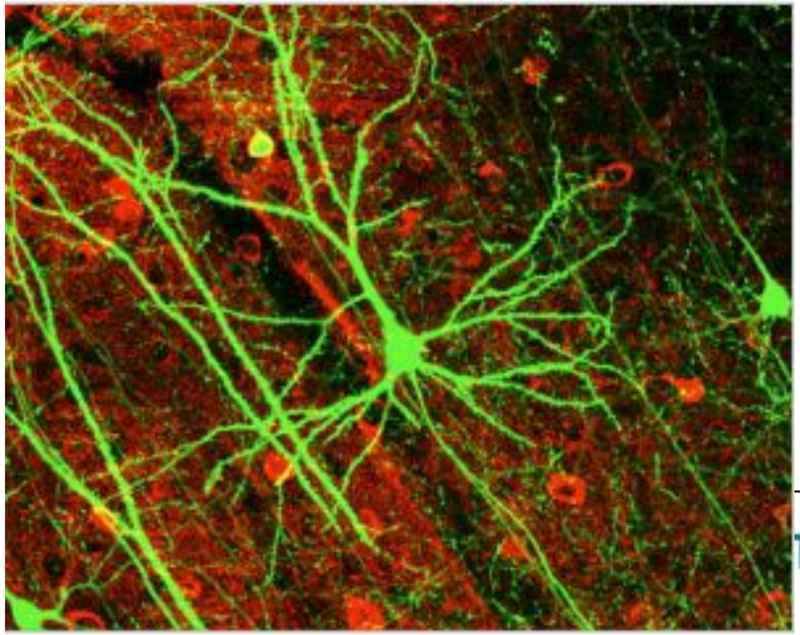
Прецептрон

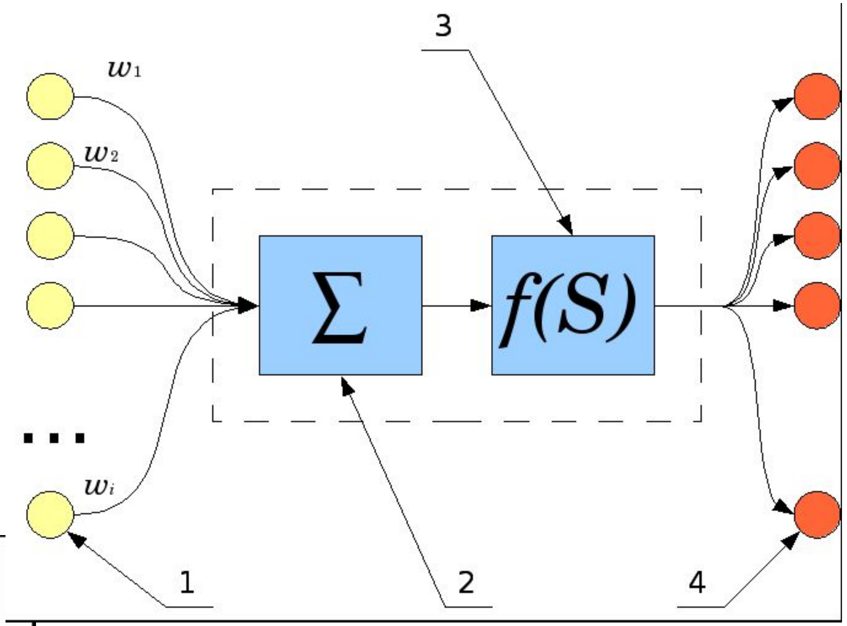
Література з курсу штучного інтелекту

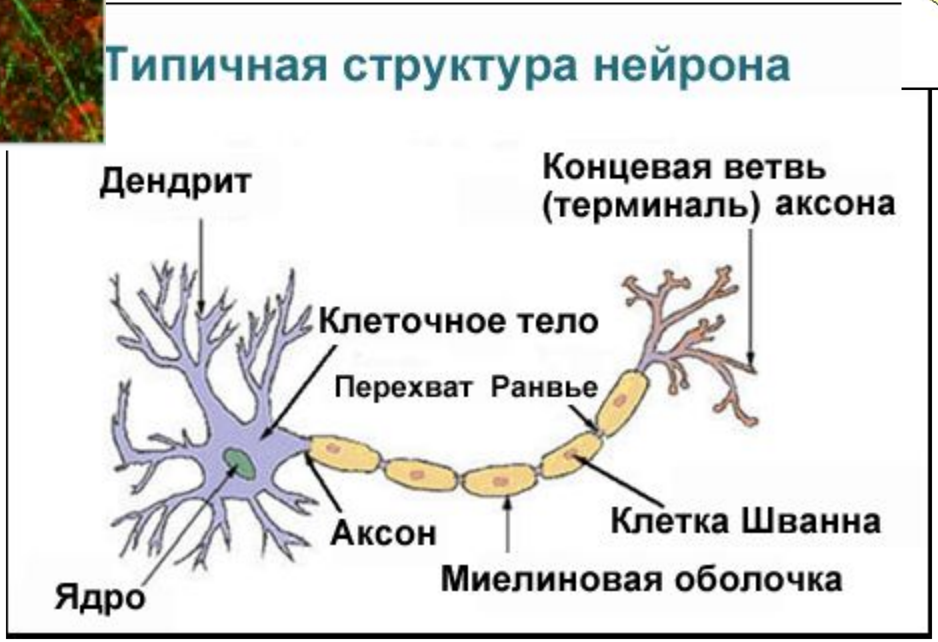
1. *Глибовець М. М., Олецький О.В.* Штучний інтелект. — Київ : «Києво-Могилянська академія», 2002. — 364 с
2. Засоби штучного інтелекту : навч. посіб. / Р. О. Ткаченко, Н. О. Кустра, О. М. Павлюк, У. В. Поліщук ; М-во освіти і науки України, Нац. ун-т "Львів. політехніка". – Львів : Вид-во Львів. політехніки, 2014. – 204 с.
3. Системи штучного інтелекту : навч. посіб. / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина ; за наук. ред. В. В. Пасічника ; М-во освіти і науки, молоді та спорту України. – 2-ге вид., виправл. та доповн. – Львів : Магнолія-2006, 2013. – 279 с.
4. Круглов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия телеком, 2012. – 382с.

Ідея виникнення



Ідея створення штучного нейрона запропонована В. Маккалоком і В. Піттсом в 1950 роках

