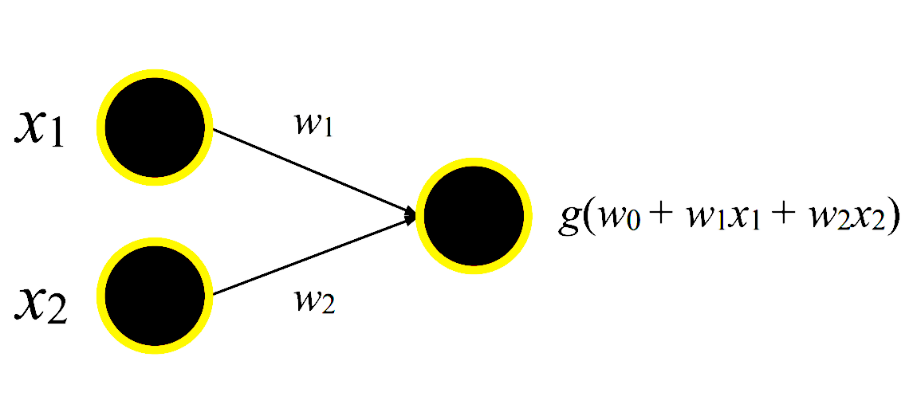
**Лекція 5. Нейронні мережі**

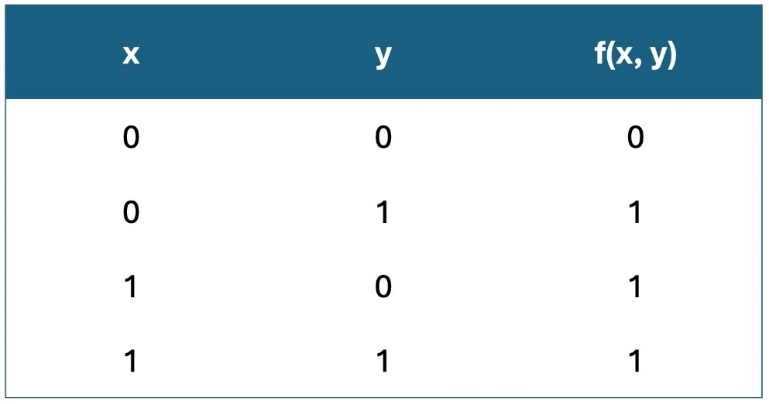
1. **Структура нейромережі**
2. **Багатошарові нейронні мережі**
3. **Алгоритм зворотного поширення помилки**
4. **Комп'ютерне бачення**
5. **Рекурентні нейронні мережі**
6. **Структура нейромережі**

Нейронна мережа може бути візуальним відображенням ідеї, описаної в лінійній регресії, де функція додає вхідні дані, щоби отримати вихідні.

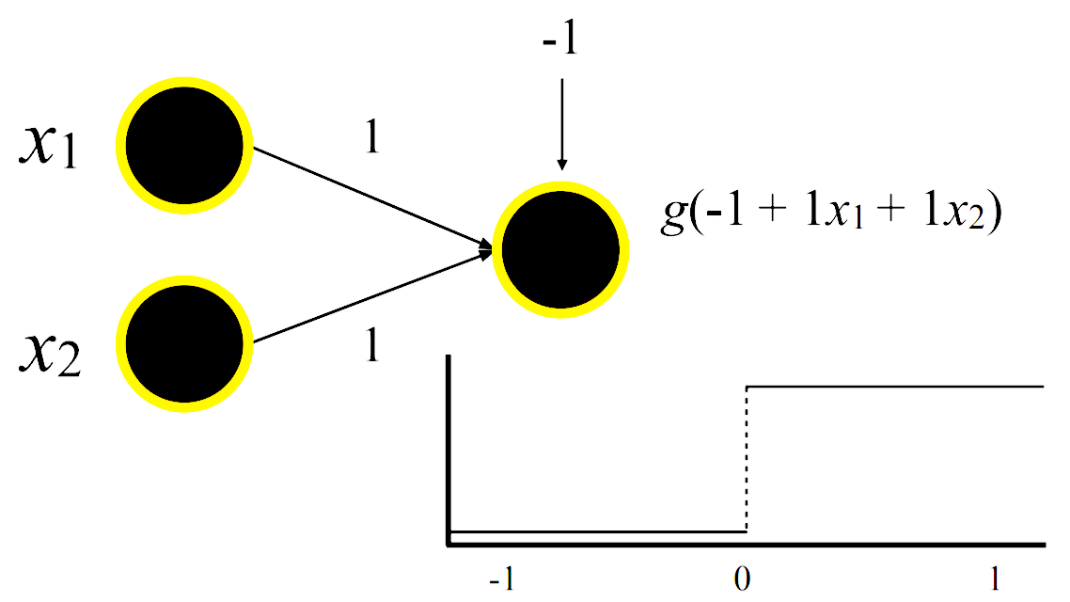


Два чорних вузли ліворуч — це входи, а вузол праворуч — вихід. Входи з’єднані з виходом за допомогою ребер із ваговими коефіцієнтами. Щоби ухвалити рішення, вихідний елемент множитиме вхідні вузли на вагові коефіцієнти, додасть зміщення (w₀), та використає функцію g, аби отримати вихід.

