

Сверточные нейронные сети

Семинар 18

Задача распознавания объектов



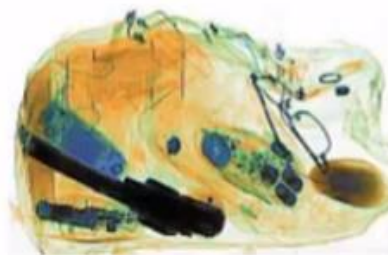
<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>



<http://news.mit.edu/2015/object-recognition-robots-0724>



<http://www.eurocarnews.com/0/0/2294/0/volvo-car-corporation-leads-the-way-in-car-safety.html>



Diana Turcsany, Andre Mouton, T. P. Breckon, Improving Feature-based Object Recognition for X-ray Baggage Security Screening using Primed Visual Words

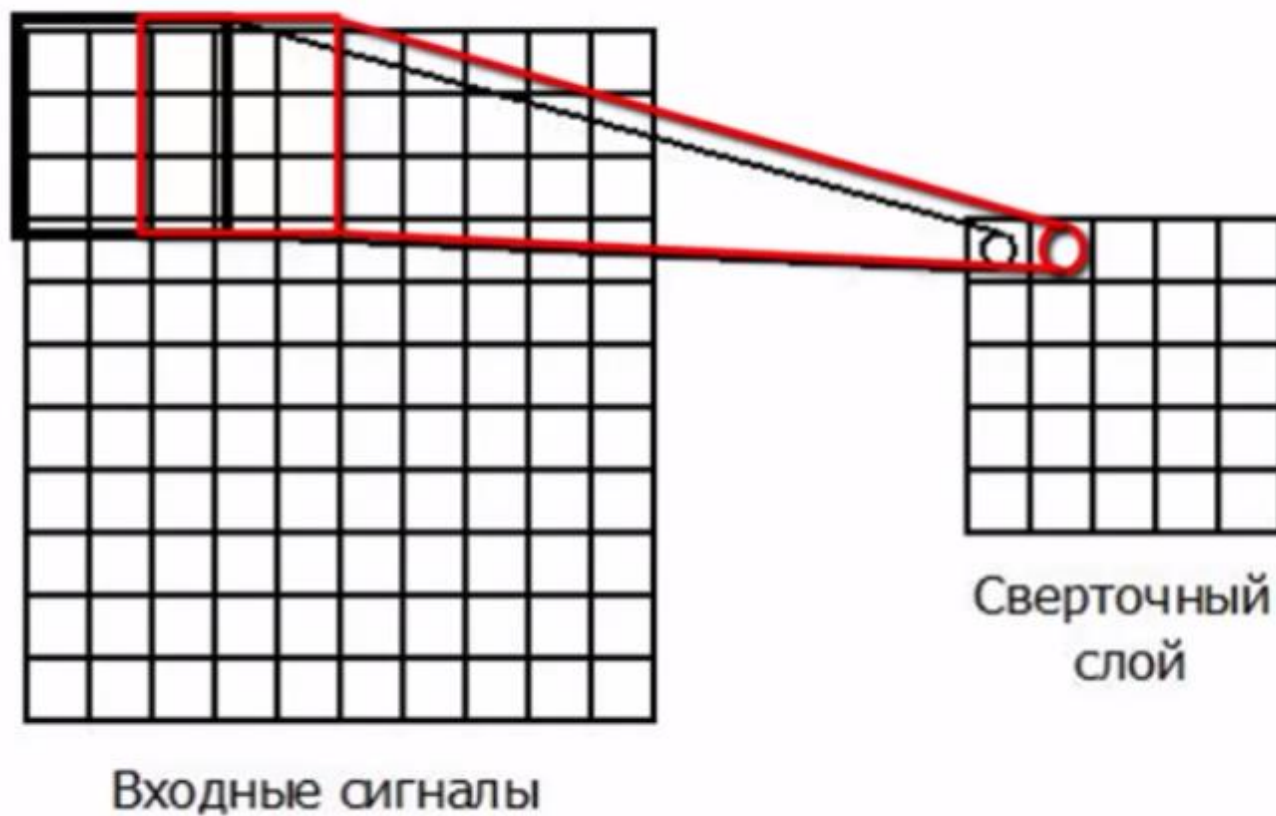
Нейронные сети для компьютерного зрения. Сверточные нейронные сети