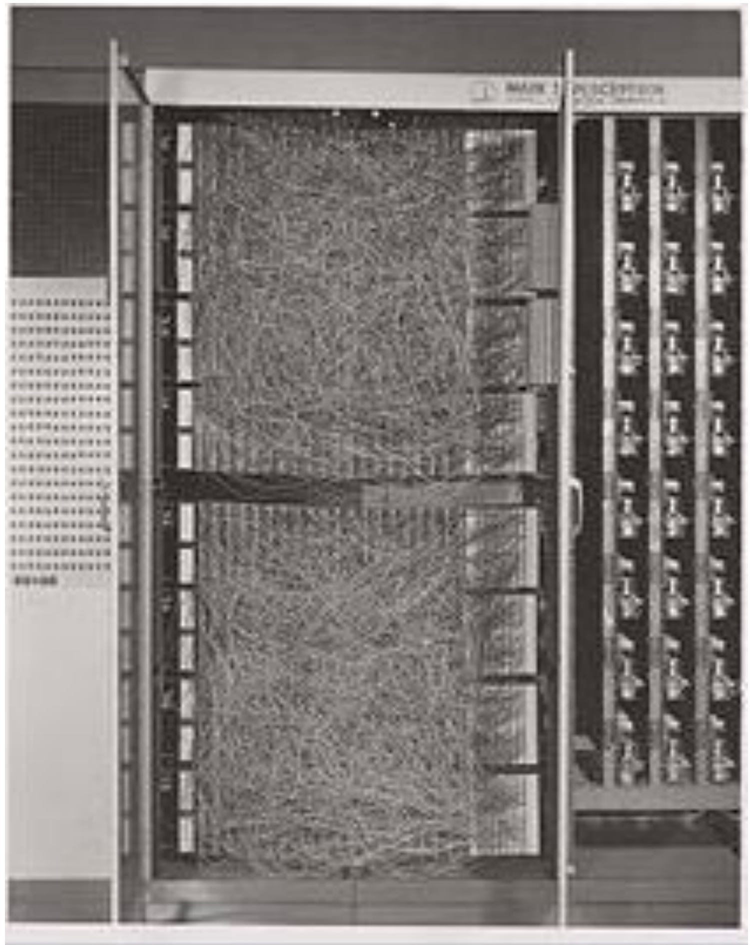


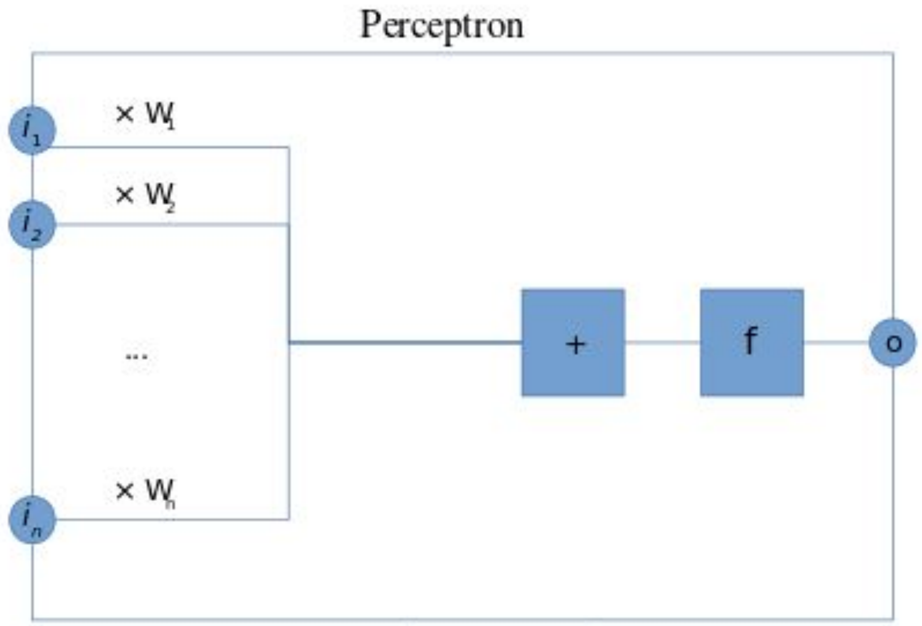


Нейрон – клітина мозку живих організмів

Практична реалізація

Практична реалізація математичної моделі штучного нейрона була здійснена в 1958 році Ф.Розенблаттом. Штучний нейрон був названий Персептроном (Perceptron). Спочатку це була програма, потім був створений пристрій.





$$o = f\left(\sum_{k=1}^{n} i_k \cdot W_k\right)$$

Зв'язки між штучними нейронами

Зв'язки, по яких вихідні сигнали одних нейронів надходять на входи інших, часто називають *синапсами* за аналогією зі зв'язками між біологічними нейронами. Кожен зв'язок характеризується своєю *вагою*. Зв'язки з позитивною вагою називаються *збудливими*, а з негативною - *гальмівними*. Нейрон має один вихід, який часто називають *аксоном* за аналогією з біологічним прототипом. З єдиного виходу нейрона сигнал може надходити на довільне число входів інших нейронів.

Математична модель персептрона та її особливості

Математично персептрон являє собою ваговий суматор єдиний вихід якого визначається через його входи і матрицю ваг таким чином:

$$y = f(u), \qquad u = \sum_{i=1}^{n} w_i x_i + w_0 x_0$$

Тут *i* - відповідно сигнали на входах нейрона і ваги входів, функція u називається індукованим локальним полем, а f(u) - передавальною функцією.

