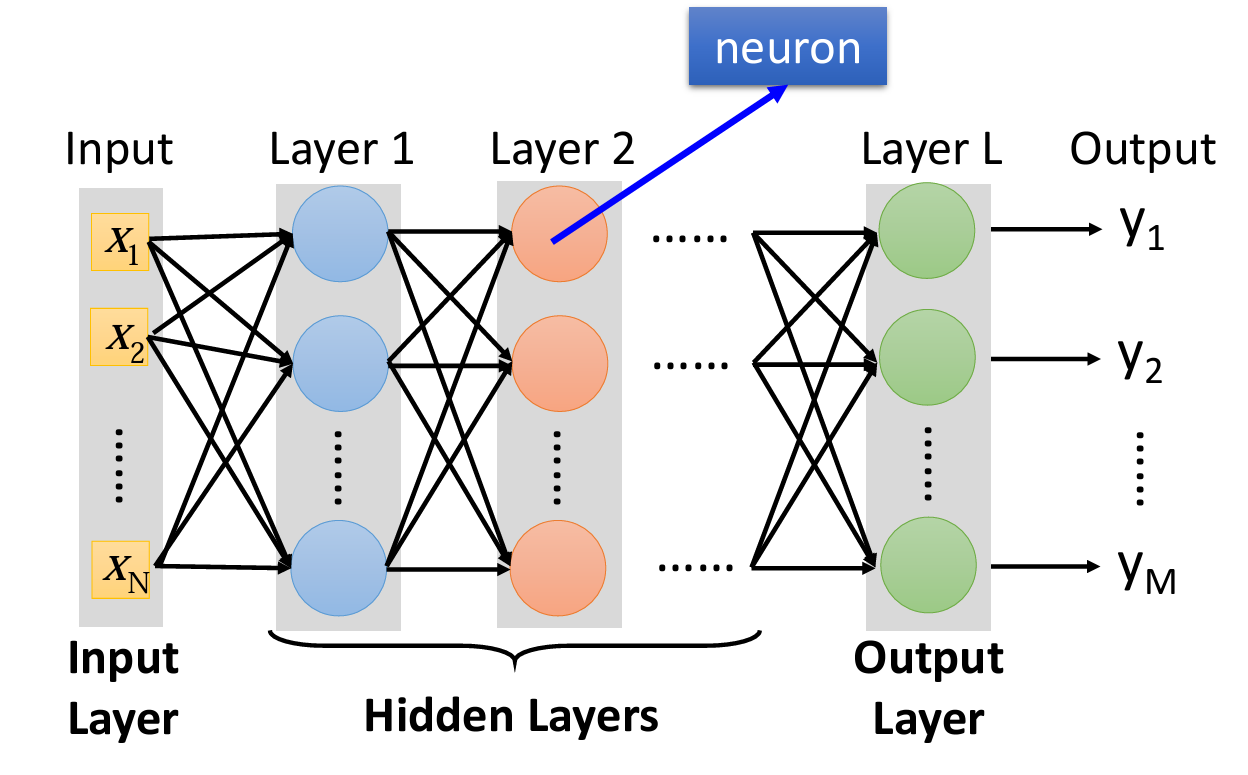
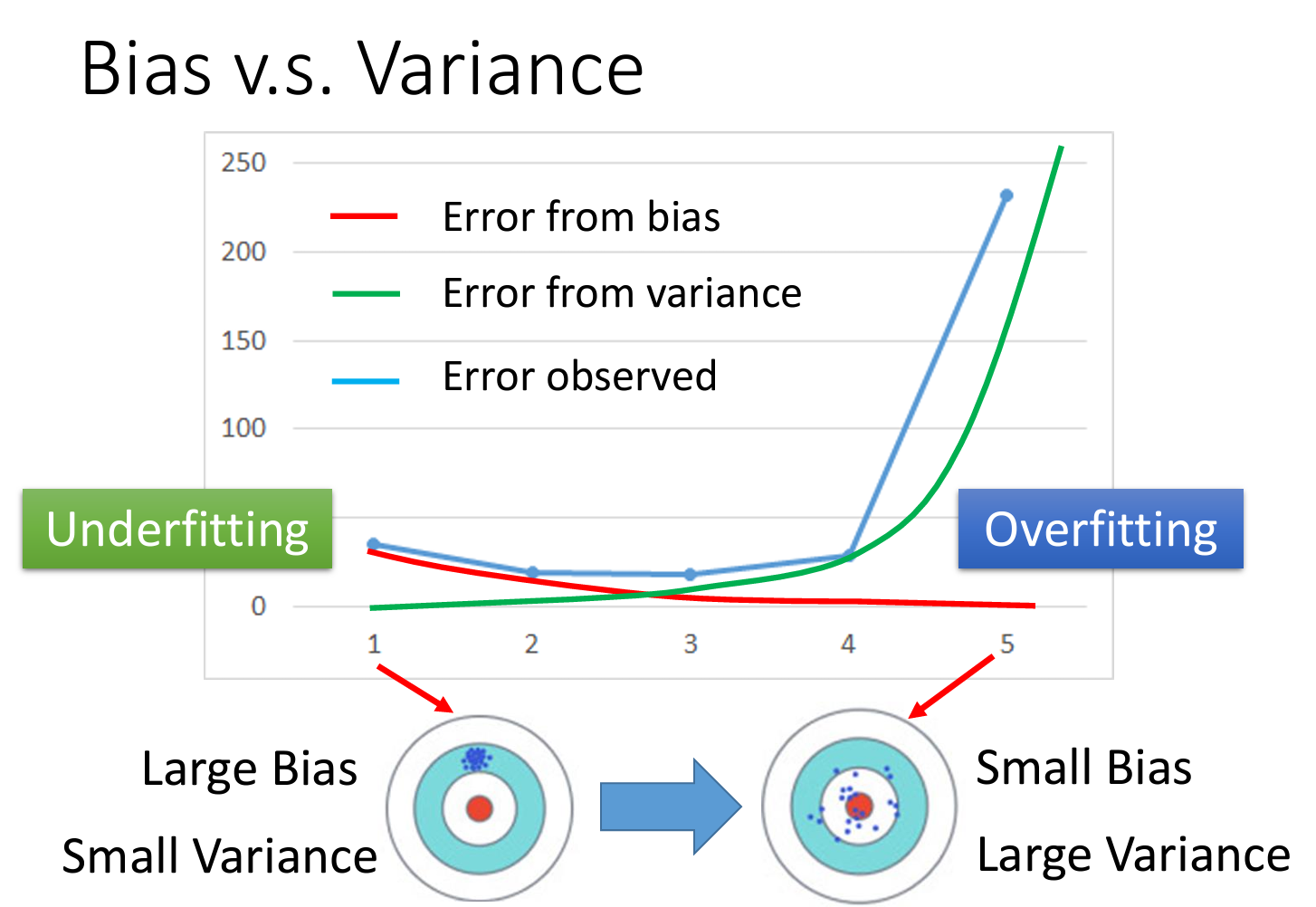
**⦾ML三步驟**

