

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Уральский федеральный университет
имени первого Президента России Б.Н.Ельцина»

Институт математики и компьютерный наук
Кафедра алгебры и дискретной математики

Применение нейронных сетей для калибровки оборудования на примере двух задач робототехники.

Допустить к защите:

« _____ » _____ 2015 г.

Выпускная квалификационная
работа на степень бакалавра
по направлению
Математика и компьютерные науки
студента группы МК-410502
Штех Геннадия Петровича
Научный руководитель
заведующий лабораторией
доктор КАКИХ?! наук
Окуловский
Юрий Сергеевич

Екатеринбург
2015

Оглавление

1	О чем диплом?	2
2	Подробные сведения об используемой нейронной сети	4
3	Описание прямого решения задачи с применением нейронной сети	5
3.1	Модель задачи	5
3.1.1	Модель	5
3.1.2	Описание сценария для сбора экспериментальных данных	6
3.1.3	Результат подхода	7
3.1.4	Попытки модификации подхода к формированию экспериментальной выборки	7
3.1.5	Вывод	8

Глава 1

О чем диплом?

Не редко при конструировании подвижных платформ применяются двигатели, управляемые постоянным током. Характерной особенностью этих двигателей является их дешевизна и долговечность. Скорость вращения вала двигателя постоянного тока обычно регистрируется с помощью дополнительных устройств. В рассмотренном случае это были энкодеры. Не будем вдаваться в принцип их работы, стоит отметить лишь тот факт, что для текущей задачи их точности хватает с запасом. Главная сложность при их использовании – это сложность управления непосредственно скоростью вращения вала. Потому что фактически управлять можно только напряжением на контактах двигателя, что хорошо связано только с моментом силы на валу. Однако при использовании двигателей для перемещения необходимо хорошо контролировать именно скорость. Поэтому остро стоит проблема определения подходящего сигнала для достижения необходимой скорости вращения.

Итак задачу можно формализовать следующим образом: нужно построить функцию, по желаемой скорости и по, возможно, необходимым дополнительным параметрам, вычисляющую напряжение на контактах двигателя, приводящее скорость к желаемой. Для простоты будем называть напряжение

на контактах двигателей сигналом.

Поскольку есть возможность собрать экспериментальные данные об искомой зависимости, подобную задачу восстановления неизвестной функции возможно решить методами регрессии. К качестве используемого метода выберем нейронную сеть. А именно – простую многослойную нейронную сеть, обучающуюся с учителем, биполярная сигмоида в качестве функции активации.

Глава 2

Подробные сведения об используемой нейронной сети

Глава 3

Описание прямого решения задачи с применением нейронной сети

3.1 Модель задачи

3.1.1 Модель

За входной вектор возьмем желаемую скорость(Spd), за выходной вектор возьмём сигнал(Sig), приводящий к этой скорости.

$$Sig \in SIG; Spd \in SPD$$

$$SIG = \{Sig \in \mathbb{N} | -256 < Sig < 256\}; SPD = \mathbb{R}$$

3.1.2 Описание сценария для сбора экспериментальных данных

Соберём экспериментальные данные по следующему сценарию: будем произвольным образом менять сигнал (Sig_l) (l – значит last, прошлый сигнал), ждать установления скорости вращения (Spd_c) (c – значит current, текущая скорость), записывать пару (Sig_l, Spd_c) в экспериментальную выборку и вновь менять сигнал.

Фактически мы получили выборку отображения из пространства сигналов в пространство скоростей.

$$Sig \in \text{SIG}; Spd \in \text{SPD}$$

$$F(Sig_l) = Spd_c$$

Но нам нужна обратная функция, по желаемой скорости (Spd_n) (n – next, следующая) возвращающая необходимый текущий сигнал (Sig_c).

$$Sig \in \text{SIG}; Spd \in \text{SPD}$$

$$F^{-1}(Spd_n) = Sig_c$$

Чтобы регрессировать обратную функцию, надо лишь перевернуть пару так, как там нужно.

$$(Sig_l, Spd_c) \rightarrow (Spd_c, Sig_l)$$

В том числе, конечно, необходимо проверить наличие обратной данной функции: исходная функция должна быть монотонной. Монотонность функции очевидна из графика, на котором отображены экспериментальные данные в

виде точек.

КАРТИНКА С ДАННЫМИ Особенность в районе нуля обусловлена наличием в системе трения и чтобы "сорвать" систему из состояния покоя требуется достаточно высокий момент вращения, который требует достаточно высокого сигнала на двигателях.

