Machine Learning - Homework 3

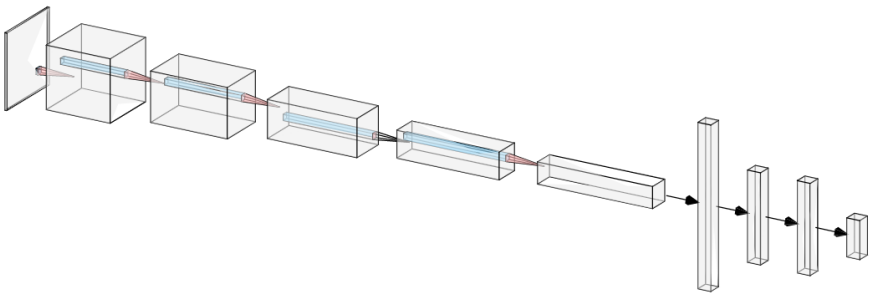
資工四 B05902023 李澤諺

November 11, 2019

Part 1. Programming Problem

1. (1%) 請說明這次使用的 model 架構，包含各層維度及連接方式。

以下爲我於本次作業中所實作的 CNN 架構：

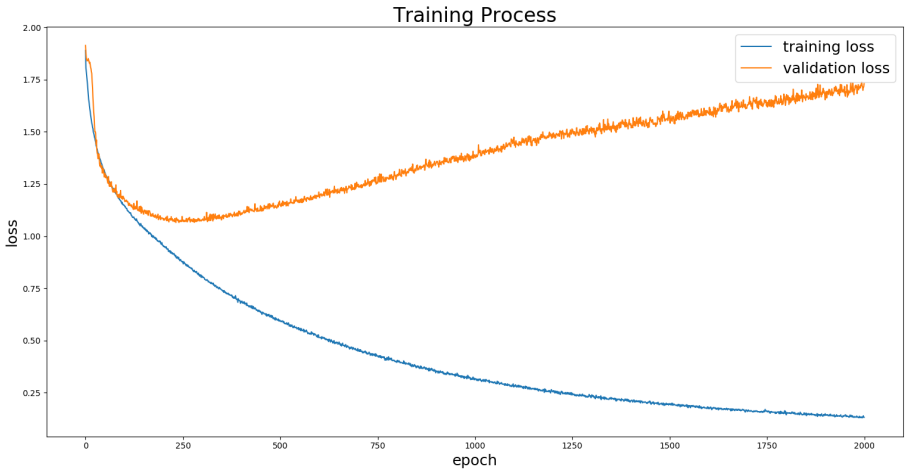


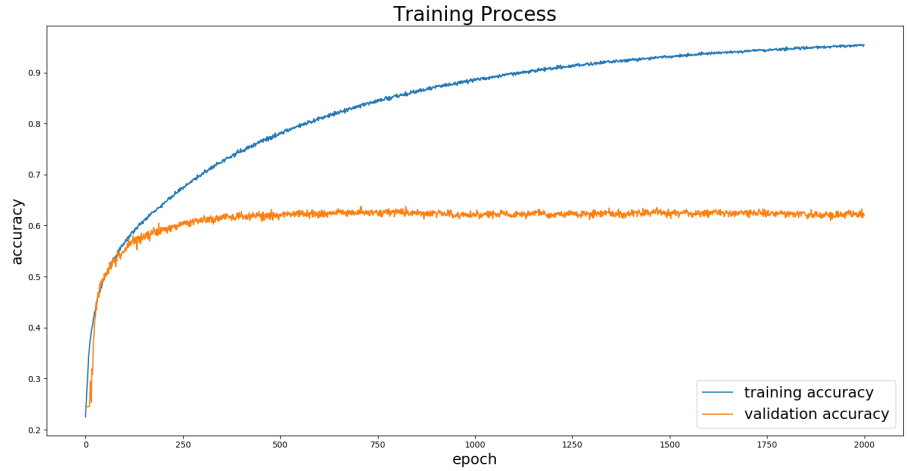
Conv2d(1 , 32 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (2 , 2))
BatchNorm2d(32)
ReLU()
MaxPool2d(2)
Dropout2d()
Conv2d(32 , 64 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (2 , 2))
BatchNorm2d(64)
ReLU()
MaxPool2d(2)
Dropout2d()
Conv2d(64 , 128 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (2 , 2))
BatchNorm2d(128)
ReLU()
MaxPool2d(2)
Dropout2d()
Conv2d(128 , 256 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (2 , 2))
BatchNorm2d(256)

