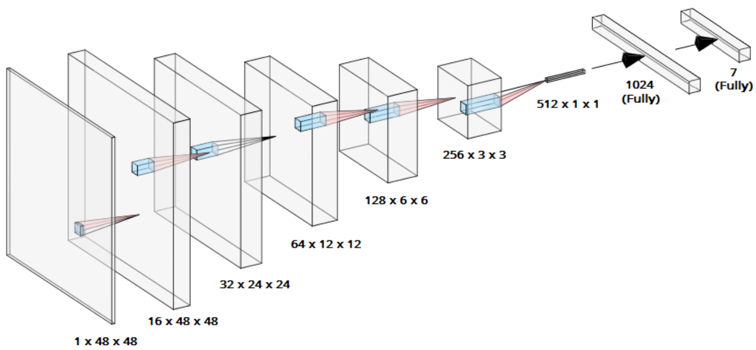
Machine Learning 2019 Fall
HW3 Report

b05902105 余友竹

1. Neural Network Architecture



上圖為我所使用的Model架構，圖中的數字分別代表$(\text{channels}, W, H)$。

架構上可以分為Convolution Layer以及Fully Connected Layer。

Convolution Layer共有6層，每層都包含Conv2d(), BatchNorm2d(), ReLU()。而導致下一層長寬有改變的，則是加上了MaxPool2d()

parameter及解釋:

- Conv2d(): kernel_size = $(3, 3)$, strides = $(1, 1)$。

  除了最後一層$(512 \times 1 \times 1)$外，每一層都有加上padding = $(1, 1)$。
- BatchNorm2d(): 對每層hidden layer做normalization, 可以讓hidden layer的feature分佈相似，加速training收斂，並降低overfitting的風險。
- ReLU(): activation function。這裡有嘗試使用過**Leaky ReLU**(在負的地方有一點斜率), **PReLU**(在learning的過程中同時learn出最好的斜率)，但沒有顯著差異。
- MaxPool2d(): pooling size $= (2, 2)$，stride = $(1, 1)$

疊Convolution Layer的精神有二，一是希望可以將output size控制在500~1000；二是希望圖片的大小僅會透過maxpooling layer改變，方便計算大小。

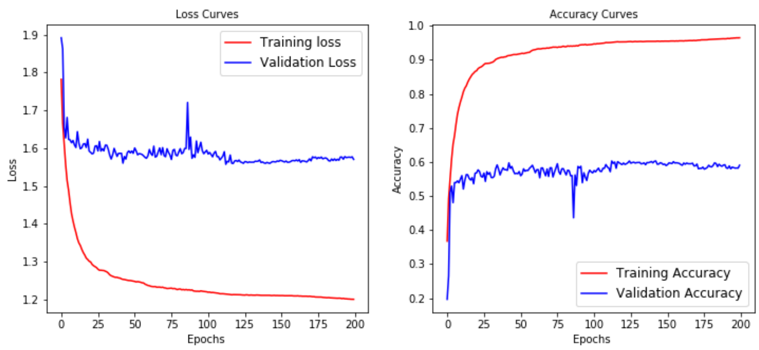
而Fully Connected Layer僅有兩層，第一層維度是1024維，搭配Batch Normalization，並使用ReLU作為activation function；第二層則是7維，直接接上Softmax做最後的分類。

疊Fully Connected Layer的精神則是希望可以適當地控制參數量，彈性調整model的大小。

在這個architecture下，參數數量總共有2109031個，大小算是適中，通過系上工作站的RTX 2080 Ti做training，batch_size設2000，每個Epoch約花8~10秒。

Note: 這次的Task我並沒有加上Dropout Layer，因為Dropout Layer主要目的是防止overfitting。但這次的Task，testing data都保留在training data中，越overfit表現會越好。

