Transformer Models for Image Processing

臺大 生醫電資所一 R11945072 張柏彥

ADSP
Final Project

一、簡介

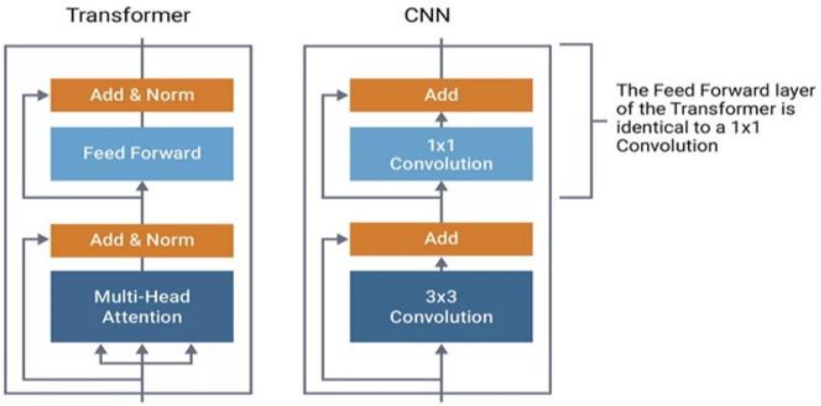
(一) 圖像處理 (Image Processing) 為何重要?

圖像處理是計算機視覺 (Computer Vision) 領域的一個重要組成，它涵蓋了從圖像分類到對圖像檢測等多種任務，此種任務在我們的生活中扮演了舉足輕重的角色，例如，自動駕駛汽車需要對圖像檢測和語義分割來理解周圍環境，醫療圖像分析需要圖像分類和分割來幫助特定病灶做疾病診斷，社交媒體平台需要圖像識別以便自動標記和過濾內容等。

過去，這些事務主要依賴卷積神經網絡 (CNN) 來完成。CNN 通過對局部圖像區域進行操作，並利用空間結構信息進行特徵學習，已經取得了顯著的效果。然而，單純使用 CNN 模型始終存在一些侷限性，例如，其對全局上下信息的處理能力有限，難以捕捉圖像中的距離依賴等。

(二) 為何需要使用到 Transformer model?

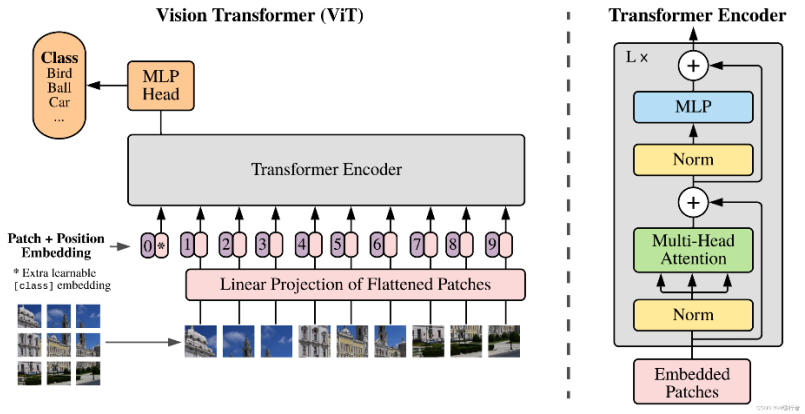
卷積神經網絡難以同時考慮上下文內容做信息處理之特性，即造成了模型對於結果輸出的侷限，而這正是 Transformer 模型發揮作用的地方。透過比較得知(圖一)，Transformer 模型引入的自注意力機制 (Self-Attention) 可以更好地把握全局上下信息以及長距離依賴關係，且能並行處理而使其具有更高的計算效率。



CNN v.s. Transformer — 多了Self-Attention架構

Transformer 模型最初是為 NLP 任務設計的，如現今最熱門的 BERT、GPT 等廣泛使用的語言模型。然而近年來，研究人員已經開始探索其在計算機視覺任務中的潛力。特別是 2020 年 Google 的研究團隊提出的 Vision Transformer (ViT) (圖二)，它首次將 Transformer 模型直接應用於圖像類任務，首次取得了出色的效成果。至此之後，探索 Transformer 模型在圖像處理中的應用，並研究如何更

好地利用 Transformer 模型來解決圖像處理任務，也可以為我們提供一種全新的、強大的視覺解析工具，可以使研究人員或開發工程師進一步提升模型的性能對許多生活中的實際應用都得以發揮重大意義。



(三) Tutorial 的目標和內容概述

1. Tutorial 的目標：

學習和理解 Transformer 模型的基礎知識和工作原理，並進一步探索其在圖像處理中的應用，包括圖像分類、對象檢測等任務，再應用基於 Transformer 的圖像處理模型。最後，了解 Transformer 模型在計算機視覺領域的發展和挑戰。

2. Tutorial 的內容概述：

(1) 神經網絡到 Transformer 模型的基礎：
Transformer 模型工作原理，包括自注意力機制、位置編碼、多頭注意力等。

(2) Transformer 在圖像處理中的應用：
討論 Transformer 在圖像處理中的應用，重點介紹 Vision Transformer，以及其在圖像分類、對象檢測等任務中的應用。

(3) Coding 實作範例：
通過實際的代碼示例，教授如何使用 Python 和深度學習框架(如 TensorFlow 或 PyTorch)來實現和訓練基於 Transformer 的圖像處理模型。

(4) 最新發展和挑戰：
探討 Transformer 模型在圖像處理中的最新發展，包括最新的研究進展和模型，如 Swin Transformer 等。同時，討論 Transformer 模型在應用過程中可能遇到的挑戰，如計算資源需求、訓練數據需求等。

