AIoT Lecture 11 CNN Pytorch

▼ 0. 前言

1. Google Meet [會議google Meet https://meet.google.com/qjv-fvrx-rka
2. 請至ilearning 下載 Lecture 11 講義

▼ 1. CNN review

1. **Parameter Sharing**
2. **Sparsity of connections**
3. **Invariance of object shift**
4. Assumption
   a. Low Level feature are local
   b. Features are translational invariant
   c. High level feature are made up by low level features

▼ [Reference Tommy Huang on mdeium]

- NN-2–1 卷積神經網路(Convolutional neural network, CNN) — 卷積運算、池化運算
- NN-2–2卷積神經網路(Convolutional neural network, CNN) — CNN運算流程
- NN-2–3卷積神經網路(Convolutional neural network, CNN):卷積計算的倒傳遞推導與稀疏矩陣觀點來看卷積計算
- NN-2–4卷積神經網路(Convolutional neural network, CNN):卷積計算中的步伐(stride)和填充(padding)
- NN-2–5卷積神經網路(Convolutional neural network, CNN): 1×1卷積計算在做什麼

▼ 2. 基礎CNN Training (forward and backward)

1. Initialize random weights
2. Forward path ⇒ propagate images through the entire network
3. Calculate loss
4. Backward propagation to tune weightings(gradient descent)
5. More images input and more iterations

▼ 3. 基礎CNN 的 Back Propagation

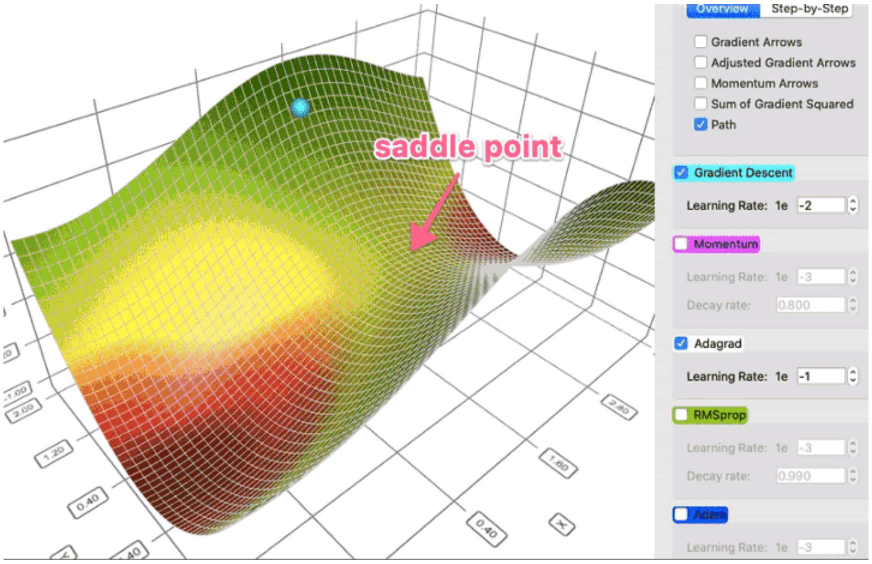
1. **Gradients Decent**

$$w \leftarrow w - \alpha \cdot \frac{\partial J}{\partial w}$$

   - Native **Gradient Decent**
   - **Stochastic Gradient Decent (S**GD, introduce fluctuation),
     - need to choose the correct learning rate (momentum + Nesterov's acceleration corrective updates)
   - **Mini-batch gradient decent** (faster)
   - Adagrad -good for sparse data
   - Adadelta - monotonically decreasing learning rtate
   - **Adam** - Adaptive Moment estimation
   - RMSProp
   - AdaMax, Nadam and AMSGrad

2. **A Visual Explanation of Gradient Descent Methods (Momentum, AdaGrad, RMSProp, Adam)**

   https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c



Overview | Step-by-Step

□ Gradient Arrows
□ Adjusted Gradient Arrows
□ Momentum Arrows
□ Sum of Gradient Squared
☑ Path

☑ Gradient Descent
Learning Rate: 1e -2

□ Momentum
Learning Rate: 1e -3
Decay rate: 0.800

☑ Adagrad
Learning Rate: 1e -1

□ RMSprop
Learning Rate: 1e -3
Decay rate: 0.990

□ Adam
Learning Rate: 1e -3

3. Learning rate
   a. Progressive decreasing
   b. Using learning rate schedules
   c. We use a big learning rate in the early stage and then decrease it slowly.
   d. Deep learning library (Pytorch, Keras 都有 LR scheduler)
4. comparison of deep learning tools Deep-learning software https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

