

НЕТРИВИАЛЬНЫЕ ВОЗМОЖНОСТИ НЕЙРОННЫХ СЕТЕЙ

Учреждение образования «Белорусская государственная академия связи», г. Минск, Республика Беларусь

Нейронные сети – одно из направлений в разработке систем искусственного интеллекта. Идея заключается в том, чтобы максимально близко смоделировать работу человеческой нервной системы – а именно, ее способности к обучению и исправлению ошибок. В этом состоит главная особенность любой нейронной сети – она способна самостоятельно обучаться и действовать на основании предыдущего опыта, с каждым разом делая все меньше ошибок. В этой статье рассматривается несколько примеров, выделяющихся на фоне других.

Pix2code. В 2017 году команда разработчиков, называющая себя FloydHub, опубликовала на GitHub алгоритм pix2code, которая генерирует готовый код на основе изображения пользовательского интерфейса [1]. Спустя год на основе работы алгоритма была создана одноименная нейросеть для верстки страниц, которую может запустить любой желающий.

Работа алгоритма делится на три этапа:

1. Нейронной сети передается картинка-исходник в формате JPEG.
2. Алгоритм конвертирует элементы в HTML- и CSS-код.
3. На выходе получается сверстанная версия.

Чтобы получился такой результат, разработчики «скармливали» алгоритму скриншоты и присваивали определенные HTML-теги, в итоге получился набор данных, с помощью которого можно генерировать шаблонные сайты.

Text2video. Разработчики из Америки создали нейронную сеть [2], генерирующую видеоряд по короткому поступающему на вход сообщению, например, «игра в гольф на траве». Длина видео – 32 кадра (около 1 секунды), разрешение – 64×64 пикселя. Генерация происходит в два этапа. На первой стадии нейросеть улавливает суть текста и переводит его в мутное изображение с пятном в том месте, где должно происходить основное действие. Следующий шаг – генерация самого действия.

Fur Real. Исследователи из Калифорнийского университета создали нейросеть, предназначенную для реалистичной отрисовки шерсти животных [3]. Новый алгоритм обучился учитывать прозрачность меха, что сделало конечный результат более качественным за счет реалистичных бликов и света. Методы рендеринга, которыми пользуются на сегодняшний день, обрабатывают волосы людей и животных по единому алгоритму, что приводит к нереалистичному виду персонажей-животных в играх и кино. Старый алгоритм не учитывает, что шерсть, в отличие от волос, отражает свет под другим углом из-за более объемной медуллы – центральной части волосяного стержня. Для рендеринга шерсти лучше подходит метод подповерхностного рассеивания. Он просчитывает, под каким углом должен отразиться свет, проникший в полупрозрачные тела. Алгоритм работает в 10 раз быстрее, чем используемые в студиях методы.

Исследовательская часть.

Попробуем создать нейронную сеть с помощью программного кода в Matlab. Прежде всего необходимо выбрать тип нейронной сети. Мы будем использовать тип «Нелинейная вход-выход», не использующий обратные связи (рис.1) [4].

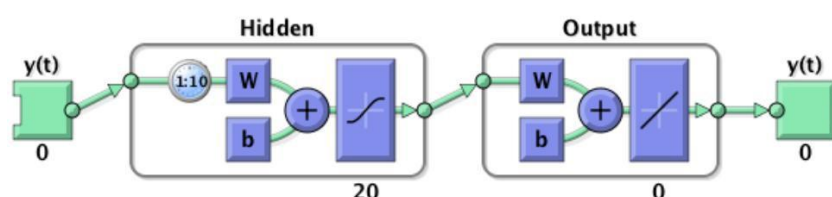


Рисунок 1 – Модель сети «Нелинейная вход-выход»

Для генерации используется следующая функция:

`narnet(feedbackDelays,hiddenSizes,trainFcn)`, где:

- `feedbackDelays` – вектор обратных задержек (По умолчанию = 1:2)
- `hiddenSizes` – Количество скрытых нейронов (По умолчанию = 10)
- `trainFcn` – Метод обучения (По умолчанию = 'trainlm')

Выбор данных для обучения, и их подготовка. Для подготовки данных воспользуемся функцией `PREPARETS`, которая подготавливает временные ряды для части сети, изменяя время до минимума, чтобы “чувствовать” входные данные и внутренние переменные.