2. Training/Validation Loss/Accuracy Curves

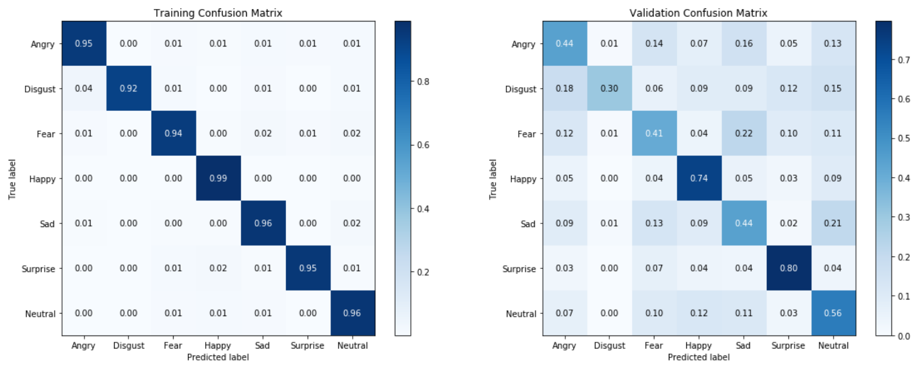


這是我的Loss, Accuracy Curves，Training 的準確度非常高，約在50個epochs以內就提升到90%的正確率了，但一直到200個epochs，Validation Accuracy都只有約60%的正確率。

可以看到，這個model非常的training data導向(biased toward training data)。

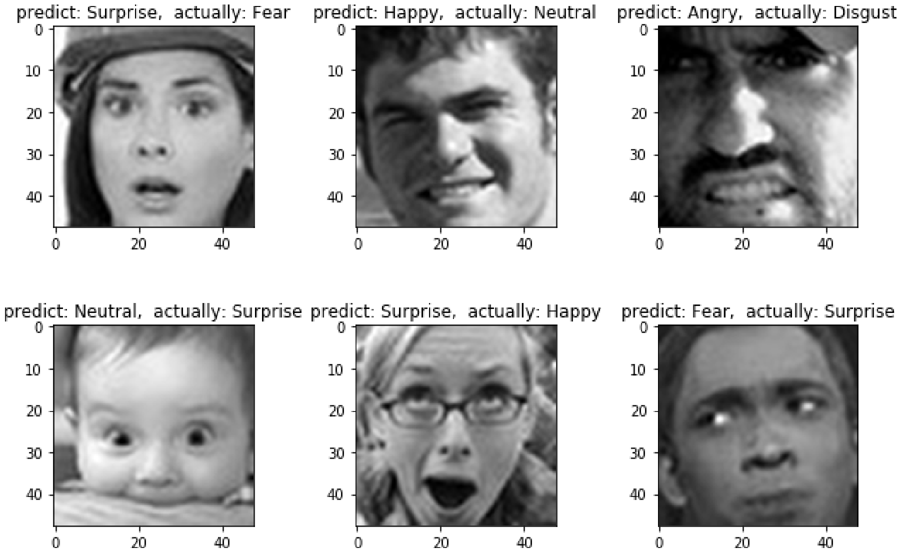
要提高validation accuracy，我想必須加上適當的regularization，如Dropout Layer, L1、L2、Weight Decay Regularization, etc. 但對這次的Kaggle競賽沒有幫助，我就沒有特別使用這些技巧了。

3. Confusion Matrix



右上角為Training Data的Confusion Matrix，可以看到，因為這個Model非常overfit Training Data，所以Confusion Matrix比例幾乎都很集中。

我滿好奇Training Accuracy這麼高的情況下，到底還有哪些表情會被混淆？因此我印出了其中幾張預測錯誤的圖片：



可以看到，這幾張照片Predict的結果還比正確結果更可信，估計是標籤錯誤。

而Validation 的Confusion Matrix被混淆的比例就相對較高了，這邊不一一列出。

可以看到，模型對Happy，以及Suprise的預測結果是最準確的，我認為這是因為Happy, Surprise的特徵最明顯，像是嘴巴微笑、嘴巴張大，這都很容易判斷。

