

# Заочное задание для стажеров Samsung

Шаталов Николай

13 мая 2018 г.

Требуется построить классификатор, разделяющий изображения из приложенного датасета: крокодил или часы.

## 1 Данные

Датасет состоит из 1000 объектов: по 500 объектов на каждый класс.



Рис. 1: Пример объектов в датасете

Для оценивания алгоритма далее будем использовать кросс-валидацию по 10 фолдам.

Также, для улучшения качества работы алгоритма каждое изображение будем нормализовать, переводя его в отрезок  $[-1, 1]$ .

## 2 Классификатор

В качестве классификатора будем использовать сверточную нейронную сеть, обучаемую методом стохастического градиентного спуска. Архитектура сети изображена на рисунке 2.

Так как решается задача классификации, в качестве оптимизируемой функции потерь была выбрана кросс-энтропия:

$$\text{logloss} = \sum_i (-y_i \log a_i - (1 - y_i) \log (1 - a_i)) \rightarrow \min_w,$$

где  $y_i$  – правильная метка класса (0 или 1),  $a_i = a(x_i|w)$  – ответ алгоритма с текущими весами  $w$  на объекте  $x_i$ , вероятность принадлежности к классу 1.

Также будем считать точность модели:

$$\text{accuracy} = \frac{\sum_{i=1}^l [y_i = \hat{y}_i]}{s},$$

где  $\hat{y}_i$  – метка класса, которого предсказывает алгоритм,  $l$  – длина тестовой выборки.

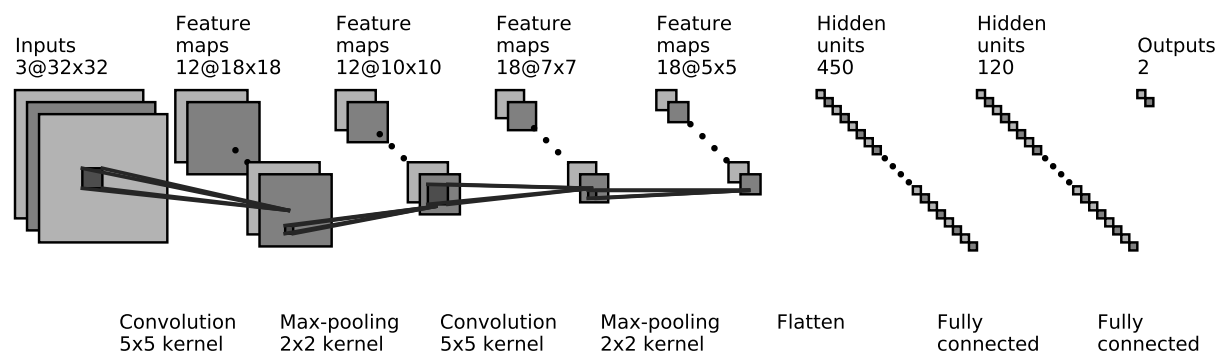


Рис. 2: Архитектура сверточной нейронной сети

На рисунках 3 и 4 можно увидеть качество работы модели. Заметную разницу в качестве работы алгоритма на валидационных выборках можно объяснить маленьким объемом исходной выборки.