Можливі значення сигналів на входах нейрона вважають заданими в інтервалі . Вони можуть бути або дискретними (0 або 1), або аналоговими. Додатковий вхід і відповідна йому вага використовується для *ініціалізації* нейрона. Під ініціалізацією мається на увазі зсув активаційної функції нейрона по горизонтальній осі, тобто формування порогу чутливості і нейрона. Крім того, іноді до виходу нейрона спеціально додають якусь випадкову величину, яка називається зсувом. Зсув можна розглядати як сигнал на додатковому, завжди навантаженому, синапсі.

Класифікація штучних нейронів

В основному, нейрони класифікують на основі їх положення в топології мережі. Розділяють:

- ***Вхідні нейрони*** - отримують вектор, що кодує вхідний сигнал. Як правило, ці нейрони не виконують обчислювальних операцій, а просто передають отриманий вхідний сигнал на вихід, можливо, посиливши або послабивши його;
- ***Вихідні нейрони*** - являють собою виходи мережі. У вихідних нейронах можуть проводитися будь-які обчислювальні операції;
- ***Проміжні нейрони*** - виконують основні обчислювальні операції.

Основні типи передавальних функцій нейронів

Лінійна передавальна функція → $f(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x \ge 1 \\ x & \text{else} \end{cases}$

Порогова передавальна функція → $f(x) = \begin{cases} 1 & \text{if } x \ge T \\ 0 & \text{else} \end{cases}$

Сигмоїдальна передавальна функція → $\sigma(x) = \frac{1}{1+e^{-x}}$

Логістична функція → $\sigma(x) = \frac{1}{(1+\exp(-tx))}$

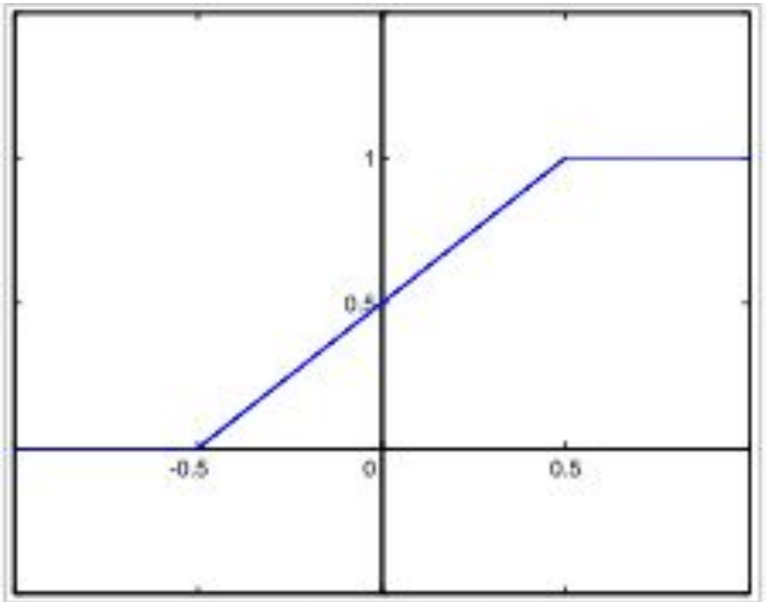
Гіперболічний тангенс → $th(Ax) = \frac{\exp(Ax) - \exp(-Ax)}{\exp(Ax) + \exp(-Ax)}$