Наприклад, логічний диз'юнктор («або») можна подати як функцію f з такою таблицею істинності:



Якщо графічно відобразити цю функцію як нейронну мережу, де x₁ — перший вхідний елемент, x₂ — другий вхідний елемент, та вони з'єднані з вихідним елементом ребром з ваговим коефіцієнтом 1. То вихідний елемент використовуватиме функцію g(-1 + 1x₁ + 2x₂) з порогом 0, щоби повернути або 0, або 1 (хибне чи істинне).



Приміром, у випадку, коли x₁ = x₂ = 0, сума становить (-1). Це нижче порогового значення, тому функція g виведе 0. Однак, якщо одне або обидва значення x₁ або x₂ є 1, сума всіх вхідних вузлів буде або 0, або 1. Обидва значення на пороговому рівні або перевищують його, тому функція виведе 1.

**2. Біологічний нейрон та формальна модель нейрона Маккалокі та Піттса**

Ідея побудови штучних нейронних мереж ґрунтується на моделюванні роботи людського мозку.

Нервова система та мозок людини складаються з клітин – нейронів, об'єднаних у біологічну нейронну мережу. Тому, перш ніж приступати до вивчення штучних нейронних мереж, корисно познайомитись із конструкцією їх біологічного прототипу.

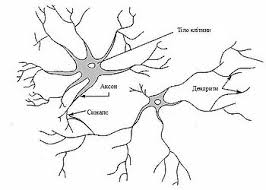
Елементом клітинної структури мозку є особлива біологічна клітина нейрон. Біологічний нейрон містить такі структурні одиниці (рис. 6.1)

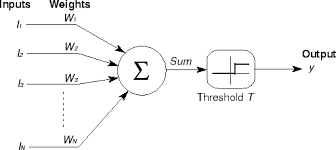
1) тіло клітини або сома (soma), яке містить ядро, та інші елементи, що підтримують її роботу,

2) два типи зовнішніх деревоподібних гілок:

- аксонів (axon), що забезпечують проведення імпульсу та передачу впливу на інші нейрони або м'язові волокна.

- дендритів (dendrites) - вхідні волокна, що збирають інформацію від інших нейронів.

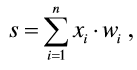
3) синапс – місце контакту нервових волокон – передає збудження від клітини до клітини. Передача через синапс майже завждиодноспрямована

У 1943 році Дж. Маккалокі та У. Пітт запропонували формальну модель біологічного нейрона як пристрою, що має кілька входів (вхідні синапси – дендрити), і один вихід (вихідний синапс – аксон) 

Класична модель нейрона Дж. Маккалокі та У. Пітта

Дендрити отримують інформацію від джерел інформації (рецепторів) Xi, якими можуть виступати і нейрони.