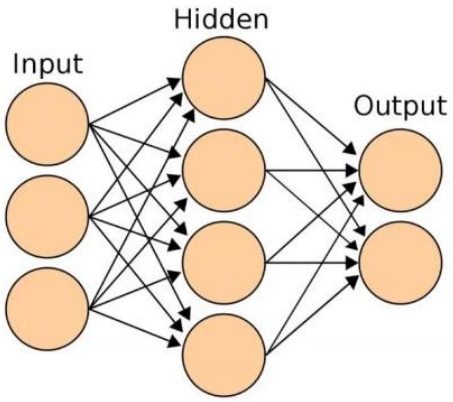
(5) 未來展望：
對 Transformer 模型在圖像處理中的未來發展進行預測和展望。

二、深度學習與圖像處理基礎

(一)、神經網絡的基本知識

人工神經網絡（Artificial Neural Networks，ANNs）是深度學習的核心組成部分，它們的設計靈感源於我們對人類大腦的理解。在生理解剖學上，神經網絡由大量的處理單元組成，這些單元被稱為「神經元」或「節點」。這些神經元被組織成許多層，每一層都與上一層和下一層的神經元相互連接。

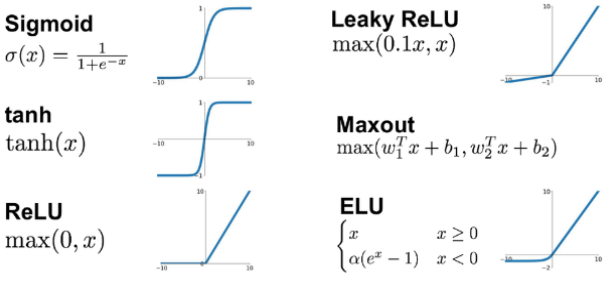
在一個基本的神經網絡中，我們可以大致區分三種類型的層 (圖四)：



(圖四) 神經網絡層示意圖

1. 輸入層 (Input Layer)：這是神經網絡接收資訊的地方。每個節點對應於我們要處理的資料的一個特徵。例如，在圖像處理中，每個像素可能作為一個輸入節點。
2. 隱藏層 (Hidden Layer)：這些層位於輸入層和輸出層之間。每個節點在這裡都會根據與之相連的上一層的節點和對應的權重來計算其值。
3. 輸出層 (Output Layer)：這是網絡的最終層，對應於我們希望神經網絡預測的結果。例如，在分類問題中，輸出層的每個節點可能對應於一個可能的標籤或類別。

在這些層中，資訊是從輸入層開始，通過隱藏層，最終到達輸出層。這種前向傳播（Feedforward）的過程涉及到將資訊加權並通過激活函數（Activation Function）傳遞。激活函數的選擇可以影響神經



(圖五) 激活函數 (Activation Function) 示意圖

網絡的性能和學習能力，常見的激活函數有 ReLU（Rectified Linear Unit）、sigmoid 和 tanh 等 (圖五)。

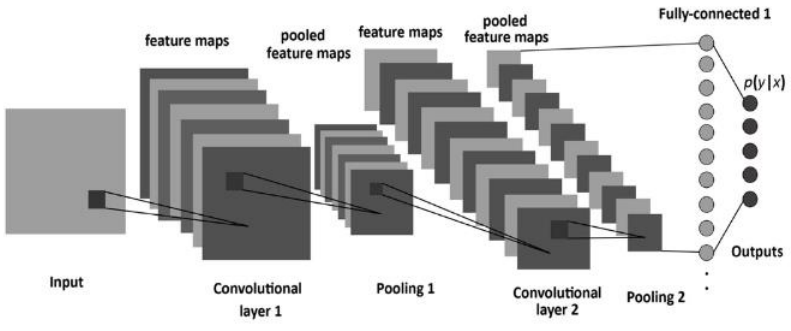
在這個過程中，神經網絡的目標是透過學習找出最佳的權重集，這可以最大程度地降低預測結果和實際標籤之間的誤差。1986 年，Rumelhart、Hinton 和 Williams 引入了一種稱為反向傳播（Backpropagation）的學習算法，使得深度神經網路的訓練允許根據預測結果的錯誤來調整權重，從而進行自我改進。它首先計算出預測錯誤，然後將這些錯誤分散到每個權重上，並根據錯誤來調整權重。這個過程會在多個資料樣本和多個學習周期（常稱 Epoch）上重複進行，直到其收斂到最佳狀態。

人工神經網絡的威力來自於它們的彈性和學習能力。隨著足夠的訓練資料和計算能力，它們可以學習從圖像到語音，再到文本的各種複雜模式。然而，它們也經常需要克服許多資源限制，例如需要大量的資料、計算資源來做訓練。此外，它們也有可能「過擬合」訓練資料（常稱 Overfitting），使得在未見過的新資料上的預測效果變差。在接下來的部分，我們將簡易討論神經網絡在圖像處理的一種最常見類型－卷積神經網絡（Convolutional Neural Networks，CNN）。

(二)、CNN 的工作原理和在圖像處理中的應用

卷積神經網絡（Convolutional Neural Networks，CNN）是一種常見的深度學習模型，它對圖像和網格結構（例如時間序列）的數據處理特別有效。CNN 的靈感來自生物視覺系統，並試圖模仿人類大腦的視覺識別過程。

一個基本的 CNN 由一系列的層組成 (圖六)：卷積層（Convolutional Layers）、激活層（Activation Layers）、池化層（Pooling Layers），並以全連接層（Fully Connected Layers）結尾，進行最終的分類或回歸任務。這些層通過特定的方式相互連接，形成了多層結構，得以自動從原始數據中提取出有意義的特徵。



(圖六) CNN 架構圖

在卷積層中，一個或多個卷積核（Convolutional Kernels，或稱為過濾器 Filters）在輸入圖像上滑動，並與其覆蓋的像素依序進行點積運算。這個過程可以捕獲到局部的視覺特徵，如邊緣、線條、角等。這些特徵隨後會通過非線性激活函數（如 ReLU）進行轉換。