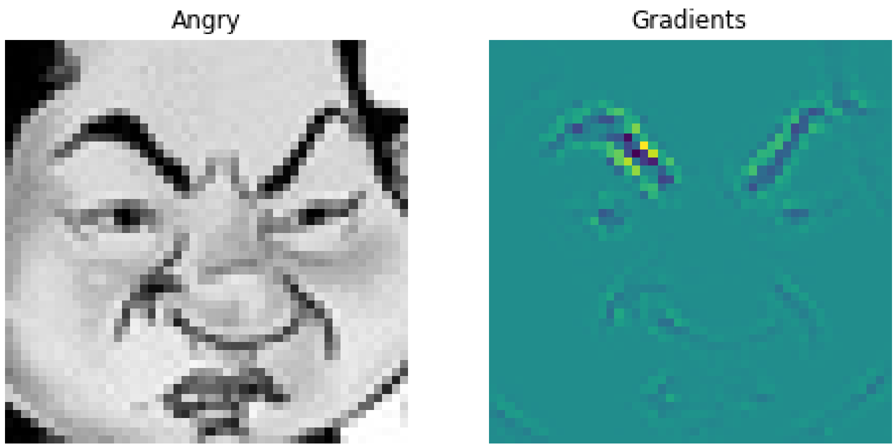
反之，Fear, disgust, neutral就沒那麼好判斷，有幾張照片人類都很難判斷出正確表情。

4. Saliency Map

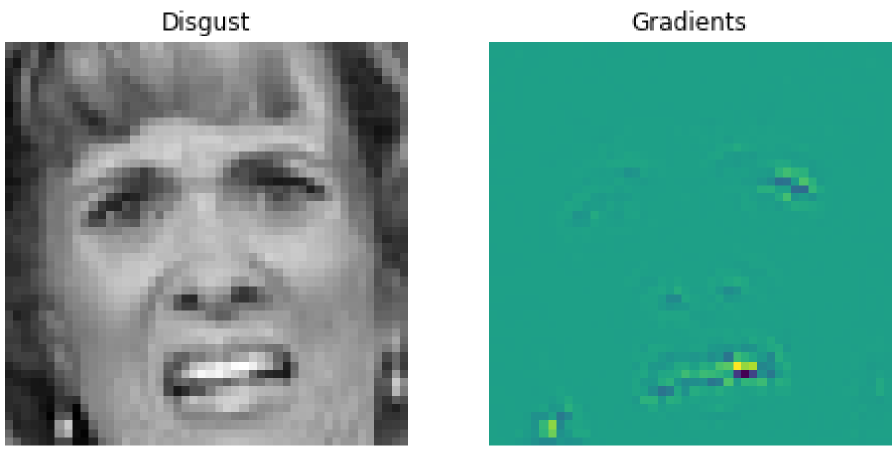
下列分析Neural Network表現出的Saliency Map，Saliency Map主要在觀察圖片每個pixel對CNN Model預測結果的影響力，以了解CNN Model主要Focus在圖片的哪個部分做決策

在這裡我參考了**Striving for Simplicity: The All Convolutional Net**所提到的Guided BackPropagation (實作於 https://github.com/MisaOgura/flashtorch/blob/master/flashtorch/saliency/backprop.py)，大幅提升了Saliency Map的可辨識度。

