Більше – не завжди краще

Оптимальний розмір апаратного забезпечення для ефективної роботи периферійного МН/ШІ

Йенн Лефау (Yann LeFaou), Microchip Technology
Переклад та редагування: Сергій Сокольський, PhD, ст. викл. каф. ПРЕ, РТФ, КПІ ім. І. Сікорського

Правильний підбір обладнання для балансу між продуктивністю, енергоспоживанням та вартістю дозволяє використовувати машинне навчання на платформах з обмеженими ресурсами. Із цієї статті ви дізнаєтесь, чому важливо не перевантажувати технічні рішення і як, маючи у своєму розпорядженні відповідне обладнання, IDE, інструменти та набори для розробки, фреймворки, набори даних та моделі з відкритим кодом, інженери можуть досить легко розробляти продукти з підтримкою МН/ШІ та обробкою на периферії.

Машинне навчання (МН) та штучний інтелект (ШІ, підвидом якого можна вважати МН) історично реалізовувалися на високопродуктивних обчислювальних платформах, а нещодавно — у хмарі. Однак зараз обидва методи дедалі частіше впроваджуються в додатках, де обробка виконується поряд з джерелом даних. Така периферійна обробка ідеально підходить для пристроїв Інтернету речей (IoT) і означає, що для аналізу потрібно надсилати менше даних у хмару. Переваги включають покращену продуктивність завдяки зменшенню затримки та підвищенню рівня безпеки.

МН/ШІ виводить периферійну обробку даних на новий рівень, роблячи можливим отримання висновків безпосередньо з джерела. Якщо дотримуватися цього прикладу, то це дозволяє пристрою Інтернету речей навчатися та вдосконалюватися на основі отриманого досвіду. Алгоритми аналізують дані, щоб знаходити закономірності та приймати обґрунтовані рішення трьома способами, якими машина (або ШІ) навчатиметься: навчання з учителем, навчання без учителя та навчання з підкріпленням.

Навчання з учителем (контрольоване навчання) передбачає використання машиною розмічених даних для тренування моделі. Наприклад, «розумну» камеру безпеки можна навчити за допомогою фото- та відеофайлів людей, які стоять, ходять, біжать або несуть коробки. До алгоритмів контрольованого машинного навчання належать логістична регресія та «наївний» алгоритм Байєса (*Naive Bayes*) зі зворотним зв'язком, необхідним для постійного вдосконалення моделі(-й), на основі якої(-их) будуть робитися прогнози.

Навчання без учителя (неконтрольоване навчання) використовує нерозмічені дані й алгоритми, такі як кластеризація K-середніх (*K Means Clustering*) або метод аналізу головних компонентів (*Principal Component Analysis*), для виявлення закономірностей. Це ідеально підходить для виявлення аномалій. Наприклад, у сценарії прогнозного технічного обслуговування або в медичній візуалізації, машина позначатиме ситуації або фрагменти зображення, які є незвичайними, на основі моделі «звичайного», яку машина створила та підтримує.

Навчання з підкріпленням (поглиблене навчання) — це процес «спроб і помилок». Як і при навчанні з учителем, потрібен зворотний зв'язок, але замість простого виправлення моделі, цей зворотний зв'язок сприймається як винагорода або покарання. До таких алгоритмів належать метод Монте-Карло та Q-навчання (*Q-learning*).

У наведених вище прикладах спільним елементом є вбудований технічний зір, що став «розумним» завдяки додаванню машинного навчання/штучного інтелекту. Будь-які інші програми також можуть отримати вигоду від отримання логічних висновків на основі технічного зору, що базується на джерелі даних. До того ж розумний вбудований технічний зір може використовувати частини спектра, невидимі для людського ока, такі як інфрачервоне (використовується для тепловізійного зображення) та ультрафіолетове випромінювання.

Надайте периферійній системі МН/ШІ інші дані, такі як показники температури та рівні вібрації від датчиків, і промислові пристрої Інтернету речей зможуть не лише відігравати значну роль у стратегії прогнозного обслуговування організації, але й забезпечувати раннє попередження про неочікувані збої, й таким чином допомагати захищати обладнання, продукцію та персонал.

