Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных Систем и Сетей

Кафедра Информатики

Дисциплина Интеллектуальный анализ данных

ОТЧЁТ
по лабораторной работе №1 по теме

ЗНАКОМСТВО С ПРОГРАММНЫМИ СРЕДСТВАМИ МАШИННОГО ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Магистрант: Н.Р. Ровдо

МИНСК 2019

Примеры команд и результаты, демонстрирующие успешность установки.

1. Установить python (3.x)

```
PS C:\WINDOWS\system32> cinst python3 -y
Chocolatey v0.10.13
2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
 - A pending system reboot request has been detected, however, this is
   being ignored due to the current Chocolatey configuration.  If you
   want to halt when this occurs, then either set the global feature
   using:
     choco feature enable -name=exitOnRebootDetected
   or pass the option --exit-when-reboot-detected.

Installing the following packages:
python3
By installing you accept licenses for the packages.

python3 v3.8.0 [Approved]
python3 package files install completed. Performing other installation steps.
Installing 64-bit python3...
python3 has been installed.
Installed to: 'C:\Python38'
  python3 can be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
 see the changes (or in powershell/cmd.exe just type `refreshenv`).
 The install of python3 was successful.
  Software installed as 'exe', install location is likely default.

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Enjoy using Chocolatey? Explore more amazing features to take your
experience to the next level at
 https://chocolatey.org/compare
```

2. Установить и настроить tensorflow и все необходимые пакеты

```
C:\Users\Harwister>C:\Users\Harwister\AppData\Local\Programs\Python\Python36\python.exe -m pip install tensorflow
Requirement already satisfied: tensorflow in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (1.12.0)
Requirement already satisfied: absl-py>=0.1.6 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (0.6.1)
Requirement already satisfied: six>=1.10.0 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.11.0)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.0.5)
Requirement already satisfied: numpy>=1.13.3 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.15.4)
Requirement already satisfied: grpcio>=1.8.6 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.16.1)
Requirement already satisfied: keras-applications>=1.0.6 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.0.6)
Requirement already satisfied: gast>=0.2.0 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: wheel>=0.26 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (0.32.2)
Requirement already satisfied: astor>=0.6.0 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (0.7.1)
Requirement already satisfied: protobuf>=3.6.1 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (3.6.1)
Requirement already satisfied: tensorboard<1.13.0,>=1.12.0 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorflow) (1.12.0)
Requirement already satisfied: h5py in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras-applications>=1.0.6->tensorflow) (2.8.0)
Requirement already satisfied: setuptools in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from protobuf>=3.6.1->tensorflow) (39.0.1)
Requirement already satisfied: werkzeug>=0.11.10 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorboard<1.13.0,>=1.12.0->tensorflow) (0.14.
)
Requirement already satisfied: markdown>=2.6.8 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from tensorboard<1.13.0,>=1.12.0->tensorflow) (3.0.1)
WARNING: You are using pip version 19.1, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Harwister>_
```

