

ИНСТИТУТ
БИОИНФОРМАТИКИ



Классификация микрофотографий гистологических препаратов при помощи свёрточных нейронных сетей

Научный руководитель: Шпильман
Алексей Александрович, СПбАУ РАН
Студент: Гайнуллина А.Н.

Санкт-Петербург
2016

Цели и задачи проекта

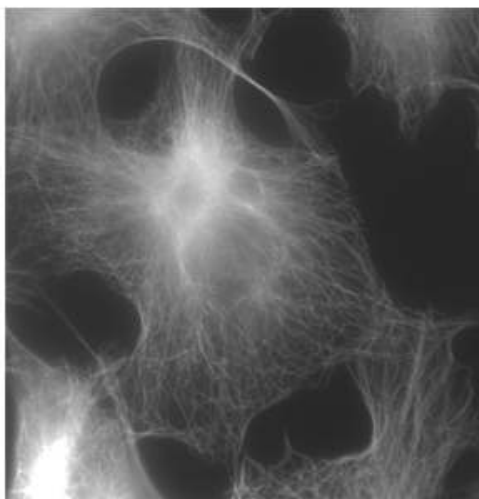
Цель:

создать нейронную сеть для классификации микрофотографий гистологических препаратов на три класса: 1. Taxol01, 2. Taxol1, 3. Control.

Задачи:

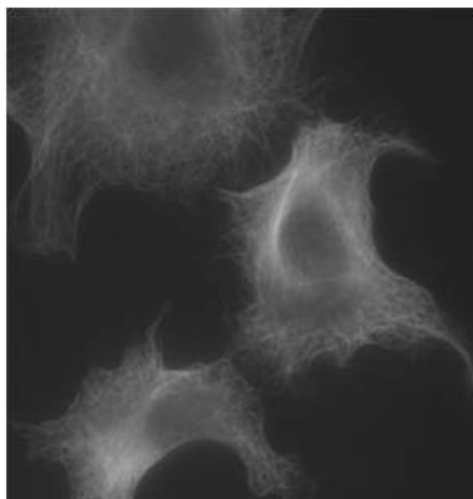
1. подготовить обучающую и тестовую выборки микрофотографий;
2. написать свёрточную нейронную сеть (python).

Контроль

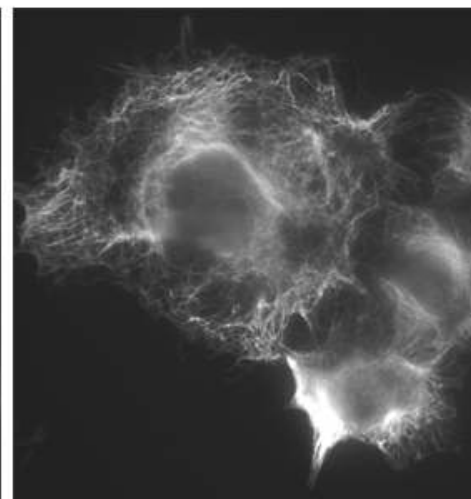


Taxol

0.1 mM



1 mM



Предварительная обработка данных (создание обучающего и тестового множеств)

Исходная выборка - 200 фотографий 1376x1038
каждого класса...



...были разбиты на **фрагменты** 100x100,
обязательно содержащие **изображение** клеток,
и **нормализованы** по яркости:

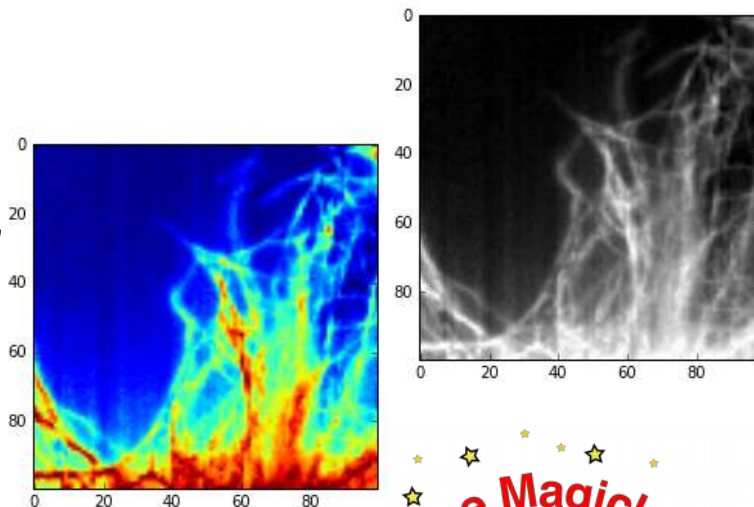
$$X_{i, 0 \text{ to } 1} = \frac{X_i - X_{\text{Min}}}{X_{\text{Max}} - X_{\text{Min}}}$$