Радіально-базисна функція передачі → $y = \exp(-\frac{(S-R)^2}{2\sigma^2})$

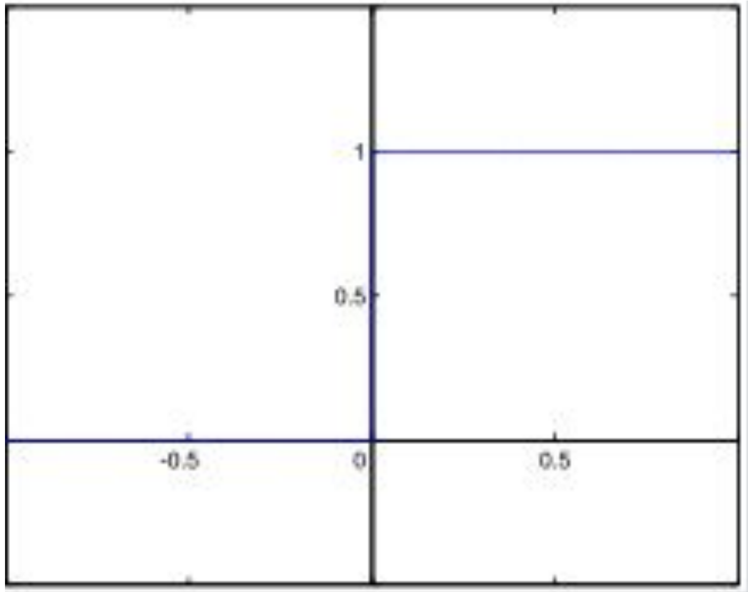
Інші функції передачі →

- Експонента $f(x) = \exp(-Ax)$;
- Тригонометричний синус;
- Модульна: $f(x) = |x|$;
- Квадратична.

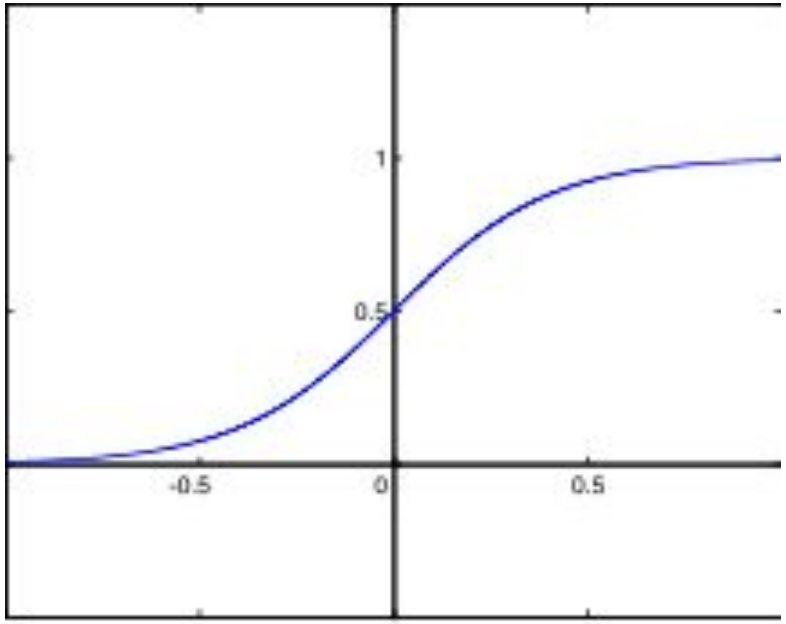
Вигляд передавальних функцій



Лінійна функція активації з насиченням



Порогова функція активації



Сигмоїдна функція активації

Моделювання формальних логічних функцій

Штучний нейрон з пороговою передавальною функцією може моделювати різні логічні функції. Зображення ілюструють, яким чином можна, задавши ваги вхідних сигналів і поріг чутливості, змусити нейрон виконувати [кон'юнкцію](#) (логічне «І») і [диз'юнкцію](#) (логічне «АБО») над вхідними сигналами, а також [Логічне заперечення](#) вхідного сигналу (логічне «НЕ»). Цих трьох операцій достатньо, щоб змоделювати абсолютно будь-яку логічну функцію будь-якого числа аргументів.

