

**ВЛИЯНИЕ ВЫБОРА ГИПЕРПАРАМЕТРОВ ПРИ ОБУЧЕНИИ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ НА ТОЧНОСТЬ ПОВТОРНОЙ ИДЕНТИФИКАЦИИ ЛЮДЕЙ В СИСТЕМАХ ВИДЕОНАБЛЮДЕНИЯ**

**С. А. ИГНАТЬЕВА**

*(Полоцкий государственный университет, Беларусь)*

***Аннотация.** В данной работе исследуется влияние на точность нейросетевого алгоритма повторной идентификации выбора гиперпараметров, таких как скорость обучения и размер пакета. Оценивается их воздействие на динамику тренировки и определяются наиболее оптимальные значения для сверточной нейронной сети ResNet-51 при использовании набора данных Market1501.*

***Ключевые слова:** скорость обучения, размер пакета, ResNet-50.*

При решении задачи повторной идентификации людей наиболее эффективным способом извлечения признаков на изображении человека является использование сверточных нейронных сетей (СНС). Для построения надежного дескриптора предварительно необходимо выполнить обучение СНС, на качество которого будут оказывать влияние обучающая выборка и гиперпараметры. Под гиперпараметрами понимают установленные перед началом тренировки настройки, такие как количество эпох, скорость обучения, размер пакета. От выбранных значений будет зависеть точность алгоритма повторной идентификации и время, затраченное на обучение.

Количество эпох определяет сколько раз каждое изображение обучающей выборки пройдет по сети. Если количество эпох невелико, то модель окажется недообученной, и точность повторной идентификации будет низкой. Слишком большое количество эпох может привести к переобучению, когда сеть запомнит все рассмотренные изображения, и не сможет работать на тестовых примерах. Для задачи повторной идентификации тренировка обычно выполняется в течение 60–100 эпох.

Изображения подаются на вход сети пакетами, обычно от 16 до 64, и увеличение размера пакета обусловлено стремлением к распараллеливанию вычислений, т.к. это позволяет сократить время, затраченное на обучение, но при этом снижается точность работы обученной модели. В [1] предлагается подход, согласно которому в процессе обучения постепенно увеличивается размер пакета. Это позволяет получить точность, близкую к той, что получена при использовании фиксированного небольшого пакета, при этом время обучения снижается.

В [2] результаты исследований также показывают, что увеличение размера пакета при обучении приводит к снижению точности работы алгоритма. Так, при сравнении пакетов с размерами 64 и 32, в первом случае точность Rank1 и mAP оказывается ниже. Кроме того, обучение на нескольких GPU показывает более низкие результаты точности, чем при использовании одного GPU.

Скорость обучения показывает, на сколько изменяются весовые коэффициенты при каждом обновлении весов. Для повышения эффективности тренировки используют планировщики скорости, которые позволяют изменять ее значения в процессе обучения по истечении определенного промежутка времени или по определенным критериям.

В [3] рассматривается механизм снижения скорости ADEL, который отслеживает степень изменения весов, и каждый раз, когда они перестают изменяться скачкообразно, скорость обучения уменьшается. Это позволяет обеспечить более быструю сходимость модели.

В [4] предлагается динамический механизм, состоящий из трех режимов изменения скорости обучения  $\eta$ , которые зависят от кривизны  $\lambda_0$  поверхности функции потерь. Первая фаза – медленная (lazy phase), когда скорость обучения имеет относительно небольшое значение  $\eta > \frac{2}{\lambda_0}$ , из-за чего с шагом  $t$  изменение

скорости остается практически постоянным в начале обучения. Вторая – быстрая фаза (catapult phase), где скорость обучения принимает значения  $\frac{2}{\lambda_0} < \eta < \eta_{\max}$ .

На этом этапе наблюдается экспоненциальный рост потерь, и при этом быстрое уменьшение кривизны  $\lambda$  до тех пор, пока не стабилизируется на значении  $\lambda_{final} < \frac{\eta}{2}$ . Как только выполняется это условие, достигается плоский минимум.

И последняя – фаза расхождения (divergent phase), когда скорость обучения превышает значение  $\eta_{\max}$  и модель перестает обучаться. Кроме того, в [4] выдвигается предположение, которое затем подтверждается исследованиями, что использование больших скоростей обучения позволяет находить плоские минимумы, которые обобщаются лучше, чем резкие минимумы. К этому же, по мнению авторов, приводит и использование небольших пакетов для обучения.