1.Function with Unknow Parameters (NN)

2.Define Loss from Training Data

3.Optimization (本課程以Gradient Descent為例)





**Underfitting：**修改model

**Overfitting：**增加訓練資練，Regularization

**⦾模型Train不起來解法**

•Gradient為0表在critical point,可為local minimum或saddle point

•Gradient Descent + Momentum 會依據Gradient的反方向以及之前的移動方向(weighted sum of the previous gradients)決定下一步，減少卡在local minimum的機率

•在Gradient Descent下，loss下降不了多半不是卡在local minimum或saddle point。

**Adapting Learning Rate :**

RMS(Root Mean Square)根據坡度調整learning rate，坡度大->調小，反之亦然。

RMSProp：可動態調整

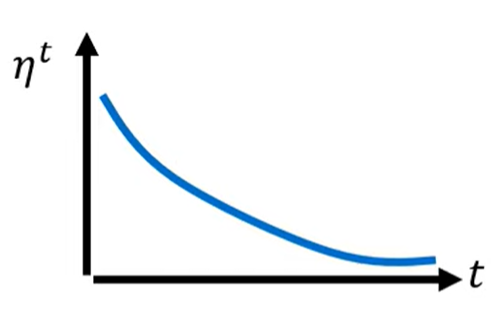
Adagrad : RMS

Adam : RMSProp + Momentum

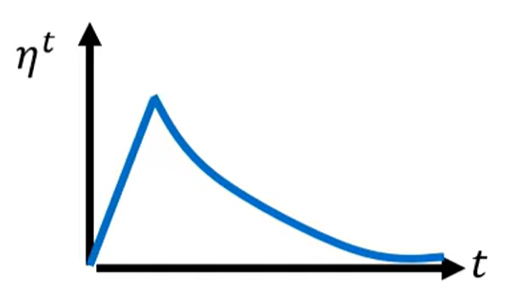
**Learning Rate Scheduling :** 隨著時間跟更新η

實作方法

1. **Learning Rate Decay**

****

1. **Warm Up**

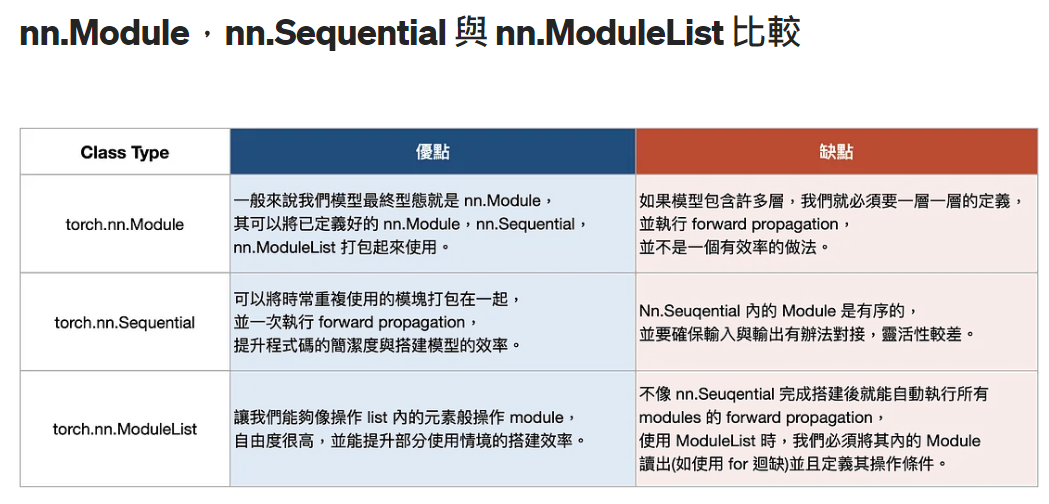
****

**Feature Normalization :**

將資料控制在範圍內如0~1`,使gradient較快收斂

**Batch Normalization :**

為了避免所有資料互相關聯導致GPU不夠，所以一次做一個Batch的資料



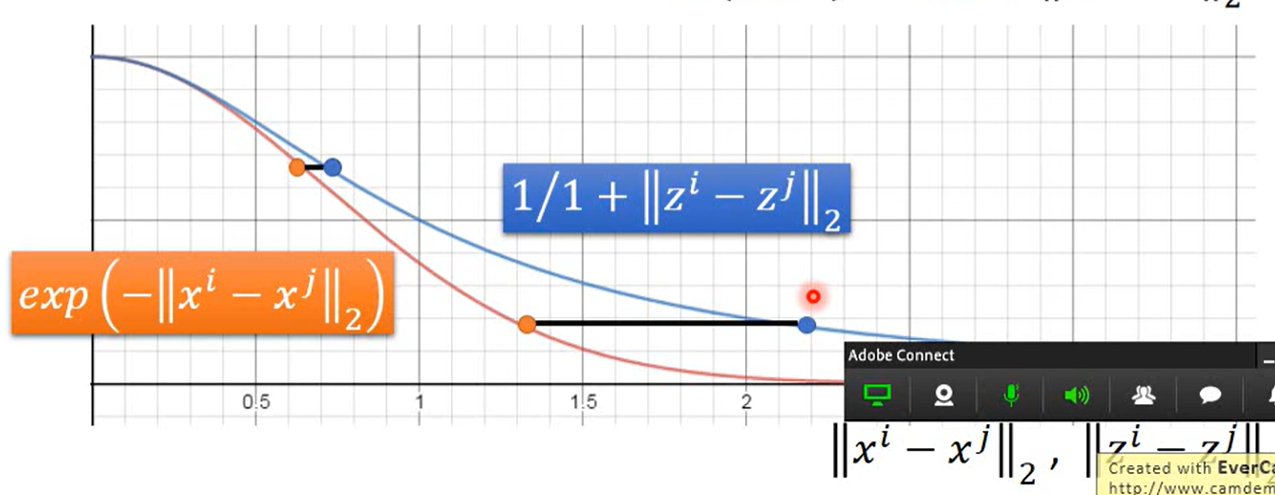
**Dropout：**是一個有效降低過擬合的方法，與權重衰減的精神相似，皆是在訓練模型時，不過度依賴某些權重，進而達到正則化的效果

