Два види рекурентних нейронних мереж: Елмана і Хопфілда.

Рекурентні нейронні мережі мають таку особливість, що вони мають **лінії затримки на вході і зворотні зв'язки** (може бути і те, і те, а може бути лише одне з того). У мережі можуть бути не лише зворотні зв'язки, але й також і прямі, але вже наявність хоч одного зворотнього зв'язку відносить мережу до рекуретної нейронної мережі. Ці мережі створенні для того, щоб моделювати якість динамічні об'єкти, якісь динамічні моделі до об'єктів, якісь різного роду коливання, якісь гармонійні ряди, якісь гармонійні повторюванні процеси. Той, хто з тим працює, той буде знати, що це саме для його процесу потрібна річ. Наприклад, з медичної сфери - електрокардіограма. Ті, хто розшифровують електрокардіограму працюють з подібними мережами. Також є спосіб розпізнавати пряму мову за допомогою рекурентних нейронний мереж. Тобто, все, що йде якиймсь таким потоком послідовним, який може апроксимуватися в більш-менш гармонійну функцію типу синусоїда, це все пробують робити за допомогою рекурентних нейронний мереж.

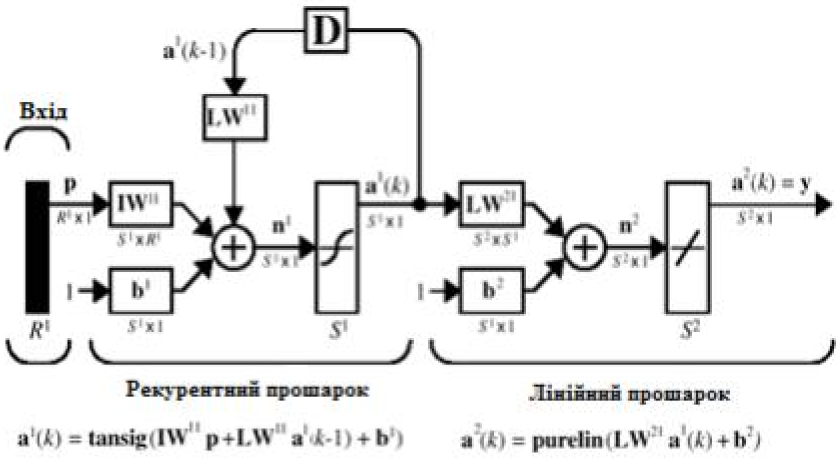
Мережі Елмана (newelm)

дозволяють нам враховувати попередню історію наших процесів і моделей і накопичувати інформацію, яка нам дозволить робити майбутню стратегію в управлінні нашими об'єктами чи в управлінні наших якихось стратегічних рішень. Також можуть застосовуватися для задачі якогось комп'ютерного зору і інших подібного характеру динамічних процесів.

Архітектура

Складається з двох прошарків. Перший прихований прошарок (рекурентний) містить динамічний зворотній зв'язок (він йде від виходу прихованого прошарку назад до його входу), а другий прошарок є вихідним і він є лінійним. На цих прошарках переважно, за замовчування, на прихованому рекурентному прошарку є тангенс гіперболічних **функцій активації tansig**, на вихідному лінійному прошарку - так звана функція **purelin** (лінійна ф-я). Таке поєднання функцій активації дозволяє максимально точно апроксимувати функції скінченного числа точок розриву. Єдина вимога до мережі, полягає в тому, щоб прихований прошарок мав досить велику кількість нейронів, що необхідно для успішної апроксимації складних функцій.

В деякій теорії можуть ще розглядати додатково в мережі Елмана так званий контекстний прошарок (він в нас можна сказати показний де буква **D** на рисунку, тобто те, що вийшло з нашого прихованого прошарку на попередньому кроці, воно не втрачається, а так якби запам'ятовується і грає роль надалі в наступному циклі нашої мережі, тому оте, що запам'яталося з попереднього кроку і передається далі на наступний крок, то що перебуває зараз в нашій пам'яті, чи в хмаринці, чи будь-яким іншим синонімічним словом сказати, це те, що називають контекстний прошарок. Це ніби така якраз динамічна частина, D позначено, бо це динаміка.



Що робиться на рисунку: при вході заходить якийсь вектор, я подаю його на прошарок (де f) перший рекурентний, отримується якийсь вихід з того прошарку (D), він собі запам'ятався, далі те, що вийшло йде на лійний прошарок і на вихід, але наступний елемент формується таким чином, що оте, що в нас щойно вийшло з першого прошарку вже заходить до f, знову на прихований прошарок і формує вже наступну відповідь. Тобто, те, що колись вийшло з попереднього кроку, приходить тепер знову на прихований прошарок і дає наступнку відповідь, воно знов виходить і знов іде на лінійний прошарок, знов дає відповідь, але те, що виходило знов запам'яталося і піде дальше для того, щоб утворювати наступний елемент.

