Школа Информационных Технологий
Университет Карнеги-Меллона

Аннотация

Рекуррентная Каскад-Корреляция (РКК) является рекуррентной версией Каскад-корреляционной обучающей архитектуры Фальмана и Либьера (Fahlman and Lebiere) [Fahlman, 1990]. РКК может научиться с помощью примеров сопоставлять последовательность входов и желаемую последовательность выходов. Новые скрытые узлы с рекуррентными соединениями добавляются в сеть во время обучения по одному в тот момент, когда они становятся необходимы. По сути, сеть строит конечный автомат специально для текущей задачи. РКК сохраняет преимущества Каскад-Корреляции: быстрое обучение, хорошее обобщение, автоматическое построение практически минимальной многослойной сети, а также возможность изучать сложные поведения через последовательность простых уроков. Мощность РКК демонстрируется на двух задачах: создание конечной грамматики из примеров правильных строк, и обучение распознавать символы азбуки Морзе.

1. Архитектура

Каскад-корреляция [Fahlman, 1990] является обучаемой системой с учителем, которая строит почти минимальную многослойную сеть во время курса обучения. Изначально сеть содержит только входы и выходы, а также связи между ними. Эта однослойная сеть обучается (с помощью алгоритма быстрого распространения Quickprop [Fahlman, 1988]) так, чтобы свести к минимуму ошибки. Когда уровень ошибок перестает уменьшаться - оценивается эффективность сети. Если ошибка достаточно мала, то мы останавливаемся. В противном случае мы добавляем новые скрытые узлы к сети, чтобы уменьшить остаточную ошибку.

Чтобы создать новые скрытые узлы, мы начинаем с набора *узлов-кандидатов*, каждый из которых получает на вход сигналы от входов сети и других скрытых узлов, уже присутствующих в сети. Выходы узлов-кандидатов еще не связаны с другими узлами. Мы совершаем несколько проходов по обучающему множеству, и каждый узел-кандидат регулирует свои весовые коэффициенты входящих связей, чтобы увеличить корреляцию между его выходом и остаточной ошибкой сети. Когда оценка корреляции перестает улучшаться, мы выбираем лучшего кандидата, замораживаем его весовые коэффициенты входящих связей и добавляем его к сети. Этот процесс называется "завладением". После завладения, кандидат становится полноценным новым узлом сети. Затем мы заново обучаем все коэффициенты на исходящих связях, в том числе и для нового узла. Этот процесс добавления нового узла и переобучения выходного слоя повторяется, пока ошибка не становится незначительной или мы не решаем сдаться и остановиться. Поскольку новые узлы получают входящие соединения от старых, каждый узел эффективно добавляет новый слой к сети. (См. рисунок 1.)

Каскад-корреляция исключает для пользователя необходимость предугадывать размеры сети, ее глубину и топологию. Достаточно малая (но не минимальная) сеть строится автоматически. Поскольку новые узлы после добавления никогда не меняют своих связей и не удаляются, сети могут быть обучены постепенно. Большой набор данных можно разбить на более мелкие "уроки" и увеличение потенциала будет нарастать.

Каскад-корреляция обучает гораздо быстрее, чем сеть с обратным распространением ошибки по нескольким причинам:
В начале только один слой весовых коэффициентов обучается в любой данный момент

времени. Никогда не возникает необходимости распространять информацию об ошибках обратно через связи, и мы избегаем драматического замедления, которое является типичным при обучении многослойных сетей с обратным распространением ошибки. Во-вторых, это "жадный" алгоритм: каждая новая единица захватывает так много оставшейся ошибки, как только может.
В стандартной сети с обратным распространением ошибки, все скрытые узлы меняются одновременно, конкурируя за различные задачи, что должны быть достигнуты — это медленный и иногда ненадежный процесс.

Каскад-корреляция, как и обратное распространение и другие архитектуры с прямой связью, не имеет кратковременной памяти в сети. Выходы в любой данный момент времени являются функцией только от текущих входов и коэффициентов при связях в сети. Конечно, многие реальные задачи требуют распознания последовательности входов и, в некоторых случаях, создания соответствующей последовательности выходов.

