

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ОБУЧЕНИЯ НЕЙРОСЕТЕЙ ДВОЙСТВЕННОГО ФУНКЦИОНИРОВАНИЯ.

Черноусова Н.В.

Научный руководитель - к.т.н., доцент Сиротинина Н.Ю.

*Сибирский федеральный университет*

Возросшие вычислительные возможности современных компьютеров привели к широкому распространению программ, использующих принципы нейросетевой обработки данных, которые предоставляют единую методологию решения большого числа практически интересных задач. Одна из основных трудностей, препятствующих более широкому распространению нейросетевых технологий, состоит в отсутствии у специалистов навыков по описанию задач из конкретной предметной области в терминах, пригодных для получения нейросетевого решения. Следовательно, изучение нейроинформационных технологий, связанных с процессом обработки и интерпретации информации, должно стать неотъемлемой составляющей учебного процесса подготовки специалистов всех уровней компетенции. Дисциплина «Нейроинформатика» включена в учебный план подготовки специалистов по направлению «Информатика и вычислительная техника». Экспериментальное исследование алгоритмов обучения в разрабатываемом обучающем комплексе позволит ознакомиться с основными методами обучения нейросетей, получить практические навыки по решению прикладных задач из различных предметных областей, оценить влияние параметров нейросети и метода обучения на характеристики работы обученной нейросети.

Модели нейронных сетей различаются между собой моделями нейрона, топологией связей и правилами определения весов (правилами обучения). По структуре сети делятся на однослойные и многослойные. В данной работе рассматривается многослойная нейронная сеть, обучаемая методом обратного распространения ошибки и состоящая из формальных нейронов с адаптивными линейными сумматорами на входе. Количество слоев и количество нейронов в слоях задается пользователем.

Формальным нейроном называется элементарный процессор, используемый в узлах нейронной сети. Математическую модель формального нейрона можно представить уравнением:

$$Y = F(S) = F\left(\sum_{i=1}^n w_i * x_i\right)$$

где:

- $Y$  – выходной сигнал нейрона;
- $F(S)$  – функция выходного блока нейрона;
- $w_i$  – коэффициент – вес  $i$ -го входа;
- $x_i$  –  $i$ -ый входной сигнал;
- $i = 1, 2, 3, \dots, n$  – номер входа нейрона;
- $n$  – число входов.

Базовые функции разрабатываемого программного продукта реализованы в дипломном проекте Невוליной С.С. «Разработка обучающего комплекса для дисциплины нейроинформатика».

В рассматриваемой работе предполагается реализация следующих дополнительных функций:

- расширение набора функций оценки;
- ведение оптимизации шага;

- реализация методов, позволяющих повысить эффективность обучения, в частности метода обучения ParTan.

#### **Функции оценки.**

Задача обучения нейронной сети сводится к нахождению некой функциональной зависимости  $Y=F(X)$  где  $X$  – входной, а  $Y$  – выходной векторы. В общем случае, такая задача при ограниченном наборе входных данных имеет бесконечное множество решений. Для ограничения пространства поиска при обучении ставится задача минимизации целевой функции ошибки сети.

Различие в способах оценки связано с различием требований, накладываемых на обучающую сеть.

Оценивание **по методу наименьших квадратов** является наиболее распространенным методом из существующих стандартных функций оценки, принятых в теории оптимизации.

Пусть пример относится к N-ой задаче. Тогда целевую функцию для метода наименьших квадратов можно записать следующим образом:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2$$

где

- $y_{j,p}^{(N)}$  - реальное выходное состояние нейрона  $j$  выходного слоя N нейронной сети при подаче на ее входы  $p$ -го образа;
- $d_{j,p}$  - идеальное (желаемое) выходное состояние этого нейрона.

Эта оценка является обычным Евклидовым расстоянием от правильного ответа до ответа, выданного сетью. Оценивание по методу наименьших квадратов является универсальным и может применяться практически во всех случаях. Однако, эта оценка плоха тем, что, во-первых, требования при обучении сети не совпадают с требованиями интерпретатора, во-вторых, такая оценка не позволяет оценить уровень уверенности сети в выданном ответе.

При решении задачи классификации возможен другой вид функции оценки - **оценка вида расстояние до множества**.

