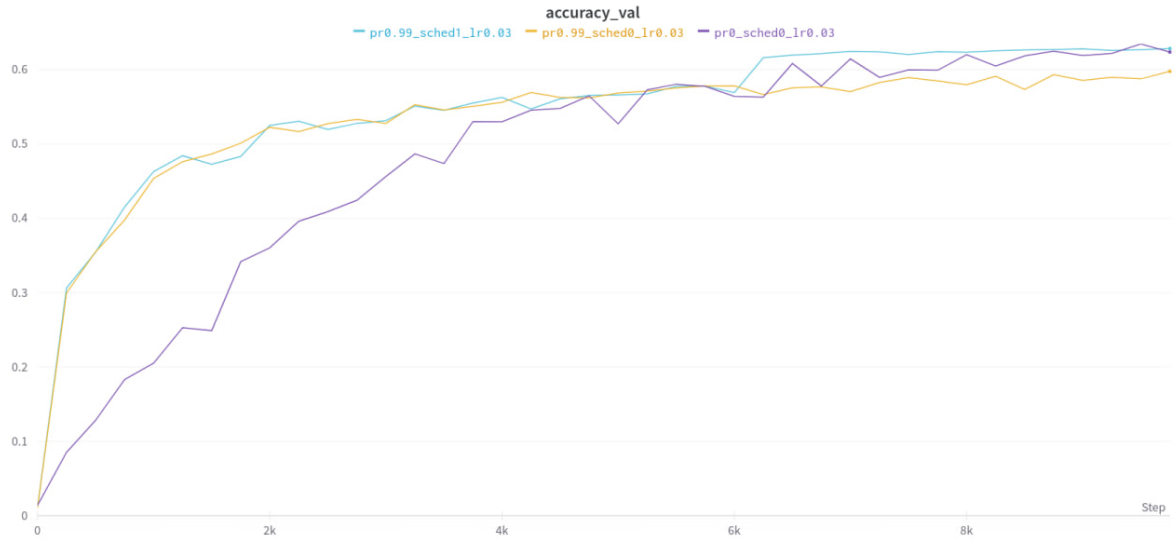

Рис. 5.3: Графики обучения 3 моделей. Фиолетовая - необрезанная. Желтая - со степенью обрезки 0.99 и без использования расписания при обучении. Голубая - со степень обрезки 0.99 и с использования расписания при обучении.

Пробуя изменять разные гиперпараметры, у нас получилось достичь желаемого. А именно найти выигрышные билеты для модели со степенью обрезки 0.99 5.3. То есть нам реально удалось получить качество выше, чем у модели, содержащей все веса. Мы решили понять, почему расписание помогло в данной ситуации. Гипотеза состояла в том, что из-за применения расписания модель при обучении попадает в другую базину. Для того, чтобы проверить это мы построили интерполяцию между необрезанной моделью и двумя со степенью пруннинга 0.99 .

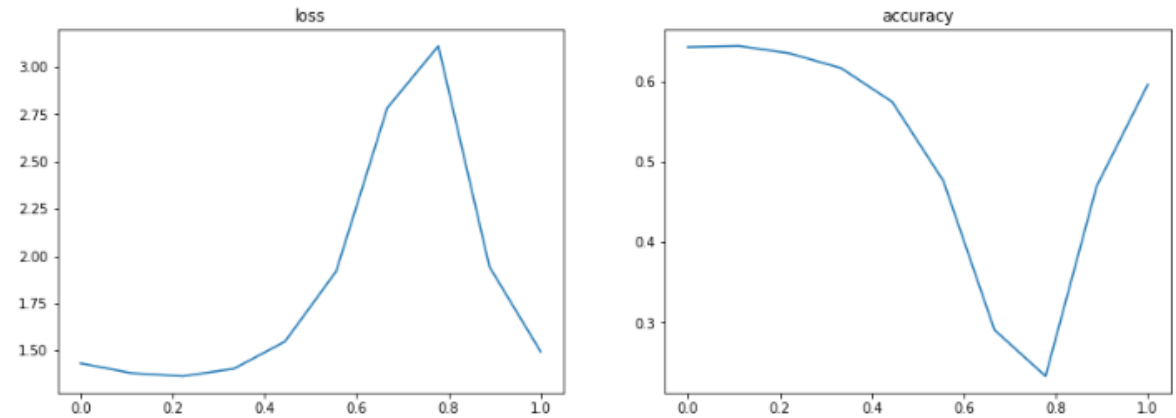

Рис. 5.4: Интерполяция между необрезанной моделью и моделью обрезанной на 0.99 без расписания

К сожалению построенные графики не дали нам доказательства желаемой гипотезы, потому что необрезанная модель и модель с шедулером попали в разные базины. Далее мы планируем продолжать исследовать, почему шедулер смог улучшить результат и изменением каких еще гиперпараметров можно добиться улучшения.

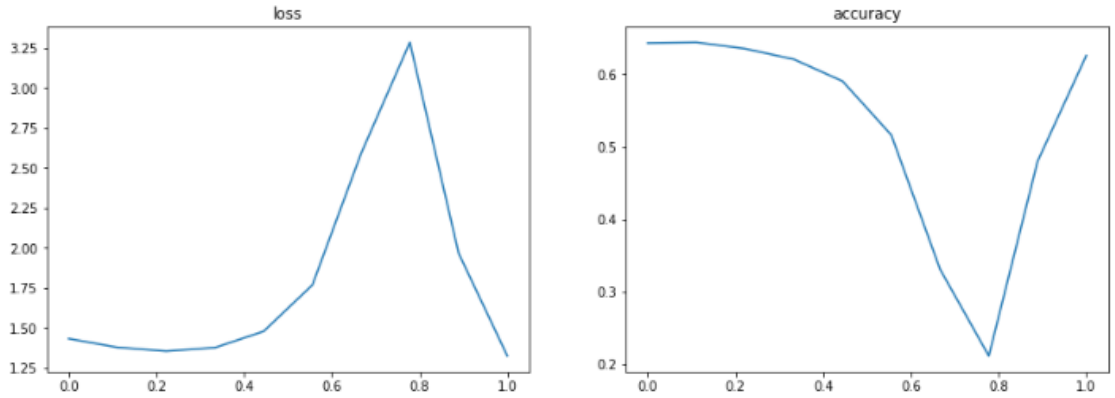

Рис. 5.5: Интерполяция между необрезанной моделью и моделью обрезанной на 0.99 с расписанием

Список литературы

[1] Ilya Sutskever Alex Krizhevsky и Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[2] Jonathan Frankle и Michael Carbin. *The lottery tickets hypothesis: finding sparse, trainable neural networks*. URL: https://arxiv.org/pdf/1803.03635.pdf.

[3] Oriol Vinyals Geoffrey Hinton и Jeff Dean. *Distilling the Knowledge in a Neural Network*. URL: http://www.cs.toronto.edu/~hinton/absps/distillation.pdf.

[4] Shaoqing Ren Kaiming He Xiangyu Zhang и Jian Sun. *Deep Residual Learning for Image Recognition*. URL: https://arxiv.org/pdf/1512.03385.pdf.

[5] Yoshua Bengio Matthieu Courbariaux и Jean-Pierre David. *Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference*. URL: https://arxiv.org/pdf/1712.05877.pdf.

[6] Yoshua Bengio Yann LeCun Léon Bottou и Patrick Haffner. *Gradient-based learning applied to document recognition*. URL: https://arxiv.org/pdf/1803.03635.pdf.