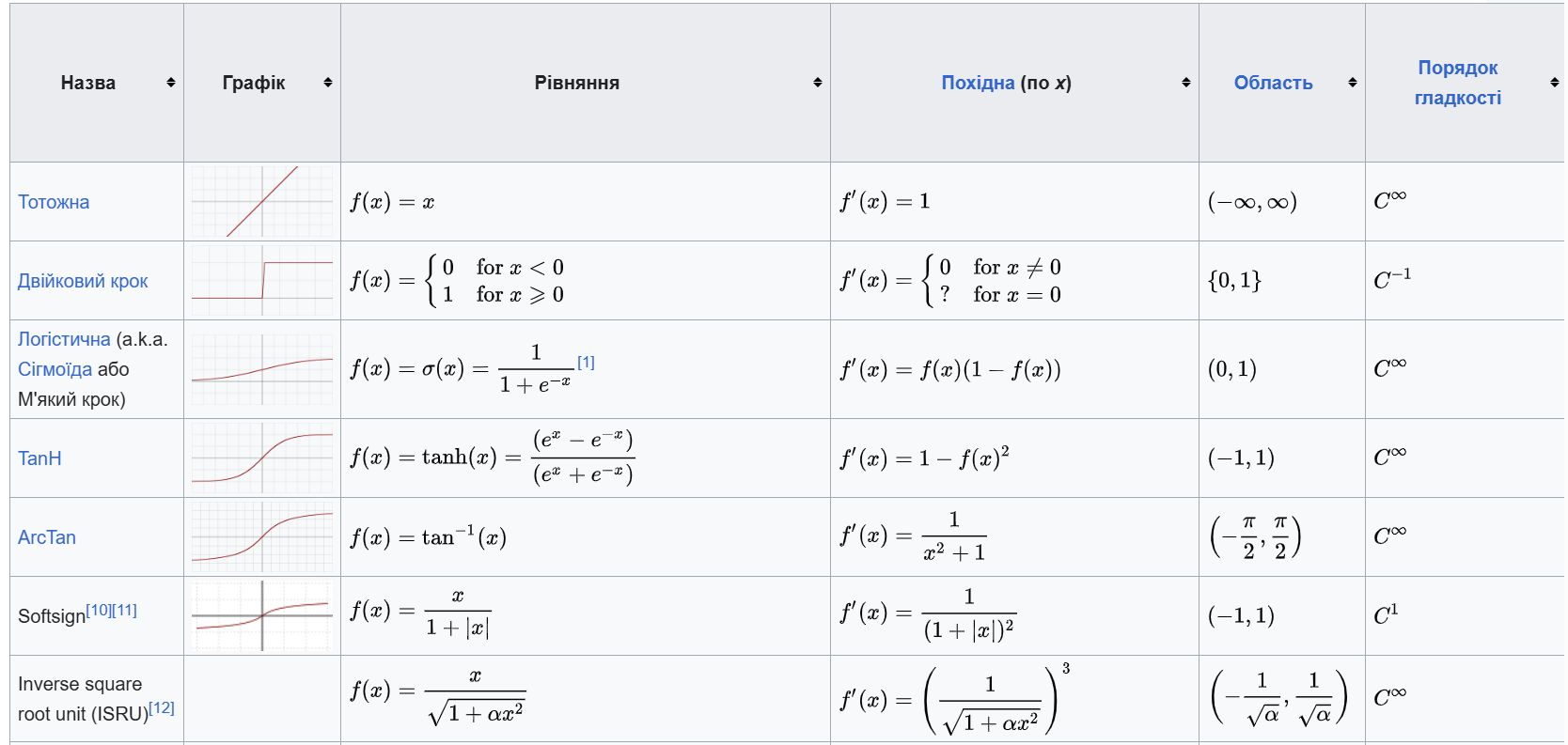
Набір вхідних сигналів {Xi} характеризує об'єкт, його стан чи ситуацію, що обробляється нейроном. Кожному i-му входу j-го нейрона ставиться у відповідність деякий ваговий коефіцієнт wij, що характеризує ступінь впливу сигналу з цього входу на аргумент передавальної (активаційної) функції, що визначає сигнал Yj на виході нейрона. У нейроні відбувається зважене підсумовування вхідних сигналів, і це значення використовується як аргумент активаційної (передавальної) функції нейрона. Таким чином, поточний стан нейрона визначається як зважена сума його входів:

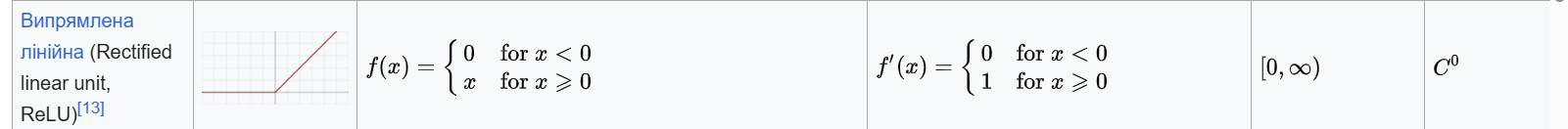


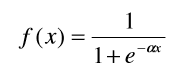
а вихід нейрона є функція його стану:



Активаційна функція нейрона f(s) може бути різних видів

: 



Однією з найпоширеніших є нелінійна функція з насиченням, так звана логістична функція або сигмоїд (тобто функція S-подібного вигляду):

При зменшенні α сигмоїд стає більш пологим, у межі при α=0 вироджуючись в горизонтальну лінію на рівні 0.5, при збільшенні α сигмоїд наближається на вигляд до функції одиничного стрибка з порогом T в точці x=0. З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить у діапазоні [0, 1].

Одна з цінних властивостей сигмоїдної функції - простий вираз для її похідної, застосування якого буде розглянуто надалі.



1. **Багатошарові нейронні мережі**

Багатошарові нейронні мережі (або багатошарові перцептрони, MLP) — це тип штучних нейронних мереж, які складаються з кількох шарів нейронів: вхідного шару, одного або більше прихованих шарів, і вихідного шару. Кожен шар нейронів підключений до наступного через ваги та зсуви, що дозволяє моделі вивчати складні залежності в даних.

Основні компоненти багатошарової нейронної мережі:

Вхідний шар: отримує вхідні дані та передає їх до прихованого шару.

Приховані шари: виконують нелінійні перетворення за допомогою активаційних функцій (ReLU, Sigmoid, tanh), що дозволяє моделі навчатися складним закономірностям у даних.

