



AI無線通訊系統實驗

Deep learning Techniques for CSI-based Indoor Wireless Sensing

方凱田 教授

TA: 蕭安紘、王庭惠

教材編寫: 蕭安紘、沈立翔 博士

Introduction (1/2)

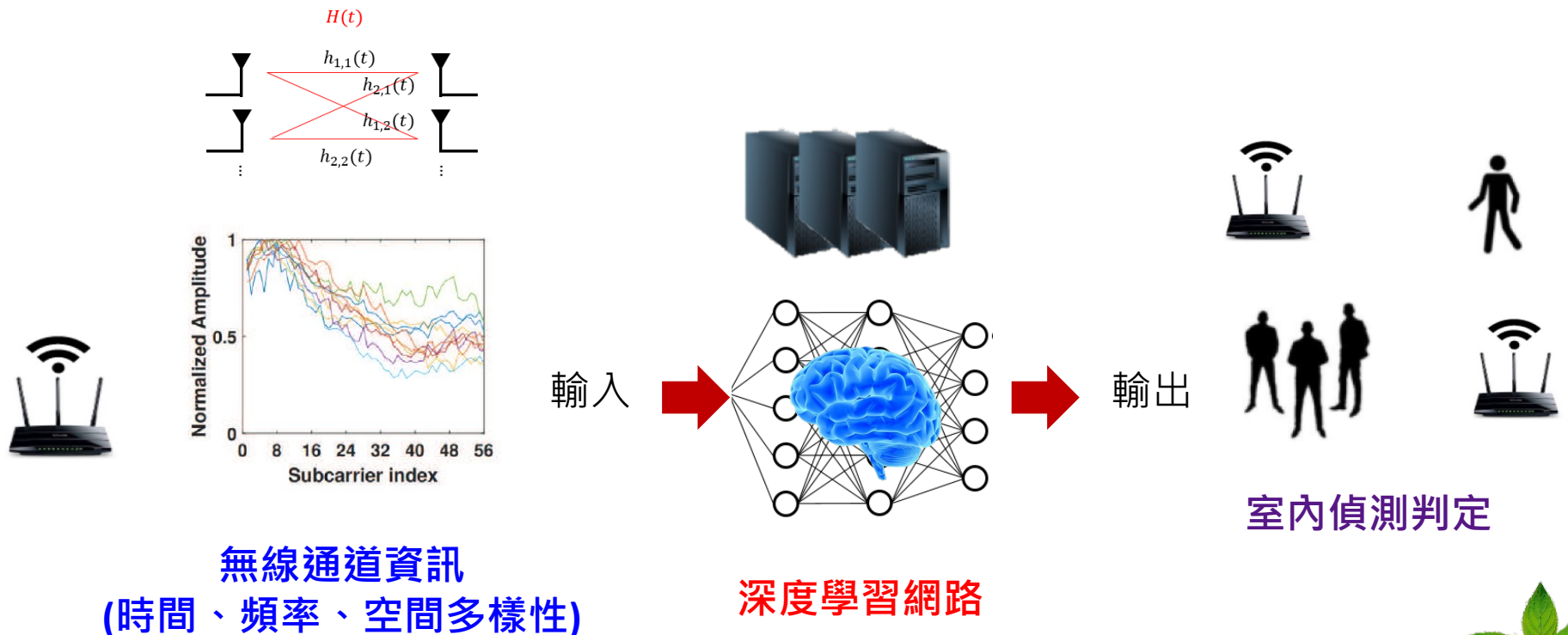
- 無線通訊(Wireless Communication)
 - 指多個節點間不經由導體或纜線傳播進行的遠距離傳輸通訊，包括各種固定式、移動式和可攜式裝置應用，手機、GPS、無線3C產品等，根據不同應用場景有各種無線通訊協定，如
 - 能支援穩定品質高速且數百公尺距離的4G/5G行動通訊
 - 巨量裝置連結的物聯網(Internet-of-Things, IoT)網路
 - 車載網路(Vehicular Communications)、衛星通訊(Satellite Communications)
 - 應用在室內短距離高速傳輸的無線區域網路(Wireless Local Area Network, WLAN)的802.11通訊協定之Wi-Fi技術
 - 更短距離之藍芽(Bluetooth)技術
 - 不同通訊協定的應用有著不一樣的通訊品質需求，且以不干擾無線通訊技術彼此之間的傳輸為原則，去符合因應的無線通訊規範



Introduction (2/2)

Mobile Intelligent Network Technology (MINT) Lab

- 本課程將介紹無線通訊通道，以及透過廣泛利用的Wi-Fi技術進行室內無線人員定位偵測，以大量蒐集Wi-Fi無線通道的資料進行資料處理，並採用人工智慧中基本的機器/深度學習方法來進行精準的室內人員判定

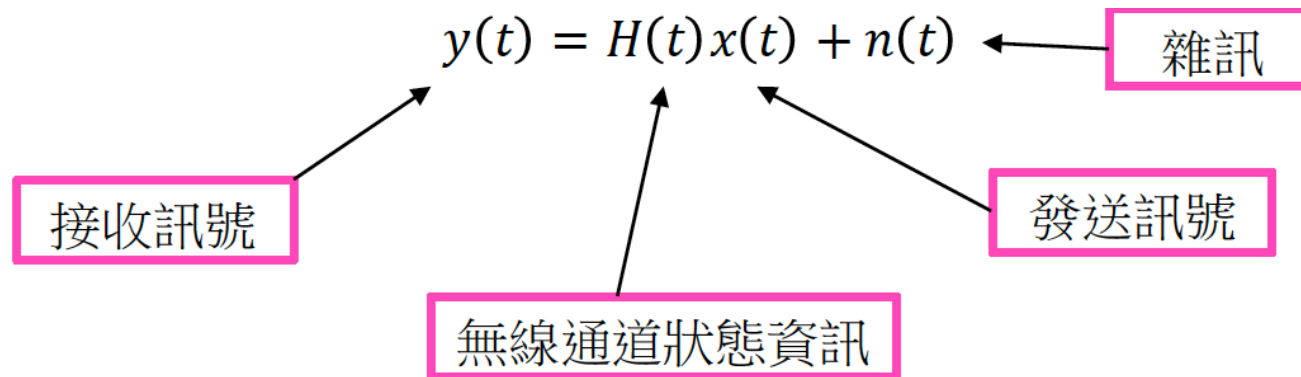




Wireless Channels

無線通道(Wireless Channel) (1/4)

- 考慮一個傳送端裝置(Transmitter, TX)發送訊號並經由無線通道到一個接收端裝置(Receiver, RX)，以把接收到的時變訊號簡單表示為以下數學式：