3. Установить keras

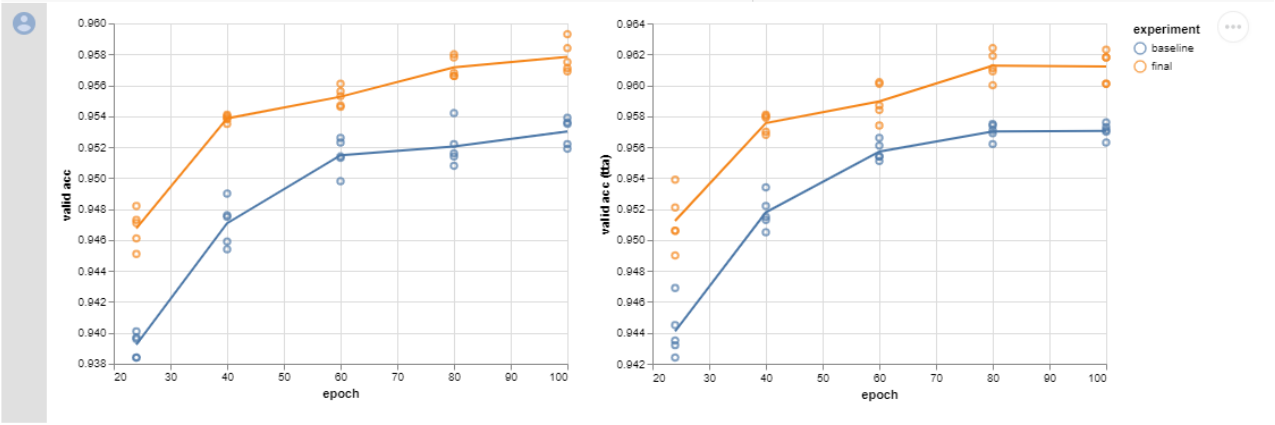
```
C:\Users\Harwister>C:\Users\Harwister\AppData\Local\Programs\Python\Python36\python.exe -m pip install keras
Requirement already satisfied: keras in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (2.2.4)
Requirement already satisfied: scipy>=0.14 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (1.1.0)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (1.0.5)
Requirement already satisfied: h5py in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (2.8.0)
Requirement already satisfied: numpy>=1.9.1 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (1.15.4)
Requirement already satisfied: six>=1.9.0 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (1.11.0)
Requirement already satisfied: pyyaml in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (3.13)
Requirement already satisfied: keras-applications>=1.0.6 in c:\users\harwister\appdata\local\programs\python\python36\lib\site-packages (from keras) (1.0.6)
WARNING: You are using pip version 19.1, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

4. Выбрать демонстрационный и запустить пример (cifar10 CNN, cifar10 ResNet)

```
[ ]  import matplotlib.pyplot as plt

     history_dict = history.history
     loss_values = history_dict['loss']
     val_loss_values = history_dict['val_loss']

     epochs = range(1, len(history_dict['acc']) + 1)

     plt.plot(epochs, loss_values, 'bo', label='Training loss')
     plt.plot(epochs, val_loss_values, 'b', label='Validation loss')
     plt.title('Training and validation loss')
     plt.xlabel('Epochs')
     plt.ylabel('Loss')
     plt.legend()

     plt.show()
```

▾ Results

```
[ ]  scale = alt.Scale(zero=False)
     c = alt.Chart(data).encode(x=alt.X('epoch', scale=scale), color='experiment')

     c1=(c.mark_point().encode(y=alt.Y('valid_acc',scale=scale, axis=alt.Axis(title='valid acc'))) +
     c.mark_line().encode(y=alt.Y('mean(valid_acc)',scale=scale)))

     c2=(c.mark_point().encode(y=alt.Y('tta_acc',scale=scale, axis=alt.Axis(title='valid acc (tta)'))) +
     c.mark_line().encode(y=alt.Y('mean(tta_acc)',scale=scale)))

     c1 | c2
```



Ответить на вопросы.

Как задать модель нейронной сети. Какие есть интерфейсы и их параметры?

Define the Model - 3 layers (2 hidden layers with 100 nodes each and 1 output layer with a single)

```
:  #defifne a sequentail Model
   model = Sequential()

   #Hidden Layer-1
   model.add(Dense(100,activation='relu',input_dim=8,kernel_regularizer=l2(0.01)))
   model.add(Dropout(0.3, noise_shape=None, seed=None))

   #Hidden Layer-2
   model.add(Dense(100,activation = 'relu',kernel_regularizer=l2(0.01)))
   model.add(Dropout(0.3, noise_shape=None, seed=None))

   #Output layer
   model.add(Dense(1,activation='sigmoid'))
```

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=x_train.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
model.add(Activation('softmax'))
```

Как задать весовые коэффициенты нейронной сети?

```
model.layers[i].set_weights(listOfNumpyArrays)
model.get_layer(layerName).set_weights(...)
model.set_weights(listOfNumpyArrays)
```

Как задать полносвязный слой нейронной сети?

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
```

Как задать свёрточный слой нейронной сети?