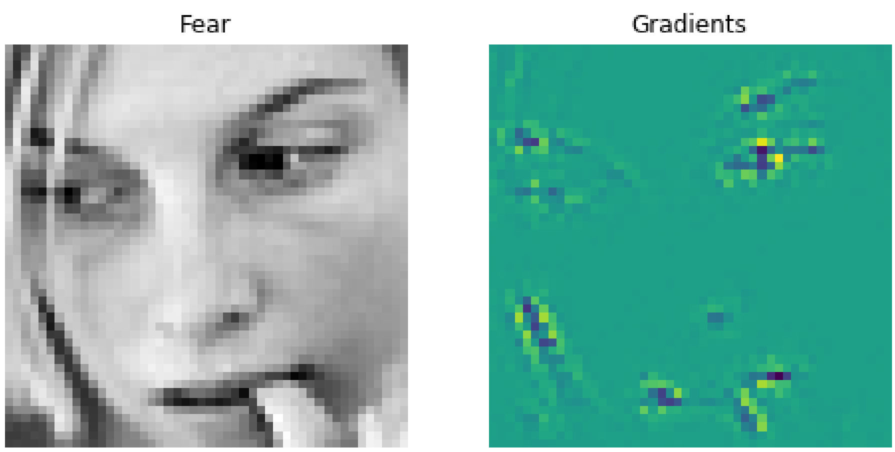
我隨機在每個類別挑出一張最有辨識度的圖片做代表，分別為: Angry(#399), Disgust(#1097), Fear(#1152), Happy(#12467), Sad(#12497), Surprise(#12451), Neutral(#12999)，以下一一說明。



這張照片的眉毛相當搶戲，大概看到眉毛就能判斷是Angry了，實際上Saliency Map在這塊區域的Gradient也是最明顯的。



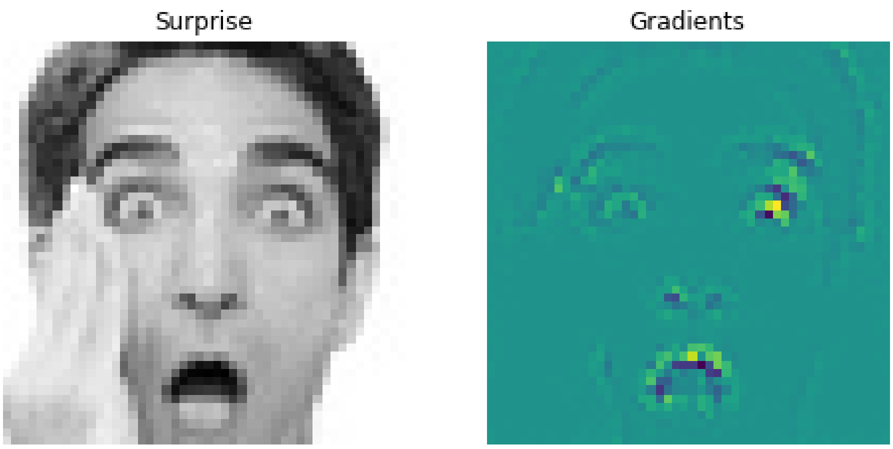
Disgust相對也比較不明顯，Modely focus在張圖的嘴巴上，但我認為眼睛也占了滿重要的一部份。



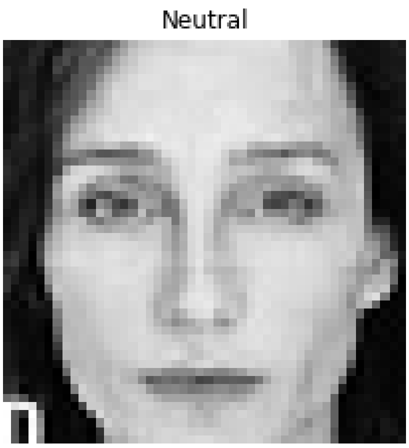
老實說Fear的特徵相當不明顯，我挑了幾張照片的標準都不太一樣，我想這也是模型在預測Fear相當不準確的原因之一。(Validation Set上只有0.41的準確率)