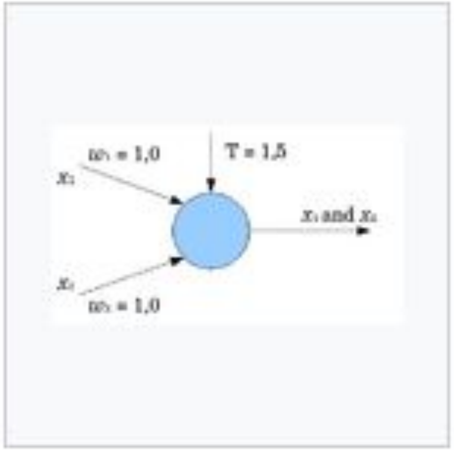


Схема нейрона, налаштованого на моделювання логічного «І»

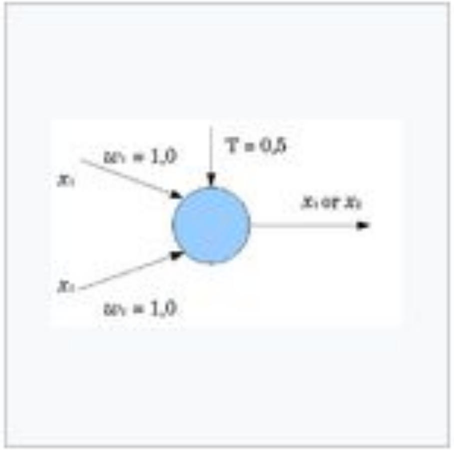


Схема нейрона, налаштованого на моделювання логічного «АБО»

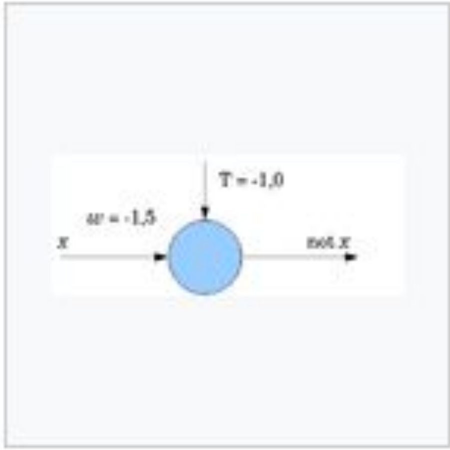
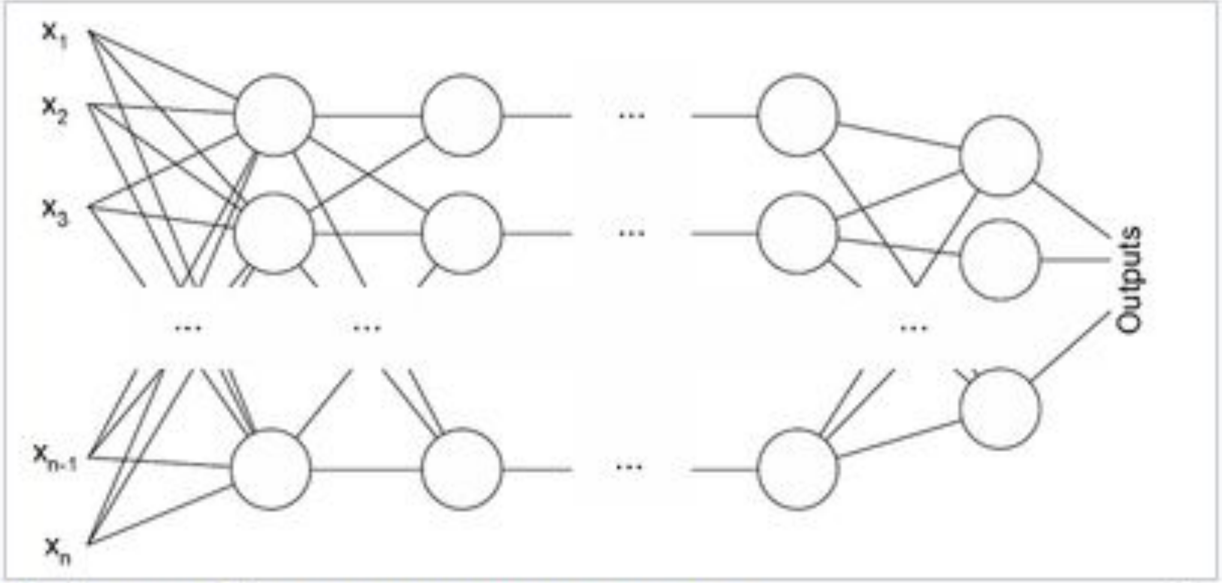


Схема нейрона, налаштованого на моделювання логічного «НЕ»