- 透過通道偵測法能獲得 $H(t)$ 的值，包含振幅 $A(t)$ 和相位 $\theta(t)$ ，表示為

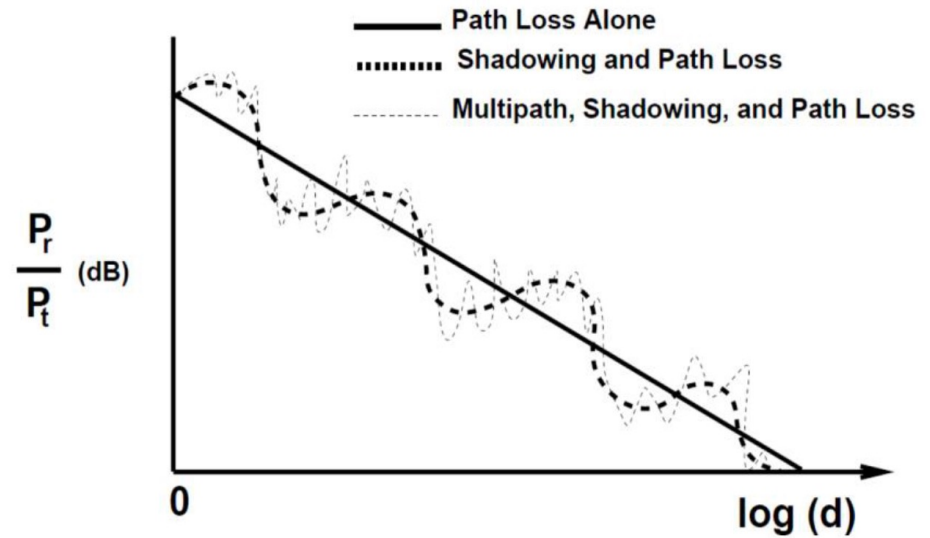
$$H(t) = A(t)e^{j\theta(t)}$$



無線通道(Wireless Channel) (2/4)

Mobile Intelligent Network Technology (MINT) Lab

- 無線通道會受到許多因素的影響
 - 傳送及接收端的距離
 - 環境遮蔽
 - 物體阻礙
 - 波反射等影響



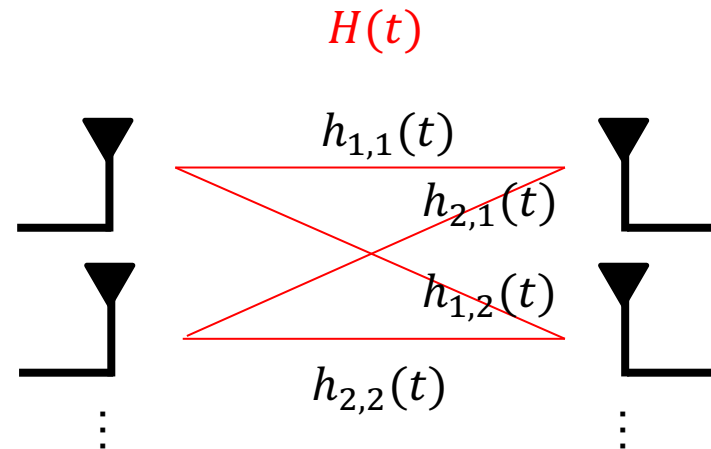
- 通道衰減(Path Loss)發生於傳輸能量在傳輸通道的耗散(Dissipation)，而遮蔽效應(Shadowing)通常是物體吸收、反射、散射和繞射等現象導致，此外亦有多重路徑(Multipath)導致的變動性
- 在複雜的環境時進行室內定位或偵測時，會因環境或人員的微小變化導致無線通道資訊(Channel State Information, CSI)之結果 $H(t)$ 有所不同



無線通道(Wireless Channel) (3/4)

Mobile Intelligent Network Technology (MINT) Lab

- 而在多天線系統架構下，我們可以透過不同天線上的無線通道變化增加我們CSI取得的資料量，例如在一多輸入多輸出(Multi-Input-Multi-Output, MIMO)下，傳送端和接收端配有多根天線， N_t 根傳輸天線和 N_r 根接收天線(Receiver, RX)，如下圖所示2x2的MIMO架構



- 因此CSI可為一矩陣表示成



無線通道(Wireless Channel) (4/4)

- 因此CSI可為一矩陣表示成

$$\mathbf{H}(\mathbf{t}) = \begin{bmatrix} h_{1,1}(t) & \cdots & h_{1,j}(t) & \cdots & h_{1,N_r}(t) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ h_{i,1}(t) & \cdots & h_{i,j}(t) & \cdots & h_{i,N_r}(t) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ h_{N_t,1}(t) & \cdots & h_{N_t,j}(t) & \cdots & h_{N_t,N_r}(t) \end{bmatrix}$$

- $h_{i,j}(t)$ 則為 i-th TX和 j-th RX之 無線通道狀態資訊 CSI值
- 透過不同時間點所蒐集到的 CSI，我們能將其輸入至深度學習網路中進行大量資料的訓練並預測人員之位置判定
- 在實際通訊系統中 (本實驗採用 Wi-Fi系統)，將單一個通道或頻帶切分成很多不同頻率的子載波 (Subcarrier)進行量測，而不同 Subcarrier會因為不同頻率的物理特性而有不同的特徵值
- 如此除了時間特性、MIMO的空間特性外，又多了頻率的特性，將此具備豐富特性的 CSI輸入學習網路進行判定





Artificial Intelligence

人工智慧(Artificial Intelligence) (1/3)

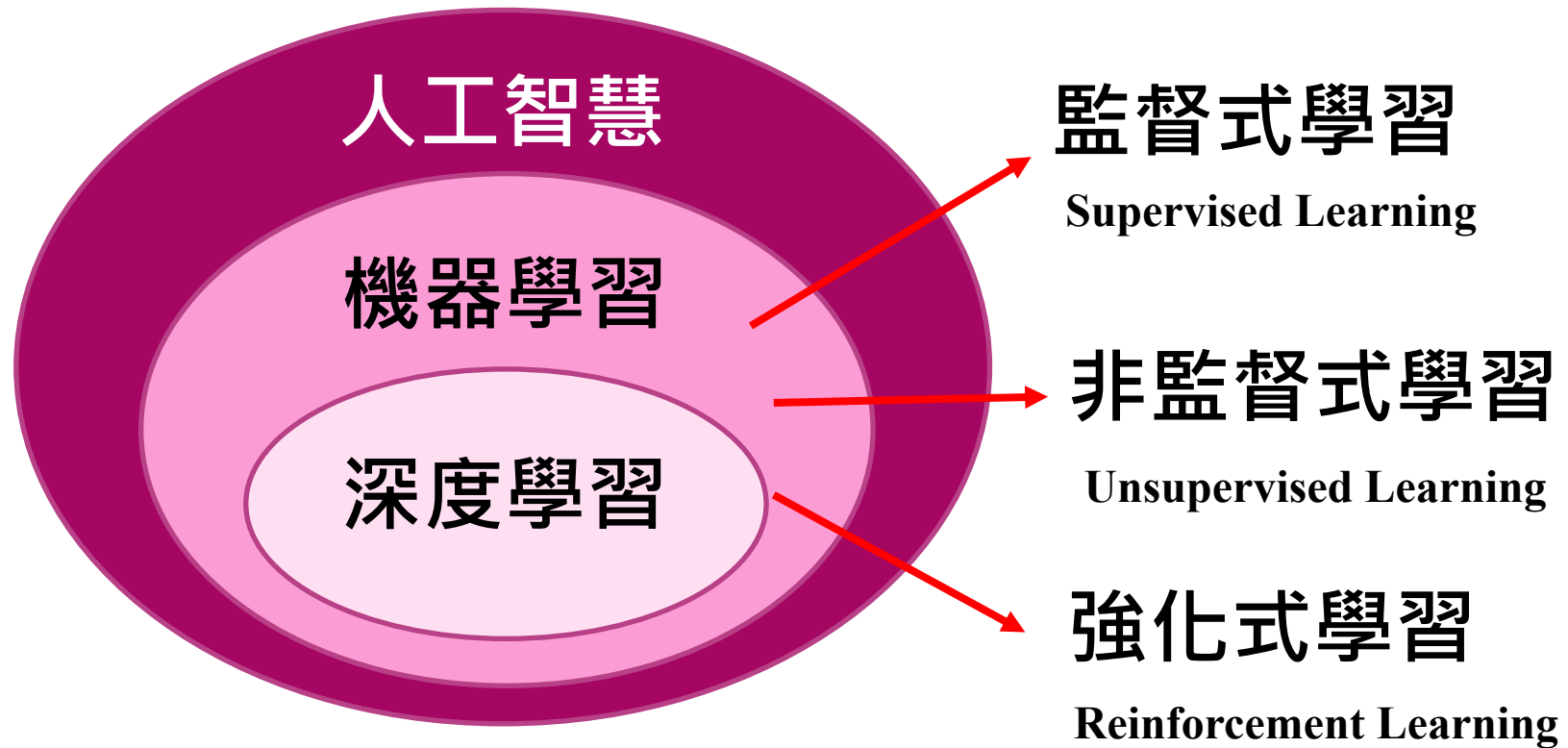
Mobile Intelligent Network Technology (MINT) Lab

