Российский Новый Университет

Курсовая Работа

По дисциплине:

“Программирование”

Тема:

“Нейронные сети”

Специальность:

“Информатика и вычислительная техника”

2023

Оглавление:

1. ВВЕДЕНИЕ
   1. Цель курсовой работы
   2. Что такое нейросеть и нейрон
   3. Описание проекта
   4. Последующее применение проекта
2. Зачем нужны нейросети
   1. Сферы применения
   2. Типы нейросетей
   3. Устройство нейросетей
   4. Что такое нейрон как математическая модель
3. История нейросетей
   1. Откуда пошло понятие
4. Разбор кода простейшего нейрона
5. Результаты тестов

Введение.

В данной курсовой работе, будут рассмотрены виды нейросетей и принципы их работы. А так же будет представлена реализация простейшего нейрона с подбором веса и функциями для его обучения и калибровки функции активации. Цель работы показать зачем и как возможно заменить функции кода нейросетью.

Что такое нейросеть и нейрон

Нейросеть – крайне сложная математическая модель, которая состоит из нейронов, связанных друг с другом синапсами.

Нейрон – основной блок для создания нейросети его задача вычислить взвешенную сумму значений его выходных данных, нормализировать значение функцией активации и подать на выход понятное другим нейронами значение.

Сигмойда – отображает выходные данные на значения от 0 до 1

Нейросети состоят из 3 и более слоёв:

1 входной слой

1 выходной слой

N скрытых слоёв

Нейросеть с одним нейроном умеет предугадывать значения легких уравнений из линейной алгебры при заданных 2 входных значениях x и z:

x + y = z

x – y = z

x // y = z

x \* y = z

[алгоритм вычисления весов]

вычисление веса (значение, ответ)

{

Результат = значение \* веса

Ошибка = значение – результат

Отклонение = (ошибка / результат) \* допустимая погрешность

Веса += отклонение

}

[алгоритм вычислений результата]

Генерация результата (значение)

{

Вывод “значение \* вес;”

}

Данный нейрон можно будет применять для предугадывания ответов пользователя на основе его предыдущих ответов или для предугадывания выдачи определённых значений. Но для более сложных математических операций одного нейрона не хватит и потребуется обработать большее количество вводных данных.

Зачем нужны нейросети?

Нейросети имеют широкий спектр применения в различных областях, таких как:

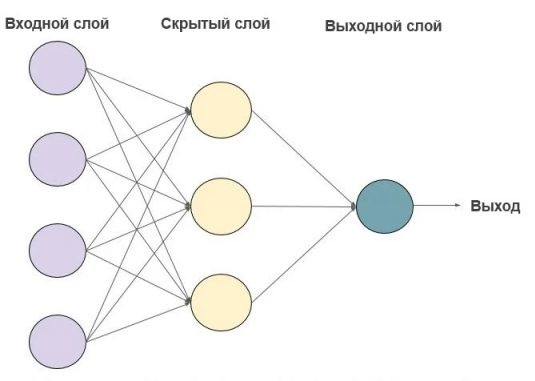
* Распознавание образов: классификация изображений, распознавание лиц, оптическая символика и т.д.
* Обработка естественного языка: машинный перевод, распознавание речи, генерация текста и т.д.
* Прогнозирование и анализ данных: прогнозирование временных рядов, анализ данных, обнаружение аномалий и т.д.
* Управление и оптимизация: автоматическое управление, оптимизация процессов и т.д.
* Медицина: диагностика заболеваний, анализ медицинских изображений, прогнозирование результатов лечения и т.д.

Типы нейросетей и их различия

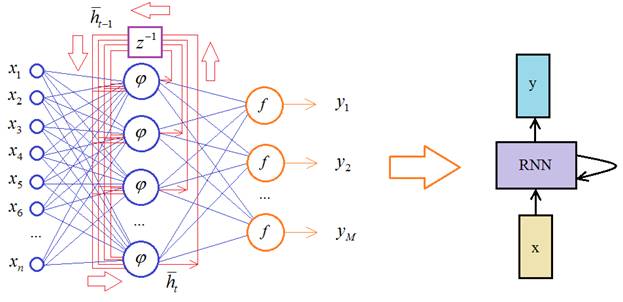
Существует несколько типов нейросетей:

1. Полносвязные [FCN]
2. Рекурентные [RNN]
3. Свёрточные [CNN]
4. Сети долгой краткосрочной памяти [LSMN]
5. Генеративно-состязательные [GAN]
6. Автоэнкодеры [Autoencoders]

