

Глубинное обучение

Большое домашнее задание 1

Разин Арслан

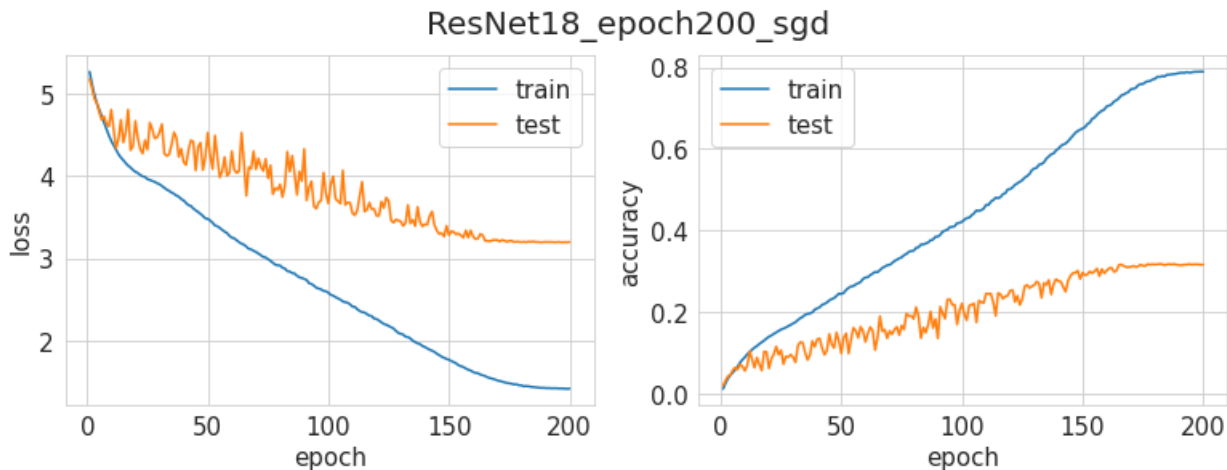
БПМИ 202

December 18, 2022

Был обычный учебный день, студент *A* закрывал очередной дедлайн, но хотел он большего – познать тайны глубинного обучения. На его счастье великий мудрец *I* объявил конкурс на создание архитектуры нейронной сети для классификации изображений. *A* помогали ему в этом великие магистры *L* и *D*. И выдали они студентам ключ от врат **Датасферы**, дабы мощнейшие видеокарты помогли им. И запустили они соревнование на **Kaggle**, чтобы решил он, кто из студентов самый достойный.

1 Первые шаги

Начал *A* с того, что переписал код с семинара мудрейшего *I*, после чего запустил свою первую модель – **ResNet18** с параметрами: num epochs = 200, batch size = 1024, lr = 0.01, momentum = 0.7, weight decay = 0.03, а расписание было CosineAnnealingLR. Вот что вышло из этого:



А **Kaggle** выдал на это 0.2938.

