Московский государственный университет
имени М.В. Ломоносова
Механико-математический факультет
Кафедра математической теории
интеллектуальных систем

Исследование влияния регуляризации на обучение нейронных сетей в случае динамического параметра dropout

Курсовая работа
Новосада Виктора Олеговича
507 группа

Научный руководитель:
научный сотрудник
кафедры математической
теории интеллектуальных систем
к.ф.-м.н. Половников Владимир Сергеевич

Москва, 2020

Аннотация

Одна из наиболее распространенных проблем, с которой сталкиваются специалисты в области машинного обучения, заключается в том, чтобы избежать чрезмерной подгонки результатов модели под тренировочную выборку. Одним из многих методов снижения переобучения является регуляризация. Многие стандартные методы, такие как l1 и l2 регуляризации зачастую показывают улучшение результатов в обучении. Однако этим методам не хватает способности к самоадаптации на протяжении всего обучения, то есть сила регуляризации фиксируется в заранее определенном разбросе или вообще берется константой, и для адаптации к различным сетевым архитектурам требуются ручные настройки. В настоящей работе мы предлагаем метод динамической регуляризации. В частности, приведены результаты использования динамического параметра dropout регуляризации. Мы моделируем силу регуляризации как еще один дополнительный параметр в обучении, изменяемый при обратном распространении ошибки вместе с весами модели. Таким образом, даже при не самом подходящем соотношении размера сети и мощности выборки можно наблюдать улучшения не только в обобщении модели, но и в результатах ее обучения. Экспериментальные результаты показывают, что предложенный метод может улучшить возможности обобщения на готовых нейросетевых архитектурах и превзойти стандартные методы регуляризации.

1 Введение

Сверточные нейронные сети (CNN) - сети, использующие последовательные операции свертки с последующей нелинейной активацией (например, ReLU) для извлечения высокоуровневых признаков выборки, достигли высоких результатов и даже вытеснили большинство стандартных методов в области визуальных данных [1–3] . Последние достижения архитектур CNN, такие как ResNet [2], DenseNet [4], ResNeXt [5], облегчают проблему исчезающего градиента и повышают производительность. Однако CNN по-прежнему подвержены проблеме переобучения, что снижает их способность к обобщению. Для снижения переобучения и уменьшения ошибки обобщения использовались самые разнообразные стратегии регуляризации. Увеличение объема данных - это простой, но эффективный способ улучшить разнообразие обучающих данных. Пакетная нормализация [7] стандартизирует среднее и дисперсию признаков каждой небольшой партии данных, что делает оптимизацию более гладкой. Ансамбли нейронных сетей с различными конфигурациями моделей, как известно, уменьшают переобучение, но требуют дополнительных вычислительных затрат на обучение и поддержание нескольких моделей. Одна модель может быть использована для моделирования наличия большого числа различных сетевых архитектур путем случайного отброса нейронов во время обучения. Dropout предлагает очень дешевый с вычислительной точки зрения и удивительно эффективный метод регуляризации для уменьшения перенапряжения и улучшения ошибки обобщения в глубоких нейронных сетях. Однако во всех моделях параметр регуляризации задается заранее в качестве гиперпараметра модели, который чаще всего подбирается кроссвалидацией, что, в теории, сильно ухудшает способность модели адаптироваться. Учитывая эти проблемы, мы предлагаем метод динамической регуляризации, в котором сила регуляризации адаптируется к изменению функции потерь во время обучения. Аналогично человеческому образованию, регуляризатор рассматривается как инструктор, который постепенно увеличивает трудность обучения примерам в форме возмущения признаков. Динамическая регуляризация может адаптироваться к различным размерам модели. На Рис. 1 показана предложенная динамическая регуляризация в структуре ResNet. Функция потери используется не только для выполнения обратного распространения, но и используется для обновления амплитуды регуляризации. Регуляризатор работает как возмущение, которое вносит увеличение в про-

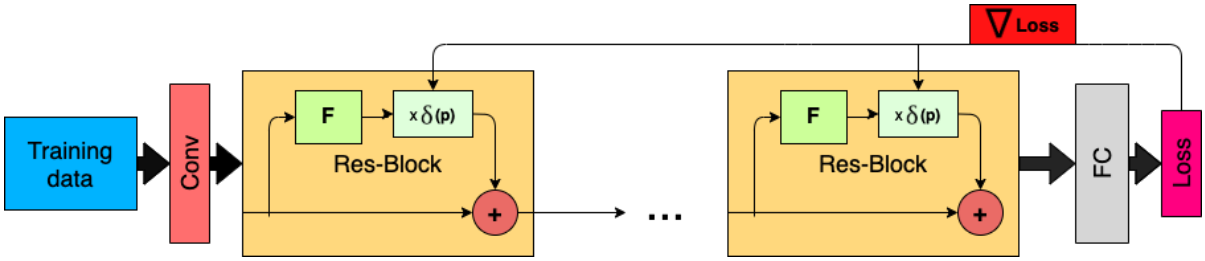

Рис. 1: Архитектура ResNet с динамическим dropout.