- 人工智慧(Artificial Intelligence, AI)是指由人類創造出來的機器所表現出來的智慧，而其子領域包含了
 - 機器學習(Machine Learning)
 - 深度學習(Deep Learning)
 - 差別在於是深度學習有深層神經網路(Neural Network, NN)的建置，能夠自動化處理巨量的資料，並學習其中特徵進行具智慧和適應性之決策
- 機器/深度學習有三種學習法：
 - 監督式學習(Supervised Learning)：擁有正確解答提供網路進行判斷
 - 非監督式學習(Unsupervised Learning)：不提供正確解答，讓網路自行分類判斷
 - 強化式學習(Reinforcement Learning)：提供針對當下環境之參考答案，並非正確解答，而是透過動作、狀態和回饋於變動環境中不斷的更新策略以達適應性



人工智慧(Artificial Intelligence) (2/3)

Mobile Intelligent Network Technology (MINT) Lab



人工智慧(Artificial Intelligence) (3/3)

Mobile Intelligent Network Technology (MINT) Lab

- 本課程將只介紹監督式深度學習中常用的
 - 深度神經網路(Deep Neural Network, DNN)
 - 卷積神經網路(Convolutional Neural Network, CNN)
 - 對抗生成網路(Generative Adversarial Network, GAN)
- 將運用到無線網路中，其它進階深度學習架構皆能藉由此三種基礎神經網路進行變化適應至不同領域的問題上
- 深度學習在輸入和輸出的過程中串接起複雜的神經網路架構，能具備非線性之複雜運算，
 - **研究挑戰**：輸入的資料集(Dataset)蒐集、輸出的解答標記(Label)和中間神經網路建立
 - 可能因為上述三種的任何原因而導致實驗結果不理想，而在研究上目前仍尚缺嚴格的學術理論來證明深度學習相關的特性
- 以下深度學習網路的輸入可想成是無線通道資訊，而輸出則可以想成是我們要進行的室內無線人員定位偵測



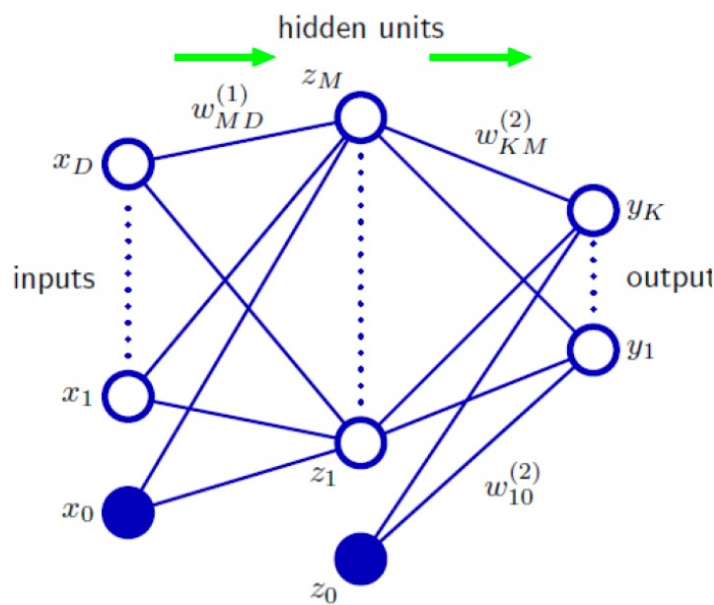


Deep Neural Networks (DNN)

深度神經網路(Deep Neural Network, DNN) (1/5)

Mobile Intelligent Network Technology (MINT) Lab

- 首先要介紹最基本的DNN深度學習網路，包含前饋(Feed Forward)、回饋傳輸(Back Propagation)和目標損失函數(Loss Function)計算



$$a_j^{(1)} = \sum_{i=1}^{D_1} w_{ji}^{(1)} x_i + w_0^{(1)} \quad \text{第一層輸入與Weight和Bias進行線性組合運算}$$

$$z_j^{(1)} = f^{(1)}(a_j^{(1)}) \quad \text{第一層輸出前經過Activation Function}$$

$$a_j^{(2)} = \sum_{i=1}^{D_2} w_{ji}^{(2)} z_i^{(1)} + w_0^{(2)} \quad \text{第二層的輸入值則為第一層之輸出值}$$

$$z_j^{(2)} = f^{(2)}(a_j^{(2)}) \quad \text{第二層之輸出}$$

$$a_j^{(3)} = \sum_{i=1}^{D_3} w_{ji}^{(3)} z_i^{(2)} + w_0^{(3)} \quad \text{第三層的輸入值則為第二層之輸出值}$$

