



# CV Research: CIFAR100 и параметры

Самородова Екатерина

МГУ, мехмат, CV SHARE  
4 апреля 2022

# Задача исследования

- Рассмотреть различные архитектуры нейросетевых моделей классификаторов с ограничением числа параметров
- Провести сравнение качества моделей на примере датасета CIFAR100 для [1K, 5K, 10K, 50K, 100K, 500K, 1M] параметров
- Представить результаты исследования



# Использованные методы

## Conv2D

Добавление/удаление сверточных слоев, изменение количества карт и размеров фильтров, использование padding="same"

## FullyConnected

Добавление/удаление полносвязных слоев, изменение количества нейронов в слоях

## BatchNormalization

Использование BatchNorm2d после блока со сверткой (Conv2D → ReLU → MaxPooling)

# Неиспользованные методы

## Dropout

Не влияет непосредственно на количество обучаемых параметров

## GlobalAveragePooling

Замена полносвязного слоя  
Не получилось обучить модель

## Transfer Learning

Большинство предобученных и SoTA-моделей слишком большие



# Модель 2К параметров



## Архитектура

MaxPool(3) →

Conv(3, 5, 3) → ReLU → MaxPool(3) → BatchNorm(5) →

Flatten → Linear(20, 100)

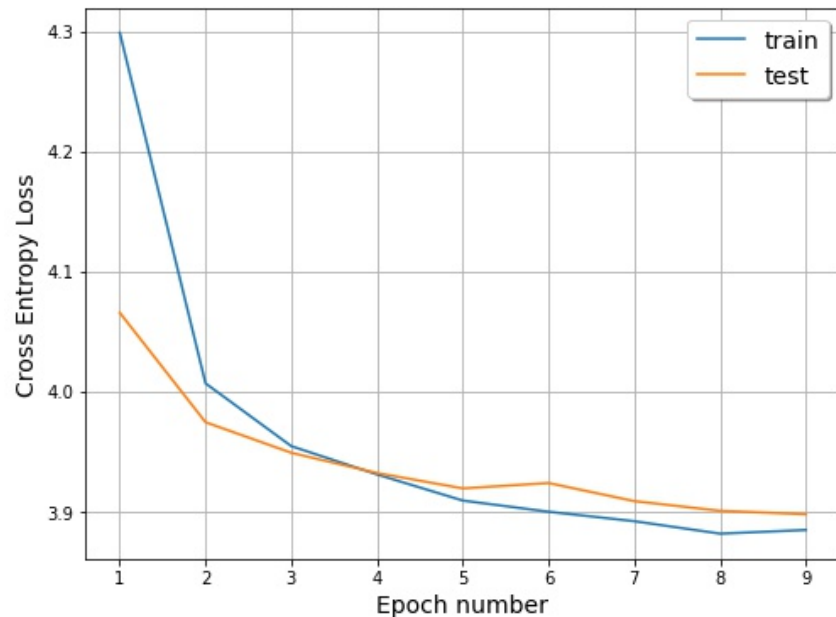
Количество обучаемых параметров: **2250**

## Результаты обучения

TOP-1 Accuracy: **11.67%**

Cross Entropy Loss: 3.89

Количество эпох: 9



# Модель 5K параметров



## Архитектура

Conv(3, 6, 3) → ReLU → MaxPool(3) → BatchNorm(6) →  
Conv(6, 10, 3) → ReLU → MaxPool(3) → BatchNorm(10) →  
Flatten → Linear(40, 100)