Вихідний шар: генерує остаточний результат, який може бути класифікацією або прогнозом.

Особливість багатошарових нейронних мереж полягає в тому, що завдяки наявності кількох прихованих шарів вони здатні моделювати більш складні нелінійні залежності між вхідними даними та виходами, ніж прості лінійні моделі.

Процес навчання в багатошарових мережах зазвичай відбувається за допомогою алгоритму зворотного поширення помилки (backpropagation), що дозволяє моделі коригувати ваги, щоб мінімізувати різницю між передбачуваним результатом і фактичним.

Найбільш поширені функції активації:

* **ReLU (Rectified Linear Unit):** дозволяє передавати тільки додатні значення, а негативні обнуляє.
* **Sigmoid:** перетворює значення в діапазон від 0 до 1, використовується для бінарних класифікацій.
* **tanh:** переводить значення в діапазон від -1 до 1, допомагає центрирувати дані.

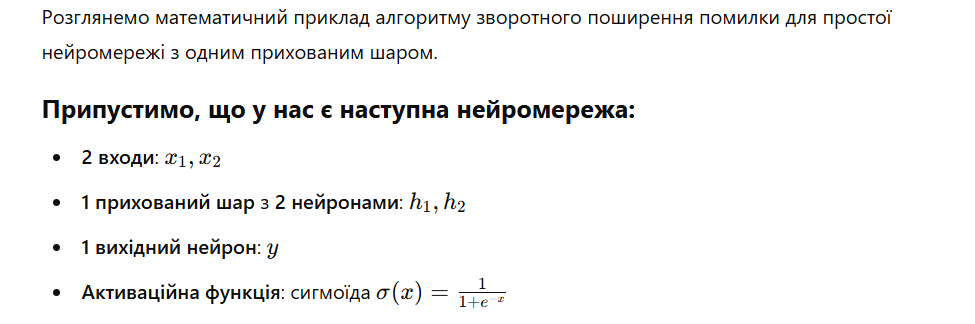
Ці нелінійні функції дозволяють нейронним мережам моделювати складніші зв'язки між вхідними даними і виходами, що значно підвищує їхню потужність у розв'язанні різних задач.

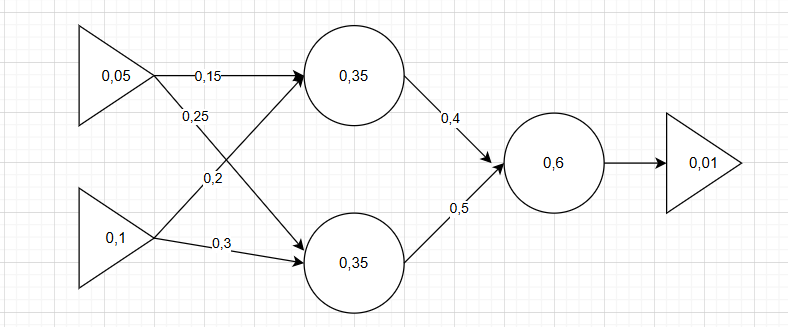
<https://developers.google.com/machine-learning/crash-course/neural-networks/interactive-exercises?hl=ru>

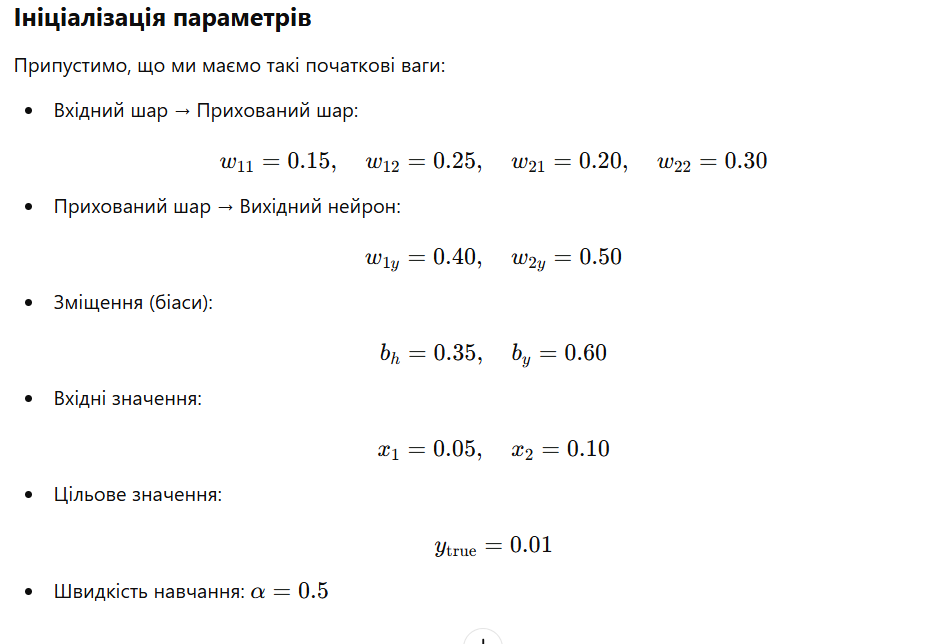
1. **Алгоритм зворотного поширення помилки**