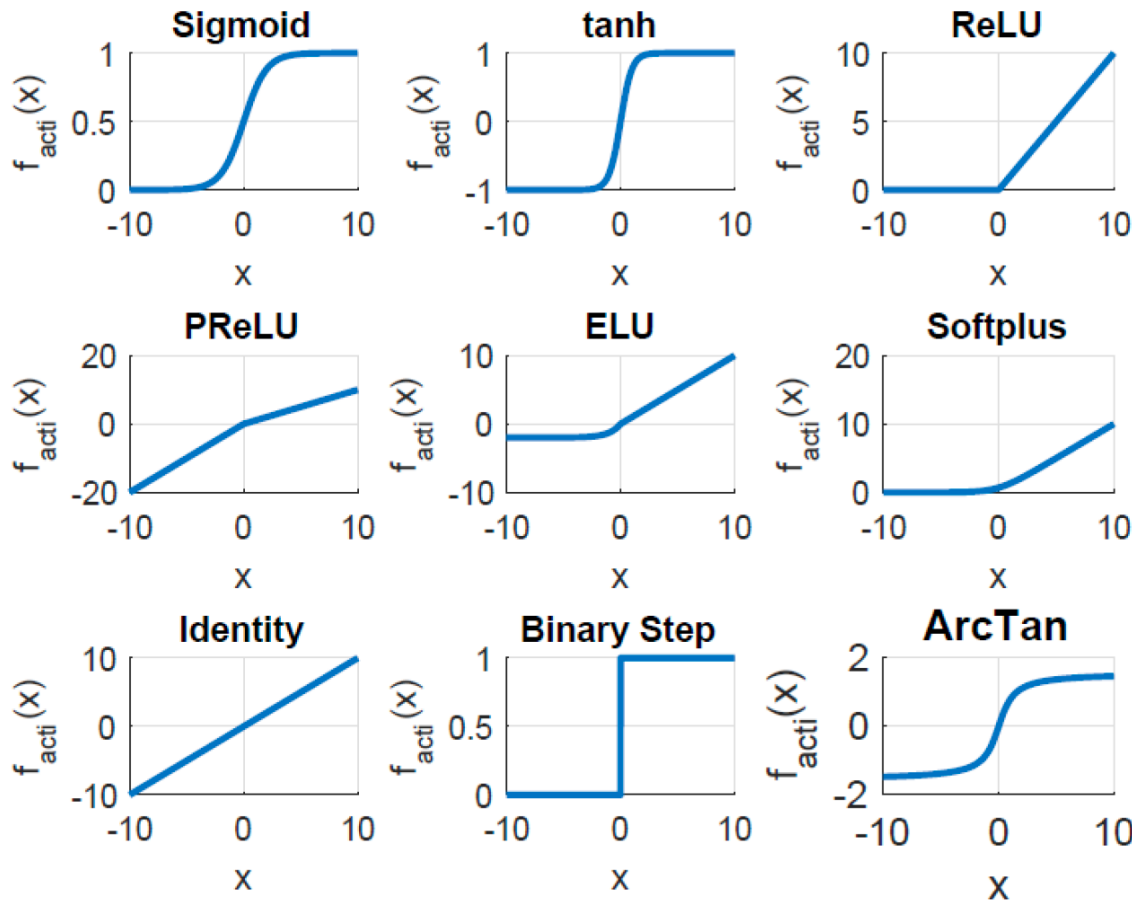
$$y_j = f^{(3)}(a_j^{(3)}) \quad \text{最後一層的輸出}$$

- 在深層網路層(Hidden Layer)中，我們考慮有 D 個特徵(Feature)的輸入資料 x_i ，在 l -th層中輸入值會和所謂的神經元的權重(Weight)和偏差值(Bias)分別為 $w_{MD}(l)$ ($M \times D$ 維度) 和 $w_0(l)$ ($M \times 1$ 維度)進行線性組合，並在輸出之前經過激活函數(Activation Function)進行線性或非線性之轉換 $f(\cdot)$ 至下一網路層
- 若網路層越多層會增加運算的複雜度，但能逐漸提高效能



深度神經網路(DNN) (2/5)

- 以下為各種常用的Activation Function，可能因為不同的Activation Function設計的特性而讓最終輸出有不同的結果



深度神經網路(DNN) (3/5)

- 在Feed Forward時，可以得到最後輸出(Output)的一般式表示為(此處式子過長忽略Bias)

$$\hat{y}_j = f^{(L)} \left(\sum_{i=1}^{D_L} w_{ji}^{(L)} \cdot f^{(L-1)} \left(\sum_{i=1}^{D_{L-1}} w_{ji}^{(L-1)} \cdot f^{(L-2)} \left(\sum \dots f^{(1)} \left(\sum_{i=1}^{D_1} w_{ji}^{(1)} \cdot x_i \right) \right) \right) \right)$$

- 得到預測的輸出之後將與真實Label比較進行判斷，並根據Loss Function評斷深度學習網路的效能，基礎的Loss Function包含兩種：
 - 均方誤差(Mean Square Error) 主要為連續值(Continuous)的效能判斷，如距離誤差：

$$E_{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2$$

- 交叉熵(Cross Entropy)為離散(Discrete)的效能判斷，如分類精準度：

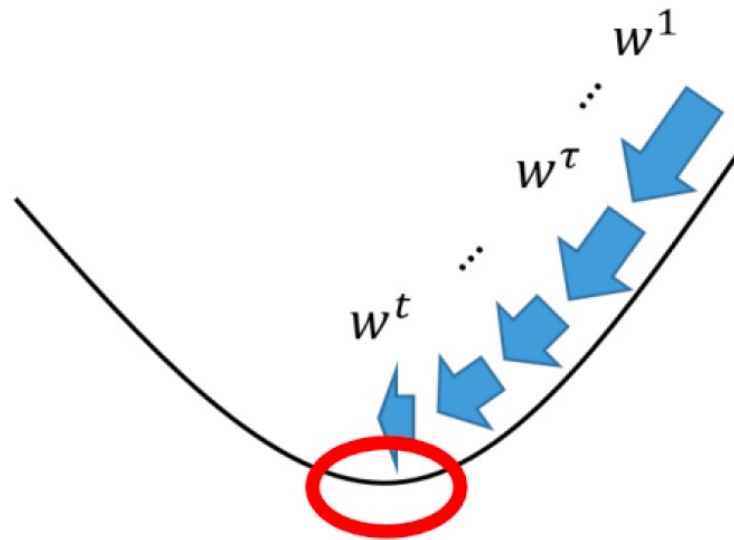
$$E_{CE}(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{i=1}^{D_{out}} y_i \log \hat{y}_i$$



深度神經網路(DNN) (4/5)

Mobile Intelligent Network Technology (MINT) Lab

- 接下來為神經網路參數Weight和Bias的更新，透過Back propagation來進行運算，也就是目標為最小化我們的Loss Function，然後透過疊代的方法更新參數
- 以下介紹簡單的隨機梯度下降法(Stochastic Gradient Descent, SGD)，其主要概念就是參數會朝向負梯度方向前進直到達到最小的Loss



Optimum with minimum loss



深度神經網路(DNN) (5/5)

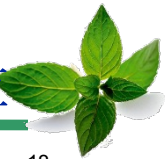
- 當我們每次計算完Loss之後，我們依照下列SGD更新數學式進行疊代更新參數

$$w^{\tau+1} = w^{\tau} - \eta \cdot \nabla E(w^{\tau})$$

- η 為學習率(Learning Rate)參數， $\nabla E(w^{\tau})$ 為Loss Function對所要更新參數之偏微分
- 舉例來說，我們要更新第三層之Weight，那我們就必須透過連鎖律(Chain Rule)對參數進行偏微分，如下式

$$\frac{\partial E}{\partial w_{ji}^{(3)}} = \frac{\partial E_n}{\partial y_j} \frac{\partial y_j}{\partial a_j^{(3)}} \frac{\partial a_j^{(3)}}{\partial w_{ji}^{(3)}}$$

- 依此類推，可得到各層的Weight和Bias之更新值
- 除了傳統SGD的做法，現有廣泛運用的是具有二階特性之Adam優化器(Adam Optimizer)，其效果相對於SGD在大多時候來的更快收斂
- 現有深度學習相關軟體工具已有建立好的函式庫能直接呼叫，並不用人為手動計算代入
 - 當學習網路層數和節點過多時是無法透過手動微分方式代入計算