ReLU()
MaxPool2d(2)
Dropout2d()
Conv2d(256 , 512 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (2 , 2))
BatchNorm2d(512)
ReLU()
MaxPool2d(2)
Dropout2d()
Conv2d(512 , 1024 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (2 , 2))
BatchNorm2d(1024)
ReLU()
MaxPool2d(2)
Dropout2d()
Linear(4096 , 100 , bias = True)
BatchNorm1d(100)
ReLU()
Dropout()
Linear(100 , 100 , bias = True)
BatchNorm1d(100)
ReLU()
Dropout()
Linear(100 , 7 , bias = True)

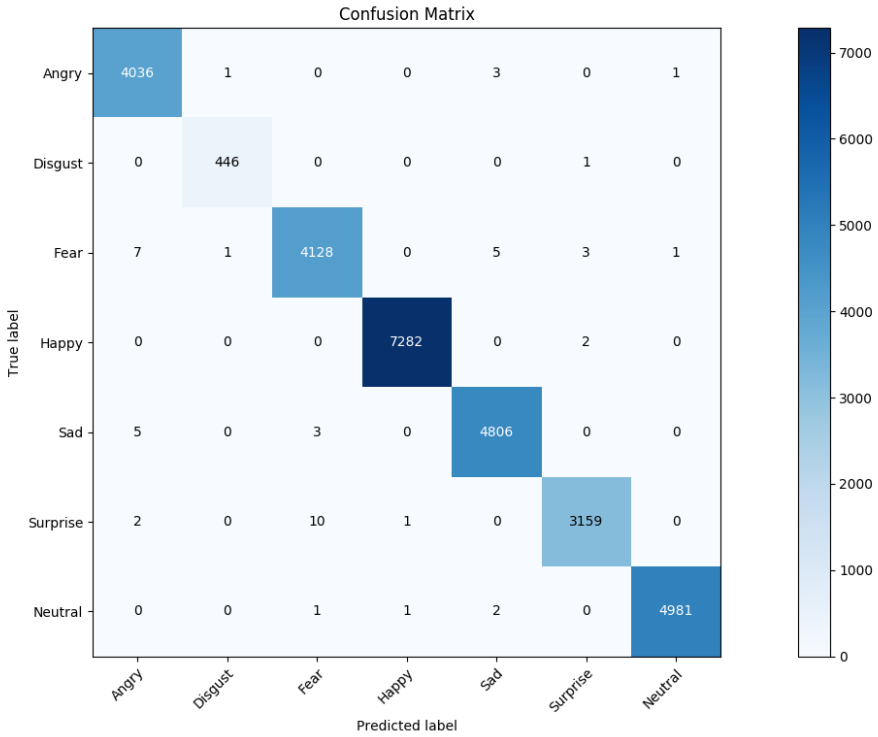
2. (1%) 請附上 model 的 training/validation history (loss and accuracy)。

我選出了 2000 張照片作爲 validation data，剩下的照片則作爲 training data，並且，我使用了 Adam 訓練 CNN，其中 learning rate 爲 0.0001，batch size 爲 1024，以此訓練了 2000 個 epoch，所得到的 loss 和 accuracy 如下圖所示：





3. (1%) 畫出 confusion matrix 分析哪些類別的圖片容易使 model 搞混，並簡單說明。(ref：https://en.wikipedia.org/wiki/Confusion_matrix)