Алгоритм зворотного поширення помилки (backpropagation) — це ключовий метод навчання багатошарових нейронних мереж. Він використовується для коригування ваг нейронів, щоб зменшити помилку між прогнозами моделі і фактичними значеннями. Ось основні етапи цього алгоритму:

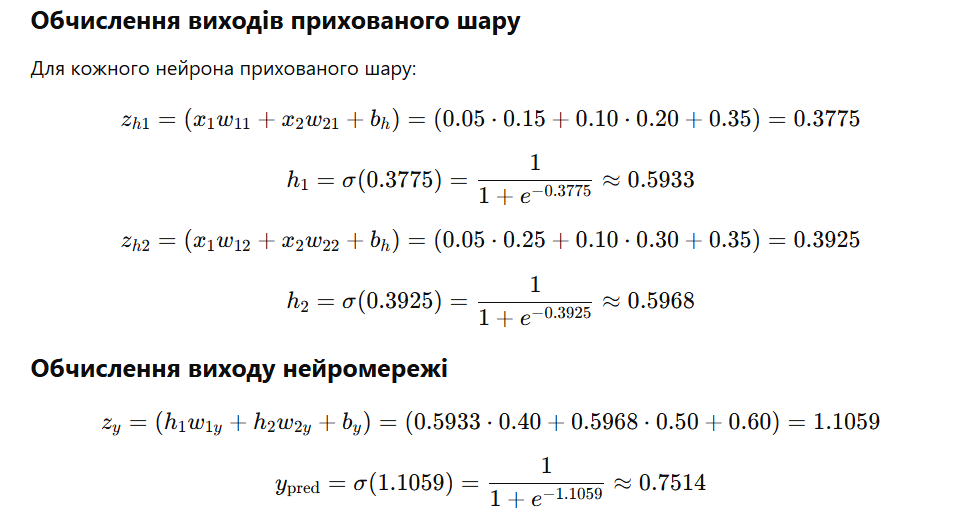
1. **Пряме поширення (Forward Pass):**
   * Вхідні дані передаються через мережу, проходячи через всі шари (вхідний, приховані та вихідний).
   * Кожен нейрон обчислює зважену суму своїх входів і пропускає її через активаційну функцію, щоб отримати вихід.
2. **Обчислення помилки:**
   * Після отримання виходу моделі обчислюється помилка (loss) між прогнозованим значенням і фактичним (цільовим) значенням за допомогою функції втрат, наприклад, середньоквадратичної помилки або крос-ентропії.
3. **Зворотне поширення (Backward Pass):**
   * Помилка поширюється назад через мережу. Для кожного нейрону обчислюється градієнт помилки по відношенню до ваги, використовуючи правило ланцюга.
   * Для виходу нейронів обчислюється похідна функції втрат по відношенню до виходу нейронів, а потім через активаційні функції — по відношенню до ваг.
4. **Оновлення ваг:**
5. **Повторення:**
   * Ці етапи повторюються для кількох ітерацій (епох), поки модель не досягне прийнятного рівня точності.