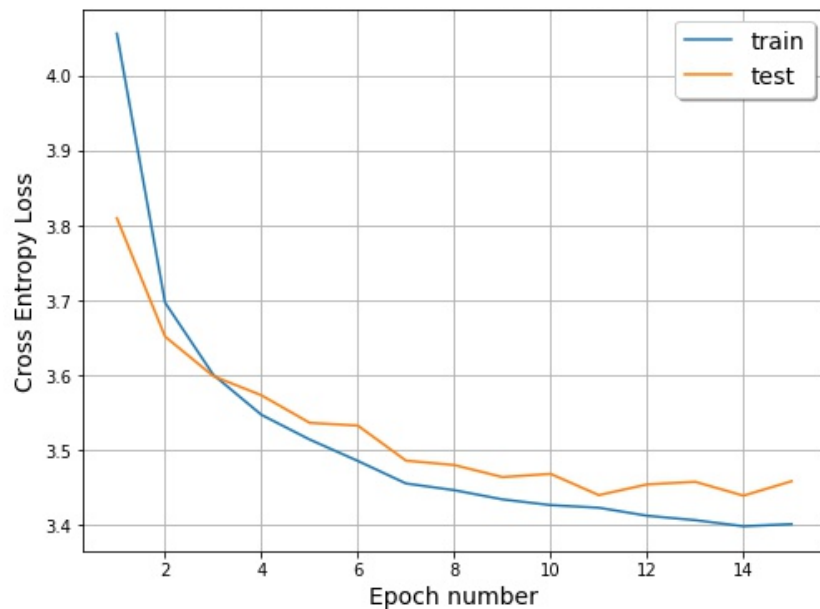
Количество обучаемых параметров: **4850**

## Результаты обучения

TOP-1 Accuracy: **19.14%**

Cross Entropy Loss: 3.45

Количество эпох: 15



# Модель 10К параметров

## Архитектура

Conv(3, 6, 3) → ReLU → MaxPool(2) → BatchNorm(6) →  
Conv(6, 10, 3) → ReLU → MaxPool(2) → BatchNorm(10) →  
Conv(10, 12, 3) → ReLU → MaxPool(2) → BatchNorm(12) →  
Flatten → Linear(48, 55) → ReLU → Linear(55, 100)

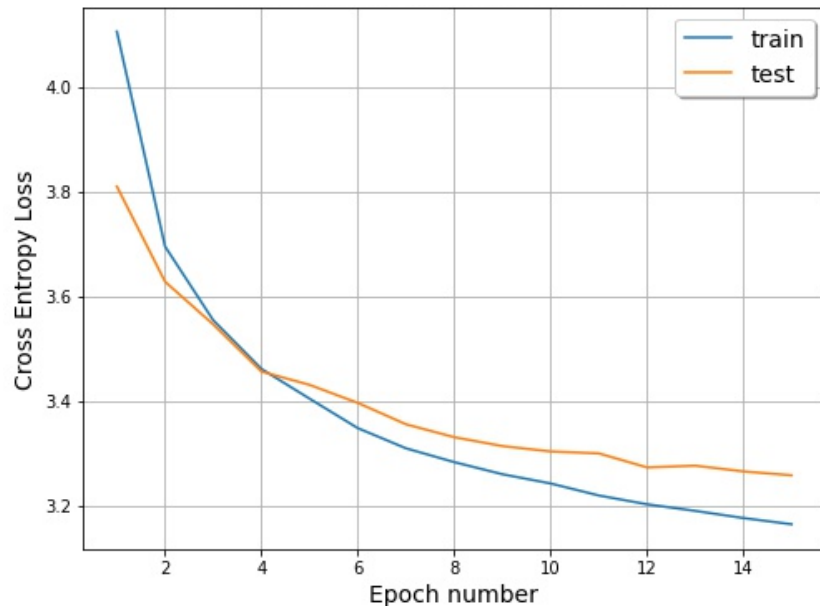
Количество обучаемых параметров: **10161**

## Результаты обучения

TOP-1 Accuracy: **21.97%**

Cross Entropy Loss: 3.25

Количество эпох: 15



# Модель 50К параметров

## Архитектура

Conv(3, 8, 3) → ReLU → MaxPool(2) → BatchNorm(8) →  
Conv(8, 16, 3) → ReLU → MaxPool(2) → BatchNorm(16) →  
Conv(16, 32, 3) → ReLU → MaxPool(2) → BatchNorm(32) →  
Flatten → Linear(128, 189) → ReLU → Linear(189, 100)