Обучающее множество:
6 000 изображений 100x100

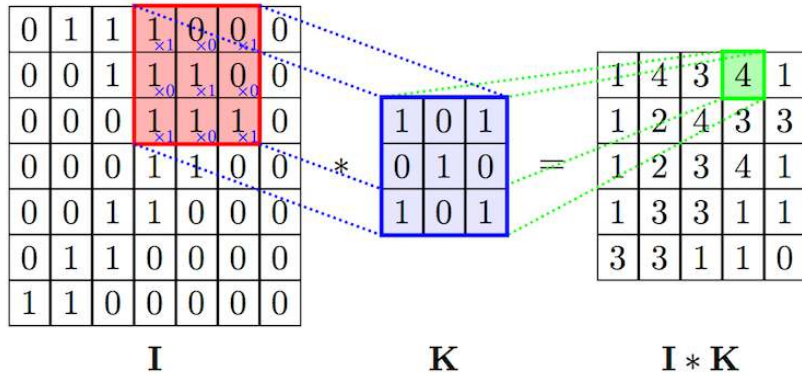
Тестовое множество:
1 200 изображений 100x100

(для каждого класса)

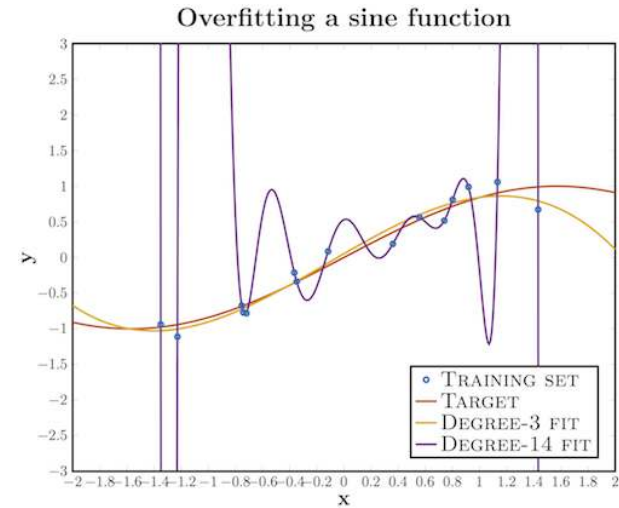


Функции некоторых слоёв нейронной сети

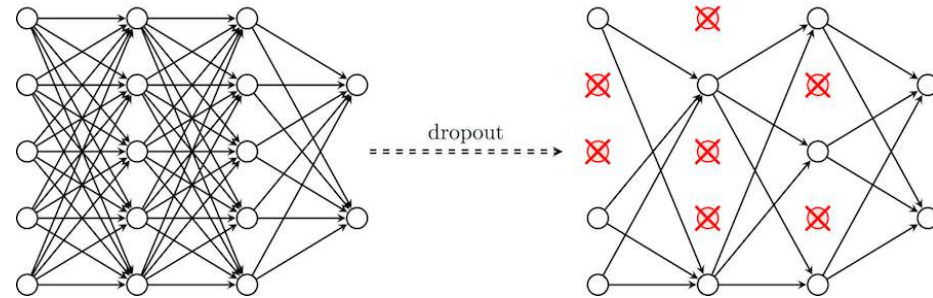
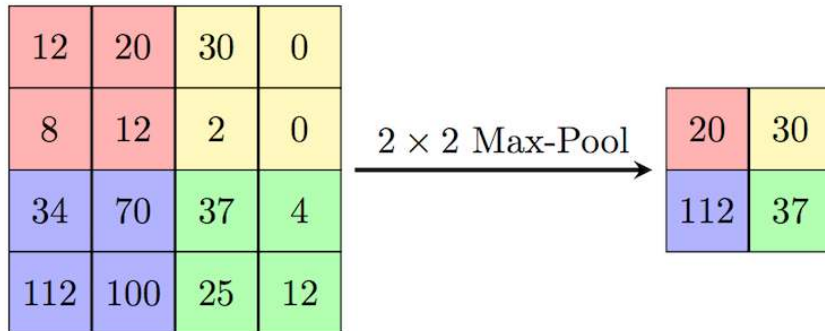
Свёртка



