

Свёрточные автокодировщики для улучшения качества классификации изображений

1 Введение

В данной работе был проведен анализ свёрточной нейронной сети для задач классификации изображений. Для реализации классов свёрточной нейронной сети и свёрточного автокодировщика использовалась библиотека `pytorch`. Эксперименты были проведены на датасете `stl-10`. В качестве мер качества использовались `accuracy` и `log loss`.

2 Эксперименты

2.1 Бейзлайны

В первом эксперименте сравнивается работа алгоритмов мультиномиальной регрессии и случайного леса. Для мультиномиальной регрессии были получены на тестовой выборке $accuracy = 36\%$ и $log_loss = 0.45$. Для случайного леса полученное значение `accuracy` равно 34% .

2.2 Подбор параметров для свёрточной сети

У рассмотренной реализации свёрточной нейросети есть параметры: размер ядра свертки и количество фильтров на свёрточном слое. Рассмотрим, как зависит качество и скорость обучения сети от значений этих параметров. В таблицах 1 и 2 приведены результаты экспериментов. Лучшее качество ($accuracy = 55\%$) было получено при размере ядра свертки 4 и числе каналов 30.

2.3 Подбор параметров автоэнкодера

Подберем значения числа фильтров и размера `padding` для автокодировщика, при которых на отложенной выборке достигается наименьшее значение `MSE` для исходного и декодированного изображений. В таблицах 3 и 4 приведены результаты экспериментов. Видно, что оптимальное число каналов равно 40, а лучшее значение $padding = 3$.

На рисунках 2, 4, 6 показаны исходные изображения и их образы, восстановленные автокодировщиком. Видно, что основная информация ясно считывается.

3 Использование признаков из автоэнкодера

Попробуем улучшить результаты моделей из первого эксперимента, используя вместо изображений их промежуточные представления, полученные автокодировщиком. У мультиномиальной

Таблица 1: kernel size

kernel size	time (s)	accuracy (%)
2	189	53
4	175	53
6	159	52
8	152	49

Таблица 2: number of channels

number of channels	time (s)	accuracy (%)
5	93	46
10	124	49
15	163	51
20	187	54
25	218	52
30	260	55
35	312	54
40	329	55

Таблица 3: number of channels

number of channels	loss (MSE)
5	0.0122
10	0.0071
15	0.0060
20	0.0053
25	0.0049
30	0.0047
35	0.0046
40	0.0043

Таблица 4: padding

padding	loss (MSE)
0	0.0074
1	0.0025
2	0.0022
3	0.0021
4	0.0023

регрессии, обученной таким образом, на тестовой выборке $accuracy = 51\%$ и $log_loss = 0.37$. Видно, что качество заметно улучшилось. Это связано с тем, что автокодировщик при обучении отобрал шумовые признаки и выделил те, которые позволяют хорошо восстановить исходное изображение. По этой же причине качество, полученное для RandomForest (36%), не сильно

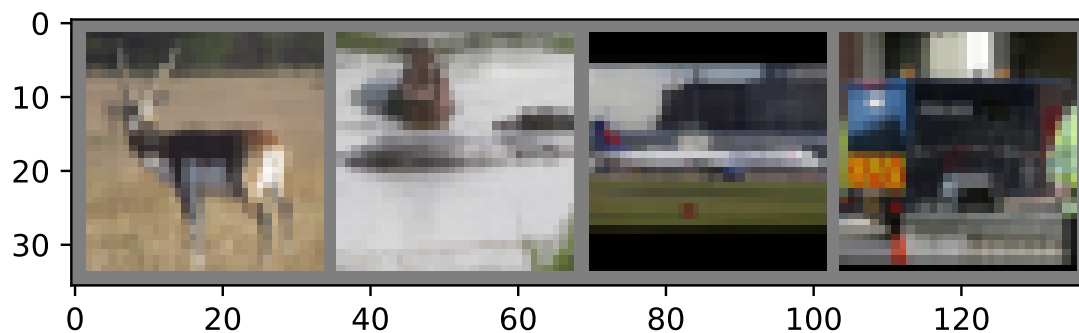


Рис. 1: original

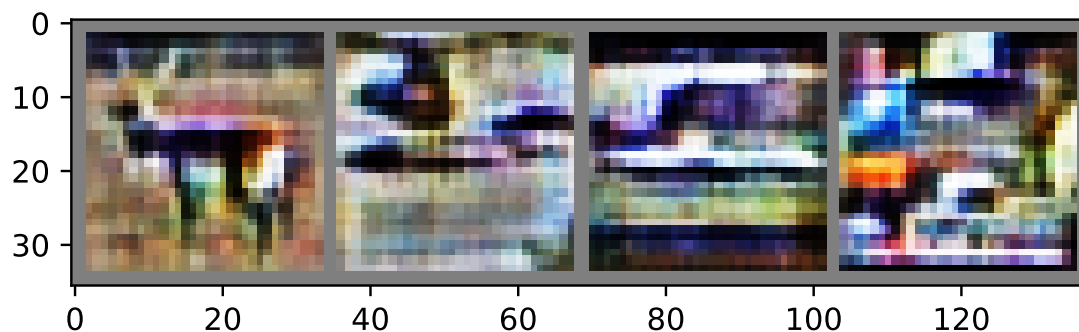


Рис. 2: result

отличается от исходного: RandomForest сам отбирает признаки.

