

Полносвязные сети

Обучение линейной регрессии в Keras

```
In # подключаем Keras
from tensorflow import keras

# создаём модель
model = keras.models.Sequential()
# указываем, как устроена нейронная сеть
# units – число нейронов в слое
# input_dim – количество входов в слое
model.add(keras.layers.Dense(units=1, input_dim=features.shape[1]))
# указываем, как обучается нейронная сеть
model.compile(loss='mean_squared_error', optimizer='sgd')

# обучаем модель
model.fit(features, target)
```

Обучение логистической регрессии в Keras

```
In # подключаем Keras
from tensorflow import keras

# создаём модель
model = keras.models.Sequential()
# указываем, как устроена нейронная сеть
# units – число нейронов в слое
# input_dim – количество входов в слое
# activation – функция активации
model.add(keras.layers.Dense(units=1, input_dim=features_train.shape[1],
activation='sigmoid'))
# указываем, как обучается нейронная сеть
model.compile(loss='binary_crossentropy', optimizer='sgd')

# обучаем модель
model.fit(features, target)
```

Обучение полносвязной нейронной сети в Keras

```
In # подключаем Keras
from tensorflow import keras

# создаём модель
model = keras.models.Sequential()
# указываем, как устроена нейронная сеть.
# Имеем два слоя: первый состоит из 10 нейронов, второй – из одного
# units – число нейронов в слое
# input_dim – количество входов в слое
# activation – функция активации
model.add(keras.layers.Dense(units=10, input_dim=features_train.shape[1],
activation='sigmoid'))
model.add(keras.layers.Dense(units=1, activation='sigmoid'))

# указываем, как обучается нейронная сеть
model.compile(loss='binary_crossentropy', optimizer='sgd', metrics=['acc'])

# обучаем модель
model.fit(features, target)
```

Работа с изображениями в Python

In

```
import numpy as np
from PIL import Image

# Чтение изображения
image = Image.open('image.png')
image_array = np.array(image)
print(image_array)

# Построение изображения
plt.imshow(image_array)

# Построение чёрно-белого изображения
plt.imshow(image_array, cmap='gray')

# Добавление шкалы цвета к изображению
plt.colorbar()
```