池化層則負責減小卷積層產生的特徵圖的維度，這通過將特徵圖的一小塊區域（如 2x2 的窗口）壓縮為一個值來實現（例如，取最大值或平均值等）。此過程有助於使模型對小的平移或變換保持不變，並且大幅減少後續層積的計算量。

最後，全連接層將池化層的輸出看作是一維向量，並進行最終的分類或回歸任務。在分類任務中，經常會使用 Softmax 函數來做每個類別輸出的預測概率。

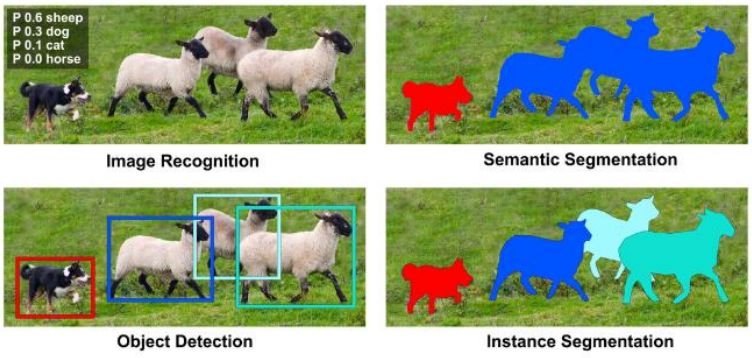
CNN 在圖像處理中的應用範圍非常廣泛，包括：

1. 圖像分類（Image Classification）： CNN 最為人所知的應用，其中最著名的案例可能是 ImageNet 挑戰賽。模型的目標是將輸入圖像分類到預定義的類別中（例如，分辨一張圖像中是否包含一隻貓）。
2. 對象檢測（Object Detection）：在對象檢測任務中，模型不僅需要識別圖像中的對象，還需要定位這些對象。著名的 CNN 如 Faster R-CNN 和 YOLO 等都被廣泛用於這種任務。
3. 義分割（Semantic Segmentation）：在圖義分割任務中，模型需要為圖像中的每個像素指定一個類別標籤。例如，給出一張城市街景的照片，模型需要識別出人行道、建築、汽車等各種對象。

總的來說，由於 CNN 強大的特徵提取能力，其以成為圖像處理領域的重要工具，並於醫療影像分析、自駕車技術、視頻處理等各種領域得到了廣泛的應用。

(三)、深度學習在圖像處理的常見任務：分類、檢測、分割等

深度學習在圖像處理中的應用非常廣泛，以下是幾種常見的任務類型 (圖七、八)：

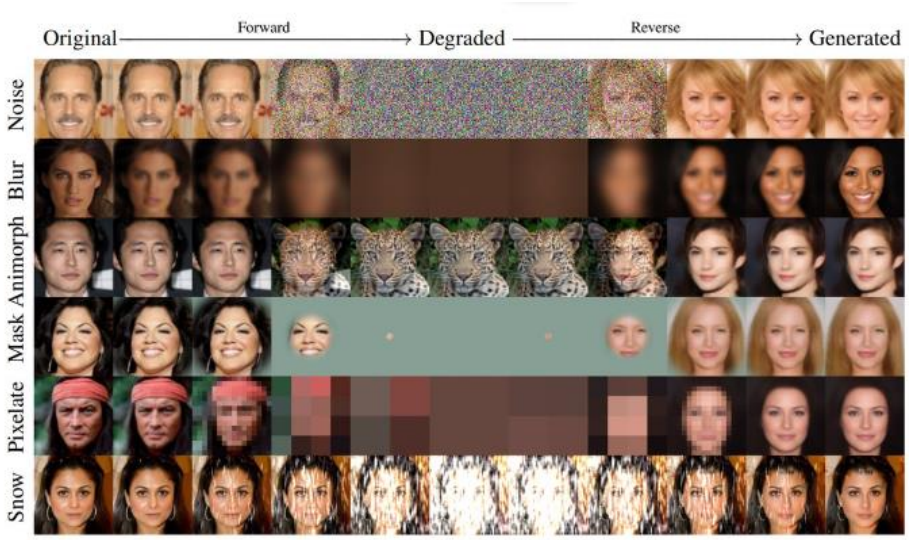


(圖七) 1.~4. 的分類差異

1. **圖像分類（Image Classification）**：圖像分類可能是最直觀且最常見的圖

像處理任務。在此任務中，模型的目標是將輸入圖像分類到一組預定義的類別中。例如，模型可能需要區分一張圖像是貓、狗還是人。這種任務的典型應用包括影像辨識，如臉部識別或疾病診斷等。

2. **對象檢測（Object Detection）**：對象檢測比單純的圖像分類更為複雜，因為它不僅需要識別圖像中的對象，還需要精確地定位這些對象。在此任務中，模型需要對每個被識別的對象輸出一個邊界框和一個類別標籤。對象檢測的應用範圍很廣，包括臉部檢測、行人檢測、車輛檢測等。
3. **語義分割（Semantic Segmentation）**：語義分割是一種更高級的圖像處理任務，模型需要為圖像中的每個像素指定一個類別標籤。換句話說，這不僅涉及到識別圖像中存在的對象，還涉及到確定這些對象在圖像中的確切位置。語義分割的應用包括自駕車、醫療影像分析等。
4. **實例分割（Instance Segmentation）**：實例分割是語義分割的一種擴展，不僅需要對每個像素進行分類，還需要區分不同的對象實例。例如，在一張包含多只貓的圖像中，實例分割不僅需要識別出所有的貓，還需要區分每一只貓。
5. **圖像超解（Image Super-Resolution）**：過去於訊號處理領域也有眾多研究，在深度學習模型中，也可以將低分辨率的圖像轉換為高分辨率的版本。這種技術可以用於老照片修復、視頻增強等，。
6. **圖像生成（Image Generation）**：目前產業界最夯的領域之一。在這種任務中，深度學習模型被用來生成新的圖像，可以是從頭開始創建，也可以是基於現有圖像進行修改。這種任務的典型應用包括生成對抗網絡（Generative Adversarial Networks, GANs），以及近期更熱門的基於 Diffusion Model（圖八）的 DALLE, Midjourney 等，多用於藝術創作、圖像翻譯（例如，將日間的風景轉換為夜間風景）等。