2 Второе пришествие

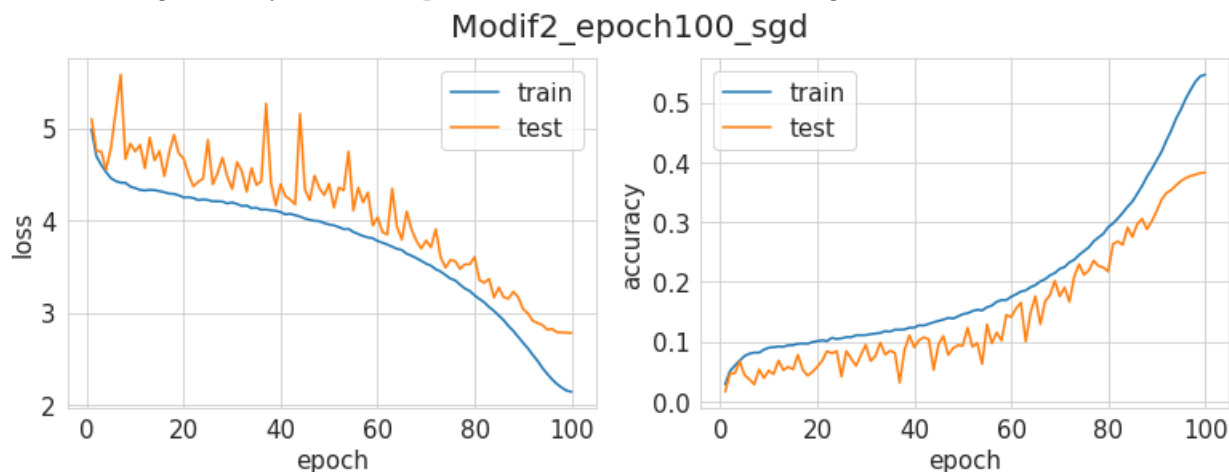
Стал думать *A*, как ему улучшить то, что он создал. Для начала он решил, что не надо так долго учить одну модель, а также стоит уменьшить штраф за веса, чтобы модель лучше училась. А ещё вспомнил он два наставления с семинаров мудрейшего *I* – дропаут перед полносвязным слоем улучшает качество, а все базовые модели торча созданы для больших картинок (то есть стоит изменить первую свёртку так, чтобы она не слишком сильно сжимала исходное изображение). Так родилась модель **Modif2** (что стало с **Modif1** история умалчивает).

```

1 model.conv1 = nn.Conv2d(in_channels = 3, out_channels = 64,
2   kernel_size = 7, padding = 2, dilation = 2)
3 model.fc = nn.Sequential(
4   nn.Dropout(inplace = True),
5   nn.Linear(in_features = 512, out_features = 200, bias = True),
6 )

```

И запустил обучение *A* с параметрами: num epochs = 100, batch size = 500, lr = 0.1, momentum = 0.9, weight decay = 0.01 и расписанием CosineAnnealingLR. Вот что вышло из этого:

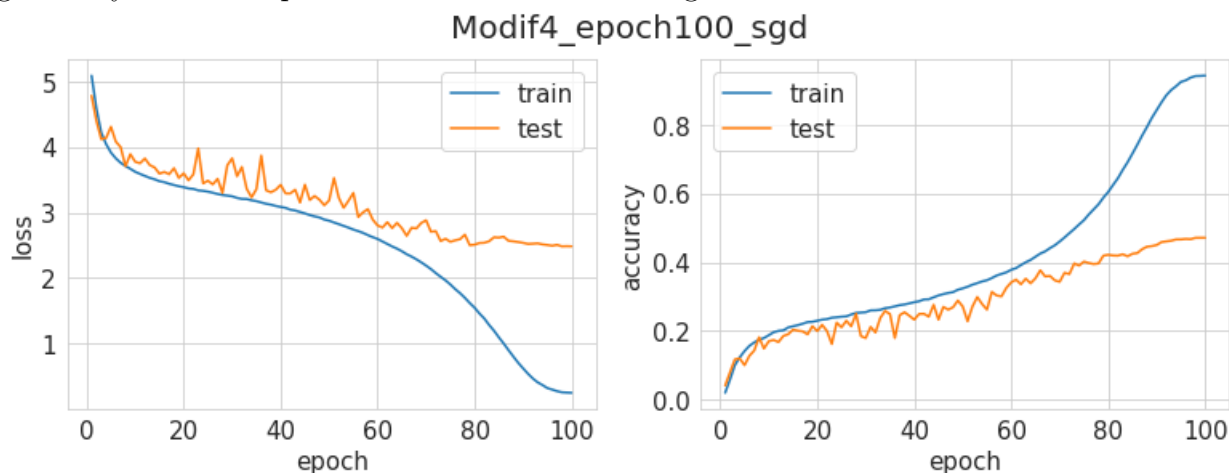


А **Kaggle** выдал на это 0.3542.

3 Третий не лишний

Ликовал *A* – он смог улучшить свой результат. Но понимал, что рано ещё останавливаться. Решил он вместо **ResNet18** взять за основу для своей модифицированной модели **ResNet34**, ведь он больше, а значит лучше. Так родилась модель **Modif4** (что стало с **Modif3** история опять же умалчивает). Решил он уменьшить размеры батча и штраф за вес, ведь у этой модели куда больше параметров, а память в **Дарасфере** была ограничена.

