

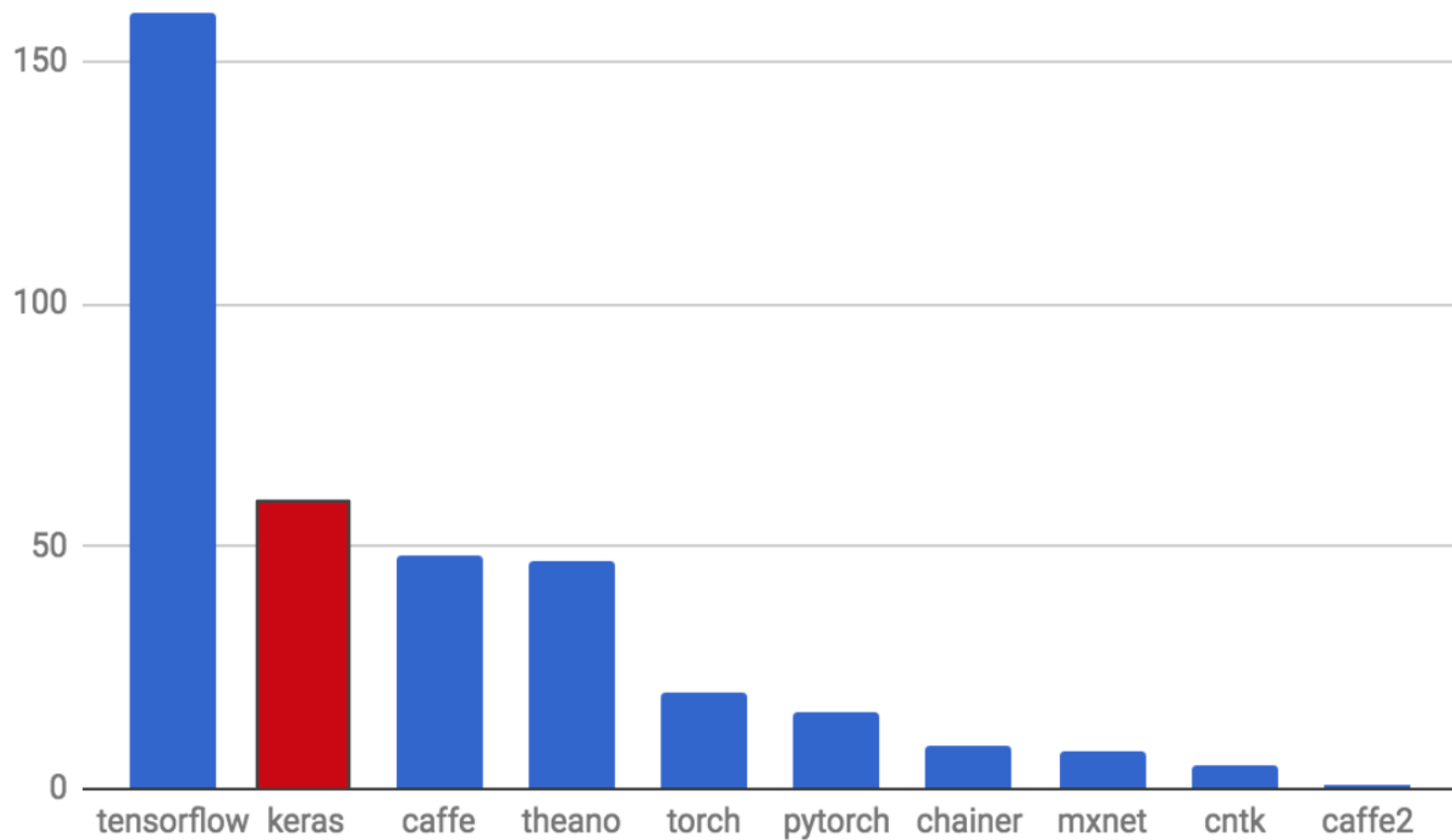
Начало работы с Keras

Гончаров Павел
Нестереня Игорь

kaliostrogooblin3@gmail.com
nesterione@gmail.com

Популярность фреймворков

arXiv mentions, October 2017



Установка

Before installing Keras, please install one of its backend engines: TensorFlow, Theano, or CNTK. We recommend the TensorFlow backend.