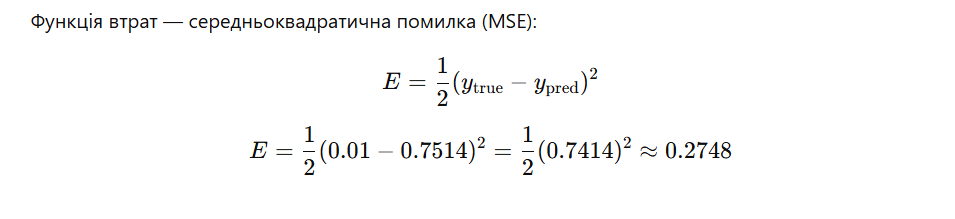




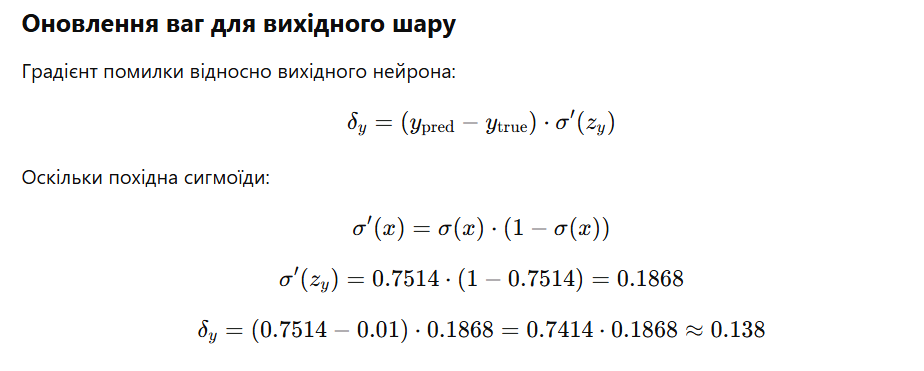
Пряме поширення

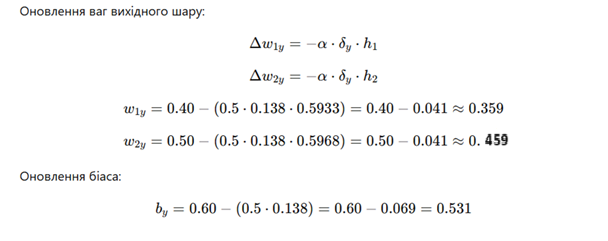


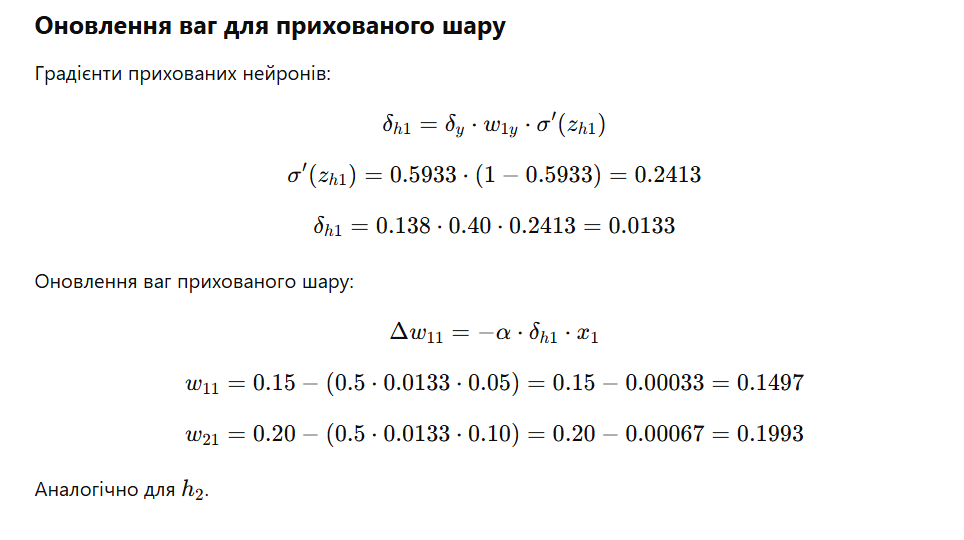
Обчислення помилки



Зворотнє поширення помилки







## **Повторення процесу**

Процес повторюється для кількох епох, поки мережа не досягне збіжності, тобто помилка не стане малою.

1. **Комп'ютерне бачення**

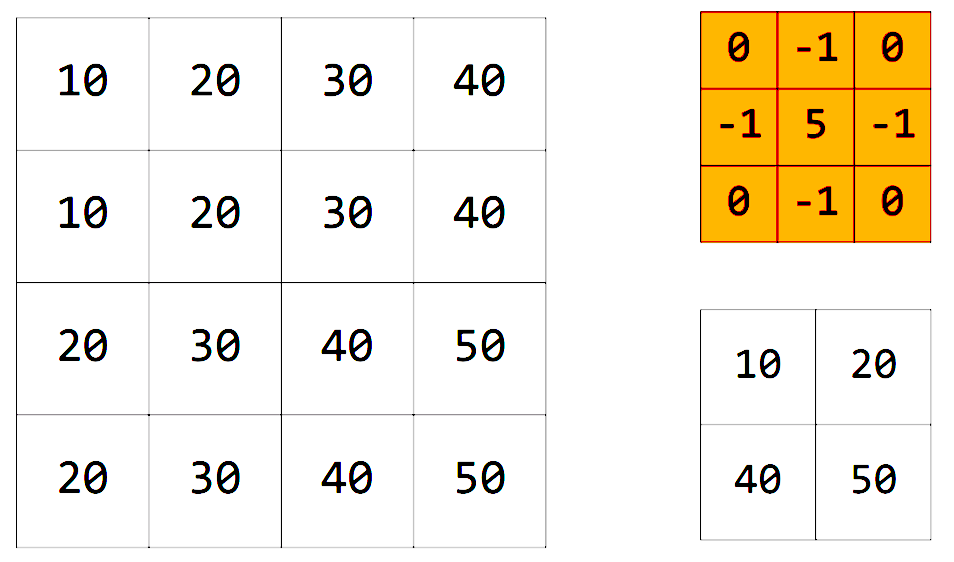
Зокрема, комп'ютерне бачення залучене в розпізнаванні обличь для автоматичного позначення людей на фотографіях у соціальних мережах. Прикладом також може бути зчитування рукописного тексту чи керування безпілотним автомобілем.