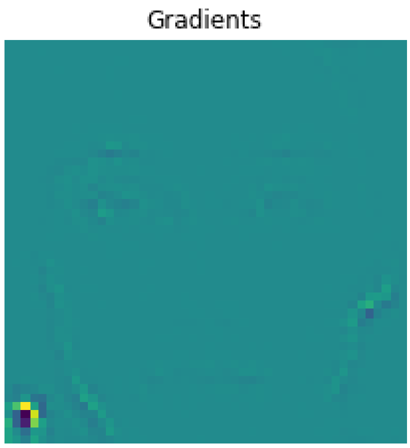
Happy最大的特徵當然是微笑或大笑的嘴巴，多張圖都顯示嘴巴是最Model判斷Happy時最大的依據。

Sad相對來講也沒有那麼明顯的特徵，但這張照片 的眼睛確實可以從Saliency Map標示的區域(眼睛)來判斷。



Surprise的特徵跟Happy很像，在Saliency Map上最常被Focus的就是眼睛跟嘴巴，但他們有本質上的不同，實際上我們根據嘴型，以及眼睛的形狀，也能輕易分出兩者的差異。Happy跟Surprise這兩個Classes也是模型預測最準確的部分，在Validation Set上分別有0.74跟0.80的準確率。





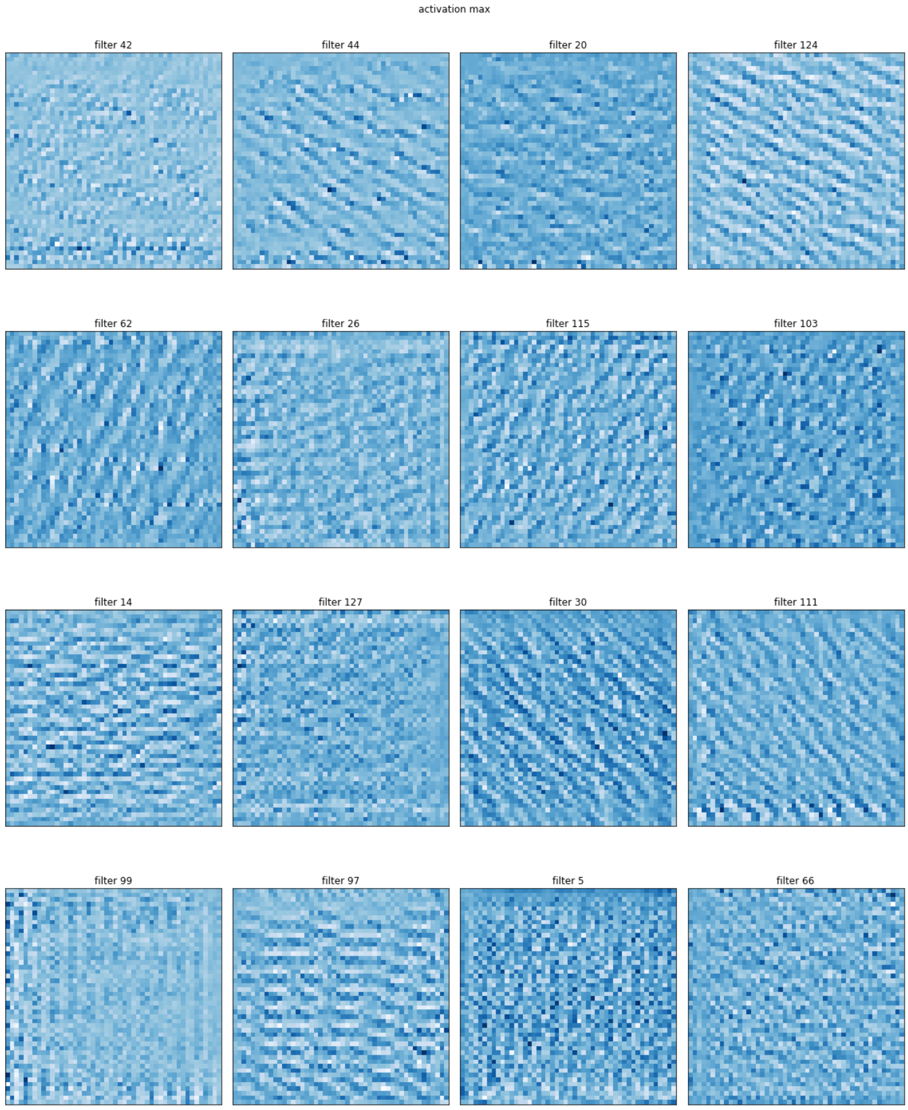
人類判斷時，會將無法歸類為前六者的表情歸類在Neutral，那麼Model是怎麼判斷某個表情是Neutral呢? 這張照片給了一個很有趣的答案──沒有特徵，就是Neutral的特徵。