```
#create model
model = Sequential ()

#add model
models.add (Conv2D (64, kernel_size = 3, activation = 'relu',
input_shape = (28,28,1)))
model.add (Conv2D (32, kernel_size = 3, activation = 'relu' ))
model.add (Flatten ())
model.add (Плотная (10, активация = 'softmax)))
```

Какие есть средства для работы с рекуррентными нейросетями?

Outputs and states

By default, the output of a RNN layer contain a single vector per sample. This vector is the RNN cell output corresponding to the last timestep, containing information about the entire input sequence. The shape of this output is `(batch_size, units)` where `units` corresponds to the `units` argument passed to the layer's constructor.

A RNN layer can also return the entire sequence of outputs for each sample (one vector per timestep per sample), if you set `return_sequences=True`. The shape of this output is `(batch_size, timesteps, units)`.

```
model = tf.keras.Sequential()
model.add(layers.Embedding(input_dim=1000, output_dim=64))

# The output of GRU will be a 3D tensor of shape (batch_size, timesteps, 256)
model.add(layers.GRU(256, return_sequences=True))

# The output of SimpleRNN will be a 2D tensor of shape (batch_size, 128)
model.add(layers.SimpleRNN(128))

model.add(layers.Dense(10, activation='softmax'))

model.summary()
```

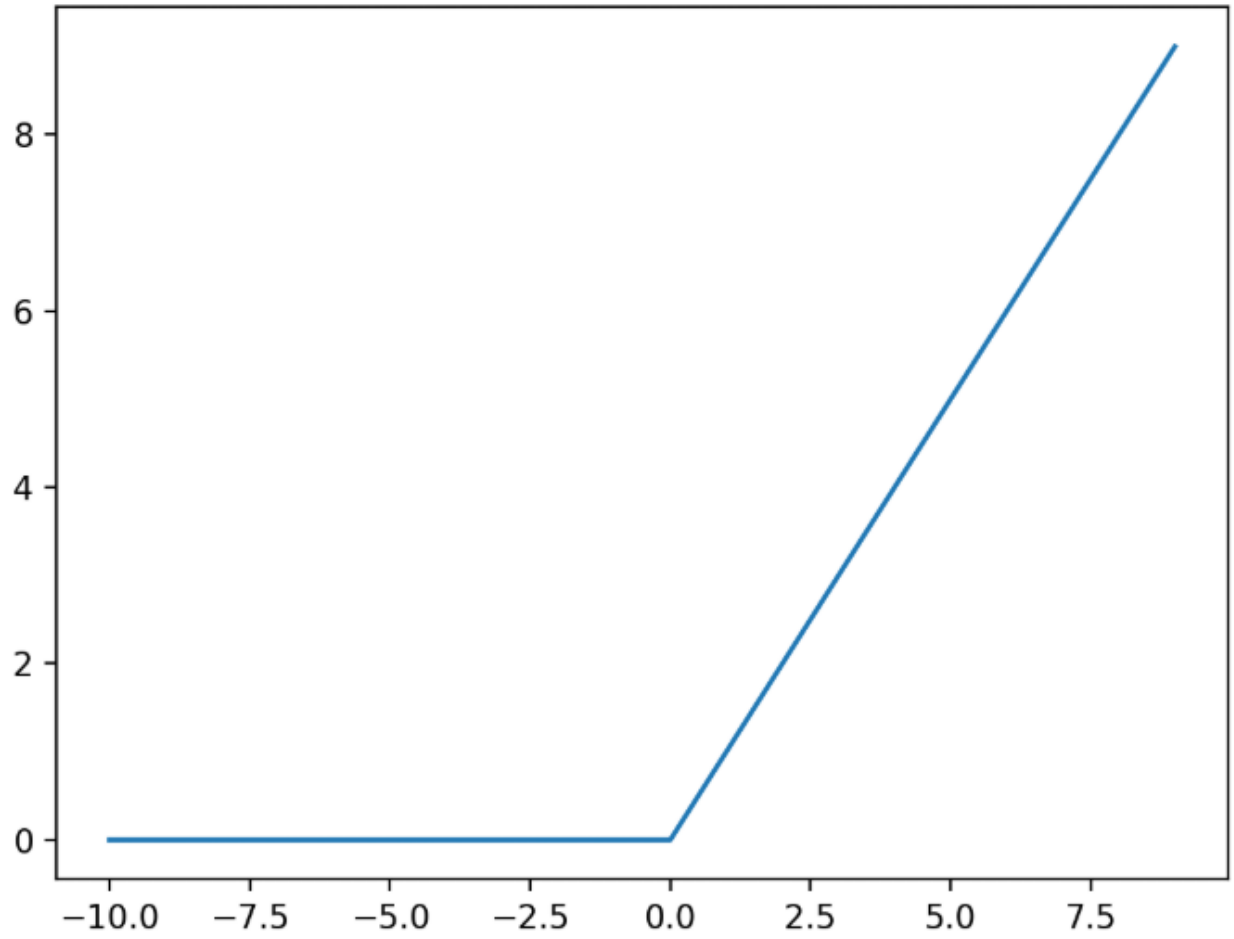
Как задать функцию активации нейронной сети и какие поддерживаются в keras?