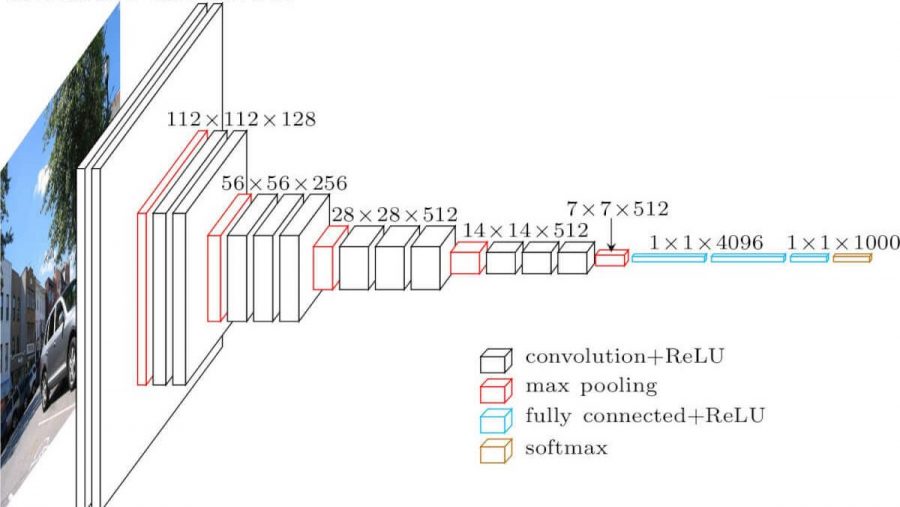
Давайте рассмотрим все виды по порядку.



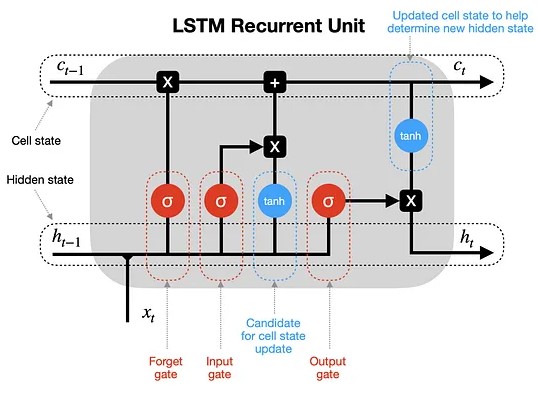
1. Полносвязные – самый простой, он действует строго от входного слоя до выходного, проходя через скрытые слои их применение возможно в таких задачах как классификация объектов или регрессии.



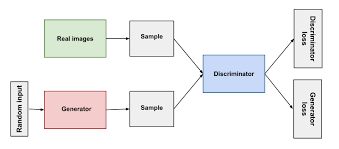
1. Рекуррентные – более сложный алгоритм и в отличие от FNN имеет обратные связи, что даёт им возможность обрабатывать последовательные данные или задействовать временные ряды или естественный язык



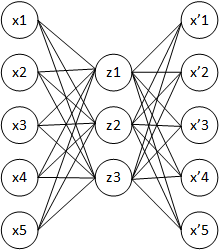
1. Свёрточные - они особенно хорошо подходят для обработки данных с пространственной структурой, таких как изображения. Они используют операцию свертки для извлечения важных признаков из входных данных.



1. Сети долгой краткосрочной памяти - специальный тип рекуррентных нейронных сетей, способных лучше сохранять и использовать информацию о длительных зависимостях в последовательных данных.



1. Генеративно-состязательные сети - особый тип нейронных сетей, состоящий из двух моделей - генератора и дискриминатора. Они применяются для генерации новых данных, таких как изображения, звуки или тексты.



1. Автоэнкодеры - они используются для изучения компактного представления входных данных. Автоэнкодеры состоят из кодировщика, который сжимает данные в скрытое представление, и декодера, который восстанавливает данные из скрытого представления.

Устройство нейросетей.

1. Полносвязная нейронная сеть - допустим у к нашей нейросети подключены 2 датчика выдающих только значения от 0 до 1, соответственно функция активации:

F(x) = {0, X<=0,5 }, {1, X<0.5}

простым языком 1 – да, 0 – нет.

Зададим конкретную задачку нашей нейросети: “Найти все круги желтого цвета”

Датчики назначаем на их задачи:

1. Датчик 1 – цвет.
2. Датчик 2 – форма.

У нас появляется 4 возможных комбинации (в скобках указан пример обектов): (красный круг) - [0,1]

(желтый треугольник) - [1,0]

(розовый треугольник) - [0,0]

(желтый круг) - [1,1]

Зададим равный вес синапсов по 0,5:

(красный круг) - x=0,5\*0+0,5\*1=0

(желтый треугольник) - x=0,5\*1+0,5\*0=0

(розовый треугольник) - x=0,5\*0+0,5\*0=0

(желтый круг) - x=-0,5\*1+0,5\*1=1

Соответственно желтый круг выдал положительный результат.