Запустил *A* третью попытку: num epochs = 100, batch size = 100, lr = 0.1, momentum = 0.9, weight decay = 0.001 и расписанием CosineAnnealingLR. Вот что вышло из этого:



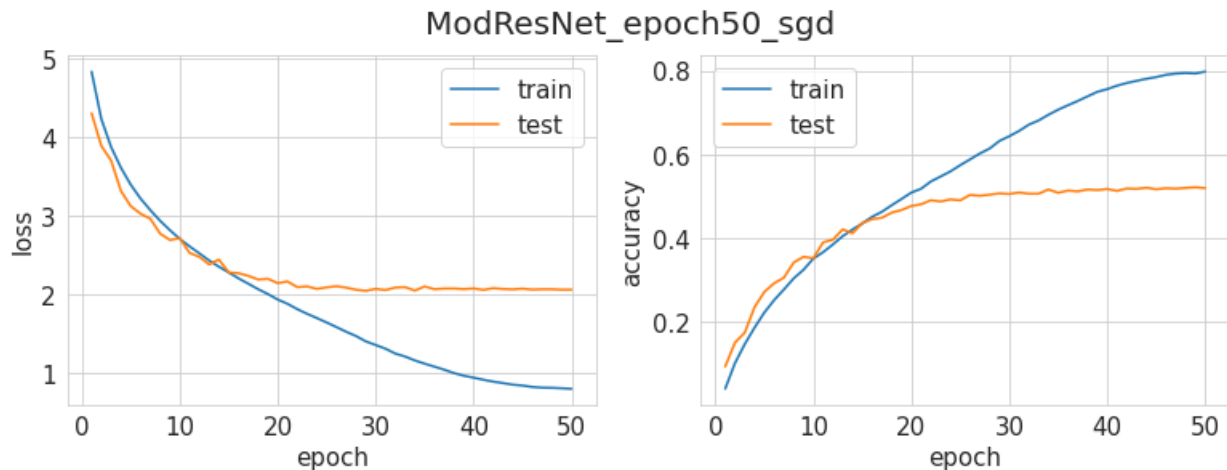
А **Kaggle** выдал на это 0.441.

4 В четырёх стенах

Стал думать *A*, как же ещё можно увеличить скор. Решил он вернуться к началам, и обучать **ResNet18**, но изменив подход. Ещё сильнее уменьшить количество эпох, но при этом взять очень маленький размер батча. Не штрафовать за большие веса, но взять изначально маленький шаг, а также немного поменять модифицированные слои. Так родилась модель **ModResNet**.

```
1 model.conv1 = nn.Conv2d(in_channels = 3, out_channels = 64,  
2   kernel_size = 3, padding = 1, stride = 1)  
3 model.maxpool = nn.Sequential()  
4 model.fc = nn.Sequential(  
5   nn.Dropout(inplace = True, p = 0.2),  
6   nn.Linear(in_features = 512, out_features = 200, bias = True),  
7 )
```

В этот раз запуск был таким: num epochs = 50, batch size = 20, lr = 0.001, momentum = 0.9, weight decay = 0 и расписанием CosineAnnealingLR. Вот что вышло из этого:

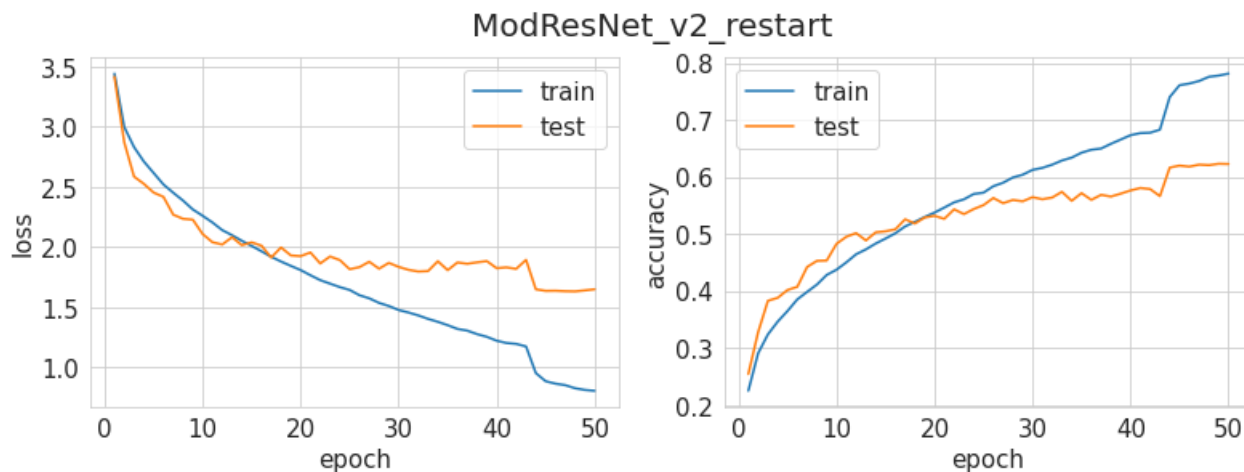


А **Kaggle** выдал на это 0.50339.

5 Пятый элемент

Но вот пришла *A* великая идея! Ведь самый лучший способ обучить модель – использовать предобученные веса. Но по велению магистра *L* такое было запрещено правилами соревнования. Но *A* был из хитрых и решил сам предобучить модель. Для этого он взял **ModResNet** и обучил его на 50 эпохах с теми же параметрами, но с batch size = 100 и вероятностью дропаута 0.5 (к сожалению **Датасфера** уничтожила график, но та модель ничем больше не отличалась, а называлась **ModResNet_v2**). После чего добавил в аугментации самую опасную, ту, что несёт в себе хаос – **RandAugment**. Но чтобы хаос подчинить и понять взял он новое расписание – **ReduceLROnPlateau**, что само понижало шаг, когда модель переставала учиться.

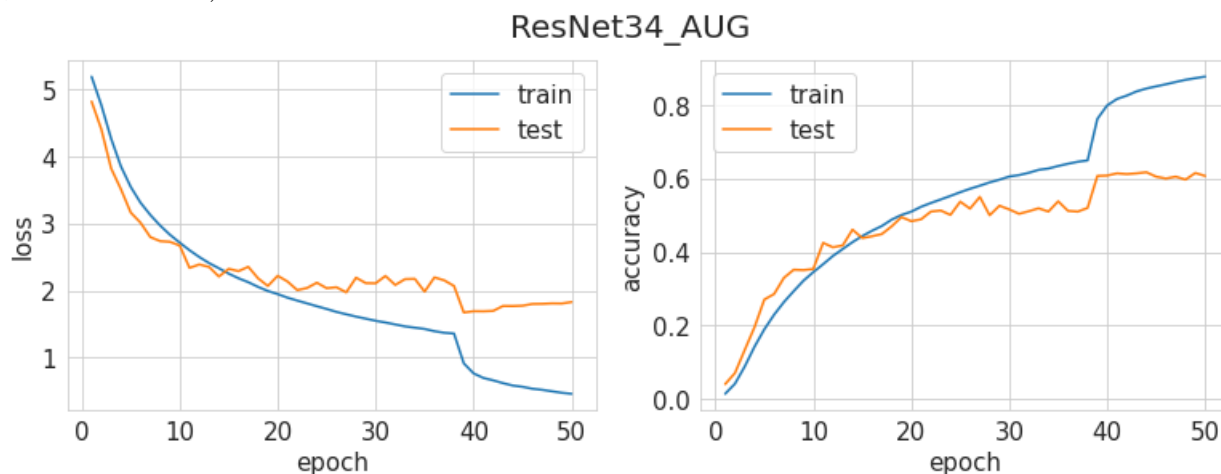
И вот так была запущена на весах предобученной модели новая: num epochs = 50, batch size = 100, lr = 0.01, momentum = 0.9, weight decay = 0 и расписанием **ReduceLROnPlateau**. Вот что вышло из этого:



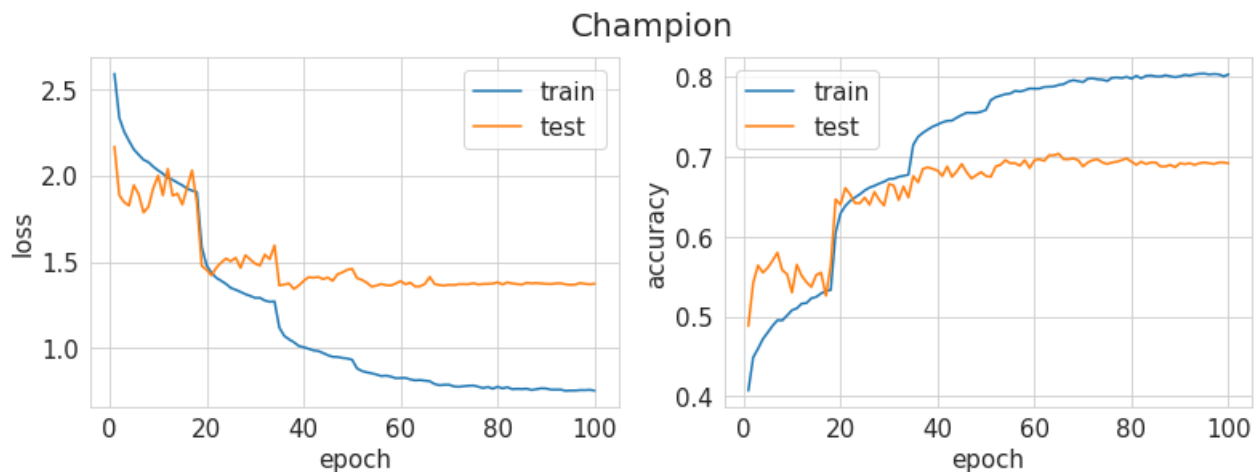
А **Kaggle** выдал на это 0.604.

6 Шестое чувство пробудилось

В этот момент нашей истории пришла великая и ужасная беда – по велению пресветлейшего *Е* отключили студентам **Датасферу**, так как превысили они бюджет. Но незадолго до этого *А* начал работать в корпорации *У*, а потому имел доступ к собственной **Датасфере** бесплатно. Так он продолжил улучшать свою модель. Для начала *А* разбил выборку на сбалансированные **train** и **val** в отношении 99 : 1, с помощью **pandas** (дабы увеличить количество данных для обучения). *А* также в этот раз он аккуратнее выбрал параметры и вспомнил уроки мудрости с **НИСа**, где рассказывали про аугментацию **CutOut**, которая в торче носит имя **RandomErasing**. Так была обучена модель **ResNet34_AUG** с параметрами **num epochs** = 50, **batch size** = 200, **lr** = 0.1, **momentum** = 0.9, **weight decay** = 0.0001 и расписанием **ReduceLROnPlateau**(**eps** = 0, **cooldown** = 1):



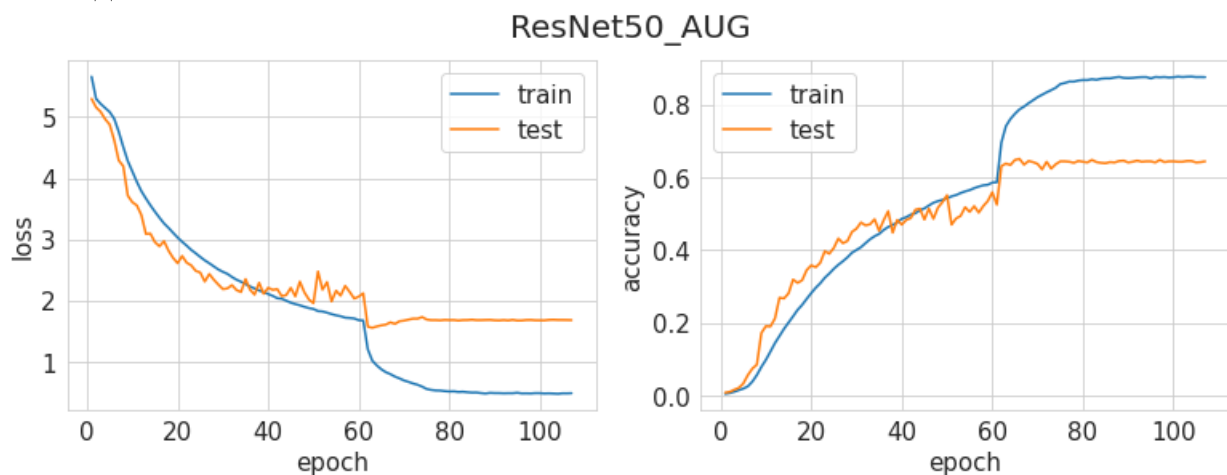
После чего *А* запустил с её весами обучение модели **Champion** с добавленным **RandAugment** и параметрами **num epochs** = 100, **batch size** = 300, **lr** = 0.1, **momentum** = 0.9, **weight decay** = 0.0001 и расписанием **ReduceLROnPlateau**(**eps** = 0, **cooldown** = 5, **factor** = 0.3):



