кривой $F(x, y) = 0$ подстановками $x = f_1(t)$ и $y = f_2(t)$, в силу этого свойства приводится к рациональной форме $\int R\{f_1(t), f_2(t)\} f_1'(t)\, dt$, т.е вычисляется в конечном виде.

Основную теорему об уникурсальных кривых можно формулировать так: "если плоская алгебраическая кривая имеет максимальное число двойных точек, допускаемое её порядком, то она уникурсальна".

Справедлива и обратная теорема: "если кривая - уникурсальна, то она имеет максимальное число двойных точек , допускаемое её порядком".

При этом число двойных точек включается все особые точки, изолированные, точки высшей кратности, пересчитанные на двойные, и точки, недоступные обозрению по внешнему виду кривой, мнимые и двойные точки в бесконечно удаленных круговых точках.

Если кривая имеет r двойных точек, а максимальное число двойных точек, допускаемое её порядком, равно δ, то разность (δ-r), т.е недостающее до максимума число её двойных точке, называется дефектом кривой. Уникурсальная кривая тогда может быть определена как кривая, дефект которой равен нулю.

В заключение хотелось бы отметить, что применение эллиптических интегралов находит все большее применение на практике. Первоначальные приложения эллиптических функций к задаче колебаний маятника распространялись затем на задачи динамики твердого тела, закреплённого в неподвижной точке, нашли себе место в теории гироскопа и других задачах механики. С развитием теории упругости эллиптические функции нашли себе применение в её задачам. Максвелл использовал эллиптические интегралы в своих работах по электричеству и магнетизму. Изучение электрических плоских полей, встречающихся в ряде задач электротехники, например в расчете электромашин, вопросы плоского движения жидкости, ряд задач аэродинамики при их решении нуждаются теперь в теории эллиптических функций. Таким образом, видно, что в настоящее время теория эллиптических функций крепко внедрилась в практику, и поле её приложения все расширяется.

Список использованных источников:

1. Тимофеев А. Ф. Интегрирование функций/ А. Ф. Тимофеев. - СССР, 1948 г., - 179 с.
2. А. М. Журавский Справочник по эллиптическим функциям/ А. М. Журавский. - СССР, 1941 г.

МНОГОСЛОЙНЫЙ ПЕРСЕПТРОН В ПРОГРАММИРОВАНИИ ИГР

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Шарай В.В.

Мардвилко Т.С. – к-т физико-математических наук

В работе исследуется проблема предсказания с использованием многослойных нейронных сетей. Написана программа, визуализирующая процесс обучения сети.

Сегодня компьютеры не только заняли огромную нишу в проведении досуга, но и мы являемся незаменимыми помощниками во многих областях человеческой деятельности. Исследователи все чаще и чаще обращают свой взор в сторону «умных машин» при работе с большими объемами данных, в сферах опасных для жизни человека и др. Неудивительно стремление ученых научить компьютеры принимать решения, “мыслить” и обучаться. Уже сейчас разработано немало алгоритмов позволяющих в той или иной степени решать различные задачи прогнозирования, классификации, кластеризации, распознавания образов и др. В нашей работе исследуется проблема предсказания на основе компьютерной игры. Для исследования проблемы нами написана игра «Ping-pong». В этой игре 1 игрок играет против компьютера. Задача состояла в том, чтобы научить компьютер на основе получаемого опыта предугадывать точку соударения шара с зоной его ворот. Таким образом, при решении данной задачи компьютер во время игры обучается у игрока и чем дольше он играет, тем лучше он начинает играть. Данную задачу мы решаем с использованием нейронных сетей, а именно многослойного персептрона. Многослойный персептрон представляют из себя сеть, состоящую из входного слоя, нескольких скрытых слоев и выходного слоя. Все нейроны связаны между собой синоптическими связями, каждая из которых имеет вес – силу связи. На рисунке 1 изображен многослойный персептрон с одним скрытым слоем и описаны те обозначения, которые мы используем для входных сигналов и весов.

