UNNAMED THEORY

Нам не важно какое количество параметров будет в определенном ноде.

Ведь у каждого Нода лишь один параметр, и это параметр это заряд.

А вот у связи нода уже другое предназначение, их может быть несколько, - Расстояние (Между группами нодов) - Вероятность

- Сила связи (Лояльность) - Частота

- Проводимость(Коэффицент потери заряда)

- Распространение (В скольких Скоплениях кластеров присутствует) {Присвоен группе}

Расстояние - Расстяжимый параметр

Проводимость - количество срабатываний \ общее количество активаций

Сила связи - Количество верных срабатываний \ общее количество активаций

(Частота это то как часто мы используем это, условно говоря слова привычки, а проводимость это уже корректность, эти параметры взаимосвязаны ибо и частота и проводимость будут влиять на то сколько в итоге очков мы получим)

К примеру при мысли о букве “H-Hello” , я решил представить места где это используется и первым что пришло мне в голову это Hello, так как я использовал его только что для тестов, а второе слово которое мне пришло в голову, это Helicopter, и это объясняется тем что начало у этих слов схожее.

Получается есть остаточное восприятие которое влияет на предпочитаемый выход?

Но в случае с беседой, я бы не стал использовать “Helicopter”, ведь это никак не ввязывается в контекст беседы. Получается у нас есть ограничения?

И подумав о контексте приветственной беседы, первое до чего я дошел это “How are you?”.

Значит у нас должны быть на данный момент 2 генерируемых объекта.

- Нод

- Связь между Нодами

\*Это является результатом слияния, но по сути группы имеют код, так что они тоже генерируются\*

- Группы\Кластеры нодов (Параметром будет наверное плотность и область покрытия)

Но в случае генерации нужно еще так же внедрить возможность слияния

Получается :

Нод(Заряд)

Группы Нодов(Слова\чувство)

Кластер Нодов(Подтекст)

Скопления Кластеров (Контекст)

Облако\Зона (Категория)

Между этими объектами так же должны формироваться связи, правда уже немного иного типа.

Так же нужна функция пересечения для больших групп.

Нод Заряд - это то насколько крепко его право на то чтобы быть в этой группе.

Сам по себе нод это просто хранилище информации.

Но погодите. Его хранение информации по сути формальное. Ведь технически нашим нейроном является именно группа нодов а не сам нод.

Начальная вероятность равна 100, но разделяется на распространение и скалируется на СИЛУ.

ФОРМА ГРУППЫ НОДОВ ОПРЕДЕЛЯЕТСЯ ТЕМ НАСКОЛЬКО БЛИЗКИ ПО “СМЫСЛУ” ПРИЛЕЖАЩИЕ ГРУППЫ.

В случае с Hello, мы можем сделать первую группу нодов HELL дальше от плавающего нода O, т.е будет происходить деформация самого нижнего по уровню слоя.

Но так же мы можем образовать Helicopter

Получается теперь доминирующая группа это Hel, которая растягивается до огромных масштабов.

Вывод: Расстояние между нодами группы определяется тем , к скольким группам входящие ноды относятся. И Самой кучной группой будет та группа что не несет в себе дополнительных дочерних вероятных групп.

Поскольку единственный объект что не может нести в себе дочерних групп это НОД, такой нод будет являться Корневым НОДОМ.

Проблема с тем как мы вообще будем рассчитывать это самое расстояние? Т.е как нам понять насколько сильно нам нужно деформировать наше облако нодов

Возможно нам нужно внедрить объекты плавающего Нода, которые будут служить скаляром того насколько мы вообще должны изменить наше облако.

Но ведь по сути это и есть нашего надо?

Поскольку у нас нет дополнительных органов чувств, нам придется полагаться на оценку человеком.

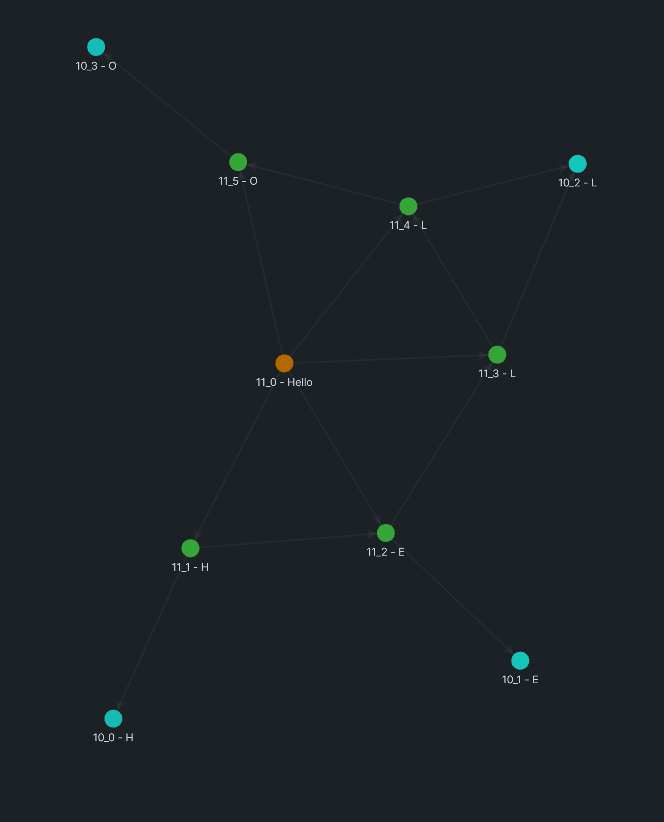
Но даже если мы внедрим дополнительные органы чувств, нам все равно понадобятся корни абстракции.

