國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science & Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

使用生成式對抗網路及最佳補全蒸餾法之多標籤分類技術

Multi-label Classification Techniques with Generative Adversarial Network and Optimal Completion Distillation

蔡哲平

Che-Ping Tsai

指導教授：李琳山 教授

Advisor: Lin-shan Lee, Ph.D.

中華民國一百零八年七月

July, 2019

doi:10.6342/NTU201901512

國立臺灣大學碩士學位論文
口試委員會審定書

使用生成式對抗網路及最佳補全蒸餾法之多標籤分類技術

Multi-label Classification Techniques with Generative Adversarial Network and Optimal Completion Distillation

本論文係蔡哲平君（學號 R06922039）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 108 年 7 月 9 日承下列考試委員審查通過及口試及格，特此證明

口試委員：李琳山

（指導教授）

陳信宏　鄭秋豫

王小川　李宏毅

系主任　莊永裕

誌謝

碩士兩年的時光，說長不長，說短也不短，在台大待了六年的時光也即將畫下句點。做研究、出遊、出國開會、忙著趕論文，碩士兩年實在有太多事值得回憶了。過程中，我得到了許多人的幫忙，不僅讓我在知識上有所長進，也從許多人的身上學習到待人處事和對生活的態度，受到的大恩大德也許無法用文字好好表達，但這些感些都會長存我心。

首先要感謝的是我的指導老師李琳山教授，從大三上學期選了老師的專題到現在碩二要結束了，也有四年的時光，從我還什麼都不懂，在老師的指導下，到現在能寫出幾篇學術論文，真的由衷的感謝老師四年來的栽培與教導。從老師身上真的學習到了很多東西，不僅僅是課堂上的知識或是論文的寫法，也學習到了很多人生哲學，讓我能在接下來的人生道路上，能更堅定自己的選擇，這份恩情永生難忘。

再來要感謝宏毅哥，老師平常就像我們的朋友一樣，會和我們一起出去吃飯，也會討論一些生活大小事或八卦，真的很好相處，但聊到研究時，老師認真的態度和專業的見解又讓我好生佩服。每次跟老師討論完，都會得到新的啓發，也會有新的動力朝向新的研究方向前進。雖然有時候老師您會用對你很期待的語氣祝福你成功，讓我懷疑老師的方向是不是對的，但後來成功訓練後，才發現老師真的有先見之明，讓我很佩服老師對研究的直覺。

還要感謝實驗室的學長們，柏儒哥、水靜、致緯哥、邦齊哥、家宏哥、舜博哥、瓊之、育軒哥、棋宇哥、佩宏哥，謝謝你們在我還在懞懂的碩一時，報了很多paper，讓我迅速學習到最新領域的進展，和你們請教問題時，你們也不厭其煩地教導我。和學長們出國開會的經驗也令我永生難忘，謝謝你們。

感謝實驗室的麻吉們，耀文、冠宇、淞楓、儒杰、政杰、靖平、耀賢、宗

doi:10.6342/NTU201901512

媛、奕禎、上銘、佳軒、達榮、思霖、元魁、逸林，真的很幸運碩士生涯有你們的陪伴，其中有幾位甚至從高中就開始了，這段時間一起打球、一起熬夜、一起趕論文、一起吃飯、一起出遊.....，太多太多的事和難關是跟你們一起完成，電腦出問的時候有你們的幫忙真的感激涕零。過程中有太多珍貴的回憶，在實驗室的時間有你們一起，帶給我許多的能量能夠繼續前進。謝謝靖平和佳軒，不僅是好同學也是好室友。謝謝儒政雙杰，資工系的梗總是特別又好笑。謝謝耀賢，在supercell的羈絆我永遠不會忘記。謝謝辛苦的網管耀文，戰艦系統實在太厲害了，我電腦有問題的時候總是向你求救，真的是十萬分的感謝，你也不厭其煩地幫我解決。謝謝奕禎，除了隊醫以外，其他方面真的都是我的老師。謝謝淞楓，從國中就認識了，謝謝你12年來的凱瑞。謝謝冠宇帶我發了一篇interspeech的paper。謝謝QA組的上銘和宗媛，坐在你們旁邊聽你們講話真的很有趣。謝謝KGB好隊友思霖和達榮，你們修課的時候很凱瑞，嘴砲的時候也很有趣。謝謝元魁在籃球和傳說上的教導。謝謝專題時期的隊友逸林，和你合作時總是感到十分安心。雖然畢業後有些人要離去了，大家也往不同的方向前進了，但大家都會是我一輩子的麻吉。

感謝實驗室的學弟和助理，雖然只有一年的相處時間，但看到你們那麼認真，學長壓力也大了起來，也會督促自己努力一點。感謝浩然和培傑去英國的時候凱瑞了很多事，浩然也讓我問了許多影像相關的問題，也感謝元瑞、杜濤陪我們這些人打球，也謝謝君璇每次我去實驗室都會跟我打招呼，謝謝網管記良、瑞陽讓我們有戰艦用，也謝謝神賢、海濱、瑞陽、博竣、仲翊、柏文、昭誼、廷緯這一年來的陪伴，實驗室有你們一定能夠更上層樓。

感謝彤恩姐許多行政事務上的協助，像我們的媽媽一樣，打理了我們各種瑣事，平常也會關心我們過得如何，我也可以和你抱怨各種瑣事，讓我心情舒坦不

doi:10.6342/NTU201901512

少。祝你接下來在實驗室能夠事事順心，不會遇到麻煩事。

最後要感謝我的家人，不僅在這兩年給我金錢上的援助，一直以來也都支持我做的決定，讓我能沒有牽掛的做我想做的事，每次回家的時候總覺得十分的放鬆，對我的養育之恩我一輩子都償還不完。

doi:10.6342/NTU201901512

摘要

本論文的主軸是多標籤分類(Multi-Label Classification)之新技術。隨著機器學習技術的日新月異，基於深層類神經網路(Deep Neural Network)的解決方法陸續被提出，前人的研究指出考慮標籤間的關聯性，是增進模型表現的關鍵。

本論文的第一個大方向是以生成對抗網路(Generative Adversarial Network)來模擬標籤關聯性。在此架構下，分類器扮演生成器(Generator)的角色，其輸入是一個物件，輸出是屬於此物件的標籤集(Label set)，鑑別器(Discriminator)則需要學習標籤之間的關聯性，來分辨此標籤集是從生成器產生還是來自真實的資料;分類器不只需要學會標籤和物件間的關係，也需要使產生出的標籤集具有正確的關聯性，以欺騙鑑別器。

本論文第二個方向是改進基於遞迴式類神經網路(Recurrent Neural Network)的多標籤分類器;這種模型使用遞迴式類神經網路解碼器來模擬標籤關聯性，並依序預測標籤。然而，此模型在訓練時，需要人為定義的標籤順序，用來將標籤集轉變成標籤序列，為訓練遞迴式類神經網路的目標序列;前人的研究已指出標籤順序對模型表現有相當大的影響，人為強加的順序性也可能會和機器推斷的標籤關係不一致。因此，本論文提出最佳補全蒸餾法(Optimal Completion Distillation)，使模型不需要標籤順序便可訓練。透過分析實驗數據，我們也證實我們提出的模型不只表現較好，廣泛化能力(Generalization ability)也較強，能夠預測出在訓練集沒有出現過的標籤集。

本論文也提供了上述兩種方法在多標籤影像分類、文件分類、環境音分類上相當豐富的測試結果。

doi:10.6342/NTU201901512

Abstract

Multi-label classification (MLC) assigns multiple labels to each sample. This paper proposes two methods that improves performance of multi-label classifiers.

Recent work has shown that exploiting relations between labels improves the performance of multi-label classification. The first direction in this paper is to use Generative Adversarial Network (GAN) to model label dependencies. The discriminator learns to model label dependency by discriminating real and generated label sets. To fool the discriminator, the classifier, or generator, learns to generate label sets with dependencies close to real data.

The second direction is to improve state-of-the-art multi-label classifiers , which utilize a recurrent neural network (RNN) decoder to model the label dependency. However, training a RNN decoder requires a predefined order of labels, which is not directly available in the MLC specification. Besides, RNN thus trained tends to overfit the label combinations in the training set and have difficulty generating unseen label sequences. Therefore, we propose a new framework for MLC which does not rely on a predefined label order and thus alleviates exposure bias. We also find the proposed approach has a higher probability of generating label combinations not seen during training than the baseline models. The result shows that the proposed approach has better generalization capability.

This paper also provides experimental results on multiple multi-label classification benchmark datasets in different domains, including text classification, image classification and sound-event classification.

doi:10.6342/NTU201901512

Contents

doi:10.6342/NTU201901512

doi:10.6342/NTU201901512

圖目錄

doi:10.6342/NTU201901512

doi:10.6342/NTU201901512

表目錄

doi:10.6342/NTU201901512

doi:10.6342/NTU201901512

第一章　導論

1.1　研究動機

過去，人類在文件、圖像、語音等資訊的管理，大多仰賴人力進行。近年來隨著科技快速發展，電腦處理的資訊也越來越豐富且多元，使用人力管理的成本也越來越高，如何快速有效的進行分類及管理，變成了現今世界的重要問題。在過去已有許多利用電腦進行的資訊分類系統，並普及應用於生活中，例如圖像分類、文本分類、垃圾信分辨等等。一般的分類系統大多是只能分出單一種類別，也稱作單標籤分類器(Single-lable Classifier)，例如垃圾信分類器只需要簡單的分辨是或不是垃圾信、某些動物的圖像分類器只能區分出一種動物。在這種單標籤分類中，每一個類別是不會重複的，並且每一個物件(instance)只會屬於某一個類別。然而，在現實生活中，有許多任務是更加複雜的，有許多物件可以同時被區分成好幾種類別，舉文件分類作為例子，一個職棒打假球的新聞，是可以被分到社會類和體育類兩種類別的。因此，單標籤分類器漸漸不敷使用，而多標籤分類(Multi-label Classification)受到越來越多的重視及研究。

在多標籤問題中，標籤中是互相有關聯性的，例如在影像分類的問題中，沙灘和海洋十分容易出現在一起，但沙灘和大象卻鮮少出現在同一張圖片上，因此，標籤之間常有一定的關聯性，某些標籤可能常常會在同一物件中出現，某些標籤卻不會一起出現。如何使模型同時學習到標籤與物體之間的對應關係和標籤間的關聯性，是多標籤分類研究的一大方向。近年來由於深層學習的興起，基於深度類神經網路的方法也陸續被提出，例如用機率圖模型(Probabilistic graphical networks) [1]、遞迴式類神經網路(Recursive Neural Network, RNN) [2] 等來模擬標籤間的關聯性。

doi:10.6342/NTU201901512

本論文第一個重點是利用生成對抗網路(Generative Adversarial Network, GAN) [3] 來模擬標籤之間的關聯性。對抗生成網路是近年來迅速火紅的深度生成模型，在圖像生成(Image Generation)有非常好的效果，是由一個生成器(Generator)和一個鑑別器(Discriminator)所組成，藉由交互對抗式的訓練，最終能使生成器產生鑑別器無法分辨真偽的圖片。在此鑑別器的工作是引導生成器學習，使生成器能產生出更貼近真實、更結構化(structural)的圖片，因此，鑑別器也能應用在其他任務上，例如在語音合成(Text to Speech, TTS)中引導模型生出更貼近真實的聲音 [4] 或是在語音辨識(Automatic Speech Recognition, ASR)系統裡讓模型生出更貼近真實的句子 [5]。同樣的，若將鑑別器應用於多標籤分類，應該也能引導模型產生真實會出現的標籤集(Label set)。換句話說，鑑別器應能使生成器學習到標籤之間的關聯性，從而了解到哪些標籤較容易一起出現，哪些標籤鮮少同時出現在同一個情境中。

本論文第二個重點是改善基於遞迴式類神經網路的多標籤分類器，先前有人將多標籤分類轉變成一個順序預測(sequential prediction)問題 [2,6]，其精神是在分類器產生標籤是順序性一個一個產生的，換句話說，在產生標籤時，會基於之前預測的標籤來做預測，而這樣的模型會接一個遞迴式類神經網路來達成這個效果，並利用此模擬標籤之間的關聯性。然而，使用這種方法需要先將訓練用的標籤排序;而排序標籤的方法會對結果的影響十分顯著，因為對原本無序的標籤集套用人為定義的順序是不自然的，況且順序的可能組合千千萬萬種，找到一個完美的順序也十分的困難，本論文提出的方法便可以不需要人為順序，便可訓練此類型的多標籤分類器。

另外，在這種基於遞迴式類神經網路的多標籤分類器，在訓練時只會基於來自正確解答的前輟(prefix)，然而在測試時，模型會基於自己的預測的前輟來做預

doi:10.6342/NTU201901512

測，而這個前輟可能會是錯的，而這種錯誤可能會一連串的發生下去，而導致錯誤越來越大，這種問題在序列到序列訓練(Sequence to Sequence Learning)中稱為曝露偏差(Exposure bias)。本論文也提出一個二元關聯(Binary Relevance, BR)的解碼器(decoder)來輔助模型訓練，在多目標學習(Multi Task Learning, MTL)的框架下讓模型能學習的更好，而另一個好處是，在測試時能同時考慮兩種解碼器的預測值，進而增進預測的表現。

因此，本論文引入最佳補全蒸餾法(Optimal Completion Distillation, OCD) [7]並提出與二元關聯解碼器共同訓練的的多目標學習來改善基於遞迴式神經網路的多標籤分類器，其中，最佳補全蒸餾法能使遞迴式類神經網路不用依賴於人為定義的順序，並且在訓練時皆是使用目前模型的預測來當前輟，從而避免了曝露偏差的問題。

1.2 研究方向

本論文之研究方向為如何增進多標籤分類器的表現，主要分為兩個方向，其一是利用生成對抗網路增進分類效能，另一者則是利用最佳補全蒸餾法和多目標學習來幫助基於遞迴式類神經網路的多標籤分類器的訓練。

- 在前人的研究已經顯示在多標籤問題中，標籤之間的關聯性是增進分類器表現的關鍵，本論文提出了一個基於生成對抗網路的方法，讓分類器不只能學到物體與標籤的對應關係，更能了解標籤之間的關聯性，進而增進分類器的表現。
- 更進一步的，由於現今對如何訓練生成對抗網路與增進其表現仍無明確的定論，因此，論文中比較了不同的訓練方法與結構，並提出一個效能最好的模型和訓練方法，並證明了加入生成對抗網路確實能幫助多標籤分類器，在實

doi:10.6342/NTU201901512

驗中，也會比較各種訓練方法及最基礎的多標籤分類器模型的優劣，並分析加入生成對抗網路後，模型更加進步的原因。

- 另一方面，因為以往基於遞迴式類神經網路的多標籤分類器需要人為定義的標籤順序，也會有上述曝露偏差的問題，因此在本論文中引入最佳補全蒸餾法來解決，
- 另外，本論文中提出利用多目標學習的方式，引進二元關聯解碼器，不但能夠使模型學得更好，並在測試階段時，也可以結合兩者的預測結果，得到一個更好的分類結果。
- 就上述更進一步，將最佳補全蒸餾法和多目標學習結合，能得到最好的模型。論文的實驗中，比較了五種不同的測量標準，不僅證明了上述方法皆能使模型表現更好，也證明了使用最佳補全蒸餾法能改善曝露偏差的問題，最後也比較了與基於生成對抗網路的模型的優劣。

1.3 章節安排

本論文之章節安排如下：

- 第二章：介紹本論文相關背景知識。
- 第三章：介紹如何用生成對抗網路使多標籤分類器學得更好。
- 第四章：介紹如何利用最佳補全蒸餾法於多標籤分類，並與前一章之方法比較。
- 第五章：本論文之結論與未來研究方向。

doi:10.6342/NTU201901512

第二章　背景知識

2.1　多標籤分類(multi-label classification)

2.1.1　簡介

在機器學習領域中，多標籤分類問題是一個歷史悠久且具有挑戰性的問題，多標籤分類不只有多種類別(class)，每個物件更可能有許多標籤。相對於多類別分類器(multi-class classifier)只需要將物件分類至某一類別，多標籤分類更加的困難，因為每個物件的標籤數是未知的。多標籤分類也有許多應用，例如：新聞或電影的分類、生物學的基因分類、醫療中的疾病分類、聲音事件檢測(sound event detection)等等都可以同時有多個標籤。

在多標籤分類中，標籤之間是有關聯性的，以多標籤圖片分類作為例子，沙灘和海洋十分容易出現在同一張圖片裡，而沙灘和大象則鮮少一起出現。因此，有些標籤較容易一起出現，有些則不是。如何模型標籤之間的關聯性來幫助分類器，是多標籤分類問題主要的一個研究方向。

2.1.2　常見的解決方法

多標籤分類問題常見的解決方法有：二元關聯(Binary Relevance, BR) [8]、分類器鏈(Classifier Chain, CC) [9]、標籤冪集(Label Powerset, LP) [10,11]等等，接下來會逐一介紹這些方法。

- 二元關聯：如圖2.1(a)，此類的方法將多標籤分類問題轉變至多個二元分類(binary classification)問題。假設在多標籤問題中有L種類別，此種方法將L個類別分開來看，變成L個二元分類問題，則這L個二元分類器便分別判

doi:10.6342/NTU201901512

斷某類的標籤是有是無。此類的方法將所有的類別分開判斷，因此，此類的方法缺乏考慮標籤之間的關聯性。

- 分類器鏈：此類的方法如同二元關聯，將多標籤分類問題轉變至多個二元分類問題，不同的地方是分類器鏈會依順序性預測，每個二元分類器皆會依照之前的標籤的預測結果來做預測。如圖2.1(b)，黃色部分是每個二元分類器的輸入，在二元分類器B和C，其不只考慮了物件的特徵(feature)，更會考慮到之前標籤的分類結果。此類的方法可以考慮到類別之間的關聯性，但卻需要人類事先定義的標籤順序來做預測。

- 標籤冪集：標籤冪集將多標籤分類問題轉變成多類別分類問題(multi-class classification)，標籤冪集先訓練集中的標籤集(label set)作統計，將同樣的標籤集分做一類，因此，若有L種標籤，最多會有2^L種獨特的類別，因此，每一個物件只會屬於某一個獨特的類別，如圖2.2，物件X_1, X_2具有相同的標籤集A, C，因此這兩個物件的標籤是屬於同一類。此類的方法在標籤集種類很多時便不適用，且無法預測沒有在訓練集中出現過的標籤集。

另一類的方法 [12–15]，是基於潛在空間(latent space)，他們將標籤集利用降維轉換至此連續空間，並訓練模型在此連續空間內學習到標籤之間的變化，進而學習到標籤關聯性。在解碼時，則在此潛在空間找尋輸入物件對應的標籤集。另外，由於近年來深層學習的興起，近年來基於上述方法的深層模型也被提出，在章節2.2.1和2.2.4也會多做介紹。

doi:10.6342/NTU201901512

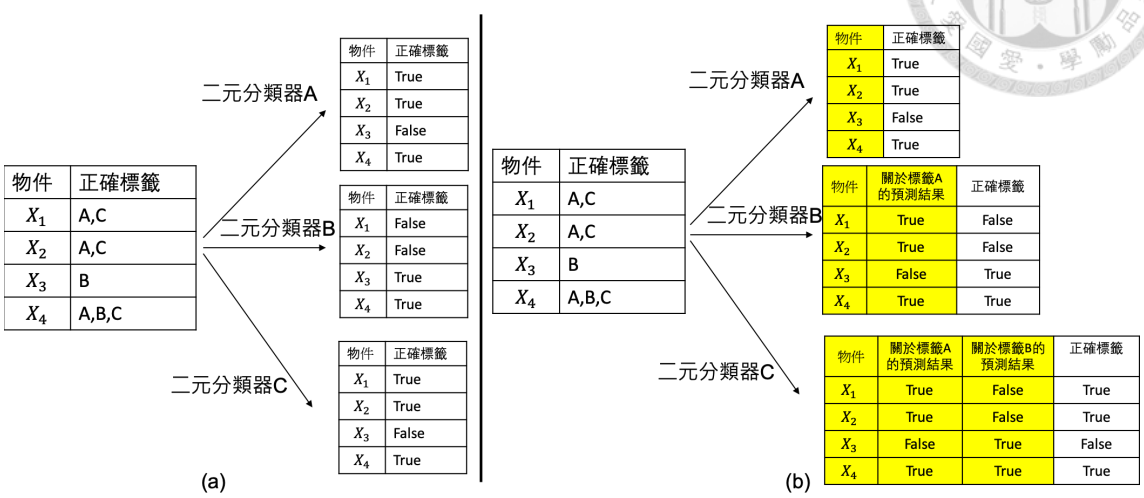


圖 2.1: (a)二元關聯示意圖，將原本A,B,C三個標籤分別用三個二元分類器來做預測(b)分類器鏈示意圖，黃色部份是指分類器的輸入，白色的部分則是分類器需要預測的標籤。

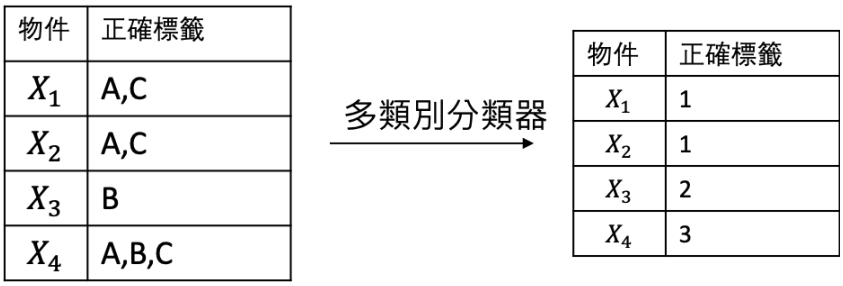


圖 2.2: 標籤冪集示意圖，因為X_1, X_2的標籤集相同，因此屬於同一個類別。

doi:10.6342/NTU201901512

2.2 序列到序列(sequence-to-sequence)模型

2.2.1 類神經網路(Neural Network, NN)

簡介

在機器學習領域中，類神經網路是一種模仿生物類神經網路的結構和功能的數學模型或計算機模型。在生物的結構上來看(如圖2.3)，神經系統是由非常多的神經元組成，彼此以樹突、軸突與突觸連結，每個神經元有活化閾值決定是否變成激發態。同樣的架構應用至電腦科學領域時，前人也設計可以計算的神經元，彼此層層連結(如圖 2.4)，每個神經元也必須被激發，才會有資訊流入下一層，此即為類神經網路的全貌。類神經網路已經被用於解決各種各樣的問題，例如電腦視覺和語音辨識。這些問題都是很難直接用傳統基於規則的程序所解決的。

運作原理

類神經網路的架構如圖2.4，是由許多的感知器(Perceptron)(圖2.5)串接而成，因此深層類神經網路又被稱為多層感知器(Multi-layer Perceptron, MLP)。每一層的感知器個數稱為此類神經網路的寬度(width)，而所有含感知器的層數稱之為深度(depth)，因為近期推出的圖形處理器(Graphics Processing Unit, GPU)大幅提高矩陣運算的速度，現今使用的類神經網路都相當多層，因此此類神經網路可稱之為深層類神經網路(Deep Neural Network, DNN)。每一層根據所在位置的不同，可以