- Недостатки полносвязной многослойной сети для анализа изображений:
 - Большое количество весов для обучения (изображения 28x28 пикселей – 784 входа, 635200 весов)
 - Изображения представляются в виде плоского массива – теряется информация о топологии

Принципы сверточных нейронных сетей

- Локальное восприятие
- Разделяемые веса для составления карт признаков
- Уменьшение размерности

Локальное восприятие



Свертка изображений

Ядро
свертки

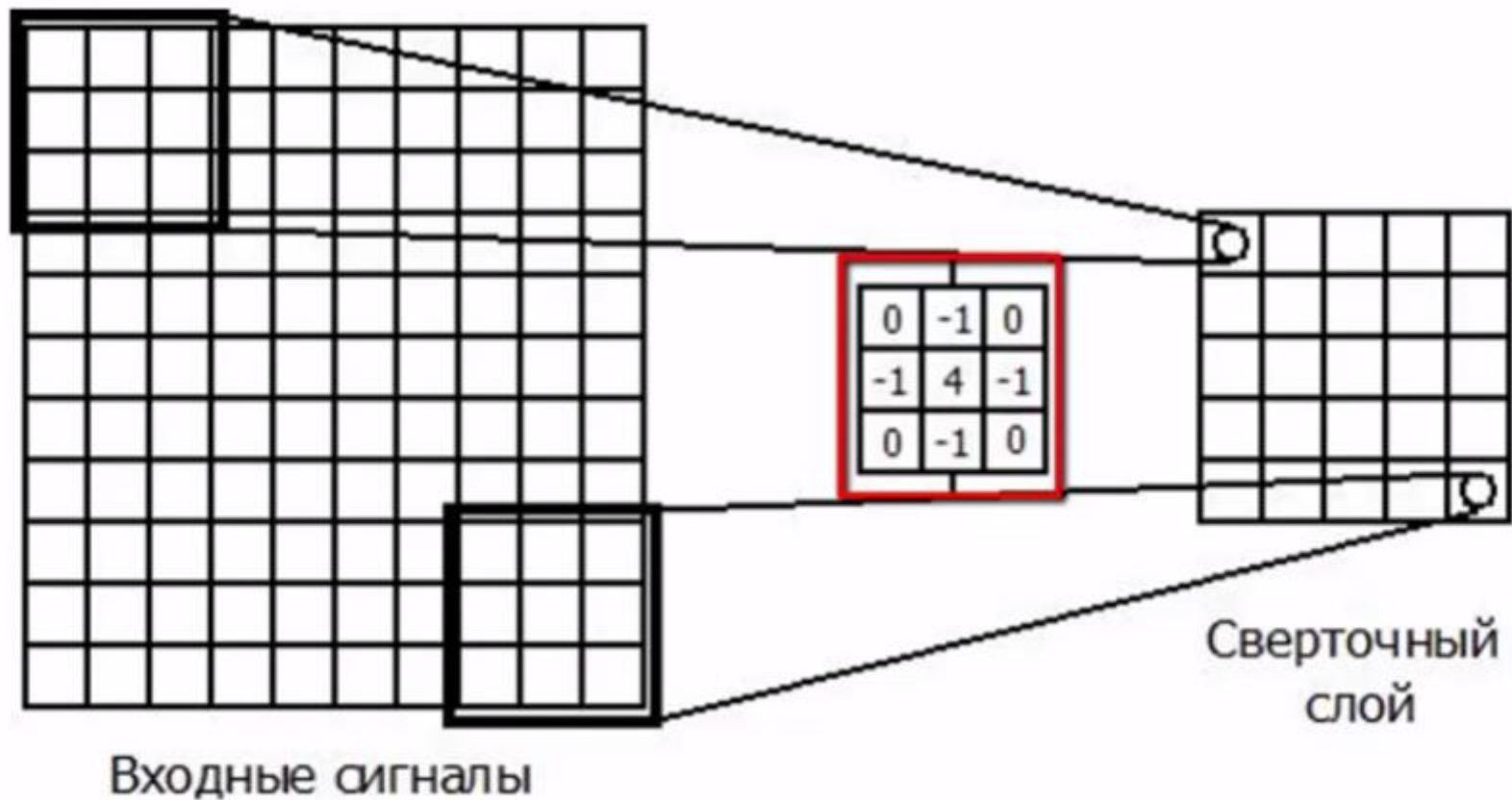
221	198	149
205	147	173
149	170	222

-1	0	1
-2	0	2
-1	0	1

$$N(x,y) = 221 * (-1) + 198 * 0 + 149 * 1 + 205 * (-2) + 147 * 0 + 173 * 2 + 149 * (-1) + 170 * 0 + 222 * 1 = -63$$