Зображення складаються з пікселів, пікселі своєю чергою представлені трьома значеннями від 0 до 255 — по одному для червоного, зеленого і синього. Ці значення зазвичай позначаються абревіатурою RGB. Це все можна використовувати для створення нейронної мережі, де кожне значення кольору будь-якого пікселя є входом із кількома прихованими шарами, а на виході є деяка кількість вузлів, які визначають, що саме зображено на фото чи картинці. Однак цей метод має певні недоліки. По-перше, якщо розкладати зображення на пікселі та значення їхніх кольорів, ви не зможете використовувати структуру зображення як орієнтир. Тобто, як люди, якщо ви бачите частину обличчя, ви розумієте, що напевне побачите й решту обличчя, і це прискорює обчислення. Непевне, ви хотіли б мати можливість застосовувати щось подібне до нейронних мереж. По-друге, кількість вхідних даних дуже велика, а це означає, що вам доведеться обчислювати багато вагових коефіцієнтів.

**Згортка (обробка зображень)**

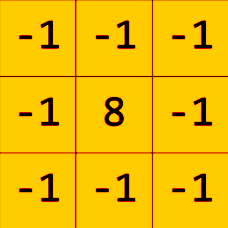
Згортка зображення — це використання фільтра, який додаватиме значення кожного пікселя зображення до його сусідів, з ваговим коефіцієнтом матриці-ядра. Це зміна зображення, яка може допомогти нейронній мережі обробити його.

Розглянемо такий приклад:



Ядром є жовта матриця, а зображенням — велика матриця ліворуч. Отримане відфільтроване зображення — це маленька матриця внизу праворуч. Щоби застосувати фільтр до зображення за допомогою ядра, можна почати з пікселя зі значенням 20 у верхньому лівому куті зображення (з координатами 1,1). Потім помножити всі значення навколо нього на відповідне значення ядра й додати їх (10\*0 + 20\*(-1) + 30\*0 + 10\*(-1) + 20\*5 + 30\*(-1) + 20\*0 + 30\*(-1) + 40\*0), отримаєте значення 10. Потім зробімо те саме для пікселя праворуч (30), пікселя під першим пікселем (30) і пікселя праворуч від нього (40). Так ви застосуєте фільтр до зображення зі значеннями, які бачите знизу праворуч.

Різні ядра можуть виконувати різні завдання. Для визначення контурів часто використовують подібне ядро:

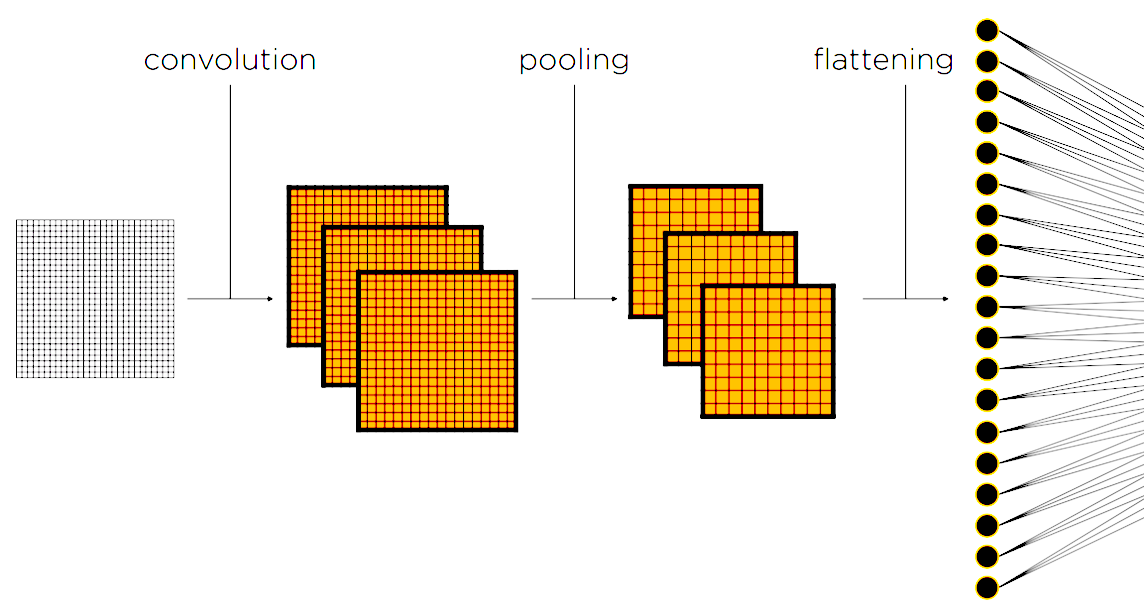


Ідея в тому, що коли піксель схожий на всіх своїх сусідів, вони мають компенсувати одне одного, щоби в результаті був 0. Відповідно, чим подібніші пікселі, тим темніша ця частина зображення, а чим більше вони відрізняються, тим світліша. Застосування цього ядра до зображення ліворуч призводить до отримання зображення з чітко окресленими межами (праворуч):



