ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В КОМПЬЮТЕРНОЙ ГРАФИКЕ

Владислав ГАВРИЩУК

The article focuses on sketch-based 3D-modeling. While drawings are easy for humans to create, they are very challenging for computers to interpret due to their ambiguity. My approach is to use neural networks to solve image recognition problems. Neural network is able to recognize a sketch and return parameters to generate a polygonal model. The result of practical work is an application written in JavaScript that combines a neural network and a visualizer of three-dimensional models.

Сегодня компьютерная графика применяется практически во всех областях человеческой деятельности. Тем не менее, создание трёхмерной компьютерной графики является непростой задачей для большинства пользователей компьютера. Развитие вычислительной техники позволило создавать специализированные программы для работы с графикой, но упор всё же делался на двухмерные изображения, так как человеку легче выполнять рисунки или скетчи в двухмерной плоскости. Что если создать программу, которая будет способна сама генерировать трёхмерные модели на основе скетча, сделанного от руки? Есть лишь одна проблема: компьютеру, с его вычислительной мощностью в миллионы операций в секунду, трудно даются задачи распознавания образов. Распознавание образов – это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы несущественных данных. Если говорить о сегодняшних достижениях, то для решения этой задачи лучше всего подойдут нейронные сети.

Искусственная нейронная сеть – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма. Искусственная нейронная сеть представляет собой систему соединённых и взаимодействующих между собой простых процессоров – **искусственных нейронов**. Будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, они

способны выполнять довольно сложные задачи, например – распознавание образов. Люди легко распознают образы и это не вызывает у них никаких вопросов, потому что люди делают это на бессознательном уровне. При этом если лишь задуматься о том, как написать алгоритм для распознавания образов, то сразу возникают трудности – какого-то определённого алгоритма просто нет. И тут на помощь приходят нейронные сети. Нейронные сети применяют для решения задач, алгоритм решения которых неизвестен. Каждый нейрон имеет дело только с сигналами, которые он преобразует по определённому алгоритму, одинаковому для всех нейронов. Другими словами, внутри любого нейрона сигнал обрабатывается всегда одинаково, но сигнал на выходе нейронной сети всегда получается индивидуальным. Дело в том, что все искусственные нейроны соединены между собой синапсами. **Синапс** – место стыка выхода одного нейрона и входа другого, в котором происходит усиление/ослабление сигнала. Значение синапса между двумя нейронами, которое также называется его весом, может изменятся, и это влияет на сигнал. У каждого нейрона может быть от одного до нескольких входов, через которые он принимает входной сигнал. Проходя по синапсу, входной сигнал «умножается» на вес этого синапса, тем самым усиливая или ослабляя его [1].

Нейронные сети применяются для классификации объектов, кластерного анализа, прогнозирования, аппроксимации, сжатия данных, оптимизации, но одним из наиболее распространённых применений нейронных сетей является распознавание образов. В распознавании образов преобладают свёрточные нейронные сети. **Свёрточные нейронные сети** пользуются тем, что вводные данные состоят из изображений, и они ограничивают построение сети более разумным путем. Работа свёрточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям и далее к ещё более абстрактным деталям, вплоть до выделения понятий высокого уровня. Хотя и у свёрточных нейронных сетей есть недостатки. Внутреннее представление данных сверточной нейронной сети не учитывает пространственные иерархии между

простыми и сложными объектами. Так, если на изображении в случайном порядке изображены глаза, нос и губы, то для свёрточной нейронной сети это явный признак наличия лица, а поворот объекта ухудшает качество распознавания.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными. Таким образом, используя нейронные сети для распознавания наброска, сделанного от руки, хорошо обученная нейронная сеть может определить то, что хотел изобразить человек, и дополнить изображение недостающими деталями или даже преобразовать набросок в данные, на основе которых можно будет сгенерировать трёхмерную модель [2].

При полигональном моделировании трёхмерная модель представляет собой набор вершин, рёбер, граней и полигонов. Вершины соединяются друг с другом рёбрами. Грань – это замкнутое множество рёбер, в котором треугольная грань имеет три ребра, а четырёхугольная – четыре. Полигон – это набор компланарных граней. Вместе они образуют полигональную сетку, которая описывает трёхмерную модель. Полигональную сетку трёхмерной модели можно создать процедурными методами [3].

Процедурная генерация контента – процесс автоматического создания контента посредством алгоритмов, а не вручную. Процедурная генерация получила самое яркое проявление в компьютерной графике, поэтому её часто применяют в компьютерных играх, при визуализации архитектурных и дизайнерских проектов, в фильмах и т.п. Так как при процедурной генерации трёхмерная модель описывается последовательностью инструкций, то необходимо на выходе искусственной нейронной сети, после обработки наброска, получить параметры для генерируемой трёхмерной модели.

Для практической реализации вышеописанной задачи был выбран язык программирования JavaScript. Так как искусственные

нейронные сети – это алгоритмы, которые могут быть реализованы на любом языке программирования, то основной задачей было найти язык программирования, который был бы способен совместить в себе реализацию алгоритмов искусственной нейронной сети и возможность визуализировать трёхмерные модели. Благодаря библиотеке Three.js в JavaScript, которая использует методы WebGL, можно визуализировать трёхмерную графику в любом современном браузере, в котором есть интерпретатор JavaScirpt.

В результате выполнения практической части работы было разработано приложение, которое применяет искусственные нейронные сети для облегчения процесса создания трёхмерной компьютерной графики. Искусственная нейронная сеть была обучена распознавать примитивные формы (куб, сферу), нарисованные от руки. После распознавания, приложение процедурными способами генерирует трёхмерные модели на основе параметров, полученных на выходе искусственной нейронной сети после обработки рисунка. Благодаря интерактивному подходу, пользователю не нужно указывать процедурные правила или параметры для генерации, вместо этого они автоматически распознаются по эскизу, что позволяет неподготовленным пользователям быстро создавать сложные процедурные модели.

Литература:

1. РАШИД, Т. *Создаём нейронную сеть*, 2016. 272 с.
2. ГУДФЕЛОУ, Я. *Глубокое обучение*/Я. Гудфелоу, Ё. Бенжио, А. Коурвиль, 2016. 777 с.
3. JAMES, M. Van Verth. *Essential Mathematics for Games and Interactive Applications*/James M. Van Verth, Lars M. Bishop. New York: CRC Press, 2016. 564 с.

Рекомендовано
Вадим СИРКЕЛИ, др., конф. унив.