



Рисунок 1 – Пример двухслойного персептрона

значение является аргументом для функции активации. Значения функций активации (выходы нейронов первого слоя) будут являться входами для персептронов второго слоя.

В качестве активационной функции чаще всего используется сигмоидальная функция

$$f_{\sigma}(S) = \frac{1}{1 + \exp(-S)}.$$

При обучении на входы подаются сигналы с известным ответом. Начальные значения весов обычно задаются случайно. В процессе обучения веса связей должны измениться таким образом, чтобы выход сети стал ближе к требуемому.

Таким образом, обучение нейронной сети сводится к минимизации функции ошибки, путем корректировки весовых коэффициентов между нейронами. Чтобы найти ошибки предыдущих слоев, используется принцип переноса, то есть сигнал ошибки распространяется по слоям в обратном направлении с учетом тех же весовых коэффициентов.

УДК 004.65

ТРЕХЭТАПНОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ДИАГРАММ СУЩНОСТЬ – СВЯЗЬ

Черненко Д.В., ст. преп., Соколова А.С., ст. преп.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Проектирование баз данных сложный и кропотливый процесс, трудно поддающийся формализации, по крайней мере, на начальном этапе. Однако относиться к этому этапу разработки информационных систем необходимо чрезвычайно ответственно, так как стоимость исправление ошибок, допущенных на начальных этапах работы над проектом может достигать значительных сумм, зачастую превышающих стоимость информационной системы.

Решением указанной проблемы может являться трехэтапный подход к проектированию базы данных.

Прежде чем начать работу по проектированию базы данных необходимо досконально изучить предметную область. Процесс изучения предметной области может проходить в форме опроса заказчика, причём опрашивать лучше всего не одного представителя, а нескольких связанных по своим профессиональным обязанностям с различными аспектами работы будущей информационной системы.

Выполнив анализ предметной области, можно приступить к первому этапу проектирования базы данных, который называют концептуальным. На этом этапе из

описания предметной области выделяют информационные сущности, их атрибуты и связи между сущностями. В качестве атрибутов используются исключительно информационные атрибуты, то есть те которые несут информационную нагрузку о свойствах сущности.

Для каждого атрибута сущностей необходимо определить их домены. Домен – это выражение, определяющее значения, разрешенные для данного атрибута. На основании информации о домене атрибута на этапе логического проектирования будут выбираться типы полей для хранения данных атрибута экземпляра сущности.

Для наглядного представления полученной модели предметной области используется диаграмма сущность-связь. На данный момент предложено множество вариантов подобных диаграмм, но на этапе концептуального проектирования наиболее удобной можно считать диаграмму, предложенную одним из основоположников реляционной модели базы данных Питером Ченом.

На основании построенной диаграммы делается выбор модели базы данных.

Второй этап проектирования базы данных – логический. На этом этапе осуществляется преобразование концептуальной модели в виде нотации Чена в диаграмму классов UML.

Заключительным является этап физического проектирования. На нем проектирование осуществляется по правилам конкретной СУБД и её диалектом языка SQL. Задачи физического этапа проектирования можно кратко сформулировать так: выбор типов данных и правильное их применение.

Описание процесса трехэтапного проектирования баз данных представленное выше несколько упрощенное. Однако оно показывает как разработать базу данных практически с нуля, при этом явно не используя правила нормализации, а лишь проверяя полученные схемы на соответствие нормальным формам, что заметно упрощает процесс проектирования.

УДК 004.42+004.896

ПРОГРАММИРОВАНИЕ МОБИЛЬНЫХ РОБОТОВ LEGO MINDSTORMS EV3

Добыш Д.С., студ., Черненко Д.В., ст. преп.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Основная задача данной работы – написание программы контроллера робота Lego Mindstorms EV3 для его прохождения по лабиринту, выложенному черной изолентой. Программирование велось на языке C для специальной ОС ev3dev, основанной на Debian Linux.

Построенный робот представляет собой базовую модель ev3 mindstorms robot, но использующую USB-порт для подключения двух периферийных устройств – веб-камеры и WiFi-адаптера.

Поставленная задача была разделена на три этапа: получение правильной картинки лабиринта, вычисление маршрута и его выполнение.

Получение картинки лабиринта выполнялось с использованием API video4linux. После получения картинки возникла проблема искажения перспективы. Камера, установленная на роботе, не имеет функции изменения фокуса. Поэтому проект был разделен на две отдельные программы: одна для ПК, предоставляющая большую производительность