

Introductory CNNs research

Author:
Nikita Kakurnikov

Содержание

1. Data	2
2. Configuration Clapeyron_CNN_v0.X	2
3. Training	3
4. Results	3
5. Augmentation	3
5.1. No augmentation (standart augmentation)	3
5.2. augm_v0.1	3
6. Clapeyron_CNN_v0.2	3
6.1. Configuration and training	3
6.2. Results	4
7. Clapeyron_CNN_v0.2_augm_v0.1	4
7.1. Configuration and training	4
7.2. Results	4
8. Clapeyron_CNN_v0.3	5
8.1. Configuration and training	5
8.2. Results	5
9. Clapeyron_CNN_v0.4	6
9.1. Configuration and training	6
9.2. Results	6
10. Clapeyron_CNN_v0.5	6
10.1. Configuration and training	6
10.2. Results	6
11. Clapeyron_CNN_v0.6	7
11.1. Configuration and training	7
11.2. Results	7

1. Data

Данные скачаны с ImageNet в количестве 10 классов. Количество фото на класс в табл. 1. Минимальное количество фотографий на класс - 749, поэтому обучение - 649, а тест - 100 фотографий.

И обучение и тест для каждого класса выбраны случайно из всех имеющихся фотографий класса, индексы этих изображений сохранены один раз и использованы для всех сетей, описанных в этой бумаге.

index	name	images
9	wallet	749
1	headphones	865
3	key	936
8	sodacan	998
6	phone	1473
0	bottle	1731
4	laptop	1771
5	pen	2360
2	human	2526
7	shoes	9337

Таблица 1: Индекс класса, имя класса, общее количество изображений для класса

2. Configuration Clapeyron_CNN_v0.X

```
Input [224x224x3]
Conv3: [224x224xD1] weights: (3x3x3)xD1
    relu
Conv3: [224x224xD1] weights: (3x3x16)xD1
    relu
    MaxPool2: [112x112xD1]
Conv3: [112x112xD2] weights: (3x3x16)xD2
    relu
Conv3: [112x112xD2] weights: (3x3x32)xD2
    relu
    MaxPool2: [56x56xD2]
Conv3: [56x56xD3] weights: (3x3x32)xD3
    relu
Conv3: [56x56xD3] weights: (3x3x64)xD3
    relu
    MaxPool2: [28x28xD3]
Conv3: [28x28xD4] weights: (3x3x64)xD4
    relu
Conv3: [28x28xD4] weights: (3x3x128)xD4
    relu
    MaxPool2: [14x14xD4]
Conv3: [14x14xD4] weights: (3x3x128)xD4
    relu
Conv3: [14x14xD4] weights: (3x3x128)xD4
    relu
    MaxPool2: [7x7xD4]
    FC: [1x1xFC1]
    relu
    dropout(FC1p)
    FC: [1x1xFC2]
    relu
    dropout(FC2p)
    FC: [1x1x10]
    softmax
```

Формат записи конфигурации сетей Clapeyron_CNN_v0.X: conf_D1_D2_D3_D4_FC1_FC2

3. Training

Формат записи обучения через обычный градиентный спуск (gradient descent) с ручным decay: `opt_GD_lrXiterY`, где `opt` - optimizer, `lr` - learning rate, `X` - значение `lr`, `iter` - iterations (количество использования оптимизатора до которого использовалось текущее `lr`, либо начиная с `iter0`, либо с предыдущего `iter`), `Y` - значение iterations.

Функция потерь на батче: $\text{loss_crossentropy} = \frac{-1}{\text{batch_size}} \sum_{k=0}^{\text{batch_size}-1} \ln p_{i_k k}$, где $p_{i_k k}$ - предсказанная вероятность верного класса на k -ом объекта батча.

Формат записи обучения с использованием dropout-регуляризации: `reg_dropout_FC1pXFC2pXiterY`, где `FC1p` - вероятность учета нейрона FC1 слоя.

При обучении мог использоваться dropout после FC-слоев. Все данные о точности на тесте получены без dropout.

Для всех сетей, исследованных в этой бумаге, `batch_size` = 100.

В батче всегда находились по 10 изображений из каждого класса, выбранных случайно из train выборки.

Начальная инициализация весов: усеченное нормальное распределение для параметров сверток, в том числе веса FC-слоев (truncated normal, `stddev` = 0.05). Все смещения (по смещению на каждый слой) задавались нулями.