(圖八) 圖像生成，此為Diffusion Model示例

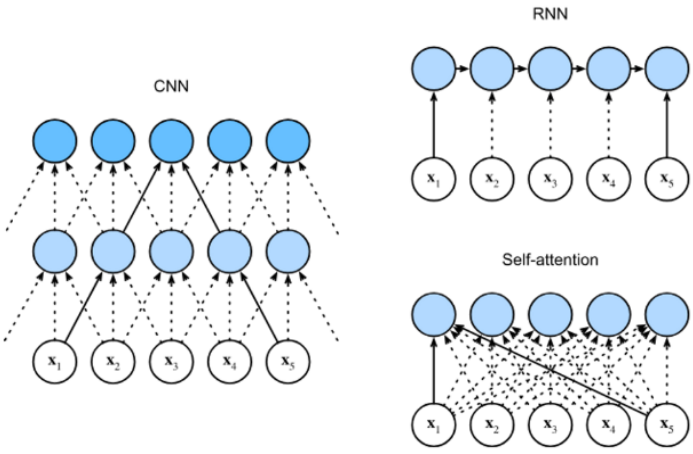
三、Transformer 模型的基礎

(一)、Transformer 的起源和背景

2017 年，Google 提出了一種新的機器學習模型概念—Transformer。該模型最初是由 Vaswani 等人在論文"Attention is All You Need"中提出而備受矚目，這種模型引入了一種全新的機制——自注意力機制（Self-Attention Mechanism），以在序列輸入中仍能考慮上下輸入之間的關係，進而針對同一輸入做出多種的輸出判斷。模型在許多自然語言處理（Natural Language Processing, NLP）任務上取得了當時最好的性能。

在這之前，自然語言處理的研究主要依賴於遞歸神經網絡（Recurrent Neural Networks, RNN）和其變型，例如長短期記憶網絡（Long Short-Term Memory, LSTM）和 GRU（Gated Recurrent Unit, 無譯名）。然而，這些模型因仍存有包含長期依賴問題（long-term dependencies）和難以平行計算等問題，使得模型訓練仍遭遇瓶頸。

為了解決這些問題，Transformer 模型應運而生。該模型棄用了 RNN 的結構，利用自我注意力機制（self-attention mechanism）來處理序列數據 (比較一圖九)。這種設計使得 Transformer 可以在處理長序列時獲得更好的性能，並且充分利用現代硬體平台的平行計算能力，大大提高訓練效率。其獨特的性能和靈活性使其成為了自然語言處理的一種新的主流模型。該模型已被廣泛應用於翻譯、文本生成、摘要生成等多種 NLP 任務，並掀起 NLP 模型架構的革新。



(圖九) CNN、RNN 到 Self-Attention 之間的結構關係比較

從首次提出以來，Transformer 模型不斷在架構上進行擴展，進而產生了許多衍生模型，例如 BERT、GPT 和 T5 等，在各種 NLP 任務上取得了優異的表現。現今，Transformer 架構及其變型已主宰當今大多數最先進的 NLP 模型。

(二)、Transformer 的結構和工作原理

1. Transformer 模型結構

Transformer 模型主要由兩大部分組成：編碼器（Encoder）和解碼器（Decoder）。每個部分都由多層的 Transformer 層堆疊組成，並且每一層又包括兩個子層結構：多頭自我注意力機制（Multi-Head Self-Attention Mechanism）和全連接前饋神經網路（Fully Connected Feed-Forward Neural Network）。每個子層結構之後都跟著一個殘差連接（Residual Connection）和層正規化（Layer Normalization）。

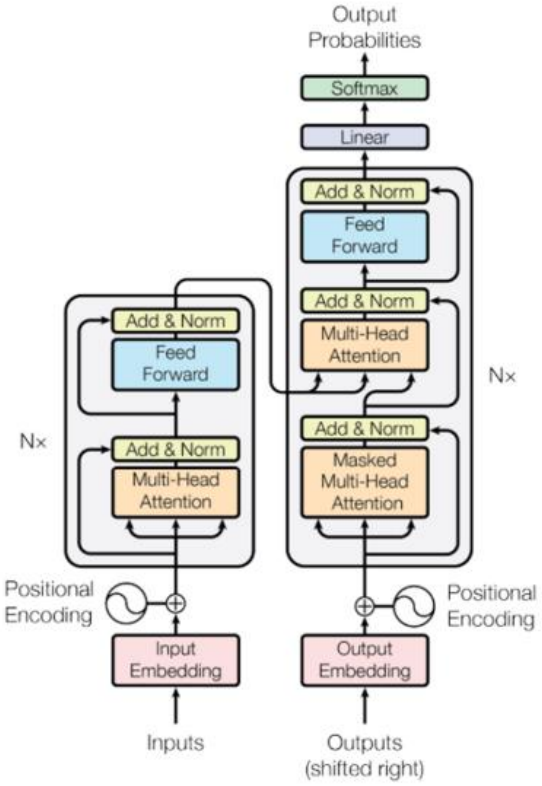


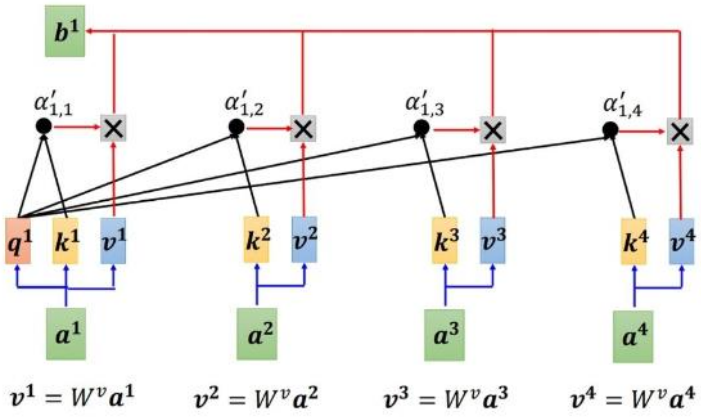
Figure 1: The Transformer - model architecture.