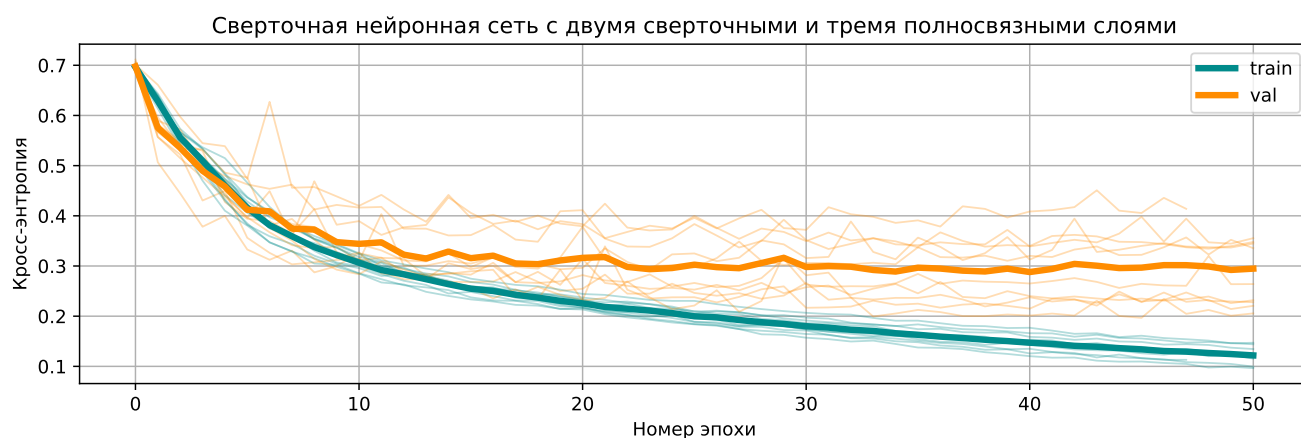


Рис. 3: Кросс-энтропия в зависимости от эпохи

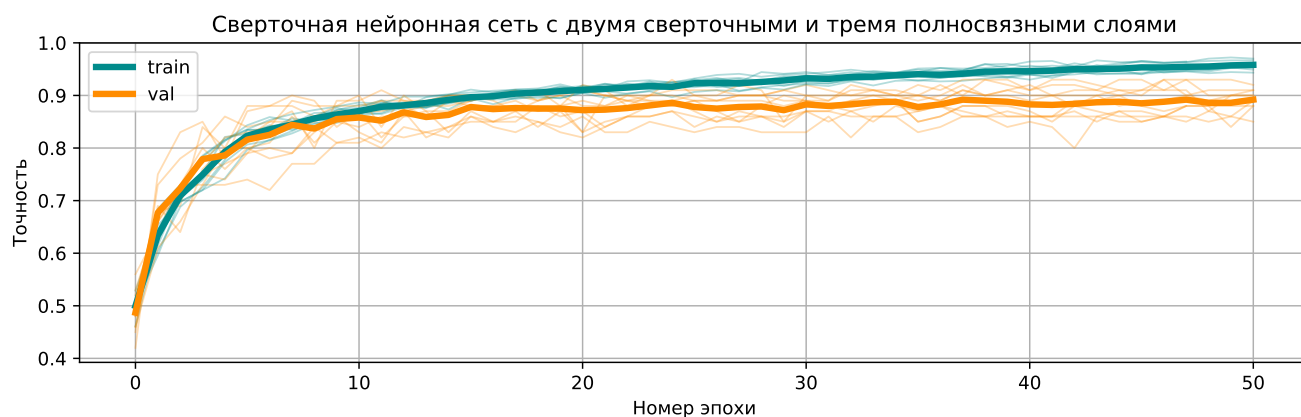


Рис. 4: Точность в зависимости от эпохи

	Обучающая выборка	Валидационная выборка
Кросс-энтропия	0.1219	0.2947
Точность	0.9582	0.8920

Таблица 1: Качество сверточной нейронной сети

### 3 «Неуверенные» изображения

Посмотрим, на каких объектах алгоритм неуверен в своем ответе (рис. 5).