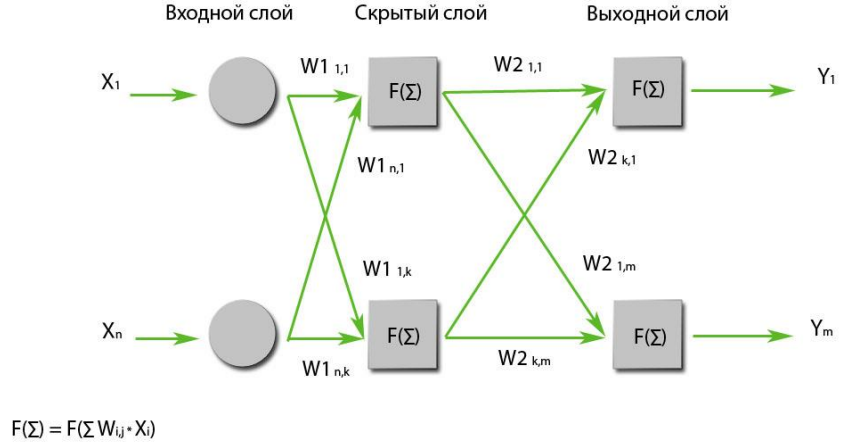

Рисунок 3

На вход сети мы подаем вектор пространства R^3, координаты которого представляют из себя координату точки соударения шара с ракеткой игрока и координаты вектора скорости движения шара. Таким образом, входной слой нашей сети состоит из трех нейронов. Выходной слой сети состоит из одного нейрона, представляющего из себя координату точки соударения шара с воротами компьютера. При построении многослойного персептрона совершенно неясен ответ на вопрос о количестве скрытых слоев и нейронов в них. Мы решаем данную проблему экспериментально исследуя ответы сети и скорость ее обучения. С математической точки зрения каждый нейрон представляет из себя скалярную функцию векторного аргумента. На вход каждый нейрон, кроме нейронов входного слоя, являющегося распределительным, принимает сигналы от всех нейронов предыдущего слоя, умноженные на соответствующие коэффициенты синоптических связей. Выходной сигнал нейрона представляет из себя значение функции активации от взвешенной суммы входных сигналов. В качестве функции активации, демонстрирующей активность нейрона, мы выбрали гиперболический тангенс $\mathrm{F} = \mathrm{th}\Sigma$. Это нелинейная функция логистического типа, обладающая рядом важных для нас свойств, таких как монотонность и дифференцируемость. Таким образом, выходной сигнал нейрона будет вычисляться следующим образом:

$$\mathrm{Y_{i,k}} = \mathrm{th}\left(\sum_{k=0}^{n} \left(\mathrm{Y_{i-1,k}} W_{k,j}^{i} \right) \right)$$

Задача обучения нейронной сети состоит в том, чтобы, предъявив сети некую обучающую выборку, настроить сеть таким образом, чтобы при предъявлении новых входных данных сеть ошибалась как можно реже. В качестве обучающей выборки мы берем реальные данные из игры. Для того, чтобы не происходило перенасыщения, и функция активации способна была дать правильный сигнал все входные данные надо нормализовать. Делаем это таким образом, чтобы координата x точки соударения находились в промежутке [-5, 5], а вектор скорости был длиной 5 единиц. Веса инициализируем рандомными значениями из отрезка [-0.3, 0.3]. Для оценки качества обучения вводится функция ошибки, характеризующая величину ошибки реального отклика от ожидаемого. В качестве функции ошибки мы взяли функцию суммы квадратов расстояний от выходных сигналов сети до их требуемых значений

$$\mathrm{E} = \frac{1}{2} \sum \left(T_j - Y_j \right)^2 ,$$

где T_j и Y_j - это ожидаемый и реальный ответы сети *j*-ого элемента вектора выходных данных обучающей пары. Сумма берется по всем элементам вектора выходных данных обучающей пары.

