

Контрастування НМ

Досліджуються два питання:

- Скільки нейронів потрібно для вирішення задачі?
- Яка має бути структура нейронної мережі?

Об'єднуючи ці два питання, ми отримуємо третє:

- Як зробити роботу нейронної мережі зрозумілою для користувача (логічно прозорою) і які вигоди може принести таке розуміння?

Скільки нейронів треба використовувати?

При відповіді на це питання існує дві протилежні точки зору. Одна з них стверджує, що чим більше нейронів використовувати, тим більше надійна мережа вийде.

Прибічники цієї позиції посилаються на приклад людського мозку. Дійсно, ніж більше нейронів, тим більше число зв'язків між ними, і тим більше складні завдання здатна вирішити нейронна мережа. Крім того, якщо використовувати свідомо більше число нейронів, чим необхідно для вирішення завдання, то нейронна мережа точно навчиться.

Якщо починати з невеликого числа нейронів, то мережа може виявитися нездатною навчитися рішенню задачі, і увесь процес доведеться повторювати спочатку з більшим числом нейронів.

Скільки нейронів треба використовувати?

Другий підхід визначає потрібне число нейронів як мінімально необхідне.

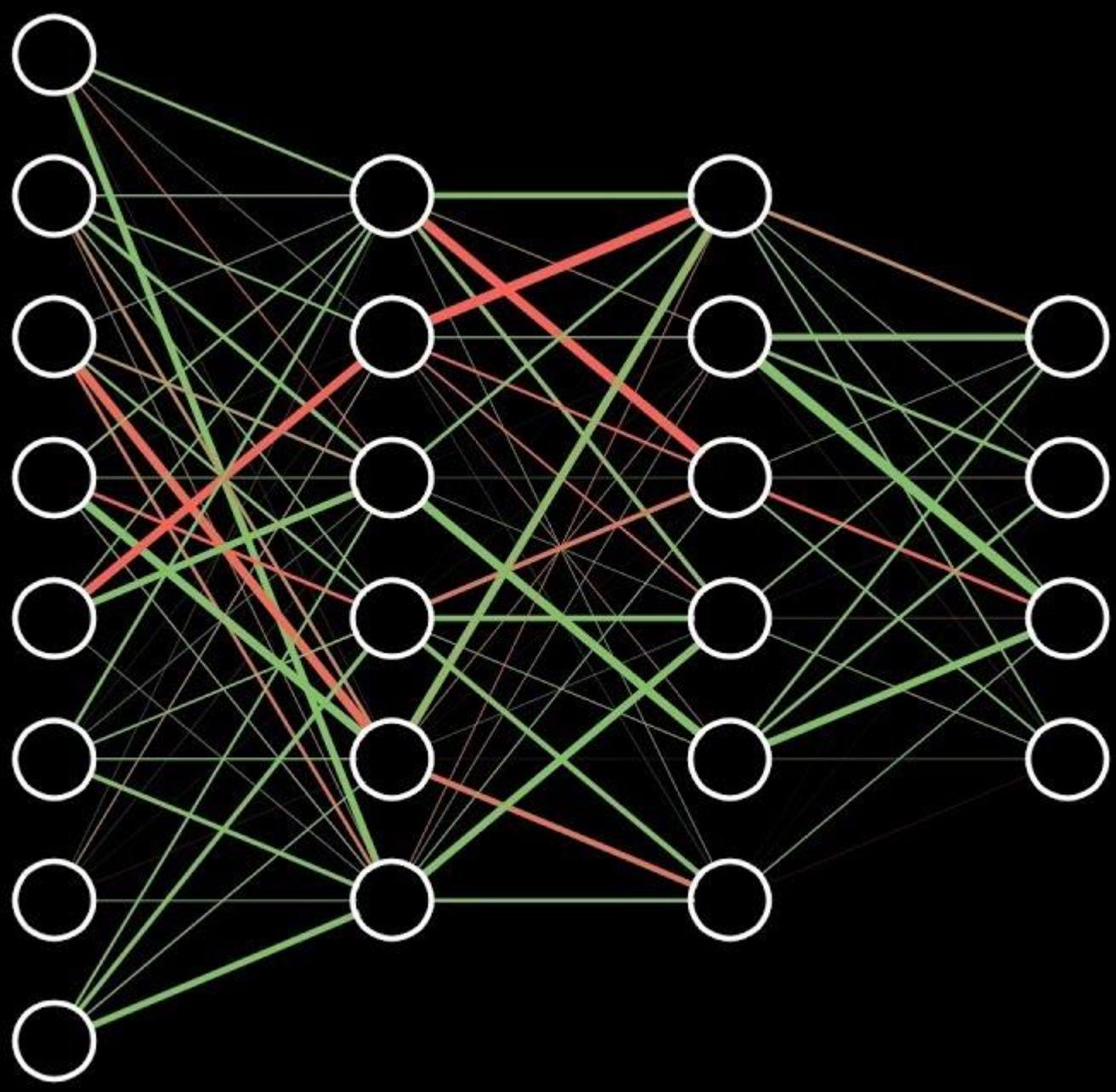
Основним недоліком є те, що це, мінімально необхідне число, заздалегідь невідоме, а процедура його визначення шляхом поступового нарощування числа нейронів дуже трудомістка.