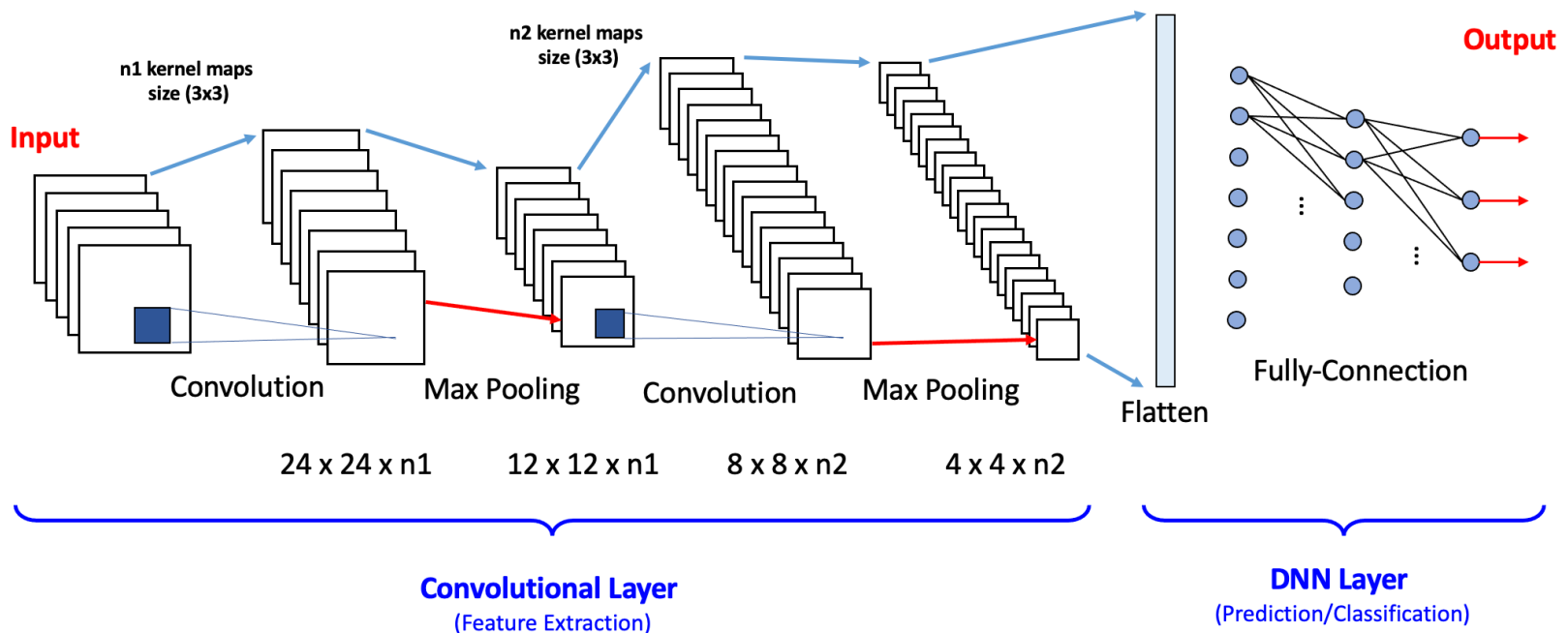


Convolution Neural Networks (CNN)

卷積神經網路(Convolution Neural Network, CNN)

Mobile Intelligent Network Technology (MINT) Lab

- 類似於DNN的網路架構，卷積神經網路CNN採用卷積的方法進行運算，並分成
 - 卷積層(Convolution Layer)
 - 全連接層(Fully-Connected Layer)，亦即DNN層



CNN - Convolution

- 首先介紹Convolution運算，數學運算符號為' \otimes ' 或' * '，運算式表示為

$$y = x \otimes w$$

- x為輸入、y為輸出、w為核心(Kernel)參數，也就是DNN裡面的Weight，w的維度跟DNN一樣可以任意調整其大小深度
- 以下範例簡介CNN的卷積運算，Input資料會和Kernel進行元素矩陣乘法(Element-Wise Product)並加總得到單一值，也就是矩陣(m,n)位置的元素值和Kernel (m,n)位置的值相乘，如紅色框框內的元素值與Kernel相乘並相加後得到Convolved Output值為0，而藍色框框經過卷積運算得到1

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Input

\otimes

0	0	1
1	0	0
0	1	1

Kernel

=

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

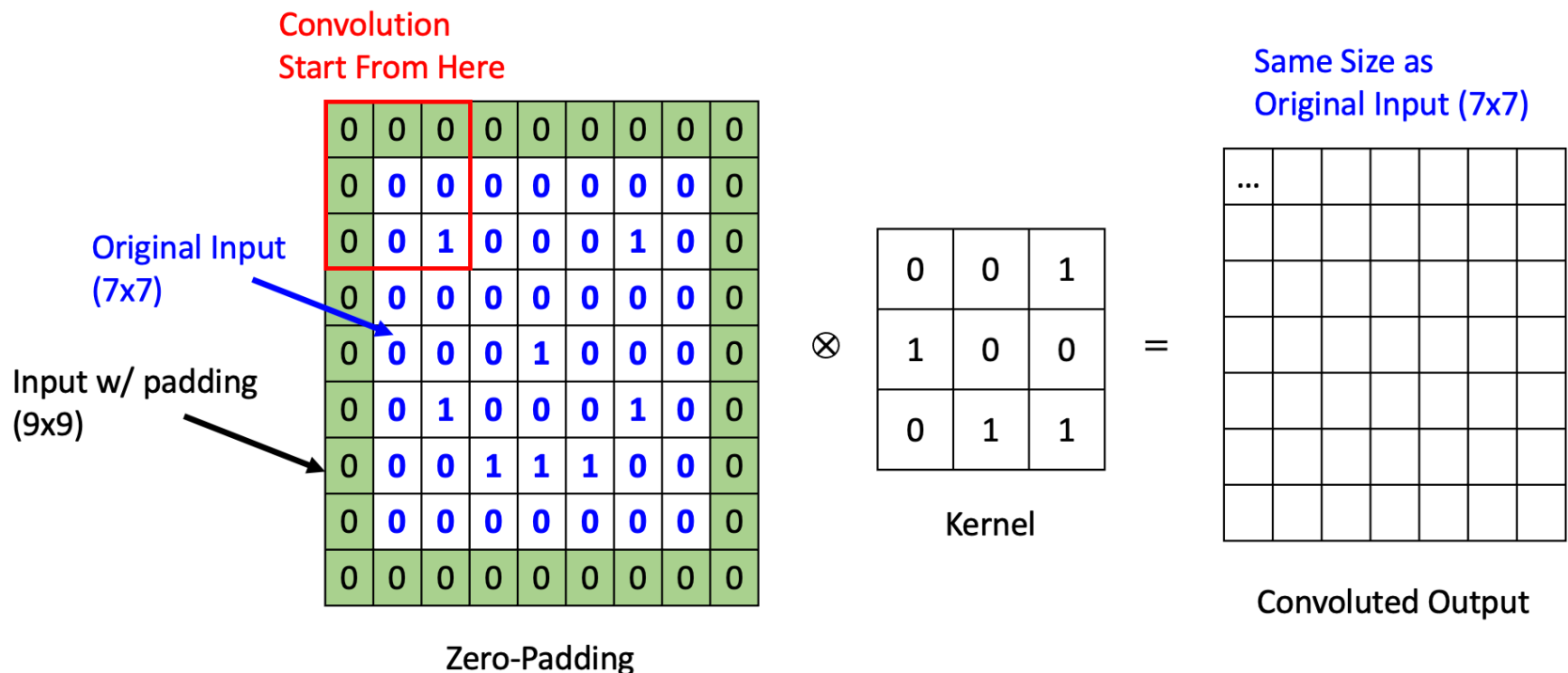
Convolved Output



CNN – Zero Padding and Stride (1/2)