(圖十) 左- Encoder 右-Decoder / 右中多一層Encoder-Decoder Attention層

首先，在編碼器部分，輸入的數據通過多頭自我注意力層，這層可以對輸入數據中的每個序列（在 NLP 中通常是單詞或者子詞）的上下文關係進行編碼。接著，數據會被送到全連接的前向神經網路中做進一步的轉換；解碼器部分，結構與編碼器部分類似，但多了一個稱為"編碼器-解碼器注意力層"（Encoder-Decoder Attention Layer）的子層。這層可以聯繫 Decoder 的輸入和 Encoder 的輸出，讓兩者做一次注意力計算，使模型權重分配到與當前解碼輸入最相關的編碼器輸出。如此，如果解碼器正在嘗試生成一個句子的下一個詞，那麼這一層可能會讓模型更多地關注到 Encoder 所有輸出中與詞語境相關的部分。

2. Transformer 模型工作原理——自注意力機制、位置編碼

其中，在 Transformer 的工作過程中，Self-Attention Mechanism 扮演了重要的角色。在這個機制下，模型不僅可以將注意力集中到當前的輸入元素，還可以考慮到前後、上下的輸入元素，做出整體性的考量。更細節地說，每個輸入元素都有一個可變動的「權重」，這個權重能有效決定該元素在輸出時的重要程度。而這些權重是通過同時計算所有輸入元素之間的相關性（經常使用 Dot Product 點積來計算）來綜合得到的 (見圖十一)。



(圖十一)、自注意力機制、各個元素點積與相加構成輸出

另一個關鍵的部分是位置編碼（Positional Encoding）。由於 Transformer 並不像 RNN 那樣具有明確的序列處理能力，因此需要通過位置編碼來為模型提供序列中元素的順序信息。位置編碼可以是學習得來的，也可以是固定的。在 "Attention is All You Need"論文中，作者使用了一種基於正弦和餘弦函數的固定位置編碼。

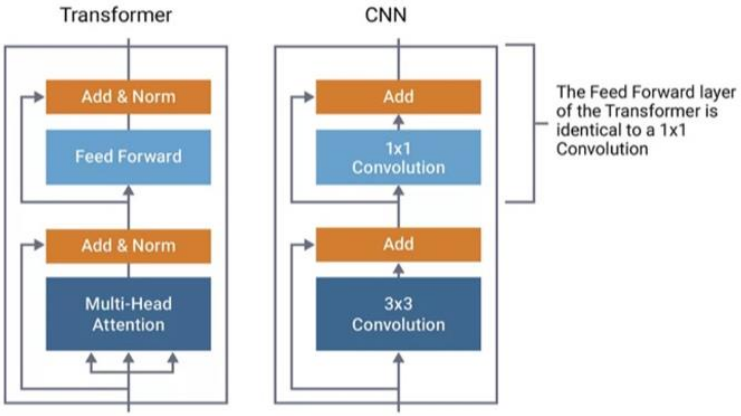
(三)、Transformer 與傳統 CNN 模型的比較

Transformer 和卷積神經網絡（Convolutional Neural Networks，CNN）都是深度學習模型，並且在許多任務中都表現出色，但是它們在設計理念、處理方式和適用場景上都存在著不少明顯的區別。

設計架構與限制——CNN

首先，從設計理念上來看，CNN 由於其卷積操作的特性，尤其擅長處理具有空間結構性的數據，例如圖像。其透過卷積核（Convolutional Kernels）在輸入上滑動，從而識別出局部的特徵，這些局部特徵可以逐層堆疊，形成更抽象的全局特徵。因此，CNN 在計算機視覺領域有著廣泛的應用。

然而，這也限制了 CNN 的某些能力。例如，對於需要長距離依賴（Long-distance Dependencies）的問題，也就是當需要捕捉上下文之間的關係做綜合判斷時，CNN 可能會表現得不夠理想。這是因為 CNN 的卷積核大小固定，只能捕捉到有限範圍內的依賴關係。



CNN v.s. Transformer — 多了Self-Attention架構

設計架構與限制——Transformer

相反地，Transformer 是基於自我注意力機制（Self-Attention Mechanism）的，它可以根據各元素之間的相關性，賦予不同的權重，無論元素間的距離多遠。這使得 Transformer 特別適合於處理需要捕捉長距離依賴的問題，例如機器翻譯、文本生成等自然語言處理任務。

然而，隨著模型如 Vision Transformer（ViT）的出現，我們發現 Transformer 也能夠在處理圖像這種具有空間結構性的數據上表現出色。這部分原因是因為 Transformer 的自我注意力機制能夠捕捉到圖像中的全局上下文關係。

計算資源 (複雜度比較)

在計算需求資源方面，由於 Transformer 的自我注意力機制需要計算所有元素之間的關係，因此其計算和內存需求隨著輸入大小的增長而急劇增長，尤其是在處理長序列時；相較之下，由於 CNN 的卷積操作只關注局部特徵，其計算需求相對較小，更適合於處理大規模圖像。