Навчання мережі

Для навчання мережі Елмана можуть бути використані як процедура адаптації, так і процедура навчання, що реалізовуються за допомогою функцій ***adapt*** і ***train*** відповідно.

У процесі процедури **адаптації** на кожному кроці виконуються такі дії:

- моделювання мережі при подачі повного набору векторів входу та обчислення помилки мережі;
- обчислення наближеного градієнта функції помилки, залежної від ваг та зміщень, методом зворотного поширення помилки;
- налаштування ваг з використанням функції налаштування, що вибирається користувачем; рекомендується функція *learngdm*.

У процесі процедури **навчання** на кожному циклі виконуються такі дії:

- моделювання мережі при подачі послідовності вхідних сигналів, порівняння з цільовими виходами та обчислення помилки;
- обчислення наближеного градієнта функції помилки, залежної від ваг та зміщень, методом зворотного поширення помилки;
- налаштування ваг з використанням функції налаштування, що вибирається користувачем; рекомендується функція *traingdx*.

Перевірка мережі

Перевірка мережі

Будемо використовувати для перевірки мережі входи навчальної послідовності:

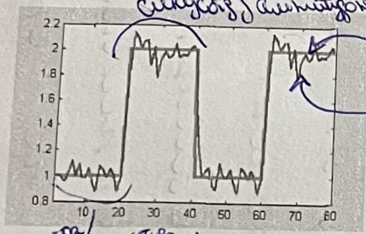
```
figure(2)
a = sim (net, Pseq);
time = 1: length (p);
plot(time, t,'--', time, cat(2,a{:}))
axis([1 80 0.8 2.2]) % Рис.3
```

На рис. 3 наведено графіки вхідного та вихідного сигналів.

синусоїда з амплітудою 2

прямі лінії — те, що мало б вийти (синусоїди)

~ — те, що апроксимувалося за допомогою мережі Елмана



синусоїда/амплітуда 1

Рис. 3

Як випливає з аналізу малюнка, мережа справляється з розв'язанням задачі виявлення амплітуди на наборах навчальної множини. Однак незрозуміло, як вона поводитиметься на інших наборах входу. Чи має побудована мережа Елмана властивість узагальнення? Спробуємо перевірити це, виконавши такі дослідження.

— це коли навчаємо на одних даних, а тоді даємо зовсім інші дані, іншого прикладу і дивимось чи мережа себе адекватно поведе на тих інших даних

Подамо на мережу набір сигналів, складений з двох синусоїд з амплітудами 1.6 та 1.2 відповідно:

```
p3 = sin(1:20)*1.6;
t3 = ones (1,20) * 1.6;
p4 = sin(1:20)*1.2;
t4 = ones (1,20) * 1.2;
pg = [p3 p4 p3 p4];
tg = [t3 t4 t3 t4];
pgseq = con2seq (pg);
figure(3)
a = sim (net, pgseq);
time = 1: length (pg);
```

Робимо **висновок**, що бажано було навчати мережу Елмана на трохи різній кількості амплітуд. Все таки навчити двом видам амплітуд це те саме, що подати два навчальні приклади на вхід. Незважаючи на те, що у нас там насправді сигнали були довгими, тобто дуже багато точок входило, але воно повелося це наче з двома навчальними прикладами, бо ми навчили розпізнавати амплітуду 2 і навчили розпізнавати амплітуду 1. Тому робимо висновок для тих, хто хоче працювати з якимись гармонійними рядами, сигналами типу електрокардіограми, що щоб вам треба було там шукати амплітуду, то треба подавати багато різних характеристик, бо ви не отримаєте властивість узагальнення. А так, рекурентні нейронні мережі мають зараз, потипу Елмана, всякі продовження, всякі новинки, в сенсі, що вони розвиваються до більш глибших нейронних мереж.

Мережі Хопфілда (newhop)

