專題

關於深度學習網絡的兩個問題

為什麼需要深度網絡與非線性機制

作者：**郭宗杰 C.-C. Jay Kuo**　譯者：**趙學信**

郭宗杰為南加州大學電機、資訊與數學系教授，現任該校多媒體通訊實驗室主任。其研究領域包括多媒體資料壓縮、傳播與網絡技術、視覺大數據分析、資訊鑑識與安全，是多媒體技術領域的領導人。

使用深度類神經網絡（deep neural network, DNN）的深度學習是機器學習的一門分支。因為以類神經網絡為基礎的學習技術，在許多語音和影像 / 視頻應用上有卓越表現，所以人們最近對它又燃起了強烈的興趣。DNN 最近的成功是因為能取得大量有標註的訓練資料（例如 ImageNet），以及更強大的電腦硬體。深度學習這個名稱的由來，是因為類神經網絡的層數愈多，表現通常愈好。最終形成的網絡及提取出的特徵，分別稱為深度網絡和深度特徵。雖然深度學習的方法，實驗上優於傳統的模式辨識（pattern recognition method），但為何如此，仍缺乏數學理論的解釋。由於對深度學習缺乏扎實的理解，我們有的只是一些經驗規則和直覺，這並不足以深刻的推動科學知識。近年來對 DNN 的詮釋已經有些研究嘗試。

類神經網絡有兩種常用架構：捲積式類神經網絡（convolutional neural network，CNN）[1]　和遞迴式類神經網絡（recurrent neural network，RNN）。CNN 經常用於從點陣圖形直接辨識出影像模式，RNN 則用於從符號或音頻 / 語音波型辨識出時間序列上的模式。CNN 和 RNN 都是多層類神經網絡的特殊類型，它們是藉由反向傳遞（backpropagation）演算法來訓練的。本文主要討論的是 CNN，並嘗試回答兩項關於 CNN 的基本問題：（1）為何每一捲積層的輸出都需要以非線性修剪（non-linear clipping）運算作為激活函數（activation function）？（2）兩層的串接較之一層的優點是什麼？其實這兩個問題是密切相關的。捲積運算本身是線性的，如果移除掉每兩個捲積層之間的非線性運算，則兩個線性系統的串接等價於一個線性系統。那麼我們直接採用一個線性系統即可，而多層網絡架構的必要性就值得商榷了。

為了闡明論點，本文將使用 MNIST 資料集 ⓘ 和 LeNet-5 為例。MNIST 資料集是由 10 個手寫數碼（0、1、……、9）所構成。所有數碼圖片的大小、位置都經過調整，使能成為 32×32 大小的點陣圖。這個資料集裡的訓練資料包含 60,000 個範例，測試資料包含 10,000 個範例。