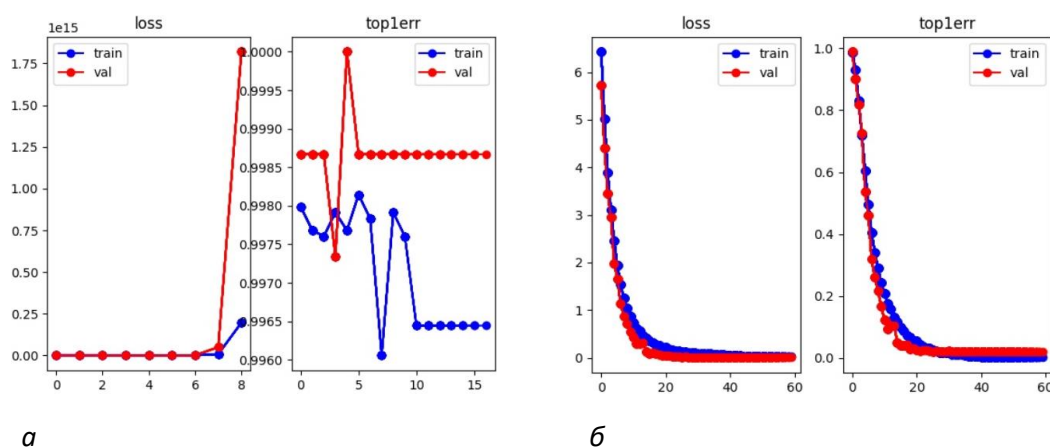
Для определения наиболее оптимального размера пакета и скорости обучения использовался алгоритм повторной идентификации [5] реализованная на фреймворке pyTorch. В качестве базовой СНС для извлечения признаков использовалась СНС ResNet-50. Первый этап экспериментов направлен на исследовании влияния скорости обучения на точность ре-идентификации, и для этого начальная скорость устанавливалась равной от 0,01 до 0,2, и после 40-й эпохи уменьшалась в 0,01 раз. Обучение осуществлялось на наборе данных Market1501[6] с размером

пакета 16 в течении 60 эпох на персональном компьютере с основными характеристиками: Intel Core i5 3.11 GHz, 16 Gb RAM, Nvidia GeForce RTX-3060 6 Gb. Результаты приведены в таблице 1.

Таблица 1. – Влияние скорости обучения на точность повторной идентификации

Скорость обучения	Rank1	Rank5	Rank10	mAP	Время обучения
0,01	72,5059	88,6283	92,2830	46,8359	87м 31с
0,03	81,3242	92,4287	95,0416	57,6058	88м 02с
0,05	81,7696	92,3990	95,3682	59,5859	87м 56с
0,07	82,8979	93,2007	95,3385	59,7440	87м 20с
0,09	82,5713	92,7850	95,6354	60,1919	88м 12с
0,1	82,6990	93,0226	95,3682	61,7585	88м 09с

Результаты экспериментов показали, что увеличение скорости обучения позволяет повысить точность повторной идентификации в метрике Rank1 более чем на 10%, и почти на 5% в метрике mAP, при этом время тренировки составляет около 88 минут и не зависит от выбранной скорости. Увеличение скорости обучения до 0,2 привело к взрывным градиентам, явлению, когда веса сети обновляются слишком быстро, а значение функции потерь сильно увеличивается. На рисунке 1, а видно, что потери принимают значения порядка  $10^{15}$  уже после 7 эпохи, и далее продолжают увеличиваться, что приводит к невозможности дальнейшего обучения модели. Для сравнения приведены графики обучения при скорости 0,1 (рисунок 1, б), где потери и ошибка top-1 уменьшаются в процессе обучения.



а – 0,2; б – 0,1

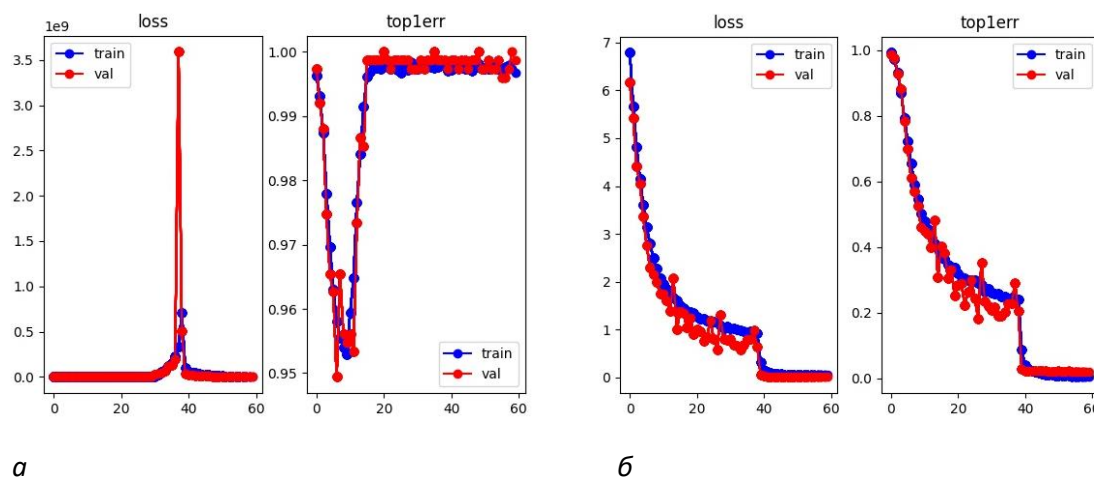
Рисунок 1. – Графики обучения модели повторной идентификации со скоростью

