

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

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет по большому домашнему заданию по курсу Deep Learning на тему:
Классификация изображений

Выполнил студент:

группы #БПМИ223, 3 курса

Тань Сипэн

Содержание

1	Введение	3
2	BasicBlockNet	4
3	ResNet50 - 1 этап	5
4	ResNet50 - 2 этап	5
5	ResNet50 - 3 этап	6
6	ResNet50 - Дообучение	7

1 Введение

Данную работу я поделю на 3 основные части:

- Функции для чтения, обучения, тестирования данных. Часть из них взято из семинара и дз.
- Простая модель, которая должна пройти порог 10
- ResNet50, потрачено больше всего времени (перебор параметров, трансформеры, дообучение)

В итоге ассигасу остановилась на **52.12%**, всего было потрачено 156 часов, и я надеюсь моему ноуту было все хорошо.

2 BasicBlockNet

- **Модель:** Данная модель взята из МДЗ2 - представляет собой простую архитектуру с остаточной связью, которая подходит для небольших изображений и задач классификации. Она сочетает в себе сверточные слои, Batch Normalization, активации ReLU, остаточные связи и пулинг, что делает ее достаточно эффективной для базовых задач.
- **Transforms:** RandomHorizontalFlip и Normalization
- **Optimizer:** `optim.SGD(net.parameters(), lr=0.1, momentum=0.9)`
- **Scheduler:** добавил MultiStepLR, но ассурасу только достигла до 6%. Решил убрать и все стало хорошо, может быть связано с тем что слишком простая модель и мало эпох.

Поменяя некоторые параметры, я начал обучать на 20 эпох и получил ассурасу 20%, даже в два раза больше чем порог контрольной точки. И теперь у меня каникулы.

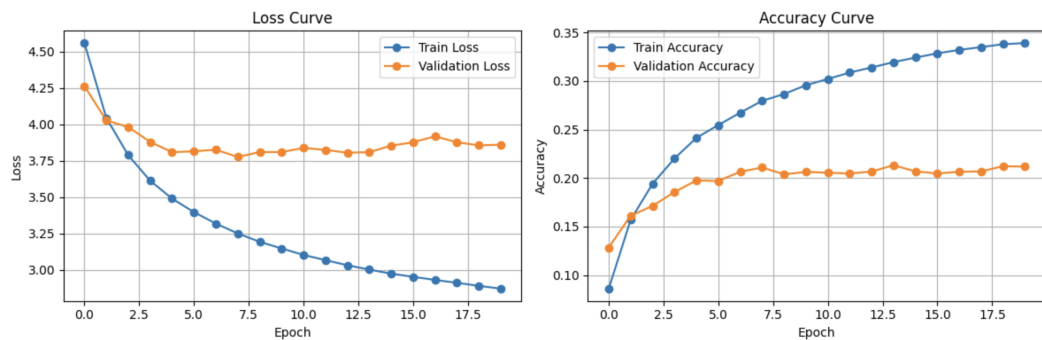


Рис. 2.1: Результаты BasicBlockNet

Я вернулся и не придумал какие то новые модели, поэтому решил обучать модель на больших эпохах - 50, в принципе было быстро но модель сильно переобучилась и качество стало еще ниже. Значит пора придумывать более сложные модели.

3 ResNet50 - 1 этап

- **Модель:** Обычный ResNet50, в модели ничего не менял, хотел посмотреть на результаты.
- **Transforms:** RandomHorizontalFlip, RandomVerticalFlip - я выбирал между ними, но оказывается качество когда как, интуитивно не получится выбрать лучший, поэтому поставил их на рандоме. И конечно же поменял Normalization с равным распределением на более лучшие значения.
- **Optimizer:** Вот тут поинтереснее, я выбирал между SGD и Adam. Сравнил их с одинаковыми условиями, Adam обучается очень медленно и качество немного хуже чем SGD, и я решил что буду использовать только `optim.SGD(net.parameters(), lr=0.1, momentum=0.9)`.
- **Scheduler:** Так как в BasicBlockNet у меня с использованием scheduler стало хуже, и модель пока что еще не совсем сложная, то я временно решил забить на него.