LeNet-5 是由勒丘（Yann LeCun）等人設計的捲積式網絡 [2]，用於處理手寫和列印的字元辨

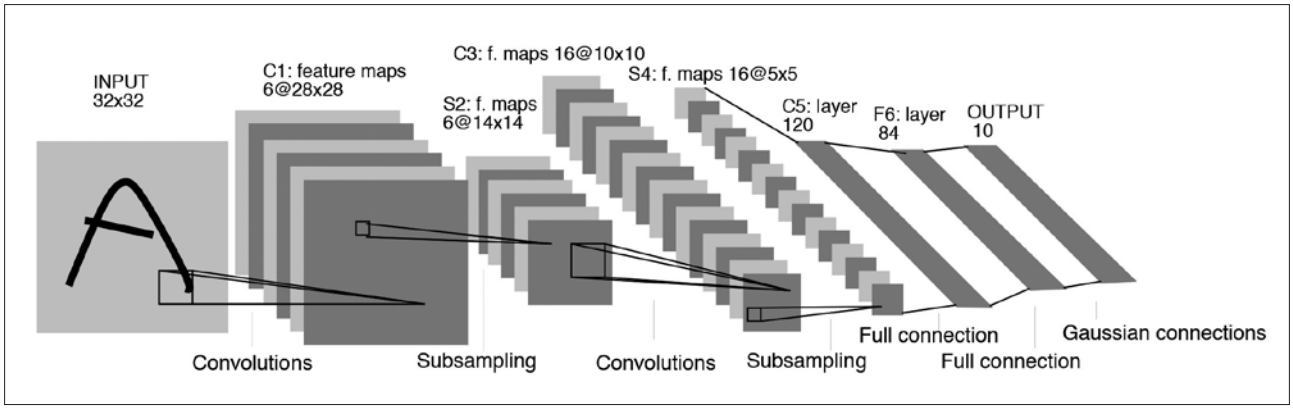

圖 1 LeNet-5 架構，本圖出自勒丘的文章 [2]。

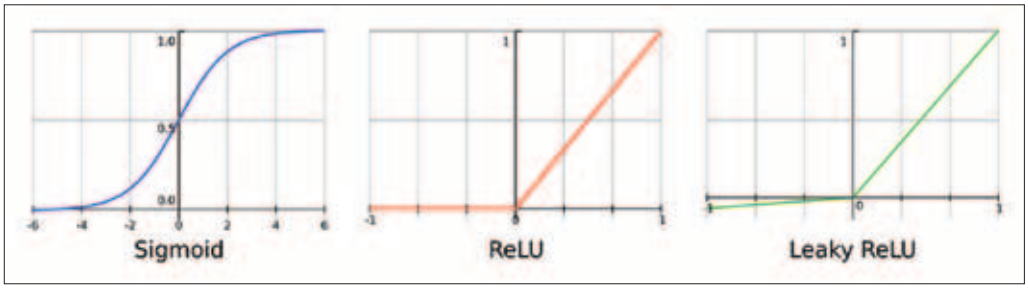

圖 2 CNN 使用的三種激活函數：S 型函數（左）、整流線性單元（中）和參數化整流線性單元（右）。

識，其架構如圖 1 所示。輸入的是 32×32 大小的 8 位元點陣圖，LeNet-5 有兩組捲積 / 池化層（convolutional/pooling layer），在圖中分別標示為 C1/S2 和 C3/S4。C1 有 6 個 5×5 的篩選器（filter），C3 有 16 個 5×5 的篩選器，它們每一個都接著一個非線性激活函數，例如 S 型函數（sigmoid function）。再來還有兩個完全連接層（fully connected layer），標示為 C5 和 F6，它們位於這兩對串接的捲積 / 池化運算之後，在輸出層之前。LeNet-5 對於近年的深度網絡設計帶來很強的衝擊，例如由克利澤夫斯基（Alex Krizhevsky）等人所提出的 AlexNet [3] 即是把 LeNet-5 的兩組捲積 / 池化 / 激活複合層推廣到五組。

問題一：為何非線性激活是必要的？

一般而言，CNN 試圖學習輸入與輸出之間的關係，並將學得的經驗儲存到它們的篩選權重（filter weight）裡。理解 CNN 的一項挑戰是理解捲積運算後，非線性激活單元所扮演的角色。以下的討論將省略池化運算，因為它主要是提供空間維度降低的技巧，並非關鍵的角色。

CNN 使用三種激活函數：S 型函數、整流線性單元（rectified linear unit, ReLU）和參數化整流線性單元（parameterized ReLU, PReLU），參數化整流線性單元也稱為 leaky ReLU，三者如圖 2 所示，它們扮演的是修剪的角色。S 型函數把輸入修剪映射到 0 到 1 的區間；ReLU 把負值修剪到零，正值則維持不變；參數化 ReLU 的作用和 ReLU 類似，但它是藉由降低映射函數的斜率，把負值縮小。根據實驗觀察，如果移除非線性激活運算，系統的表現將失色不少。

每個捲積層都是由它的篩選權重所決定，而這些權重是在訓練階段藉由遞迴更新過程（iterative update process）得到的。亦即，我們先初始化權重，再藉由反向傳遞來將成本函數最小化，於是權重到了測試階段都是定值，這些權重擔任的是「系統記憶」的角色。

在本文中，我們將採用另一個名稱來稱呼篩選權重，以強調它們在測試階段的角色，我們將稱之為「錨向量」（anchor vector），因為它們為每一輸入區塊的測試影像，擔任了參照訊號（或視覺模式）的角色。眾所皆知，所謂訊號捲積（signal convolution）也可以看成是訊號相關度（correlation）或投影量。對於每一輸入影像，我們計算它和各錨向量的相關度（correlation）來衡量它們的相似度。顯然，將輸入影像對這些錨向量的投影提供了該輸入的譜分解（spectral decomposition）。

錨向量通常並非正交也不完備。試考慮 LeNet-5，對於第 1 捲積層（C1），輸入區塊的大小是 5×5=25，它有 6 個同大小的篩選器（或錨向量），因此 C1 中的錨向量維數是 25，個數是 6。第 2 捲積層（C3），輸入是空間與譜的混合表現，維數是 (5×5)×6=150，所以 C3 中錨向量的維數是 150、個數是 16。

在此，我們將捲積之後作非線性激活的合成運算，解釋成是執行「球面整流相關性」（REctified COrrelations on a Sphere, RECOS）的機制。在不