Функция ошибки представляет из себя функцию многих переменных, и задача состоит в том, чтобы минимизировать эту функцию. Таким образом, настройка весов представляет из себя задачу оптимизации. Решаем эту задачу мы классическим для многослойного персептрона методом - методом обратного распространения ошибки. Данный алгоритм представляет из себя модификацию метода градиентного спуска. Мы будем двигаться в многомерном пространстве весов в сторону, противоположенную градиенту. При этом подстройка весов осуществляется после каждой обучающей пары, т.е. осуществляется так называемый стохастический градиентный спуск. Согласно этому методу, веса изменяются следующим образом

$$\mathrm{W}_{j,k}^{i} = \mathrm{W}_{j,k}^{i} - \eta \nabla E\left(W_{j,k}^{i} \right),$$

где $\nabla E\left(W_{j,k}^{i} \right) = \dfrac{\partial E}{\partial W_{j,k}^{i}}$ - градиент функции ошибки, а η - скорость (шаг) обучения.

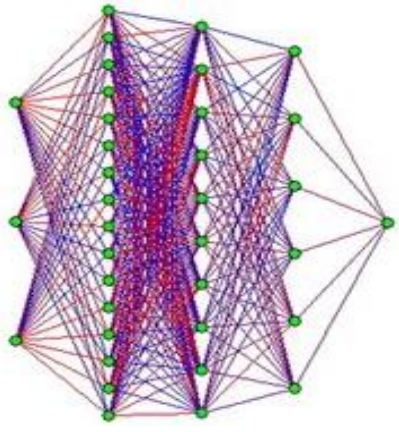

Рисунок 4

Для исследования различных эвристик и выбора оптимального числа слоев, нейронов в каждом слое, данных для обучающей выборки, скорости обучения и других необходимых нам параметров нами написана специальная утилита, визуализирующая процесс обучения и позволяющая изучать ответы сети при разных ее конфигурациях. Данная утилита представляет из себя меню, где исследователь настраивает различные конфигурации сети, такие как количество слоев, нейронов в каждом слое, скорость обучения, загружает данные для обучения, а также настраивает различные параметры игры. Тут же мы можем видеть результаты наших настроек, таблицу с загруженными данными, а также построенную нами нейронную сеть. На рисунке 2 приведен пример как выглядит в нашей программе сеть с 3-мя скрытыми слоями по 16, 10 и 6 нейронов на 1-ом, 2-ом и 3-ем слое соответственно. Как видно из рисунка линии связи имеют различную окраску, соответствующую весовому коэффициенту и получаемую смешением красного и синего цветов. Так связи с положительными весами, так называемые возбуждающие, отображаются оттенками красного цвета, а с отрицательными, тормозящие, - оттенками синего. При этом чем больше коэффициент возбуждающей связи, тем больше красного цвета в отображаемой линии. Для отображения связей с коэффициентом больше единицы, используется только красный. Аналогичная ситуация с отображением тормозящих связей. Каждый раз после обучения сети программа изменяет цвета связей, тем самым визуализируя наблюдателям процесс настройки весов. После обучения сети мы можем тут же нажатием кнопки «Train» отобразить график, демострирующий ошибку обучаения зависящую от количества итераций. Меняя конфигурацию сети мы можем сравнивать скорость и качество обучения, отображая несколько графиков на одной плоскости. Щелкая левой кнопкой мыши по точке графика мы увидим соответсвующее этой точке количество пройденных итераций и ошибку сети. Меню также содержит поля, отображающие средние и максимальные ошибки обучаеющей и тестовой выборкок. А также, выбирая в таблице обучающей выборки отдельную пару, можно просмотреть ошибку для нее.

Список использованных источников:

1. [Электронный ревурс] http://www.machinelearning.ru
2. Марк Лутц. Программирование на Python. Том 2 / Марк Лутц // Учебник по языку Python. - Символ-Плюс, Санкт-Петербург, 2011. – 992 с.