В итоге у меня получился **accuracy 20%**, как так? Даже хуже чем базовую модель, теперь придется много менять в модели.

4 ResNet50 - 2 этап

- **Модель:** Так, что же можно поменять в модели, мне в голову первое пришло - это `kernelseize`, `stride`, `padding`... интуитивно уменьшил их размеры. И конечно же убрал Maxpooling, так как картинки и так уже маленькие, нельзя больше терять.
- **Transforms:** Выбирал между - RandomGrayscale, GaussianBlur, ColorJitter, оказывается каждый раз ColorJitter ухудшает оценку (хотя в мдз он выдавал неплохое качество).
- **Optimizer:** Добавил `weightdecay`, `optim.SGD(net.parameters(), lr=0.1, momentum=0.9, weightdecay=0.0001)`
- **Scheduler:** Пора его добавить, выбирал между ReduceLROnPlateau и MultiStepLR, первый дал лучший результат.

В итоге результат действительно стало лучше, но после 10 эпохи качество остановилась улучшаться и остановилась на 30%.

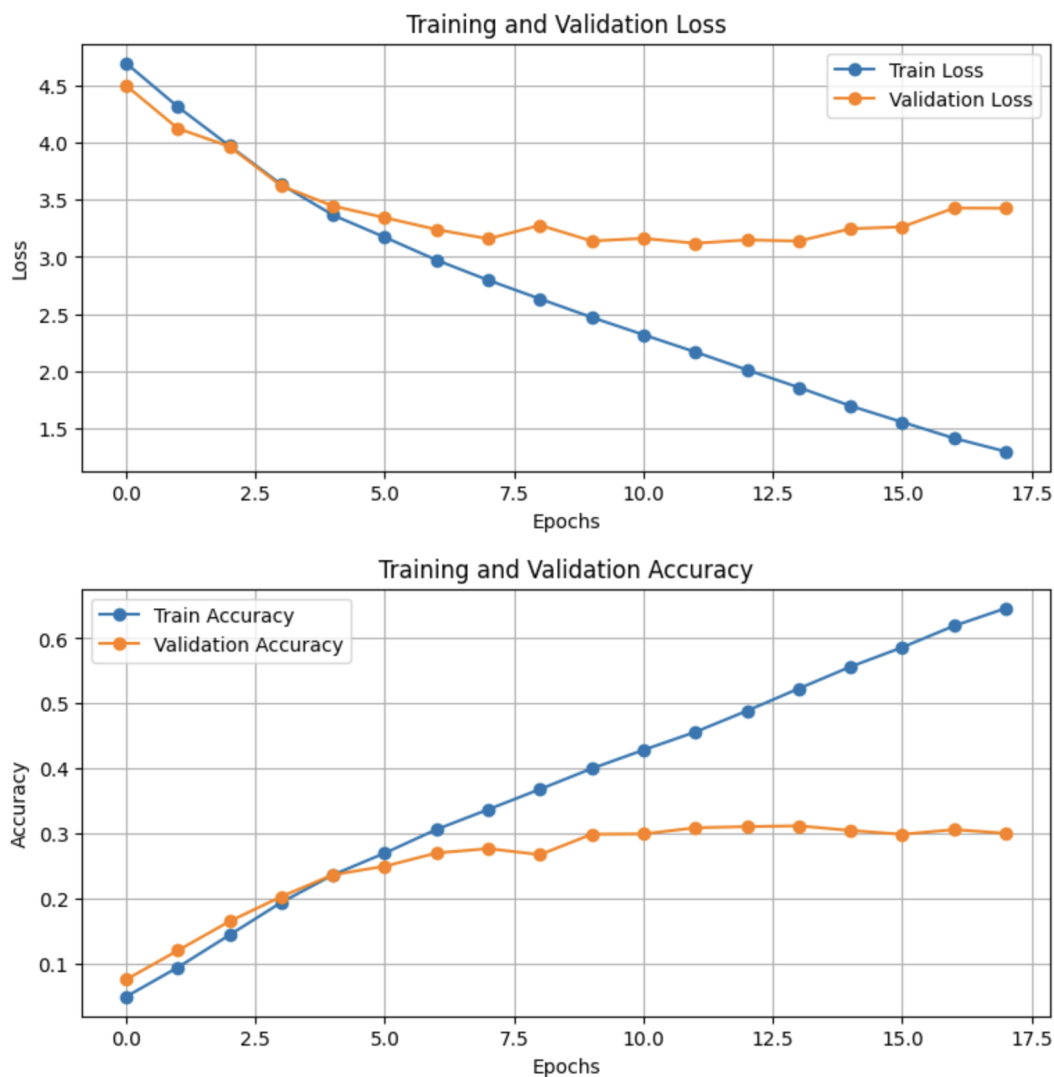


Рис. 4.1