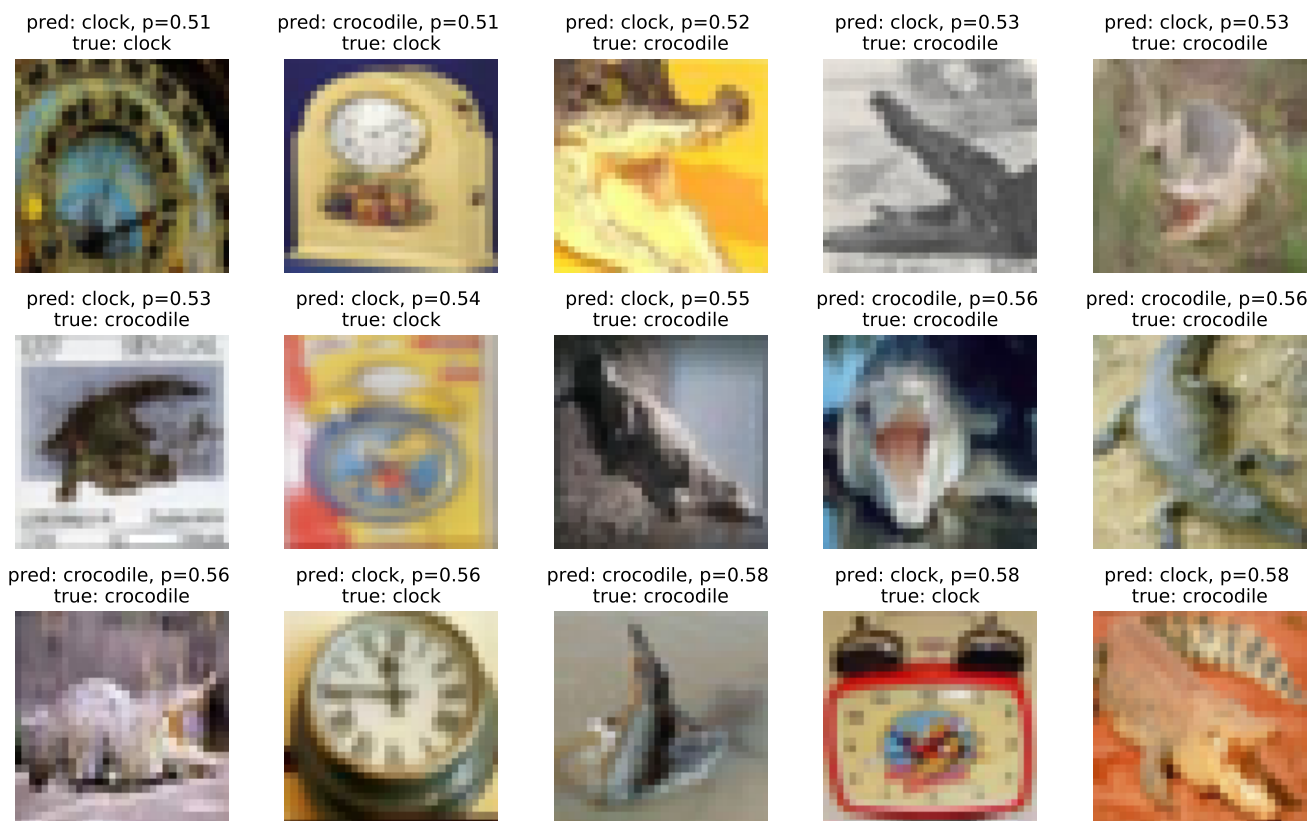


Рис. 5: Изображения, на которых алгоритм не уверен как отвечать

Как видно, алгоритм не может дать точного ответа, если у часов «пестристого» цвета, или если пасть крокодила открыта (можно спутать с круглым циферблатом, если фотография в профиль, или можно спутать с стрелками часов, если фотография в анфас).

Теперь посмотрим, на каких объектах алгоритм сильно ошибается (уверенно говорит за неправильный класс) на рисунке 6.