我找了幾張Neutral的照片，都發現，Saliency Map都沒有明顯突出的區域，忠實地反映出Neutral的特性。

5. Activation Maximization

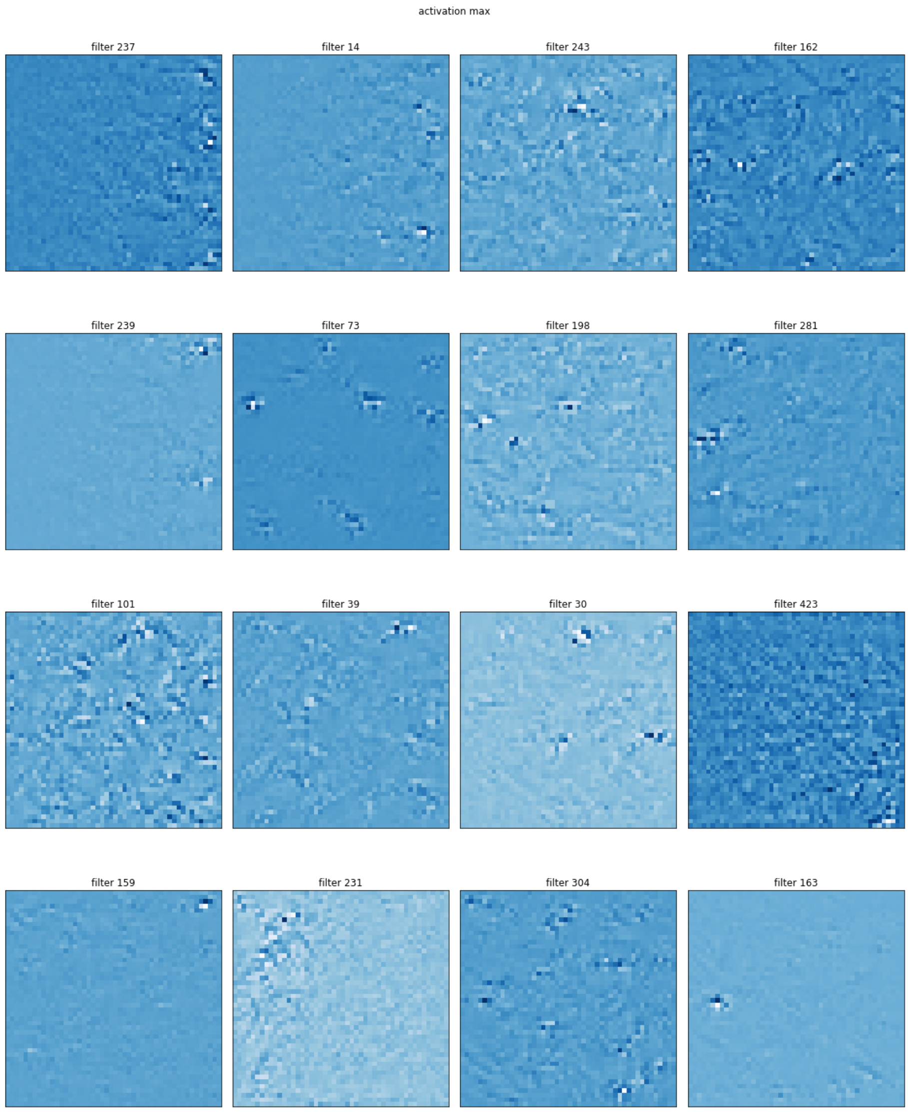
在這邊我測試的Model有5層Convolution Layer，通過100個Epochs的Training後，在Training Set上的Accuracy有91%