Mobile Intelligent Network Technology (MINT) Lab

- 在執行Convolution時，有時候為了Input和Output大小一致，會採用填充(Padding)和步伐(Stride)的方法
- Padding通常在矩陣的最外面補0，範例如下圖



CNN – Zero Padding and Stride (2/2)

- Stride即Kernel下次移動的步伐大小，範例如下圖

Stride Size = 2

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Input

\otimes

0	0	1
1	0	0
0	1	1

Kernel

=

0	0	0
1	...	

Convolved Output

- 可以依照需求結合兩者產生不同維度(Dimension)的Output，而輸出大小公式為

$$\text{Output Size} = \text{floor} \left(\frac{\text{Original Size} + 2 \times \text{Padding Size} - \text{Kernel Size}}{\text{Stride}} \right) + 1$$



CNN – Max-Pooling

Mobile Intelligent Network Technology (MINT) Lab

- 做完Convolution之後會執行池化(Pooling)運算
 - 好處是當輸入整個平移幾個步伐對判斷上不會造成影響，並具有不錯的抗雜訊功能
 - 我們常用最大化Pooling方法(Max-Pooling)，也就是選擇其中最大的值當作Output
 - 如下圖範例，Output (1,1)的值為原本2x2子矩陣中最大的值
 - 然而Pooling也可以選擇不同大小，一般來說會選擇2x2以不失去太多原本卷積結果特徵的維度
 - 而根據需求，不一定要做Pooling的運算(Optional)，比方說在維度過小時執行Pooling有可能無法擷取重要特徵出來而導致最終預測的失誤

Max-Pooling

0	1	0	0
0	1	1	1
1	0	1	2
1	4	2	1

Convolved
Output



1	1
4	2

Pooling
Output



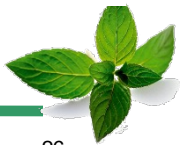
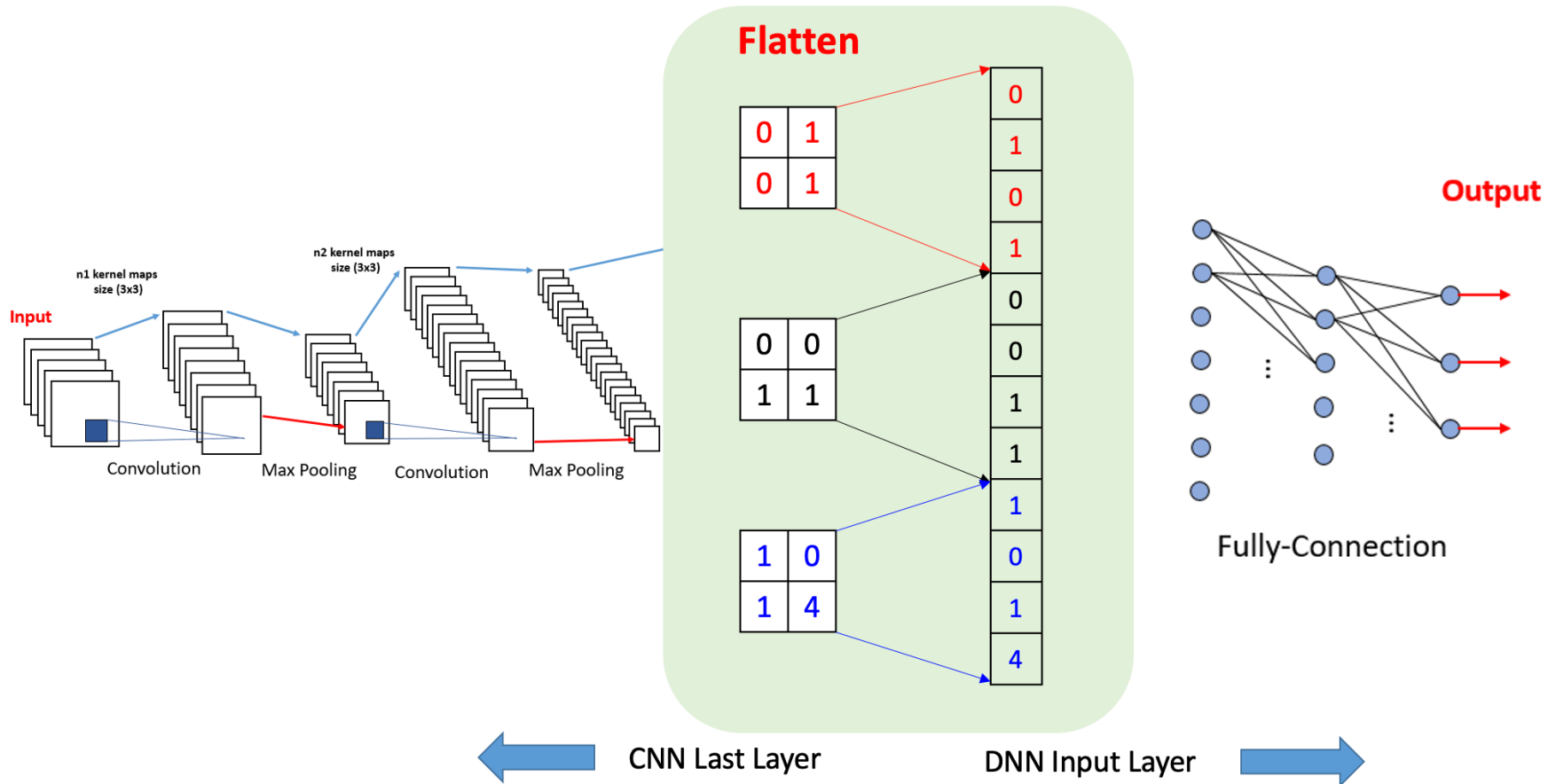
CNN – Flatten (1/2)

- Convolution和Pooling執行完後須要進入Activation Function進行非線性之映射轉換，請參考DNN內容，然後進入下一層CNN直到最終層
- 在最後一層的CNN因為Output是矩陣形式，我們必須在進入DNN前攤平(Flatten) Output變成單一向量形式(Vector)
 - 最後進入DNN層進行運算並預測輸出，可以參考DNN的部分
- 相較於DNN線性向量運算的方法，CNN透過矩陣卷積的運作擷取出具代表性的特徵，保有矩陣鄰近元素的相關性
 - CNN資料也可以透過DNN呈現，然而元素之間的相關性可能較難透過DNN向量形式的運算取得，必須依據資料的特性去設計網路架構和調整內部參數以達所要的輸出效能