❶ 見 http://yann.lecun.com/exdb/mnist/。

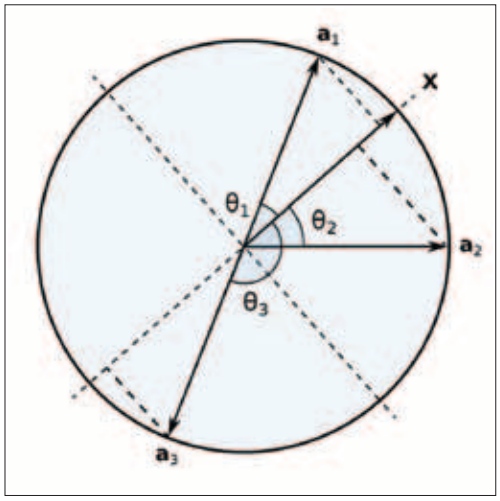

圖 3 以單位圓為例，說明相關度整流的必要性。

致過度簡化的前提下，以下討論將採用 ReLU 作為激活函數。RECOS 模組中的 ReLU 運算會將所有負的相關度化為零（整流的必要性後面會解釋）。首先，我們考慮一個以原點為球心的單位球面。

以原點為球心單位球面的情況

令 $\mathbf{x}=(x_1,\cdots,x_N)^T$ 為 N 維空間中的向量，以原點為球心的單位球面可定義為

$$S=\left\{\mathbf{x}\middle|\|\mathbf{x}\|=(\textstyle\sum_{n=1}^{N}x_n^2)^{1/2}=1\right\}$$

我們感興趣的是球面 S 上向量 $\mathbf{x}$ 彼此之測地距離（亦即大圓距離），因為兩向量的測地距離愈近，（在影像測試時）表示它們愈相似。向量 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 在 S 上的測地距離即其夾角（使用弧度量），可藉由下式算出

(1) $\theta(\mathbf{x}_i,\mathbf{x}_j)=\cos^{-1}(\mathbf{x}_i^T\mathbf{x}_j)$

因為 $\cos\theta$ 在 $0^\circ\le|\theta|\le90^\circ$ 時是單調遞減函數，我們可使用相關度 $0\le\mathbf{x}_i^T\mathbf{x}_j=\cos\theta\le1$ 做為兩向量近似度的測量方式，同理亦可如此測量 S 上叢集向量（cluster vectors）的近似度。但是當 $90^\circ\le|\theta|\le180^\circ$ 時，相關度 $\mathbf{x}_i^T\mathbf{x}_j=\cos\theta$ 是負值，此時相關度就不是測地距離的良好測量。❷

我們以圖 3 的 2 維類比來說明整流的必要性，其中 $\mathbf{x}$ 和 $\mathbf{a}_k$（$k=1,2,3$）分別表示單位圓上的輸入和三個錨向量，θ_i 是各個錨向量與輸入的夾角。因為 θ_1、θ_2 小於 90 度，$\mathbf{a}_1^T\mathbf{x}$ 和 $\mathbf{a}_2^T\mathbf{x}$ 是正值，相關度可以視為是錨向量在輸入上的投影量（反之亦然）。對於正相關，測地距離是投影值的單調遞減函數。這表示相關度愈大，則距離愈短。

但是，角 θ_3 大於 90 度，它的相關度 $\mathbf{a}_3^T\mathbf{x}$ 是負值。$\mathbf{x}$ 和 $\mathbf{a}_3$ 的測地距離很遠，但它們卻有很強的相關度（儘管是負值）。試考慮 $\mathbf{a}_3=-\mathbf{x}$ 的極端例子，這時 $\mathbf{x}$ 和 $\mathbf{a}_3$ 在單位圓上有最遠的測地距離，但它們是完全負相關（請參考圖 4）。因此之故，當相關度為負時，它並不是測地距離的良好指標。

可能有人會想：我們何不用負號來表示較遠的測地距離。但這在沒做非線性修剪運算的多層 RECOS 系統卻行不通。當兩個 RECOS 單元串接時，第 2 個 RECOS 單元的篩選權重可以是正值或負值。如果第 1 個 RECOS 單元的回應（輸出）是負的，負回應和負篩選權重的乘積會是正值；然而正回應和正篩選權重的乘積也是正值。如此一來，系統便無法區分這兩種情形。同理，不做整流的系統也無法區分下列兩種情形：在第 1 層是正回應，接著第 2 層是負篩選權重；或者在第 1 層是負回應，接著在第 2 層是正篩選權重。因此，很重要的一件