**⦾CNN**

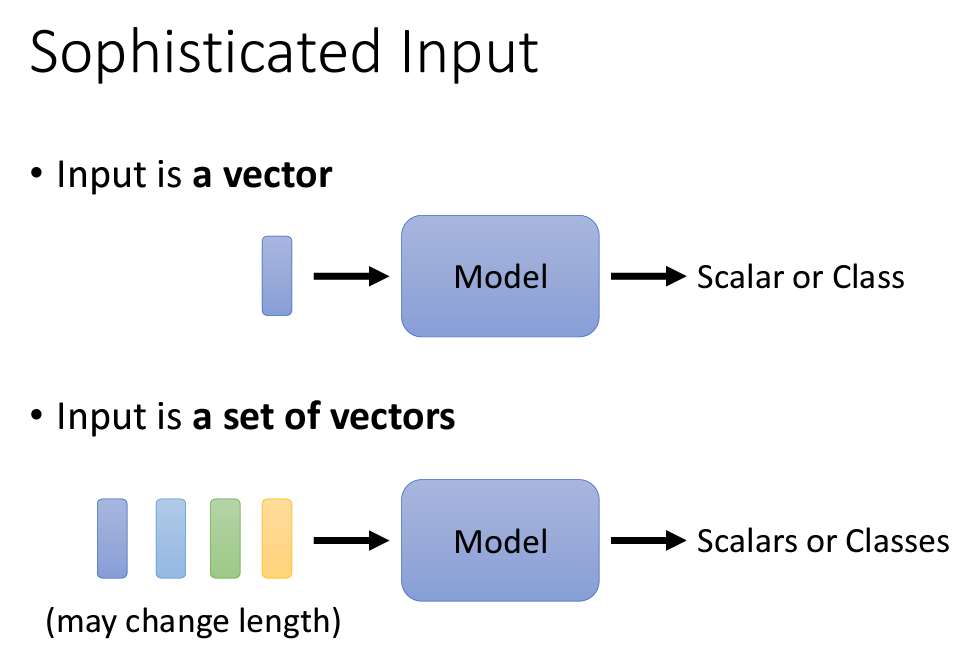
**Convolution：**filter 和對應區塊作內積，要設定filter size, stride, padding