3.1 Начальная инициализация весов нейросети

Попробуем улучшить качество работы нейросети, изначально проинициализировав ее весами, которые обучил автокодировщик на первом слое. Итоговое $accuracy = 55\%$, и видно, что оно не улучшилось по сравнению с исходной моделью.

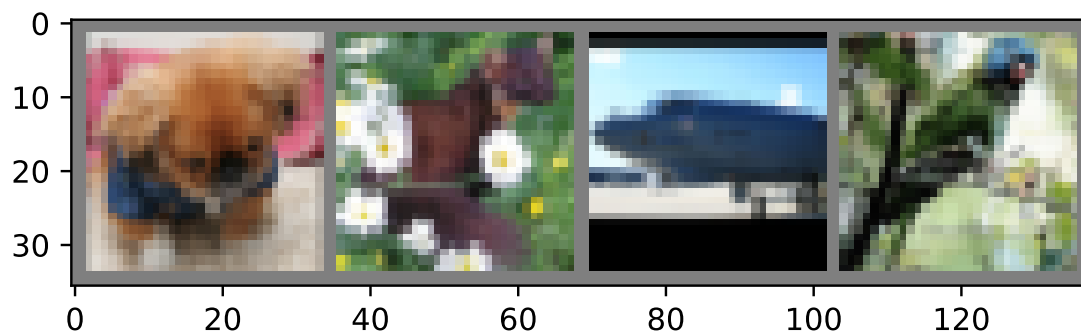


Рис. 3: original

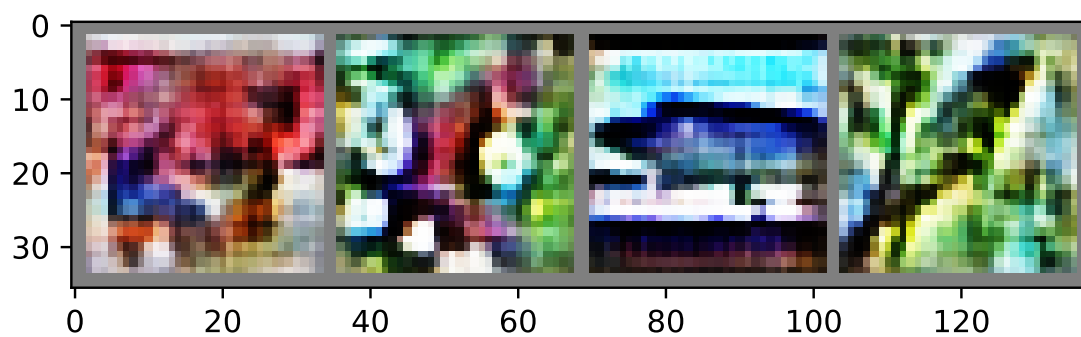


Рис. 4: result

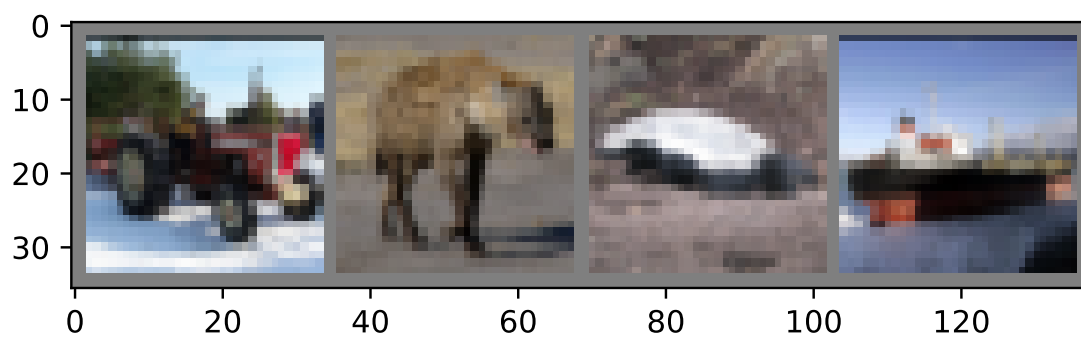


Рис. 5: original

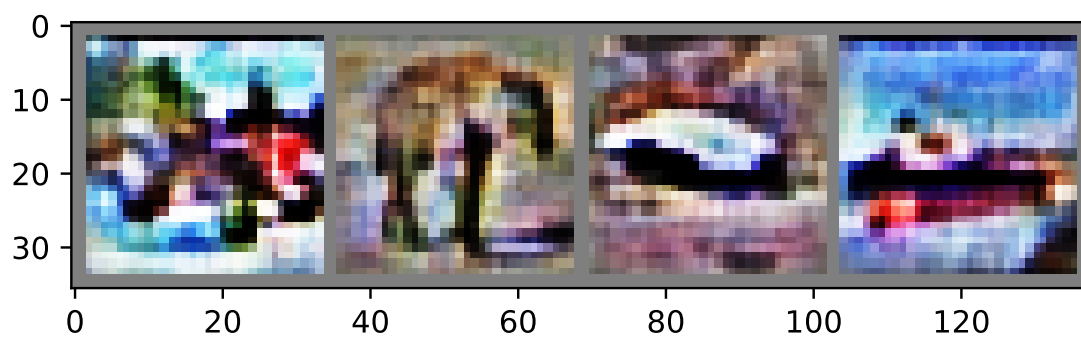


Рис. 6: result