

Библиотека Tensorflow

Руднев Виктор

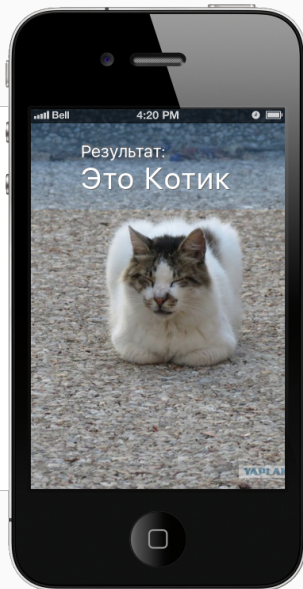
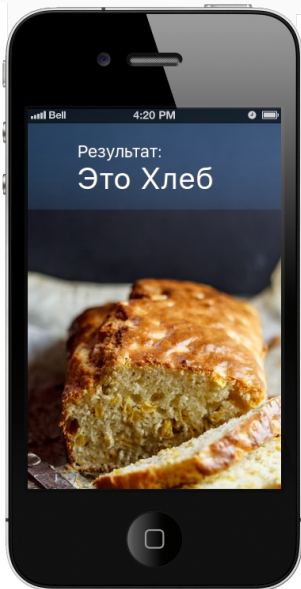
5 декабря 2017 г.

ВМК МГУ

1. Введение
2. Пример
3. Заключение

Введение

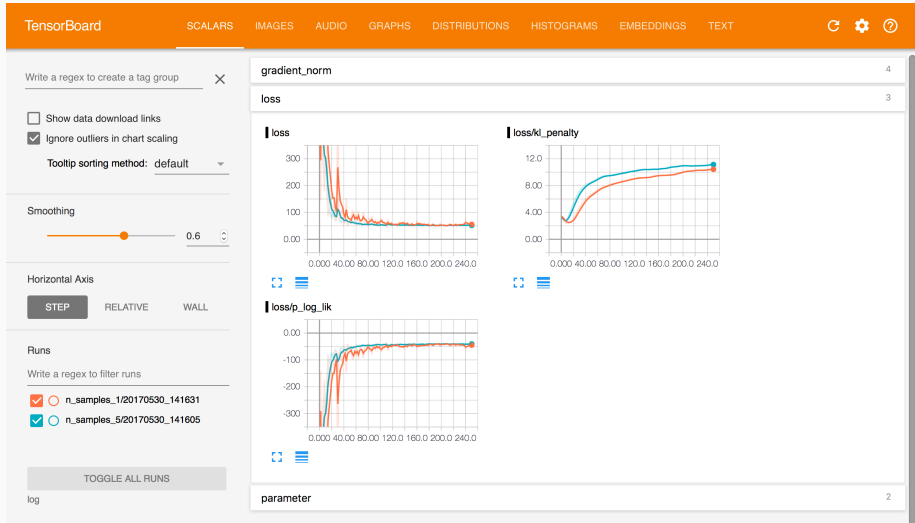
Зачем изучать Tensorflow?



Зачем изучать Tensorflow?

- Грубо говоря, это NumPy+SciPy с поддержкой GPU
- Считает за Вас производные
- Удобно встраивается в разные приложения
- Есть удобная визуализация (TensorBoard)

TensorBoard



TensorBoard

TensorBoard

SCALARS

IMAGES

AUDIO

GRAPHS

DISTRIBUTIONS

HISTOGRAMS

EMBEDDINGS

TEXT



Write a regex to create a tag group



Horizontal Axis

STEP

RELATIVE

WALL

Runs

Write a regex to filter runs

☒ ☐ n_samples_1/20170530_141631

☒ ☐ n_samples_5/20170530_141605

TOGGLE ALL RUNS

log

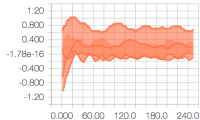
gradient

8

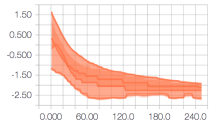
parameter

4

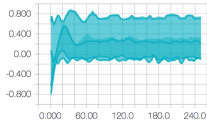
parameter/posterior/qw/loc/0
n_samples_1/20170530_141631



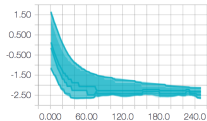
parameter/posterior/qw/unconstrained_scale/0
n_samples_1/20170530_141631