```
model = tf.keras.Sequential()
model.add(layers.Embedding(input_dim=1000, output_dim=64))

# The output of GRU will be a 3D tensor of shape (batch_size, timesteps, 256)
model.add(layers.GRU(256, return_sequences=True))

# The output of SimpleRNN will be a 2D tensor of shape (batch_size, 128)
model.add(layers.SimpleRNN(128))

model.add(layers.Dense(10, activation='softmax'))

model.summary()
```

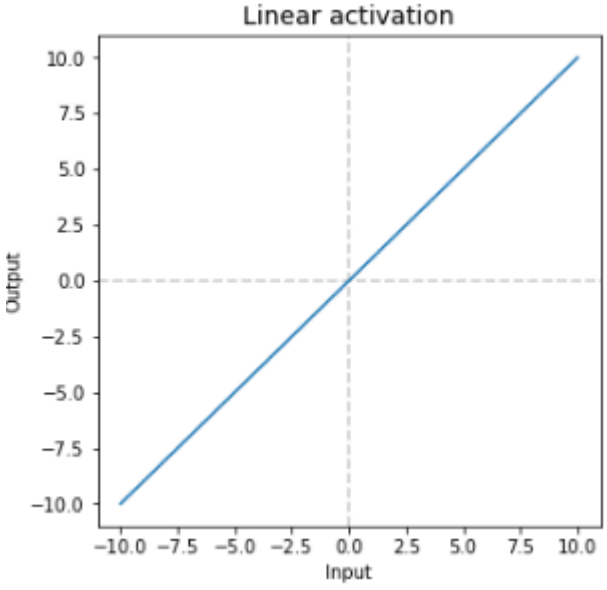
Available activations

- Elu
- Softmax
- Selu
- Softplus
- Softsign
- Relu
- tanh
- sigmoid
- hard_sigmoid
- exponential
- linear

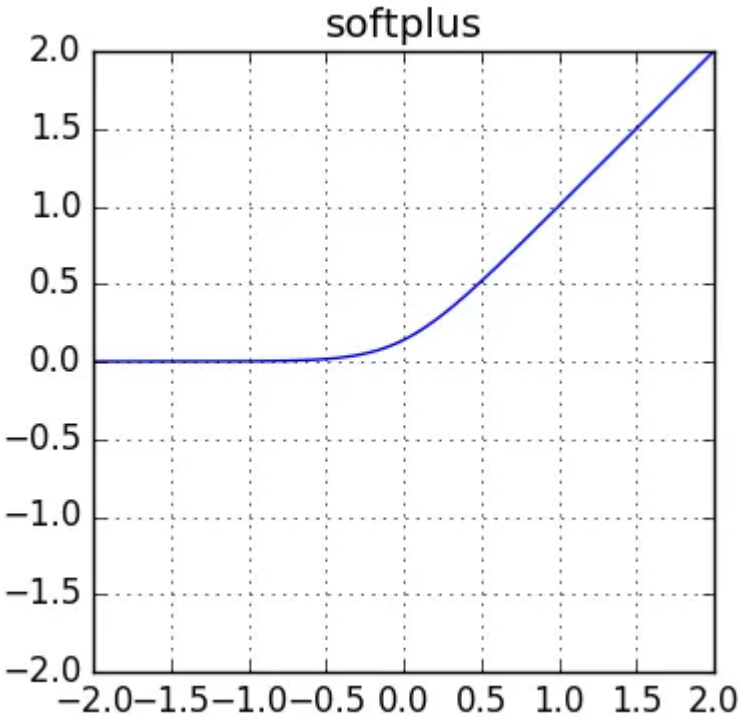
Чем отличается linear от ReLU, softplus?


8
6
4
2
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

Relu


Linear activation
Output
Input
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0 7.5 10.0

Linear



Softplus

Как задать функцию ошибки\ потерь нейронной сети?

Usage of loss functions

A loss function (or objective function, or optimization score function) is one of the two parameters required to compile a model:

```
model.compile(loss='mean_squared_error', optimizer='sgd')
```

Чем отличается mean_squared_error от cosinus_proxmity, по каким формулам они вычисляются?

$$\mathrm{MSE} = \frac{1}{n}\sum_{i-1}^{n}(Y_i - \hat{Y}_i)^2.$$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Как задать метод обучения нейронной сети?

```
model.compile(loss=custom_objective,
              optimizer='adam',
              metrics=['accuracy'])
```

Чем отличается SGD от rprop, Adadelta, Adam; nesterov от momentum?

Adam [source]

```
keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, amsgrad=False)
```

Adam optimizer.

Default parameters follow those provided in the original paper.

SGD [source]

```
keras.optimizers.SGD(learning_rate=0.01, momentum=0.0, nesterov=False)
```

Stochastic gradient descent optimizer.

Includes support for momentum, learning rate decay, and Nesterov momentum.

Adadelta [source]

```
keras.optimizers.Adadelta(learning_rate=1.0, rho=0.95)
```

Adadelta optimizer.

Adadelta is a more robust extension of Adagrad that adapts learning rates based on a moving window of gradient updates, instead of accumulating all past gradients. This way, Adadelta continues learning even when many updates have been done. Compared to Adagrad, in the original version of Adadelta you don't have to set an initial learning rate. In this version, initial learning rate and decay factor can be set, as in most other Keras optimizers.

It is recommended to leave the parameters of this optimizer at their default values.

RMSprop [source]

```
keras.optimizers.RMSprop(learning_rate=0.001, rho=0.9)
```

RMSProp optimizer.

It is recommended to leave the parameters of this optimizer at their default values (except the learning rate, which can be freely tuned).

Nadam [source]

```
keras.optimizers.Nadam(learning_rate=0.002, beta_1=0.9, beta_2=0.999)
```

Nesterov Adam optimizer.

Much like Adam is essentially RMSprop with momentum, Nadam is RMSprop with Nesterov momentum.

Как указать обучающую выборку?

```
history = model.fit(x_train, y_train,
                    batch_size=batch_size,
                    epochs=epochs,
                    verbose=1,
                    validation_split=0.1)
```

СПИСОК ИСТОЧНИКОВ

[1] Оф. сайт Chocolatey. — Электронный ресурс. — Режим доступа: https://chocolatey.org/ — Дата доступа: 12.09.2019.

[2] Оф. сайт Python. — Электронный ресурс. — Режим доступа: https://www.python.org/ — Дата доступа: 12.09.2019.

[3] Оф. сайт PIP. — Электронный ресурс. — Режим доступа: https://pypi.org/project/pip/ — Дата доступа: 12.09.2019.

[4] Документация виртуальной среды Python. — Электронный ресурс. — Режим доступа: https://docs.python.org/3/library/venv.html — Дата доступа: 12.09.2019.

[5] Пример глубокого обучения с помощью библиотеки Keras. — Электронный ресурс. — Режим доступа: https://keras.io — Дата доступа: 12.09.2019.