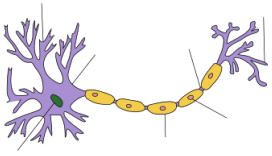


圖 2.3: 神經示意圖

doi:10.6342/NTU201901512

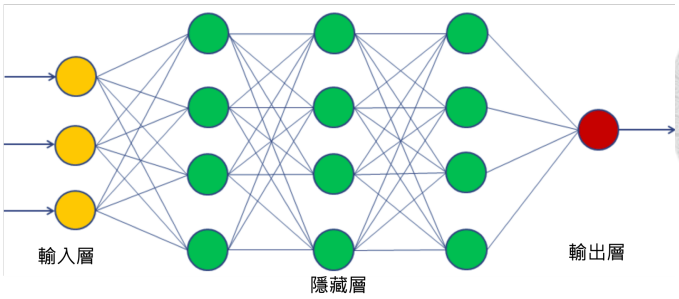


圖 2.4: 類神經網路示意圖

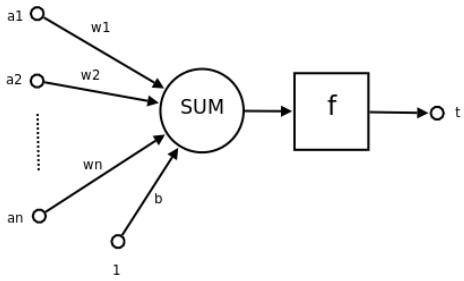


圖 2.5: 運算神經元示意圖

分為三種:

- 輸入層(Input Layer):類神經網路由此層輸入特徵向量，輸入的訊息稱為輸入向量。
- 輸出層(Output Layer):訊息在神經元連結中經傳輸、運算，輸出的訊息稱為輸出向量。
- 隱藏層(Hidden Layer):是輸入層和輸出層中間的各個層。可以有一層或是多層，深度和寬度數目不定。

在圖2.5中，$\{a_1, a_2, ...a_n\}$是輸入向量的各個分量，n是前一層的神經元個數，而$\{w_1, w_2, ..., w_n\}$是每個神經元的加權值，$b = \{b_1, b_2, ...b_n\}$是偏移量(bias)，f則是活化函數(activation function)，常見的是S函數(sigmoid)，整流線性單元(Rectified Linear Unit, ReLU)，運算的數學式分別如下：

doi:10.6342/NTU201901512

$$t = f(\sum_{i=1}^{n} w_i a_i + b) \tag{2.1}$$

$$sigmoid(x) = 1/(1 + e^{-x}) \tag{2.2}$$

$$ReLU(x) = max(0, x) \tag{2.3}$$

若寫成矩陣的形式，則每層之間的轉換可以當作是一個從M維實數空間映射至N維空間的函數$R^N \leftarrow R^M$，M和N分別是輸入向量X和輸出向量Y的維度。

$$Y = f(WX + b) \tag{2.4}$$

訓練類神經網路

在深層類神經網路內，最常見的訓練方法為反向傳播演算法(back propogation)，通常會搭配一些最佳化演算法，例如梯度下降法(gradient descent algorithm)。在訓練時，會先定義一個特定的減損函數(loss function)$\sum_{n=1}^{N} L(y_n, \hat{y}_n, \theta)$，來衡量目前的模型的輸出$\hat{y}_n$和正確答案$y_n$的差距有多少，一般來說，減損函數愈大，代表誤差愈大，而訓練的目的就是減小誤差。其訓練的目標可以表示成如下的最佳化的問題:

$$min_\theta \sum_{n=1}^{N} L(y_n, \hat{y}_n, \theta) \tag{2.5}$$

常用的減損函數有多類別分類問題使用交叉熵(Cross Entropy, CE)或在二元分類上使用的對數機率回歸(logistic regression)。以多類別分類器為例，假設有C個類別，在機器學習領域中，會先將其表示成一個維度C的獨一餘零(1-hot)的向

doi:10.6342/NTU201901512

量$y = [0, 0...1, ..., 0]^T$，只有在正確類別l的維度是1，而其餘是0。分類器的輸出也會是一個維度C且總和為1的向量$\hat{y}$，第i個維度代表屬於第i類的機率，是一個機率分佈。訓練需要使y和$\hat{y}$的距離越近越好，因此通常會使用交叉熵作為減損函數，定義為：

$$L_{CE}(y, \hat{y}) = \sum_{i=1}^{C} y_i log(\hat{y}_i) \tag{2.6}$$

其中，減少交叉熵等同於減少兩機率分佈y和$\hat{y}$的庫雷散度(Kullback-Leibler divergence)，值越小代表兩機率分佈的距離越近，反之則代表兩分佈越不相似。

對於二元分類，通常使用對數機率回歸的減損函數，定義如下：

$$L_{logistic}(y, \hat{y}) = ylog(\hat{y}) + (1 - y)log(1 - \hat{y}) \tag{2.7}$$

其中，$y \in \{0, 1\}$代表是否屬於此類別，屬於的話是1，反之則為0，$\hat{y} \in [0, 1]$為分類器的輸出機率，代表屬於此類別的機率，此減損函數是交叉熵在$C = 2$時的特例，此時只有兩個類別，分別代表“屬於”和“不屬於”，因此可以只用一個數字1或0來表達。

當設計者決定好減損函數後，接下來就是要找到模型的參數集θ，使得減損函數最小。

$$\theta^* = arg\min_{\theta} \sum_{n=1}^{N} L(y_n, \hat{y}_n, \theta) \tag{2.8}$$

其中N為資料量的個數。由於θ的參數空間太大，很難直接找到公式2.5最佳解(optimal solution)，因此常用的方法是用梯度下降法，藉由一步一步的更新參數，使減損函數愈來愈小。因此，減損函數通常是定義成可微分的。簡單的演算法是統計式梯度降低(Stochastic Gradient Descent, SGD)，損失函數沿著該參數上的

doi:10.6342/NTU201901512

梯度方向更新，可以表達成：

$$\theta_{k+1} \rightarrow \theta_k - \eta \triangle \theta_k$$
$$\triangle \theta_k = \left. \frac{\partial L}{\partial \theta} \right|_{\theta=\theta_k} \tag{2.9}$$

其中η為學習率(learning rate)，調控最佳化的速度與精細度，k為更新的迭代次數，隨著k的增加，減損函數的值能逐步減少，模型就能訓練得更好。

在使用統計式梯度降低訓練類神經網路時，在模型完成順向預測(forward prediction)後，為了要算每層神經元的梯度，會先算出減損函數對輸出層的梯度，再一層一層使用鏈鎖律(chain rule)往反向的算至輸入層，也就是反向傳播演算法。近年來，也有許多對於學習率的研究，例如基於物理學的動量(momentum)的演算法，對於學習率有限制的Adagrad更新法，和融合上述的Adam更新法，都是目前深層學習廣泛應用的演算法。

然而，深層學習常常遇到過度貼合(overfitting)的問題，也就是說，模型在訓練集(training set)的表現遠遠優於在驗證集(validation set)的表現，這代表類神經網路已經偏向在“記憶”訓練集的正確答案，而缺乏廣泛化能力(generalization ability)，因此，在訓練類神經網路時，我們常會使用L1,L2正規化(regularization)或是丟棄法(dropout) [16]，避免過度貼合的問題。

深度深層模型如何應用於多標籤分類

類神經網路的架構可以輕易地和二元關聯的方法做結合，也就是說，可以將多標籤分類轉變成多個二元分類問題，但是所有的分類器使用同一個類神經網路。假設總共有L個類別，則類神經網路可以如圖2.6架構設計。

其中，使用的減損函數是L個二元分類的減損函數總和，如式2.10

doi:10.6342/NTU201901512

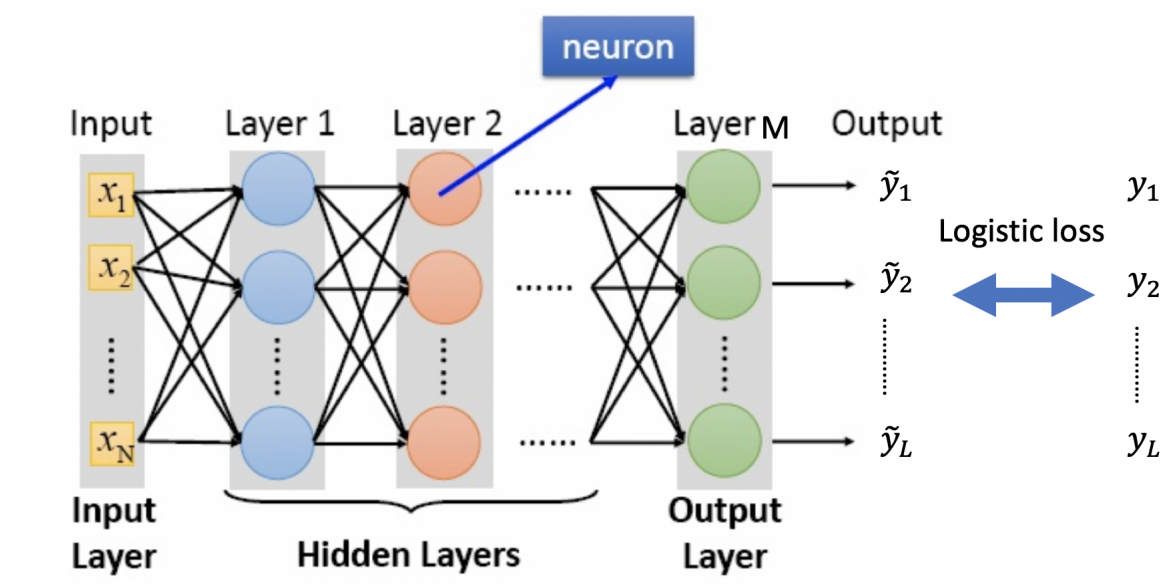


圖 2.6: 二元關聯應用於深度深層模型之示意圖

$$L_{logistic}(y, \hat{y}) = \sum_{i=1}^{L} [y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)] \tag{2.10}$$

如此便是使用深度類神經網路在多標籤分類上的一個基本架構。

2.2.2 遞迴式類神經網路(Recurrent Neural Network, RNN)

上述類神經網路是深層學習中的基本網路架構，可以應用於許多問題，但是在圖像、聲音、文字這種訊號時，因為訊號十分複雜，如果每層都使用都全連接(fully connected)的深度深層網路，會使訓練參數過大，效果也十分不好。因此，便有了針對圖像的卷積式類神經網路(Convolutional Neural Network, CNN)和對於時間序列所設計的遞迴式類神經網路。遞迴類神經網路的隱藏層具有「記憶」功能，除了考慮當下時間點的資訊外，也會參考過去所輸入的資訊，因此，對於處理文字和聲音訊號這種時間序列經常使用此類型的類神經網路。以下我們將分別介紹遞迴類神經網路的基本數學原理以及長短期記憶單元(Long Short-Term

doi:10.6342/NTU201901512

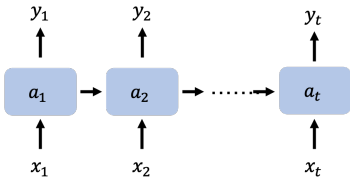


圖 2.7: 基本的遞迴式類神經網路

Memory，LSTM)。

數學原理

圖2.7為基本的遞迴式類神經網路，圖中的a是指該網路的記憶單元，而此記憶單元a_{t-1}在時間點t，會與該時間點的輸入x_t，經過一連串的矩陣變換，爾後產生輸出y_t，如此一來，在輸出y_t時，此網路不只會考慮該時間點的輸入x_t，更會融合時間t之前所保留下來的資訊，因此在訓練類神經網路時，此網路不只會學習到輸入和輸出之間的對應關係，更會學習到哪些資訊需要保留下來，而哪些資訊可以捨棄。

長短期記憶單元

然而，遞迴式類神經網路也存在了許多問題，例如梯度消失(gradient vanishing)，原因是因為遞迴式類神經網路在傳遞梯度時，會經過許多活化函數，而此活化函數若沒有設計好，則容易造成梯度越來越小，而此類網路又會輸入十分長的時間序列，梯度每個時間點皆會經過一次記憶單元，因此會有梯度便會越來越小。因此，在後人的研究中，提出了長短期記憶單元模型，其概念如圖2.8所示。

長短期記憶單元是更加複雜的遞迴式類神經網路的組成單元，其透過不同的閘門(gate)來控制網路中資訊的流動，而閘門的設計是輸出介於0到1的S函數，其值代表這個閘的輸出和輸入的比例。如圖2.8，共有三種閘門，分別是輸入閘

doi:10.6342/NTU201901512

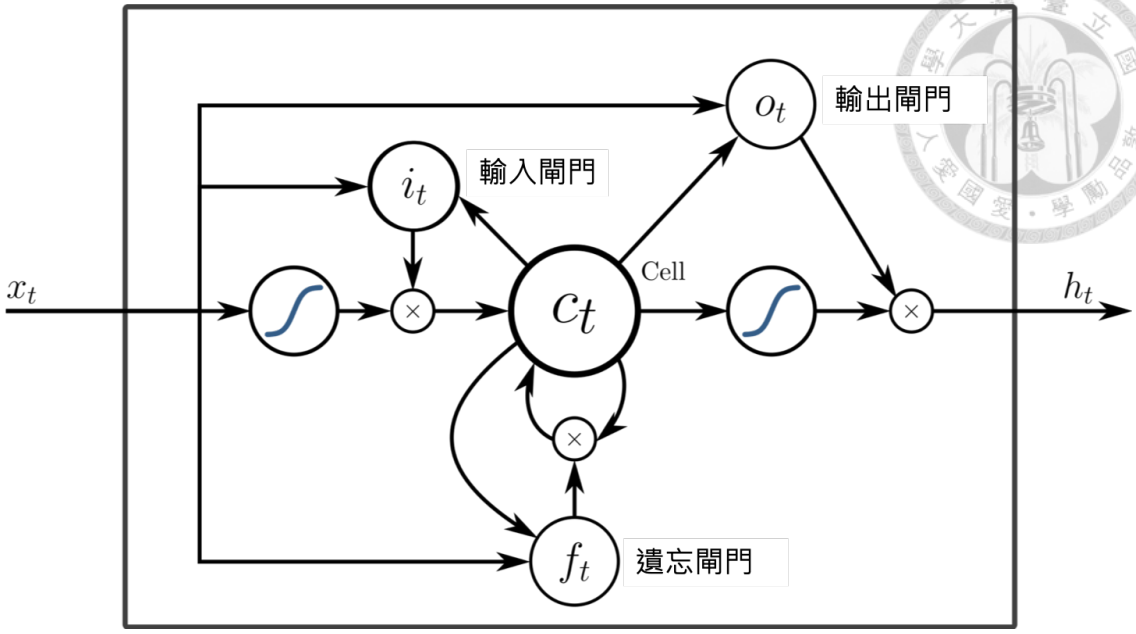


圖 2.8: 長短期記憶單元示意圖

門(input gate)、輸出閘門(output gate)和遺忘閘門(forget gate)，輸入閘門負責控制輸入的資訊量有多少需要存入記憶，輸出閘門負責決定多少比例的記憶需要用來輸出，而遺忘閘門則負責決定多少比例的記憶資訊可以被拋棄，如此一來，梯度便只會影響到開關開啓的時候，而減少梯度消失的問題。

2.2.3 序列到序列模型

序列到序列模型是由兩個部分組成，分別是編碼器(encoder)和解碼器(decoder)，皆是由遞迴式類神經網路所組成，如圖2.9。當我們需要處理輸入和輸出皆是序列的情況時，特別是輸入輸出序列不等長的時候，便會使用此種模型，如機器翻譯(machine translation)、語音辨識等等。其中，$\{x_1, x_2, ..., x_m\}, \{y_1, y_2, ..., y_n\}$分別代表編碼器和解碼器的輸入資料序列和輸出資料序列。編碼器負責將輸入資料編碼成一個維度固定的代表向量，而解碼器需要根據此代表向量，進行解碼，而得到一串輸出序列。然而，解碼器在產生資料序列時，是一個一個單位生成的，並

doi:10.6342/NTU201901512

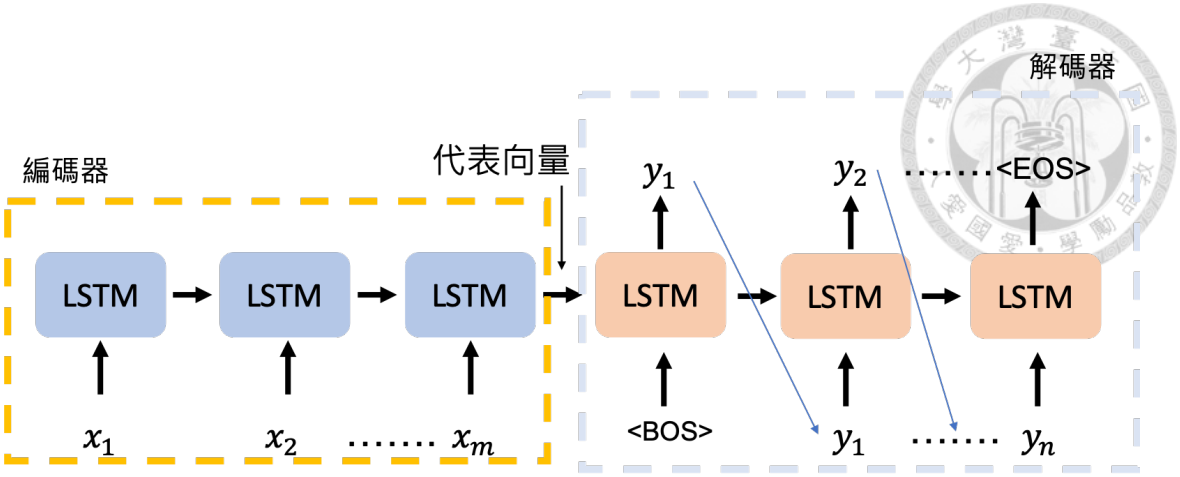


圖 2.9: 長短期記憶單元示意圖

且會基於之前所做的預測來產生這一步的輸出，因此，解碼器在每一步的輸入皆會包括上一個時間點的輸出。而因為解碼器可以無限制的產生序列，我們會用SOS和EOS來代表開始(start-of-sentence)和結束(end-of-sentence)，在第一個時間點輸入SOS來告訴解碼器這是序列的開始，而解碼器輸出EOS時，就是這個序列的結束。

在訓練序列到序列模型時，我們常會使用最大似然估計(Maximum Likelihood Estimation, MLE)，訓練使用的減損函數如下：

$$L_{MLE}(y,\hat{y}) = -\mathbb{E}_{(\mathbf{x},\mathbf{y}^*)\sim data}\sum_{t=1}^{|y^*|} logp_{\theta,t}(y_t^*|\mathbf{y}_{<\mathbf{t}}^*,\mathbf{x}) \tag{2.11}$$

其中，$\mathbf{x},\mathbf{y}^*$是配對好的資料(paired data)，訓練的目標是最大化目標資料序列$\mathbf{y}^*$，也就是說，在每一個時間點t，會基於時間t之前的正確答案$\mathbf{y}_{<\mathbf{t}}^*$，而做出時間t的預測，而訓練的目標便是最大化時間$t$產生正確結果的機率，此訓練方法也稱之為教師强迫(teacher forcing)。

然而，因為序列到序列模型中，輸入序列通常會十分的長，而輸入序列的資訊又只由代表向量提供，因此輸入序列前段的訊息很難傳到解碼器，而且會有梯度消失的問題，因此 就有人提出專注(attention)機制的想法，將在下一小節做介

doi:10.6342/NTU201901512

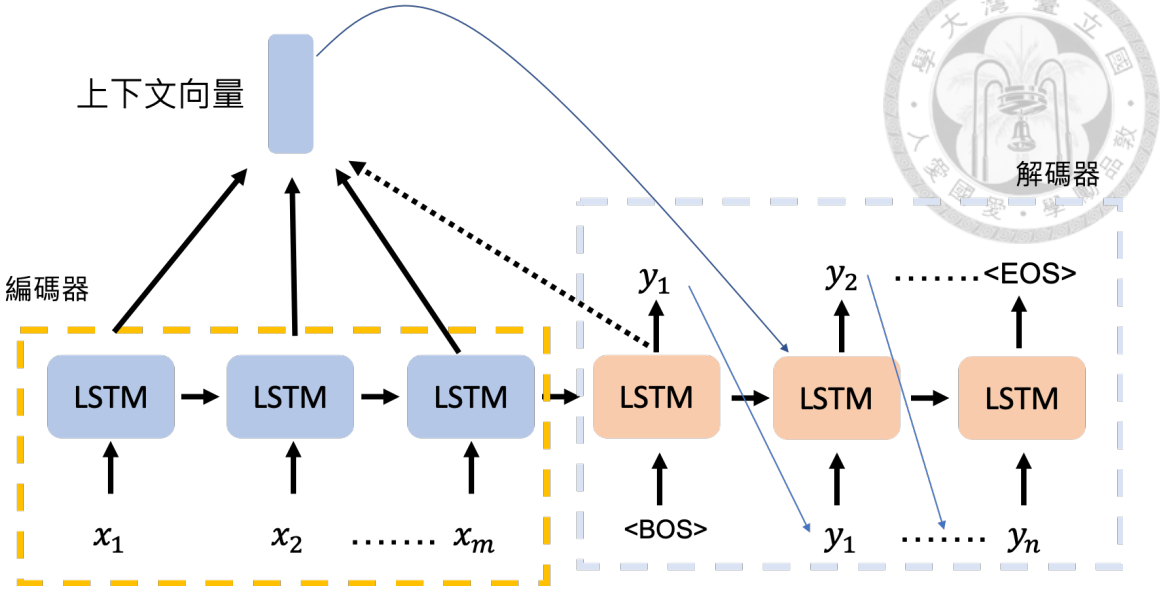


圖 2.10: 長短期記憶單元示意圖

紹。

專注機制

專注機制可以使解碼器直接取到編碼器的資訊，而不是只從代表向量中得到，因此梯度較容易傳回編碼器，而能夠使一般序列到序列模型能表現更好。另外，因為在在解碼時的每一個時間點，解碼器考慮每個輸入的權重不應相同，像是在機器翻譯中，每個詞之間應有對應語言的翻譯，因此，在解碼每個詞的時候，需要考慮對應的輸入文字也不盡相同。專注機制給予了模型這個彈性，使模型可以在訓練時，動態的調整對於輸入序列的權重。以下我們以數學式來解釋專注機制，給定編碼器的隱藏層序列$h_1^e, h_2^e, ...h_m^e$及解碼器在時間$t-1$的隱藏層h_{t-1}^d，上下文向量如以下的方法計算。

$$e_{tj} = v_a^T tanh(W_a h_{t-1}^d + U_a h_j^e) \tag{2.12}$$

doi:10.6342/NTU201901512

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{m} \exp(e_{tk})} \tag{2.13}$$

$$c_t = \sum_{j=1}^{m} \alpha_{tj} h_j^e \tag{2.14}$$

$$h_t^d = LSTM(h_{t-1}^d, [e(\tilde{y}_{t-1}); c_{t-1}]) \tag{2.15}$$

其中，$[e(\tilde{y}_{t-1}); c_{t-1}]$是指將標籤$\tilde{y}_{t-1}$的嵌入向量(embedding vector)和上下文向量c_{t-1}做連接，W_a和U_a是專注機制的參數。如圖2.10，上下文向量的計算先是由解碼器的隱藏層和編碼器每個時間點的隱藏層做一些矩陣運算，計算出每一個時間點的重要性e_{tj}(式2.12)，爾後，式2.13藉由此重要性的數值來算出對於每個時間點的權重α_{tj}，上下文向量便是編碼器隱藏層的加權平均(式2.12)，此上下文向量會加入記憶單元的隱藏層中，並算出解碼器的下一個時間點t的隱藏層(式2.15)。