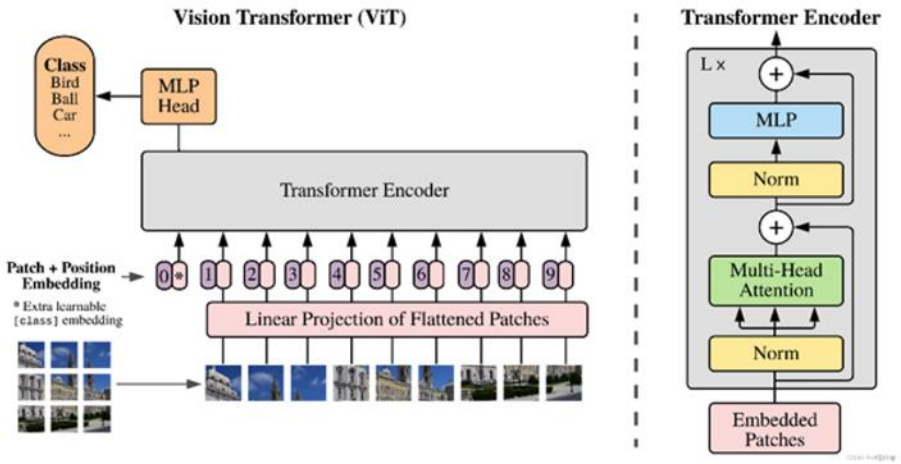
總的來說，這兩種模型在不同的場景和任務中各有優勢，實際模型的選擇必須大幅取決於具體的應用需求。

四、Transformer 在圖像處理中的應用

(一)、Vision Transformer (ViT)

Vision Transformer (ViT) 是一種基於 Transformer 架構的模型，專門用於處理視覺任務，特別是圖像分類。在眾多基於卷積神經網絡（Convolutional Neural Networks，CNN）的圖像處理模型中，ViT 的出現提供了一種新的思路。

ViT 的主要工作方式是將輸入的圖像分割成一系列的小塊或稱為「圖塊」（patches）。每個圖塊會被視為序列中的一個元素，並且這些圖塊會被展平並輸入到一個線性轉換層（linear transformation layer），該層將每個圖塊轉換為一個固定維度的向量。然後，這些向量會被送入標準的 Transformer 編碼器結構進行處理 (圖十二)。



(圖十二) ViT 模型架構圖—含區塊切分、自注意力編碼器

由於 ViT 並沒有依賴任何卷積操作，因此其主要是透過自注意力模型的優勢，得以使其在全局上下文中進行特徵學習，對比傳統的 CNN 局限性有很大的躍進。也就是說，ViT 能夠學習到圖像的各個部分之間的關聯性，並且這種關聯性的學習並不依賴於他們的空間位置或鄰近性。

然而，由於 ViT 的自我注意力機制需要考慮所有圖塊間的配對關係，因此在計算上它相對於 CNN 來說昂貴許多，特別是當輸入圖像的分辨率極高時。此外，ViT 對大量訓練數據的需求也比傳統的 CNN 要高，這限制了其在有限數據集上的應用。

總的來說，Vision Transformer 為視覺任務創造了一種新的概念與模型，並在一些標準視覺任務表現上，得以匹配甚至超越最先進的 CNN 模型的性能。

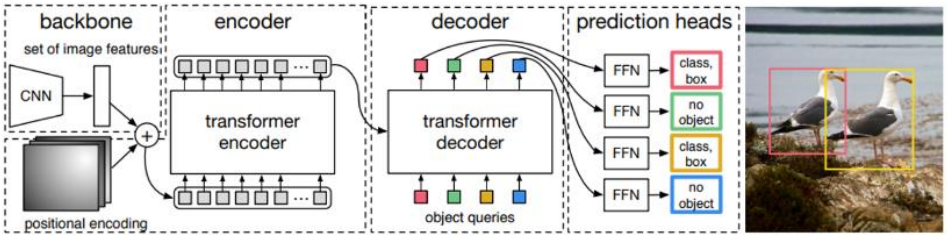
(二)、Transformer 在分類、檢測、分割任務中的應用

1. 圖像分類（Image Classification）：

上文所述，Vision Transformer (ViT)在圖像分類任務中展現了強大的性能。ViT 將輸入圖像分割成一系列的圖塊（patches），然後將這些圖塊作為序列輸入到 Transformer 中。這種設計允許模型捕捉到圖像各個部分間的全局關聯性，從而有效地學習到更具辨識力的特徵來進行圖像分類。

2. 物件偵測（Object Detection）：

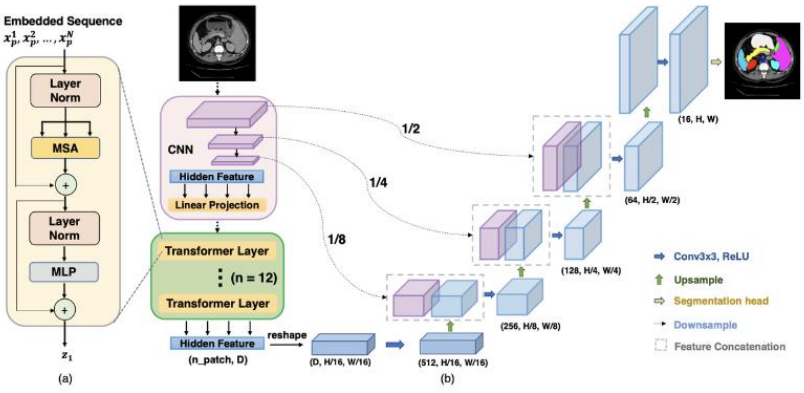
在物件偵測任務中，Facebook 開發了一種名為 DEtection TRansformer (DETR)的模型使用了 Transformer 的架構。DETR 結合了 CNN 和 Transformer 的特點，利用 CNN 提取圖像特徵，然後使用 Transformer 處理特徵序列以進行物體檢測。DETR 的優勢在於，它能夠直接輸出物體的邊界框和類別，無需對檢測任務進行複雜的手工設計和後處理。