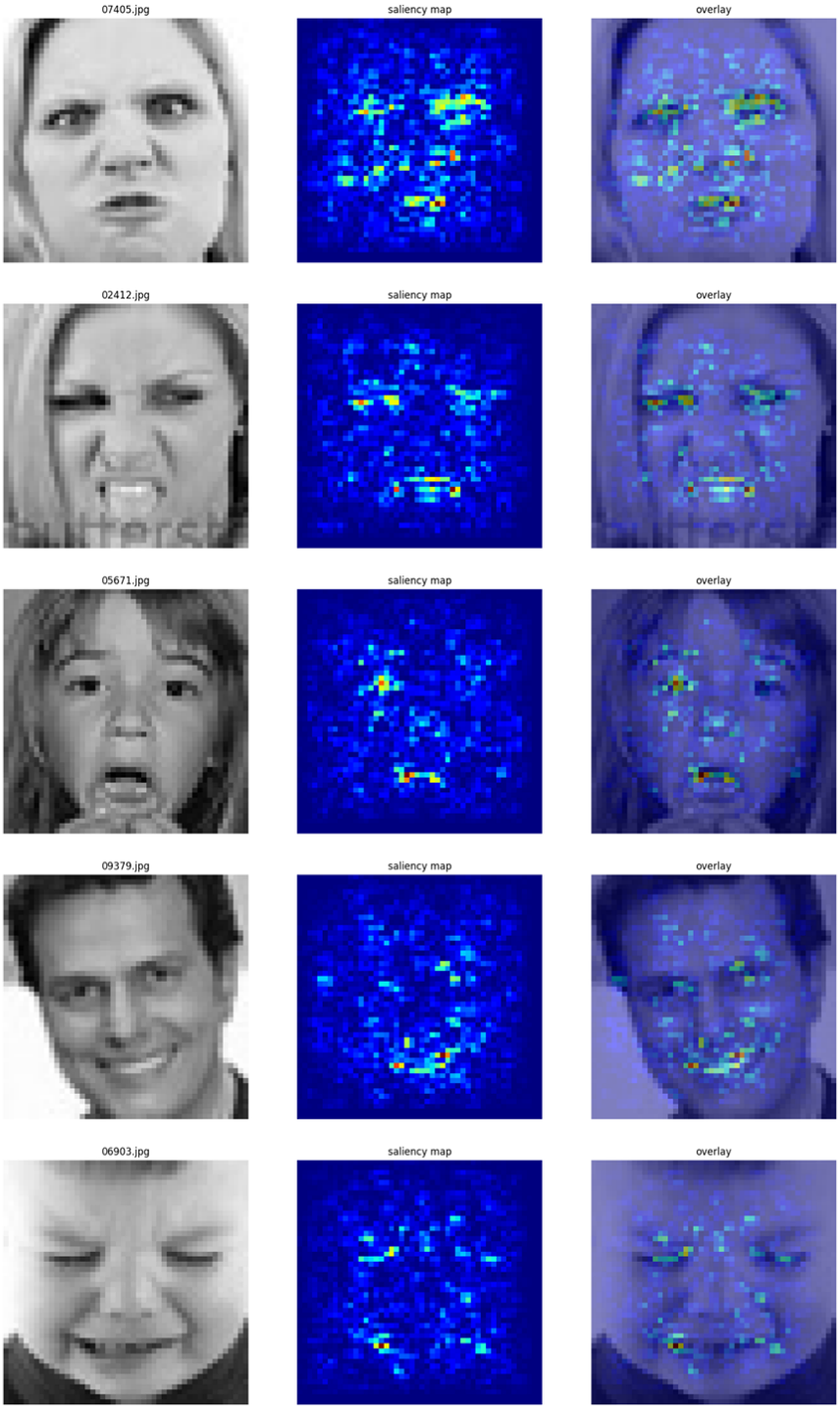


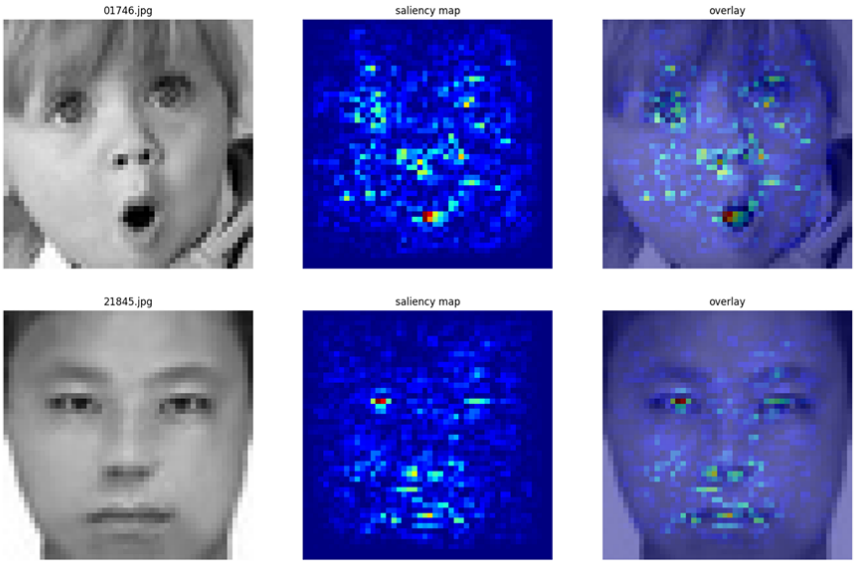
由 confusion matrix 可以看出，最容易辨識錯誤的情緒爲將 surprise 辨識爲 fear，其次爲將 fear 辨識爲 angry，再者爲將 fear 辨識爲 sad 以及將 sad 辨識爲 angry，我認爲其原因爲這些被辨識錯誤的情緒多爲負面情緒，其在人的面部表情呈現上其實相差無多，連人們也經常會辨識錯誤，在電腦的表現上似乎也是如此。