1. Рекуррентная нейронная сеть - тип нейросети по большей части использущейся для работы и генерации текста и изображений.

Например, нашумевшая “ChatGPT”, все версии данной нейросети являются рекуррентными, так как оперируют огромной базой/моделью/словарём слов и предложений.

Основное отличие RNN от FCN в том, что RNN использует информацию последовательно, т.е. опирается на предыдущие ответы и запросы.

Такой метод генерации можно назвать “памятью”, и это даёт возможность поддержания здравого смысла при генерации текстов.



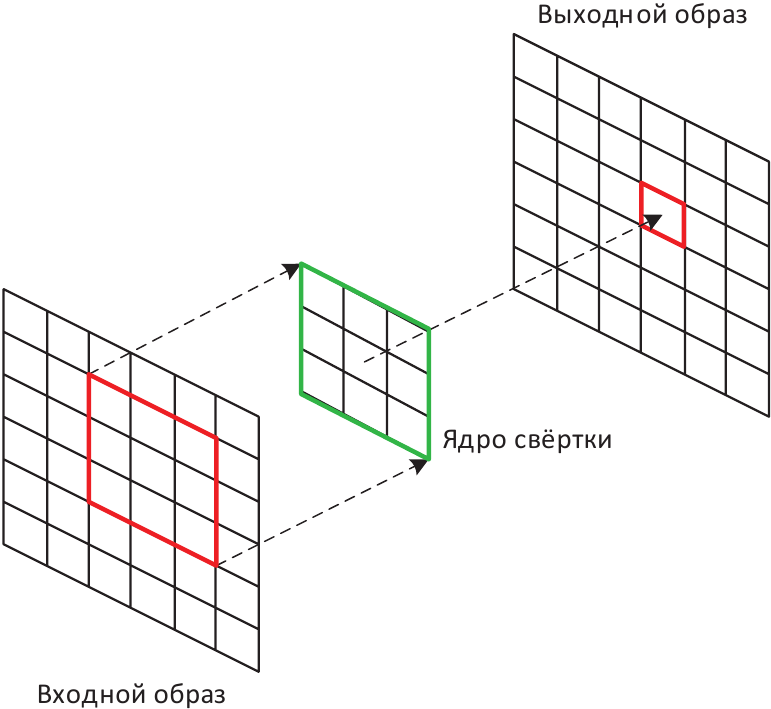
На схеме показано как RNN разворачивается в полную сеть.

x\_t - вход на временном шаге t, например вектор с одним состоянием.

s\_t - это скрытое состояние на шаге t или память.

o\_t - выход на шаге t. Например вектор предсказанных вероятностей.

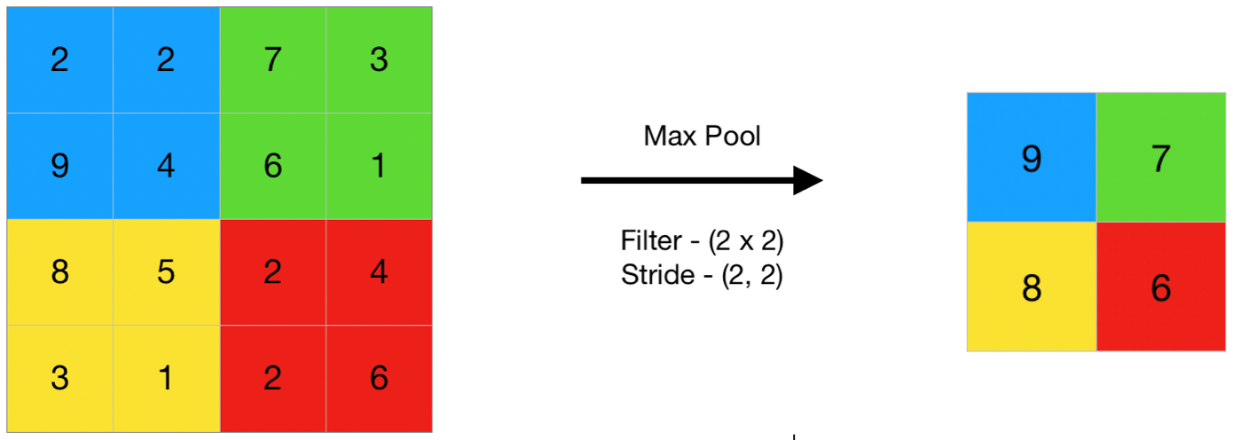
Обучение RNN так-же как и у FCN происходит через алгоритм обратного распространения ошибки, но с поправкой на то, что изменения зависят не только от текущего шага, но и от предыдущих.