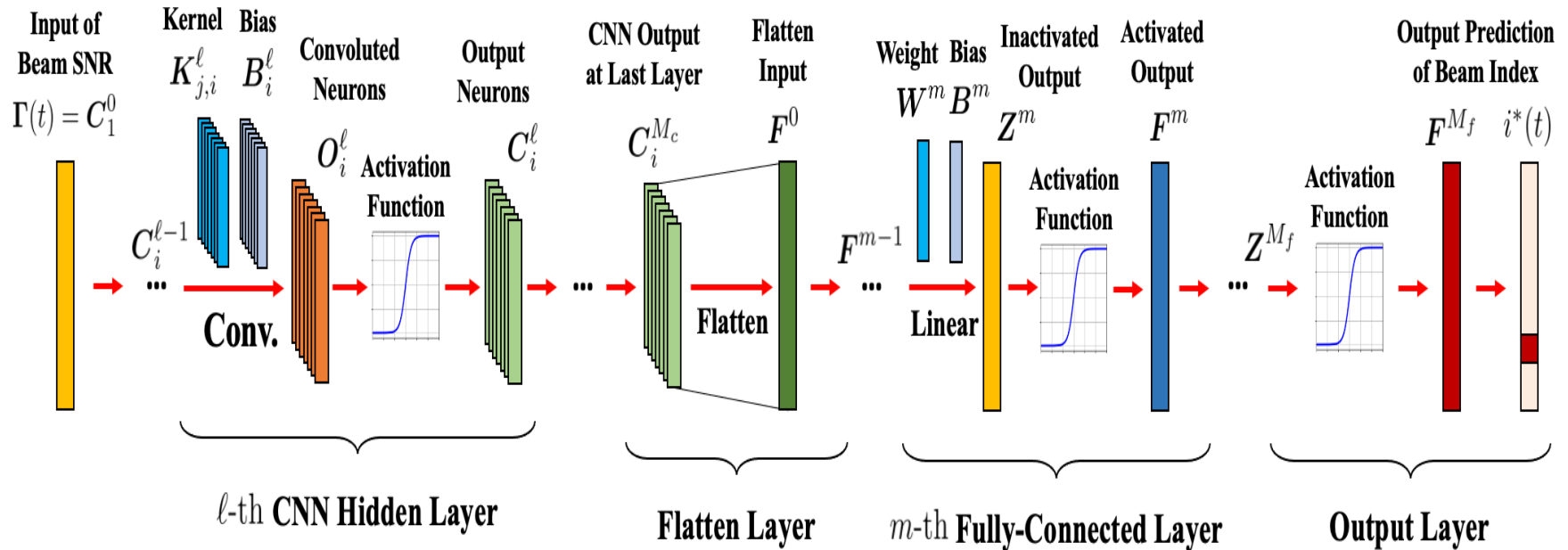
CNN – Flatten (2/2)

Mobile Intelligent Network Technology (MINT) Lab



CNN – Complete Example

Mobile Intelligent Network Technology (MINT) Lab



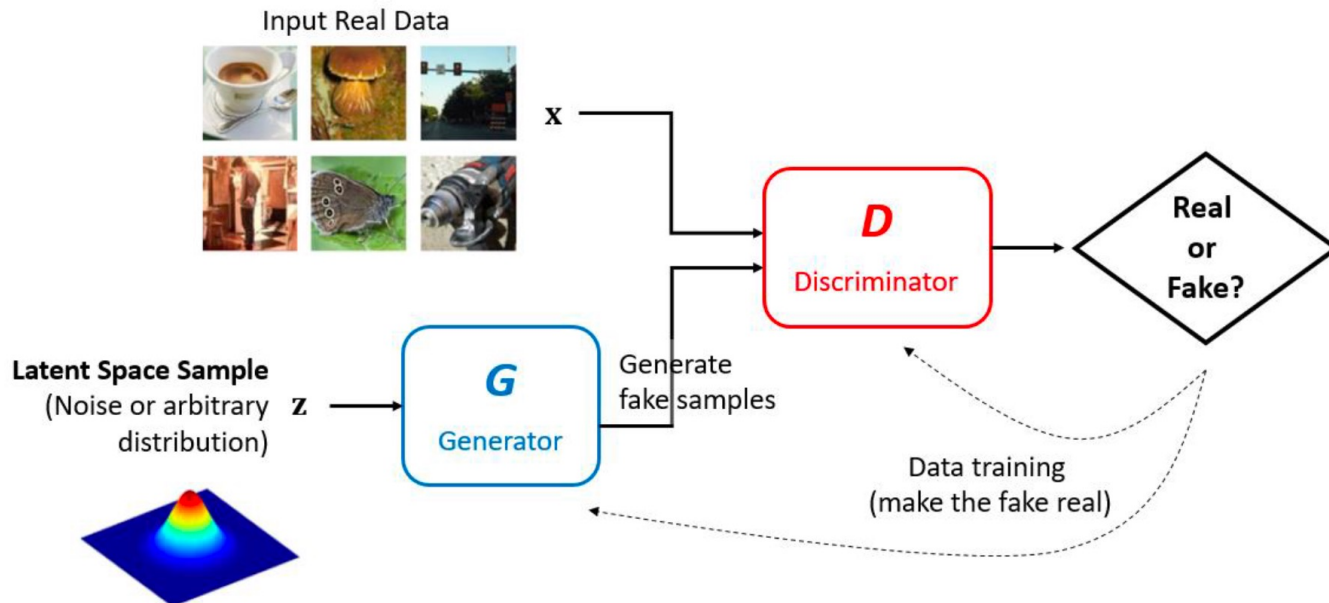


Generative Adversarial Networks (CNN)

對抗生成網路(Generative Adversarial Network, GAN) (1/2)

Mobile Intelligent Network Technology (MINT) Lab

- 對抗生成網路GAN包含
 - 產生器(Generator)網路G： Generator能根據任意的分佈(Distribution)產生出很像真實資料的假資料(Fake Sample)
 - 分辨器(Discriminator)網路D： 能從Generator所產生出來的假資料去分辨真偽性
- 這種訓練方式就是像是兩個玩家參與彼此對抗的同時，假玩家要學的像真玩家的遊戲，故稱為對抗生成網路



對抗生成網路(GAN) (2/2)

- 在無線系統中，因為無線通道不易量測導致資料量過少，所以可以透過GAN的方式將少量的真實資料丟入機器學習中，使得產生的假資料能夠像真實的無線通道資訊，進而降低量測無線通道上的人力成本
- GAN的目標數學式為最大化分辨器的分辨能力，同時最小化產生器產生假資料的誤差