解碼時的每一個時間點都會重新計算一次前後文向量，使不同的時間點的輸出可以根據不同的輸入序列資訊做計算，因此專注機制可以解決序列到序列模型因輸入序列過長造成梯度消失的問題。

時序採樣(Scheduled Sampling, SS)

在訓練序列到序列模型時，因為使用教師强迫進行訓練，解碼器皆是基於正確的前綴$\mathbf{y}^*_{<\mathbf{t}}$來做時間$t$的預測，然而，在測試階段時，模型可能會產生錯誤的前綴，若模型根據錯誤的結果繼續做預測，這個結果可能會更加糟糕，此現象又稱為曝光偏差(exposure bias)，因此，有人提出了時序採樣 [17]來解決此問題。

圖2.11是序列到序列模型的解碼器。在訓練模型階段，解碼器中的每個時刻t，皆會擲硬幣或隨機決定是否使用正確的標籤y_{t-1}或從上一時間點的預測的分

doi:10.6342/NTU201901512

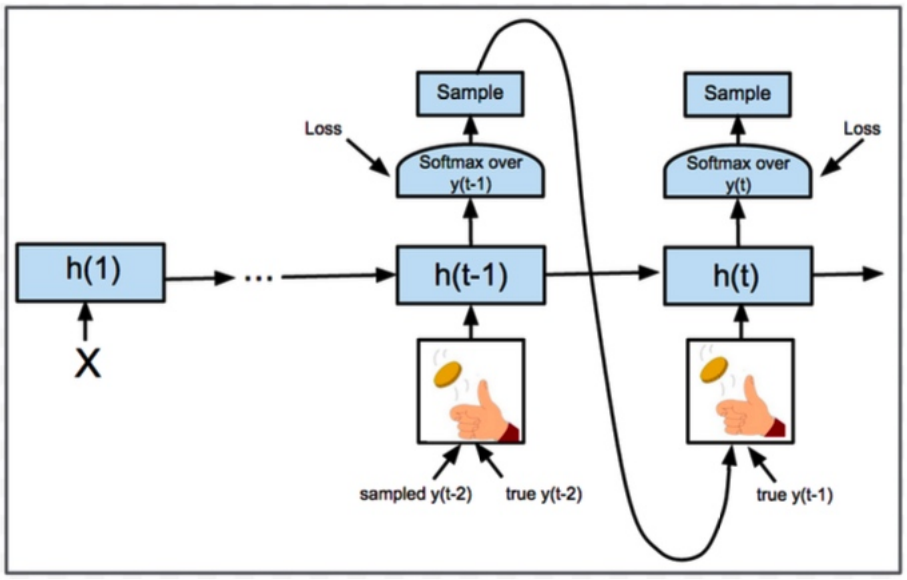


圖 2.11: 時序採樣示意圖

佈隨機採樣來作為時間點t的輸入。在訓練初期，使用正確標籤的機率會大一些，讓模型較好訓練;在訓練末期，此機率則會較小，讓模型能夠多學習發生錯誤前綴時的情況。

2.2.4 序列到序列模型應用於多標籤分類

序列到序列模型可以和2.1.2中提到的分類器鏈方法做結合。所謂分類器鏈是指分類器會基於先前預測標籤的結果，來預測之後的標籤，此特性和遞迴式類神經網路十分相似，因為此類神經網路會囊括先前的預測結果，來做下一步的預測。結合的方法便是由遞迴式類神經網路所構成的解碼器，順序性的預測每一個屬於此物件的標籤。若分類器的輸入也是序列，例如文章、音訊等，可以用遞迴式類神經網路來做編碼，此模型便是序列到序列模型，由編碼器編碼輸入訊息，再由解碼器依照順序解碼標籤。

因為序列到序列模型的輸出也是一串有順序性的序列，因此訓練時必須將標

doi:10.6342/NTU201901512

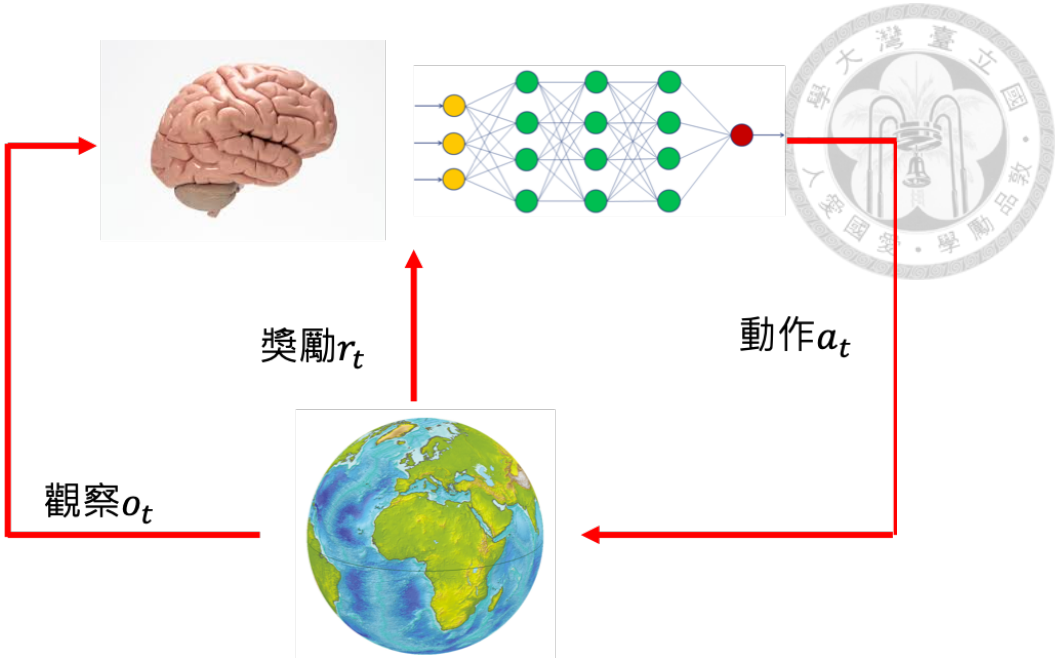


圖 2.12: 强化學習示意圖

籤進行排序，經前人的研究，將標籤依照出現次數由最多次到最少次來做排序，能使序列到序列模型有較好的表現。

2.2.5 强化學習(Reinforcement Learning, RL)應用於序列到序列模型

本章將介紹如何將强化學習中的訓練方法應用至序列到序列模型，會先介紹基本的强化學習，爾後介紹如何將强化學習應用於訓練序列到序列模型。

强化學習簡介

强化學習是機器學習中的一個領域，和先前提到的監督式學習(supervised learning)不同，它並沒有資料的標籤，學習的方法是由模型和環境(environment)互動，藉由環境給予的獎勵(reward)來做學習。

圖2.12中，大腦代表模型(agent)，在深層學習中，此模型便是一個類神經網

doi:10.6342/NTU201901512

路，地球代表環境，而獎勵是指環境給與模型的反饋，通常是由人事先定義的。訓練模型的目標便是訓練出一個模型來適應環境且最大化預期獎勵。在每個時間點t中，模型做出一個動作a_t，而環境會接收模型的動作而改變，並給予模型相對的獎勵r_t及觀察(observation)o_t，模型再根據觀察o_t來更新策略(policy)π_θ，θ是指模型的參數集，直到互動結束。其中，模型的狀態(state)s_t是指模型從開始互動後所有的觀察、動作和得到的獎勵所構成的函數:

$$s_t = f(o_1, r_1, a_1, ..., a_{t-1}, o_t, r_t) \tag{2.16}$$

而訓練模型其中一個著名的方法便是策略梯度(policy gradient)，這是一種策略基礎(policy based)的方法，其所想要最大化的就是根據策略所能得到的獎勵的期望值$J_\theta = \mathbb{E}[R(s,a)|\pi_\theta]$，其中，$R(s,a)$是指環境在狀態$s$對模型做動作$a$給予的獎勵，$\pi_\theta$則是動作的分佈，$\pi_\theta(s)$是指模型在狀態$s$下執行每個動作的機率分佈。為了能使用梯度下降法更新模型，需要對此數學式微分，而得到:

$$\nabla_\theta J(\theta) = E[\nabla_\theta log \pi_\theta R(s,a)] \tag{2.17}$$

此數學是可以用以下方式理解，若獎勵$R(s,a)$是正的，則鼓勵模型在狀態s時做動作a，反之若獎勵是負的，則使模型在狀態s時做動作a的機率少一點。

强化學習應用於序列到序列模型

如圖2.13，在基於遞迴式類神經網路的解碼器中，每個時間點t的隱藏層紀錄了狀態s_t，儲存了先前輸出和編碼器的資訊，而輸出的分佈可以看作是策略的分佈$\pi_\theta(s_t)$，而下一個時間點的輸入便是由策略分佈取樣得到a_t，而與環境互動結束時，環境會比對輸出序列和正確的標籤序列的分數，並給予模型獎勵R，而模型

doi:10.6342/NTU201901512

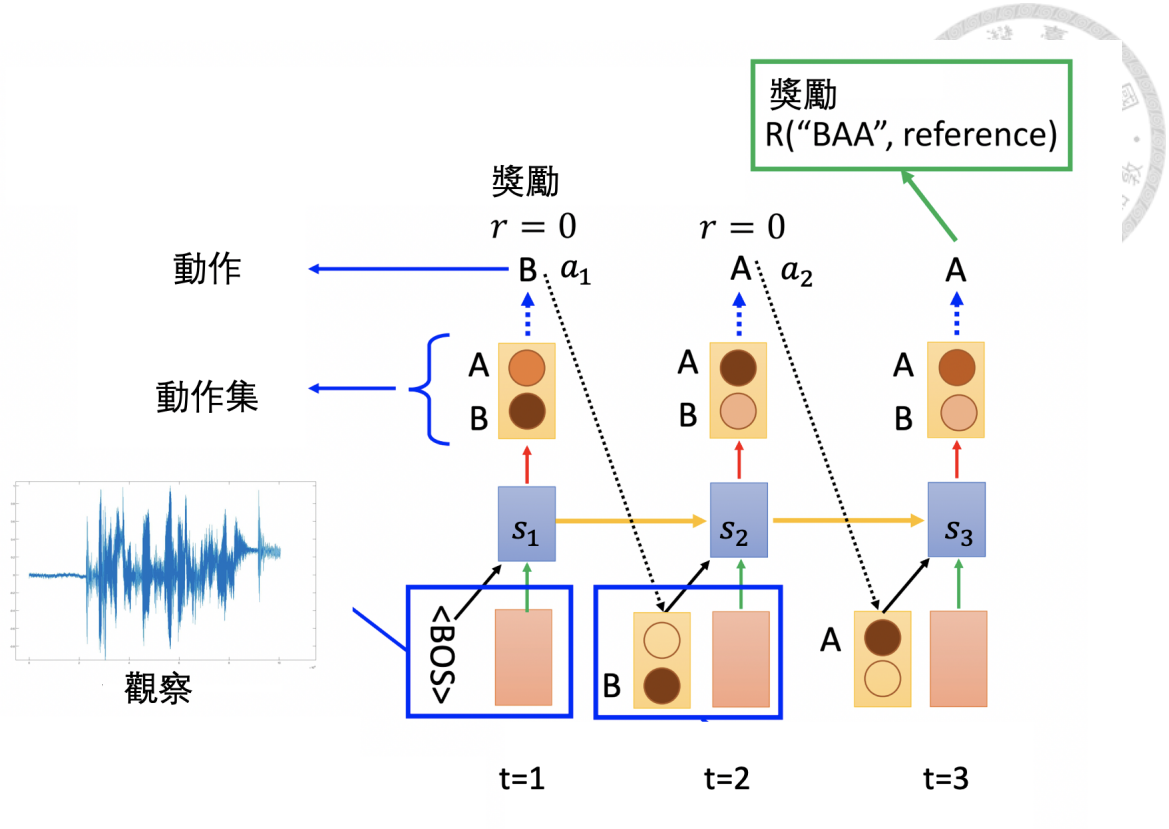


圖 2.13: 强化學習應用於序列到序列模型示意圖

更新的方法便是上一節講到的策略梯度。例如在語音辨識中，獎勵便是由詞錯誤率計算的，錯誤的越少，獎勵就會越多，模型輸出此序列的機率便會上升。

與基於最大似然估計的訓練方法最大的不同點是，其輸入的是由上一時間點輸出分佈做隨機取樣得到的結果，而不是正確標籤，且强化學習可以使用不可微分的值做為獎勵，而原本的訓練方法則不行。强化學習應用於訓練序列到序列模型一個有名的例子便是在序列到序列語音辨識模型上最小化詞錯誤率(Word Error Rate, WER)，因為詞錯誤率的算法是離散而無法微分的，需要使用强化學習的方法來最小化詞錯誤率。然而，因為此種方法皆需要模型做完預測後，才能拿到一個值進行更新，因此這種方法不容易訓練類神經網路，通常需要使用最大似然估計來作預訓練(pretrain)。

doi:10.6342/NTU201901512

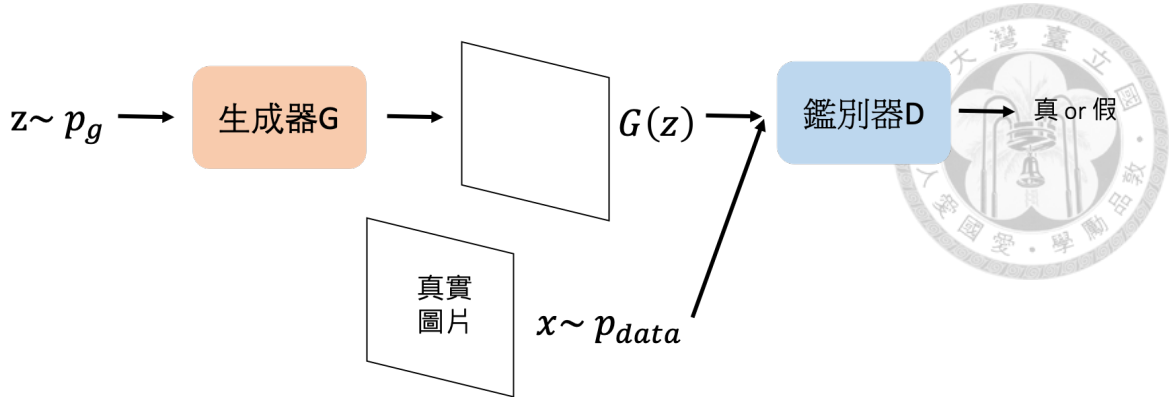


圖 2.14: 生成對抗網路基本架構

2.3 生成對抗網路(Generative Adversarial Network, GAN)

2.3.1 簡介

生成對抗網路 [3]是現今被廣泛研究的生成模型之一，除了在在圖片生成(image generation)，在風格轉換(style transfer) [18]、語音增强(speech enhancement) [19]、聲音轉換(voice conversion) [4]等領域都有生成對抗網路的應用。

模型架構

圖2.14是生成對抗網路的基本架構，其主要分成兩個部分，分別為生成器(generator)G和鑑別器(discriminator)D，生成器的輸入是從人定義的事前機率p_g隨機取樣的一個點，而輸出為一張圖片，並希望能生成接近真實的圖片，以欺騙鑑別器。而鑑別器的工作是判別此圖片是來自於生成器或從真實圖片分佈取樣得到的圖片$x \sim p_{data}$，生成器的目標是欺騙鑑別器，然而，鑑別器的目標是鑑別出生成器產生的圖片，兩者目標相反。生成式對抗網路便藉由兩個部分輪流更新來訓練。訓練完成時，理想狀況是生成器能夠學習到如何從事前機率分佈映射到真實的圖片分佈，換句話說，便是生成器生成的圖片分佈，和真實圖片的分佈相同，在這個理想情況下，鑑別器便無法分辨真實和生成的圖片。

doi:10.6342/NTU201901512

減損函數與訓練方法

接下來要介紹生成對抗網路的數學式，也就是生成對抗網路的價值函數(Value function)，定義如下：

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \quad (2.18)$$

其中，$D(x)$的輸出表示此圖片是真實圖片的機率，介於零到一之間。式2.18的第一項表示鑑別器需最大化真實圖片的機率，第二項表示鑑別器需最小化生成器圖片的機率，然而生成器需要最大化此值。式中的$\min_G \max_D$則表示D和G是互相拮抗的。在一般的情況下，G和D是交互訓練的，訓練方法如下圖2.15描述: 其中，k是生成器和鑑別器訓練次數的比例，上面的演算法是更新k次

1: **for** number of iterations **do**

2: **for** k steps **do**

3: 從事前機率$p_g(z)$中抽樣出m筆資料$\{z_1, z_2, ..., z_m\}$

4: 藉由梯度下降更新生成器： $\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} [log(1-D(G(z_i)))]$

5: **end for**

6: 從事前機率$p_g(z)$中抽樣出m筆資料$\{z_1, z_2, ..., z_m\}$

7: 從真實資料$p_{data}(z)$中抽樣出m筆資料$\{x_1, x_2, ..., x_m\}$

8: 藉由梯度下降更新鑑別器： $\nabla_{\theta_D} \frac{-1}{m} \sum_{i=1}^{m} [log(D(x_i)) + log(1-D(G(z_i)))]$

9: **end for**

圖 2.15: 生成對抗網路的訓練方法

生成器後，更新一次鑑別器，通常k是一個敏感的超參數(hyperparameter)，對訓練成功與否有相當大的影響。

doi:10.6342/NTU201901512

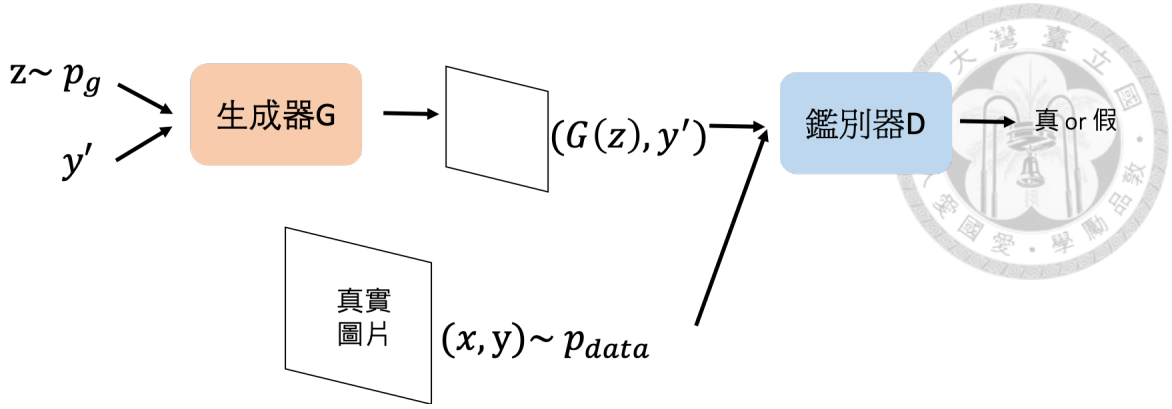


圖 2.16: 條件式生成對抗網路基本架構

2.3.2 條件式生成對抗網路

簡介

在生成式對抗網路中，生成器的輸入是事前機率隨機取樣一個點，然後生成圖片。然而，真實的圖片卻有非常多種類別，若需要生成特定種類的圖片時，例如貓、狗等，則需要在訓練過程中，使用圖片的類別標籤，使生成對抗網路了解圖片和標籤的對應關係。

模型架構

如圖2.16，訓練條件式生成對抗網路需要有標記的圖片標籤配對(x, y)，加入條件的方法是在生成器和鑑別器的輸入都加入條件y(或y')，生成器必須產生符合條件y(或y')的圖片，鑑別器不只需要考慮圖片的真實程度，也需要考量輸入圖片是否符合條件y(或y')。

減損函數與訓練方法

條件式生成對抗網路的價值函數可以寫成下式：

doi:10.6342/NTU201901512

$$\min_G \max_D V(G,D) = \mathbb{E}_{(x,y)\sim p_{data}}[log(D(x|y))] + \mathbb{E}_{z\sim p_z(z),y'\sim p_y}[log(1-D(G(z)|y'))] \tag{2.19}$$

訓練方法和一般的生成對抗網路大同小異，只是在訓練生成器時，除了要從p_z取樣，還要隨機選取一個類別y'，使生成器生成屬於類別y'的圖片，此y'也會是鑑別器的輸入，在從真實數據取樣時，也需要取樣和圖片相符的類別y。生成器需要最小化式2.19的第二項，然而，鑑別器兩項都需要最大化。

2.3.3 霍式生成對抗網路(Wasserstein GAN)

然而，上述都只是最基本的生成對抗網路，基本的模型有許多問題，例如其訓練過程十分困難、對於超參數十分敏感、模式崩潰(mode collapsed)等等，其中，模式崩潰是指生成的圖片缺乏多樣性，生成器只能生成特定幾個種類的圖片。霍式生成對抗網路 [20]解決了底下幾個問題：

- 讓生成對抗網路的訓練更加穩定，使其對超參數較不敏感。
- 解決了模式崩潰的問題，確保了生成圖片的多樣性。

基本的生成對抗網路遇到的問題

霍式生成對抗網路的作者在論文中提到了基本的生成對抗網路遇到的問題。在最優(optimal)鑑別器的情況下，其實鑑別器的數值是衡量真實分佈p_{data}和生成的分佈p_g的JS散度(Jensen-Shannon Divergence)(式2.20)，但是JS散度在衡量兩分佈的距離時，若兩分佈沒有交集，他們的JS散度就會是一個定值$log2$，因此，微分後便會是0，梯度便會消失。因此訓練初期兩分佈相差甚遠或鑑別器訓練得較好時，就有可能遇到這個狀況，導致訓練變得困難。

doi:10.6342/NTU201901512

$$\max_D V(G,D) = -2log2 + 2JSD(p_{data}||p_g) \tag{2.20}$$

霍氏距離

在霍氏生成對抗網路中，作者引用了霍氏距離來代替JS散度，並將其作為優化目標，霍氏距離相對於JS散度平滑，因此較不容易有梯度消失的問題。其中，霍氏距離可以用以下式子表達:

$$W(p_g, p_{data}) = \sup_{\|f\|_L \leq 1} \{\mathbb{E}_{x\sim p_g}[f(x)] - \mathbb{E}_{x\sim p_{data}}[f(x)]\} \tag{2.21}$$