Спираючись на досвід роботи в області медичної діагностики, космічній навігації і психології можна відмітити, що в усіх цих завданнях жодного разу не було потрібно більше декількох десятків нейронів.

Скільки нейронів треба використовувати?

Істина, як завжди буває в таких випадках, лежить посередині: треба вибирати число нейронів більшим чим необхідно, але не набагато. Це можна здійснити шляхом подвоєння числа нейронів в мережі після кожної невдалої спроби навчання.

Проте існує надійніший спосіб оцінки мінімального числа нейронів - використання процедури контрастування. Крім того, процедура контрастування дозволяє відповісти і на друге питання: яка має бути структура мережі.



Контрастування на основі оцінки

Розглянемо мережу, що правильно вирішує усі приклади повчальної великої кількості. Позначимо через $w_p, p=1, \dots, n$ ваги усіх зв'язків. При зворотному функціонуванні мережі за принципом двоїстості або методом зворотного поширення помилки мережа обчислює вектор градієнта функції оцінки E по вагах зв'язків w_k :

$$\nabla E = \left\{ \frac{\partial E}{\partial w_k} \right\}_{k=1}^K$$

Нехай w_p^0 - поточний набір ваг зв'язків, $\left\{ \frac{\partial E}{\partial w_k} \right\}_{k=1}^K$ - помилка поточного прикладу. Тоді в лінійному наближенні можна записати функцію оцінки E в точці w_k як :

Використовуючи це наближення можна оцінити зміну оцінки при заміні w_k^0 на w_k^* як:

$$\chi(k, p) = \left| \frac{\partial E}{\partial w_k} \right| \cdot |w_k^* - w_k^0|$$

де p - номер прикладу навчальної вибірки, для якого були обчислені оцінка і градієнт.

Контрастування на основі оцінки

Величину $\chi(k, p)$ називатимемо показником чутливості до заміни для прикладу p . Далі необхідно обчислити показник чутливості, незалежний від номера прикладу. Для цього можна скористатися будь-якою нормою. Зазвичай використовується рівномірна норма (максимум модуля):

$$\chi(k) = \max_p \chi(k, p)$$

Уміючи обчислювати показники чутливості, можна приступати до процедури контрастування.

Приведемо простий варіант цієї процедури :

1. Обчислюємо показники чутливості для кожної ваги w_k .
2. Знаходимо мінімальний серед показників чутливості для різних прикладів навчальної вибірки.
3. Замінюємо відповідний цьому показнику чутливості вагу на w_k^0 , і виключаємо його з процедури навчання.
4. Пред'явимо мережі усі приклади навчальної вибірки. Якщо мережа не допустила жодної помилки, то переходимо до другого кроку процедури.
5. Намагаємося навчити відконтрастовану мережу. Якщо мережа навчилася безпомилковому рішенню задачі, то переходимо до першого кроку процедури, інакше переходимо до шостого кроку.
6. Відновлюємо мережу в стан до останнього виконання третього кроку. Якщо в ході виконання кроків з другого по п'ятий була відконтрастована хоч би одна вага, (число навчених ваг змінилося), то переходимо до першого кроку. Якщо жодна вага не була відконтрастована, то отримана мінімальна мережа.

Контрастування на основі оцінки

Можливе використання різних узагальнень цієї процедури.

Наприклад, контрастувати за один крок процедури не одну вагу, а задане користувачем число ваг.