下圖為通過第三層Convolution Layer(128個filter)後得到的activation maximization Feature，我擷取了隨機16個Filter。



可以看到，基本上還是一些紋理圖，但如果跟前面幾層的filter做比較(這邊沒有列出來)，就會發現第三層的filter多了許多曲線的紋理，可以想像，越深的layer，可以觀察出越複雜的紋理。

但觀察更深的Layer(這邊我選擇觀察第五層的Convolution Layer，有512個filter)，卻發現有許多filter很難被acitivate。



我認為可能是Regularize制約的關係。舉例，隨著架構變深，feature通過一層層的Layer，每層都會做一次標準化(Batch Normalization)，可能還會有weight decay，到越深的Layer時，差異已經很小，帶給Layer變化也跟著變小，自然很難activate filter。但這純屬推測，實際情形還有待查證。

6. Math Problem

Convolution

假設進入Convolution Layer的size為：

$$(B, W, H, \text{input\_channels})$$

則Output的size為：

$$(B, W_{out}, H_{out}, \text{output\_channels})$$

其中，output_channels即為Convolution中的參數: output_channels

W_{out}跟圖片padding後的寬度，以及水平向stride的距離s_1有關，公式為：

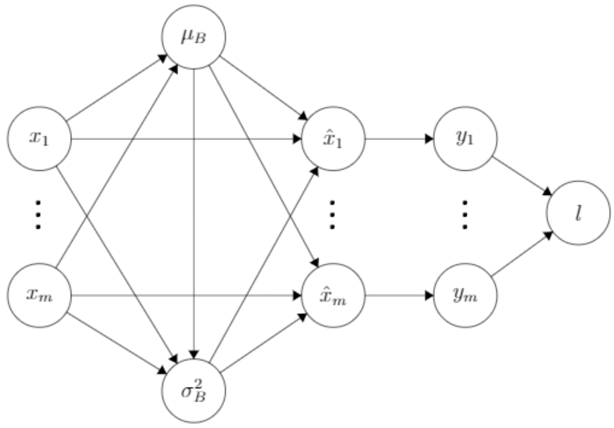
$$W_{out} = \lfloor \frac{W_{in} + 2p_1 - k_1}{s_1} + 1 \rfloor$$

H_{out}跟圖片padding後的高度，以及垂直向stride的距離s_2有關，公式為：

$$H_{out} = \lfloor \frac{H_{in} + 2p_2 - k_2}{s_2} + 1 \rfloor$$

Batch Normalization

下圖為各個參數之間的關係示意圖



1. $\frac{\partial l}{\partial \hat{x}_i}$

$$\frac{\partial \ell}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \frac{\partial y_i}{\partial \hat{x}_i} = \gamma \cdot \frac{\partial \ell}{\partial y_i}$$

2. $\frac{\partial \ell}{\partial \sigma_B^2}$

$$\begin{aligned}\frac{\partial \ell}{\partial \sigma_B^2} &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial \sigma_B^2} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{\partial}{\partial \sigma_B^2}\left(\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}\right) \\ &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{-(x_i - \mu_B) \cdot \frac{1}{2} \cdot (\sigma_B^2 + \epsilon)^{-\frac{1}{2}}}{\sigma_B^2 + \epsilon} \\ &= \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot -\frac{1}{2}(x_i - \mu_B)(\sigma_B^2 + \epsilon)^{-\frac{3}{2}}\end{aligned}$$