Та же самая операция свертки выполняет линейный нейрон, после к нейрону применяется нелинейная функция активации

Разделяемые веса

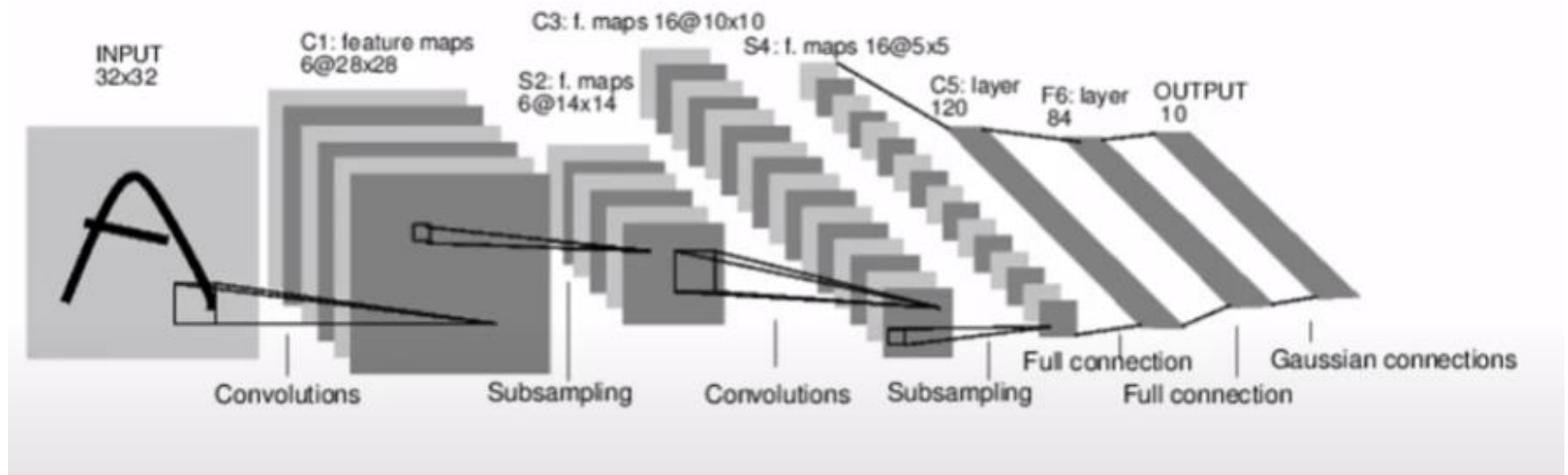


Уменьшение количества весов в процессе обучения за счет одинаковых весов ядра свертки

Уменьшение размерности

- Распознавание объектов вне зависимости от масштаба
- Факт наличия признака важнее его местоположения на картинке

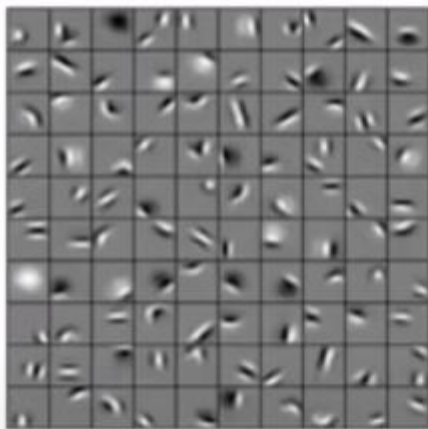
Сверточная нейросеть LeNet-5 (для распознавания цифр)



Слой характеризует конкретный признак

В конце решается задача классификации. 10 нейронов, каждый говорит о наличии одно

Распознавание лиц сверточными сетями



Выводы

- Архитектура сверточных нейронных сетей
 - Сверточные слои
- Преимущества
 - Небольшое количество параметров для обучения
- Обучение сверточных сетей
 - Алгоритм обратного распространения ошибки с ограничением на веса

Библиотеки для глубокого обучения

- Tensorflow – будет работать только на 64-разрядной машине
- Keras – позволяет на питоне описать нейронную сеть