Найбільш радикальна процедура полягає в контрастуванні половини ваг зв'язків. Якщо половину ваг відконтрастувати не вдається, то намагаємося відконтрастувати чверть і так далі.

Відмітимо, що при описаному методі обчислення показників чутливості, передбачається можливим обчислення функції оцінки і проведення процедури навчання мережі, а також передбачається відомою навчальна множина.

Контрастування без погіршення

Нехай нам дана тільки навчена нейронна мережа і навчальна множина.

Допустимо, що вид функції оцінки і процедура навчання нейронної мережі невідомі. В цьому випадку так само можливе контрастування мережі.

Припустимо, що ця мережа ідеально вирішує задачу. Тоді нам необхідно так відконтрастувати ваги зв'язків, щоб вихідні сигнали мережі при рішенні усіх завдань змінилися не більше ніж на задану величину.

В цьому випадку контрастування ваг виконується по нейронах. На вході кожного нейрона стоїть адаптивний суматор, який підсумовує вхідні сигнали нейрона, помножені на відповідні ваги зв'язків. Для нейрона найменш чутливою буде та вага, яка при рішенні прикладу дасть найменший вклад в суму. Позначивши через x_k^p вхідні сигнали даного нейрона при рішенні p -го прикладу отримуємо формулу для показника чутливості ваг:

Аналогічно раніше розглянутому отримуємо
$$\chi(k, p) = \left| x_k^p \right| \left| w_k^* - w_k^0 \right|$$

В самій процедурі контрастування $\chi(k)$ - це тільки одна відмінність - замість перевірки на наявність помилок при пред'явленні усіх прикладів перевіряється, що нові вихідні сигнали мережі відрізняються від первинних не більше ніж на задану величину.

Логічно прозорі нейронні мережі

Одним з основних недоліків нейронних мереж, з точки зору багатьох користувачів, є те, що нейронна мережа вирішує задачу, але не може розповісти як. Іншими словами з навченої нейронної мережі не можна витягнути алгоритм рішення задачі. Проте спеціальним чином побудована процедура контрастування дозволяє вирішити і це завдання.

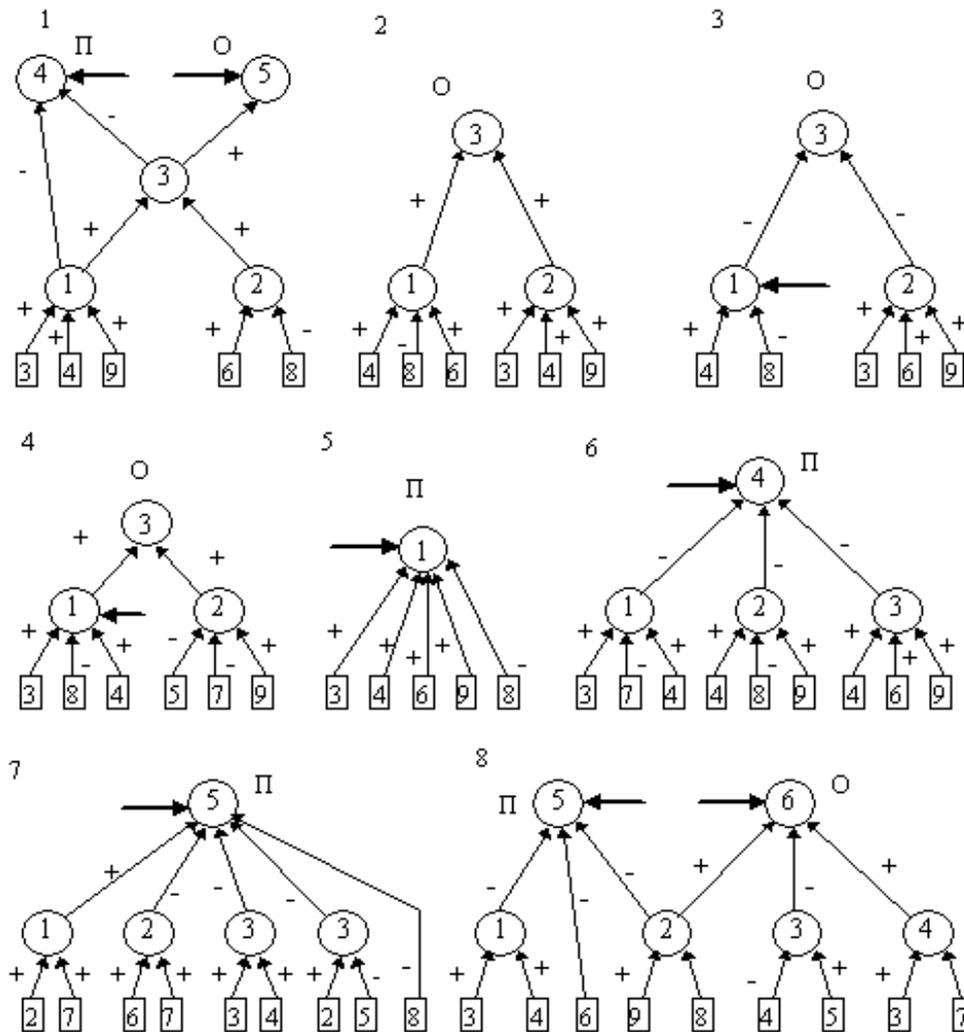
Задамося класом мереж, які вважатимемо логічно прозорими (тобто такими, які вирішують задачу зрозумілим для нас способом, для якого легко сформулювати словесне описи у вигляді явного алгоритму). Наприклад зажадаємо, щоб усі нейрони мали не більше трьох вхідних сигналів.