Синтаксис команды:

`[Xs,Xi,Ai,Ts,EWs,shift] = preparets(net,Xnf,Tnf,Tf,EW)`, где:

- `Net` – Нейронная сеть
- `Xnf` – Не возвращаемые входные данные
- `Tnf` – Не возвращаемые выходные данные
- `Tf` – Выходные данные
- `EW` – Вес ошибок (По умолчанию = {1})
- `Xs` – Перемещенные входы
- `Xi` – Изначальные состояния входных задержек
- `Ai` – Изначальные состояния скрытых нейронов
- `Ts` – Перемещенные выходы
- `EWs` – Перемещенные веса ошибок

Также, перед применением `preparets` воспользуемся функцией `tonndata`, которая конвертирует данные в стандартный тип для нейронных сетей.

Синтаксис команды:

`[y, wasMatrix] = tonndata(x, columnSamples, cellTime)`, где:

- `x` — Матрица или ячейка – массив матриц
- `columnSamples` `True` если исходные данные представлены в виде колонок, `false` если в виде строк
- `cellTime` – `True` если исходные данные представлены колонками ячеек матрицы, `false` если в виде матрицы
- `y` – Исходные данные, конвертированные в стандартный вид
- `wasMatrix` – `True` если исходные данные представлены в виде матриц

Выбор назначения данных. В этой части рассматривается выбор процентного соотношения данных: Тренировка, Проверка, Тестирование. Для этого воспользуемся следующими командами:

```
net.divideParam.trainRatio = 85/100; %Тренировка
net.divideParam.valRatio = 15/100; %Проверка
net.divideParam.testRatio = 5/100; %Тестирование
```

Обучение нейросети

Операция обучения кодом выглядит так:

```
[net,tr] = train(net,inputs,targets,inputStates,layerStates);
```

При помощи следующих команд можно получить различные графики, характеризующие процесс и качество обучения сети:

```
outputs = net(inputs,inputStates,layerStates)
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)
```

Получение нейронной сети. Для работы с обученной сетью необходимо выполнить экспорт полученной нейросети. Для получения Simulink модели после всех операций над нейронной сетью с использованием кода, необходимо вызвать функцию:

```
gensim(net,time)
```

где `time` – время дискретизации, а `net` – нейронная сеть.

Вывод. Разработчики по всему миру используют нейросети в своих проектах и нет никаких предпосылок к прекращению этой тенденции, поэтому можно с уверенностью сказать, что нейросети и искусственный интеллект, в дальнейшем станут еще популярнее и появятся на свет куда более интересные проекты.

ЛИТЕРАТУРА

1. Turning Design Mockups into Code with Deep Learning // floydhub.com [Электронный ресурс]. URL: <https://blog.floydhub.com/Turning-design-mockups-into-code-with-deep-learning/> (Дата обращения: 09.03.2018)
2. New algorithm can create movies from just a few snippets of text // sciencemag.org [Электронный ресурс]. URL: <https://www.sciencemag.org/news/2018/02/new-algorithm-can-create-movies-just-few-snippets-text> (Дата обращения: 09.03.2018)
3. Fur Real - Scientists Improve Computer Rendering of Animal Fur// jacobsschool.ucsd.edu [Электронный ресурс]. URL: http://jacobsschool.ucsd.edu/news/news_releases/release.sfe?id=2480 (Дата обращения: 09.03.2018)
4. Нейронные сети в MatLab // digiratory.ru [Электронный ресурс]. URL: <https://digiratory.ru/508> (Дата обращения: 09.03.2018).