Потому что слово “Hello”, не имеет за собой какого действительного объекта на который мы можем опираться, а значит его значение, приходит лишь с опытом, того как взаимодействует авторитет.

Вопрос :

- Нужно подумать не о том как это происходит, а зачем?

В общем абстракции это будут еще и своеобразные правила, что то вроде абстракции знания о написании предложения с большой буквы, и просто итоговое значение будет скалироваться на абстракцию



Голубые это кор ноды, в них не может быть дубликата, это нейрон содержащий заряд.

Зеленые, это абстрактные ноды созданные лишь для визуализации Линков

Оранжевый отображает группу нодов, тоже абстракция, но для ясности связал их.

Обсидиан плох в визуализации так что не слишком фокусируемся на это.

Предположительно у нас есть Нод старта, у него фиксированный заряд 1, мы переходим к следующему ноду, и заряд проходя путь к следующему ноду затраичивает какую то часть этой энергии.

Неправильный исход будем уменьшать проводимость энергии , правильный исход будет увеличивать проводимость, и мы будем терять меньшее количество энергии на путь.

Вопрос:

-Как избежать повторение одного и того же слова?

-Как поступать с зарядом в случае длинных слов? Он иссякнет прежде чем мы достигнем? Увеличить итоговое вознаграждение? Но мы тогда будем использовать длинные слова в приоритет коротких? Велик ли смысл?

С другой стороны поощрять сложносочиненные предложения это хорошо, грубо говоря можно сделать так:

Высокая проводимость с маленьким количеством задействованных нейронов, при правильном результате будет минимально влиять на наш I\_Score, и в случае ошибки, сильно уменьшать проводимость.

Низкая проводимость с большим количеством нейронов, будет слабо наказывать, и хорошо поощрять. Однако при достижении высокой проводимости профит будет так же сводится к минимуму.

Правильная конструкция , но неподходящее под контекст\смысл, не должно сильно влиять на проводимость.

\_\_

Нодам не нужны координаты но группам нужны, по сути группы нодов это слова, кластеры это контексты и категории, а облака тогда получается смысл?

Вероятно нам также нужно разделять ноды, на конкретные и обобщающие, ну по сути у нас как раз таки уже есть эти конкретные и обобщающие.

Полагаю у обобщающих заряд изначальный должен быть выше.

Крч у линков нодов, будет только два параметра это сила и проводимость, но в коде, они будут именно высчитываться, т.е храниться в линках будет именно количество активаций и количество верных активаций, чтобы изменение нейрона происходило динамически а не статично.

А расстояние было перенесено для линков групп, потому что вот там оно как раз таки нужно.

Function Link Welder :

Node\_A = Корневой Нод

Node\_B = Нод ветвь

Source = SV|AI

Сначала нужно проверить наличие в памяти словаря нашего корневого нода, потом надо проверить наличие ветви в этом словаре

(Заглавные и строчные это 2 разных нода, и их связи тоже разные)

(Хотя можно было бы оптимизировать это и попробовать ссылаться из заглавного словаря на прописной, но со скаляром, хотя в этом логики мало так что пока что бракуем это)

Источник будет влиять на итог в рассчетах, т.е несмотря на то что параметры у нас считаются отношением, конечный результат это не будет прямой кореляцией отношения, а еще и доверия к источнику этой информации, хотя конечно по началу едва ли это будет сильно влиять на что либо.

Проходя через нейрон или любой раз как мы будем ссылаться на него, мы должны увеличивать его заряд

Формирование связи требует определенного уровня заряда, так что мы не можем выучить все слово за раз

Нод группа,

При установке начальных координат, нам стоит так же опираться на остаточную память, ибо мы предполагаем что новое слова фигурирующее после известного, не находится далеко по смыслу от известного, а значит координаты нового внесенного будут устанавливаться опираясь на соседнюю известную группу.