ВБУДОВАНІ СИСТЕМИ

Як зазначалося на початку, МН/ШІ спочатку вимагали значних обчислювальних ресурсів. Сьогодні ж, залежно від складності алгоритмів, МН та ШІ можна реалізувати на компонентах, характерних для вбудованих систем (наприклад, IoT-пристроїв).

Наприклад, виявлення й класифікацію зображень можна реалізувати у

програмованих користувачем вентильних матрицях (*Field Programmable Gate Arrays, FPGA*) та на мікропроцесорних модулях (*Microprocessor Units, MPU*). Ба більше, відносно прості застосування, такі як моніторинг та аналіз вібрацій (наприклад, для прогнозного технічного обслуговування) — можуть бути реалізовані навіть на 8-бітному мікроконтролері (*Microcontroller Unit, MCU*).

Крім того, тоді як МН/ШІ спочатку вимагали висококваліфікованих фахівців з обробки даних для розробки алгоритмів розпізнавання образів — а також моделей, які могли б автоматично оновлюватися, забезпечуючи коректні прогнози — зараз це вже далеко не так. Інженери вбудованих систем, які вже знайомі з обробкою периферійних даних, тепер мають у своєму розпорядженні апаратне та програмне забезпечення, інструменти та методології для розробки продуктів на основі МН/ШІ. Крім того, багато моделей та наборів даних для тренування мережі доступні безплатно, а численні виробники мікросхем пропонують інтегровані середовища розробки (integrated development environments, IDE) та комплекси інструментів, що істотно пришвидшують створення МН/ШІ додатків.

Розглянемо, наприклад, MPLAB X IDE від Microchip. Це програмне забезпечення містить інструменти, які допомагають інженерам виявляти, налаштовувати, розробляти, налагоджувати та кваліфікувати вбудовані проєкти для багатьох пристроїв компанії. Плагін пакету розробки машинного навчання (*ML Development Suite*) дозволяє безпосередньо прошивати МН-моделі у цільове апаратне забезпечення. Цей пакет використовує так зване автоматизоване машинне навчання (*AutoML*), яке є процесом автоматизації багатьох трудомістких та ітеративних завдань, таких як розробка та навчання моделей (рис. 1).

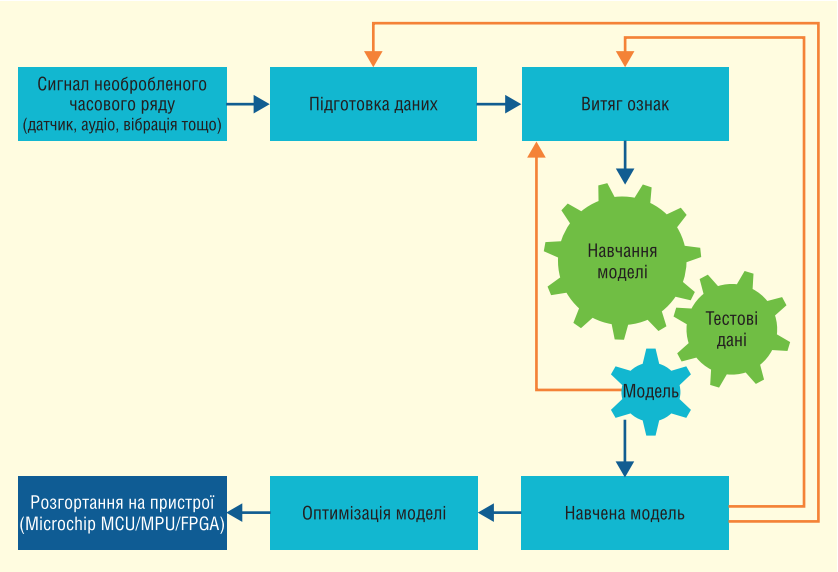

Рис. 1. Розробка моделей МН/ШІ — це ітеративний процес

Хоча ці ітеративні кроки можна автоматизувати, оптимізація проєктування — це зовсім інша справа. Навіть інженери, що мають досвід розробки програм для обробки периферійних даних, можуть мати труднощі з деякими аспектами МН/ШІ. Доведеться ухвалювати багато компромісних рішень між продуктивністю системи (яка значною мірою визначається розміром/складністю моделі та обсягом даних), енергоспоживанням і вартістю. Щодо останніх двох показників, на рисунку 2 зображено обернену залежність між (необхідною) продуктивністю та вартістю, а також наведено рівні енергоспо-