5 ResNet50 - 3 этап

- **Модель:** Видимо, проблемы с регуляризацией, так как на маленьком эпохе уже переобучилась, тогда надо добавить в модель dropout, так как тут перебором очень сложно подобрать (потому что надо много эпох, чтобы понять какой лучше), то я выбрал сначала 0.5. Чем меньше значение тем быстрее модель сходится, но наверно и качество будет не очень.
- **Transforms:** Добавил RandomErasing и стало лучше.
- **Optimizer:** Оставил такой же, `optim.SGD(net.parameters(), lr=0.1, momentum=0.9, weight_decay=0.0001)`.
- **Scheduler:** Добавил в `ReduceLROnPlateau` новые признаки - `patience` и `cooldown`, хорошие штуки, контролируют модель LR после ухудшения.

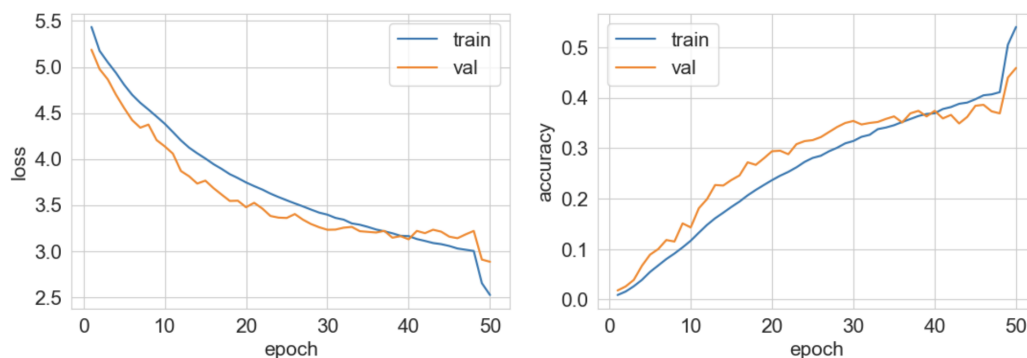


Рис. 5.1: Ура модель достиг до 40%

Несложно увидеть, что у модели появился "прыжок" и еще не закончилась, поэтому я решил поставить на 100 эпох, и сделать dropout 0.7, я считаю что 0.5 все таки мало, надо побольше чтобы помедленнее сходилась. И получил:

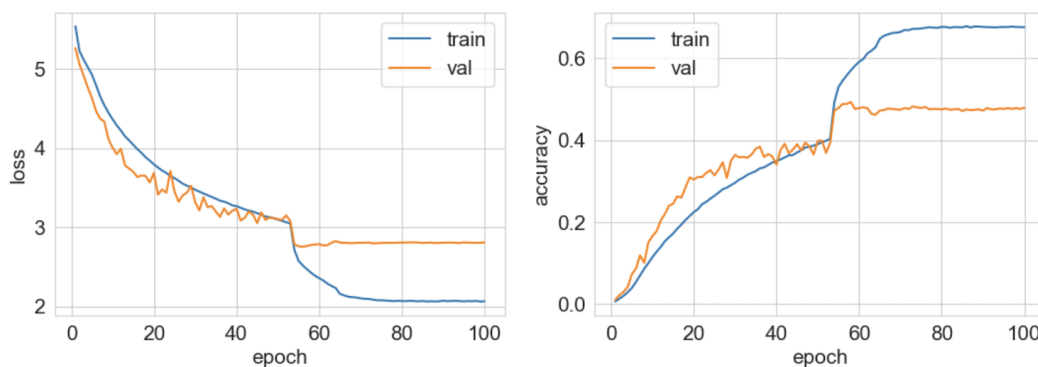


Рис. 5.2: Ассурасу на тесте 45%

6 ResNet50 - Дообучение

Теперь пришла самая сложная часть - после того как получил 45% я расслабился, думал поменять там какие то параметры, поменять transform и все добыю до 48%, но я на это потратил 3 дня (по 20ч) и никак не могу пробить порог, наверно уже ясно - придется дообучить модель, но до того как я додумался до этого, следующий день уже был дедлайн. Я решил убрать предыдущие transforms и добавить RandAugment, и поменял размер batchsize (batchsize в данном бдз имеет большой вес, так как при изменении размера качество на много отличается, но не очень понял почему)

И вот что я получил: **51.98%** на тесте (Дадада, 0.02 не хватает до макс балла)

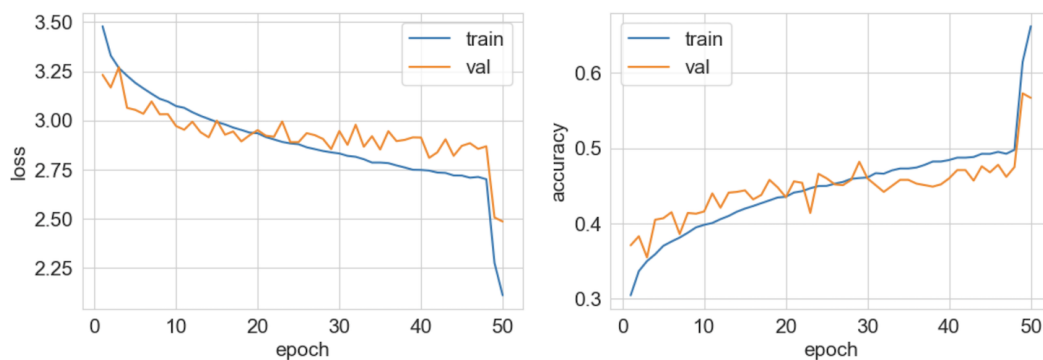


Рис. 6.1: Думаю всем в голову первое что пришло, это надо сделать побольше эпох и модель поднимется на 0.02%, поэтому дообучил до 75 эпох, но качество стало ниже.

Значит надо сделать еще одно дообучение, но до дедлайна у меня остался только 4ч, значит могу только обучать на маленькtx эпохах. И у меня начался перебор параметров на маленьких эпохах, в итоге я поменял lr на 0.2, и пробил 52% (думаю было логично менять lr, так как модель должна быстро обучаться на маленьких эпохах).

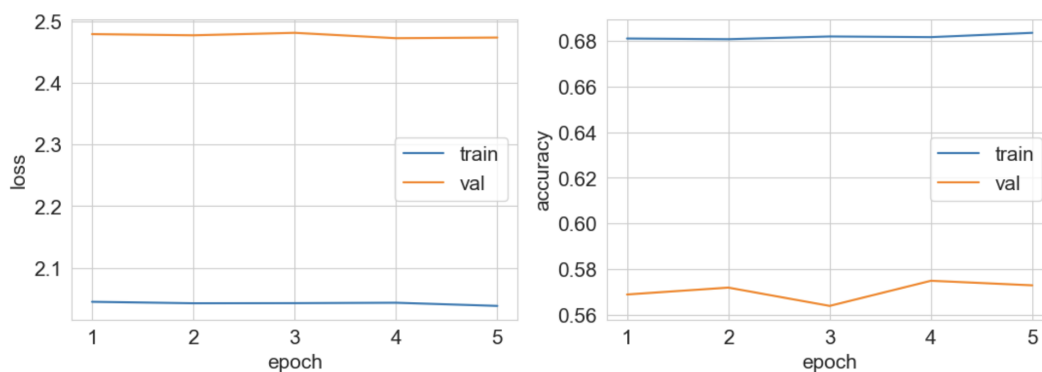


Рис. 6.2: На тесте это **52.12%**