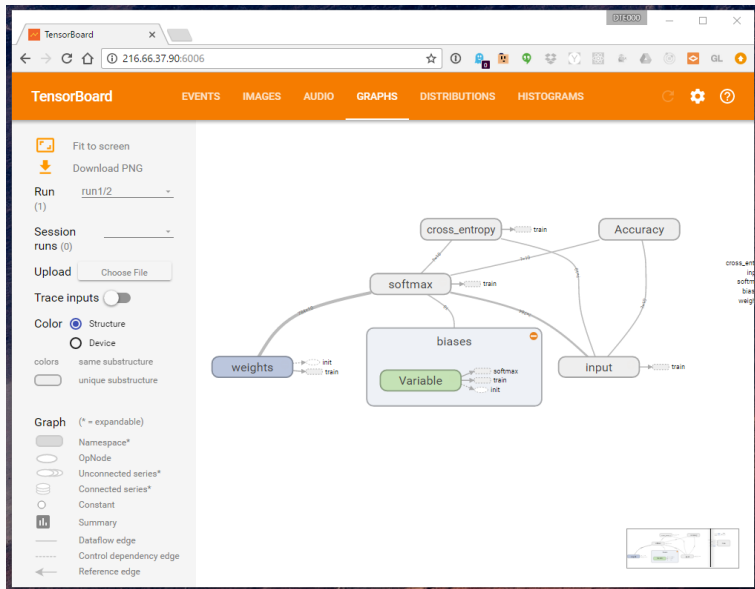
parameter/posterior/qw/loc/0
n_samples_5/20170530_141605



parameter/posterior/qw/unconstrained_scale/0
n_samples_5/20170530_141605



TensorBoard



Пример

Напомню, что мультиклассовую логистическую регрессию можно определить так:

$$P(y = j \mid x) = \frac{\exp(W_j^T x)}{\sum_k \exp(W_k^T x)} = \text{SoftMax}(W^T x)_j,$$

где x — признаки, а W — матрица весов.

Тогда функция потерь будет выглядеть так:

$$\text{Loss} = - \sum_j \log \left(\text{SoftMax}(W^T x^{(j)})_{y^{(j)}} \right),$$

где $x^{(j)}$, $y^{(j)}$ — признаки и отклик j -ого объекта.

Пусть, y — one-hot представление откликов, тогда

```
logits = X.dot(w.T)
proba = np.exp(logits)
proba /= proba.sum(1, keepdims=True)

loss = -np.log(proba*y).mean()
```

Почти то же, но на Tensorflow

NumPy

```
logits = X.dot(w.T)
proba = np.exp(logits)
proba /= proba.sum(1, keepdims=True)

loss = -np.log(proba*y).mean()
```

Tensorflow

```
logits = tf.matmul(X, w)
proba = tf.nn.softmax(logits)

loss = tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(logits, y)
)
```

„Всё остальное”

```
import tensorflow as tf
```

```
train_X, train_Y, test_X, test_Y = load_data()
```

```
num_features = train_X.shape[1]
```

```
num_classes = train_Y.shape[1]
```

„Всё остальное”

```
X = tf.placeholder(tf.float32, [None, num_features])
y = tf.placeholder(tf.float32, [None, num_classes])

w = tf.Variable(tf.zeros([num_features, num_classes]))

logits = X.dot(w.T)
proba = np.exp(logits)
proba /= proba.sum(1, keepdims=True)

loss = tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(logits, y)
)

opt = tf.train.AdamOptimizer(0.01).minimize(loss)
# Есть, естественно, и SGD:
# opt = tf.train.GradientDescentOptimizer(0.01).minimize(loss)
```

„Всё остальное”

```
with tf.Session() as sess:
    init = tf.global_variables_initializer()
    sess.run(init)

    num_epochs = 40
    for i in range(num_epochs):
        sess.run(opt, feed_dict={X: train_X, Y: train_Y})

    accuracy = tf.reduce_mean(
        tf.cast(
            tf.equal(tf.argmax(proba, 1), tf.argmax(y, 1)),
            tf.float32
        )
    )
    accuracy_value = sess.run(
        accuracy,
        feed_dict={X: test_X, Y: test_Y}
    )
    print('accuracy:', accuracy_value)
```

Заключение

Библиотека полезная, но объемная.

Так что изучайте, читайте tutorиалы, например, по адресу

[`www.tensorflow.org/tutorials/`](http://www.tensorflow.org/tutorials/)

Вопросы?