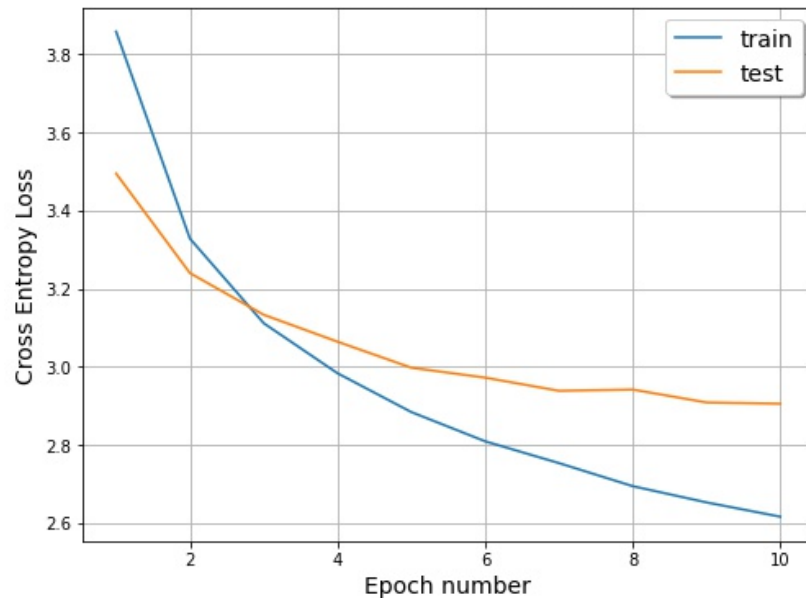
Количество обучаемых параметров: **49525**

## Результаты обучения

TOP-1 Accuracy: **29.39%**

Cross Entropy Loss: 2.90

Количество эпох: 10



# Модель 100K параметров

## Архитектура

Conv(3, 16, 3) → ReLU → MaxPool(2) → BatchNorm(16) →  
Conv(16, 32, 3) → ReLU → MaxPool(2) → BatchNorm(32) →  
Conv(32, 64, 3) → ReLU → MaxPool(2) → BatchNorm(64) →  
Flatten → Linear(256, 213) → ReLU → Linear(213, 100)

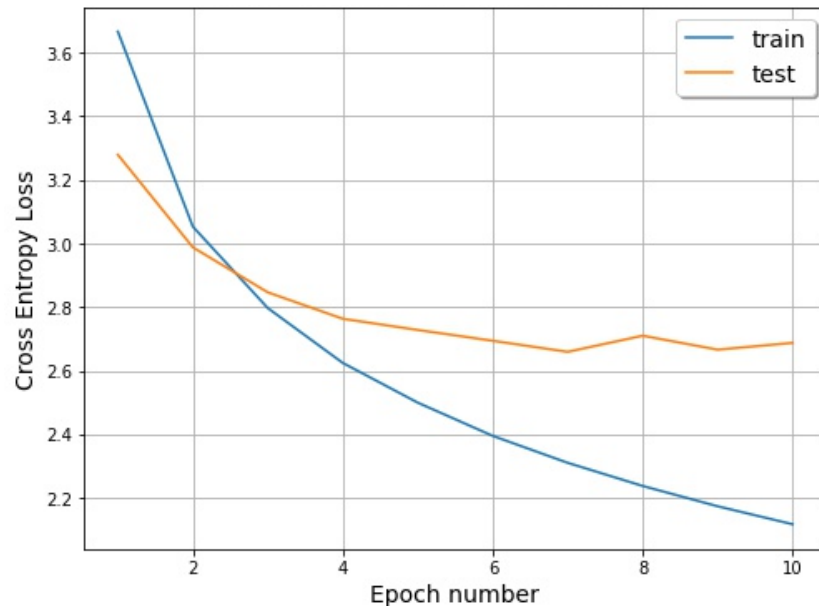
Количество обучаемых параметров: **99949**