Некоторое количество рекуррентных архитектур было предложены в ответ на эту необходимость. Пожалуй, наиболее широко используемой в настоящее время является модель Эльмана [Elman, 1988], которая предполагает, что сеть оперирует дискретными промежутками времени. Выходы скрытых узлов сети в момент времени T подаются обратно для использования в качестве дополнительных входов сети в момент времени T + 1. (См. Рисунок 2). Эти дополнительные входы могут рассматриваться как переменные состояния, содержание и интерпретация которых определяется меняющимися весами сети. По сути, сеть свободна в выборе ее собственного представления прошлой истории в ходе обучения.

Рекуррентная Каскад-корреляция (РКК) это архитектура, которая добавляет в Каскад-корреляционную архитектуру рекурсивное поведение в стиле Эльмана. Однако, были необходимы некоторые изменения для того, чтобы две модели подходили друг к другу. В оригинальной архитектуре Эльмана существует общая связь между переменными состояния (предыдущими выходами скрытых узлов) и слоем скрытых узлов.

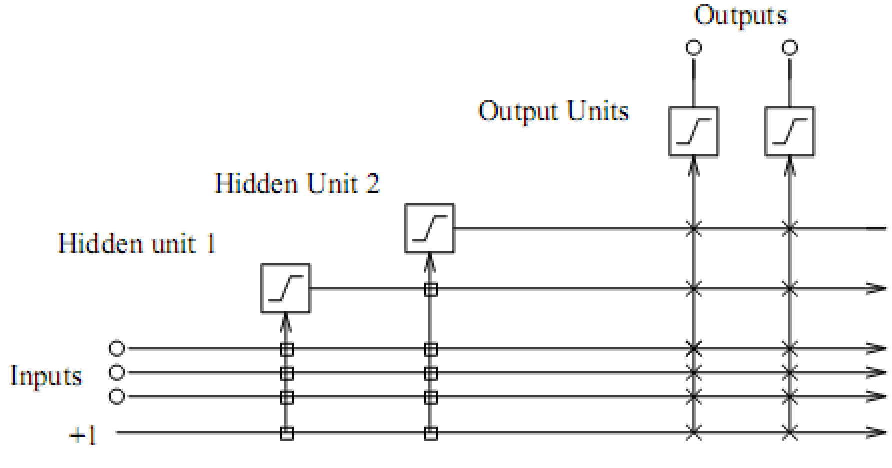

Рисунок 1: Каскадно-корреляционная архитектура, после того как было добавлено два скрытых узла. Вертикальные линии суммируют все входящие импульсы. Соединения в квадратах заморожены, а X-соединения непрерывно обучаются.

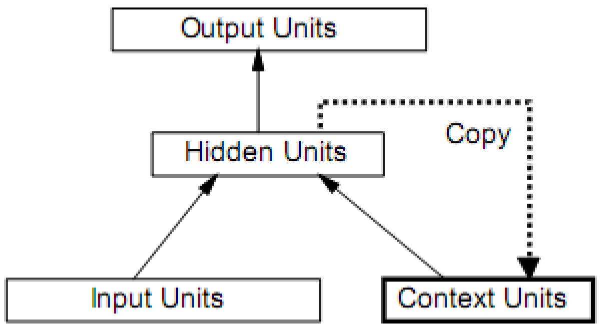

Рисунок 2: Рекуррентная структура сети Эльмана

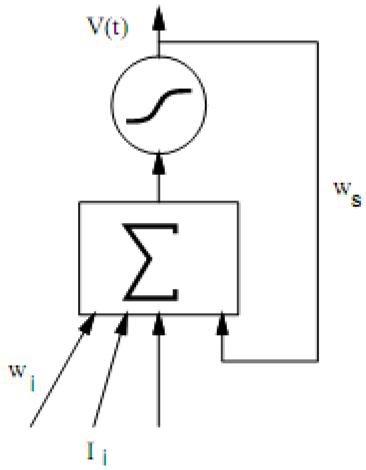

Рисунок 3: Узел-кандидат или скрытый узел с само-рекуррентной связью.