圖 4 一張灰階的貓咪照片及其負片影像。它們在移除平均（mean removal）後是負相關的，兩者距離很大（白貓對黑貓）。

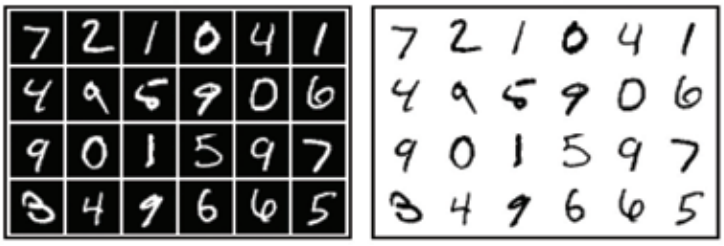

圖 5 MNIST 資料集樣本，圖左是原始影像，圖右是對應的負片影像。

事是，在每一層把負的相關值（亦即回應）設成零（或幾乎為零），以避免在多層 RECOS 系統中造成混淆。

我們不妨做個實驗來驗證整流的重要性。我們用 MNIST 訓練資料集來訓練 LeNet-5，然後在 MNIST 測試資料集得到 98.94% 的正確辨識率。接著，如圖 5，我們用同一網絡來測試原圖的負片灰階影像，正確率陡降到 37.36%。下一步，我們將 C1 的所有篩選權重皆改成其負值，其餘各層維持不變。做過這項微小修改的 LeNet-5，反過來對負片影像的測試集有 98.94% 的正確辨識率，但對原始測試集卻只有 37.36%。我們可以設計一個對圖 5 兩種測試資料都具有高辨識率的新網絡，方法是把第 1 層的錨向量數目加倍。

上述討論可以寫成如下形式。考慮 N 維單位球面上有 K 個錨向量的情形，記為 $\mathbf{a}_k \in \mathbb{R}^N$，$k = 1, \cdots, K$。給定 $\mathbf{x} \in S$，它對 $\mathbf{a}_k$，$k = 1, \cdots, K$ 的 K 個整流相關度定義了一個從 $\mathbf{x}$ 到輸出向量

$$\mathbf{y} = (y_1, \cdots, y_k, \cdots, y_K)^T$$

的非線性變換，其中

$$\text{(2)}\quad y_k(\mathbf{x}, \mathbf{a}_k) = \max(0, \mathbf{a}_k^T \mathbf{x}) \equiv \text{Rec}(\mathbf{a}_k^T \mathbf{x})$$

（2）式的形式就是 ReLU。其他激活函數的形式，例如 S 型函數和參數化 ReLU 也可接受。只要負的相關值都很小，這些向量呈現弱相關，對最終結果的影響就不大。

平移單位球面的情況

我們可以把這個 RECOS 模型進一步推廣到平移的單位球面：

$$\text{(3)}\quad S_\mu = \left\{ \mathbf{x} \middle| ||\mathbf{x} - \mu\mathbf{1}|| = \left[\sum_{n=1}^N (x_n - \mu)^2 \right]^{1/2} = 1 \right\}$$

其中 $\mu = \frac{1}{N} \sum_{n=1}^N x_n$ 是所有 x_n 的平均，且 $\mathbf{1} \equiv (1, \cdots, 1, \cdots 1)^T \in \mathbb{R}^N$ 是所有分量皆為 1 的常數向量。球面 S_μ 把 S 的球心從原點平移到 $\mu\mathbf{1}^T$。需要這個推廣的原因解釋如下。

對於影像問題，$\mathbf{x}$ 的元素 x_n，$n = 1, \cdots, N$ 代表的是輸入影像的 N 個像素值，而 μ 則是所有像素的平均。如果輸入是整張圖，則其平均是對影像理解沒有影響的整體平均，可在處理之前就先移除，因此可設 $\mu = 0$。但如果輸入影像很大，需要把圖分割成較小的區塊，再平行處理所有區塊。此時，每一區塊的平均是不可移除的局部平均，因為保持局部平均，整合起來才能提供整張影像的概貌。這就對應到（3）式的一般情形。

根據（2）式，S_μ 上的輸出可以寫成 $\mathbf{y} = (y_1, \cdots, y_K)$，

$$\text{(4)}\quad y_k(\mathbf{x} - \mu\mathbf{1}, \mathbf{a}_k) = \text{Rec}(\mathbf{a}_k^T \mathbf{x} + \mu a_{k,0})$$

其中 $a_{k,0} = -\sum_{n=1}^N a_{k,n}$。分別在 $\mathbf{x}$ 和 $\mathbf{a}_k$ 中補上一個元素，定義

$$\mathbf{x}' = (\mu, x_1, \cdots, x_N)^T, \quad \mathbf{a}'_k = (a_{k,0}, a_{k,1}, \cdots, a_{k,N})^T$$

就可以把（4）式重寫成

$$\text{(5)}\quad y_k(\mathbf{x}', \mathbf{a}'_k) = \text{Rec}(\mathbf{a}'^T_k \mathbf{x}'), \quad k = 1, \cdots, K$$

雖然 $\mathbf{x}'$、$\mathbf{a}'_k \in \mathbb{R}^{N+1}$，但它們只有 N 個獨立元素，因為它們的第一個元素是從後面 N 個元素計