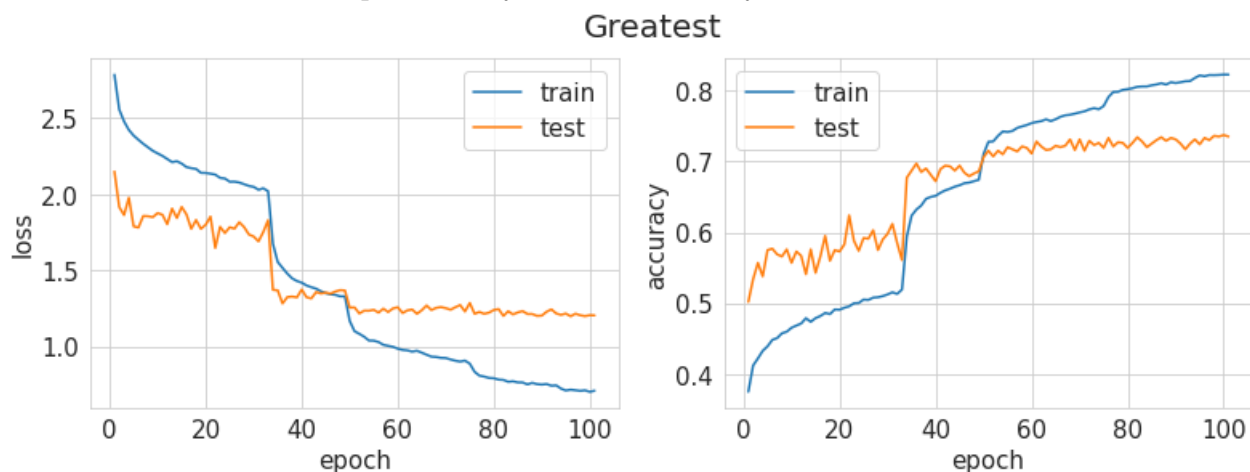
А **Kaggle** выдал на это 0.6768.

7 Семеро одного подождали

И вот настал последний день и стал думать *A*, как бы стать лучшим, среди всех студентов, дабы посвятить это достижение прекрасной *P*. И понял он – можно заменить **ResNet34** на **ResNet50**, только так, чтобы в дропауте оставалась та же доля ненулевых коэффициентов (стало быть $p = 0.875$). В остальном ничем от прошлых запусков не отличаясь обучилась первая модель так:



А с её весами на более серьёзных аугментациях получилось:



А **Kaggle** выдал на это 0.7064.

8 Восьмое чудо света

И так понял А, что стал он чуть-чуть умнее, так как понял, что нейронки надо обучать с умом, а не просто увеличивать количество эпох (хотя в ходе экспериментов он так делать пытался). Многие его провальные попытки не вошли в сказ (в них он пытался обучать более умные нейросети или просто перебирал гиперпараметры). Все совпадения в сказе **НЕ** случайны. Надеюсь, что второе БДЗ будет таким же интересным.

