

РАЗРАБОТКА МЕТОДИКИ ПОСТРОЕНИЯ НЕЙРОННОЙ СЕТИ НА ОСНОВЕ ПРОСТЫХ АДАПТИВНЫХ ЭЛЕМЕНТОВ.

Ефимов Е. Н., Шевгунов Т. Я.

Московский авиационный институт (национальный исследовательский
университет),
г. Москва, Россия

В данной работе рассматриваются искусственные нейронные сети – математические модели, построенные по принципам, заимствованным из сетей нервных клеток живых организмов. Традиционно нейронная сеть представляет собой систему взаимодействующих адаптивных элементов – нейронов, каждый из которых выполняет определенное функциональное преобразование над входным сигналом и генерирует выходной сигнал.

Существуют два основных методологических подхода к описанию процессов нейросетевой обработки сигналов: математический, основанный на описании сети в терминах функциональных преобразований, и системный, представляющий сеть в форме взаимодействующих подсистем, в которых происходит преобразование входных сигналов и сигналов. В настоящем исследовании используется системный подход, на основе которого предложена методика, предполагающая разбиение всей нейронной сети на блоки простых адаптивных элементов. Такие элементы не обязательно являются классическими нейронами – это могут быть любые системы, удовлетворяющие требованиям модели, т.е. способные пропускать через себя сигнал при прямом прохождении и ошибку при обратном прохождении. Практической целью работы является разработка прототипа программного обеспечения для моделирования нейронных сетей, составленных из таких элементов.

Каждый адаптивный элемент представляет собой подсистему, для которой определены вход и выход, а также задано функциональное преобразование, связывающее входной и выходной сигналы. Формально подсистема реализует две передаточные функции: одну для прямого прохождения сигнала со входа на выход, а другую – для обратного распространения ошибки с выхода на вход. Примерами простейших адаптивных элементов являются умножитель, сумматор, разветвитель и безынерционный функциональный элемент, позволяющие построить на их основе более сложный элемент сети – классический нейрон, описываемый адаптивным взвешивающим сумматором с произвольной функцией активации. Элемент так же может хранить информацию о своём состоянии и результаты, полученные в процессе обратного распространения ошибки для последующего их использования при обучении сети. Таким образом, задача обучения сводится к итеративному изменению для каждого элемента параметров его передаточной функции по некоторому известному алгоритму, называемого правилом обучения. Обученная нейронная сеть способна представлять обобщенную зависимость между входными и выходными данными, следовательно, она может генерировать корректный выходной сигнал на основании данных, отсутствующих в использованной при обучении выборке, неполных или зашумленных данных.

Для реализации прототипа программного обеспечения в качестве языка программирования выбран высокоуровневый язык программирования общего назначения *Python*, предоставляющий с одной стороны простую форму записи математических выражений, с другой – широкие возможности в области объектно-ориентированного программирования. Кроме того данный язык обладает свободной лицензией и большой библиотекой готовых разработок для решения ряда вспомогательных задач моделирования.

Базовая часть выполнена в качестве пакета и включает необходимую для работы приложения иерархию классов. Дополнительная часть предназначена для работы в интерактивном режиме в составе системы компьютерной математики *Sage*, пакете

программ со свободной лицензией, объединенных единым пользовательским интерфейсом. Дополнительная часть использует возможности *Sage* для ввода и вывода данных: генерация обучающих последовательностей, отображение графов, построение графиков и таблиц.

Разработанный с использованием *Python* пакет подлежит последующей оптимизации по скорости выполнения с использованием компилятора *Cython*. Для реализации завершено настольного приложения с графическим пользовательским интерфейсом предполагается использование библиотеки *Qt*, а для организации в системе «тонкий клиент» – специализированный пакет разработчика *web*-приложений *Django*.

В качестве примера в работе приведены результаты решения двух задач: задачи аппроксимации сигнала и задачи классификации. В качестве исходного сигнала для задачи аппроксимации используется сумма импульсов в форме функции Гаусса с различными параметрами амплитуды и временной задержки, наблюдаемая в присутствии аддитивного шума. Задача классификации заключается в отнесении наблюдаемых данных к одному из двух классов, описывающих комплексные нормальные случайные процессы с различными средними и дисперсиями.

Разработанный прототип программного обеспечения, основанный на методах системного анализа, позволил успешно решить две представленные задачи, обеспечивая при этом вывод всей необходимой информации о процессе обучения сети.