3.1.3 Результат подхода

КАРТИНКА РЕЗУЛЬТИРУЮЩЕЙ ЛИНИИ На практике такой подход показывает себя плохо: поскольку сценарий обучения не соответствует сценарию использования. При управлении платформой скорость двигателей меняется достаточно часто (до двадцати раз в секунду) в достаточно широком диапазоне. Но функция регрессии спроектирована так, что желаемая скорость достигается через некоторое неопределённое время после выставления сигнала. Экспериментально установлено, что в среднем это несколько секунд. При динамическом управлении платформой это неприемлемый результат. Особенно заметна эта особенность при торможении.

КАРТИНКА ЛИНИИ ТОРМОЖЕНИЯ Более того, видно, что особенность вокруг нуля нейронная сеть не выучила. Эту особенность удалось выучить только разбиением пространства на 2: $Sig > 0$ и $Sig \leq 0$ с независимым обучением двух сетей на этих пространствах.

3.1.4 Попытки модификации подхода к формированию экспериментальной выборки

Для приведения в соответствие сценария использования и сценария обучения была сделана модификация сценария сбора данных. Изменение: не ждать установления скорости вала и добавлять в выборку все пары (Sig_l, Spd_c) . Этот

путь тоже не приводит к нужному результату, поскольку выборка становится зашумлённой и обилие информации нейронная сеть воспринимает как шум. В результате использование этой функции регрессии для управления двигателем ни в каких сценариях не даёт приемлимого результата.

3.1.5 Вывод

Прямой подход не даёт пригодного к использованию результата. Главным образом из-за того, что сеть не имеет достаточной информации для выдачи сигнала, подходящего к текущей ситуации. Например ситуации ускорения, торможения и равномерного движения требуют различного подхода к формированию сигнала: завышение, занижения и непосредственное соответствие. В том числе у сети нет временных рамок достижения скорости по сигналу, что даёт непредсказуемое время приведения системы в необходимое состояние. Таким образом мы получили список недостатков:

- Отсутствие временных рамок
- Отсутствие контекста применения сигнала
 - Режим езды: ускорение, торможение, равномерное движение
 - Предыдущий выставленный сигнал индуцирует в катушке двигателя ток, который тоже нужно преодолеть
- Сложная особенность в районе нуля

Ну мы взяли нейронную сеть и попробовали решить проблемы калибровки как умеем: записали как-то входной вектор(желаемая скорость), взяли как-то выходной(какой сигнал к этой скорости приводит), обучили сеть и обосрались: то, как мы калибровали и то, как мы использовали – это разные сценарии.

Основная проблема оказалась вот в чём: хуйню засунул – хуйня выпала, мы собирали данные по следующему сценарию: выставляли сигнал, ждали, пока скорость установится, и в экспериментальную выборку добавляли пару (текущий сигнал, текущая установившаяся скорость). Понятно, что в динамике это показывало ужасающий результат. Наиболее заметен он при торможении, кривая скорости падала недостаточно быстро и робот очень сильно переезжал.

Понятно, кажется, для начала нужно было собрать данные, соответствующие хотя бы сценарию использования. Данные стали собирать так: выставляли сигнал и все скорости, которые регистрировали, записывали в выборку обучения. К сожалению, это ни к чему не привело. Это очевидно, если взглянуть на получившуюся картинку. Множество различных данных для сети оказалось лишь шумом и улучшения не дало. (КАРТИНКА!)

В целом жизнь устроена так: чтобы управлять штуками, нужно постоянно контролировать, как они отреагировали на прошлое воздействие. Иначе штуки будут делать хуеу. Фактически, за штуками нужно следить постоянно, с какой-то периодичностью. Поскольку жизнь вокруг штук тоже может поменяться и прежде годное управляющее воздействие прямо сейчас мчит вас в пропасть. Весь этот процесс называется циклом регуляции. Фактически, устроен он так: мы получаем обратную связь от штуки, вспоминаем, в каком состоянии нам необходимо её поддерживать и на основе обратной связи и наших грязных желаний вырабатываем управление. Возможно, даже учитываем наши предыдущие управляющие воздействия.

В голову пришла мысль о том, что чтобы управлять динамической системой недостаточно просто построить регрессионную модель её отзывает. Оказалось, что наш двигатель – это классическая динамическая система, требующая построения регулятора для работы. Тут мы малясь загрузили, ибо

ПИД-регулятор если брать обобщенный, то там объебаться можно с коэффициентами, а если по уму его строить, то е нас такого ума нет.

Ее мы взяли нейронную сеть и попробовали решить проблемы как умеем: записали входной вектор(прошлый сигнал, прошлая скорость, текущий сигнал), взяли выходной(текущая скорость), собрали данных по сценарию(сценарий сцукорегулятора). Обучили сеть, а она сука поехала и стали мы счастливые очень. Фактически нейронная сеть стала инверсным нейрорегулятором, а не просто регрессионной калибровкой.