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

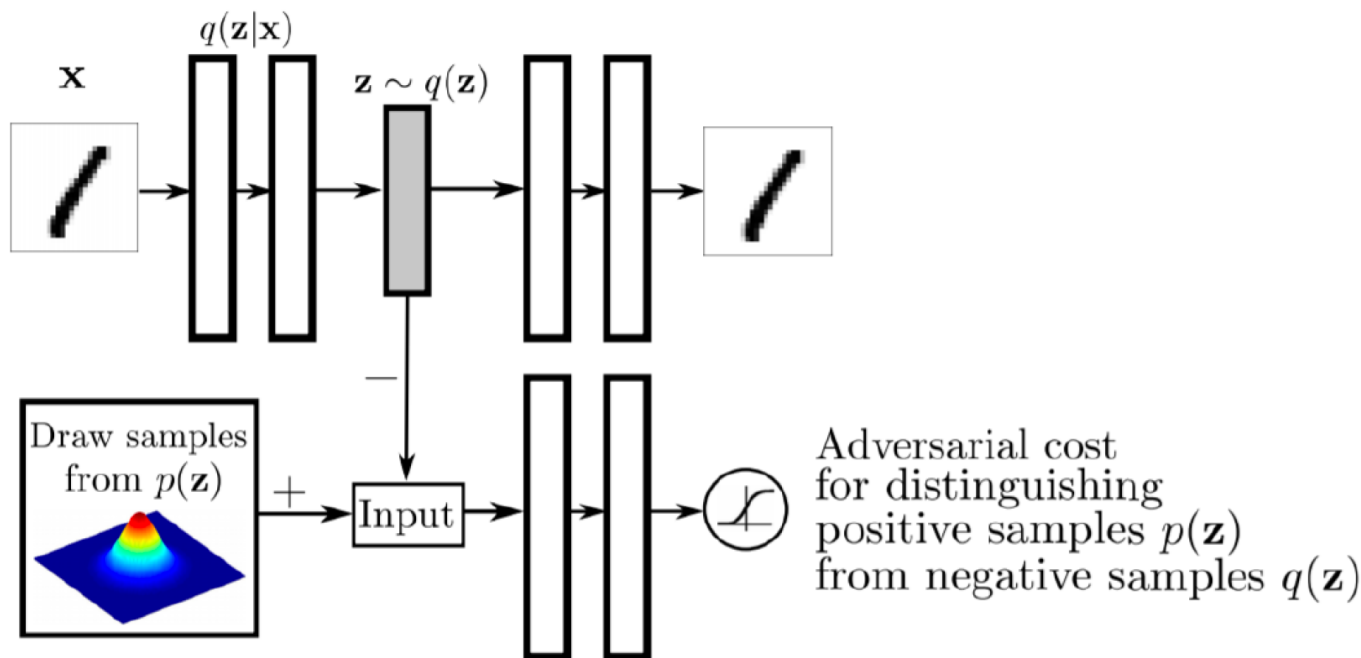
- \mathbf{x} 為真實資料
- \mathbf{z} 為任意分佈的假資料
- D 和 G 網路可以使用前面所介紹的DNN或CNN串接深層網路
- 在執行訓練的時候，必須先固定某一網路去訓練另一個網路
 - 比方說先固定 D 去訓練 G 的參數，再反過來輪流訓練直到收斂為止
 - 所以GAN在一般訓練的時候，因為具有Max-Min對抗關係，收斂時間比起單一DNN或CNN網路來的久，且每次收斂結果有可能不太一樣，需要嘗試多次才能穩定



對抗自編碼器(Adversarial Autoencoders, AAE) (1/2)

Mobile Intelligent Network Technology (MINT) Lab

- GAN有另一種常使用的變形，稱為對抗自編碼器(Adversarial Autoencoders, AAE)或稱為Autoencoder-GAN (AE-GAN)
- AAE概念是要透過深層隱含參數(Latent Variable)辨別真假之外，同時要利用自編碼器學習用低維度但具特徵代表的Latent Variable去產生與原本相像的資料，如此一來，AAE就能透過隨機分佈的假資料去產生所需要真資料。



對抗自編碼器(AAE) (2/2)

- AAE有三個Loss Functions：
 - Autoencoder重建資料的相像性，採用MSE當作Loss Function

$$\mathcal{L}_{\text{recons}} = \sum_{n=1}^N \| \mathbf{x}_n - \hat{\mathbf{x}} \|^2$$

- 訓練GAN分辨器去從假資料辨別真實資料的Loss，即GAN的Max-Min對抗中Max的部分

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log D(\mathbf{z})] - \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log(1 - D(\text{Enc}(\mathbf{x})))]$$

- 訓練編碼器(視為產生器)以讓其能騙過分辨器以為是隨機分布的假資料，即GAN中Min的部分

$$\mathcal{L}_{\text{Enc}} = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log(1 - D(\text{Enc}(\mathbf{x})))]$$





Related Applications

相關應用介紹(Applications) (1/2)

Mobile Intelligent Network Technology (MINT) Lab

- 隨著科技的進步，室內無線偵測(Indoor Wireless Sensing)的應用越來越廣泛，如：
 - 被照護者無需配戴裝置，並透過在環境感應裝置(Sensor)來定位並偵測其行為及睡眠呼吸等生理資訊、利用智慧家庭之感測器協助監測觀察家中的現況等
- 基於人員偵測的服務廣泛應用於室內，如醫療照護、室內人員控管、室內入侵者警報、和商場客群行銷、資產定位及人流分析等，藉由巨量物聯網(Internet of Things, IoT)發展的普及，服務將備受重視，因此發展精準室內無線偵測日趨重要
- 在複雜的室內遮蔽環境下，隨時間改變之物體和人員眾多，使用攝影機及紅外線感測會產生影像訊號死角、高成本建置及個人隱私等皆為室內人員偵測的挑戰
 - 此外有別於穿戴式裝置偵測系統，電池續航力、攜帶性及設備成本等問題都能透過非接觸裝置式(Device-Free)無線偵測系統來解決



相關應用介紹(Applications) (2/2)

Mobile Intelligent Network Technology (MINT) Lab

- 然而大數據時代來臨，各種爆炸量的資訊之大數據包含人類形態、數量及行為對於瞬時通道的變化無法利用傳統優化技術進行無線偵測
- 基於機器學習(Machine Learning)之蓬勃發展，能在難以建模的問題有突破，在各式各樣的服務中，透過機器學習處理複雜的室內無線偵測訊號，應用於人員行為及生理特徵偵測，將提供各種智慧型產業快速蓬勃發展，並同時滿足未來眾多新興需求
 - 在眾多基於人員的服務中隱含許多商機：智慧家庭入侵保全系統、空間佔用偵測、商場人員數偵測下進行廣告推播、人員於工廠及險峻環境之探勘行為的監控系統等
 - 而智慧醫院老人醫療照護和智能呼吸偵測系統，可以讓我們了解睡眠品質與週期，並根據偵測數據改善日常生活品質





References

References

1. <https://zh.wikipedia.org/wiki/%E7%84%A1%E7%B7%9A%E9%80%9A%E8%A8%A>.
2. Fundamentals of Wireless Communication, D. Tse, Cambridge, 2005
3. Wireless Communications, by A. Goldsmith, Cambridge, 2005.
4. Diederik P. Kingma et al, “Adam: A Method for Stochastic Optimization,” in Proc. ICLR, 2015.
5. Ian Goodfellow et al, “Deep Learning,” MIT Press, 2016.
6. Rachel Howard, “Pattern Recognition and Machine Learning,” Springer-Verlag New York, 2006.
7. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley et al, “Generative Adversarial Nets,” arXiv preprint arXiv:1406.2661v1, 2014.
8. Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow, “Adversarial autoencoders,” arXiv preprint arXiv:1511.05644, 2015.