3. $\frac{\partial l}{\partial \mu_B}$

$$\begin{aligned}\frac{\partial l}{\partial \mu_B} &= \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial \mu_B} + \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial \mu_B} \\ &= \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\partial}{\partial \mu_B}\left(\frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2\right) + \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot -(\sigma_B^2 + \epsilon)^{-\frac{1}{2}} \\ &= \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{1}{m} \sum_{i=1}^{m} -2(x_i - \mu_B) + \sum_{i=1}^{m} \frac{\partial \ell}{\partial \hat{x}_i} \cdot -(\sigma_B^2 + \epsilon)^{-\frac{1}{2}}\end{aligned}$$

4. $\frac{\partial l}{\partial x_i}$

$$\begin{aligned}\frac{\partial l}{\partial x_i} &= \frac{\partial l}{\partial \sigma_B^2} \frac{\partial \sigma_B^2}{\partial x_i} + \frac{\partial l}{\partial \mu_B} \frac{\partial \mu_B}{\partial x_i} + \frac{\partial l}{\partial \hat{x}_i} \frac{\partial \hat{x}_i}{\partial x_i} \\ &= \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{1}{m} \cdot 2(x_i - \mu_B) + \frac{\partial l}{\partial \mu_B} \cdot \frac{1}{m} + \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \epsilon}}\end{aligned}$$

5. $\dfrac{\partial l}{\partial \gamma}$

$$\frac{\partial l}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \frac{\partial y_i}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \cdot \hat{x}_i$$

6. $\dfrac{\partial l}{\partial \beta}$

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \frac{\partial y_i}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i}$$

Softmax and Cross Entropy

針對第t項的Cross Entropy:

$$L_t(y_t,\ \hat{y}_t) = -(y_t \log \hat{y}_t + (1 - y_t) \log(1 - \hat{y}_t))$$

根據助教解釋，假設題目為binary classification

若$y_t = 1, L(y_t,\ \hat{y}_t) = -y_t \log \hat{y}_t$

$$\begin{aligned}
\frac{\partial L_t}{\partial z_t} &= \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \\
&= \frac{\partial}{\partial \hat{y}_t}(-y_t \log \hat{y}_t) \cdot \frac{\partial}{\partial z_t}(\frac{e^{z_t}}{\sum_i e^{z_i}}) \\
&= -\frac{y_t}{\hat{y}_t} \cdot \frac{e^{z_t}(\sum_i e^{z_i}) - e^{z_t} \cdot e^{z_t}}{(\sum_i e^{z_i})^2} \\
&= -\frac{y_t}{\hat{y}_t} \cdot (\frac{e^{z_t}}{\sum_i e^{z_i}} - \frac{e^{z_t}}{\sum_i e^{z_i}} \cdot \frac{e^{z_t}}{\sum_i e^{z_i}}) \\
&= -\frac{y_t}{\hat{y}_t} \cdot (\hat{y}_t - \hat{y}_t^2) \\
&= -y_t + y_t \hat{y}_t \\
&= -y_t + \hat{y}_t = \hat{y}_t - y_t
\end{aligned}$$

若$y_t = 0, L(y_t,\ \hat{y}_t) = -(1 - y_t) \log(1 - \hat{y}_t)$

$$\begin{aligned}
\frac{\partial L_t}{\partial z_t} &= \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \\
&= \frac{\partial}{\partial \hat{y}_t}(-(1 - y_t) \log(1 - \hat{y}_t)) \cdot (\hat{y}_t - \hat{y}_t^2) \\
&= \frac{1 - y_t}{1 - \hat{y}_t} \cdot (\hat{y}_t - \hat{y}_t^2) \\
&= (1 - y_t) \cdot \hat{y}_t = \hat{y}_t - y_t \cdot \hat{y}_t \\
&= \hat{y}_t = \hat{y}_t - y_t
\end{aligned}$$

可知，$\dfrac{\partial L_t}{\partial z_t} = \hat{y}_t - y_t$