При Каскад-корреляции, новые скрытые узлы добавляются по одному и их состояние замораживается, как только они будут добавлены в сеть. Эта концепция была бы нарушена, если бы выходы новых узлов соединялись со входами уже существующих. С другой стороны, сеть должна иметь возможность создавать рекуррентные петли, если она хочет поддерживать свое состояние в течение неопределенного времени.

Решение, которое мы приняли в РКК: дополнить каждый узел-кандидат одной само-рекуррентной связью с коэффициентом, которая возвращает на вход узла его вывод на предыдущем шаге (рис. 3). Эта само-рекуррентная связь обучается вместе с другими входными коэффициентами узла, чтобы увеличить корреляцию узла с остаточной ошибкой. Если рекуррентная связь примет строго положительное значение — узел будет функционировать в качестве триггера, сохраняя свое прежнее состояние, если другие узлы не заставят его измениться. Если рекуррентная связь примет строго отрицательное значение - узел, как правило, будет колебаться между положительным и отрицательным значением на выходе в каждый следующий шаг времени, если значение на другом входе не удержит его на месте. А если рекуррентный вес близок к нулю, то узел будет выступать в качестве своеобразного логического вентиля. Когда узел-кандидат добавляется к активной сети как новый скрытый узел, его рекуррентный вес замораживается наряду со всеми другими весами. Каждый новый скрытый узел, по сути, является одной переменной состояния в конечном автомате, создаваемом специально для выполнения этой задачи. При использовании только само-рекурентных связей модель РКК походит на алгоритм «Сфокусированного обратного распространения» Мозера [Mozer, 1988].

Выход V (T) каждого само-рекуррентного узла рассчитывается следующим образом:

$$V(t)=\sigma(\sum_i I_i(t)w_i+V(t-1)w_s)$$

где σ является некой нелинейной сжимающей функцией, применяемой к взвешенной сумме входов I плюс рекуррентный вес w_s , умноженый на результат предыдущего выхода. В примерах, описанных тут σ всегда является гиперболическим тангенсом или «симметричной сигма-фунцкией» в пределах [-1; 1]. В течение фазы обучения узла-кандидата мы изменяем веса w_i и w_s для каждого узла так, чтобы увеличить его корреляцию. Это требует вычисления производной от V(t) с учетом этих весов:

$$\partial V(t)/\partial w_i=\sigma'(t)(I_i(t)+w_s\partial V(t-1)/\partial w_i)$$

$$\partial V(t)/\partial w_s=\sigma'(t)(V(t-1)+w_s\partial V(t-1)/\partial w_s)$$

Крайне правый член отражает влияние веса в вопросе на предыдущее состояние узла. Т.к. Мы вычислили $\partial V(t-1)/\partial w$ на предыдущем шаге, мы можем просто сохранить это значение и использовать его на текущем шаге. Итак, рекуррентная версия обучающего алгоритма вынуждает нас хранить одно дополнительное число для веса каждого узла-кандидата, плюс $V(t-1)$ для каждого узла. В момент t = 0 мы подразумеваем, что предыдущее значение узла и предыдущие производные все равны нулю.

Как побочный эффект, обычная формулировка сетей Эльмана считает предыдущие значения скрытых узлов *независимыми* входами, игнорируя зависимость этих предыдущих значений от изменения весов. В сущности, крайние правый части уравнений выше отбрасываются, хотя в общем случае они не являются незначительными. Это грубое приближение очевидно причиняет небольшие проблемы на практике, но оно может объяснить ту нестабильность, которую отметили некоторые исследователи, когда сети Эльмана выполняются с агрессивными процедурами обучения второго порядка, такими как quickprop (быстрое распространение). Алгоритм Мозера принимает во внимание эти дополнительные условия.