Набор данных CIFAR10

Самолет



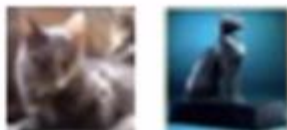
Автомобиль



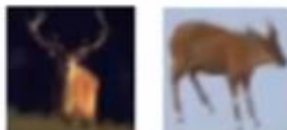
Птица



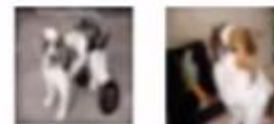
Кот



Олень



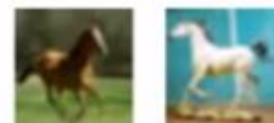
Собака



Лягушка



Лошадь



Корабль



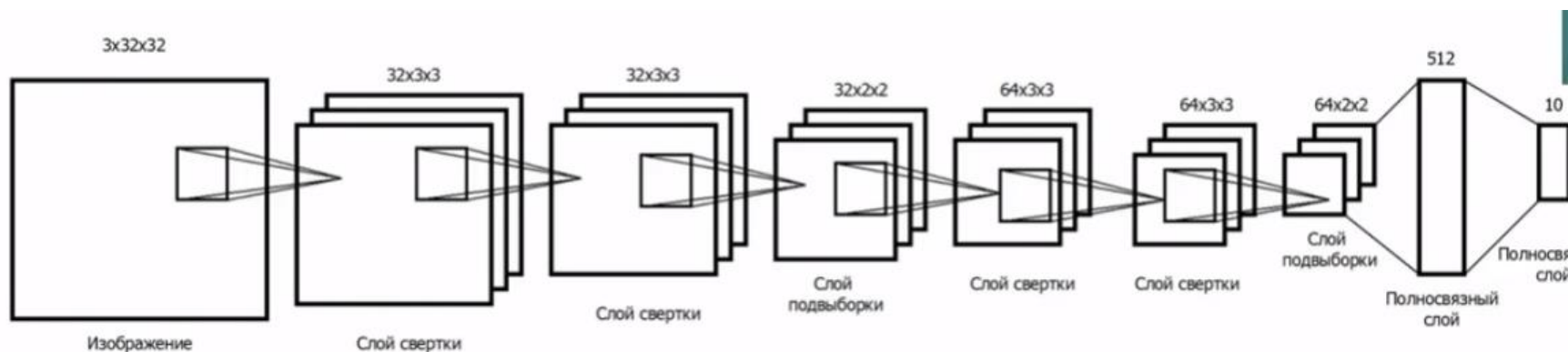
Грузовик



Набор данных CIFAR10

- Открытый набор данных
- Изображения в CIFAR10:
- Размер 32x32
- Цветные изображения
- Набор данных для обучения – 50000
- Набор данных для тестирования – 10000
- На каждом изображении только один объект
- Нет пересечений

Сверточная сеть для распознавания CIFAR10



32 и 64 карты (ядро свертки для поиска признаков) признаков

Подключение библиотек

```
import numpy
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Flatten, Activation
from keras.layers import Dropout
from keras.layers.convolutional import Convolution2D,
MaxPooling2D
from keras.utils import np_utils
```

```
# Задаем seed для повторяемости результатов
numpy.random.seed(42)
```

```
# Загружаем данные
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```


Предварительная обработка данных

```
# Нормализуем данные о интенсивности пикселей изображения
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255

# Преобразуем метки классов в категории
Y_train = np_utils.to_categorical(y_train, 10)
Y_test = np_utils.to_categorical(y_test, 10)
```

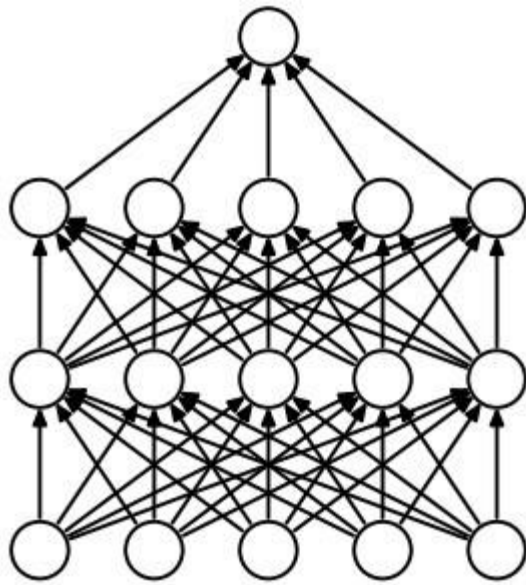
Создание глубокой сверточной сети