4. Results

В качестве результатов обучения сети предоставлена следующая информация:

- 1) Продолжительность обучения: итерация оптимизации, на которой самая высокая точность предсказаний тестовой выборки. Эта точность снималась каждую 1000 итераций.
- 2) Точность на тесте: значение этой максимальной точности на тестовой выборке.
- 3) Время до переобучения: первая итерация, на которой точность предсказаний батча равна 100%. Вывод предсказаний на батче был каждые 100 итераций.
- 4) Максимальная точность на батче.
- 5) Время максимальной точности на батче: время = итерация.

Дальнейшие данные - результаты работы сети с самой точной конфигурацией (на тесте) из всех итераций обучения: 6) Таблица с точностью предсказания каждого класса на трайне, тесте и на всех доступных изображениях без трайна.

5. Augmentation

5.1. No augmentation (standart augmentation)

Если не указано, что при обучении была использована аугментация, значит, исходная картинка преобразовывалась к размеру 256×256 , затем центральное окно (вырез) размера 224×224 подавался на вход нейросети.

5.2. augm_v0.1

Для каждого класса в батче любая картинка из десяти имела вероятность 0.3 остаться "стандартной". С вероятностью 0.7 к ней могла быть применена аугментация со следующими последовательными действиями:

- 1) изменение размеров картинки (длины и ширины) на новые из множества $[224, 226, 228, \dots, 370]$, каждое значение равновероятно.
- 2) с вероятностью 0.5 изображение могло быть отражено по горизонтали.
- 3) изображение могло быть повернуто на 0, 90, 180, 270 градусов равновероятно.
- 4) сдвиг окна 224×224 от центра получившегося изображения. Генерировались равновероятные значения (сдвиги) для горизонтали и вертикали из множества $[0.1, 0.2, 0.3, \dots, 0.9]$. Сдвиг окна по горизонтали от $x = 0$ равнялся $(\text{image_width} - 224) \cdot \text{shiftx}$, т.е. в случае положения окна по центру на горизонтали $\text{shiftx} = 0.5$ (аналогично для вертикали).

Формат записи аугментации `augm_v0.1`:

`a1_step2_w×h[224,370]×[224,370]_a2_0.5_a3_a4_step0.1_w×h[0.1,0.9]×[0.1,0.9]`

6. Clapeyron_CNN_v0.2

6.1. Configuration and training

```
conf_16_32_64_128_128_512_256
opt_GD_lr0.5iter8000_lr0.001iter15000
reg_dropout_FC1p0.8FC2p0.8iter15000
loss_crossentropy, batch_size= 100
augm_No
```

6.2. Results

Test: 34.5%, *iter* = 14000, *loss* = 5.7; 34.2%, *iter* = 7000

Batch 100%: *iter* = 3400

index	images	train	test	all	all-test
9	749	99.8	42.0	42.0	0.0
1	865	100.0	37.0	42.1	5.1
3	936	99.7	28.0	26.1	1.9
8	998	99.8	33.0	35.8	2.8
6	1473	100.0	19.0	23.1	4.1
0	1731	99.2	34.0	21.4	12.6
4	1771	100.0	40.0	31.0	9.0
5	2360	100.0	46.0	24.4	21.6
2	2526	100.0	19.0	22.3	3.3
7	9337	100.0	47.0	25.2	21.8

Таблица 2: Результаты (ассурагу) обучения сети Clapeyron_CNN_v0.2

index	images	train	test	all	all-test
9	749	0.0	5.4	5.4	0.0
1	865	0.0	5.2	5.1	0.1
3	936	0.0	6.5	6.8	0.3
8	998	0.0	5.4	5.2	0.2
6	1473	0.0	7.9	7.5	0.4
0	1731	0.0	5.4	6.7	1.2
4	1771	0.0	5.2	6.7	1.4
5	2360	0.0	4.9	7.6	2.6
2	2526	0.0	6.4	6.6	0.2
7	9337	0.0	4.1	7.1	3.0

Таблица 3: Результаты (loss) обучения сети Clapeyron_CNN_v0.2

7. Clapeyron_CNN_v0.2_augm_v0.1

7.1. Configuration and training

conf_16_32_64_128_128_512_256
opt_GD_lr0.1iter6000_lr0.07iter20000
reg_dropout_FC1p0.8FC2p0.8iter20000
loss_crossentropy, batch_size= 100
augm_v0.1