其中，$\|f\|_L \leq 1$代表所有1-Lipschitz的函數，也就是所有滿足$|f(x_1) - f(x_2)| \leq |x_1 - x_2|, \forall x_1, x_2$的函數，而sup表示最小上界(supremum)，式2.21代表對所有滿足1-Lipschitz的函數取到$\mathbb{E}_{x\sim p_g}[f(x)] - \mathbb{E}_{x\sim p_{data}}[f(x)]$的上界。 然而，若我們使用一組參數集為w的類神經網路f_w來表示1-Lipschitz的函數，則式2.21可以近似成:

$$W(p_g, p_{data}) = \max_{w:\|f_w\|_L \leq 1} \{\mathbb{E}_{x\sim p_g}[f(x)] - \mathbb{E}_{x\sim p_{data}}[f(x)]\} \tag{2.22}$$

因此，這個式子就和就可以和原本生成對抗網路的式2.18十分相似，將f_w改寫成D後，霍氏生成對抗網路的價值函數便可以寫成式2.23。

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[D(x)] - \mathbb{E}_{z\sim p_z(z)}[D(G(z))] \tag{2.23}$$

而因為f_w需要是1-Lipschitz的函數，作者限制所有類神經網路的參數不得超過某個範圍$[-c, c]$，因此，鑑別器D的微分值也會被限制在某個常數內，就滿足

doi:10.6342/NTU201901512

了1-Lipschitz的條件，在實作上也十分簡單，只要將每次更新完的參數超過範圍$[-c, c]$的部分扣掉就好了。

因此，霍式生成對抗網路的改進部分主要是下列幾點:

- 去掉鑑別器最後的sigmoid層，因為不需要限制輸出至0到1之間。
- 改寫價值函數，使鑑別器改為計算霍氏距離。
- 限制鑑別器的所有參數在更新完後皆不超過範圍$[-c, c]$。

增進版霍式生成對抗網路(Improved Wasserstein GAN)

增進版霍式生成對抗網路 [21]是改良版的霍式生成對抗網路，其作者發現了原本霍式生成對抗網路為了要限制1-Lipschitz的條件，限制了參數的範圍，然而在實際訓練時，模型對於限制範圍的大小的數值十分敏感，且大部分的參數都會集中在範圍的邊界，因此，他提出了增進版霍式生成對抗網路，使用了不同的方法限制1-Lipschitz的條件。

作者提出的方法是梯度懲罰(Gradient Penalty, GP)，原理是限制鑑別器的梯度，若鑑別器的梯度小於一 ($\|\nabla_x D(x)\| \leq 1, \forall x$)，則鑑別器必然滿足1-Lipschitz的條件，因此梯度懲罰鼓勵鑑別器的梯度越接近1越好，數學式如下:

$$L_{gp} = \mathbb{E}_{\hat{x} \sim p_{penalty}}[(\|\nabla_x D(\hat{x})\| - 1)^2] \tag{2.24}$$

其中，$p_{penalty}$是指梯度懲罰分佈。下列對此做簡短說明，因為若要限制空間中的每一個點的梯度是十分困難的。然而，在此空間中較重要的點就是p_g和p_{data}中的點，還有這兩個分佈中間的點，因此，作者認為只需要限制這三個範圍內的點的梯度，因此做法便是隨機取樣p_g和p_{data}中的點，並使用此兩點隨

doi:10.6342/NTU201901512

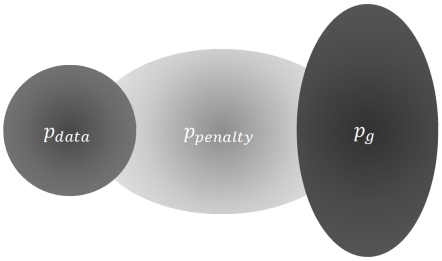


圖 2.17: 梯度懲罰分佈

機內差得到一個新的點，並限制這個點梯度的大小，在實作中此方法較為容易，且有好的效果，如圖2.17。

表2.18是增進版霍式生成對抗網路的訓練方法，其中λ是一個超參數，調控梯度懲罰損失的大小。

doi:10.6342/NTU201901512

1: **for** number of iterations **do**

2: **for** k steps **do**

3: 從事前機率$p_g(z)$中抽樣出m筆資料$\{z_1, z_2, ..., z_m\}$

4: 藉由梯度下降更新生成器： $\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} [-D(G(z_i))]$

5: **end for**

6: 從事前機率$p_g(z)$中抽樣出m筆資料$\{z_1, z_2, ..., z_m\}$

7: 從真實資料$p_{data}(z)$中抽樣出m筆資料$\{x_1, x_2, ..., x_m\}$

8: 藉由梯度下降更新鑑別器： $\nabla_{\theta_D} \frac{-1}{m} \sum_{i=1}^{m} [D(x_i) - D(G(z_i))]$

9: 從均勻分佈$U(0, 1)$中隨機取樣m個點$\{\alpha_1, \alpha_2, ..., \alpha_m\}$

10: 計算內差的點$\{\hat{x}_i = \alpha_i z_i + (1 - \alpha_i) x_i | i = 1, 2, ..., m\}$

11: 藉由梯度下降更新鑑別器： $\lambda \nabla_{\theta_D} \sum_{i=1}^{m} [(\|\nabla_x D(\hat{x}_i)\| - 1)^2]$

12: **end for**

圖 2.18: 增進版霍式生成對抗網路的訓練方法

2.4 本章總結

本章一開始先介紹多標籤分類的基本算法，爾後，從基本的類神經網路開始，介紹遞迴式類神經網路及序列到序列模型，以及其如何應用在多標籤分類問題上，也稍微簡介了如何利用强化學習訓練序列到序列模型。在本章的後半段，主要是介紹生成對抗網路，從條件式生成對抗網路的架構，並演進到霍氏生成對抗網路，以及說明其解決了原本生成對抗網路的什麼問題。

doi:10.6342/NTU201901512

第 三 章　以生成式對抗網路幫助多標籤分類器

3.1 簡介

3.1.1 研究動機

多標籤分類是一個基本卻十分困難的問題，並且有許多應用，像是音樂中的曲風分類、文章分類、醫療疾病分類等等，相對於單標籤分類，多標籤分類器不只需要了解物件和標籤之間的關係，更需要了解標籤之間的關聯性。例如在一張圖片中，海洋和沙灘比較容易一起出現，而沙灘和大象就不容易一起出現在同一張圖片裡。

多標籤分類在深層學習中有許多方法，像是利用深層神經網路搭配對數機率回歸，近年來，也有人利用機率圖模型(probabilistic graphical networks)、遞迴式類神經網路來模擬標籤間的關聯性。而在本論文，將介紹如何使用生成式對抗網路來模擬此標籤關聯性。

本章介紹的模型是以條件式生成網路作為基礎，分類器在此扮演生成器的角色，其輸入是一個物件，而輸出是屬於此物件的標籤集，鑑別器則需要學習標籤之間的關聯性，其輸入是標籤集和一個物件，而鑑別器需要輸出一個值。此標籤集可能來自真實的資料也可能來自生成器的輸出，而鑑別器需要學習如何分辨此標籤集是從生成器產生還是從真實的資料，因此，鑑別器不只需要了解標籤之間的關聯性，還需要了解物件和標籤集的關係，才能學習到如何分辨真偽。而在此生成對抗網路的架構下，分類器(生成器)也需要學習如何從輸入的物件，讓產生

doi:10.6342/NTU201901512

之標籤集的關聯性更貼近真實資料，以欺騙鑑別器。如同一般的生成對抗網路，分類器和鑑別器會交替訓練。

本論文提出的架構是和分類器的結構無關的，因此，我們相信這個架構可以套用在其他模型上，使其他模型學習如何模擬標籤間的關聯性。模型架構將會在章節3.2中說明，而生成對抗網路中的分類器和鑑別器的訓練，將會在章節3.3.1和章節3.3.2說明，最後的實驗部分會在章節3.4做說明。

3.2 本論文所提出之模型

本論文所提出之模型如圖3.1，在此，以多標籤影像分類作為解釋多標籤分類的一個例子，x是指輸入影像，而對應的正確答案標籤集$y \in \{0,1\}^{|S|}$, $|S|$是指標籤的種類數。

模型包含分類器(生成器)和鑑別器，生成器G是一個輸出層是S函數作為激活函數的類神經網路。此生成器在此可以有很多種架構，例如，VGG-16 [22], Inception_v3 [23], Resnet-101 [24], 或Resnet-152 [24]等。G 將一張圖片x當作輸入，並輸出圖片x的標籤集機率分佈$\tilde{y} \in R^{|S|}$，$\tilde{y}$的每一維皆介於0和1之間，代表具有某個標籤的機率。在訓練時，預測的標籤集$\hat{y}$會從$\tilde{y}$取樣，取樣的機率會根據每個標籤輸出機率的大小，而有所不同。在測試階段，則不會有取樣的步驟，而是選輸出機率大於0.5的標籤當作模型預測的標籤集。

鑑別器D的輸入是標籤集y(或$\hat{y}$)和圖片x，並產出一個分數$D(y,x)$(或$D(\hat{y},x)$)，代表這個標籤集有多"真實"或標籤集和此圖片的匹配程度。其中，特徵抽取器f_{ext}是用來抽取圖片x的特徵$z = f_{ext}(x)$，並包含在鑑別器裡。爾後接了一個全連接(fully-connected)的類神經網路，此全連接層的輸入是z和y(或$\hat{y}$)，並輸出一個值$D(y,x)$(或$D(\hat{y},x)$)。在測試階段，不需要鑑別器，只需要分類器的輸出作為預

doi:10.6342/NTU201901512

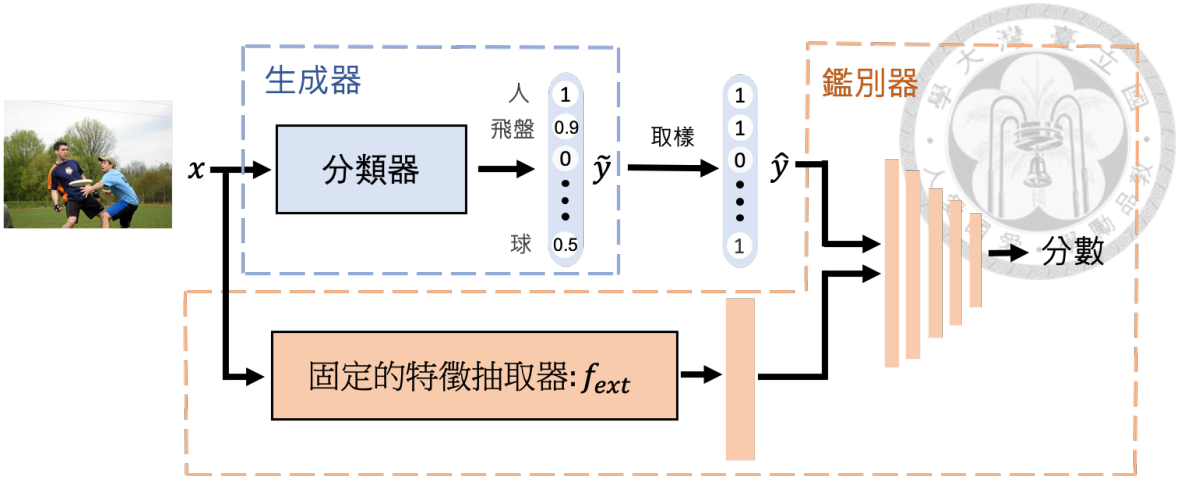


圖 3.1: 模型架構示意圖

測結果。

doi:10.6342/NTU201901512

3.3 模型之訓練方式

3.3.1 分類器之訓練方式

訓練分類器的方法如圖3.2，分為兩個部分，一是使用對數機率回歸的減損函數，二是增進版霍氏生成對抗網路損失，總和的減損函數則是兩者以一加權值相加，接下來將逐一說明。

對數機率回歸的減損函數

如同一般的多標籤分類器，本論文也使用了對數機率回歸的損失$\mathcal{L}_{logistic}$，在給定正確標籤集$y = [y_1, y_2, ..., y_{|S|}]^T$ 和輸出的機率分佈$\tilde{y} = [\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_{|S|}]^T$，減損函數如下:

$$\mathcal{L}_{logistic} = \mathbb{E}_{(x,y)\sim data}[\sum_{i=1}^{|S|} y_i log\, \tilde{y}_i + (1 - y_i) log(1 - \tilde{y}_i)], \tag{3.1}$$

其中，x, y是⟨圖片、標籤集⟩配對，$\tilde{y}$是指分類器的輸出標籤機率分佈。因為在此損失函數中，所有的標籤是獨立的，使用此減損函數不一定能夠保證模型能夠學習到標籤之間的關聯性，因此，本論文提出了一個基於生成對抗網路的減損函數。

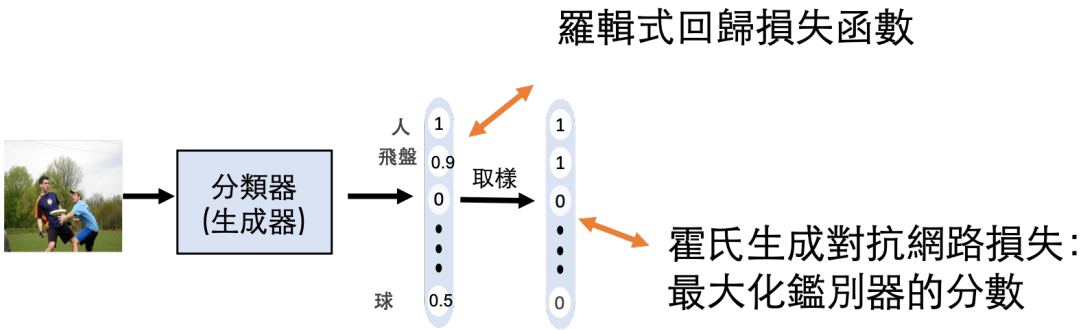


圖 3.2: 分類器的訓練示意圖

doi:10.6342/NTU201901512

增進版霍氏生成對抗網路減損函數

基於增進版霍氏生成對抗網路減損函數的減損函數如下式:

$$\mathcal{L}_G = -\mathbb{E}_{x\sim data,\hat{y}\sim G(x)}[D(\hat{y},x)], \tag{3.2}$$

其中，x是從真實資料分佈取樣的圖片，也是分類器的輸入。$\hat{y}$則是從輸出標籤分佈$\tilde{y}$取樣而得的標籤集。$D(\hat{y},x)$是鑑別器基於⟨圖片,標籤集⟩輸出的分數。從式3.2，分類器不只學習到要最小化$\mathcal{L}_{logistic}$，也需要藉由最大化$D(\hat{y},x)$，學習產生合理的標籤集，以欺騙鑑別器。因為D是將整個標籤集作為輸入，可以利用標籤之間的關聯性來判斷是否為生成的標籤，分類器也需要學習模擬標籤間的關聯性。

分類器的減損函數

總和的減損函數如下:

$$\mathcal{L}'_G = \mathcal{L}_G + \alpha\mathcal{L}_{logistic}, \tag{3.3}$$

其中，α是一個加權值，決定兩減損函數間的比例。

因為式3.2需要G和D皆是可以微分的，且$\hat{y}$需要是離散的分佈，因此，我們在取樣階段，使用了岡氏軟性最大化(Gumbel-softmax)技法並套用在伯努利分佈(Bernoulli distribution)，且在分類器的每個標籤的機率分佈都使用。岡氏軟性最大化技法藉由再參數化(reparameterization)，使得我們可以從機率分佈上做取樣，並且使取樣過程是可微分的。

3.3.2 鑑別器之訓練方式

訓練鑑別器的方式如下，(真實的標籤集y, 圖片x)是從真實資料中取樣的，並作為

doi:10.6342/NTU201901512

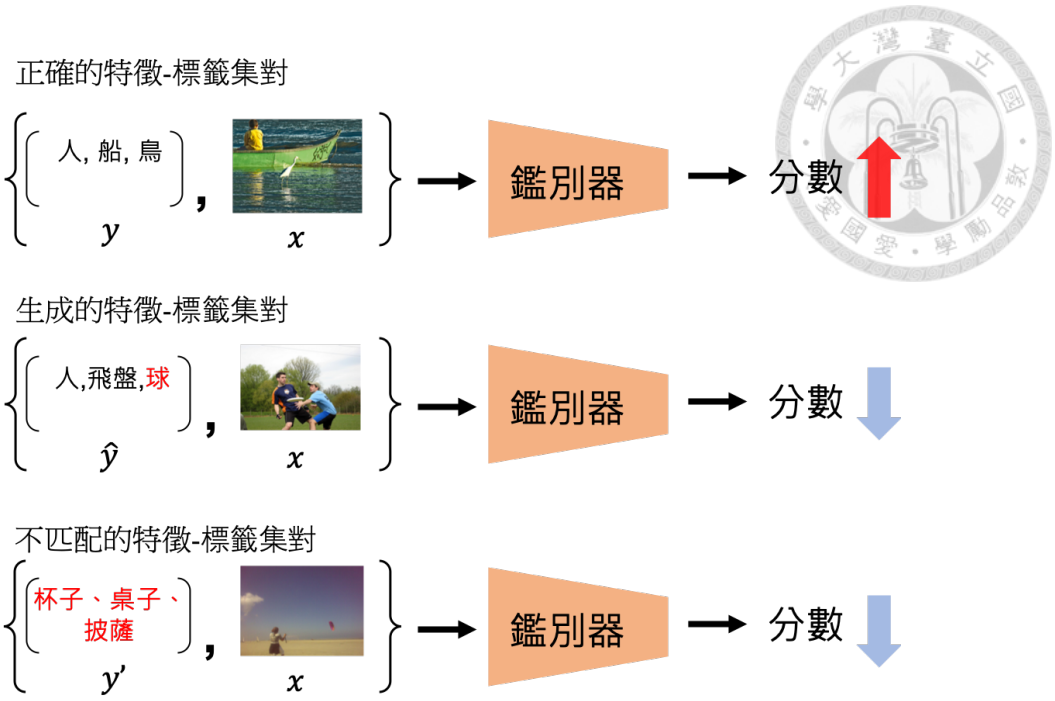


圖 3.3: 訓練鑑別器的三種輸入，第一種是正確的特徵-標籤集對，兩者皆是從真實數據取樣而得，第二種是生成的特徵-標籤集對，標籤集是從生成器的輸出而得，第三種是不匹配的特徵-標籤集對，特徵和標籤集雖然皆是從真實數據而得，卻是不匹配的。鑑別器需要學習最大化第一種配對的分數，而最小化後兩個配對的分數。

鑑別器的正範例(positive example)，鑑別器需要最大化正範例的分數。相反的，鑑別器需要最小化負範例(negative example)的分數，其中，我們不只使用了生成器所產生的(生成的標籤集$\hat{y}$, 圖片x)作為負範例，還有不匹配的配對(隨機取樣的標籤集y',圖片x)，其中，y'是從資料隨機取樣的，和原本圖片x的標籤並不相同，此方法稱為負取樣(negative sampling) [25]，圖3.3解釋了三種鑑別器的輸入。

而鑑別器的損失函數如下式，是使用了增進版霍氏生成對抗網路的減損函數。

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{(x,y)\sim data}[D(y,x)] + \frac{1}{2}\mathbb{E}_{x\sim data,\hat{y}\sim G(x)}[D(\hat{y},x)] \\ &+ \frac{1}{2}\mathbb{E}_{x\sim data,y'\sim data}[D(y',x)] + \lambda\mathcal{L}_{gp}\end{aligned} \tag{3.4}$$

doi:10.6342/NTU201901512

其中，第一項是正確的特徵-標籤集對，兩者皆是從真實資料取樣而得，鑑別器需要最大化此分數，第二項是生成的特徵-標籤集對，標籤集是從生成器的輸出而得，第三項是不匹配的特徵-標籤集對，是從資料中負採樣而得，鑑別器需最小化後兩者的分數。其中，鑑別器必須要從兩種錯誤做學習，一是從分類器生成的標籤集，二是不匹配的特徵-標籤集對，這能使鑑別器更能學習到特徵和標籤集之間的關係。最後一項是增進版霍氏生成對抗網的梯度懲罰，是為了讓訓練更穩定必須加入的減損函數：

$$\mathcal{L}_{gp} = \mathbb{E}_{x\sim data,\hat{y}\sim G(x),y^*\sim interp(\hat{y},y^*)}[(\|\nabla D(y^*,x)\|-1)^2] \tag{3.5}$$

其中，λ是加權值，y^*是藉由真實標籤集和生成標籤集以一個介於0到1的值做差值而得到的，並對y^*做梯度懲罰，此項對於穩定的訓練是必須加入的。

3.4 系統評估

3.4.1 實驗設定

模型參數、訓練方式、實驗資料集、機器評分機制將會在以下詳細介紹，而實驗中，也比較了四種也有考慮標籤關聯性的方法，分別是WARP [26], CNN-RNN [27], 無序且有專注力機制的CNN-RNN (Att-RNN) [28], and RLSD [29]。為了證明我們提出的架構是通用的，我們實作了4種圖片分類器的架構，分別為VGG-16 [22], Inception_v3 [23], Resnet-101 [24], 或Resnet-152 [24]，並比較有無使用增進版霍氏生成對抗網路的結果。

模型架構和訓練方式

實驗中的分類器事先預訓練於大型影像分類資料庫Imagenet，爾後我們移除了分

doi:10.6342/NTU201901512

類器的最後一層，作為固定的特徵抽取器f_{ext}，在訓練時，特徵抽取器的參數並不會被更新。在鑑別器中，影像的特徵$z = f_{ext}(x)$和標籤集會分別被線性投射到256維的向量，然後兩者會被連結成512維的向量，接著送入8層具有溢出整流線性單元(leaky relu)的全連結層。根據我們實驗的觀察，在生成對抗網路的訓練中，我們更新鑑別器3次才會更新分類器一次。在每次更新鑑別器時，我們會隨機取樣3個批次(batch)的資料，分別為正確的特徵-標籤集對、生成的特徵-標籤集對和不匹配的特徵-標籤集對，如圖3.3。

訓練分類器和鑑別器時，我們使用了亞當優化器(Adam optimizer)，學習率是0.0001，對數機率回歸損失和生成器損失的加權值λ是10，而岡氏軟性最大化使用的溫度T是0.9。

我們在訓練時，我們執行了資料增強(data augmentation) [30]。詳細的做法如下，我們先將圖片改變大小至256x256，爾後，我們會分別從四個角落以及圖片中央，切取了5種不同大小的圖片，大小分別是{256,224,192,168,128}，最後再將切取的圖片調整大小至224x224。而在測試階段，我們直接使用224x224的圖片做為輸入。

實驗資料集

實驗使用了兩個多標籤圖像分類的資料集，分別是80種標籤的MS-COCO和81種標籤的NUS-WIDE，兩個資料集的一些範例如表3.4。

MS-COCO是Microsoft COCO的縮寫，是一個用來做物件偵測(object detection)、物件分割(object segmentation)、影像標題(image captioning)的大型資料集，並且也有使用多標籤分類，共有80個標籤類別，訓練集有82,081張圖片、驗證集有40137張圖片，因為測試集並沒有提供，我們像 [27,28]一樣皆使用驗證集作為

doi:10.6342/NTU201901512

測試集。

NUS-WIDE是一個網路圖片大型資料集，具有269,648張圖片，皆來自Flickr，這些圖片被人標注了總共81種標籤，我們考慮WARP [26]和Att-RNN [28]的作法，將沒有標注標籤的圖片刪除，並使用了150,000張圖片作為訓練集，59,347張圖片作為驗證集。

doi:10.6342/NTU201901512

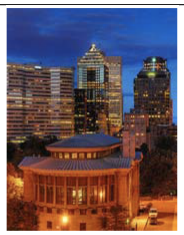
文章	標籤集
MS-COCO	
	人、球、棒球棍、棒球手套
	酒杯、杯子、叉子、刀子、披薩、餐桌
	椅子、沙發、床、書
NUS-WIDE	
	機場、雲、軍隊、飛機、天空
	雲、人、天空、水
	建築物、市容、夜晚、天空