(圖十三)、DETR模型架構圖—物件偵測

3. 語義分割（Semantic Segmentation）：

在語義分割任務，一種名為 TransUNet 的模型將 Transformer 和 U-Net 結合在一起。該模型首先利用 ViT 將圖像分割成一系列的圖塊，並從這些圖塊中提取特徵。然後，利用 U-Net 的上採樣（Upsampling）機制將這些特徵映射回像素級別的分割結果。這種方法有效地結合了 Transformer 在建模全局上下文關係方面的優勢和 U-Net 在處理局部細節方面的強大能力。



(圖十四)、TransUNet模型架構圖—語義分割

(三)、Transformer 在圖像處理的優勢和挑戰

1. 優勢：

(1) 全局視野（Global View）：

傳統的 CNN 通過滑動窗口的方式從局部視野提取特徵，而 Transformer 的自注意力機制可以在全局範圍內識別相關特徵，無需人工設定窗口大小形狀。

(2) 序列化輸入（Sequential Input）：

與 CNN 相比，Transformer 對輸入的順序更為敏感，使其可以處理一系列圖像或視頻中的時間序列數據。

(3) 靈活性和擴展性（Flexibility and Scalability）：

Transformer 架構可以被輕易地擴展或縮小，以應對不同的計算能力和任務。

2. 挑戰：

(1) 計算效能（Computational Efficiency）：

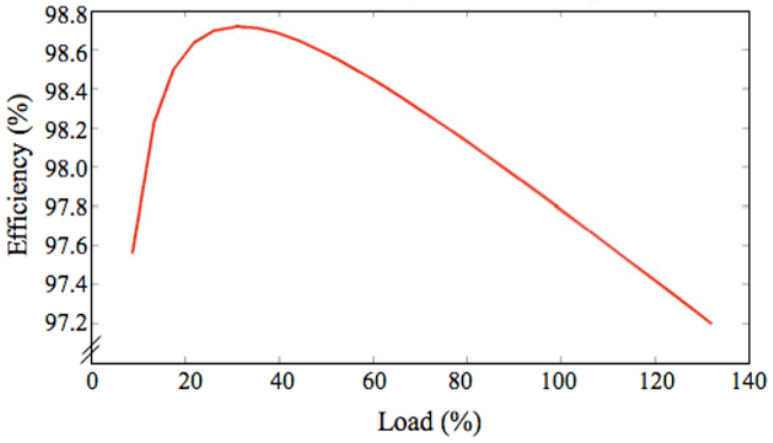
由於自注意力機制需要對每一對輸入元素進行計算，當輸入的圖像分辨率非常高時，會使 Transformer 模型效率下降、需承擔更大的計算挑戰。

(2) 訓練數據需求（Data Requirements）：

Transformer 模型通常需要大量的訓練數據以達到好的效果。對於有限的訓練數據，或者數據分佈不均的圖片任務，Transformer 較難達到理想的性能。

(3) 欠完備的位置感知能力（Imperfect Positional Awareness）：

儘管 Transformer 模型通過位置編碼解決了不具有固有位置感知能力的問題，但這種方法可能並不足以捕捉到所有圖片類型的空間關係。



(圖十五)、Transformer模型—Loading大時效率大幅下降

五、Transformer 在圖像處理中的應用

實作(一)、利用 Transformer 進行圖像分類

在這個實例中，我們將介紹如何使用 PyTorch 實現 Vision Transformer (ViT)來進行圖像分類任務。首先，我們先透過在 Terminal 使用 pip 完成安裝所需的套件。

```
pip install torch torchvision
pip install transformers
```

接著，我們可以從 Hugging Face 的 Transformers 函式庫中載入我們的模型。我們將使用預訓練的 ViT 模型，並在我們的數據集上微調它：

```
from transformers import ViTFeatureExtractor, ViTForImageClassification
import torch

feature_extractor = ViTFeatureExtractor.from_pretrained('google/vit-base-patch16-224')
model = ViTForImageClassification.from_pretrained('google/vit-base-patch16-224')

inputs = feature_extractor(images, return_tensors="pt")
outputs = model(**inputs)
logits = outputs.logits
predicted_class_idx = logits.argmax(-1).item()
```

在上面的代碼中， images 應該是一個 list，其中包含你的圖像。這些圖像可以是 NumPy 數組，也可以是 PIL 圖像。在 model(**inputs)這一行，模型將輸入的圖像進行分類，並返回一個包含 logits 的輸出。然後，我們可以通過找到最大的 logit 來確定預測的類別索引。

在微調模型之前，我們需要定義我們的數據集和數據加載器。你可以使用 PyTorch 的 ImageFolder 或 Dataset 來實現這一點，並且在微調期間，我們需要適當的資料擴增（Data Augmentation）：

```
from torchvision import transforms
from torchvision.datasets import ImageFolder
from torch.utils.data import DataLoader

# 資料擴增
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
```

```
])

# 定義數據集
train_dataset = ImageFolder(root='path_to_your_training_data', transform=transform)
val_dataset = ImageFolder(root='path_to_your_validation_data', transform=transform)

# 定義數據加載器
train_dataloader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_dataloader = DataLoader(val_dataset, batch_size=32, shuffle=False)
```

