

РАСПОЗНАВАНИЕ ДВИЖЕНИЙ НА КВАТЕРНИОНАХ ВРАЩЕНИЯ ДАТЧИКА ПОВОРОТА ПОСРЕДСТВОМ НЕЙРОННОЙ СЕТИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ерома А. П.

Сурков К. А. – ассистент

Информационные технологии проникли во все аспекты жизнедеятельности современного общества. Одним из следующих шагов является повсеместное внедрение разнообразных датчиков, т.н. сенсоров, с помощью которых программные средства могут распознавать деятельность и состояние человека, при этом способствуя повышению эффективности данной деятельности. В данном докладе рассматривается способ применения сенсоров, а также программного комплекса для распознавания движений человека в различных видах спорта, с возможностью применения в процессе тренировок и соревнований. Проблема, одно из решений которой представлено далее, - это машинное распознавание движений человека.

Датчик движения размещён на руке человека, по протоколу Bluetooth датчик передает данные сенсоров на вычислительное устройство, вычислительное устройство преобразовывает данные и передаёт на вход нейронной сети. Нейронная сеть на выходе даёт вероятности отношения обрабатываемого движения к различным классам движений.

В качестве обрабатываемых данных выбраны кватернионы вращения, т.к. они позволяют в наиболее краткой форме описать поворот объекта в пространстве.

В качестве классификатора выбрана нейронная сеть типа: многослойный перцептрон. Разрабатываемая нейронная сеть состоит из 3-х слоёв: входного, выходного и промежуточного (скрытого). Ценность такой сети заключается в том, что она показывает вероятность отношения распознаваемого образа к определённому классу, что в частных случаях, например в спортивных тренировках, может рассматриваться как некий коэффициент, характеризующий то, насколько правильно было выполнено то или иное движение. В качестве функции активации нейронов используется сигмоидная функция (1). Схема описанной нейронной сети представлена на рисунке 1.

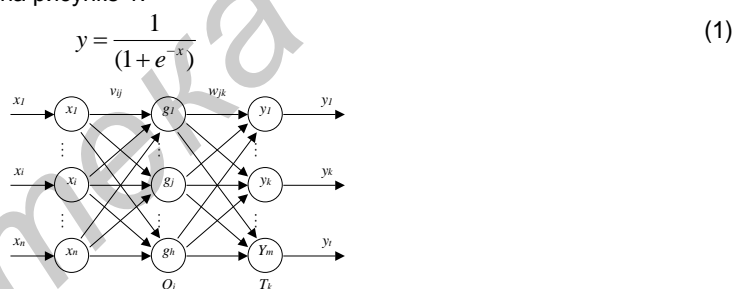


Рис. 1 – Многослойный перцептрон

Наиболее эффективным обучение такой сети будет, если в качестве входных образов использовать биполярные образы, т.е. входные сигналы со значениями [1, -1]. В случае если для обучения сети напрямую использовать кватернионы вращения, то обучение станет крайне затруднительной задачей, т.к. входные образы будут плохо разделены в пространстве признаков.

Было экспериментально установлено, что в общем случае, значение кватернионов полученных с датчика поворота варьируются в диапазоне от -1 до 1. Таким образом, в процессе распознавания и обучения можно использовать не конкретное значение кватерниона, а только его знак, т.к. в общем случае этого достаточно для того чтобы получить представление о совершённом движении. Однако в ряде случаев, было необходимо различать движения, которые характеризовались кватернионами с одинаковыми знаками компонент. Для этого была введена таблица преобразования значения компоненты кватерниона к биполярному образу (рис. 2), при этом значению компоненты кватерниона соответствует двухзначный биполярный образ, где первая компонента образа соответствует знаку компоненты кватерниона, а вторая компонента образа характеризует величину модуля компоненты кватерниона.