7.2. Results

Test: 56.6%, *iter* = 17000, *loss* = 1.38; 52.8%, *iter* = 8000

Batch 100%: No

Batch max: 82.0%, *iter* = 17900

index	images	train	test	all	all-test
9	749	99.8	72.0	72.0	0.0
1	865	99.8	62.0	70.8	8.8
3	936	99.5	53.0	53.3	0.3
8	998	99.4	56.0	61.3	5.3
6	1473	99.8	46.0	48.8	2.8
0	1731	98.9	50.0	36.0	14.0
4	1771	99.8	49.0	44.7	4.3
5	2360	99.5	64.0	34.5	29.5
2	2526	99.7	58.0	62.0	4.0
7	9337	99.4	56.0	26.1	29.9

Таблица 4: Результаты (ассурагу) обучения сети Clapeyron_CNN_v0.2

index	images	train	test	all	all-test
9	749	0.1	0.9	0.9	0.0
1	865	0.1	1.1	0.8	0.2
3	936	0.1	1.6	1.5	0.1
8	998	0.1	1.2	1.2	0.1
6	1473	0.1	1.6	1.6	0.0
0	1731	0.1	1.3	2.1	0.8
4	1771	0.1	1.6	1.7	0.1
5	2360	0.1	1.5	2.7	1.2
2	2526	0.1	1.4	1.1	0.3
7	9337	0.1	1.6	2.8	1.2

Таблица 5: Результаты (loss) обучения сети Clapeyron_CNN_v0.2

8. Clapeyron_CNN_v0.3

8.1. Configuration and training

conf_32_64_128_224_224_1024_1024
 opt_GD_lr0.001iter10000_lr0.0001iter24000
 reg_dropout_FC1p0.8FC2p0.8iter24000
 loss_crossentropy, batch_size= 100
 augm_v0.1

8.2. Results

Test: 38.6%, *iter* = 23000, *loss* = 1.78; 38.0%, *iter* = 18000

Batch 100%: No

Batch max: 45.0%, *iter* = 19900

index	images	train	test	all	all-test
9	749	58.4	56.0	56.0	0.0
1	865	48.8	50.0	50.9	0.9
3	936	21.9	18.0	14.6	3.4
8	998	46.1	42.0	47.9	5.9
6	1473	36.7	34.0	31.1	2.9
0	1731	37.3	36.0	21.3	14.7
4	1771	26.8	27.0	20.3	6.7
5	2360	55.8	55.0	33.3	21.7
2	2526	31.1	24.0	29.5	5.5
7	9337	57.0	44.0	21.9	22.1

Таблица 6: Результаты (ассигуру) обучения сети Clapeyron_CNN_v0.2

index	images	train	test	all	all-test
9	749	1.3	1.5	1.5	0.0
1	865	1.6	1.7	1.6	0.1
3	936	1.9	1.9	2.0	0.1
8	998	1.5	1.6	1.5	0.1
6	1473	1.8	2.0	2.0	0.0
0	1731	1.7	1.8	2.2	0.4
4	1771	2.0	2.0	2.1	0.1
5	2360	1.6	1.6	2.2	0.6
2	2526	1.8	2.0	1.9	0.2
7	9337	1.6	1.8	2.4	0.6

Таблица 7: Результаты (loss) обучения сети Clapeyron_CNN_v0.2

9. Clapeyron_CNN_v0.4

9.1. Configuration and training

```
conf_16_32_64_128_128_512_512
opt_GD_lr0.1iter10000_lr0.05iter15000_lr0.001iter20000_lr0.0001iter25000
_lr0.00002iter30000_lr0.000001iter35000_lr0.0000001iter40000
reg_dropout_FC1p0.8FC2p0.8iter25000_FC1p1.0FC2p1.0iter40000
loss_crossentropy, batch_size= 100
augm_v0.1
```

9.2. Results

Test: 62.8%, *iter* = 27000, *loss* = 1.33;

Batch 100%: No

Batch max: 90.0%, *iter* = 37500; 89.0%, *iter* = 40000

index	images	train	test	all	all-test
9	749	100.0	73.0	73.0	0.0
1	865	99.2	70.0	72.2	2.2
3	936	100.0	66.0	60.6	5.4
8	998	99.4	68.0	64.8	3.2
6	1473	100.0	52.0	54.4	2.4
0	1731	99.4	68.0	46.1	21.9
4	1771	100.0	59.0	52.4	6.6
5	2360	100.0	71.0	44.4	26.6
2	2526	99.8	41.0	48.4	7.4
7	9337	99.8	60.0	30.7	29.3