[關於第四及第五題]
可以使用簡單的 3-layer CNN model [64, 128, 512] 進行實作。

4. (1%) 畫出 CNN model 的 saliency map，並簡單討論其現象。(ref：https://reurl.cc/Qpjg8b)

以下爲從 7 種情緒中各隨機挑出一張照片，並使用 3-layer CNN model [64, 128, 512] 所畫出的 saliency map：

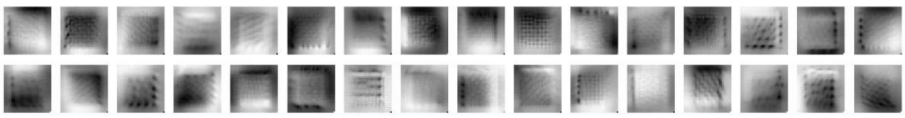

07405.jpg
saliency map
overlay
02412.jpg
saliency map
overlay
05671.jpg
saliency map
overlay
09379.jpg
saliency map
overlay
06903.jpg
saliency map
overlay



由此可以看出，saliency map 中數值較大的地方主要是集中在面部五官的部分，因此可以推測 CNN 主要是透過五官來辨識情緒。

5. (1%) 畫出最後一層的 filters 最容易被哪些 feature activate。(ref : https://reurl.cc/ZnrgYg)

以下爲在訓練完 3-layer CNN model [64, 128, 512] 之後，從最後一個 convolution layer 中取出 32 個 filter，找出可以使其最爲 activate 的各個 feature :



以上的 feature 看起來都是一些簡單的幾何圖案或材質，比預想中的簡單許多，我猜測其原因爲照片中人臉佔了絕大部分的面積，而人的五官本來就是由簡單的幾何形狀所構成，沒有太複雜的特徵，因此只要能偵測簡單的幾何形狀，便有可能辨識出人的五官，進而將照片中的人臉情緒做分類，此外，CNN 的架構簡單也有可能爲造成該現象的原因之一。

Part 2. Math Problem

1. Convolution (1%)

As we mentioned in class, image size may change after convolution layers. Consider a batch of image data with shape $(B, W, H, input\_channels)$, how will the shape change after the convolution layer?

$Conv2D(input\_channels,\ output\_channels,\ kernel\_size = (k_1,\ k_2),\ stride = (s_1,\ s_2),\ padding = (p_1,\ p_2))$

To simplify the answer : the padding tuple means that we pad p_1 pixels on both left and right side, and p_2 pixels for top and bottom.

solution

由 PyTorch 官方 API (https://pytorch.org/docs/stable/nn.htmlconv2d)，可知 $(B, W, H, input\_channels)$ 在經過題幹中所述的 convolution layer 之後，會變爲 $(B, \left\lfloor \frac{W+2P_1-k_1}{s_1} + 1 \right\rfloor, \left\lfloor \frac{H+2P_2-k_2}{s_2} + 1 \right\rfloor, output\_channels)$。 □

2. Batch Normalization (1%)

Besides Dropout, we usually use Batch Normalization in training nowadays [ref]. The trick is popular within the deep networks due to its convenience while training. It preserves the distribution within hidden layers and avoids gradient vanish.