圖 3.4: MS-COCO和NUS-WIDE兩個資料集的一些範例。

doi:10.6342/NTU201901512

機器評分機制

實驗中，我們使用了兩類評分機制，每類分別有三種，分別是微精確率(micro precision)、微召回率(micro recall)、微F1分數(micro F1 score)和宏精確率(macro precision)、宏召回率(macro recall)、宏F1分數(macro F1 score)。以下介紹他們的計算方式:

假設總共有L種標籤，則真陽(True Positive, TP)、真陰(True Negative, TN)、偽陽(Flase Positive, FP)和偽陰(Flase Negative, FN)的計算方式為:

- 真陽：預測為Positive且預測準確True，例如預測有下雨且真的下了。
- 真陰：預測為Negative且預測準確True，例如預測不會下雨且真的沒下。
- 偽陽：預測為Positive但預測錯False，例如預測有下雨不過實際上沒下。
- 偽陰：預測為Negative但預測錯False，例如預測不會下雨不過實際下雨了。

而對於每一種標籤，我們先分別計算每一類的精確率、召回率:

$$precision_i = \frac{TP_i}{TP_i + FP_i} \tag{3.6}$$

$$recall_i = \frac{TP_i}{TP_i + FN_i} \tag{3.7}$$

則宏精確率(宏召回率)的計算方式便是每一種標籤的精確率(召回率)的平均:

$$precision_{ma} = \frac{1}{L}\sum_{i=1}^{L} precision_i \tag{3.8}$$

$$recall_{ma} = \frac{1}{L}\sum_{i=1}^{L} recall_i \tag{3.9}$$

doi:10.6342/NTU201901512

有了宏精確率和宏召回率，便可以計算宏F1分數:

$$F1_{ma} = 2\frac{precision_{ma} \times recall_{ma}}{precision_{ma} + recall_{ma}} \tag{3.10}$$

因為每類標籤的數目可能會不一樣，在宏F1分數的計算方式中，有些標籤較罕見，但卻和較頻繁的標籤具有相同的權重，因此，微F1分數變考慮了每種標籤的種類數目，較頻繁出現的標籤會對F1分數有較大的影響，接下來，我們來計算微精確率和微召回率，

$$precision_{mi} = \frac{\sum_{i=1}^{L} TP_i}{\sum_{i=1}^{L}(TP_i + FP_i)} \tag{3.11}$$

$$precision_{mi} = \frac{\sum_{i=1}^{L} TP_i}{\sum_{i=1}^{L}(TP_i + FN_i)} \tag{3.12}$$

$$F1_{mi} = 2\frac{precision_{mi} \times recall_{mi}}{precision_{mi} + recall_{mi}} \tag{3.13}$$

一般來說，微F1分數和宏F1分數較準確率和召回率為重要。

3.4.2 實驗結果

MS-COCO

表3.1是MS-COCO的實驗結果，我們發現Inception_v3, Resnet-101和Resnet-152在沒有使用增進版霍氏生成對抗網路(WGAN-gp)的情況下，在微/宏F1分數的結果便已超越前人的結果，可能是因為現今深層類神經網路的架構已十分發達。更進一步地，這四種模型使用本章提出的方法訓練後，皆得到了較高的微/宏F1分數，這些證明了若使模型學會標籤間的關聯性，便可以增進多標籤分類模型的表現。然

doi:10.6342/NTU201901512

方法	微精確率	微召回率	微F1分數	宏精確率	宏召回率	宏F1分數
前人的方法						
WARP	59.3	52.5	55.7	59.8	61.4	60.7
CNN-RNN	66.0	55.6	60.4	69.2	66.4	67.8
Att-RNN	71.6	54.8	62.1	74.2	62.2	67.7
RLSD	67.6	57.2	62.0	70.1	63.4	66.5
提出的方法						
VGG-16	74.2	44.8	56.0	77.6	52.5	62.6
+ WGAN-gp	62.6	58.3	60.4	67.5	63.3	65.3
Inception_v3	76.4	52.8	62.4	80.0	58.8	67.8
+ WGAN-gp	70.5	58.2	63.8	73.2	63.8	68.2
Resnet-101	76.2	53.4	62.8	**80.8**	58.9	68.1
+ WGAN-gp	70.5	**58.7**	**64.0**	72.3	**64.6**	68.2
Resnet-152	**76.6**	53.9	63.3	80.6	59.6	**68.6**
+ WGAN-gp	71.4	57.9	63.9	73.6	64.2	**68.6**

表 3.1: MS-COCO多標籤分類的實驗結果。其中，WARP, CNN-RNN, 和RLSD的結果是取前三高分標籤作為預測標籤集。WGAN+gp是指增進版霍氏生成對抗網路

doi:10.6342/NTU201901512

而，模型的進步量會隨著基準模型(baseline model)(沒有使用WGAN-gp)的增强而減少，例如，對於VGG-16模型，使用WGAN-gp有4.4%和2.7%微/宏F1分數的進步量，然而對於Resnet-152，只有0.6%和0% 微/宏F1分數的進步量，這可能是因為對於越深層的網路，例如Resnet-152，可能模型自身已學習到了標籤之間的關聯性，因此限制了WGAN-gp的進步量。另外，我們發現四個基準模型平均來說每張圖片預測了2.09個標籤，然而對於有使用WGAN-gp訓練的模型，卻預測了2.61個標籤，比基準模型多出了大約25%，因此有使用WGAN-gp訓練的模型相對於基準模型，有較高的召回率但精確率卻較低。

NUS-WIDE

表3.2是NUSWIDE的實驗結果，模型的表現和MSCOCO有相同的趨勢。

3.4.3 實驗結果分析

在這個章節中，我們想要分析模型在遇到沒有在訓練集見過的標籤集組合時的表現如何。因此，我們將MS-COCO的測試集依照標籤集切成兩部分，第一部分S_{seen}圖片的標籤集組合皆是在訓練集有出現過的，共33,139張圖片，第二部分S_{unseen}的圖片的標籤集組合是在訓練集沒有出現過的，共6,998張圖片。下表3.3是模型VGG-16、Resnet-101以及兩者使用WGAN-gp進行訓練的模型，分別在測試集S_{seen}, S_{unseen}的微/宏F1分數。比較4個模型在S_{seen}, S_{unseen}中的表現，顯而易見的，可以發現模型在有看過的標籤集的S_{seen}上表現較好，而在S_{unseen}上表現較差。然而在S_{unseen}中，不管是VGG-16或Resnet-101，使用WGAN-gp進行訓練皆帶來相當的進步量，這可能是因為使用WGAN-gp的模型學習標籤間的關聯

doi:10.6342/NTU201901512

方法	微精確率	微召回率	微F1分數	宏精確率	宏召回率	宏F1分數
前人的方法						
WARP	31.7	35.6	33.5	48.6	60.5	53.9
CNN-RNN	40.5	30.4	34.7	49.9	61.7	55.2
Att-RNN	59.4	50.7	54.7	69.0	71.4	70.2
RLSD	44.4	49.6	46.9	54.4	67.6	60.3
提出的方法						
VGG-16	53.3	24.9	33.9	73.9	59.6	66.0
+ WGAN-gp	51.6	34.3	41.2	68.8	67.3	68.1
Inception_v3	67.9	44.1	53.5	74.7	64.8	70.3
+ WGAN-gp	62.4	50.5	**55.8**	71.4	70.9	**71.2**
Resnet-101	67.0	44.0	53.1	76.3	65.0	70.2
+ WGAN-gp	59.6	**51.8**	55.4	68.9	**72.8**	70.8
Resnet-152	**69.1**	41.8	52.1	**75.9**	65.1	70.1
+ WGAN-gp	65.2	46.2	54.1	71.3	70.8	71.1

表 3.2: NUS-WIDE多標籤分類的實驗結果。其中，WARP, CNN-RNN, 和RLSD的結果是取前三高分標籤作為預測標籤集。WGAN+gp是指增進版霍氏生成對抗網路

doi:10.6342/NTU201901512

性學得較好，因此，在面對標籤集組合未曾出現的圖片時，能夠基於已知的標籤推論出其他標籤。然而對於Resnet-101，使用WGAN-gp進行訓練雖稍微降低了在S_{seen}的表現，但在S_{unseen}上有較强的廣泛化能力，並且在原本MS-COCO綜合兩部分的測試集也表現較好(表3.1)。

使用WGAN-gp進行訓練的模型，廣泛化能力較好的推論，也可以在表3.4中再次印證，表中的值代表模型產生出沒有在訓練集看過的標籤集的種類數。顯而易見的，使用WGAN-gp進行訓練後，模型可以產生出較多種沒有見過的標籤集組合，因此具有較强的廣泛化能力。

	S_{seen}		S_{unseen}	
方法	微F1分數	宏F1分數	微F1分數	宏F1分數
(1) VGG-16	0.546	0.651	0.358	0.446
(2) VGG-16 + WGAN-gp	0.617	0.694	0.456	0.537
(3) Resnet-101	**0.665**	**0.735**	0.491	0.550
(4) Resnet-101 + WGAN-gp	**0.665**	0.725	**0.520**	**0.577**

表 3.3: 模型VGG-16、VGG-16 + WGAN-gp、Resnet-101和Resnet-101 + WGAN-gp在資料集MS-COCO的S_{seen}, S_{unseen}上分別的微/宏F1分數。其中S_{seen}中的正確標籤集組合是在訓練集中有出現的，S_{unseen}則否。

3.4.4 切除研究

在這個章節中，我們將證明本章提出的方法確實是有用的，我們將在MS-COCO資料集上，比較Resnet-101模型有/無本章提出的訓練方法的微/宏F1分數，表3.5是切除研究的結果。

doi:10.6342/NTU201901512

	正確答案	VGG-16	VGG-16 + WGAN-gp	Resnet-101	Resnet-101 + WGAN-gp
$S_{test-train}$	6544	216	962	695	2365

表 3.4: 模型在MS-COCO測試集上，產生沒有在MS-COCO訓練集見過的標籤集組合的種類數($S_{test-train}$)，其中正確答案欄代表測試集的正確標籤集有多少種沒有出現在訓練集過。

列(a)和列(b)是呈現了Resnet-101基準模型和使用所有本章所提出的方法進行訓練Resnet-101而得到的模型的結果，和表3.1的數字相同。列(c),(d),(e)中的模型架構皆和列(a),(b)的模型架構相同，但是我們移除了一些提出的訓練方法。在列(c)和(d)中，我們不使用負採樣，也就是說，我們移除了式3.4中的$\mathcal{L}_D$第三項，並將第二項的權重從$\frac{1}{2}$改為1。在列(d)中，我們將條件式鑑別器改成使用非條件式的，因此，此鑑別器唯一的輸入便只有標籤集，沒有使用這兩種方法，皆會使傷害到F1分數。在列(e)的模型中，我們直接將分類器的輸出標籤分佈$\tilde{y}$輸入給鑑別器，沒有用岡氏軟性最大化技法，然而，因為這個輸入是連續的，生成器必須要使輸出分佈更銳利(sharp)，才能騙過鑑別器，因此降低了模型的表現。

方法	微F1分數	宏F1分數
(a): Resnet-101	62.8	68.1
(b): Resnet-101 + WGAN-gp	**64.0**	**68.3**
(c): (b) 不使用負採樣	62.2	67.5
(d): (b) 鑑別器非條件式	62.5	67.6
(e): (b) 直接將輸出分佈輸入給鑑別器(不做採樣)	62.3	67.1

表 3.5: 在MS-COCO資料集上，Resnet-101模型有/無本章提出的訓練方法的微/宏F1分數。

doi:10.6342/NTU201901512

3.4.5 模型輸出範例

表3.5是一些Resnet-101在MS-COCO資料集多標籤分類的結果。

doi:10.6342/NTU201901512


	(A)	(B)
正確標籤集:	人、球、 棒球棍、棒球手套	酒杯、杯子、叉子、 刀子、披薩、餐桌
Resnet-101	人、棒球棍	叉子、刀子、披薩、餐桌
Resnet-101 + WGAN-gp	人、球、 棒球棍、棒球手套	酒杯、杯子、叉子、 刀子、披薩、餐桌


	(C)	(D)
正確標籤集:	椅子、沙發、床、書	人、筆電
Resnet-101	沙發、電視、書	人、筆電
Resnet-101 + WGAN-gp	椅子、沙發、電視、 筆電、書	筆電、滑鼠、 鍵盤

圖 3.5: 一些Resnet-101在MS-COCO資料集多標籤分類的結果。若使用WGAN-gp進行訓練，分類器可以將較小的物件預測得較準確，例如在範例(A)中，Resnet-101 + WGAN-gp模型基於人和棒球棍，而正確的預測出棒球手套和球，然而在範例(D)中，它錯誤的將和筆電相關的鍵盤和滑鼠納入了預測結果。

doi:10.6342/NTU201901512

3.5 本章總結

在本章中，我們介紹了基於增進版霍氏生成網路的架構以及訓練模型的方法，並在實驗中證明了模擬標籤間的關聯性不只能使多標籤分類器進步，使用所提出的方法訓練的模型，廣泛化能力也較強。並在切除實驗中，證明了提出的方法皆是有效的，最後展現了一些不同模型預測的例子。

doi:10.6342/NTU201901512

第 四 章　最佳補全蒸餾法應用於多標籤分類

4.1 簡介

上一章介紹了如何使用生成對抗網路幫助多標籤分類器，而這種多標籤分類器是使用對數機率回歸訓練的。此章中，會將重點放在基於遞迴式類神經網路的多標籤分類器。因為遞迴式類神經網路解碼器會依序做標籤的預測，所以在訓練此模型時，必須要事先將標籤集做排序，作為解碼器的目標序列供模型學習。然而，强加人為定義的標籤順序是不自然的，我們發現使用此方法訓練的解碼器容易過度貼合(overfit)到標籤組合(label combination)，使模型在測試階段，很難產生出沒有在訓練階段看過的標籤組合。因此，本論文提出了一個不需要人為定義的標籤順序，便可訓練遞迴式類神經網路解碼器的新架構，並在三個多標籤文件分類的資料集上，皆勝過了具挑戰性的基準模型。我們也發現使用此方法訓練的模型較能產生沒有見過的標籤組合。

4.1.1 研究動機

近年來的研究已顯示多標籤分類可以被轉變成一個順序預測的問題，並使用遞迴式類神經網路解碼器依序預測每個標籤 [2,6,31]。然而，這一種基於遞迴式類神經網路解碼器的多標籤分類模型，存在許多問題。

第一個問題是，因為這些模型是使用最大似然估計在正確標籤序列上訓練的，需要人為定義好的標籤順序，以將標籤集轉變成標籤序列。而在前人的研究中 [2,6]，已顯示如何作標籤排序對模型的表現有相當大的影響，雖然他們討論了許多標籤排序的方法，並建議將標籤依照出現頻率由最多次到最少次進行排序，

doi:10.6342/NTU201901512

然而，加入序列訊息到標籤集中是不自然的，這可能會導致有些標籤的推論關係並不正確，更進一步的，我們發現這類型的模型容易過度貼合至標籤序列，而缺乏廣泛化能力。

第二個問題是使用最大似然估計訓練時，模型永遠是基於正確的前綴標籤序列做接下來的預測，然而，在測試階段時，模型可能預測錯誤的前綴標籤，若模型繼續根據錯誤的結果做預測，這個結果可能會更加糟糕，而導致曝光偏差(章節2.2.3)這個問題。

在本章中，我們提出了一個基於遞迴式類神經網路解碼器的多標籤分類器的新訓練架構，此架構不需要仰賴人為事先定好的標籤順序。此方法是啓發自最佳補全蒸餾法。在我們提出的訓練方法中，此遞迴式類神經網路解碼器的輸入是從上一個時間點的標籤分佈取樣而得的標籤，因此，模型在訓練階段可能會遇到不一樣的標籤順序和錯誤的前綴標籤，因此可以減輕曝光偏差的問題。另外，我們又提出了一個輔助的二元關聯解碼器，在多目標訓練的架構下，和遞迴式類神經網路解碼器共同訓練。此輔助用的解碼器能夠使模型學得更好，並且在測試階段時，我們提出了兩種方法結合兩個解碼器的預測標籤機率，使測試結果更好。

模型將會在章節4.2介紹，而模型的訓練方式和測試方式會分別在章節4.3和章節4.4介紹，最後的實驗部分則在章節4.5中介紹，最後則會在章節4.6與前一章提出的方法比較。

4.2 模型簡介

在此章節中，會以多標籤文件分類為例，介紹模型的架構。模型概觀如圖4.1，分成三個部分，分別是編碼器$\mathcal{E}$、遞迴式類神經網路解碼器$\mathcal{D}_{rnn}$和二元關聯解碼器$\mathcal{D}_{br}$，會在接下來的章節分別介紹。

doi:10.6342/NTU201901512

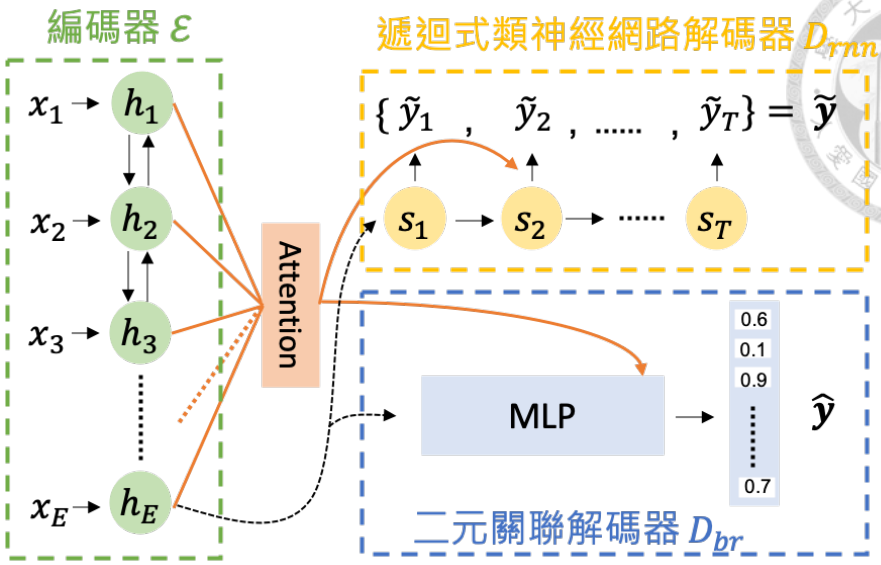


圖 4.1: 模型概觀

4.2.1 編碼器$\mathcal{E}$架構

我們使用了雙向長短期記憶單元(Bidirectional LSTM, BiLSTM)作為編碼器$\mathcal{E}$，此編碼器會以前向和後向讀過m個詞的詞序列$\mathbf{x} = x_1, x_2, ..., x_m$，並計算出每個詞的隱藏層數值$h_1^e, h_2^e, ...h_m^e$。

$$h_1^e, h_2^e, ...h_m^e = \text{BiLSTM}(x_1, x_2, ..., x_m) \tag{4.1}$$

4.2.2 遞迴式類神經網路解碼器$\mathcal{D}_{rnn}$架構

這個遞迴式類神經網路解碼器會依序預測標籤，此解碼器會基於先前預測的標籤來做之後標籤的預測，並從此中學習到標籤關聯性。在實作上，我們使用了具有專注機制的單向的長短期記憶單元(章節2.2.3)，其首個隱藏層會被初始化成編碼器的最後一個隱藏層$h_0^d = h_m^e$，爾後我們會依照下式計算時間t的隱藏層和上下文向量:

doi:10.6342/NTU201901512

$$e_{tj} = v_a^T \tanh(W_a h_{t-1}^d + U_a h_j^e) \tag{4.2}$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{m} \exp(e_{tk})} \tag{4.3}$$

$$c_t = \sum_{j=1}^{m} \alpha_{tj} h_j^e \tag{4.4}$$

$$h_t^d = \text{LSTM}(h_{t-1}^d, [e(\tilde{y}_{t-1}); c_{t-1}]) \tag{4.5}$$

其中，$[e(\tilde{y}_{t-1}); c_{t-1}]$表示將前一時間點$t-1$的標籤$\tilde{y}_{t-1}$的嵌入向量$e(\tilde{y}_{t-1})$和上下文向量$c_{t-1}$做串接，$W_a$和$U_a$則是參數的矩陣。接下來，用來預測標籤$\tilde{y}_t$的條件機率$p_{rnn}(\tilde{y}_t|\tilde{\mathbf{y}}_{<t}, \mathbf{x})$的算法如下，

$$o_t = W_o f(W_d h_t^d + V_d c_t), \tag{4.6}$$

$$p_{rnn}(\tilde{y}_t|\tilde{\mathbf{y}}_{<t}, \mathbf{x}) = \text{softmax}(o_t) \tag{4.7}$$

$$\tilde{y}_t \sim \text{softmax}(o_t + M_t), \tag{4.8}$$

其中，W_o, W_d和V_d皆是參數矩陣，f是一個非線性的激活函數。而在取樣時，我們會用一個遮罩(mask)$M_t \in \mathcal{R}^L$來避免遞迴式類神經網路解碼器產生重複的標籤，其中，L是指標籤的種類，遮罩的定義如下:

$$(M_t)_j = \begin{cases} -\infty & \text{如果第}j\text{個標籤已經在時間}t\text{之前被預測過} \\ 0 & \text{第}j\text{個標籤沒有被預測過} \end{cases} \tag{4.9}$$

4.2.3 二元關聯解碼器$\mathcal{D}_{br}$架構

二元關聯解碼器在此是一個輔助的解碼器，在多目標訓練的框架內，能夠使編碼器能夠學得更好。另一個優點是在測試階段時，我們可以結合兩解碼器的預測結果作為最終預測，再使模型的預測結果更好。

doi:10.6342/NTU201901512

在實作上，我們將編碼器最後一個隱藏層h_m^e輸入至一個深層類神經網路，這個網路的最後一層是由L個S型激活函數組成的，可以獨立的一次性預測每一個標籤存在的機率。由於輸入文字序列可能會十分長，反向傳播的梯度可能會消失，因此我們在此解碼器也加入了專注機制，並加至此深層類神經網路的最後一層，詳細的做法如下，我們將式4.2的h_{t-1}^d換成MLP(h_m^e):

$$e_{tj} = v_a^T \tanh(W_{a'} MLP(h_m^e) + U_{a'} h_j^e) \tag{4.10}$$

其中$W_{a'}, U_{a'}$是參數矩陣，接下來再照式4.3和式4.4算出上下文向量c_t^{br}。最後，此解碼器輸出機率如下:

$$p_{br}(\tilde{y}|\mathbf{x}) = \text{sigmoid}(W_{br}[\text{MLP}(h_m^e); c_t^{br}]), \tag{4.11}$$