Таблица 8: Результаты (ассурасу) обучения сети Clapeyron_CNN_v0.2

index	images	train	test	all	all-test
9	749	0.0	0.9	0.9	0.0
1	865	0.1	1.1	1.0	0.1
3	936	0.0	1.3	1.3	0.0
8	998	0.1	1.0	1.2	0.2
6	1473	0.0	1.5	1.6	0.0
0	1731	0.0	1.2	1.9	0.7
4	1771	0.0	1.6	1.6	0.0
5	2360	0.0	1.0	2.2	1.2
2	2526	0.1	1.9	1.6	0.4
7	9337	0.0	1.7	3.0	1.3

Таблица 9: Результаты (loss) обучения сети Clapeyron_CNN_v0.2

10. Clapeyron_CNN_v0.5

10.1. Configuration and training

```
conf_8_16_32_64_64_256_256
opt_GD_lr0.1iter10000_lr0.01iter20000_lr0.001iter25000_lr0.0001iter30000
_lr0.00001iter35000_lr0.000001iter40000
reg_dropout_FC1p0.8FC2p0.8iter30000_FC1p1.0FC2p1.0iter40000
loss_crossentropy, batch_size= 100
augm_v0.1
```

10.2. Results

Test: 57.1%, *iter* = 33000, *loss* = 1.36;

Batch 100%: No

Batch max: 80.0%, *iter* = 33200

index	images	train	test	all	all-test
9	749	99.8	71.0	71.0	0.0
1	865	99.7	64.0	74.1	10.1
3	936	99.2	54.0	51.2	2.8
8	998	99.4	55.0	61.9	6.9
6	1473	100.0	45.0	45.1	0.1
0	1731	99.2	50.0	36.0	14.0
4	1771	99.5	52.0	48.8	3.2
5	2360	100.0	66.0	40.0	26.0
2	2526	99.8	54.0	59.2	5.2
7	9337	100.0	60.0	30.5	29.5

Таблица 10: Результаты (ассигасу) обучения сети Clapeyron_CNN_v0.2

index	images	train	test	all	all-test
9	749	0.1	0.9	0.9	0.0
1	865	0.1	1.0	0.9	0.2
3	936	0.1	1.5	1.5	0.0
8	998	0.1	1.2	1.2	0.0
6	1473	0.1	1.8	1.7	0.0
0	1731	0.1	1.4	2.0	0.5
4	1771	0.1	1.6	1.6	0.0
5	2360	0.0	1.2	2.4	1.2
2	2526	0.1	1.4	1.3	0.1
7	9337	0.1	1.5	2.4	0.9

Таблица 11: Результаты (loss) обучения сети Clapeyron_CNN_v0.2

11. Clapeyron_CNN_v0.6

11.1. Configuration and training

```

conf_16_32_64_128_128_512_512
opt_GD_lr0.1iter10000_lr0.01iter20000_lr0.001iter25000_lr0.0001iter30000
_lr0.00001iter35000_lr0.000001iter40000
reg_dropout_FC1p0.8FC2p0.8iter30000_FC1p1.0FC2p1.0iter40000
loss_crossentropy, batch_size= 100
augm_v0.1

```

11.2. Results

Test: 60.9%, *iter* = 33000, *loss* = 1.38;
Batch 100%: No
Batch max: 80.0%, *iter* = 33200

index	images	train	test	all	all-test
9	749	100.0	72.0	72.0	0.0
1	865	100.0	69.0	77.8	8.8
3	936	99.5	55.0	54.0	1.0
8	998	99.1	62.0	62.2	0.2
6	1473	100.0	44.0	45.0	1.0
0	1731	99.8	58.0	39.6	18.4
4	1771	100.0	61.0	53.5	7.5
5	2360	99.8	67.0	35.1	31.9
2	2526	100.0	55.0	61.9	6.9
7	9337	100.0	66.0	32.1	33.9

Таблица 12: Результаты (ассигасу) обучения сети Clapeyron_CNN_v0.2

index	images	train	test	all	all-test
9	749	0.1	0.9	0.9	0.0
1	865	0.1	1.0	0.9	0.2
3	936	0.1	1.5	1.5	0.0
8	998	0.1	1.2	1.2	0.0
6	1473	0.1	1.8	1.7	0.0
0	1731	0.1	1.4	2.0	0.5
4	1771	0.1	1.6	1.6	0.0
5	2360	0.0	1.2	2.4	1.2
2	2526	0.1	1.4	1.3	0.1
7	9337	0.1	1.5	2.4	0.9

Таблица 13: Результаты (loss) обучения сети Clapeyron_CNN_v0.2