**Pooling：**提取圖片部分特徵，用來縮小data size，減輕計算量，但有些task不建議使用，會嚴重影響準確率

**⦾t-SNE**

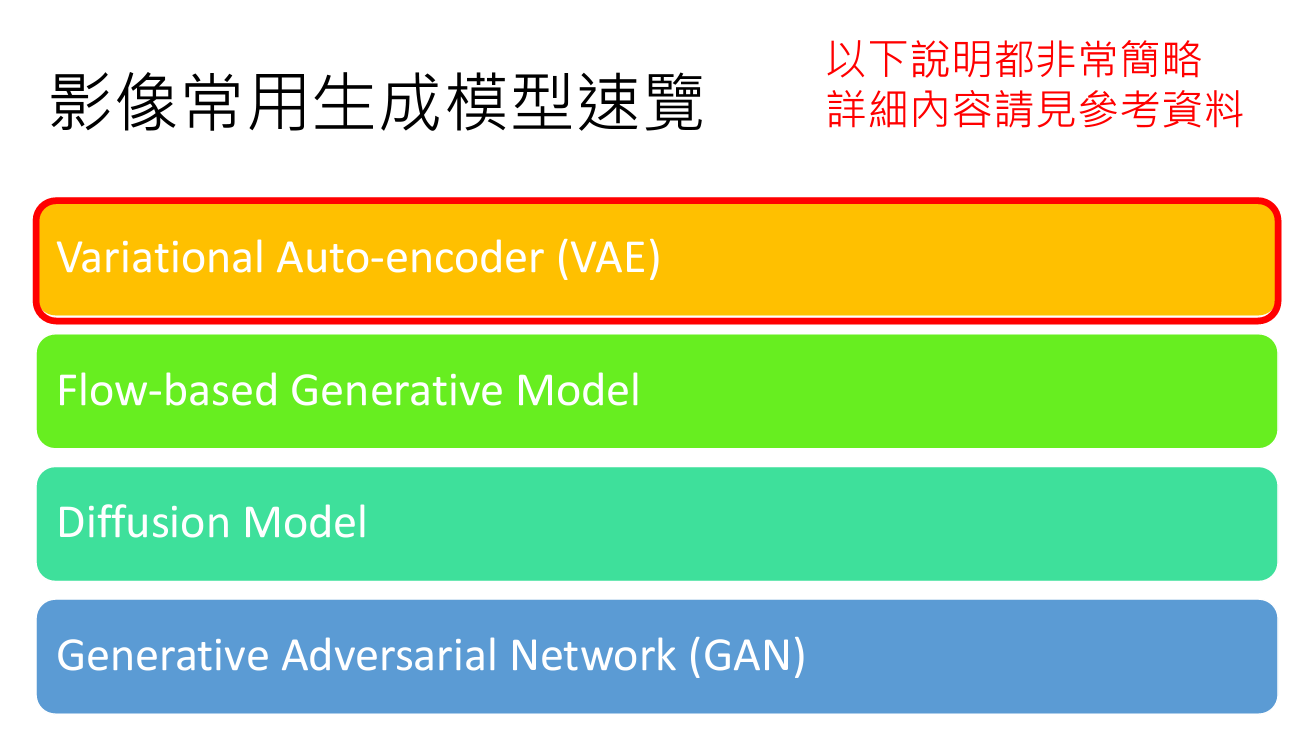


**⦾self-attention**