Когда решается задача классификации, обычно требуется, чтобы с выхода НС соответствующему выбранному классу поступал сигнал максимального уровня, а с остальных - минимального. При этом о надежности полученного результата говорят не только значения выходных сигналов, но и разница между полученным максимальным выходным сигналом и прочими. Если разница между максимальным выходным сигналом и ближайшим к нему по уровню велика, можно говорить о высокой надежности результата (или о более высокой уверенности нейросети при решении предложенного примера).

Формально, оценка вида «расстояние до множества», которая учитывает данную зависимость, описывается следующим образом:

1.  $y_{max} = (y_1, \dots, y_n)$
2.  $E(w) = \sum_1^k f(y_{max}, y_i)$
3.  $f(y_{max}, y_i) = \begin{cases} 0, & \text{если } y_{max} - y_i < U \\ y_{max} - y_i, & \text{если } y_{max} - y_i > U \end{cases}$

где  $(y_1, \dots, y_n)$  - выходные сигналы НС;  
 $y_{max}$  - максимальный выходной сигнал НС;

U - параметр оценки, задающий требуемый уровень надежности, который задает пользователь в настройках программы.

Таким образом, в случае метода расстояние до множества требование только одно – разница между выходным сигналом N-го нейрона и выходными сигналами остальных нейронов должна быть не меньше уровня надежности.

Для решения задач классификации оценка вида "расстояние до множества" позволяет повысить скорость обучения.

#### **Оптимизация шага.**

При обучении НС осуществляется коррекция обучаемых параметров нейросети (весов) согласно следующему выражению:

$$x_{ij} = h * \frac{\partial H}{\partial x_{ij}}$$

где h - шаг перемещения по выбранному направлению.

Величина шага оказывает большое влияние на эффективность процесса обучения.

Если шаг h является постоянной величиной, то при больших значениях шага существует вероятность, что в результате очередного перемещения по градиенту сеть проскочит минимум, и дальнейшее обучение пойдет в другом направлении, либо, если впадина вокруг минимума достаточно велика, сеть войдет в режим колебаний вокруг точки минимума. При этом скорость обучения может стать практически нулевой. При малом значении шага скорость обучения также становится малой, и обучение сети останавливается даже в неглубоких локальных минимумах с малой окрестностью. Следовательно, для сохранения приемлемой скорости обучения необходимо проводить оптимизацию (подстройку) шага в ходе обучения.

Достаточно простой и эффективной является двумерная оптимизация шага. При этом предполагают, что изменение функции в выбранном направлении может быть достаточно точно описано параболой. Для того, чтобы построить эту параболу, достаточно знать значение функции оценки в трех точках: исходной (из которой осуществляется очередной шаг в выбранном направлении) и двух вспомогательных. По этим трем точкам строится парабола и определяется ее минимум. Оптимальная величина шага принимается равной расстоянию от текущей точки нахождения сети до минимума параболы.

#### **Методы, повышающие эффективность обучения.**

В случае эллиптических линий уровня методом наискорейшего спуска за один шаг минимум достигается только из точек, расположенных на осях эллипсов. Из любой другой точки спуск будет происходить по ломаной, каждое звено которой ортогонально к соседним звеньям, а длина звеньев убывает. Для достижения минимума потребуется бесконечное число шагов метода градиентного спуска. Этот эффект получил название овражного, а методы оптимизации, позволяющие бороться с этим эффектом – антиовражных.

Метод обучения РarТap является одним из простейших антиовражных методов. Входными параметрами данного метода являются: начальная карта, процедура вычисления направления спуска, локальное обучающее множество и процедура вычисления оценки.

Идея метода состоит в том, чтобы запомнить начальную точку, затем выполнить k шагов оптимизации по методу наискорейшего спуска, затем сделать шаг оптимизации по направлению из начальной точки в конечную.

В ходе реализации проекта в ранее разработанном обучающем комплексе введены соответствующие программные модули: функция оценки вида расстояние до мно-

жества, оптимизация шага, антивращный метод обучения PageRank. Данные методы позволяют студентам в ходе экспериментальных исследований освоить методы обучения нейросетей и сделать выводы о зависимости скорости обучения от заданных параметров.