其中，W_{br}是參數矩陣。

4.3 訓練方式

在這個章節中，我們會分別介紹如何訓練遞迴式類神經網路解碼器(章節4.3.1)、二元關聯解碼器(章節4.3.2)和多目標訓練的目標函數(章節4.3.3)。

4.3.1 訓練遞迴式類神經網路解碼器

利用强化學習訓練遞迴式類神經網路解碼器

為了解決曝光偏差的問題，我們並不使用正確標籤序列來做訓練，而是以强化學習的角度來看待多標籤分類的問題。在此，這個模型的動作a_t代表在時間t輸出的標籤，狀態s_t代表了時間t之前預測的標籤序列$\tilde{\mathbf{y}}_{<t}$，此模型的策略$\pi(s)$是指在狀

doi:10.6342/NTU201901512

態s下，動作a的機率分佈。當模型輸出結束符號⟨eos⟩(end-of-sentence,eos)時，解碼過程結束，模型會得到一個獎勵R。

在我們提出的方法中，獎勵的定義如式4.12。

$$R(\mathbf{y}^*, \tilde{\mathbf{y}}) = -|\{\mathbf{y}^*\} \setminus \{\tilde{\mathbf{y}}\}| - |\{\tilde{\mathbf{y}}\} \setminus \{\mathbf{y}^*\}|, \tag{4.12}$$

其中，$\mathbf{y}^*$和$\tilde{\mathbf{y}}$分別代表正確標籤集和遞迴式類神經網路解碼器輸出的標籤集，$B \setminus A$是指集合A在集合B中的差集。此獎勵函數的兩項分別代表正確標籤集中沒有被預測到的標籤數目和模型預測的標籤集中錯誤標籤的數目。

最佳補全蒸餾法(Optimal Completion Distillation, OCD)

然而，在現階段普遍使用的强化學習演算法中，例如Q學習和策略梯度，無法很有效的從正確的標籤序列學到所需的資訊，只能根據解碼過程結束後得到的獎勵R來做學習。這裡我們使用了最佳補全蒸餾法，以下會逐步介紹。我們先介紹最佳Q值(optimal Q-value)，它衡量了每個時間點t的每一個動作a_t。

最佳Q值$Q^*(s, a)$代表了模型在狀態s執行動作a_t後，在未來能得到的最大獎勵是多少，因此，對於前綴序列$\tilde{\mathbf{y}}_{<t}$和在時間點t輸出的標籤a的最佳Q值是由接上最佳後綴序列$\mathbf{y}_{opt}$所得到的，寫成數學式如下:

$$Q^*(\tilde{\mathbf{y}}_{<t}, a) = \max_{\mathbf{y}_{opt} \in \mathcal{Y}} R(\mathbf{y}^*, [\tilde{\mathbf{y}}_{<t}, a, \mathbf{y}_{opt}]). \tag{4.13}$$

給定模型在狀態s所有可能的動作的Q值，我們定義的最佳策略如式4.14，將每個可能動作的Q值套用溫度為τ的軟性最大化(softmax)。

$$\pi^*(a|\tilde{\mathbf{y}}_{<t}) = \frac{\exp(Q^*(\tilde{\mathbf{y}}_{<t}, a)/\tau)}{\sum_{a'} \exp(Q^*(\tilde{\mathbf{y}}_{<t}, a')/\tau)}, \tag{4.14}$$

doi:10.6342/NTU201901512

$\tau \geq 0$是一個代表溫度的參數，如果τ趨近於0，則最佳策略只會在有最大的最佳Q值的動作上有機率。

給定一個訓練的配對$(\mathbf{x}, \mathbf{y}^*)$，我們先由模型獨立取樣出一個完整的序列$\tilde{\mathbf{y}} \sim p_{rnn}(\cdot|\mathbf{x}), \Gamma\text{ff}$ *i.i.d.*，然後模型便會向最佳策略蒸餾(distill)知識，方法是最小化每個時間點的最佳策略和模型輸出分佈的克雷散度，數學式如下:

$$\mathcal{L}_{OCD} = \mathbb{E}_{(\mathbf{x},\mathbf{y}^*)\sim data}\mathbb{E}_{\tilde{y}\sim p_{rnn}(\cdot|\mathbf{x})}[\sum_{t=1}^{|\mathbf{y}^*|} \mathrm{KL}(\pi^*(\cdot|\tilde{\mathbf{y}}_{<t})||p_{rnn,t}(\cdot|\tilde{\mathbf{y}}_{<t}, \mathbf{x}))] \tag{4.15}$$

最佳補全蒸餾法在每個時間點t，皆鼓勵模型輸出所有可以使獎勵最大化的標籤，然而最大似然估計是從一個給定的標籤序列學習所得，因此在每個時間點只會有一個標籤。因此，最佳補全蒸餾法的目標是所有還未預測過的正確標籤，並會給每個未預測過的正確標籤相同的機率，然後在模型將所有正確標籤的輸出完後，此目標函數便會鼓勵模型輸出⟨eos⟩。表格4.1和圖4.2是一個使用最佳補全蒸餾法的範例。

因為最佳補全蒸餾法的目標標籤只和模型之前輸出的標籤有關，因此，我們並不需要人為事先定義的標籤順序去訓練遞迴式類神經網路解碼器，而另一個好處是，我們使用從模型取樣出的標籤作為遞迴式類神經網路下一個時間點的輸入，這樣做可以減少曝光偏差的問題。值得一提的是，我們若使用有別於式4.12的獎勵函數，算出來也會是相同的最佳補全蒸餾法目標標籤，因為對於所有正確標籤，最佳Q值皆會是相同的。

4.3.2 訓練二元關聯解碼器

給定向量形式的正確標籤集$\mathbf{y}^*_{vec} = [y^*_{vec,1}, y^*_{vec,2}\cdots, y^*_{vec,L}]^T \in \{0,1\}^L$，我們使用二元

doi:10.6342/NTU201901512

時間t	最佳補全蒸餾法的目標標籤	標籤的最佳Q值	最佳策略π^* ($\tau \to 0$)	預測標籤$\tilde{y}_t$
0	A, B, D	$[0, 0, -1, 0, -3]$	$[\frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3}, 0]$	B
1	A, D	$[0, -1, -1, 0, -2]$	$[\frac{1}{2}, 0, 0, \frac{1}{2}, 0]$	C
2	A, D	$[-1, -2, -2, -1, -3]$	$[\frac{1}{2}, 0, 0, \frac{1}{2}, 0]$	A
3	D	$[-2, -2, -2, -1, -2]$	$[0, 0, 0, 1, 0]$	D
4	⟨eos⟩	$[-2, -2, -2, -2, -1]$	$[0, 0, 0, 0, 1]$	⟨eos⟩

表 4.1: 一個由最佳補全蒸餾法的範例，和圖4.2的範例相同。在此範例中，總共有4種標籤A,B,C,D和⟨eos⟩，而此物件的正確標籤有A,B,D三種。標籤的最佳Q值和最佳策略的向量中的每個值分別代表標籤A,B,C,D和⟨eos⟩的最佳Q值和策略的機率。我們在此將軟性最大化的溫度τ設為一個接近0的數值，因此最佳策略只會在有最大的最佳Q值的動作上有機率。例如，在時間點$t = 1$時，有兩個最佳補全蒸餾法的目標標籤，分別是A和D，能最大化最佳Q值(0)，然後我們從模型的輸出分佈取樣而得標籤C，作為$t = 2$時的輸入，因為C不在正確標籤集中，而使模型能拿到最大的獎勵值變為-1。

的交叉熵來訓練二元關聯解碼器:

$$\mathcal{L}_{logistic} = \mathbb{E}_{(\mathbf{x}, \mathbf{y}^*_{vec}) \sim data}[\sum_{i=1}^{L} y^*_{vec,i} log\, \hat{y}_i (1 - y^*_{vec,i}) log(1 - \hat{y}_i)], \tag{4.16}$$

其中，$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2 ..., \hat{y}_L]^T$是一個長度為$L$的向量，代表每個標籤的預測機率。

4.3.3 多目標訓練

多目標訓練的目標函數如下:

$$\mathcal{L}_{MTL} = \mathcal{L}_{OCD} + \lambda \mathcal{L}_{logistic}, \tag{4.17}$$

doi:10.6342/NTU201901512

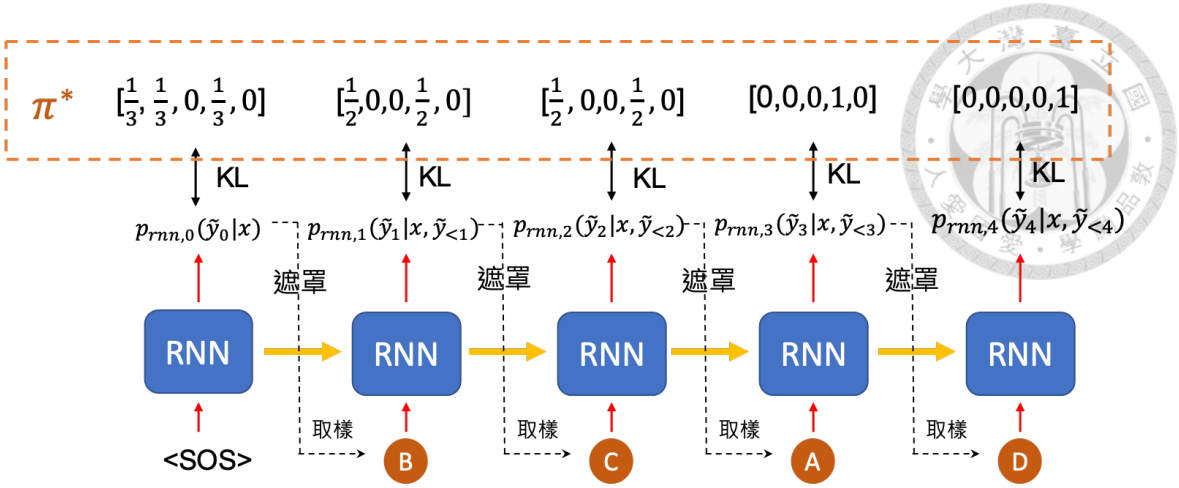


圖 4.2: 最佳補全蒸餾法的訓練過程示意圖，和表4.1的範例相同。在進行取樣時，輸出機率會先經過一個遮罩，防止模型輸出重複的標籤。而模型會向最佳策略學習，學習的方法便是最小化兩機率分佈的克雷散度。

其中，λ是兩損失函數間的權重。

doi:10.6342/NTU201901512

4.4 測試方式

在這個章節中，我們會在章節4.4.1分別介紹兩解碼器的測試方式，並在章節4.4.2介紹如何結合兩解碼器的預測結果，作為最終的預測結果$\mathcal{H} = \{l_1, l_2, ..., l_T\}$，其包含了$T-1$個標籤和⟨eos⟩，代表遞迴式類神經網路解碼器的解碼過程結束。

4.4.1 基本的測試方式

對於二元關聯解碼器，最後一層S函數的輸出是每一個標籤的機率，因此，理論上最好的門檻值(threshold)是0.5，若此機率大於0.5，便將此標籤納入預測結果，和最大化式4.18的預測標籤集$\mathcal{H}$是一樣的，式4.18是所有選中的標籤機率和沒選到的標籤的機率的乘積。

$$P_{br}(\mathcal{H}) = \prod_{l \in \mathcal{H}} p_{br}(y_l = 1|\mathbf{x}) \times \prod_{l \notin \mathcal{H}} p_{br}(y_l = 0|\mathbf{x}) \tag{4.18}$$

對於遞迴式類神經網路解碼器，一個典型的測試方法是使用集束搜尋(beam search)去解式4.19 [28,31]。給定模型的輸入x，預測結果$\mathcal{H}$的機率為:

$$P_{path}(\mathcal{H}) = \prod_{i=1}^{i=T} p_{rnn}(l_i|\mathbf{x}, l_1, ...l_{i-1}). \tag{4.19}$$

4.4.2 結合兩解碼器的測試方式

要結合兩解碼器，我們將式4.18和式4.19的乘積作為我們最終的目標函數:

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \{P_{path}(\mathcal{H}) \times P_{br}(\mathcal{H})\}. \tag{4.20}$$

doi:10.6342/NTU201901512

然而，式4.20並不好解，因為遞迴式類神經網路解碼器的輸出是一個序列的標籤機率，而二元關聯解碼器是一次性的預測所有的標籤的獨立機率。在本論文中，我們提供了兩種結合兩解碼器的測試方式，來解式4.20。

對數機率重新計分(logistic rescoring)

在這個方法中，我們先使用集束搜尋在遞迴式類神經網路解碼器上得到一些預測結果，再使用二元關聯解碼器(式4.18)重新計分，最後，再將機率$P_{br}(\mathcal{H}_{best})$最高的預測結果作為最終的預測結果。

對數機率共同解碼(logistic joint decoding)

這個方法是一次完成(one-pass)的解碼方法，我們先將式4.18改寫:

$$P_{br}(\mathcal{H}) = \prod_{l \in \mathcal{H}} \frac{p_{br}(y_l = 1|\mathbf{x})}{p_{br}(y_l = 0|\mathbf{x})} \times \prod_{l} p_{br}(y_l = 0|\mathbf{x}) \tag{4.21}$$

因為式中的第二項和預測結果$\mathcal{H}$無關，因此，我們將此式代入式4.20:

$$\begin{aligned} \hat{H} &= \arg\max_{\mathcal{H}} \{P_{path}(\mathcal{H}) \times P_{br}(\mathcal{H})\} \\ &= \arg\max_{\mathcal{H}} \{\prod_{i=1}^{i=T} p_{rnn}(l_i|\mathbf{x}, l_1, ...l_{i-1}) \frac{p_{br}(y_{l_i} = 1|\mathbf{x})}{p_{br}(y_{l_i} = 0|\mathbf{x})}\} \end{aligned} \tag{4.22}$$

便可以使用此式做集束搜尋。注意到我們設定⟨eos⟩ (l_T)的機率$p_{br}(y_{l_T} = 0|\mathbf{x}) = p_{br}(y_{l_T} = 1|\mathbf{x}) = 0.5$ ，因為它不會出現在二元關聯解碼器中。

4.5 模型表現評估

為了檢驗我們提出的方法，我們在實驗中使用了三個多標籤文件分類的資料集，三者在標籤數量、文章長度和資料集大小皆有很大的差異。實驗資料集的介紹在

doi:10.6342/NTU201901512

章節4.5.1中，實驗比較的基準模型和評估指標分別在章節4.5.2和章節4.5.3介紹，超參數的設定和模型訓練的細節在章節4.5.4，最後的實驗結果則在章節4.5.5，並在章節4.6與前一章的方法做比較。

4.5.1 實驗資料集

我們使用了三個多標籤文件分類的資料集，資料集的詳細數據在表4.2一些資料集的例子在表4.3，介紹如下:

- **路透社-21758(Reuters-21758)**:路透社-21758是路透社(Reuters)1987年在路透社新聞上搜集下來而成的，裡面有大約10000篇文章共有90種標籤。
- **Arxiv學術論文資料集(Arxiv Academic Paper Dataset, AAPD)** [31]:其包含了55,840篇在arxiv上關於電腦科學的學術論文的論文摘要和所屬類別，一篇論文平均有2.41個標籤。
- **路透社資料集卷一(Reuters Corpus Volume I, RCV1-V2)** [32]:這個資料集含有大量人工標注的新聞文章，共有804,414篇文章和103個標籤種類。

我們考慮了南氏等人(Nam et al.) [2]在路透社-21758的預處理方法，隨機取了10%的訓練集資料作為驗證集。對於Arxiv學術論文資料集和路透社資料集卷一，我們參考了楊氏等人(Yang et al.) [31]訓練集/驗證集/測試集的切法。對於這三個資料集，我們將多於500詞的文章刪除，在每一個資料集中大約移除了0.5%的文章。

4.5.2 基準模型介紹

為了證明我們提出的模型是比較好的，我們比較了幾個具競爭力的基準模型，其

doi:10.6342/NTU201901512

資料集	$N_{training}$	N_{val}	N_{test}	標籤種類	每篇文章平均的字數	每篇文章平均的標籤數量
路透社-21758	6993	776	3019	90	53.94	1.24
Arxiv學術論文資料集	53840	1000	1000	54	163.42	2.41
路透社資料集卷一	802414	1000	1000	103	123.94	3.24

表 4.2: 資料集的統計資料。$N_{training}$、N_{val}、N_{test}分別是指訓練集、驗證集、測試集的資料筆數。

中，除了二元關聯模型++，編碼器架構都是相同的。序列到序列模型、無序遞迴式類神經網路模型和我們提出的模型使用的遞迴式類神經網路解碼器皆是相同的，以下是比較的基準模型的介紹:

- **二元關聯**:是使用對數機率回歸的減損函數(式4.16)訓練的模型，是由編碼器和二元關聯解碼器所組成。
- **二元關聯++**:此模型是加大版的二元關聯模型，訓練方法也和它相同。因為多目標學習的模型使用的參數量較多，這個模型增加了編碼器的參數量，使其和多目標學習的模型有差不多的參數量。
- **序列到序列**:這個模型是由基於遞迴式類神經網路的編碼器和遞迴式類神經網路解碼器所組成。這個模型是用最大似然估計訓練的，目標標籤序列是由出現次數最多次到最少次所排序 [2,31]。
- **無序遞迴式類神經網路(Order Free RNN, OfRNN)** [28]:這個做法原先是在多標籤影像分類上，可以無序訓練遞迴式類神經網路解碼器的方法。我們重新實作在多標籤文件分類上。它的作法如下:

doi:10.6342/NTU201901512

文章	標籤集
路透社-21758	
UK money market offered early assistance. The bank of england said it had invited an early round of bill offers from the discount houses after forecasting a shortage of around 950 mln stg in the money market today. Among the main factors affecting liquidity, bills maturing in official hands and the take-up of treasury bills will drain around 572 mln stg while a rise in note circulation wil take out some 280 mlns stg. In addition, exchequer transactions and bankers' balances below target will remove some 85 mln stg and 15 mln stg for the system respectively.	**money-fx**, **interest**
Arxiv學術論文資料集	
Our paper explores contribution patterns of creativity and collaboration of wikipedia editors as manifestations of social dynamics between the editors. We find support for existence of four socially constructed personas among the editors and difference in distribution of personas in articles of different qualities.	**cs.SI** (Social and Information Networks), **physics.soc-ph** (Physics and Society)
路透社資料集卷一	
The stock of Tylan General Inc. jumped Tuesday after the maker of process-management equipment said it is exploring the sale of the company and added that it has already received some inquiries from potential buyers. Tylan was up 2.50 to 12.75 in early trading on the Nasdaq market. The company said it has set up a committee of directors to oversee the sale and that Goldman, Sachs amp; Co. has been retained as its financial adviser.	**C15** (performance), **C152** (comment/forecasts), **C18** (ownership changes), **C182** (asset transfer), **CCAT** (corporate/indusrial)

表 4.3: 三個多標籤文件分類的資料集中的例子。

doi:10.6342/NTU201901512

在訓練遞迴式類神經網路解碼器的每一個時間點t，此時間點的目標標籤是在正確標籤集中，模型輸出機率最高且模型尚未預測過的標籤，因此此訓練方法會動態的決定模型的目標標籤序列，而標籤的順序會依照模型的輸出機率來自動決定，不需要仰賴人為定義的標籤順序。然而，此方法的遞迴式類神經網路解碼器在訓練時，都是基於正確的標籤做預測，因此也會有曝光偏差的問題。

為了討論曝光偏差的問題，我們也比較了序列到序列模型和無序遞迴式類神經網路模型，使用時序採樣訓練的模型，分別是**序列到序列+時序採樣**和**無序遞迴式類神經網路+時序採樣**。

4.5.3 評估指標介紹

我們在此實驗中，使用了五種評估指標，這些指標可以分成兩類，介紹如下:

- **基於實例的評分標準(example-based measures)**:這些標準是可以每筆資料分開計算的，計算的方法是比較每一筆資料中，模型的輸出和正確標籤集，最後的結果便是每一筆資料所得評分標準的平均分數，共有三種，介紹分別如下:
  - 子集正確率*(subset accuracy, ACC)*:這個是最嚴格的評分標準，模型預測的標籤集需和正確標籤集完全相同，才算是正確的，此分數代表有多少比例的資料模型預測完全正確。

$$\mathrm{ACC}(\mathbf{y}^*, \hat{\mathbf{y}}) = \mathbb{I}[\mathbf{y}^* = \hat{\mathbf{y}}] \tag{4.23}$$

  - 漢明正確率*(hamming accuracy, HA)*:這個分數是指模型答對的標籤數和

doi:10.6342/NTU201901512

所有的標籤數(資料筆數乘標籤種類數)的比例。

$$\mathrm{HA}(\mathbf{y}^*, \hat{\mathbf{y}}) = \frac{1}{L}\sum_{i=1}^{i=L} \mathbb{I}[\mathbf{y}_i^* = \hat{\mathbf{y}}_i] \tag{4.24}$$

其中，L是指標籤的數目。

– 基於實例的*F1*分數*(example-based F1, ebF1)*:這個F1分數是藉由比較每一筆資料的模型預測標籤集和正確標籤集，然後將所有資料的F1分數作平均而得到的，計算方法是正確標籤的數目和預測標籤數加上正確標籤數的比例。

$$\mathrm{ebF1}(\mathbf{y}^*, \hat{\mathbf{y}}) = \frac{2\sum_{i=1}^{L} \mathbf{y}_i^* \hat{\mathbf{y}}_i}{\sum_{i=1}^{L} \mathbf{y}_i^* + \sum_{i=1}^{L} \hat{\mathbf{y}}_i} \tag{4.25}$$

其中，L是指標籤的數目。

- **基於標籤的評分標準**:這些評分標準是將每個標籤分別當作二元分類問題，然後再將所有標籤的分數做某種平均得到的。我們使用了微*F1*分數和宏*F1*分數兩種，已經在章節3.4.1介紹過。通常較高的微F1分數代表模型在出現次數較多的標籤上表現較好，而較高的宏F1分數代表模型在出現次數較少的標籤上表現較好。

4.5.4 實驗設定

我們使用了Pytorch來實作我們的實驗。有一些因資料集而異的超參數在表4.4。二元關聯解碼器是一個3層的深層類神經網路且每層有512個溢出線性整流單元。實驗中使用的詞嵌入大小是512，且我們隨機初始化詞嵌入。我們使用學習率是0.0005的亞當優化器。為了避免過度貼合的問題，我們不但使用了丟棄法，也限制了梯度的大小在$[-10, 10]$之間。在使用最佳補全蒸餾法訓練的模型中，我

doi:10.6342/NTU201901512

資料集	單字量	長短期記憶單元的層數	批次大小	丟棄法的機率大小
Reuters	22747	(2,2)	96	0.5
AAPD	30000	(2,2)	128	0.5
RCV1-V2	50000	(2,3)	96	0.3

表 4.4: 一些在不同資料集使用的超參數，其中，長短期記憶單元的層數(2,3)是指在編碼器長短期記憶單元的層數為2，而在解碼器為3。

們將溫度τ設為10^{-8}。對於使用時序採樣的模型，我們初始的教師强迫的機率設為1，並會在訓練過程中線性降低，直到訓練結束時為0.7。在多目標學習的模型中，兩個損失函數間的加權值λ為1。