странство признаков, поэтому CNN обучаются разнообразию расширенных признаков. Кроме того, амплитуда регуляризации изменчива по отношению к изменению потерь на обучение. Мы провели эксперименты по задаче классификации изображений, чтобы оценить нашу стратегию регуляризации. Экспериментальные результаты показывают, что предложенная динамическая регуляризация превосходит результаты аналогичной модели, но без регуляризации, или со статическим параметром регуляризации. Однако, вопреки логичным рассуждениям о том, что сила регуляризации должна постоянно расти по мере обучения модели - эксперименты показывают, что данное утверждение является правдой только в случае нехватки тренировочных данных, а также в самом начале обучения.

2 Связанные работы

2.1 Глубокие CNN

Вместе с увеличением производительности машин - CNN также стали глубже и шире [2, 4, 6]. Поскольку предлагаемая регуляризация основана на ResNet, ниже приведено краткое рассмотрение базовой структуры данной сети, так называемый остаточный блок. Остаточный блок. Остаточный блок (Res-Block, показанный на Рис. 1) определяется следующим образом:

$$x_k + 1 = x_k + F(x_k, W_k)$$

Где x_k - это входные признаки k-го Res-блока, с добавлением остаточной ветви F, представляющей собой нелинейное преобразование между x_k и набором параметров W_k (W_k будет опущен для простоты ниже). F состоит из двух архитектур $Conv - BN - ReLU$ или $Bottleneck$ в исходной структуре ResNet [2].

2.2 Регуляризация

В дополнение к достижениям сетевых архитектур, многие методы регуляризации, например, увеличение объема данных [1, 11], стохастическое падение [8–10], и тд, были успешно применены, чтобы избежать переобучения CNN моделей. Увеличение объема данных (например, случайное обрезание, переворачивание и настройка цвета [1]) - это простая, но эффективная стратегия увеличения разнообразия данных. Dropout [8] - это метод регуляризации, который аппроксимирует обучение большого числа нейронных сетей с различными архитектурами параллельно. Во время обучения некоторое количество выходных данных слоя, нейронов, случайным образом игнорируется. Это приводит к тому, что слой выглядит и обрабатывается как слой с другим числом нейронов и связей с предыдущим слоем. По сути, каждое обновление слоя во время обучения выполняется с другим “видом” настроенного слоя. На основе этой идеи было предложено большое количетсво усовершенствований [10, 12, 14, 15]. Почему

dropout является методом регуляризации? Можно попытаться объяснить это с некоторыми допущениями (для урощения) с помощью несложной математики:

1. Пусть у нас есть однослойная нейронная сеть с dropout регуляризацией с параметром p
2. Функция активации - тождественная ($f(x) = x$)
3. Функция потерь - среднеквадратичная

Тогда:

$$Loss = \frac{1}{2}\left(y - \sum_{i=1}^{n} w_i' x_i\right)^2$$

Или

$$Loss = \frac{1}{2}\left(y - \sum_{i=1}^{n} \delta_i w_i x_i\right)^2$$

Где where $\delta \quad Bernoulli(p)$, это означает, что δ равно 1 с вероятностью p и 0 иначе. Для градиентного спуска необходимо вычислить производные функции потерь по весам:

$$\frac{\partial Loss}{\partial w_i} = -y\delta_i x_i + w_i \delta_i^2 x_i^2 + \sum_{j=1, j\neq i}^{n} w_j \delta_i \delta_j x_i x_j$$

Теперь предположим, что у нас есть аналогичная сеть с весами $w' = p * w$. Тогда

$$Loss = \frac{1}{2}\left(y - \sum_{i=1}^{n} p_i w_i x_i\right)^2$$

$$\frac{\partial Loss}{\partial w_i} = -y p_i x_i + w_i p_i^2 x_i^2 + \sum_{j=1, j\neq i}^{n} w_j p_i p_j x_i x_j$$

Далее, интересная часть заключается в подсчете математического ожидания величины $\frac{\partial Loss}{\partial w_i}$ в первом случае:

$$E[\frac{\partial Loss}{\partial w_i}] = -y p_i x_i + w_i p_i^2 x_i^2 + w_i Var(\delta_i) x_i^2 + \sum_{j=1, j\neq i}^{n} w_j p_i p_j x_i x_j =$$
$$= \frac{\partial Loss}{\partial w_i} + w_i Var(\delta_i) x_i^2 = \frac{\partial Loss}{\partial w_i} + w_i p_i (1 - p_i) x_i^2$$

Отсюда можно увидеть, что в среднем градиент функции потерь с dropout равен градиенту функции потерь сети с весами $w' = p * w$.