**Згорткові нейронні мережі**

Згорткова нейронна мережа — це нейронна мережа, яка використовує згортку, переважно для аналізу зображень. Спочатку вона застосовує фільтри, які можуть допомогти визначити деякі особливості зображення відповідно до різних ядер. Ці фільтри можна покращувати так само, як й інші вагові коефіцієнти нейромережі: налаштовувати їхні ядра, залежно від помилок на вході. Після цього отримані зображення агрегуються, а пікселі передаються на вхід традиційної нейронної мережі (цей процес називається **згладжуванням**).



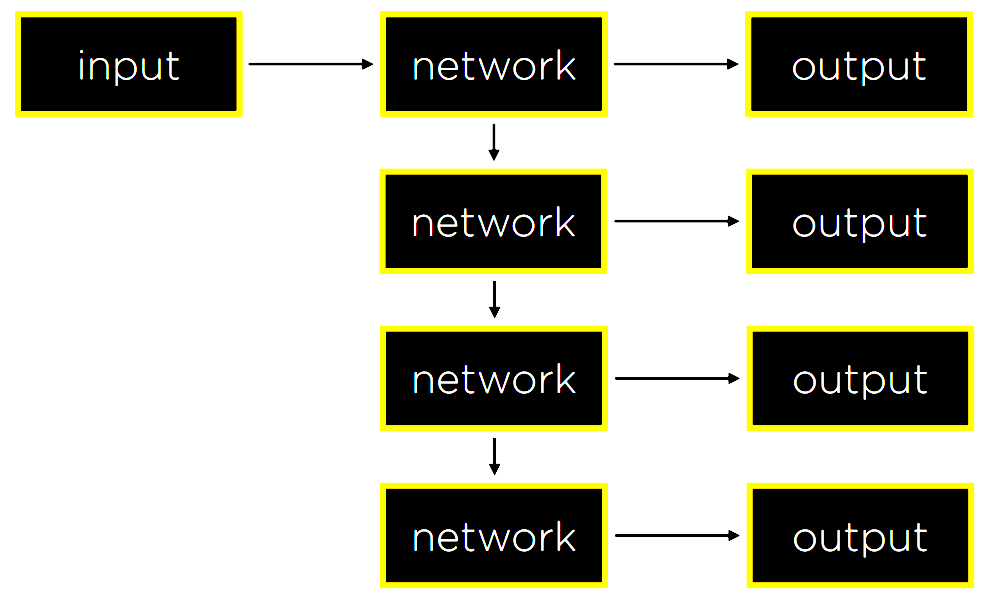
Згортку й агрегацію можна повторювати стільки, скільки буде необхідно, щоби виокремити додаткові особливості й зменшити кількість вхідних даних для нейронної мережі. У цьому полягає одна з переваг цих процесів — завдяки згортці та об'єднанню нейронна мережа стає менш чутливою до змін. Тобто, якщо одну й ту ж картинку сфотографувати під трохи різними кутами, вхідні дані для згорткової нейронної мережі будуть схожими, однак без згортки й агрегації вхідні дані кожного зображення будуть дуже різними.

1. **Рекурентні нейронні мережі**

Нейронна мережа прямого поширення — різновид нейронних мереж, які вже розглядалися в лекції, коли мережі надаються вхідні дані, з яких врешті-решт продукується певний вихід. Схема роботи нейронних мереж прямого поширення наведена нижче.



На відміну від них, **рекурентні нейронні мережі** містять нелінійні структури, тобто мережа використовує власний вихід як вхід. Так, наприклад, бот [captionbot](https://www.captionbot.ai/" \t "_blank)від Microsoft вміє складати слова в речення зі змістом зображеного. Від класифікації це відрізняється тим, що вихідні дані можуть бути різної довжини (яка залежить від характеристик зображення). Тоді як нейронні мережі прямого поширення не можуть змінювати кількість виходів, рекурентні нейронні мережі здатні на це завдяки своїй структурі. Наприклад, при створенні титрів мережа обробляє вхідні дані, щоб згенерувати вихідні, продовжує обробку даних з цього місця й створює ще один вихід, та повторює це стільки разів, скільки необхідно.



Рекурентні нейронні мережі корисні у випадках, коли мережа працює з послідовностями, а не з певним окремим об'єктом. Нейронна мережа вище мала створити послідовність слів. За цим же принципом можна аналізувати відеофайли, які складаються з послідовності зображень, або перекладати якісь матеріали, де послідовність вхідних даних (слова мовою оригіналу) обробляються для отримання послідовності вихідних даних (слів мовою перекладу).