▼ 4. 進階訓練技巧

- **4.1 Dropout**
- 4.2 L1 and L2 regularization

  也就是trade-off (1) prefer smaller weights (2) minimizing the original loss function

  $Loss + \lambda \cdot \Sigma_{i=1}^{p} w_i^2$
- 4.3 Result in less but important weightings present
- 4.4 資料擴增

- 4.5 Early stopping to avoid overfitting- (通常loss stop decrease 之後就會increase)

▼ 4.6 Batch-Normalization (幫助多層次訓練的配合)-reduce internal covariate shift (https://ithelp.ithome.com.tw/articles/10241052)

標準化的好處就是讓收斂速度快一點，不作的話，通常先導向梯度較大的方向前進，造成收斂路線曲折前進，如下圖。

圖一. 不作標準化 vs. 作標準化 優化過程的示意圖，圖片來源：

Why Batch Normalization Matters?

Batch Normalization 另外再引進兩個變數 -- γ(Gamma)、β，分別控制規模縮放(Scale)及偏移(Shift)。

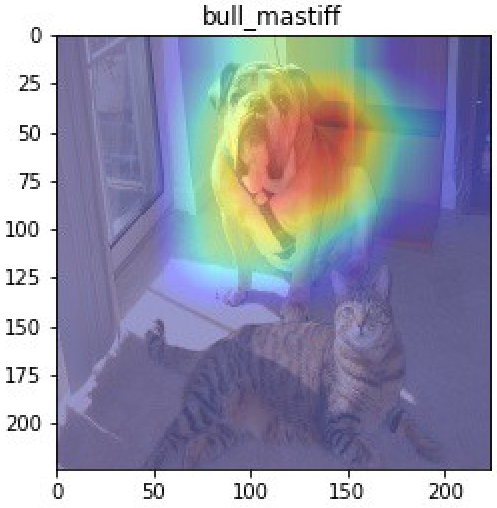
Internal Covariate Shift

假設我們訓練辨識狗的模型，訓練時我麼使用黃狗的圖片作訓練資料集，完成後，我們拿來辨識花狗，這時效果就不好了，必須拿全部資料再訓練一次，這種現象就稱為【Internal Covariate Shift】，正式的定義是【假設我們要使用X預測Y時，當X的分配隨著時間有所變化時，模型就逐漸失效了】。

▼ 5. CNN超炫工具

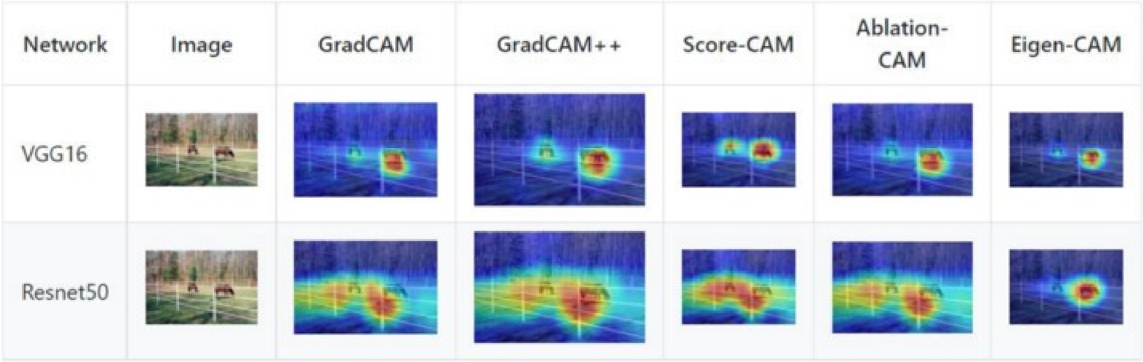
▼ 5.1 Grad-CAM = Keras CAM (Class Activation Mapping)

https://medium.com/手寫筆記/grad-cam-introduction-d0e48eb64adb



▼ 5.2 Grad-CAM Pytorch

https://yanwei-liu.medium.com/pytorch-with-grad-cam-6a92a54bfaad



Network	Image	GradCAM	GradCAM++	Score-CAM	Ablation-CAM	Eigen-CAM
VGG16						
Resnet50						