```
# Создаем модель
model = Sequential()
# Первый сверточный слой
model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=(3, 32, 32),
                        activation='relu'))
# Второй сверточный слой
model.add(Convolution2D(32, 3, 3, activation='relu'))
# Слой подвыборки
model.add(MaxPooling2D(pool_size=(2, 2)))
# Слой регуляризации
model.add(Dropout(0.25))
```

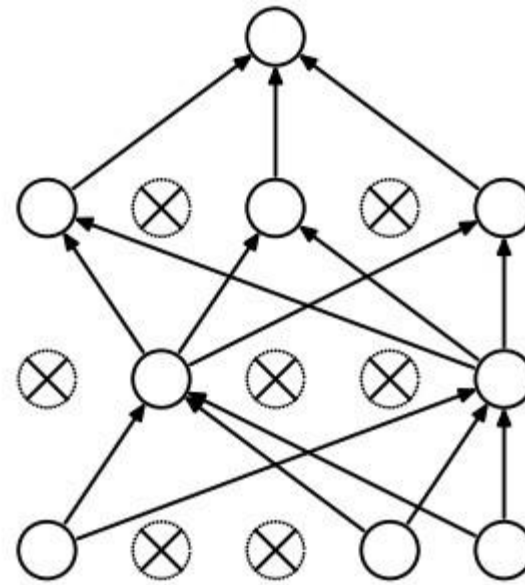
Dropout

- Dropout нужен для предотвращения переобучения
- При предъявлении каждого объекта обучения, нейроны с заданной вероятностью отключаются
- Оставшиеся нейроны обучаются распознавать необходимые признаки без участия соседних

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Продолжение сверточной сети

```
# Третий сверточный слой
model.add(Convolution2D(64, 3, 3, border_mode='same',
activation='relu'))
# Четвертый сверточный слой
model.add(Convolution2D(64, 3, 3, activation='relu'))
# Второй слой подвыборки
model.add(MaxPooling2D(pool_size=(2, 2)))
# Слой регуляризации
model.add(Dropout(0.25))
```

Классификатор

```
# Преобразование из двумерного вида в плоский
model.add(Flatten())
# Полносвязный слой
model.add(Dense(512, activation='relu'))
# Слой регуляризации
model.add(Dropout(0.5))
# Выходной слой
model.add(Dense(10, activation='softmax'))
```

Компиляция и обучение сети

```
# Компилируем сеть
model.compile(loss='categorical_crossentropy',
              optimizer='SGD',
              metrics=['accuracy'])

# Обучаем сеть
model.fit(X_train, Y_train,
        batch_size=32,
        nb_epoch=25,
        validation_split=0.1,
        shuffle=True)
```

В обучении выборка вариантов и ответы. Shuffle – перемешивание данных

Пример вывода в процессе обучения

Epoch 1/25

45000/45000 [=====] - 370s - loss:
1.7926 - acc: 0.3386 - val_loss: 1.3840 - val_acc: 0.4936

Epoch 2/25

45000/45000 [=====] - 359s - loss:
1.3837 - acc: 0.4957 - val_loss: 1.2487 - val_acc: 0.5492

Epoch 3/25

45000/45000 [=====] - 366s - loss:
1.2151 - acc: 0.5635 - val_loss: 1.0457 - val_acc: 0.6304

...

Epoch 25/25

45000/45000 [=====] - 354s - loss:
0.6231 - acc: 0.7850 - val_loss: 0.6644 - val_acc: 0.7750

Проверка сети на тестовых данных

```
# Проверяем точность модели на тестовых данных
scores = model.evaluate(X_test, Y_test, verbose=0)

# Печатаем точность
print("Точность работы на тестовых данных: %.2f%%" %
      (scores[1]*100))
```

Пример вывода:

Точность работы на тестовых данных: 76.29%

Задание

- Реализовать свой пример обучения и работы сверточной сети
- Аналог:
<https://colab.research.google.com/drive/1B-HSO8E9Wahc1VcrWKKHesdVMTCekXwl?usp=sharing#scrollTo=qrfd6FzSLrE2>