1. Свёрточная нейронная сеть – алгоритм, принимающий изображения и присваивая веса аспектам картинки может определять объекты или различать изображения. Для данного типа нейросетей лучше всего подходит плотный поток взаимосвязанных данных, таких как пиксели. В изображениях есть визуальные данный такие как цвет и яркость, соседние пиксели содержат похожие данные с минимальными отклонениями. Основные сферы использования данных нейросетей это классификация и распознавание объектов.

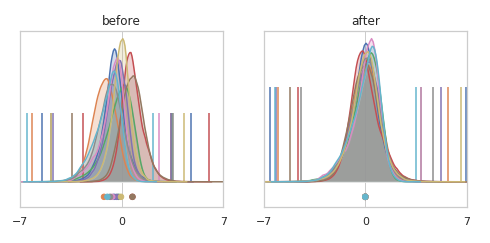
Данный тип нейросетей работает с Датасетами (Datasets) в которых содержится информация о классификации объектов. Сравнивая входной образ с образами из датасета, нейросеть может последовательно определить несколько схожих частей тела/объекта и выдать правильный ответ.

Архитектура свёрточной нейросети состоит из 4 пунктов:

1. Свёрточный слой (convolutional layer) - основной блок свёрточной нейронной сети. Он отвечает за удаление лишних деталей и оптимизацию данных для последующего анализа. Свёртку возможно применять для каждого признака, нейросеть будет подбирать недостающие данные сама, сравнивая из с даннами из датасета.
2. Пулинг (pooling) - нелинейное уплотнение карты признаков, которое следует после свёрточного слоя. Из оптимизированных данных данный слой выбирает самые важные и удаляет несущественные признаки. К результату пулинга можно применить свёрточный слой и получить более точные данные. Это требуется для выстраивания иерархии признаков. Начиная от контура и мелких деталей, заканчивая формой глаз, ушей, лица\морды и ушей.



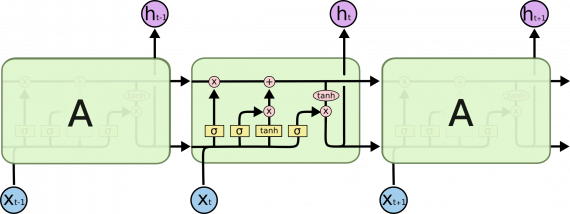
1. Нормализация по батчу (bath-normalization) - Пакетная нормализация, она выполняется путём масштабирования данных, подаваемых в функцию активации. Обычно это принято делать в скрытых слоях нейронной сети. Данный метод помогает ускорить обучение.



1. Полносвязный слой – выходные нейроны связанные со всеми нейронами нейросети.

У свёрточной нейросети есть свои камни преткновения. Ведь она не может работать с табличными данными и им крайне сложно работать с данными разного масштаба такими, как картинки разного разрешения и документы разного формата. Если задать цель сравнить документы А5 и А3 то надо свести их в один формат А4 иначе нейросеть не поймёт что это одинаковый текст.

1. Сети долгой краткосрочной памяти (LSTM) - Особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям. Данный тип нейросети разработан специально для исключения проблем долгосрочной зависимости. Такие нейросети изначально запоминают информацию на долгие промежутки времени, не обучаясь этому.



Структура LSTM очень похожа на RNN, но модули немного отличаются. Вместо одного слоя, они содержат четыре слоя, и эти слои взаимодействуют особым способом.

Основа LSTM – cell state – ячейка, проходящая через всю цепочку и участвующая лишь в нескольких линейных преобразованиях. LSTM имеет возможность удаления информации из ячейки, данный процесс регулируется фильтрами.

Шаги работы алгоритма LSTM:

1. Лишняя информация фильтруется и удаляется из дальнейшей обработки. Фильтрация осуществляется сигмовидным слоем (гейт утраты).
2. Решаем какую информацию надо сохранить в состоянии ячейки.
3. Гейт входа решает какие значения требуется обновить
4. Слой tanh создаёт вектор новых значений, которые добавляются в состояние.
5. Объединяем значения со второго щага.
6. Обновляем предыдущее состояние ячейки для получения нового состояния (C). Затем добавляем новые значений кандидатов (i\*C).
7. Пропускаем состояние ячейки через tanh и умножаем на его на входной сигнал сигмовидного гейта.

Данная схема является традиционной.

1. Генеративно-состязательные сети (GAN) – данный тип нейросетей является для меня самым интересным экземпляром. В ней задействована архитектура включающая генератор и дискриминатор которые работают друг против друга, поэтому GAN и называется генеративно-состязательной.

GAN нейросеть одна из самых потенциальных, она может работать с