

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет

Компьютерных сетей и систем

Обработка больших объемов информации

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ 2

по предмету «Интеллектуальный анализ информации»

Студент
гр. 758601
Лимонтов А. С.

Проверил
Ивашенко В. П.

Цель	2
Назначение платформы Keras	2
Установка	2
Преимущества Keras	3
Keras модели	3
Пример	4
Выводы	6
Литература	7

Цель

Изучить возможности, решаемые задачи, характерные свойства применения платформы Keras. Привести пример создания простой модели.

Назначение платформы Keras

Как известно, Theano и Tensorflow являются широко применимыми платформами для глубоких нейронных сетей. Кроме того, они устанавливают свой “язык” для создания вычислительных графов. Keras, в свою очередь, представляет из себя инструмент (API), который помогает строить модели для глубокого обучения, совмещая такие “блоки”. Исходя из такого определения, сравнение производительности Keras и, например, Tensorflow, не имеет смысла, поскольку под собой Keras будет вызывать методы Tensorflow. Поэтому можно представлять себе Keras как более удобный пользовательский интерфейс при работе с глубоким обучением и используется для быстрого запуска и проверки экспериментов.

Установка

Установка Keras очень проста, достаточно выполнить следующую команду, поскольку это обычный пакет Python:

```
pip install keras
```

Для установки версии Keras, работающей на GPU, стоит выполнить следующую команду:

```
pip install keras-gpu
```

Но для успешной работы Keras, должна быть предустановлена еще хотя бы одна из платформ (Theano или Tensorflow), поскольку является надстройкой над ними.

Преимущества Keras

1. Легкое и быстрое прототипирование решений
2. Поддерживает сверточные и рекуррентные сети, их комбинацию
3. Может запускаться как на CPU, так и на GPU
4. Keras был создан для применения человеком, не машиной: минимизирует необходимое количество совершенных действий для общих задач
5. Легкая настройка при интеграции с такими платформами, как Tensorflow, Theano, CNTK.
6. У Keras очень большое сообщество - это позволяет достаточно быстро находить и устранять ошибки в его работе, создавать более детальную и доступную документацию, быстрее обновлять и расширять исходный код. Помимо прочего, он поддерживается такими крупными корпорациями, как Google, Microsoft, Nvidia, Amazon, которые являются ведущими в этой области
7. Keras модели запускаются на большем количестве платформ, чем какие-либо другие аналоги
8. Поддерживает несколько запущенных бекендов, не оставляя пользователя в рамках одной экосистемы.

Keras модели

Бывают два типа моделей:

1. Sequential. Представляют собой стек слоев. Можно создать такую модель, передав необходимые слои в конструктор:

```
from keras.models import Sequential
from keras.layers import Dense, Activation

model = Sequential([
    Dense(32, input_shape=(784,)),
    Activation('relu'),
```

```
Dense(10),  
Activation('softmax'),  
)
```

Но можно и добавлять их после создания объекта:

```
model = Sequential()  
model.add(Dense(32, input_dim=784))  
model.add(Activation('relu'))
```

2. Functional. Такие модели можно создавать, передавая лишь входные/выходные тензоры:

```
from keras.models import Model  
from keras.layers import Input, Dense
```

```
a = Input(shape=(32,))  
b = Dense(32)(a)  
model = Model(inputs=a, outputs=b)
```

Такая запись создаст все необходимые слои для вычисления b по заданному a.

Помимо базовых моделей, можно создавать свои модели, наследуясь от класса Model.

Пример

Приведем самый простой пример использования Keras:

1. Создадим модель:

```
from keras.models import Sequential  
model = Sequential()
```

И добавим в нее слои:

```
model.add(Dense(units=64, activation='relu',  
input_dim=100))
```

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

Функциями активации в данном примере являются relu и softmax. Конечно, это не единственные доступные функции, существуют другие предоставленные функции активации, например:

a. sigmoid

- b. tanh
- c. linear

Выбор функции активации зависит от решаемой задачи.

2. Скомпилируем модель, указав оптимизатор, функцию для вычисления loss (оптимизируемую функцию) и дополнительные метрики:

```
model.compile(loss=mean_squared_error,  
              optimizer='sgd',  
              metrics=['accuracy'])
```

accuracy не единственная предлагаемая метрика. Можно воспользоваться и другими:

- a. binary_accuracy
- b. categorical_accuracy
- c. sparse_categorical_accuracy
- d. top_k_categorical_accuracy
- e. sparse_top_k_accuracy

3. Обучим модель:

```
model.fit(x_train, y_train, epochs=5,  
          batch_size=32)
```

4. Валидация модели (batch_size - количество примеров, которые обрабатываются за одну итерацию обновления градиента).

```
loss_and_metrics = model.evaluate(x_test,  
y_test, batch_size=128)
```

Помимо всего, функция fit также принимает следующие параметры:

- a. verbose - уровень логирования
- b. callbacks - массив классов, которые вызываются на разных стадиях обучения (например, on_train_start, on_batch_end). Они позволяют изменить параметры в процессе обучения (например, изменить learning rate или прекратить обучение)
- c. validation_split - здесь указывается часть данных для валидации

- d. `shuffle` - стоит ли перемешивать данные перед обучением
- e. `class_weight` - заранее указать веса. Бывает полезно при несбалансированной выборке, чтобы уравнивать шансы у разных классов.

5. Запуск обученной модели:

```
classes = model.predict(x_test, batch_size=128)
```

Выводы

В ходе выполненной лабораторной работы был произведен краткий обзор Keras, были выявлены его преимущества, назначение, разобраны базовые понятия, а также приведен простой пример использования.

Литература

1. How does Keras compare to other Deep Learning frameworks like Tensor Flow, Theano, or Torch? [Электронный ресурс] Режим доступа:
<https://www.quora.com/How-does-Keras-compare-to-other-Deep-Learning-frameworks-like-Tensor-Flow-Theano-or-Torch>. Дата доступа: 01.12.2018
2. Keras Intro [Электронный ресурс] Режим доступа:
<https://keras.io/>. Дата доступа: 01.12.2018
3. Keras [Электронный ресурс] Режим доступа:
<https://en.wikipedia.org/wiki/Keras>. Дата доступа: 01.12.2018
4. Tensorflow on Keras [Электронный ресурс] Режим доступа:
<https://www.tensorflow.org/guide/keras>. Дата доступа: 01.12.2018
5. 7 Steps to Mastering Deep Learning with Keras [Электронный ресурс] Режим доступа:
<https://www.kdnuggets.com/2017/10/seven-steps-deep-learning-keras.html>. Дата доступа: 01.12.2018