然後，我們可以開始微調我們的模型了：

```
from torch import nn, optim

# 定義損失函數和優化器
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# 開始訓練
for epoch in range(num_epochs):
    for images, labels in train_dataloader:
        inputs = feature_extractor(images, return_tensors="pt")
        outputs = model(**inputs)
        loss = criterion(outputs.logits, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

過程中，我們使用了交叉熵損失（CrossEntropyLoss）來衡量模型的性能，並使用 Adam 優化器來更新模型的權重。我們對數據集進行多次迭代，每次迭代我們都會經過一個前向傳播和一個反向傳播階段。在前向傳播階段，模型將對輸入圖像進行預測；在反向傳播階段，我們將計算損失並更新模型的權重。

經過訓練後，我們的模型現在可以用於新的圖像分類任務。結果的解析可以通過簡單地將模型的輸出與實際的標籤進行比較來完成。

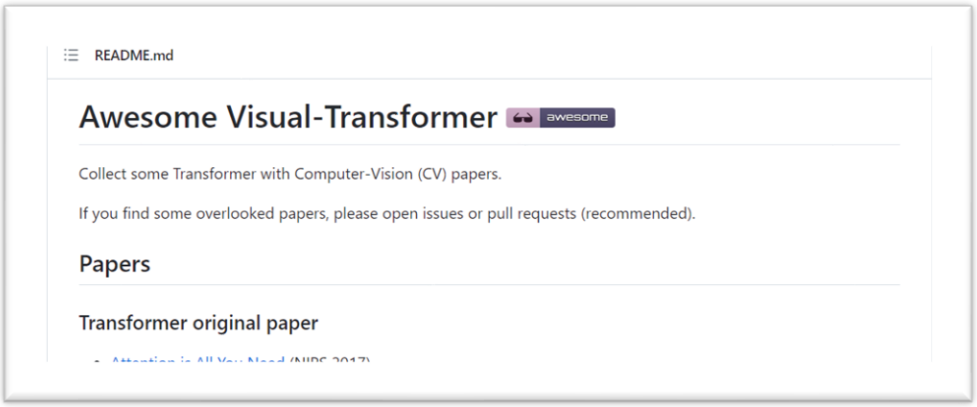
【實作連結】：[點此 COLAB]
可將圖片 List 上傳至一旁作為 Image 後，微調模型並進行圖像分類任務的訓練

六、Transformer 的未來發展和挑戰

(一)、研究進展和新模型介紹——Swin Transformer 與最新發展

2021-2022 年，當中最值得一提的進展就是 Swin Transformer，Swin Transformer 的名字由 "Shifted Window"（移位窗口）縮寫而來。Swin Transformer 則將注意力機制限制在局部範疇下進行，透過移位的窗口策略（Shifted Window Strategy），結合局部與全域的信息，打破了全連接自注意力機制的限制。Swin Transformer 引進的這種移位窗口策略，具有局部感知域的自注意力結構，它的效率和效果在一定程度上都超越了傳統的卷積神經網路。

Transformer 模型持續在自然語言處理和圖像處理領域做出重要突破，變化非常迅速，需要透過經常大量閱讀期刊與頂會來極實跟上產業趨勢，在此附上一處作者查到經常更新的相關文獻收集處 (點此)，若需要跟上最新趨勢，請務必實時查閱與大量學習以追上其研究與業界的發展。



(二)、現存問題和挑戰

1. **計算需求**：儘管 Transformer 模型在許多領域已取得重大突破，我們仍需認識到這些模型的計算需求（Computational Demand）。對於大型 Transformer 模型，如 GPT-3 或 BERT，他們需要龐大的訓練數據和強大的計算能力才能實現其最佳效能。這對於資源有限的研究者或小型企業而言，無疑是一項重大挑戰。
2. **空間結構和空間相關性的處理**：雖然在處理序列數據，如文本時，Transformer 模型表現出色，但在處理圖像數據時卻可能面臨挑戰。由於缺乏固定的空間結構，Transformer 可能不如卷積神經網絡（CNNs）那樣敏感於圖像的空間關聯性。這可能會在需要捕捉圖像的細微結構或空間規律的應用中影響其效果。

3. **動態環境與時間序列數據的處理**：儘管有一些 Transformer 的變體，如 Transformer-XL 和 Reformer，已經試圖解決處理長序列的問題，但在動態環境或處理時間序列數據方面，現有的 Transformer 模型仍有改進的空間。這些模型通常需要大量的計算資源來保持內部狀態，並且可能在處理長序列時面臨效率問題。
4. **模型解釋性和可視化**：雖然 Transformer 模型在許多任務上表現出色，但其內部運作方式仍然相對模糊，使得模型的解釋性和可視化變得困難。在許多實際應用中，模型的可解釋性和可視化是非常重要的，特別是在需要了解模型決策過程的場合，如醫學影像診斷和司法決策等。
5. **對抗攻擊和模型安全性**：與所有機器學習模型一樣，Transformer 模型也會面臨對抗攻擊（Adversarial Attacks）的挑戰。研究人員需要找到更有效的方法來提高模型的魯棒性並防止此類攻擊。

(三)、對未來發展的預測和展望

儘管目前的 Transformer 模型存在一些問題，但是我們從 Transformer 的演進來預見其模型的巨大潛力。越來越多的研究正在嘗試優化 Transformer 的結構，比如減少模型的大小、增加模型的效率等，這些都將使得 Transformer 能夠更好地融入與應用於各種任務和環境中。

隨著硬體技術的進步、算法優化的進一步研究，我們可以預期未來的 Transformer 模型將更有效地利用計算資源。這可能包括利用較少的數據或計算力來訓練模型，或改進模型以適應更大的數據集。此外，新的訓練策略和技術，如知識蒸餾（Knowledge Distillation）和模型壓縮（Model Compression），也可能有助於降低 Transformer 模型的計算需求。此外，隨著模型解釋性研究的發展（Explanable AI），我們可以預期會有更多的工具和方法來理解和視覺化 Transformer 模型的內部運作。這將有助於提升模型的可信度和可用性，並使得非專業用戶能夠更好地理解和利用這些模型。

另一方面，我們期待透過研究和實驗，能夠進一步理解 Transformer 的工作機制，並解釋其在各種任務中的成功。這不僅將推動 Transformer 模型的發展，也將有助於我們理解和建構更有效的深度學習模型。隨著 Transformer 等模型的普及，其對社會和個人的影響也逐漸浮現。我們可以期待在未來，人工智慧的倫理問題，如隱私、公平性和責任將受到更多的關注。將在，我們可以預見到 Transformer 在更多領域和應用中發揮作用，包括醫療診斷、自駕車、語言翻譯、音樂生成、遊戲 AI 等多種領域。

Bo-Yen Chang 2023/6/21