Для исследования влияния размера пакета проведен второй этап экспериментов, при котором обучение осуществлялось в течении 60 эпох с начальной скоростью 0,09, которая была уменьшена после 40 эпохи в 0,01 раз. Результаты экспериментов с разным размером пакета приведены в таблице 2.

Таблица 2. – Влияние размера пакета изображений при обучении на точность повторной идентификации

Размер пакета	Rank1	Rank5	Rank10	mAP	Время обучения
8	0,02969	0,08314	0,18409	0,02020	104м 45с
16	82,5713	92,7850	95,6354	60,1919	88м 12с
32	80,0178	91,7755	94,8634	56,9912	80м 08с
64	75,7423	90,4691	94,0024	51,0595	77м 10с

Результаты эксперимента, представленные в таблице 2 показывают, что увеличение размера пакета позволяет уменьшить время обучения, однако при этом снижается точность модели. При использовании пакета из 8 изображений точность повторной идентификации меньше 1%, что говорит о том, что модель не смогла обучиться. На рисунке 2, а представлен график обучения с размером пакета 8, из которого видно, что в начале тренировки ошибка Top1 уменьшалась, однако 8 изображений оказалось мало, и не позволило получить сети достаточную обобщающую способность, и привело к резкому увеличению потерь. Корректировка скорости после 40-й эпохи позволила уменьшить значение функции потерь, однако этого оказалось недостаточно, и их значение осталось велико (около  $2 \cdot 10^6$ ), при начальном значении равном 8,24. Масштаб графика потерь не позволяет отразить эти изменения из-за большого диапазона значений, однако график ошибки Top1 указывает, что рост потерь начался после 10 эпохи. На рисунке 2, б приведен график обучения с размером пакета 16, при котором удалось получить наилучшие показатели точности модели ре-идентификации. Резкое уменьшение значения потерь и ошибки Top1 связано со снижением скорости обучения в 0,01 раз после 40-й эпохи, что обеспечило более плавную корректировку весов при поиске локального минимума.



а – 8; б – 16

Рисунок 2. – Графики обучения модели повторной идентификации с размером пакета

Таким образом установлено, что наиболее оптимальными параметрами для обучения на наборе данных Market1501 является размер пакета 16 и скорость обучения 0,09. Повысить точность алгоритма повторной идентификации можно так же применяя корректировку скорости обучения в процессе тренировки.

#### ЛИТЕРАТУРА

1. Lee, S., Kang, Q., Madireddy, S., Balaprakash, P., Agrawal, A., Choudhary, A.N., Archibald, R., & Liao, W. (2019). Improving Scalability of Parallel CNN Training by Adjusting Mini-Batch Size at Run-Time. 2019 IEEE International Conference on Big Data (Big Data), 830–839. DOI: 10.1109/Big-Data47090.2019.9006550.
2. Person-reid-GAN-pytorch. URL: <https://github.com/qiaoguan/Person-reid-GAN-pytorch>.
3. Lewkowycz, A. (2021). How to decay your learning rate. ArXiv, abs/2103.12682.
4. Lewkowycz, A., Bahri, Y., Dyer, E., Sohl-Dickstein, J., & Gur-Ari, G. (2020). The large learning rate phase of deep learning: the catapult mechanism. ArXiv, abs/2003.02218.
5. Person reID baseline PyTorch. URL: [https://github.com/layumi/Person\\_reID\\_baseline\\_pytorch](https://github.com/layumi/Person_reID_baseline_pytorch)
6. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable Person Re-identification: A Benchmark. 2015 IEEE International Conference on Computer Vision (ICCV), 1116-1124. DOI: 10.1109/ICCV.2015.133.