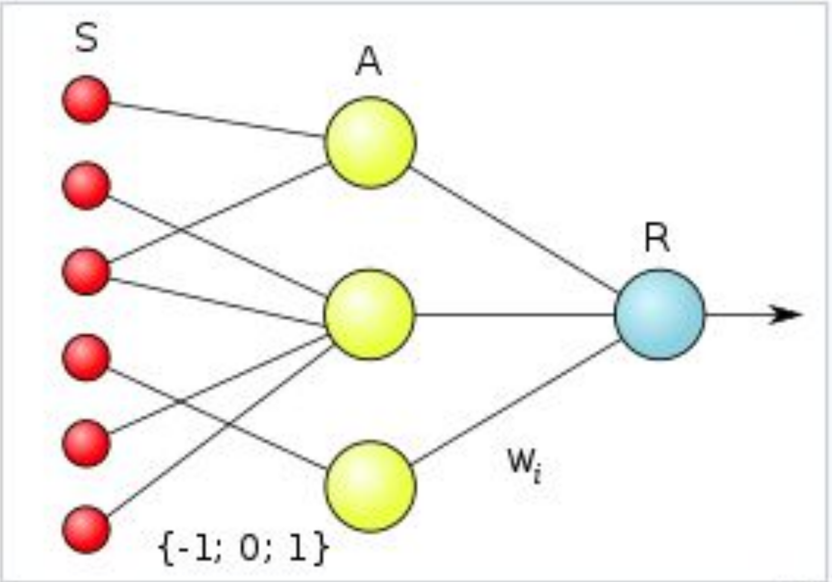
Відмінності між біологічним і штучним нейроном

Нейронні мережі, побудовані на штучних нейронах, виявляють деякі ознаки, які дозволяють зробити припущення про подібність їх структури до структури мозку живих організмів. Тим не менше, навіть на нижчому рівні штучних нейронів існують суттєві відмінності. Наприклад, штучний нейрон є безінерційною системою, тобто сигнал на виході з'являється одночасно з появою сигналів на вході, що зовсім не характерно для біологічного нейрона.

Класифікація персептронів



Архітектура багатошарового перцептрона



Логічна схема елементарного перцептрону. Ваги зв'язків S-A можуть мати значення −1, 1 або 0 (тобто відсутність зв'язку). Ваги зв'язків A-R W можуть мати будь-яке значення

Багатошаровий персептрон

Одношаровий персептрон

Відмінності багатошарового перцептрона від перцептрону Розенблатта

- Використання нелінійної функції активації, як правило сигмоїдної.
- Число шарів, які навчають, більше одного. Найчастіше використовується не більше трьох.
- Сигнали, що надходять на вхід, та одержувані з виходу не бінарні, а можуть кодуватися десятковими числами, які потрібно нормалізувати, так щоб значення були на відрізку була від 0 до 1 (нормалізація необхідна як мінімум для вихідних даних, згідно з функцією активації — сигмоїдою).
- Допускається довільна архітектура зв'язків (у тому числі, і повнозв'язані мережі).
- Помилка мережі обчислюється не як число неправильних образів після ітерації навчання, а як деяка статистична міра нев'язаності між потрібним і одержаним значенням.
- Навчання проводиться не до відсутності помилок після навчання, а до стабілізації вагових коефіцієнтів при навчанні або переривається раніше, щоб уникнути перенавчання.

Відмінності багатошарового перцептрона від перцептрону Розенблатта

Багатошаровий перцептрон буде володіти функціональними перевагами в порівнянні з перцептроном Розенблатта лише в тому випадку, якщо у відповідь на стимули не просто буде виконана якась реакція (оскільки вже в перцептроні може бути отримана реакція кожного типу), а виразиться у підвищенні ефективності вироблення таких реакцій. Наприклад, покращиться здатність до узагальнення, тобто до правильних реакцій на стимули, яким перцептрон не навчався. Але зараз таких узагальнюючих теорем немає, існує лише маса досліджень різних стандартизованих тестів, на яких порівнюються різні архітектури.

Навчання персептрона

Важливою властивістю будь-якої нейронної мережі є здатність до навчання. Процес навчання є процедурою налаштування ваг та порогів з метою зменшення різниці між бажаними (цільовими) та отримуваними векторами на виході. У своїй книзі Розенблат намагався класифікувати різні алгоритми навчання перцептрону, називаючи їх системами підкріплення.

Система підкріплення — це будь-який набір правил, на підставі яких можна змінювати з плином часу матрицю взаємодії (або стан пам'яті) перцептрону.

Описуючи ці системи підкріплення і уточнюючи можливі їхні види, Розенблат ґрунтувався на ідеях Д. Хебба про навчання, запропонованих ним 1949 року, які можна перефразувати в наступне правило, яке складається з двох частин:

- Якщо два нейрони з обох боків синапсу (з'єднання) активізуються одночасно (тобто синхронно), то міцність цього з'єднання зростає.
- Якщо два нейрони з обох боків синапсу активізуються асинхронно, то такий синапс послаблюється або взагалі відмирає.