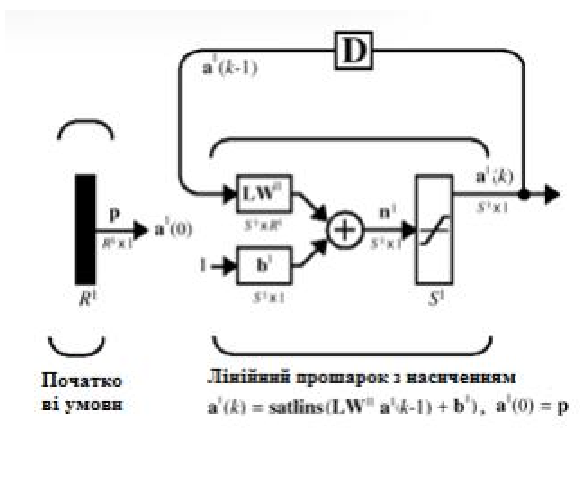
Часто порівнюють з асоціативною пам’ятю людини. Що це таке? Людина має таку властивість, що коли вона бачить якийсь предмет, чи якусь особу, чи будь-що, будь-який предмет, вона починає в своїй пам’яті згадувати. Що таке згадування? Це постійна прогонка крізь мозок, крізь нейроні зв’язки візуального вигляду і пошук у своїй базі даних чогось подібного до того предмету, який ти бачиш. Наприклад, зустрів знайоме лице і починаєш багато-багато раз проганяти під свій мозок, шукаючи все більше відновлюючи лице, бо ти побачив вже його і все більше підганяючи під одне з тих, яке є в твоїй умовній базі даних всіх людей, яких ти знаєш. Тобто, я бачу об’єкт, допустимо, зашумлено і я постійно проганяючи крізь мозок позбуваюся тих шумів, доходячи до якоїсь точки насичення і доходячи до якогось елементу, який є в мене в базі. Так само і працюють мережі Хопфілда.

Архітектура мережі

Вони складаються з одного прошарку. Він має відразу зворотній зв’язок на початок. З кінця на свій же початок задається якась початкова умова, яка задає старт роботи нейронної мережі і від цієї початкової умови мережа має відновитися до певних точок, до однієї з точок в своїй базі даних, а така точка називається ***точка рівноваги***. Тобто, якщо ми проводимо аналогію з лицем, допустимо, наша мережа має в своїй пам’яті кілька лиць і вона там побачила якесь лице і вона проганяє крізь себе багато-багато раз з початку на кінець, з кінця на початок, з початку на кінець, з кінця на початок поки це лице не позбудиться шумів настільки, щоб стало подібне до одного із лиць у базі даних цієї мережі Хопфілда, до однієї з точок рівноваги.

Про точки рівноваги: в кожній мережі Хопфілда є якась база точок рівноваги. Тобто точок, до однієї з яких моя мережа прагнутиме при подачі того чи іншого вхідного сигналу, а вхідний сигнал для мережі Хопфілда називається ***початковою умовою***. Саме тому кажуть(точніше математично доведено), що мережа Хопфілда це щось подібне (точніше, так і є) на систему дифернціальних рівнянь з початковими або крайовими умовами.

Функція активації в середині мережі Хопфілда: функція активації з насиченням ***satlins.*** Вона по краяках така як порогова (вона йде приблизно як y=x (бісектриса 1 і 3 чверті). В нашому випадку a=n.



Є кілька **проблем**, які можуть виникати в мережах Хопфілда при задані того чи іншого початкового статусу. Іноді може бути недостатньо точок рівноваги. До чого це призводить? Якщо я задала якийсь початковий стан в яку з трьох ситуацій (положень) може потрапити моя мережа:

- мережа може перейти в стійкий стан, який відповідає певному еталонному образу, тобто в одній з точок з рівноваги. Тобто, кілька раз то жуючи і перетравлюючи вона може все таки дойти до однієї з точок рівноваги. Це для нас ідеальний випадок.

- мережа може перейти в стійкий стан, який не відповідає ніякому еталонному образу, це тоді кажуть, що вона прийшла в паразитну точку рівноваги. Тобто, те, що нами не назване точкою рівноваги, а вона туди взяла і прийшла.

- мережа може взагалі зациклитися, попавши у якийсь коливальний процес.

Наше завдання щоби відбувався перший випадок, щоб яке б ми не задали початкове значення ми дійшли до однієї з точок рівноваги. Для того, в нас має бути достатня база точок рівноваги. Бо якщо я задаю те, що в моїй базі нема, нейронна мережа трошки схибиться. Також має бути достатня кількість нейронів в мережі Хопфілда.

Як зрозуміти якою має бути база і якою має бути кількість нейронів?

Для цього існує багато підходів:

- одразу задавати таку кількість еталонів і таку кількість нейронів, які пов’язані між собою певними співвідношеннями; на то існують різні гіпотетичні теореми (гіпотетичні, бо вони переважно перевіряються методом спроб і помилок, а не мають якогось прям доведення, хіба є такі які доводять згрубша, але ті доведення часто ламаються на конкретних прикладах)

- є ітеративний формат, який в нас був подібний в радіально-базисній нейронній мережі, коли ми саму нейронну мережу і нейронами, і точками рівноваги, збагачуємо поступово, тобто спочатку вона така бідненька, в неї мало точок рівноваги, мало нейронів, але потіи ми все по трошки додаємо і від того дивимося чи покращується розв’язок нашої задачі.

Гіперкуб - це куб, який залежить від розмірності простору, в тривимірному просторі - це тривимірний кубок, і часто точками рівноваги виступають вершини отих гіперкубів.