## Результаты обучения

TOP-1 Accuracy: **34.97%**

Cross Entropy Loss: 2.68

Количество эпох: 10





# Модель 500K параметров

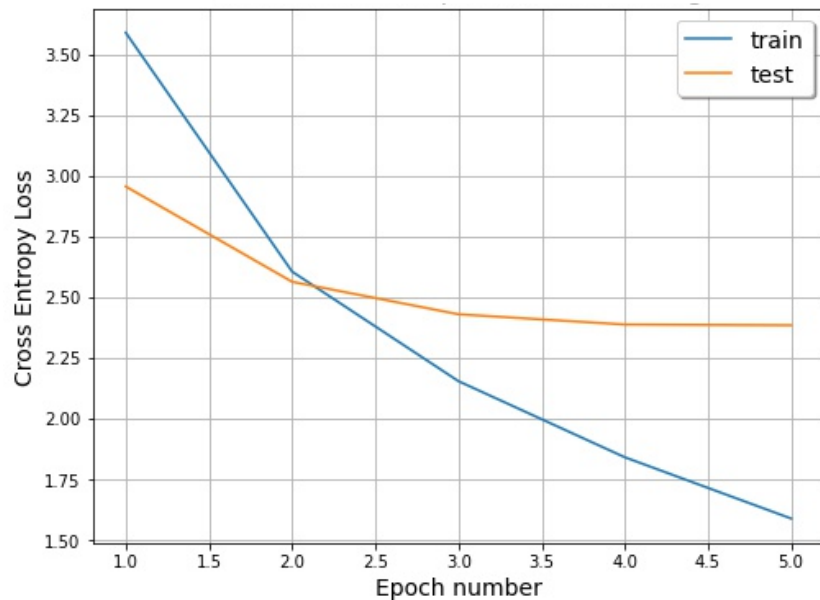


## Архитектура

Conv(3, 32, 3) → BatchNorm(32) → ReLU →  
Conv(32, 64, 3) → BatchNorm(64) → ReLU → MaxPool(2) →  
Res(64, 64) → MaxPool(2) → Flatten → Linear(4096, 194)  
Количество обучаемых параметров: **503396**

## Результаты обучения

TOP-1 Accuracy: **41.34%**  
Cross Entropy Loss: 2.38  
Количество эпох: 5



# Модель 1М параметров



## Архитектура

Conv(3, 64, 3) → ReLU → MaxPool(2) → BatchNorm(64) →  
Conv(64, 128, 3) → ReLU → MaxPool(2) → BatchNorm(128) →  
Conv(128, 256, 3) → ReLU → MaxPool(2) → BatchNorm(256) →  
Flatten → Linear(4096, 150) → ReLU → Linear(150, 100)

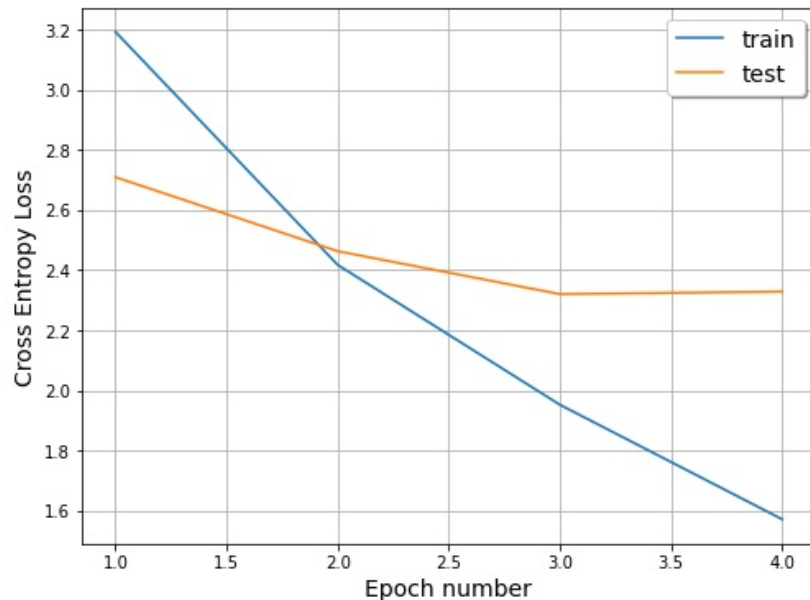
Количество обучаемых параметров: **1001362**