Навчання персептрону з учителем

Класичний метод навчання перцептрону — це *метод корекції помилки*. Він являє собою такий вид навчання з учителем, при якому вага зв'язку не змінюється до тих пір, поки поточна реакція перцептрона залишається правильною. При появі неправильної реакції вага змінюється на одиницю, а знак (+/-) визначається протилежним від знаку помилки.

Припустимо, ми хочемо навчити перцептрон розділяти два класи об'єктів так, щоби при пред'явленні об'єктів першого класу вихід перцептрона був позитивний (+1), а при пред'явленні об'єктів другого класу — негативним (-1). Для цього виконаємо наступний алгоритм:

Навчання персептрону без учителя

Крім класичного методу навчання перцептрону, Розенблат також ввів поняття про навчання без учителя, запропонувавши наступний спосіб навчання:

Альфа-система підкріплення — це система підкріплення, за якої ваги всіх *активних* зв'язків, що ведуть до елемента, змінюються на однакову величину r, а ваги *неактивних* зв'язків за цей час не змінюються.

Пізніше, з розробкою поняття багатошарового перцептрону, альфа-систему було модифіковано, і її стали називати *дельта-правилом*.

Модифікацію було проведено з метою зробити функцію навчання диференційовною (наприклад, сигмоїдною), що в свою чергу потрібно для застосування методу градієнтного спуску, завдяки якому можливе навчання більше ніж одного шару…

Застосування перцептронів

Перцептрон може бути використано, наприклад, для апроксимації функцій, для задачі *прогнозування* (й еквівалентної їй задачі *розпізнавання образів*), що вимагає високої точності, та задачі *керування агентами*, що вимагає високої швидкості навчання.

У практичних задачах від перцептрона вимагатиметься можливість вибору більш ніж з двох варіантів, а отже, на виході в нього має бути більше одного R-елемента. Як показано Розенблатом, характеристики таких систем не відрізняються суттєво від характеристик елементарного перцептрона.

Апроксимація функції за допомогою персептрона

Теорема Цибенко, доведена Георгієм Цибенко[en] 1989 року, стверджує, що штучна нейронна мережа прямого поширення з одним прихованим шаром може апроксимувати будь-яку неперервну функцію багатьох змінних з будь-якою точністю. Умовами є достатня кількість нейронів прихованого шару, вдалий підбір $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N, \alpha,$ і θ, де

- $\mathbf{w}_i$ — ваги між вхідними нейронами і нейронами прихованого шару
- α — ваги між зв'язками від нейронів прихованого шару і вихідним нейроном
- θ — коефіцієнт «упередженості» для нейронів прихованого шару.

