* model architecture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer | Input shape | Filter | Stride | Padding | Output shape |
| Conv1 | (16, 1, 32, 32) | (3, 5, 5) | 1\*1 | no | (16, 3, 28, 28) |
| ReLU | (16, 3, 28, 28) |  |  |  | (16, 3, 28, 28) |
| Maxpooling | (16, 3, 28, 28) | (3, 3) | 2\*2 | no | (16, 3, 13, 13) |
| Flatten | (16, 3, 13, 13) |  |  |  | (16, 3\*13\*13) |
| Fully Connected 1 | (16, 3\*13\*13) |  |  |  | (16, 64) |
| ReLU | (16, 64) |  |  |  | (16, 64) |
| Fully Connected 2 | (16, 64) |  |  |  | (16, 3) |
| Softmax | (16, 3) |  |  |  | (16, 3) |

註1: Conv1的input shape=(16, 1, 32, 32)，16為batch size，1為channel(將照片灰階模式讀取並做normalization)

註2: Conv1的filter=(3, 5, 5)，3為產生三張feature map，5為kernel size

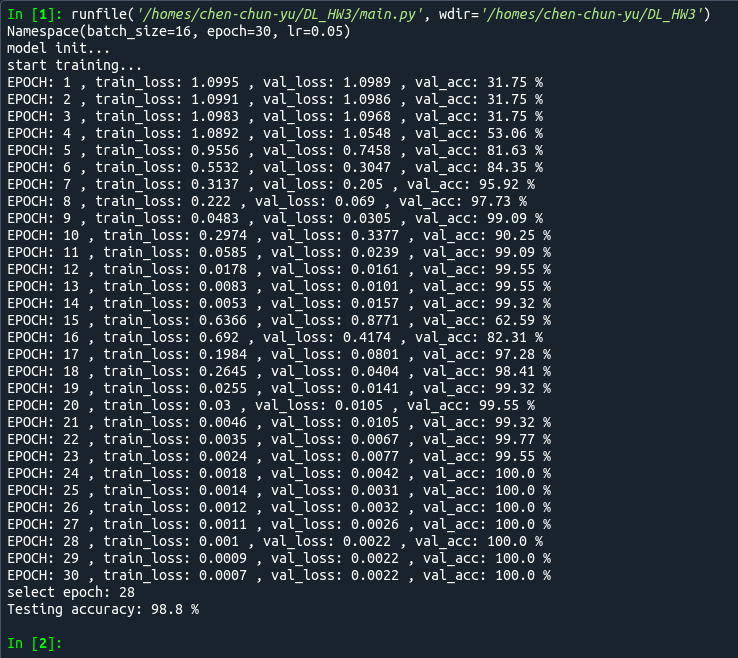
註3: Maxpooling的filter=(3, 3)，意思是在feature map中每個3\*3的2D區域內取一個最大值做降維，並且步幅為2

* loss function

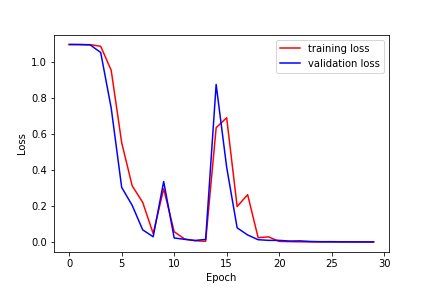
採用多類別交叉熵，公式如下:

N為樣本總數，M為類別數，為label(one hot vector)，為model predicted probability

* screenshot of testing result



* Plot training loss and validation loss



* Describe the major problem you encountered and how did you deal with it

手刻CNN的難度相較手刻DNN難上許多，不像DNN可以將數組攤平(Flatten)再處理，CNN要考慮到多維度之間的轉換，光是input image的shape就有四個維度(batch\_size, channel, width, height)，若input image的channel為3 (RGB)，那情況會更加複雜，我的作法是將其以灰階模式讀取(一層channel)再處理。另外就是模型參數的梯度更新，因為層數多，需要非常細心地做偏微分與連鎖率運算。