❷ 儘管（1）式仍然正確得到測地距離。

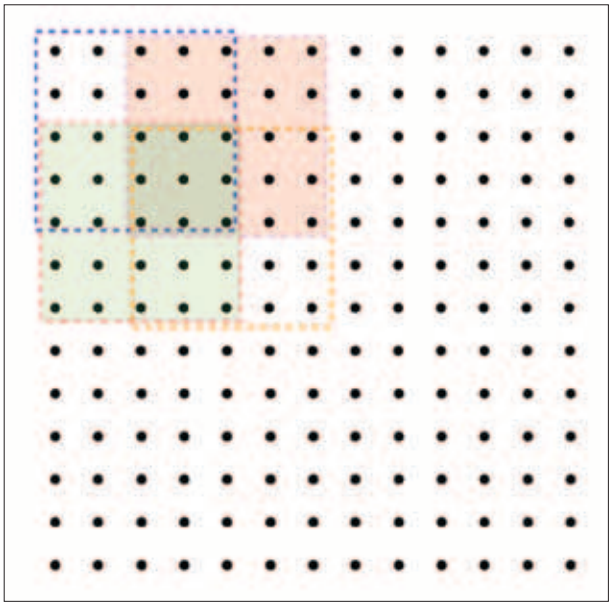

圖 6 LeNet-5 第 1 層和第 2 層篩選器的感知域，圖中每一點代表輸入圖的一個像素，圖中的 5×5 方框表示第 1 層篩選器的感知域，整個 13×13 的點陣代表第 2 層篩選器的感知域。第 2 層篩選器接收第 1 層 5×5 篩選器的輸出。為了閱讀方便，圖中只畫出四個第 1 層篩選器。

算出來的。

進一步而言，輸入向量和錨向量的長度也不一定得是 1。若用 $\mathbf{x}''$ 和 $\mathbf{a}''_k$ 來表示一般情形，然後設

$$\mathbf{x}' \equiv \frac{\mathbf{x}''}{||\mathbf{x}''||}, \quad \mathbf{a}'_k \equiv \frac{\mathbf{a}''_k}{||\mathbf{a}''_k||}$$

則（5）式可重寫成

$$y_k(\mathbf{x}'', \mathbf{a}''_k) = ||\mathbf{x}''||||\mathbf{a}''_k||\text{Rec}(\mathbf{a}'^T_k \mathbf{x}')$$

如果輸入資料中有 K 個經常出現的模式形成 K 個群聚，我們可指定錨向量 $\mathbf{a}_k$（$k = 1, \cdots, K$）為第 k 個叢集的形心。那麼在這叢集裡的資料點將會和 $\mathbf{a}_k$ 產生強相關，而和其他錨向量產生較弱的相關度。在此值得一提的是，論文 [4] 觀察到 K 平均叢集（K-means clustering）在單層網絡是有效的。一個 CNN 會包含多個合作運行的 RECOS 單元，這些單元可以組織成多層架構。多層 CNN 的好處將隨後解釋。

問題 2：多層串接的優點何在？

LeNet-5 基本上是具有兩個捲積層的類神經網絡，因為在類神經網絡的現代文獻中，捲積 / 取樣 / 非線性激活的合成運算被視為是一個完整層。

LeNet-5 第 1 層的輸入是純空間訊號。第 2 層的輸入則是譜 / 空間混合訊號，包含來自 6 個譜帶（spectral bands）的空間訊號 [3]。篩選器所覆蓋的輸入影像空間範圍稱為它的感知域（receptive field），LeNet-5 第 1 層和第 2 層的感知域分別是 5×5 和 13×13。13×13 感知域中的每一空間位置，可能會被 1, 2 或 4 個第 1 層篩選器所覆蓋，如圖 6。

接下來我們要對串接系統的行為進行數學分析，以說明深度網絡的一些優點。在下面的討論中，我們先談一個第 1 層 RECOS 單元和一個第 2 層 RECOS 單元的串接，然後推廣到多個第 1 層 RECOS 單元和一個第 2 層 RECOS 單元的串接。為了便於討論，我們假定所有輸入的平均為零。

一對一串接

我們定義兩個錨矩陣：

$$\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_k \cdots, \mathbf{a}_K], \ \mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_l \cdots, \mathbf{b}_L]$$

其中矩陣的行向量分別是兩個 RECOS 單元的錨向量 $\mathbf{a}_k$ 和 $\mathbf{b}_l$。顯然 $\mathbf{A} \in \mathbb{R}^{N \times K}$ 且 $\mathbf{B} \in \mathbb{R}^{K \times L}$。為了便於處理，我們先做相關分析，最後再考慮非線性整流的效應。在相關部分，令 $\mathbf{y} = \mathbf{A}^T\mathbf{x}$ 且 $\mathbf{z} = \mathbf{B}^T\mathbf{y}$，我們有

$$\mathbf{z} = \mathbf{B}^T\mathbf{A}^T\mathbf{x} = \mathbf{C}^T\mathbf{x}, \ \mathbf{C} \equiv \mathbf{AB}$$

顯然，$\mathbf{C} \in \mathbb{R}^{N \times L}$，且其第 (n, l) 分量為

$$c_{n,l} = \boldsymbol{\alpha}_n^T \mathbf{b}_l,$$

其中 $\boldsymbol{\alpha}_n^T \in \mathbb{R}^K$ 是 $\mathbf{A}$ 的第 n 個列向量。$\boldsymbol{\alpha}_n$ 的意義如圖 7 所示。