## Результаты обучения

TOP-1 Accuracy: **43.13%**

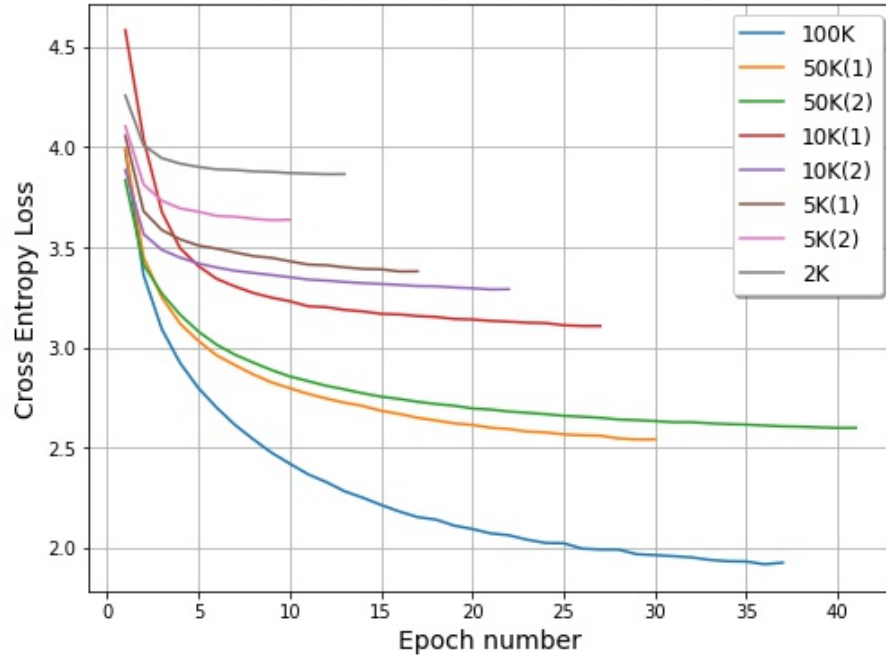
Cross Entropy Loss: 2.32

Количество эпох: 4

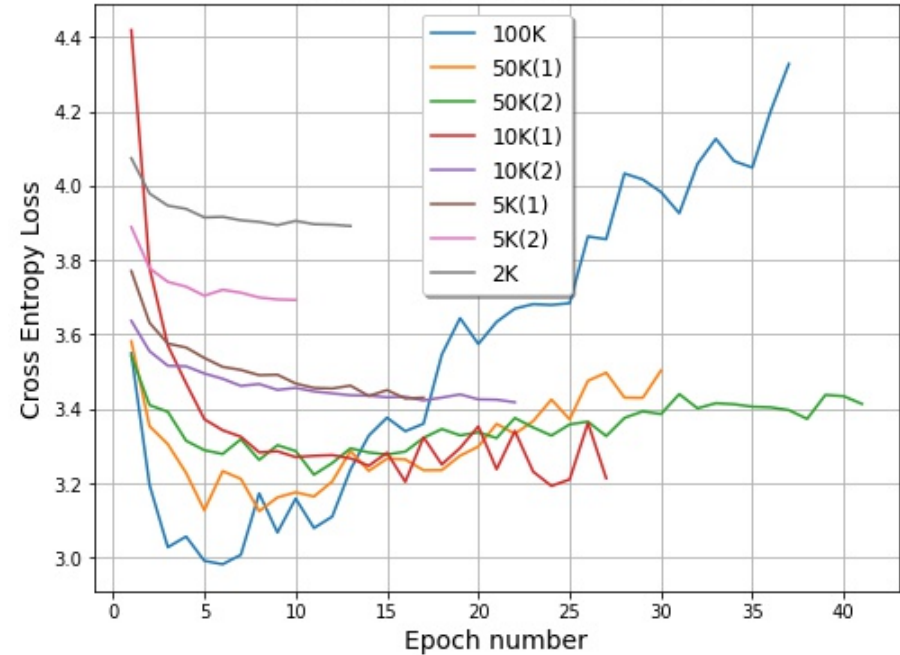


# Проблема: переобучение

Training loss for different models

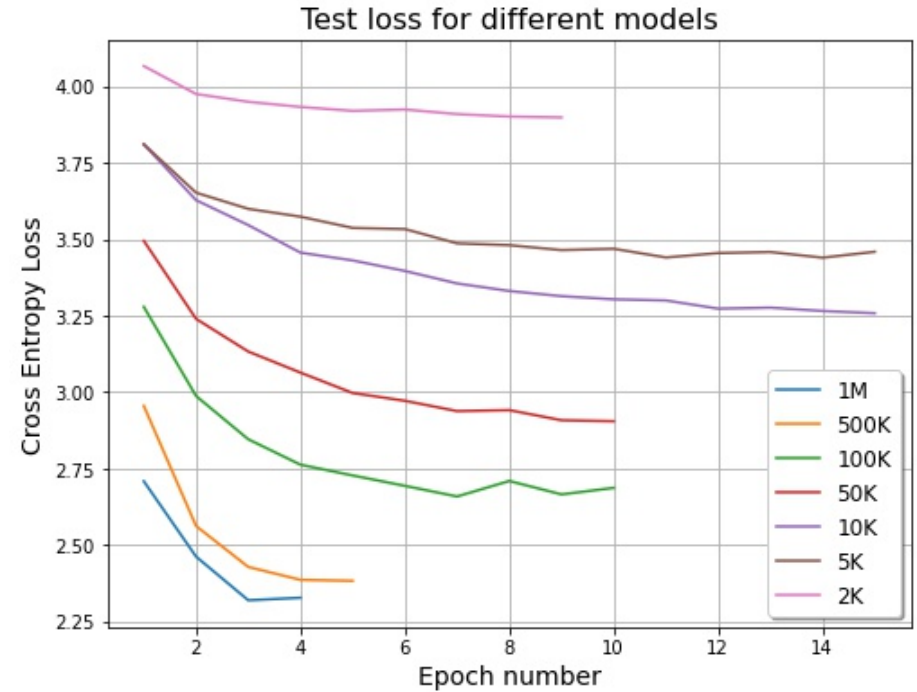
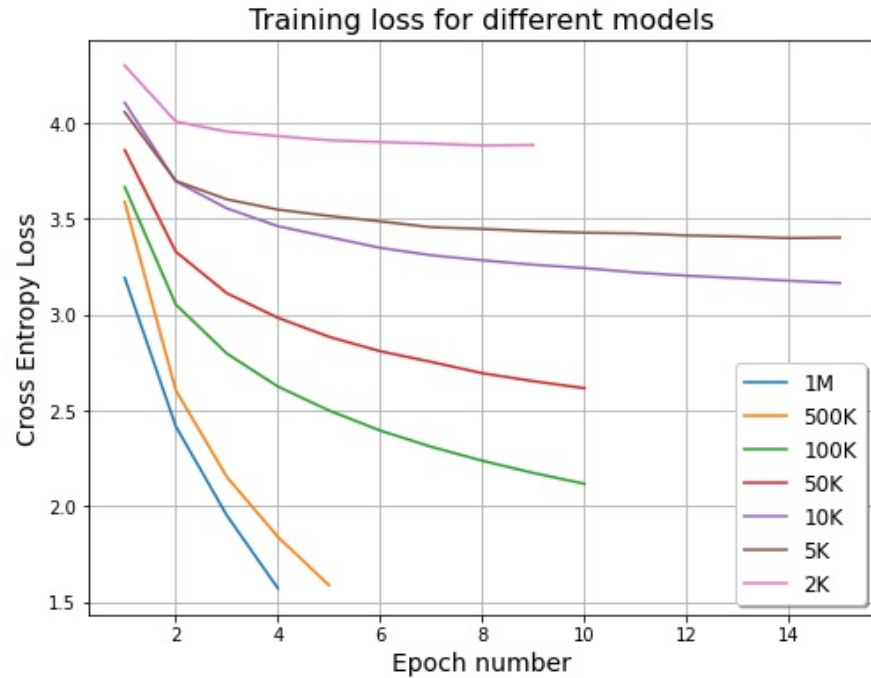


Test loss for different models



Получается, что нужно уменьшать количество эпох обучения модели при увеличении числа обучаемых параметров модели

# Решение: сокращение обучения



Большие модели в данном случае достигают своего максимума примерно за 5 эпох, более маленькие – за 10-20 эпох