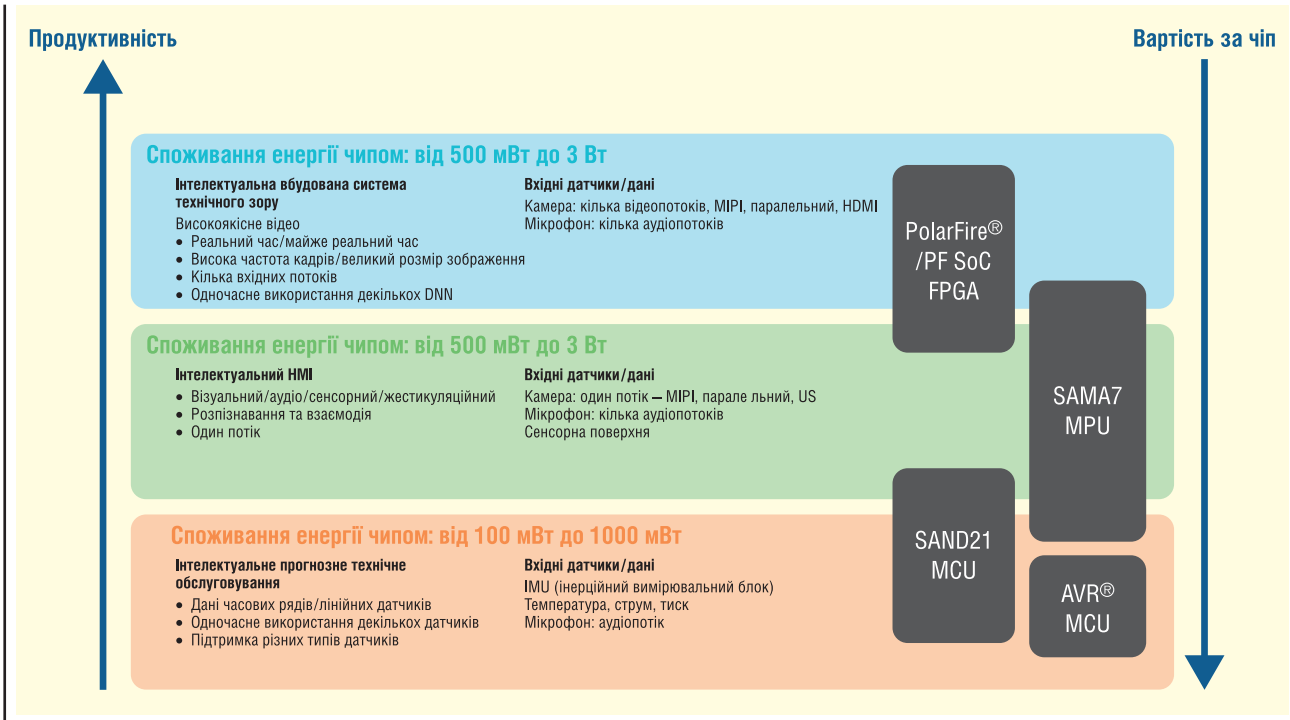

Рис. 2. Приклади типів пристроїв та застосувань машинного навчання, в яких вони зазвичай використовуються

Рис. 3. Структурна схема процесу машинного навчання

живання для типів пристроїв Microchip, що використовуються в типових додатках з машинним навчанням.

ПРОЄКТУВАННЯ ДЛЯ МАЛИХ ФОРМ-ФАКТОРІВ

Як уже згадувалося, навіть 8-бітні мікроконтролери можуть бути використані для деяких програм машинного навчання. Одним із факторів, що роблять це можливим і значно сприяють поширенню МН/ШІ у спільноті інженерів, є популярність tinyML, яка дозволяє запускати моделі на мікроконтролерах, що мають обмежені ресурси.