就數學而言，我們分解

$$\mathbf{x} = \sum_{n=1}^{N} x_n \mathbf{e}_n$$

其中 $\mathbf{e}_n \in \mathbb{R}^N$ 是第 n 個坐標基底向量。則列向量可表為

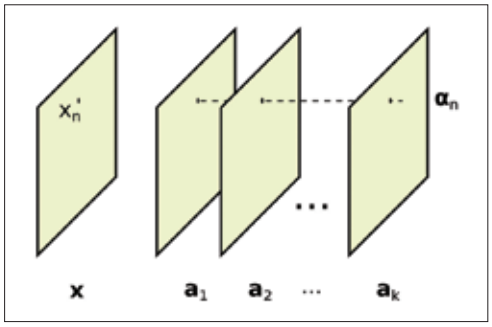

圖 7 錨位置向量 $\boldsymbol{\alpha}_n$ 示意圖。

$$\boldsymbol{\alpha}_n = \mathbf{A}^T \mathbf{e}_n$$

因為 $\boldsymbol{\alpha}_n$ 捕捉了錨向量在 $\mathbf{A}$ 中的位置資訊，所以被稱為錨位置向量（anchor-position vector）。最後，我們對 $\mathbf{C}$ 的所有負元素做整流，有此可獲得從 $\mathbf{x}$ 到 $\mathbf{z}'$ 的錨矩陣 $\mathbf{C}'$：

$$\textbf{(6)}\ \mathbf{z}' = \mathbf{C}'^T \mathbf{x}, \quad \mathbf{C}' = [c'_{n,l}]_{N\times L}$$

其中

$$\textbf{(7)}\ c'_{n,l} = \mathrm{Rec}(c_{n,l}) = \mathrm{Rec}(\boldsymbol{\alpha}_n^T \mathbf{b}_l)$$

嚴格來說，$\mathbf{z}$ 和 $\mathbf{z}'$ 是不同的。前者未做整流運算，而後者則對矩陣積做整流。因為真正的系統對兩層的輸出都做了整流，而最後的結果 $\mathbf{z}''$可能和 $\mathbf{z}$ 與 $\mathbf{z}'$ 都不同。然而，我們最感興趣的是在兩層都具有強烈正相關且不必整流的區域。在這些區域有 $\mathbf{z} \approx \mathbf{z}' \approx \mathbf{z}''$，因此以上的分析只針對這種情形。

多對一串接

把一對一串接的情形推廣到多對一是很直接的。第 1 層 RECOS 單元的相關度可以寫成

$$\mathbf{Y} = \mathbf{A}^T \mathbf{X}$$

其中

$$\mathbf{Y} = [\mathbf{y}_1, \cdots \mathbf{y}_P], \quad \mathbf{X} = [\mathbf{x}_1, \cdots \mathbf{x}_P]$$

在第 1 層有 P 個 RECOS 單元平行運作（如圖 6）。它們覆蓋了空間上鄰近的區域，但共享同一個錨矩陣，可以用來提取不同區域的共同表現模式（common representative patterns）。第 2 層 RECOS 的相關度可以表成 $\mathbf{z} = \mathbf{B}^T \tilde{\mathbf{y}}$，其中 $\mathbf{z} \in \mathbb{R}^L$、$\mathbf{B} \in \mathbb{R}^{PK\times L}$ 和 $\tilde{\mathbf{y}} = (\mathbf{y}_1^T, \cdots, \mathbf{y}_P^T)^T \in \mathbb{R}^{PK}$ 是由第 1 層 RECOS 單元的 P 個輸出向量串接所形成的。

錨矩陣 $\mathbf{A}$ 提取不同區域的表現模式，同時錨矩陣 $\mathbf{B}$ 則是把這些與空間相關的表現模式綴合起來，以形成更大的表現模式。打個比方，考慮一片由多塊小草皮所構成的大草地，假設草皮模式可由 $\mathbf{A}$ 中的一個錨向量捕捉，那麼 $\mathbf{B}$ 中的錨向量可以提供疊加規則，把這些空間分布不同的草皮錨向量綴合起來，以形成更大的草地。

一層系統和兩層系統的比較

要解釋深度網絡的優點，試比較（6）式的兩層系統與下式的一層系統

$$\mathbf{z} = \mathbf{D}^T \mathbf{x}$$

其中 $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_L] \in \mathbb{R}^{N\times L}$ 是以 $\mathbf{d}_l$ 為其錨向量的錨矩陣。錨矩陣 $\mathbf{A}$ 和 $\mathbf{D}$ 在捕捉 $\mathbf{x}$ 全域經常出現的模式時，基本上扮演了相同的角色。然而兩階段系統另有一個錨矩陣 $\mathbf{B}$ 用於捕捉 $\mathbf{y}$ 的表現模式。若要完全理解 $\mathbf{A}$ 和 $\mathbf{B}$ 的合成效應，最好的辦法是檢驗 $\mathbf{C}'$ 的錨向量。根據（6）式，$\mathbf{C}'$ 的錨向量是