<https://keras.io/#installation>

<https://www.tensorflow.org/install/>

- cuDNN (recommended if you plan on running Keras on GPU).
- HDF5 and h5py (required if you plan on saving Keras models to disk).
- graphviz and pydot (used by [visualization utilities](#) to plot model graphs).

```
pip install keras
```

Keras. Модели

Sequential

```
model = Sequential()
model.add(Dense(64, input_dim=784))
model.add(Activation('relu'))

model.add(Dense(64))
model.add(Activation('relu'))

model.add(Dense(10))
model.add(Activation('softmax'))

# For a binary classification problem
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Functional

```
# This returns a tensor
inputs = Input(shape=(784,))

# a layer instance is callable on a tensor, and returns a
# tensor
x = Dense(64, activation='relu')(inputs)
x = Dense(64, activation='relu')(x)
predictions = Dense(10, activation='softmax')(x)

# This creates a model that includes
# the Input layer and three Dense layers
model = Model(inputs=inputs, outputs=predictions)
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Инициализаторы

В keras `glorot_normal` используется по умолчанию, можно настроить другой инициализатор:

```
kernel_initializer = keras.initializers.RandomNormal(mean=0.0, stddev=0.05, seed=None)  
kernel_initializer = keras.initializers.Zeros()
```

Нельзя инициализировать сеть одинаковыми значениями, сеть не будет обучаться

<https://keras.io/initializers/>

Оптимизаторы

Выбор и настройка алгоритма оптимизации

```
model.compile(  
    #optimizer='sgd',  
    optimizer=optimizer,  
    loss='categorical_crossentropy',  
    metrics=['accuracy'])  
  
from keras.optimizers import SGD  
optimizer =SGD(lr=0.1, momentum=0, nesterov=False)
```

<https://keras.io/optimizers/>

SGD

- Batch mode: $N=B$, one epoch is same as one iteration.
- Mini-batch mode: $1 < B < N$, one epoch consists of N/B iterations.
- Stochastic mode: $B=1$, one epoch takes N iterations.

N – размер обучающей выборки

B – размер мини-батча

```
history = model.fit(x_train, y_train, batch_size=256, epochs=10)
```

Overfitting и проверка модели

Метод `fit` вернёт объект `history` где можно посмотреть промежуточные результаты. Также в `fit` можно передать данные для валидации.

```
history = model.fit(x_train, y_train, validation_data=(x_test, y_test), batch_size=256, epochs=10)
```

```
score = model.evaluate(x_test, y_test, verbose=0)  
print('Test loss:', score[0])  
print('Test accuracy:', score[1])
```

Регуляризация

В keras доступны L1 и L2 регуляризация

```
keras.regularizers.l1(0.)  
keras.regularizers.l2(0.)  
keras.regularizers.l1_l2(0.)
```

Регуляризация настраивается для каждого слоя

```
model.add(Dense(512, kernel_regularizer=keras.regularizers.l1(0.)))
```

<https://keras.io/regularizers/>

Функции активации

```
model.add(Dense(512, ))  
model.add(Activation('sigmoid'))
```

```
model.add(Dense(num_classes,))  
model.add(Activation('softmax'))
```

Softmax и categorical crossentropy

$$y_i = \frac{e^{z_i}}{\sum e^{z_j}}$$

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^n y_i \log(p_i)$$

Binomial cross-entropy loss is a special case of multinomial cross-entropy

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(p_{ij})$$

<https://www.quora.com/Artificial-Neural-Networks-Why-do-we-use-softmax-function-for-output-layer>

Сохранение и загрузка модели

```
model.save('nn_dence.h5')
```

```
model.save_weights('nn_dence_weights.h5')
```

```
model = load_model('./nn_dence.h5')
```

```
model.load_weights('./nn_dence_weights.h5')
```