Логічно прозорі нейронні мережі

Задамося нейронною мережею у якої усі вхідні сигнали подаються на усі нейрони вхідного шару, а усі нейрони кожного наступного шару приймають вихідні сигнали усіх нейронів попереднього шару. Навчимо мережу безпомилковому рішенню задачі.

Після цього виконуємо контрастування у декілька етапів. На першому етапі контрастуватимемо тільки ваги зв'язків нейронів вхідного шару. Якщо після контрастування у деяких нейронів залишилися більше трьох вхідних сигналів, то збільшимо число вхідних нейронів. Потім аналогічну процедуру виконаємо по черзі для усіх інших шарів.

Після завершення описаної процедури буде отримана логічно прозора мережа. Можна зробити додаткове контрастування мережі, щоб отримати мінімальну мережу



Якщо під логічно прозорими мережами розуміти мережі, у яких кожен нейрон має не більше трьох входів, то усі мережі окрім п'ятої і сьомої є логічно прозорими. П'ята і сьома мережі демонструють той факт, що мінімальність мережі не спричиняє логічної прозорості.

Логічно прозорі нейронні мережі

Технологія отримання явних знань з даних за допомогою навчених нейронних мереж виглядає досить просто і начебто не викликає проблем - необхідно її просто реалізувати і користуватися.

Перший етап: навчаємо нейронну мережу вирішувати базову задачу. Зазвичай базовим є завдання розпізнавання, пророцтва (як в попередньому розділі) і тому подібне. В більшості випадків її можна трактувати як завдання про заповнення пропусків в даних. Такими пропусками є і ім'я образу при розпізнаванні, і номер класу, і результат прогнозу, та ін.

Другий етап: за допомогою аналізу показників значущості, контрастування і доучування (усе це застосовується, найчастіше, неодноразово) приводимо нейронну мережу до логічно прозорого виду - так, щоб отриману мережу можна було "прочитати".

Отриманий результат неоднозначний - якщо стартувати з іншої початкової карти, то можна отримати іншу логічно прозору структуру. Кожній базі даних відповідає декілька варіантів явних знань. Можна вважати це недоліком технології, але ми вважаємо, що, навпаки, технологія, що дає єдиний варіант явних знань, недостовірна, а неєдиність результату є фундаментальною властивістю виробництва явних знань з даних.

Логічно прозорі нейронні мережі

Другий етап: за допомогою аналізу показників значущості, контрастування і доучування (усе це застосовується, найчастіше, неодноразово) приводимо нейронну мережу до логічно прозорого виду - так, щоб отриману мережу можна було "прочитати".

Отриманий результат неоднозначний - якщо стартувати з іншої початкової карти, то можна отримати іншу логічно прозору структуру. Кожній базі даних відповідає декілька варіантів явних знань.

Можна вважати це недоліком технології, але ми вважаємо, що, навпаки, технологія, що дає єдиний варіант явних знань, недостовірна, а не єдиність результату є фундаментальною властивістю виробництва явних знань із даних.