$$\mathbf{c}'_l = (c'_{1,l}, \cdots, c'_{n,l})^T,\ l = 1, \cdots, L$$

其中 $c'_{n,l}$如（7）式，是 $\boldsymbol{\alpha}_n$ 和 $\mathbf{b}_l$ 的整流內積。錨向量 $\mathbf{a}_k$ 捕捉區域的整體表現模式，但在捕捉位置敏感資訊（position sensitive information）方面的能力很弱。要彌補這項缺陷，可以用錨位置向量 $\boldsymbol{\alpha}_n$ 的分量對 $\mathbf{b}_l$ 來做調節（modulating）。

底下用一個例子來說明。首先，我們擴大 MNIST 的訓練和測試資料集 [2]，方法是為原來的 MNIST 手寫數字影像加上 10 種不同的背景，如圖

❸ 記得第 1 層有 6 個錨向量，因此譜分解有 6 個分量。

圖 8 上方兩列是加上十種不同背景的 MINST 資料集；下方三列的左邊是輸入圖，中間是第 1 層的 6 個譜頻道，右邊是第 2 層的 16 個譜頻道。

8 的上方兩列所示。在下方三列裡，最左邊是輸入的三張數位影像，中間是捲積層和 ReLU 層中 6 個譜輸出影像，最右邊兩欄是第二層的 16 個譜輸出影像。因為背景差異很大，第 1 層很難找到好的錨矩陣。然而從空間區域位置著眼，這些圖的背景雖不一致，但前景的數字卻是一致的。因此，藉由使用第 1 層的錨位置向量 $\boldsymbol{\alpha}_n$ 來調節第 2 層的錨向量 $\mathbf{b}_l$，背景可以更容易被濾掉。

完全連接層的角色

一個 CNN 可以分解成兩個子網絡：特徵提取子網絡（feature extraction subnet）和決策子網絡（decision subnet）。對 LeNet-5 而言，特徵提取子網絡包含了 C1、S2、C3 和 S4，決策子網絡包含了 C5、F6 和輸出層（如圖 9）。決策子網絡負有下列三項任務：（1）將從 S4 輸出的譜 / 空間特徵圖 （spectral-spatial feature map）轉換成 C5 中的一個 120 維特徵向量；（2）調整錨向量，使得

表 1 LeNet-5 中 RCS 單元的規格

LeNet-5	RECOS	N	K
C1/S2	S1	(5×5) +1	6
C3/S4	S2	(6×5×5)	16
C5	S3	16×5×5	120
F6	S4	120×1×1	84
Output	S5	84×1×1	10

第 3 欄（N）是該層輸入的維度，第 4 欄（K）是輸出的維度。注意 K 也是錨向量的個數。

表 2 AlexNet 中 RCS 單元的規格

AlexNet	RECOS	N	K
Conv_1	S^1	(3×11×11) + 1	96
Conv_2	S^2	(96×5×5) + 1	256
Conv_3	S^3	(256×3×3) + 1	384
Conv_4	S^4	(384×3×3) + 1	384
Conv_5	S^5	(384×3×3) + 1	256
FC_6	S^6	256×1×1	4096
FC_7	S^7	4096×1×1	4096
FC_8	S^8	4096×1×1	1000

第 3 欄（N）是 SI（I = 1,…8）輸入的維度，第 4 欄（K）是輸出的維度。K 也是錨向量的個數。

它們與坐標基底向量對齊，而在 F6 裡有正確的特徵 / 數字配對；（3）在輸出層做最終的數字分類決策。

C5 和 F6 函數如圖 9 所示。如圖 9(a) C5 為特徵叢集指定錨向量。在 LeNet-5 裡，400-D 空間需要指派（或訓練）120 個錨向量。然後，F6 使用一個錨矩陣來把 C5 中的錨向量旋轉和縮放到新位置。目的是要確保一個物件類別的特徵叢集，可以對齊到同一物件類別的坐標基底向量，以供輸出層決策之用，見圖 9(b)。在輸出層裡，每個坐標基底向量都是錨向量，各別代表一個數位類別。在輸出層做決策時，最常用的是軟最大法（softmax rule，即正規化指數函數〔normalized exponential〕）。

多層 CNN

我們在表 1 列出 LeNet-5 各層的慣用名稱、RECOS 記法和它們的輸入、輸出向量維度。輸出向量的維度數與該層的錨向量個數相同。在 S^1 中，必須做向量擴增，因為它們的局部平均可能不是零；但在 S^2、S^3、S^4 和 S^5 則不需要，因為整體平均已經移除了。

