

Разработка и использование библиотеки поддержки организации экспериментов в области машинного обучения

Крощенко Александр Александрович

УО «Брестский государственный университет имени А.С. Пушкина»

В настоящее время в научном мире широко используются различные библиотеки и прикладные фреймворки, направленные на решение определенных научно-технических задач. Подобные системы есть и в области машинного обучения, но их применение ограничивается, как правило, узко специализированными задачами (например, обучение глубоких нейронных сетей, реализация генетических алгоритмов и т.д.). В качестве примеров можно назвать такие фреймворки как Caffe, Tensorflow, Theano и др.

Нами был разработан прототип системы поддержки проведения эксперимента в области машинного обучения, основывающийся на абстрактных концептуальных сущностях (например, модель, эксперимент и т.д.). Благодаря такому подходу стало возможным простое расширение системы добавлением в него новых алгоритмов и моделей. Например, иерархия объектов, непосредственно участвующих в обучении нейронных сетей, состоит из сущностей «Слой», конкретизируемой типом слоя сети (например, полносвязный, сверточный, подвыборочный и т.д.), «Алгоритм обучения», конкретизируемой, например, алгоритмом обратного распространения или алгоритмом предварительного обучения и т.д.

В системе существует конфигуратор, отвечающий за загрузку параметров проведения эксперимента из файла специального вида, а также поддерживается интеграция с СУБД SQLite, в базе данных на основе которой сохраняются результаты и параметры обучения. Подобная функциональность упрощает протоколирование результатов вычислительных экспериментов.

Файл специального вида для хранения настроек проведения эксперимента оформляется в соответствии с форматом JSON. Структура файла позволяет описывать модели нейронных сетей произвольной сложности, включающие различные нейронные слои в любом порядке.

Формат файла настроек для выполнения эксперимента (классификатор выборки MNIST [1]) приведен ниже.

```
{
  "layers":
  [
    ["conv", 8, "sigm", [5, 5]],
    ["pooling", [2, 2], "max"],
    ["conv", 16, "sigm", [5, 5]],
    ["pooling", [2, 2], "max"],
    ["fullycon", 100, "sigm"],
    ["fullycon", 10, "sigm"]
  ],
  "train_data": "MNIST",
  "pretrain":
  {
    "max_epochs": 10,
    "batch_size": 50,
    "start_momentum": 0.5,
    "final_momentum": 0.9,
```

```
"weight-loss": 0,  
"rates": [0.01, 0.01, 0.01, 0.01, 0.01],  
"pretrain_type": "reba"  
},  
"backprop-finetune":  
{  
  "max_epochs": 300,  
  "batch_size": 50,  
  "momentum": 0.9,  
  "weight-loss": 0.00001,  
  "rates": [0.8, 0.8, 0.8, 0.8, 0.8]  
},  
"MNIST":  
{  
  "train": "train-images.idx3-ubyte",  
  "test": "t10k-images.idx3-ubyte",  
  "train_labels": "train-labels.idx1-ubyte",  
  "test_labels": "t10k-labels.idx1-ubyte",  
  "path": "D:\\Datasets\\MNIST\\"  
}  
}
```

Результатом разработки указанной библиотеки стала возможность ее непосредственного интегрирования в системы, которые используют интеллектуальные алгоритмы обработки данных.

Например, специализированную программную систему распознавания образов, построенную на основе указанной библиотеки, можно представить следующей схемой (рис. 1).

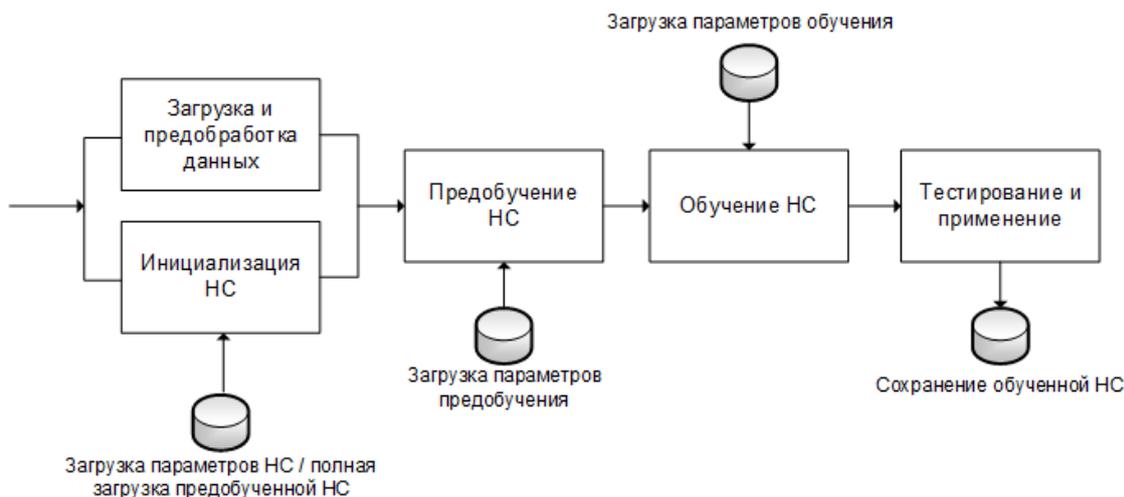


Рисунок 1 - Программная система распознавания образов

В реализации программных модулей на языках более высокого уровня были предусмотрены средства резервного копирования полной информации о нейронных сетях. Подобное копирование выполняется средствами внутриязыковой сериализации и позволяет сохранить не только основные параметры сети (количество слоев, нейронов в каждом слое, функции активации и т.д.), но и настраиваемые параметры (веса и пороги).

Подобная функциональность позволяет сохранить состояние обученной нейронной сети и сформировать пул доступных для использования сетей. Далее при необходимости уже обученная НС может быть загружена из пула и использоваться для решения задач.

Использованные источники

1. LeCun, Y., Cortes, C., Burges, C. MNIST handwritten digit database : <http://yann.lecun.com/exdb/mnist/>.