Переваги цього підходу можна побачити, порівнюючи вимоги традиційних МН/ШІ-платформ і tinyML. Високопродуктивні мікроконтролери або мікропроцесори для програм МН/ШІ зазвичай працюють на частотах 1–4 ГГц, потребують 512 МБ – 64 ГБ оперативної пам'яті й використовують 64 ГБ – 4 ТБ енергонезалежної пам'яті для зберігання. Вони також споживають 30–100 Вт енергії. Натомість tinyML призначене для мікроконтролерів, що працюють у діапазоні частот 1–400 МГц, мають 2–512 КБ оперативної пам'яті та використовують 32 КБ – 2 МБ енергонезалежної пам'яті для зберігання даних. Споживана потужність зазвичай становить 150 мкВт – 23.5 мВт, що ідеально підходить для пристроїв, які живляться від батарей, значною мірою залежать від зібраної енергії або живляться за допомогою індукції.

Ключові етапи впровадження tinyML — це збір і підготовка даних, а також генерація й оптимізація моделі. Серед них збір і підготовка даних є критично важливими, якщо ми хочемо забезпечити наявність релевантного набору даних на всіх етапах процесу МН (рис. 3).

Для тренування необхідні датасети в системах із контрольованим навчанням (та із частковим контролем). Тут датасет — це колекція даних, структурованих у певному порядку. Як було зазначено раніше, для контрольованих систем дані вже розмічені. Тому, повертаючись до прикладу зі «смарт»-камерою безпеки, машина тренувалася б на фотографіях людей, які стоять, ходять чи біжать (та інших ситуаціях). Датасети можуть бути створені вручну, але багато з них комерційно доступні, наприклад, MPII Human Pose, який містить близько 25 000 зображень, отриманих з онлайнових відео.

Однак набор даних обов'язково потрібно оптимізувати. Надмірна кількість даних швидко заповнить пам'ять. Занадто мала — і машина або не зможе робити передбачення, або робитиме прогнози, які будуть помилковими чи хибними.

Крім цього, модель МН/ШІ повинна бути невеликою. У цьому відношенні одним із популярних методів стиснення є обрізання ваг нейронних зв'язків (*Weight Pruning*), коли вага деяких з'єднань усередині моделі встановлюється в нуль, що дозволяє моделі не враховувати її під час отримання висновків. Можливе також «обрізання» й нейронів. Іншим методом стискання є квантування, що знижує точність параметрів, зміщень і активацій моделі, перетворюючи дані з формату високої точності, наприклад, 32-бітний з рухомою крапкою (FP32), у формат нижчої точності, скажімо, 8-бітне ціле число (INT8).

За допомогою оптимізованого набору даних та компактної моделі можна вибрати відповідний мікроконтролер. Для спрощення цього процесу існують спеціальні фреймворки. Наприклад, TensorFlow дозволяє створити нову модель або перенавчити ту, що вже існує. Потім модель можна стиснути в плоский буфер (*Flat-buffer*) за допомогою конвертора TensorFlow Lite Converter, завантажити на цільовий пристрій та провести квантування.

ПІДСУМОК

Машинне навчання та ШІ використовують алгоритмічні методи для виявлення закономірностей і тенденцій у даних та здійснення прогнозу. Розміщуючи МН/ШІ безпосередньо біля джерела даних (на периферії), додатки можуть робити висновки, приймати рішення та вживати заходів на місці та в режимі реального часу, що робить систему в цілому ефективнішою (з погляду продуктивності та енергоспоживання) та безпечнішою.

Завдяки наявності відповідного обладнання, IDE, інструментів і наборів для розробки, фреймворків, датасетів та моделей з відкритим кодом, інженери можуть порівняно легко створювати продукти з МН/ШІ та периферійною обробкою даних.

Це захопливий час для інженерів вбудованих систем і всієї галузі загалом. Однак важливо уникати надмірної інженерної складності. Надто багато часу й коштів можна витратити на розробку додатків, що потребують мікросхем із більшими ресурсами, ніж необхідно, а отже вони споживають більше енергії та мають вищу вартість. CN