Значение компоненты кватерниона	Биполярный образ
$(-\infty..-05]$	-1,1
$(-05..0]$	-1,-1
$(0..0.5]$	1,-1
$(0.5..\infty)$	1,1

Рис. 2 –Таблица преобразования компонент кватерниона в биполярный образ

Таким образом, каждый 4-х компонентный кватернион поворота преобразовывается в 8-ми компонентный биполярный образ. Входной образ состоит из 20 кватернионов, или 160 биполярных

сигналов. Такой подход не является затратным с точки зрения нагрузки на вычислительный процессор и использования аппаратной памяти. Описанный выше подход показал хорошие результаты в процессе обучения, т.е. обучение на поданной на вход сети обучающей выборке происходит быстро, при условии, что образы более-менее компактно разделены в пространстве признаков, а так же высокую точность при распознавании, что позволяет применять данный подход для решения широкого класса задач.

Список использованных источников:

1. Ф. Уоссермен "Нейрокомпьютерная техника - Теория и практика". Перевод на русский язык, Ю. А. Зуев, В. А. Точенов, 1992. 184с.
2. Лабораторный практикум по дисциплине «Цифровая обработка сигналов и изображений» и «Методы и средства обработки изображений» для студ. спец. I-40 02 01 «Вычислительные машины, системы и сети» и I-40 01 01 «Программное обеспечение информационных технологий»: В 2 ч. Ч. 2 /Р.Х. Садыхов, М.М. Лукашевич. – Мн. : БГУИР, 2006. – 32 с.
3. И.Л. Кантор, А.С. Солодовников Гиперкомплексные числа. – М.: Наука, 1973. – 144 с.

РАБОТА ПРЕПРОЦЕССОРОВ НА ОСНОВЕ ZEN CODING И EMMET

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Бардт Д.В., Кузнецов А.А.

Бранцевич П.Ю. – к. т. н., доцент

В нынешнее время каждый человек пытается облегчить себе жизнь. В сфере IT также, каждый программист старается не писать код несколько раз, а использовать каждый раз один и тот же листинг кода при необходимости.

Работа с препроцессорами всегда являлась одной из самых обсуждаемых тем. Они являются одним из самых удобных инструментах, которые очень сильно упрощают процесс разработки кода, ускоряя его, и делают его более доступным к последующей его модернизации.

Сегодня одним из самых интересных моментов являются сокращение кода до минимума. И с этим легко справляется Zen Coding. С его помощью можно легко и быстро написать необходимый код, уместив его практически в две строчки. После нажатия выбранной комбинации клавиш появится ваш результат кода полностью на HTML или CSS. Zen Coding уже как два года в разработке и присутствует различные его версии. Работает он на многих текстовых редакторах.

Начнем с того, что определим, чем Zen Coding не является: не является препроцессором в том плане, в котором мы видим LESS, Sass, Stylus, Jade, Haml и так далее. Хотя здесь и есть сходства. Подобно всем этим инструментам, Zen Coding имеет уникальный синтаксис, направленный на упрощение процесса разработки HTML-кода и CSS.

Тем не менее, в отличие от других инструментов, Zen Coding не добавляет никаких особых свойств типа переменных или миксинов, а также не требует дополнительных файлов, которые нужно пропускать через компилятор. Zen Coding на выходе дает уже готовые коды HTML и CSS и никто, глядя на ваш код «изнутри», не сможет определить, что вы использовали для его разработки. Ознакомление с синтаксисом самого написания является очень простым, если вы знакомы с основами CSS и HTML.

Emmet – не новая технология. Это продолжения Zen Coding, но сделана намного удобней для разработчика. Там работа произведена практически как с обычными селекторами в CSS. Что касается разработки вложенности, то Zen Coding сильно уступает Emmet, так как Emmet производит ссылочную вложенность. Также в Emmet улучшен модуль определения неявных имен, теперь он смотрит в какой тег вложен, и, исходя из этого, делает, определенный вид аббревиатуры.

Список литературы:

1. Official Emmet site [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://emmet.io/>
2. Emmet Documentation [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://docs.emmet.io/>
3. Zen Coding Documentation [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://code.google.com/p/zen-coding/>