Прогнозування та розпізнавання образів

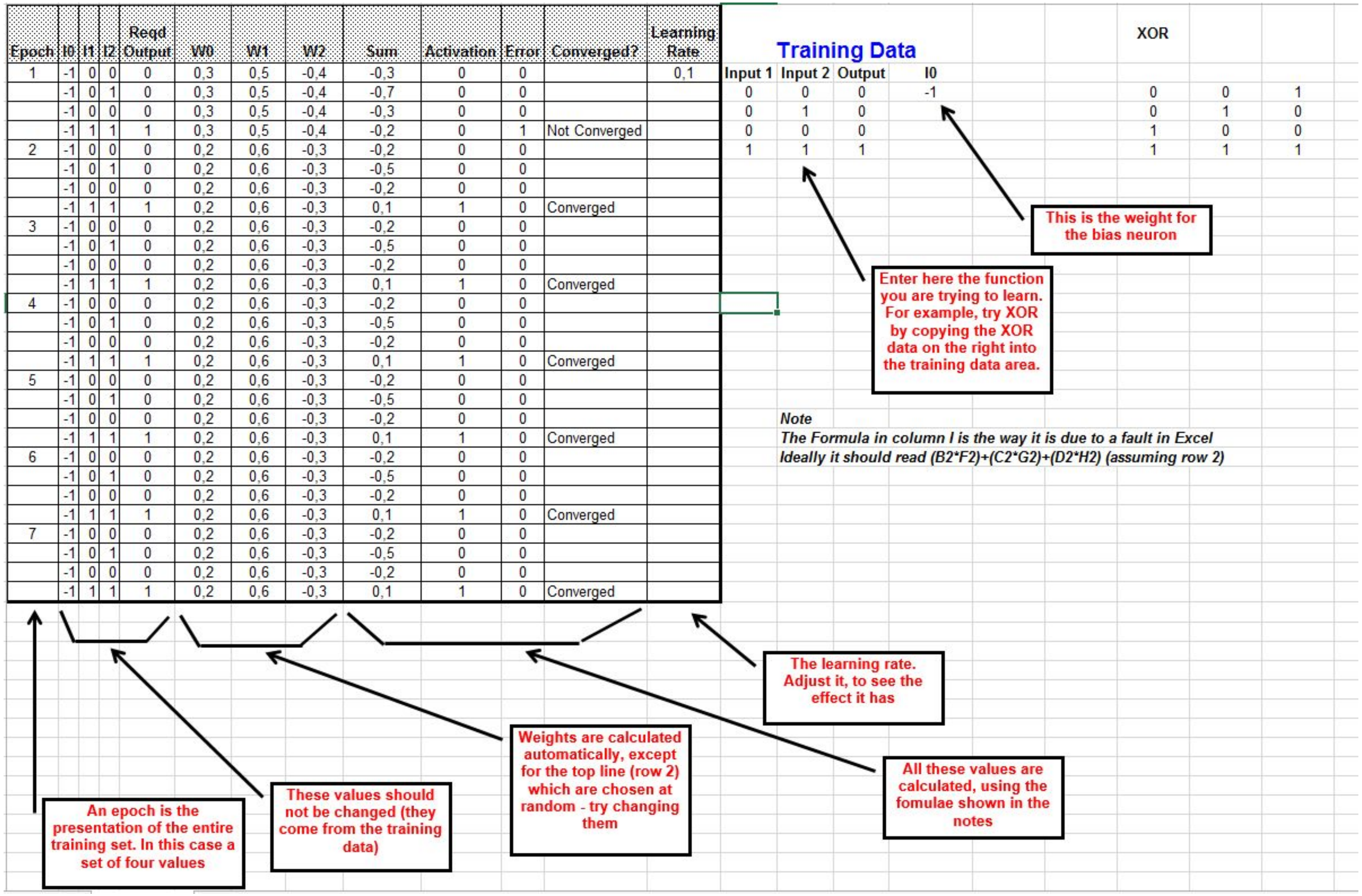
У цих завданнях перцептронові потрібно встановити приналежність об'єкта до якогось класу за його параметрами (наприклад, за зовнішнім виглядом, формою, силуету). Причому точність розпізнавання багато в чому залежатиме від представлення вихідних реакцій перцептрону. Тут можливі три типи кодування: конфігураційне, позиційне та гібридне. Позиційне кодування, за якого кожному класові відповідає свій R-елемент, дає точніші результати, ніж інші види. Такий тип використано, наприклад, у праці Е. Куссуль та ін. «Перцептрони Розенблата для розпізнавання рукописних цифр». Однак воно є незастосовним у тих випадках, коли кількість класів є значною, наприклад, кілька сотень. У таких випадках можна застосовувати гібридне конфігураційно-позиційне кодування, як це було зроблено у праці Яковлева «Система розпізнавання рухомих об'єктів на базі штучних нейронних мереж».

Керування агентами

У теорії штучного інтелекту часто розглядають агентів, що навчаються (адаптуються до навколишнього середовища). При цьому в умовах невизначеності стає важливим аналізувати не лише поточну інформацію, а й загальний контекст ситуації, в яку потрапив агент, тому тут застосовують перцептрони зі зворотним зв'язком.

Крім того, в деяких задачах стає важливим підвищення швидкості навчання перцептрона, наприклад, за допомогою моделювання рефрактерності.

Підготовка до семінару



Epoch	I0	I1	I2	Reqd Output	W0	W1	W2	Sum	Activation	Error	Converged?	Learning Rate
1	-1	0	0	0	0,3	0,5	-0,4	-0,3	0	0		0,1
	-1	0	1	0	0,3	0,5	-0,4	-0,7	0	0		
	-1	0	0	0	0,3	0,5	-0,4	-0,3	0	0		
	-1	1	1	1	0,3	0,5	-0,4	-0,2	0	1	Not Converged	
2	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	0	1	0	0,2	0,6	-0,3	-0,5	0	0		
	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	1	1	1	0,2	0,6	-0,3	0,1	1	0	Converged	
3	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	0	1	0	0,2	0,6	-0,3	-0,5	0	0		
	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	1	1	1	0,2	0,6	-0,3	0,1	1	0	Converged	
4	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	0	1	0	0,2	0,6	-0,3	-0,5	0	0		
	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	1	1	1	0,2	0,6	-0,3	0,1	1	0	Converged	
5	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	0	1	0	0,2	0,6	-0,3	-0,5	0	0		
	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	1	1	1	0,2	0,6	-0,3	0,1	1	0	Converged	
6	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	0	1	0	0,2	0,6	-0,3	-0,5	0	0		
	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	1	1	1	0,2	0,6	-0,3	0,1	1	0	Converged	
7	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	0	1	0	0,2	0,6	-0,3	-0,5	0	0		
	-1	0	0	0	0,2	0,6	-0,3	-0,2	0	0		
	-1	1	1	1	0,2	0,6	-0,3	0,1	1	0	Converged	

Training Data

Input 1	Input 2	Output	I0
0	0	0	-1
0	1	0	
0	0	0	
1	1	1	

XOR

0	0	1
0	1	0
1	0	0
1	1	1

Note
The Formula in column I is the way it is due to a fault in Excel
*Ideally it should read (B2*F2)+(C2*G2)+(D2*H2) (assuming row 2)*