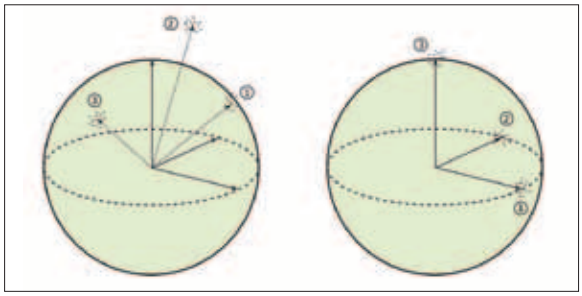

圖 9 函數 C5（左）和 F6（右）的示意圖。

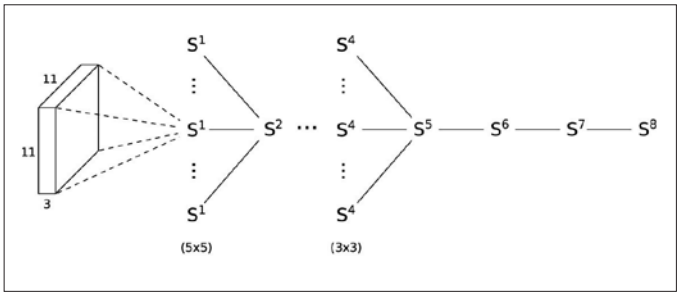

圖 10 使用樹狀結構 RECOS 單元的 AlexNet 組織圖。

接著再以克利澤夫斯基等人在 [3] 中提出的 AlexNet 為例，說明多層 CNN。我們在表 2 列出 AlexNet 各層的慣用名稱、RECOS 記法和它們的輸入、輸出向量維度。AlexNet 樹狀結構 RECOS 單元組織圖如圖 10 所示。我們把第 l 層的 RECOS 單元記為 S^l（$l = 1, \cdots 8$）。S^l 的輸入是一個彩色影像區塊，其大小是 11×11，具有 R、G、B 三個頻道。當我們從 S^l 推進到 S^5，覆蓋區域逐漸變大。它們是用於捕捉位於各個空間位置的各種尺寸的表現視覺模式。

結語與後續問題

本文使用 RECOS 模型來解釋非線性修剪函數在 CNN 中的角色，並使用簡單的矩陣分析來解釋雙層 RECOS 模型優於單層 RECOS 模型的理由。文中提出的 RECOS 數學模型集中在錨向量的選擇。CNN 確實為影像處理和理解提供了非常強大的工具，然而在 CNN 解釋性和更廣泛的應用上，仍有一些待解決的課題，以下列出四項議題：

○**特定應用的 CNN 架構**。在訓練 CNN 時，CNN 架構（包括層數以及每層的篩選器數等）都必須預先指定。給定一個固定架構，篩選權重是藉由一個端對端的優化架構（end-to-end optimization framework）來做優化的。一般而言，小型的 CNN 對簡單的任務可以勝任愉快，然而對於一整類的應用，CNN 架構的設計仍沒有明確的指導原則。錨向量觀點鼓勵我們仔細檢視來源資料的屬性，對來源資料分布的良好理解將有助於設計效率更高的 CNN 架構和更有效的訓練。

○**各種輸入的穩健性（robustness）**。在 [2] 中已證明，LeNet-5 在處理各式各樣的輸入時是穩健的（robust），但是最近也有研究質疑 CNN 的穩健性，例如 [5]。理解這些問題的成因，進以設計出更能防錯的 CNN，將會是有趣的課題。

○**低度監督的學習**。CNN 的訓練需要大量加上標註的資料。蒐集與標註資料的成本很高，而且即使用途相同，不同資料集所用的標註規則也可能不同。減輕標註的負擔，讓 CNN 可以用部分標註或彈性標註的資料來訓練，是很重要的一件事。換句話說，CNN 若要能被廣泛應用，需要將學習從高度監督轉變成低度監督。

○**視頻處理和理解**。目前 CNN 的應用大多是在影像處理和理解上，將 CNN 用於視頻處理和理解的研究仍屬有限。一套有效的視頻處理和理解的對策顯然還付之闕如，考慮 CNN/RNN 混合系統是一種可能的思路。∞

本文參考資料請見〈數理人文資料網頁〉http://yaucenter.nctu.edu.tw/periodical.php

本文出處

本文原稿為英文，感謝作者同意翻譯刊登。作者感謝丁哲航為本文做實驗與製圖。

譯者簡介

趙學信為網路工程師。兼事翻譯、寫作。

延伸閱讀

▶ "LeNet-5, convolutional neural networks" 這是深度學習學者勒丘的網頁，其中有 LeNet-5 的許多資料。

http://yann.lecun.com/exdb/lenet/

▶要順利閱讀本文，可以先閱讀捲積式類神經網絡的簡介。網路上有許多資源，底下是兩個例子：

http://cs231n.github.io/convolutional-networks/#conv

http://deeplearning4j.org/convolutionalnets.html

▶本文作者在南加大主持的 MediaComm 實驗室網站

http://mcl.usc.edu