3 Предложенный метод

Как уже упоминалось выше, фиксированнный параметр регуляризации в существующих методах регуляризации, отходит от парадигмы человеческого обучения. Одним из способов добавить "динамику"в изменение параметра регуляризации является предварительное определение схемы обновления параметра регуляризации, такого как схема линейного приращения в [9, 16], которая линейно увеличивает параметр регуляризации от низкого до высокого. Как будет видно из результатов - заранее определенный график недостаточно гибок, чтобы

выявить процесс обучения. Исходя из того, что потеря обучающей системы может полностью обеспечить статус обучения, мы предлагаем динамическую регуляризацию, которая способна адаптивно регулировать силу регуляризации. Наша динамическая регуляризация для CNN использует динамику функции потерь при обучении. То есть в начале обучения и тренировочные, и тестовые потери продолжают снижаться. Через определенное число итераций сеть перекрывает обучающие данные, в результате чего потери при обучении уменьшаются быстрее, чем потери при тестировании. Описываем в данной статье подход стремится уменьшить разброс между точностью на тренировочной выборке и точностью на тестовой выборке. Как уже было показано выше на Рис. 1, для результатов данной статьи использовалась структура сети ResNet [2] с добавленными слоями dropout в каждом Res-Block. Обучение параметра dropout происходит ровно так же как и обучение весов модели, то есть через градиентный спуск.

4 Вычислительные эксперименты

4.1 Детали реализации

Эксперименты проводились на базе данных $cifar10$. Параметры модели были выбраны следующим образом:

1. Коэффициент обучения - 0.001
2. Функция оптимизации - *Adam*
3. Функция потерь - *categorical crossentropy*
4. Размер пакета - 128
5. Количество эпох - 50
6. Стартовый параметр dropout в обоих случаях (статики и динамики) определялся с помощью кросс-валидации с шагом 0.1, от 0.2 до 0.7

Модель и все вспомогательные классы/функции были реализованы на $python3$, используя фреймворк $tensorflow.keras$. Полный код доступен на github. Обучение происходило на *google colab* с одним GPU.