在訓練模型時，我們使用一個固定的期數(epoch)訓練模型，並在每1000次更新後，測試模型在驗證集上的表現，並將模型存下來。當訓練結束後，我們會從中挑選在驗證集上微F1分數最高的模型為最終模型，並在測試集上測試。

在測試模型時，對於所有的遞迴式類神經網路解碼器，我們在集束搜尋時使用的集束(beam)數目為6。對於所有的二元關聯解碼器，我們在驗證集上找到一個最佳的閾值使微F1分數最高，然後在測試時，挑選所有輸出機率大於此閾值的標籤 [33,34]。

4.5.5 實驗結果

在這個章節中，我們會列出在三個資料集上，基準模型與我們提出的模型的結果。對於多目標學習的模型，我們使用了4種不同的解碼方式(章節4.4)，分別是只使用遞迴式類神經網路解碼器解碼、只使用二元關聯解碼器解碼、對數機率重新計分和對數機率共同解碼，其中，後兩者是結合兩者解碼器預測結果的方法(章

doi:10.6342/NTU201901512

節4.4.2)。另外，因為我們使用了五個評分標準進行衡量，為了有一個易於比較的標準，我們也計算了五種評分標準的平均值，並列為參考。

Arxiv學術論文資料集(AAPD)

Arxiv學術論文資料集上的實驗結果可以參考表4.5，我們可以觀察到不同的模型擅長於不同的評量標準，例如:序列到序列模型(列(f))和無序遞迴式類神經網路(列(h))在子集正確率上有較好的表現，而二元關聯模型在漢明正確率上表現較好，但面對出現次數較少的標籤時，預測結果卻不太理想(宏F1分數較低)。然而對於最佳補全蒸餾法的模型(列(j))，除了子集正確率之外，其他的評分標準皆勝過基準模型，尤其是在微F1分數(0.707)和基於實例的F1分數(0.737)上，這代表我們提出的訓練遞迴式類神經網路解碼器的方法較好。

對於多目標學習的模型，我們使用了4種解碼方法，前兩個解碼方法(列(k),(l))是指只用其中一個解碼器的預測結果，而後兩個方法(列(m),(n))是指結合兩種解碼器的預測方法。若使用多目標學習，我們發現模型在除了子集正確率之外的評分標準皆變得更好(列(j)對比列(k))。另外，結合兩種解碼器的預測方法得到了最好的結果，也勝過了前人的方法(列(a)(b)(c))。我們還發現二元關聯的方法在多目標學習上有顯著的進步(列(d)對比列(l))，可能是因為此模型的編碼器也藉由遞迴式類神經網路解碼器學習到了標籤關聯性，是原本二元關聯模型忽略的。

圖4.3展示了最佳補全蒸餾法、二元關聯和最佳補全蒸餾法+多目標學習模型(3種解碼方式)，在驗證集上子集正確率和微F1分數的在模型訓練時的變化，其中，二元關聯模型表現得最不好且收斂速度最慢。然而，在多目標學習的幫忙下，二元關聯解碼器收斂的快了許多也較好，並且可以看到多目標學習也幫助了

doi:10.6342/NTU201901512

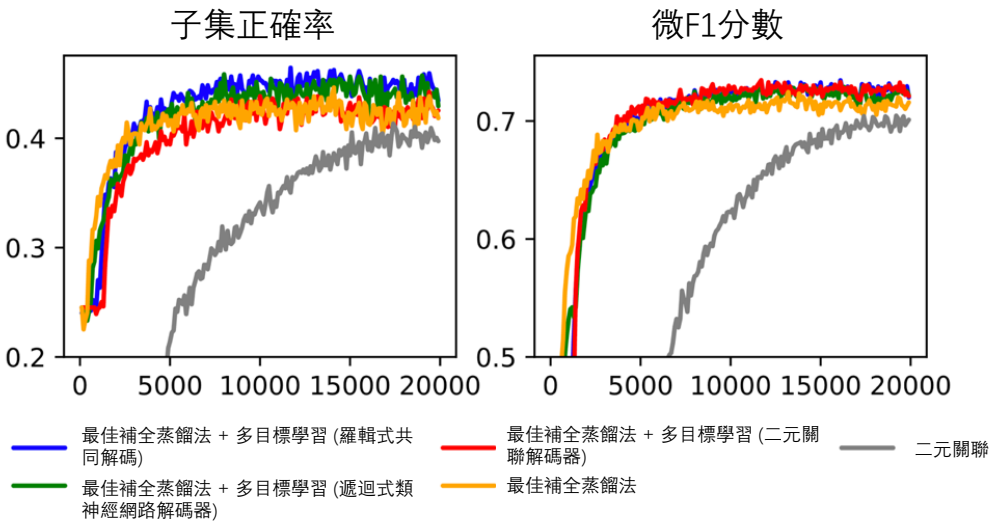


圖 4.3: 最佳補全蒸餾法、二元關聯和最佳補全蒸餾法+多目標學習模型(3種解碼方式)，在Arxiv學術論文資料集的驗證集上，子集正確率和微F1分數的在模型訓練時的變化示意圖。x軸表示模型的更新次數，y軸則是評分標準的變化。

最佳補全蒸餾法的表現。

doi:10.6342/NTU201901512

模型		宏F1分數	微F1分數	基於實例的F1分數	子集正確率	漢明正確率	衡量標準的平均值
(a) 序列到子集簡單版 (Seq2set (simp.)) [35]		-	0.705	-	-	0.9753	-
(b) 序列到子集(Seq2set) [35]		-	0.698	-	-	0.9751	-
(c) SGM+GE [31]		-	0.710	-	-	0.9755	-
基準模型							
(d) 二元關聯		0.523	0.694	0.695	0.368	0.9741	0.651
(e) 二元關聯++		0.521	0.700	0.703	0.390	0.9750	0.658
(f) 序列到序列		0.511	0.695	0.707	**0.421**	0.9743	0.662
(g) 序列到序列+時序採樣		0.541	0.703	0.713	0.406	0.9742	0.667
(h) 無序遞迴式類神經網路		0.539	0.696	0.708	0.413	0.9742	0.666
(i) 無序遞迴式類神經網路+時序採樣		0.548	0.699	0.709	0.416	0.9743	0.669
本論文提出的方法							
(j) 最佳補全蒸餾法		0.541	0.707	0.723	0.403	0.9740	0.670
最佳補全蒸餾法+多目標學習	(k) 遞迴式類神經網路解碼器	0.578	0.711	0.727	0.391	0.9742	0.676
	(l) 二元關聯解碼器	0.562	0.711	0.718	0.382	**0.9760**	0.670
	(m) 對數機率重新計分	**0.585**	**0.720**	**0.736**	0.395	0.9749	**0.682**
	(n) 對數機率共同解碼	0.580	0.719	0.731	0.399	0.9753	0.681

表 4.5: Arxiv學術論文資料集上的實驗結果。

doi:10.6342/NTU201901512

路透社-21758(Reuters-21758)

對比於Arxiv學術論文資料集，路透社-21758是一個較小的資料集，其每筆資料平均的標籤數只有1.24個，且超過80%的資料只有一個標籤。表4.6是路透社-21758上的實驗結果，這些結果再次證明了最佳補全蒸餾法較好，且多目標學習的確能幫助訓練模型。

doi:10.6342/NTU201901512

模型		宏F1分數	微F1分數	基於實例的F1分數	子集正確率	漢明正確率	衡量標準的平均值
支援向量機(SVM) [36]		0.468	0.787	-	-	-	-
編碼器解碼器(EncDec) [2]		0.457	0.855	0.891	0.828	0.996	0.805
基準模型							
二元關聯		0.442	0.861	0.878	0.817	0.9964	0.799
二元關聯++		0.407	0.852	0.861	0.812	0.9962	0.786
序列到序列		0.465	0.862	0.895	0.834	0.9965	0.811
序列到序列+時序採樣		0.464	0.856	0.895	0.834	0.9965	0.809
無序遞迴式類神經網路		0.445	0.862	0.901	0.835	0.9963	0.806
無序遞迴式類神經網路+時序採樣		0.452	0.859	0.896	0.836	0.9962	0.808
本論文提出的方法							
最佳補全蒸餾法		0.458	0.872	0.903	0.839	0.9966	0.814
最佳補全蒸餾法+多目標學習	遞迴式類神經網路解碼器	0.475	0.874	**0.905**	**0.844**	0.9966	0.819
	二元關聯解碼器	0.459	**0.877**	0.898	0.835	0.9966	0.813
	對數機率重新計分	0.477	0.875	0.903	0.842	**0.9967**	0.819
	對數機率共同解碼	**0.490**	0.874	0.904	0.843	**0.9967**	**0.822**

表 4.6: 路透社-21758上的實驗結果。

doi:10.6342/NTU201901512

路透社資料集卷一(RCV1-V2)

相較於Arxiv學術論文資料集和路透社-21758，路透社資料集卷一是由相當多的資料所組成的。特別的是，這些標籤有階層性的特性，也可以說是樹的架構，在上層的標籤是一個大的分類標準，並且下層會有幾個屬於它的子標籤，因此，若有一個在葉節點(leaf node)的標籤屬於此文章，則所有從根節點(root node)到此葉節點的標籤，皆會屬於此文章。若我們將標籤由出現次數從多到少排序，則父結點(parent node)永遠會在其子結點的前方。

在此資料集中，最佳補全蒸餾法模型的進步量較小，這可能是因為人事先定義好的標籤順序存有標籤間的階層順序的資訊，是最佳補全蒸餾法模型忽略的。並且我們觀察到無序遞迴式類神經網路模型會自己學到先預測父結點再預測子結點，而最佳補全蒸餾法模型因為在訓練時傾向於將所有的標籤給予同樣的機率，因此會有各式各樣的標籤順序。

doi:10.6342/NTU201901512

模型		宏F1分數	微F1分數	基於實例的F1分數	子集正確率	漢明正確率	衡量標準的平均值
基準模型							
二元關聯		0.671	0.868	0.881	0.642	0.9919	0.811
二元關聯++		0.650	0.867	0.881	0.646	0.9919	0.807
序列到序列		0.654	0.864	0.881	**0.662**	0.9916	0.811
序列到序列+時序採樣		0.653	0.860	0.878	0.658	0.9914	0.809
無序遞迴式類神經網路		0.660	0.863	0.878	0.650	0.9917	0.809
無序遞迴式類神經網路+時序採樣		0.637	0.862	0.876	**0.662**	0.9917	0.806
本論文提出的方法							
最佳補全蒸餾法		0.668	0.866	0.882	0.654	0.9918	0.812
最佳補全蒸餾法+多目標學習	遞迴式類神經網路解碼器	0.671	0.867	0.882	0.651	0.9918	0.813
	二元關聯解碼器	0.663	0.869	**0.885**	0.637	**0.9920**	0.813
	對數機率重新計分	**0.676**	0.869	0.884	0.653	0.9919	0.815
	對數機率共同解碼	0.674	**0.871**	**0.885**	0.658	**0.9920**	**0.816**

表 4.7: 路透社資料集卷一上的實驗結果。

doi:10.6342/NTU201901512

三個資料集的綜合排名

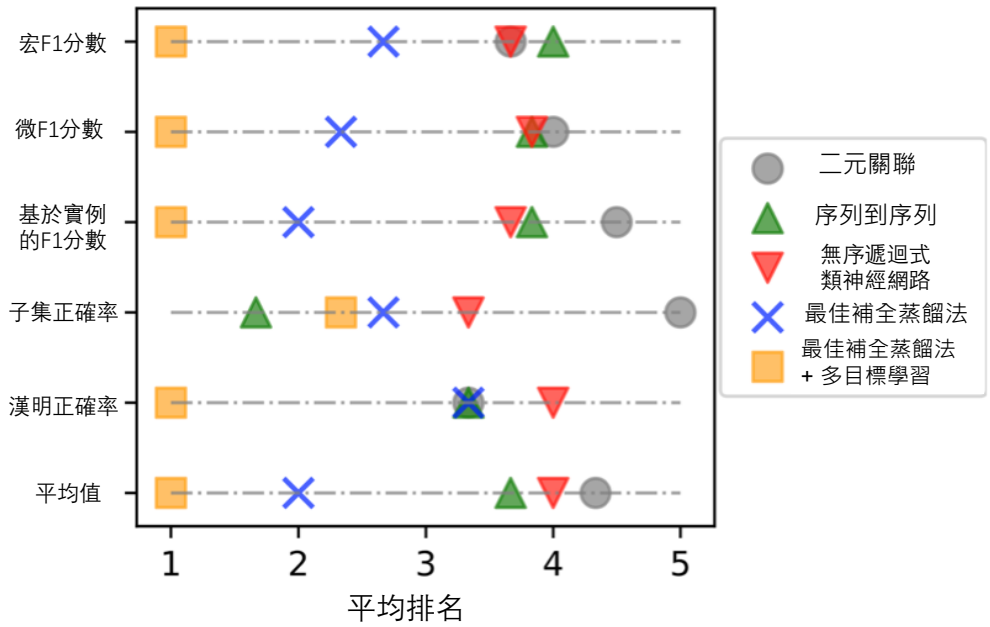


圖 4.4: 每個模型在三個資料集上，在每個評分標準下的平均排名，排名的值越小，代表模型表現越好，其中，最佳補全蒸餾法+多目標學習模型是使用對數機率共同解碼。

圖4.4展現了各個模型在三個資料集的綜合排名，明顯的可以發現，最佳補全蒸餾法+多目標學習模型表現得最好，接下來則是最佳補全蒸餾法。值得注意的是序列到序列模型在子集正確率上有最好的表現，但在其他的評分標準上卻表現較差。

4.5.6 實驗結果討論

關於曝光偏差現象的討論

圖4.5是在Arxiv學術論文資料集上，各個模型對於標籤組合在訓練集出現次數的基於實例的F1分數。顯然地，若標籤組合在訓練集上出現的越多次，模型對於此類

doi:10.6342/NTU201901512

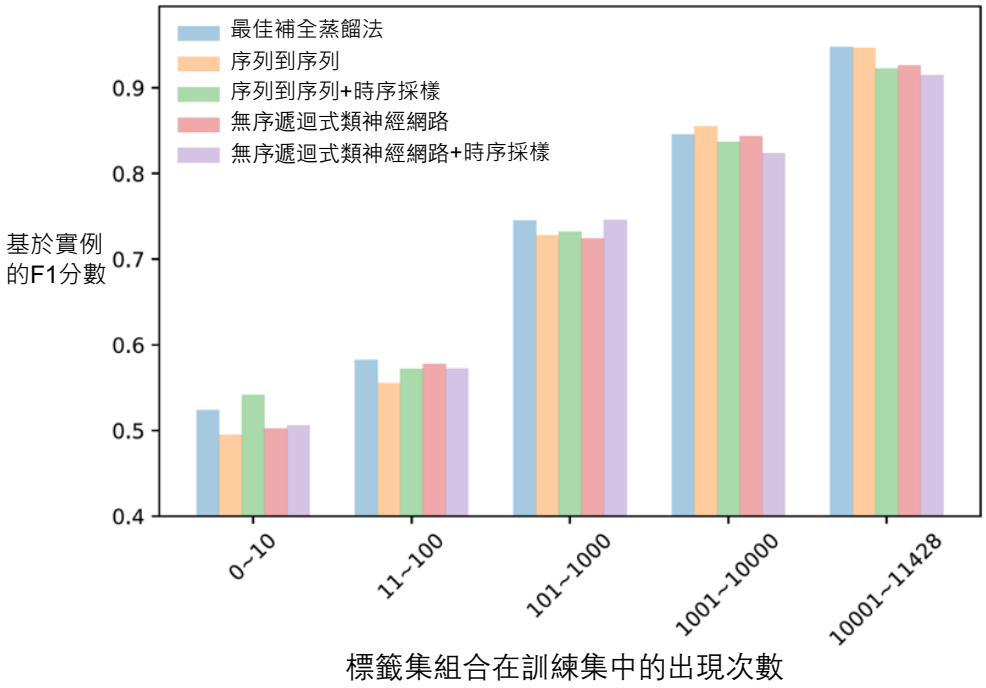


圖 4.5: 在Arxiv學術論文資料集上，各個模型對於標籤組合在訓練集出現次數的基於實例的F1分數。0 ～ 10代表此類的測試資料的標籤組合在訓練集中只出現0 ～ 10次。

的標籤組合就會表現得越好，一個有趣的發現是時序採樣對於序列到序列模型和無序遞迴式類神經網路模型，在較少見的標籤組合上皆有進步，但對於較常見的標籤組合卻是退步。這個也許是因為模型在沒見過的標籤組合表現較差，而曝光偏差的問題會因為模型表現不好而更加嚴重，因此，時序採樣在此情況下帶來較多的進步。

然而，在時序採樣中，目標的標籤序列和沒有使用時序採樣的情況是一樣的，但解碼器的輸入卻會不一樣，因此，在時序採樣的取樣過程中，我們可能會採樣到不符合事先定好的標籤順序的標籤，也有可能取樣到重複的標籤，這會在訓練時誤導模型，這也許是時序採樣在較常見的標籤組合表現較差的原因。

doi:10.6342/NTU201901512

	正確答案	最佳補全蒸餾法	序列到序列	序列到序列+時序採樣	無序遞迴式類神經網路	無序遞迴式類神經網路+時序採樣
Arxiv學術論文資料集						
S_{test}	392	302	214	293	251	259
$S_{test-train}$	43	30	1	3	1	4
路透社-21758						
S_{test}	210	159	135	140	139	144
$S_{test-train}$	94	37	15	16	23	26
路透社資料集卷一						
S_{test}	279	222	233	197	184	168
$S_{test-train}$	5	18	11	0	0	3

表 4.8: 在各個資料集的測試集上，不同模型產生出的標籤組合的種類數(S_{test})，還有產生出在訓練集中沒有出現過的標籤組合的種類數($S_{test-train}$)。

在此圖中，最佳補全蒸餾法模型在各個情況下都有好的表現，因為最佳補全蒸餾法的減損函數會按照模型的輸出而決定，並且因為我們在訓練過程中，沒有使用正確標籤序列來訓練模型，模型在訓練時能夠探索更多的狀態，因此在每個情況下表現能夠更加穩定。

表4.8能夠再次證明上述的想法，序列到序列模型和無序遞迴式類神經網路模型在路透社-21758和Arxiv學術論文資料集上產生了較少的標籤組合，它們可能傾向於"記憶"標籤組合，因此產生出的標籤組合較為相似，表示它的廣泛化能力較差。對於有使用時序採樣進行訓練的模型，他們也產生了較多的標籤組合(除了在路透社資料集卷一上)。然而，最佳補全蒸餾法模型產生了最多的沒有在訓練集出現過的標籤組合，這可能是因為其在訓練時，看過各式各樣的標籤順序。因為序列到序列模型傾向於產生訓練集上見過的標籤組合，因此其在Arxiv學術論文資料集子集正確率較高(表4.5)。

doi:10.6342/NTU201901512

模型	案例一	案例二	案例三
正確答案	**cs.it**, **math.it**, cs.ds	cs.lg, stat.ml, math.st, stat.th	**cs.it**, **math.it**, cs.ds, cs.dc
序列到序列	**cs.it**, **math.it**, cs.ni	**cs.it**, **math.it**, math.st, stat.th	**cs.it**, **math.it**, cs.ni
無序遞迴式類神經網路	**math.it**, **cs.it**	**math.it**, **cs.it**, stat.th, math.st	**math.it**, **cs.it**, cs.ni
最佳補全蒸餾法	**math.it**, **cs.it**, cs.ds	stat.ml, stat.th, cs.lg, math.st	**cs.it**, **math.it**
最佳補全蒸餾法 + 多目標學習 + 對數機率共同解碼	**cs.it**, **math.it**	math.st, stat.th, stat.ml, cs.lg, **cs.it**, **math.it**	**math.it**, **cs.it**

表 4.9: 一些模型在Arxiv學術論文資料集上的預測結果的例子

模型預測結果的例子

表4.9是在Arxiv學術論文資料集上，不同模型的預測結果的例子。注意到**math.it**和**cs.it**兩個標籤，序列到序列模型只會依照出現次數多到出現次數少的順序預測標籤，和正確答案的標籤順序相同。然而無序遞迴式類神經網路會自己學習到一個特定的標籤順序，而最佳補全蒸餾法模型會以不同的順序產生標籤，因為它在訓練時就會遇到各式各樣的標籤順序。

4.6 與基於生成對抗網路的多標籤分類器比較

為了比較本章提出的方法和前一章基於增進版霍氏生成網路的模型，我們在谷歌音訊集(Google Audio Set)上比較此兩種方法的優劣。

4.6.1 實驗資料集

我們使用了谷歌音訊集 [37]來進行以下的實驗。谷歌音訊集是一個環境音分類的資料集，環境音的種類總共有527種，包含了來自人或動物的聲音、樂器聲等環境音。此資料集總共有2,084,320段由人類標注的來自Youtube的10秒音訊，因為谷歌只有釋出由ResNet-50模型抽取的特徵，每段的10秒音訊會有10個128維度的特徵向量，我們便使用此作為模型的輸入。一些谷歌音訊集標籤的例子如下表4.10

doi:10.6342/NTU201901512

Speech(語音), Male speech(男性語音), Female speech(女性語音), Child speech(小孩語音), Snoring(打鼾), Breathing(呼吸聲), Finger snapping(打響指), Dog(狗), Animal(動物), Orchestra(樂隊), Vibraphone(音琴), Bass drum(低音鼓), Ocean(海洋), Waterfall(瀑布), Doorbell(門鈴), Eruption(火山噴發), Television(電視), Radio(收音機), Machine gun(機槍), Hammer(鐵鎚)

表 4.10: 谷歌音訊集標籤的一些例子。

4.6.2 模型介紹

在這個章節中，會介紹如何使用第三章基於生成對抗網路的多標籤分類器和第四章提出基於最佳補全蒸餾法的方法，兩個模型的架構。

編碼器架構

首先我們介紹實驗中所有的模型使用的編碼器，我們採用了余氏等人(Yu et al.) [38]提出的架構，模型架構如圖4.6。

如圖4.6，輸入向量會先經過4層全連接層和批次正規化(batch normalization)，並取最後兩層的輸出向量給專注模組，最後將兩個專注模組的輸出向量做連接，得到編碼器的輸出。