## Итоговая таблица

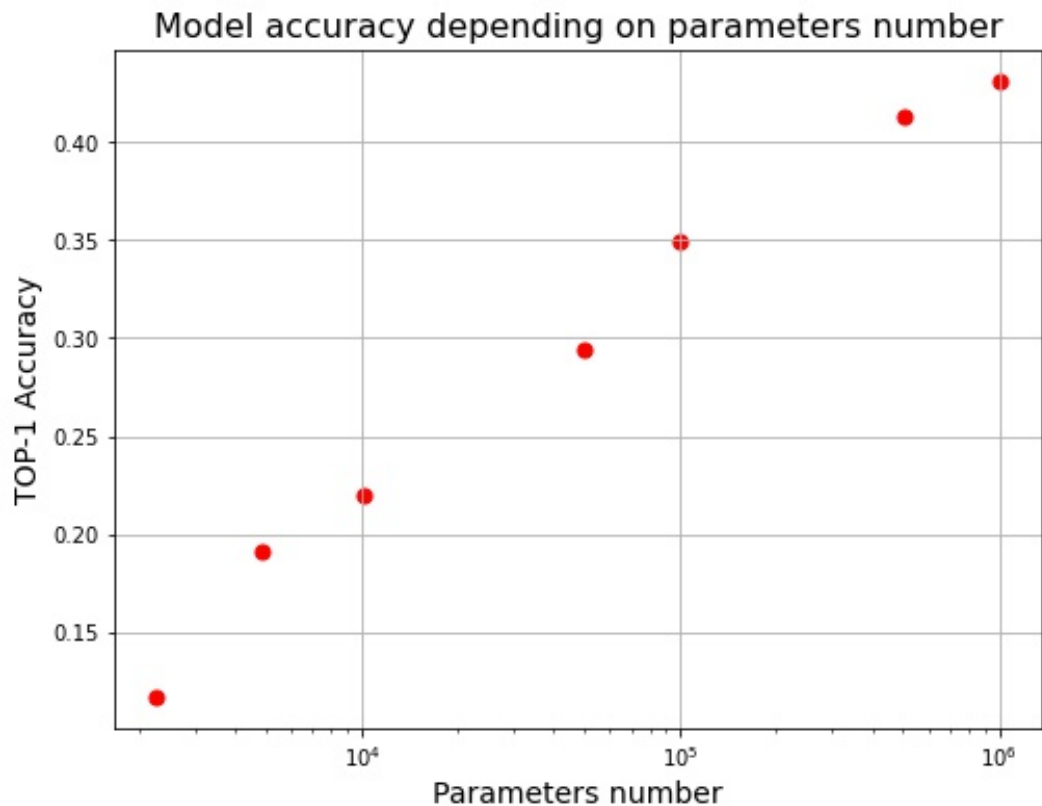
Количество параметров	TOP-1 Accuracy	CELoss (train)	CELoss (test)	Количество эпох
<b>2250</b>	11.67%	3.88	3.89	9
<b>4850</b>	19.14%	3.40	3.45	15
<b>10161</b>	21.97%	3.16	3.25	10
<b>49525</b>	29.39%	2.58	2.87	10
<b>99949</b>	34.97%	2.11	2.68	10
<b>503396</b>	41.34%	1.58	2.38	5
<b>1001362</b>	43.13%	1.57	2.32	4



## Результаты и выводы

В среднем, при увеличении числа параметров модели качество возрастает. При этом также влияют другие факторы, в первую очередь – архитектура модели (количество сверточных слоев), количество эпох обучения сети и т. д.

При помощи нескольких сверточных слоев и батч-нормализации можно построить достаточно эффективную модель для решения задачи многоклассовой классификации.





# Источники

- 1) <https://towardsdatascience.com/how-to-reduce-training-parameters-in-cnns-while-keeping-accuracy-99-a213034a9777>
  - 2) <https://medium.com/@alitbk/image-classification-in-a-nutshell-5-different-modelling-approaches-in-pytorch-with-cifar100-8f690866b373>
  - 3) <https://www.kaggle.com/code/yiweiwangau/cifar-100-resnet-pytorch-75-17-accuracy>
- 