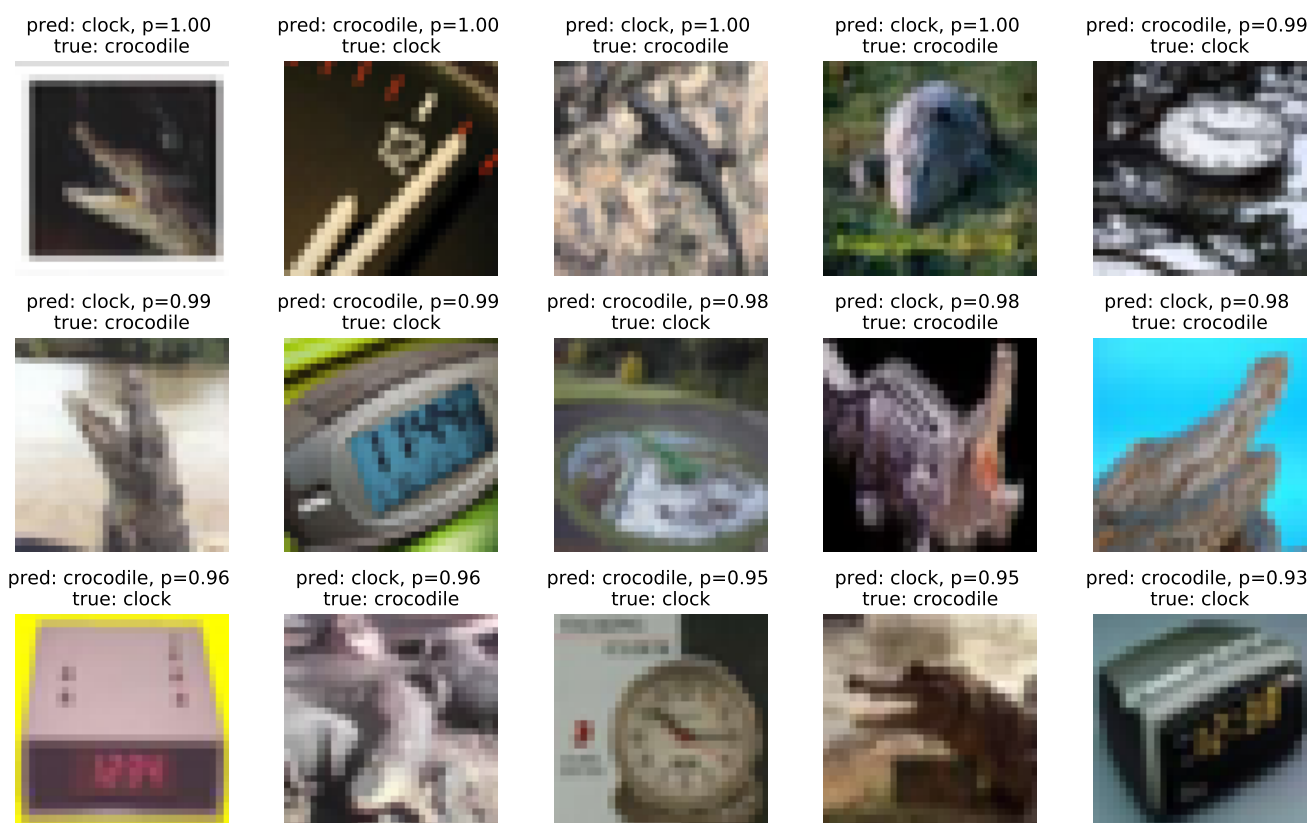


Рис. 6: Изображения, на которых алгоритм сильно ошибается

Алгоритм ошибается если крокодил сфотографирован сверху или с открытой пастью (и похож на стрелки часов), если у фотографии с крокодилом присутствуют рамки (путает с квадратными часами). В случае часов алгоритм ошибается, если они сфотографированы под острым углом или с другого странного ракурса, а также, если на фотографии цифровые часы.

## 4 Генерация новых изображений

Разобьем датасет на две подвыборки: выборка только с крокодилами и выборка только с часами. На них обучим два автоэнкодера. После, закодируем изображение крокодила и часов соответствующими автоэнкодерами, возьмем взвешенное среднее полученных векторов и декодируем одним из автоэнкодеров полученный вектор.

Ниже приведен пример «смешивания» часов и крокодила с весами 0.8 и 0.2 соответственно, а после полученный вектор был декодирован автоэнкодером, обученным на часах.