4.2 Сравнение результатов

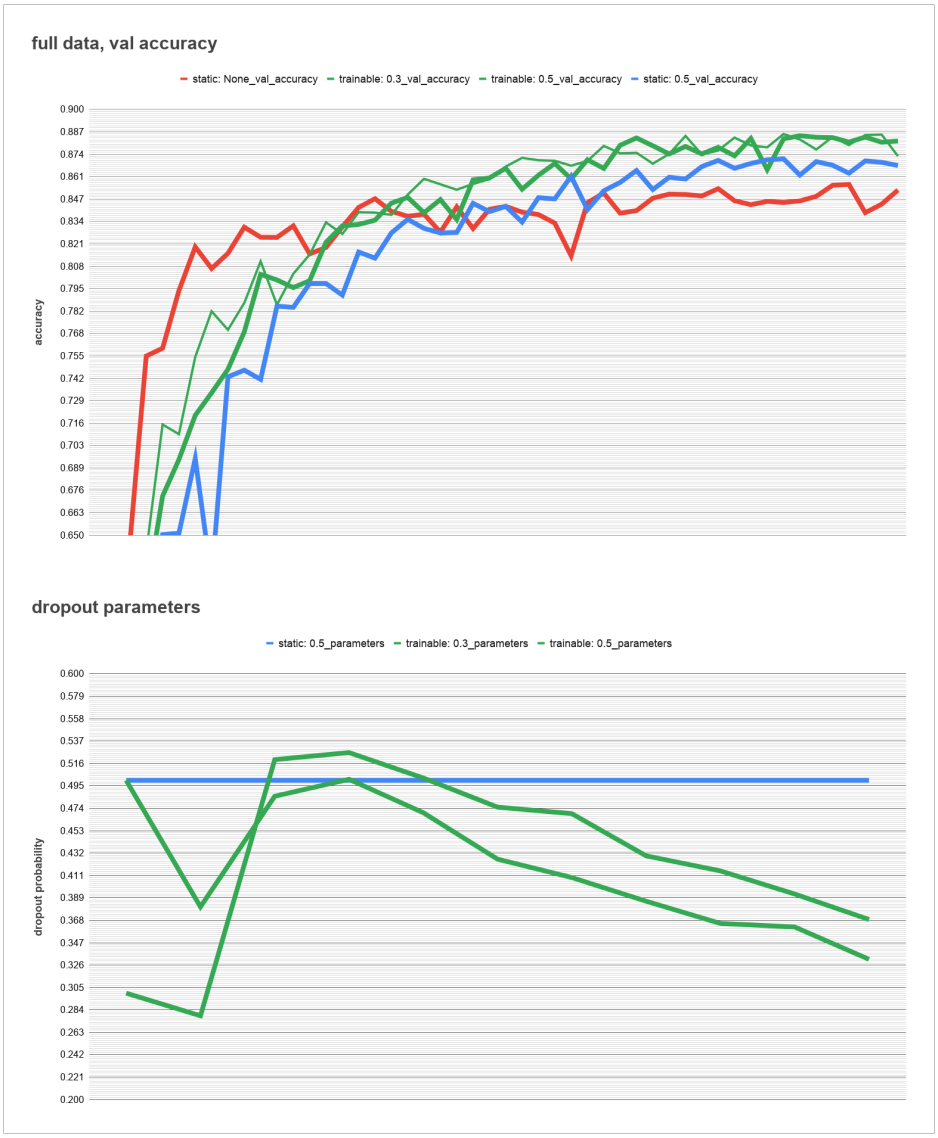

Рис. 2: Лучшие результаты трех вариантов сетей на полном обучающем наборе данных (красное - без dropout, синее - статический dropout, зеленое - предложенный вариант динамического dropout.

В результате экспериментов, лучшими начальными параметрами dropout оказались:

1. Для статической модели - 0.3
2. Для динамической модели - 0.5 или 0.3

Как видно из Рис. 2, использование статической регуляризации дает результат лучше, чем аналогичная модель без регуляризации, а использование динамической регуляризации - лучше, чем статической регуляризации. Если выписать результаты в цифрах, то получится следующее:

Тип регуляризации	Параметр dropout	Точность	Мин. dropout	Макс. dropout
База (без регуляризации)	-	85.6%	-	-
Статический dropout	0.3	87%	0.3	0.3
Динамический dropout	0.5	88.5	0.33	0.5%