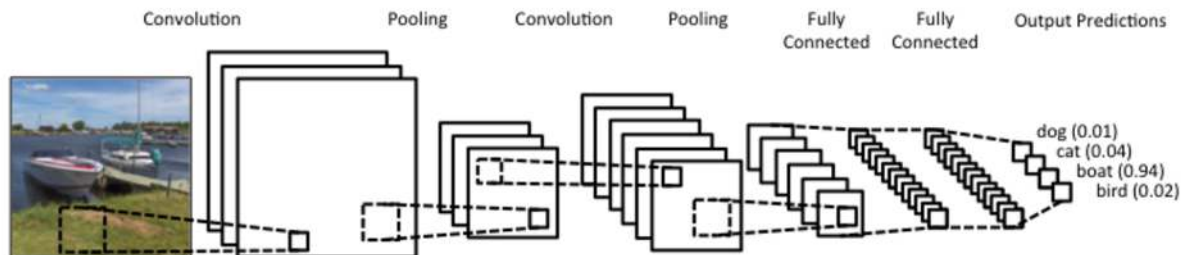
Регуляризация



Субдискретизация



Свёрточная нейронная сеть (keras, tensorflow backend)



```
model = Sequential()

model.add(Convolution2D(nb_filters, nb_conv, nb_conv,
                        border_mode='valid',
                        input_shape=(1, img_rows, img_cols)))
model.add(Activation('relu'))
model.add(Convolution2D(nb_filters, nb_conv, nb_conv))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(nb_pool, nb_pool)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer='adadelta',
              metrics=['accuracy'])
```

In [9]: model.summary()

Layer (type)	Output Shape	Param #	Connected to
convolution2d_1 (Convolution2D)	(None, 98, 98, 32)	320	convolution2d_input_1[0][0]
activation_1 (Activation)	(None, 98, 98, 32)	0	convolution2d_1[0][0]
convolution2d_2 (Convolution2D)	(None, 96, 96, 32)	9248	activation_1[0][0]
activation_2 (Activation)	(None, 96, 96, 32)	0	convolution2d_2[0][0]
maxpooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0	activation_2[0][0]
dropout_1 (Dropout)	(None, 48, 48, 32)	0	maxpooling2d_1[0][0]
Flatten_1 (Flatten)	(None, 73728)	0	dropout_1[0][0]
dense_1 (Dense)	(None, 128)	9437312	Flatten_1[0][0]
activation_3 (Activation)	(None, 128)	0	dense_1[0][0]
dropout_2 (Dropout)	(None, 128)	0	activation_3[0][0]
dense_2 (Dense)	(None, 3)	387	dropout_2[0][0]
activation_4 (Activation)	(None, 3)	0	dense_2[0][0]
Total params: 9447267			

Результаты

Нейронная сеть
(batch_size = 128, nb_epoch = 24):

Классификация по трём классам: 52%

Классификация **Control:Taxol01**: 78.5%

Классификация **Control:Taxol1**: 81%

Классификация **01Taxol:1Taxol**: 59%

Ручная классификация:

Классификация по трём классам: 64%

Классификация **Control:Taxol01**: *NA*

Классификация **Control:Taxol1**: 88%

Классификация **01Taxol:1Taxol**: *NA*

Спасибо за внимание!