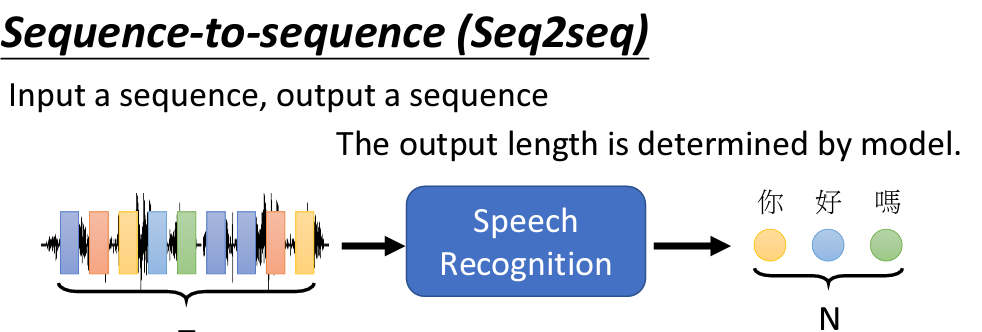


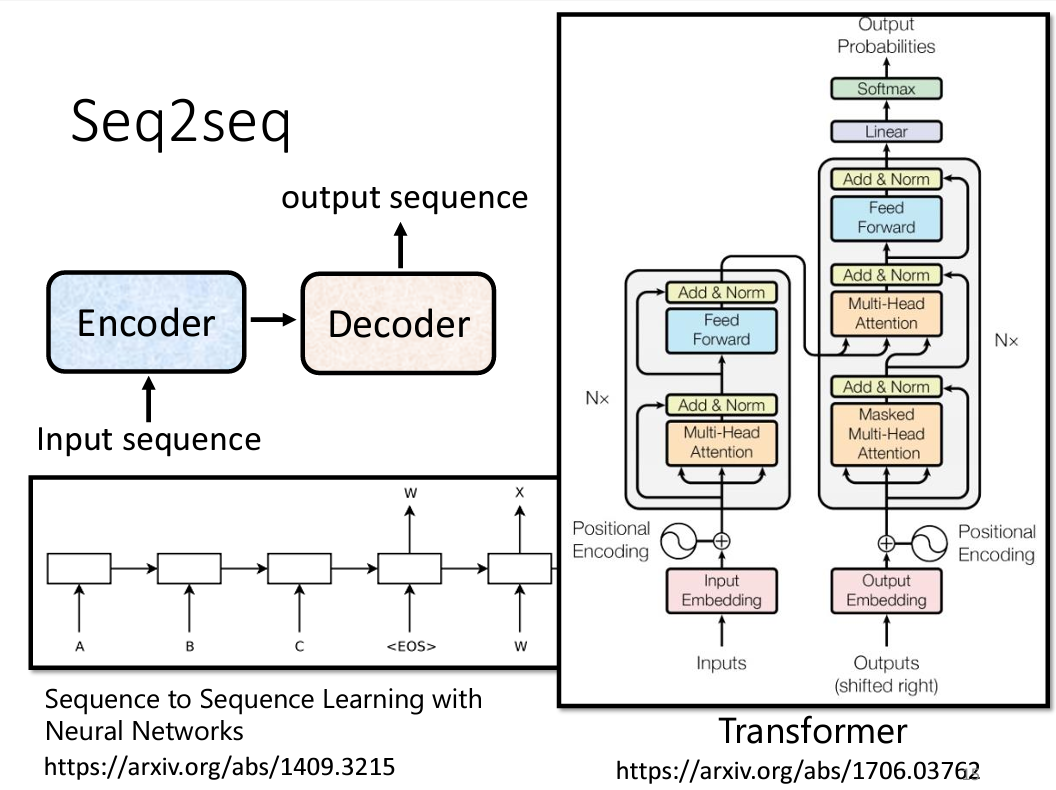
Receptive filed為整個network

**⦾**



**⦾Transformer**

****

****