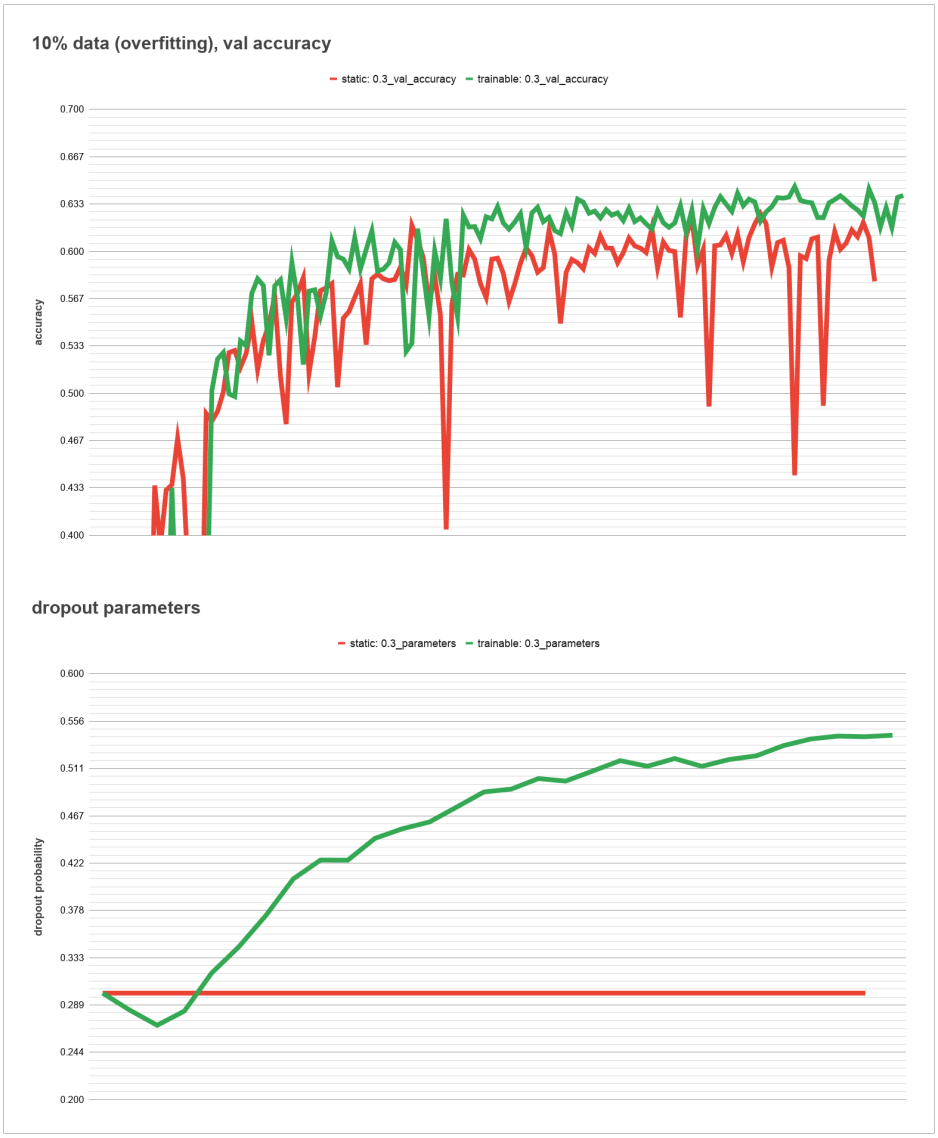

Рис. 3: Лучшие результаты трех вариантов сетей на полном обучающем наборе данных (красное - без dropout, синее - статический dropout, зеленое - предложенный вариант динамического dropout.

Также был проведен эксперимент на оценку "логичности"данного подхода. При переобучении модели - параметр регуляризации должен постепенно увеличиваться, чтобы ослабить переобучение. Для этого было сильно урезано количество обычающих данных (до 10% от общего количества, 5к изображений) и обучены две модели: одна со статическим параметром, другая с динамическим. Как видно из Рис. 3, хотя точность на тестовой выборке возросла незначительно, но видно сильное стремление параметра dropout к 1, что как раз соответствует нашим ожиданиям.

5 Вывод

Выше мы представили архитектуру сети с динамическим параметром регуляризации, изменяющимся в зависимости от изменения функции потери во время обучения. Данный прием можно использовать не только для ResNet [2] или сверточных сетей, а для любой другой нейросетевой архитектуры. Хотя и было проведено немного экспериментов, и необходимо

провести эксперименты на других сетях и других, более больших, базах данных, все равно виден потенциал данного приема (особенно по сравнению с базовыми методами), а также открыта новая область для изучения.

Список литературы

[1] A. Krizhevsky, I. Sutskever, G. E Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012

[2] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016

[3] Z.-Q. Zhao, P. Zheng, S.-tao Xu, X. Wu, "Object detection with deep learning: a review," IEEE Transactions on Neural Networks and Learning Systems, 2019

[4] G. Huang, Z. Liu, L. Van Der Maaten, K. Q Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017

[5] S. Xie, Ross G., P. Dollar, Z. Tu, K. He, "Aggregated residual transformations for deep neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017

[6] D. Han, J. Kim, J. Kim, "Deep pyramidal residual networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017

[7] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, 2015

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014

[9] G. Ghiasi, T.-Yi Lin, Q. V Le, "Dropblock: A regularization method for convolutional networks," in Advances in Neural Information Processing Systems, 2018

[10] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Q Weinberger, "Deep networks with stochastic depth," in European Conference on Computer Vision. Springer, 2016

[11] T. DeVries G. W Taylor, "Improved regularization of convolutional neural networks with cutout," CoRR, vol. abs/1708.04552, 2017

[12] G. Larsson, M. Maire, G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," in International Conference on Learning Representations, 2017

[13] P. Morerio, J. Cavazza, R. Volpi, R. Vidal, V. Murino, "Curriculum dropout," in Proceedings of the IEEE International Conference on Computer Vision, 2017

[14] Ian J Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, "Maxout networks," in International Conference on Machine Learning, 2013

[15] X. Shen, X. Tian, T. Liu, F. Xu, D. Tao, "Continuous dropout," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 9

[16] B. Zoph, V. Vasudevan, J. Shlens, Q. V Le, "Learning transferable architectures for scalable image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern, Recognition, 2018