The algorithm can be written as below :

Algorithm 1: Batch Normalization

Input : Values of x over a mini−batch: $\mathcal{B} = \{x_{1\cdots m}\}$;
Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$;
Parameters to be learned: γ, β;

$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$ //mini−batch mean
$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$ //mini−batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$ //normalize
$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$ //scale and shift

How to update γ and β from the optimization process of loss?

Just try to derive $\frac{\partial l}{\partial \hat{x}_i}$, $\frac{\partial l}{\partial \sigma^2_{\mathcal{B}}}$, $\frac{\partial l}{\partial \mu_{\mathcal{B}}}$, $\frac{\partial l}{\partial x_i}$, $\frac{\partial l}{\partial \gamma}$, $\frac{\partial l}{\partial \beta}$.

solution

因爲

$$\mu_{\mathcal{B}} = \mu_{\mathcal{B}}(x_1, x_2, \cdots, x_m) = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_{\mathcal{B}}^2 = \sigma_{\mathcal{B}}^2(x_1, x_2, \cdots, x_m, \mu_{\mathcal{B}}) = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$$

$$\hat{x}_i = \hat{x}_i(x_i, \mu_{\mathcal{B}}, \sigma_{\mathcal{B}}^2, \epsilon) = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

$$y_i = y_i(\hat{x}_i, \gamma, \beta) = \gamma\hat{x}_i + \beta$$

所以

$$\frac{\partial l}{\partial \hat{x}_i} = \frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \hat{x}_i} = \frac{\partial l}{\partial y_i} \cdot \frac{\partial}{\partial \hat{x}_i}(\gamma\hat{x}_i + \beta) = \frac{\partial l}{\partial y_i} \cdot \gamma$$

$$\begin{aligned}\frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i} \cdot \frac{\partial}{\partial \sigma_{\mathcal{B}}^2}\frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2}(\sigma_{\mathcal{B}}^2 + \epsilon)^{-\frac{3}{2}}\end{aligned}$$

$$\begin{aligned}\frac{\partial l}{\partial \mu_{\mathcal{B}}} &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial \mu_{\mathcal{B}}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2}\frac{\partial \sigma_{\mathcal{B}}^2}{\partial \mu_{\mathcal{B}}} \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i} \cdot \frac{\partial}{\partial \mu_{\mathcal{B}}}\frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\partial}{\partial \mu_{\mathcal{B}}}\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \\ &= \sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\mathcal{B}})}{m}\end{aligned}$$

$$\begin{aligned}\frac{\partial l}{\partial x_i} &= \frac{\partial l}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial x_i} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2}\frac{\partial \sigma_{\mathcal{B}}^2}{\partial x_i} + \frac{\partial l}{\partial \mu_{\mathcal{B}}}\frac{\partial \mu_{\mathcal{B}}}{\partial x_i} \\ &= \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{\partial}{\partial x_i}\frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\partial}{\partial x_i}\frac{1}{m}\sum_{j=1}^{m}(x_j - \mu_{\mathcal{B}})^2 + \frac{\partial l}{\partial \mu_{\mathcal{B}}} \cdot \frac{\partial}{\partial x_i}\frac{1}{m}\sum_{j=1}^{m} x_j \\ &= \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial l}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}\end{aligned}$$

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

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \frac{\partial y_i}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \cdot \frac{\partial}{\partial \beta}(\gamma \hat{x}_i + \beta) = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \ \square$$

3. Softmax and Cross Entropy (1%)

In classification problem, we use softmax as activation function and cross entropy as loss function.

$$\begin{aligned}
&softmax(z_t) = \frac{e^{z_t}}{\sum_i e^{z_i}} \\
&cross_entropy = L(y, \hat{y}) = -\sum_i y_i log \hat{y}_i \\
&cross_entropy = L_t(y_t, \hat{y}_t) = -y_t log \hat{y}_t \\
&\hat{y}_t = softmax(z_t)
\end{aligned}$$

Derive that $\frac{\partial L_t}{\partial z_t} = \hat{y}_t - y_t$.

solution

題幹中的 $L_t(y_t, \hat{y}_t) = -y_t log \hat{y}_t$ 爲當 $y_t = 1$ 時的 cross entropy loss，此時

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

而當 $y_t = 0$ 時，cross entropy loss 應爲 $L_t(y_t, \hat{y}_t) = -(1 - y_t) log(1 - \hat{y}_t)$，此時

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