專注模組如圖4.7，模組的輸入是大小為(10,512)的向量$x = (x_1, x_2, ..., x_{10}), x_1, x_2...x_{10} \in \mathcal{R}^{512}$，爾後會分別經過兩個大小為(512,527)的全連接層，其中一個全連接層FC_1的激活函數是軟性最大化$v(x_t) = \text{softmax}(FC_1(x_t))$;另一個全連接層$FC_2$則是S函數，$f(x_t) = \text{sigmoid}(FC_2(x_t))$，最後$v(x_t)$會對第一個維度做正規化，爾後和$f(x_t)$逐點相乘，得到最後的結果，式子如下:

doi:10.6342/NTU201901512

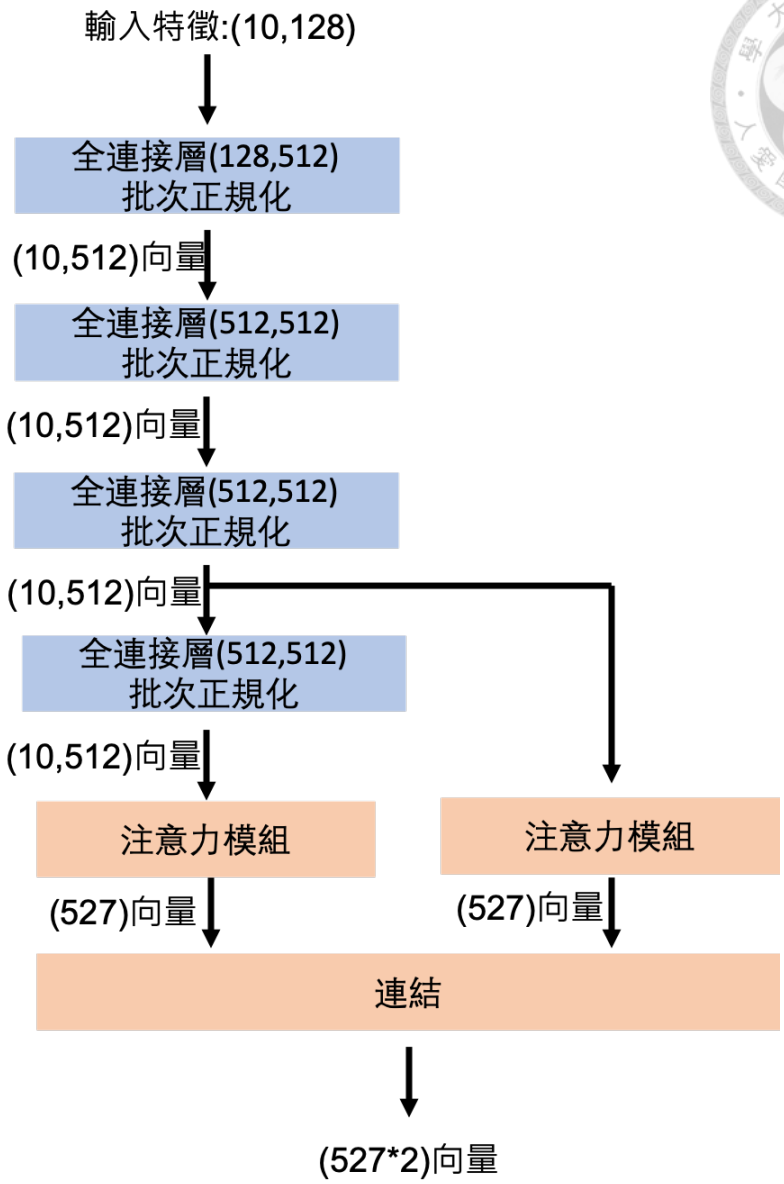


圖 4.6: 編碼器架構示意圖。

$$y = \frac{1}{\sum_{t=1}^{10} v(x_t)} \sum_{t=1}^{10} v(x_t) f(x_t) \tag{4.26}$$

得到的y大小是(10,527)，此向量會沿著第一個維度相加，得到最後專注模組的輸出結果。

doi:10.6342/NTU201901512

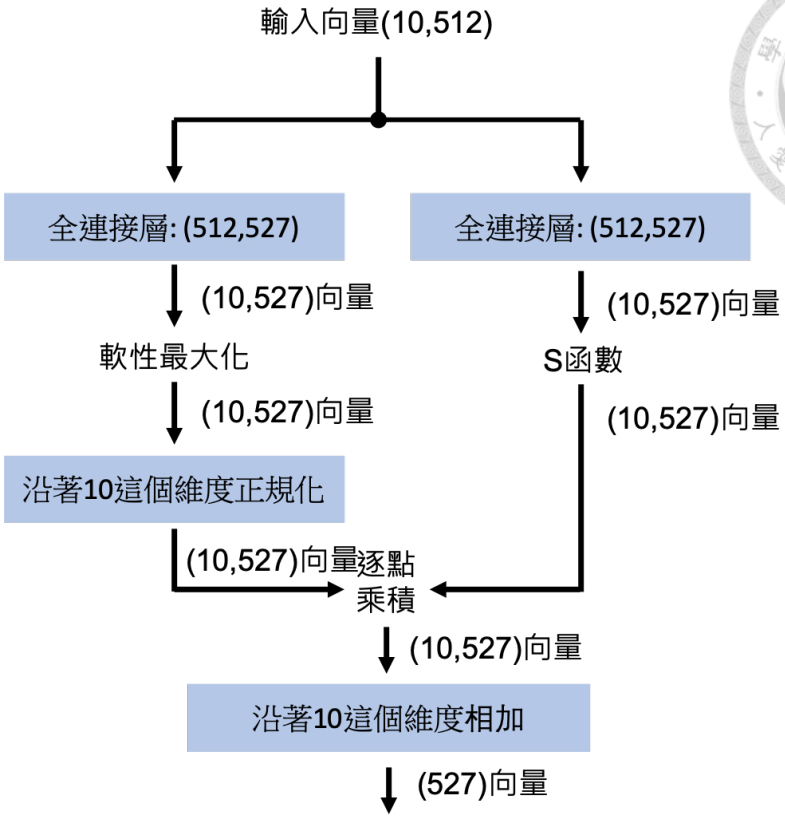


圖 4.7: 專注模組示意圖。

基於生成對抗網路的多標籤分類器

其分類器是由編碼器後面接一層全連接層和S函數所組成，最後會輸出527維的向量，是每個標籤的輸出機率。

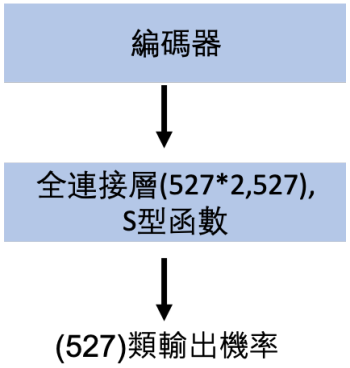


圖 4.8: 基於生成對抗網路的多標籤分類器示意圖。

doi:10.6342/NTU201901512

分類器也是生成對抗網路中的生成器，而鑑別器和章節3.2的模型相同，是由一個固定的特徵抽取器(模型架構和編碼器相同)和幾層的全連接層構成，訓練方法也相同。基準模型中的二元關聯模型也使用此架構。

遞迴式神經網路的多標籤分類器

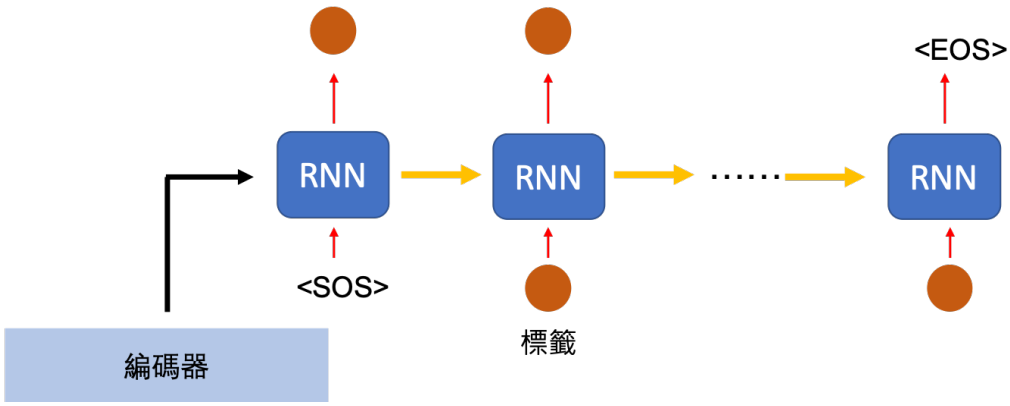


圖 4.9: 遞迴式神經網路的多標籤分類器示意圖。

基於遞迴式神經網路的多標籤分類器的模型架構如圖4.9，是由編碼器和遞迴式神經網路解碼器所構成，解碼器的第一個隱藏層會由編碼器的輸出來初始化，其中，最佳補全蒸餾法模型和基準模型中的遞迴式類神經網路解碼器模型和無序遞迴式類神經網路模型也使用此架構。

多目標訓練的多標籤分類器

最佳補全蒸餾法 + 多目標學習的模型架構如上圖4.10，結合了上述兩種的解碼器做多目標訓練。

doi:10.6342/NTU201901512

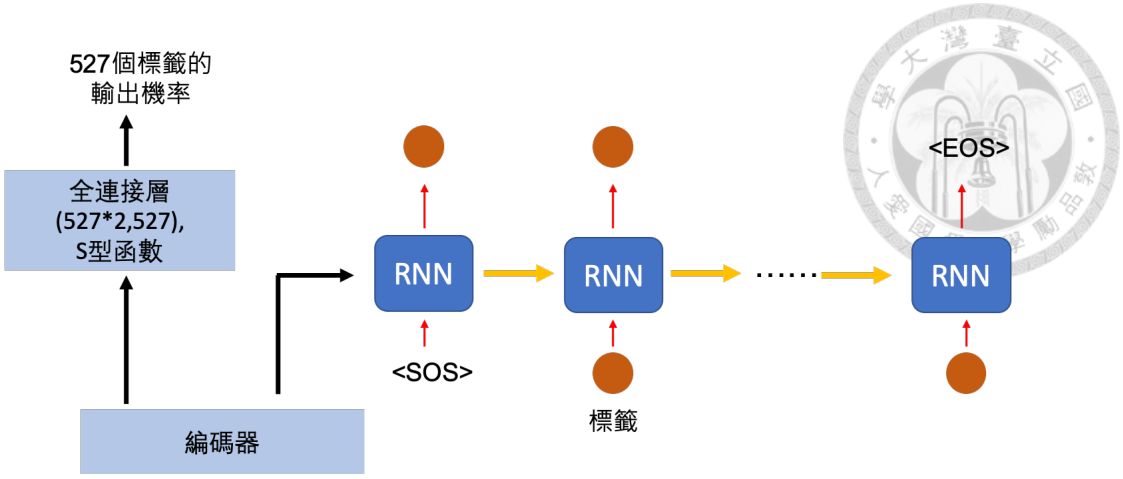


圖 4.10: 多目標訓練的多標籤分類器示意圖

4.6.3 實驗設定

本章將會分別介紹基準模型、實驗使用的超參數和模型訓練的細節，使用的評分標準和章節4.5.3所述的相同，使用了五種評分標準，分別是宏F1分數、微F1分數、基於實例的F1分數、子集正確率和漢明正確率。

基準模型介紹

- **二元關聯**:是使用對數機率回歸的減損函數(式4.16)訓練的模型，模型架構和基於增進版霍氏生成網路模型的分類器架構相同。
- **遞迴式類神經網路解碼器**:這個模型是使用最大似然估計來訓練，和章節4.5.2中的序列到序列模型的訓練方法相同，因為編碼器沒有使用遞迴式類神經網路，因此沒有使用序列到序列這個名字。其網路架構如圖4.9。
- **無序遞迴式類神經網路**:和章節4.5.2中的無序遞迴式類神經網路訓練方法相同。使用了和遞迴式類神經網路解碼器相同的網路架構，而訓練方法不同。

doi:10.6342/NTU201901512

超參數和訓練細節

我們使用了Pytorch來實作實驗。以下會分別介紹基於生成對抗網路的多標籤分類器和最佳補全蒸餾法模型的訓練細節。

- **基於生成對抗網路的多標籤分類器**: 實驗中的鑑別器中的特徵抽取器是由用對數機率回歸訓練於同一資料集而得的編碼器構成，在之後訓練特徵抽取器的參數時並不會被更新。在鑑別器中，音訊的特徵$z = f_{ext}(x)$和標籤集會分別被線性投射到256維的向量，然後兩者會被連結成512維的向量，接著送入5層具有溢出整流線性單元(leaky relu)的全連結層。根據我們實驗的觀察，在生成對抗網路的訓練中，我們更新鑑別器2次才會更新分類器一次。訓練分類器和鑑別器時，我們使用了亞當優化器(Adam optimizer)，學習率是0.0001，對數機率回歸損失和生成器損失的加權值λ是10，而岡氏軟性最大化使用的溫度T是0.9。

- **最佳補全蒸餾法模型**:我們使用學習率是0.0005的亞當優化器。為了避免過度貼合的問題，我們不但使用了丟棄法，也限制了梯度的大小在$[-10, 10]$之間。在使用最佳補全蒸餾法訓練的模型中，我們將溫度τ設為10^{-8}。在多目標學習的模型中，兩個損失函數的加權值λ為1。在測試模型時，我們使用貪婪解碼(greedy decoding)，即在每個時間點皆選取機率最大的標籤。

4.6.4 實驗結果

表4.11是在谷歌音訊集上的實驗結果。我們可以發現不同的模型擅長於不同的評分標準，像是二元關聯(列(a))和二元關聯 + WGAN-gp(列(d))在宏F1分數和微F1分數上表現較好，子集正確率卻十分的低，而基於遞迴式類神經網路的多標籤分類

doi:10.6342/NTU201901512

器的模型(列(b),(c),(e))，在子集正確率上比二元關聯類的模型高上不少。最佳補全蒸餾法 + 多目標學習模型則在每種評分標準都有不錯的表現。

比較列(a)和列(d)，二元關聯 + WGAN-gp子集正確率相對高了一些(0.008對比0.073)，綜合起來也表現較好(平均值0.461對比於0.473)，證明了生成對抗網路幫助了二元關聯分類器。另外，本章提出的模型(列(e),(f))，除了在子集正確率上，其餘標準皆較基於遞迴式類神經網路的多標籤分類器好(列(b),(c))，和在多標籤文件分類的實驗結果相似。

比較列(d)和列(f)，可以發現最佳補全蒸餾法 + 多目標學習模型在基於實例的F1分數和子集正確率皆超過二元關聯 + WGAN-gp許多，而其餘三個評分標準的數值皆差不多，因此本章提出的方法應較基於生成對抗網路的多標籤分類器(第三章)好。

doi:10.6342/NTU201901512

模型	宏F1分數	微F1分數	基於實例的F1分數	子集正確率	漢明正確率	衡量標準的平均值
基準模型						
(a)二元關聯	**0.377**	0.488	0.437	0.008	0.9942	0.461
(b)遞迴式類神經網路解碼器	0.332	0.428	0.415	0.132	0.9940	0.460
(c)無序遞迴式類神經網路	0.333	0.432	0.415	**0.136**	0.9943	0.462
本論文提出的方法						
(d)二元關聯 + WGAN-gp (第三章)	0.372	**0.491**	0.435	0.073	**0.9953**	0.473
(e)最佳補全蒸餾法 (第四章)	0.355	0.463	0.439	0.129	0.9946	0.476
(f)最佳補全蒸餾法 + 多目標學習 + 對數機率共同解碼 (第四章)	0.376	0.485	**0.457**	0.134	0.9950	**0.489**

表 4.11: 谷歌聲音集上的實驗結果。

doi:10.6342/NTU201901512

4.7 本章總結

在本章中，我們介紹了基於最佳補全蒸餾法和多目標學習的模型架構和訓練方法，其中，多標籤分類器會有兩個解碼器，分別是遞迴式類神經網路解碼器和二元關聯解碼器。我們也提出了兩種方法結合兩解碼器的預測機率，作為最終的預測結果。最後在實驗中，我們在三個多標籤文件分類資料集上，證明了提出的模型表現較好，並在實驗結果討論中，分析了在多標籤分類上時序採樣的有效性和曝光偏差的問題。最後，我們在多標籤環境音分類上和上一章提出的方法做比較。

doi:10.6342/NTU201901512

第 五 章　結論與展望

5.1 研究貢獻

本論文主要的研究方向是多標籤分類系之新技術統，在第三章時，我們介紹了如何利用生成式對抗網路來幫助二元關聯模型的表現，而第四章則是使用最佳補全蒸餾法改進基於遞歸式類神經網路的多標籤分類器。本論文的主要貢獻條列如下:

- 在前人的研究中，已經證明了模擬標籤間的關聯性，可以增進分類器的表現，本論文提出了基於增進式霍氏生成對抗網路的模型架構，在此架構下，多標籤分類器不僅需要學會正確的標籤間的關係以欺騙鑑別器，也需要學會輸入物件和標籤的關係。在實驗中，對於不同的資料集和不同的分類器架構，我們提出的訓練方法皆能使模型表現較好，另外在後續的實驗中也證明了其廣泛化能力也較强。

- 另一方面，以往基於遞歸式類神經網路的多標籤分類器，不僅需要人為定義的標籤順序來做訓練，而且會有過度貼合的問題。因此，我們提出了使用最佳補全蒸餾法來改進此類多標籤分類器，其不僅不需仰賴標籤順序便可訓練，也能改善曝光偏差和過度貼合的問題。

- 另外，我們提出了使用二元關聯解碼器在多目標訓練的框架下，使基於遞歸式類神經網路的多標籤分類器學習得更好，並在測試階段，我們提出了兩種解碼方式來結合兩解碼器的預測結果，更加改進了模型的預測結果。在實驗中，我們也使用了三種多標籤文件分類的資料集，證明了我們提出的模型較好，並改善曝光偏差和過度擬合的問題。

doi:10.6342/NTU201901512

- 最後，我們在多標籤環境音分類上，比較了第三章和第四章的方法，證明了使用最佳補全蒸餾法及多目標學習的模型有較好的表現。

5.2 未來展望

以下分別說明第三章和第四章提出的模型未來可以改進的方向。

5.2.1 以生成式對抗網路幫助多標籤分類器

歷經了數次的嘗試後，我們使用了增進版霍氏生成對抗網路，並有最好的表現，但生成對抗網路的訓練方法日新月異，目前已有新的模型架構或正則化方法能嘗試 [39] [40]，或許能使模型的表現更好。在實驗的部分，我們只使用了多標籤影像分類和多標籤環境音分類的資料庫，若要證明模型在任何情況下，都能藉由模擬標籤關聯性增進模型的表現，或許可以再嘗試別的資料集，例如第四章的多標籤文件分類資料集。

5.2.2 最佳補全蒸餾法應用於多標籤分類

在實驗部分的路透社資料集卷一上，最佳補全蒸餾法帶來的進步量十分的少，我們猜測的原因可能是因為其丟棄了標籤的順序性，但是在此資料集上，標籤具有階層性的性質，因此，先預測大分類再預測小分類，會是一個較合理的推斷標籤關聯性的方法，也是序列到序列模型在此資料集表現相當好的原因。因此，或許我們能藉由稍微更動最佳補全蒸餾法的訓練方法，使此模型能利用標籤階層性的性質，或許能使此模型在這類型的多標籤分類問題上得到更好的預測結果，其中一個可能的方法便是使用兩種解碼器，一個是序列到序列模型的解碼器，另一個

doi:10.6342/NTU201901512

是最佳補全蒸餾法使用的解碼器，或許模型便能藉由序列到序列模型的解碼器學到標籤的順序性。

doi:10.6342/NTU201901512

參 考 文 獻

[1] Qiang Li, Maoying Qiao, Wei Bian, and Dacheng Tao, "Conditional graphical lasso for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2977–2986.

[2] Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Advances in neural information processing systems*, 2017, pp. 5413–5423.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[5] Alexander H Liu, Hung-yi Lee, and Lin-shan Lee, "Adversarial training of end-to-end speech recognition using a criticizing language model," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6176–6180.

[6] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings*

doi:10.6342/NTU201901512

of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2285–2294.

[7] Sara Sabour, William Chan, and Mohammad Norouzi, "Optimal completion distillation for sequence learning," *arXiv preprint arXiv:1810.01398*, 2018.

[8] Grigorios Tsoumakas and Ioannis Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

[9] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333, 2011.

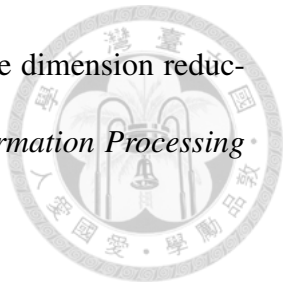
[10] Grigorios Tsoumakas and Ioannis Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *European conference on machine learning*. Springer, 2007, pp. 406–417.

[11] Jesse Read, Bernhard Pfahringer, and Geoffrey Holmes, "Multi-label classification using ensembles of pruned sets," in *8th IEEE international conference on data mining*. IEEE, 2008, pp. 995–1000.

[12] Krishnakumar Balasubramanian and Guy Lebanon, "The landmark selection method for multiple output prediction," *arXiv preprint arXiv:1206.6479*, 2012.

[13] Wei Bi and James Kwok, "Efficient multi-label classification with many labels," in *International Conference on Machine Learning*, 2013, pp. 405–413.

doi:10.6342/NTU201901512



[14] Yao-Nan Chen and Hsuan-Tien Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Advances in Neural Information Processing Systems*, 2012, pp. 1529–1537.

[15] Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang, "Learning deep latent space for multi-label classification.," in *AAAI*, 2017, pp. 2838–2844.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[19] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[20] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

doi:10.6342/NTU201901512

[22] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[26] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe, "Deep convolutional ranking for multilabel image annotation," *arXiv preprint arXiv:1312.4894*, 2013.

[27] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 464–472.

[28] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang, "Order-free rnn with visual attention for multi-label classification," *arXiv preprint arXiv:1707.05495*, 2017.

doi:10.6342/NTU201901512

[29] Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, and Jianfeng Lu, "Multi-label image classification with regional latent semantic dependencies," *IEEE Transactions on Multimedia*, 2018.

[30] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[31] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang, "Sgm: sequence generation model for multi-label classification," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3915–3926.

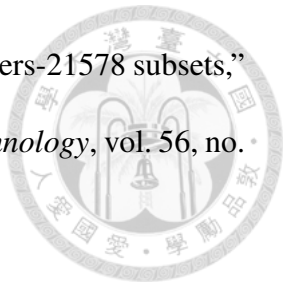
[32] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of machine learning research*, vol. 5, no. Apr, pp. 361–397, 2004.

[33] Lifu Tu and Kevin Gimpel, "Learning approximate inference networks for structured prediction," *arXiv preprint arXiv:1803.03376*, 2018.

[34] José RamóN Quevedo, Oscar Luaces, and Antonio Bahamonde, "Multilabel classifiers with a probabilistic thresholding strategy," *Pattern Recognition*, vol. 45, no. 2, pp. 876–883, 2012.

[35] Pengcheng Yang, Shuming Ma, Yi Zhang, Junyang Lin, Qi Su, and Xu Sun, "A deep reinforced sequence-to-set model for multi-label text classification," *arXiv preprint arXiv:1809.03118*, 2018.

doi:10.6342/NTU201901512



[36] Franca Debole et al., "An analysis of the relative hardness of reuters-21578 subsets," *Journal of the American Society for Information Science and technology*, vol. 56, no. 6, pp. 584–596, 2005.

[37] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[38] Changsong Yu, Karim Said Barsim, Qiuqiang Kong, and Bin Yang, "Multi-level attention model for weakly supervised audio classification," *arXiv preprint arXiv:1803.02353*, 2018.

[39] Cyprien de Masson d'Autume, Mihaela Rosca, Jack Rae, and Shakir Mohamed, "Training language gans from scratch," *arXiv preprint arXiv:1905.09922*, 2019.

[40] Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine, "Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow," *arXiv preprint arXiv:1810.00821*, 2018.

doi:10.6342/NTU201901512