微積分 + AI Deep Learning + PyTorch

註記:

$$\text{括號符號}\left(scalar\right),\left[tensor\right],||norm||$$

$$\text{如：向量可使用} v_I = [v_i]_{i\in I}$$

$$\text{如：矩陣可使用} a_{IJ} = [a_{ij}]_{(i,j)\in I\times J}\text{表示}$$

$$\text{單一大寫字母代表集合，如：} I,J,K,L\ldots \text{ 代表指標集，如太多維可使用} I^1, I^2,\ldots$$

$$\text{對於取用同指標集} I \text{子集的元素，可使用} i,i',i'',\text{如太多個可使用} i^{(2)},i^{(3)}\ldots$$

$$\text{紅色代表為資料流，運算結果流}$$

$$\text{藍色代表模型參數}(parameters),\text{訓練時會受到最佳化演算法改變}$$

$$\text{綠色代表模型超參數}(hyper-parameters),\text{不會受訓練影響}$$

$$\text{紫色括號}[P]\in\{0,1\}\text{，代表 Iverson Bracket}$$

torch.nn

1. Linear Layers

$$Linear(x_J; w_{IJ}, b_I) := w_{IJ}\cdot x_J + b_I = \left[\sum_{j\in J} w_{ij}x_j + b_i\right]_{i\in I}$$

Gradient :

$$\nabla_{\omega_{IJ}} Linear = [x_j]_{(i,j)\in I\times J}$$
$$\nabla_{b_I} Linear = 1_I$$

---

2. Nonlinear Activations

$$Softmax(x_I) = \left[\frac{e^{x_i}}{||e^{x_I}||_1}\right]_{i\in I} \quad \text{其中分母} \; ||e^{x_I}||_1 := \sum_{i'\in I} e^{x_{i'}}$$

$$Tanh(x_I) := \left[\frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}}\right]_{i\in I}$$

Gradient :

$$\nabla_{x_I} Tanh(x_I) = [1 - Tanh^2(x_i)]_{i \in I}$$

$$Sigmoid(x_I) := \left[\frac{1}{1 + e^{-x_i}} \right]_{i \in I}$$

Gradient :

$$\nabla_{x_I} Sigmoid(x_I) = Sigmoid(x_I) \odot (1_I - Sigmoid(x_I))$$

$$LogSoftmax(x_I) := ln \circ Softmax(x_I) = \left[ln \left(\frac{e^{x_i}}{||e^{x_I}||_1} \right) \right]_{i \in I}$$

$$Softplus(x_I; \beta) = \left[\frac{1}{\beta} ln(1 + e^{\beta x_i}) \right]_{i \in I}$$ 註: β 每個維度都是同個參數

$$Softsign(x_I) = \left[\frac{x_i}{1 + |x_i|} \right]_{i \in I}$$

$$Thershold(x_I, \alpha) = \left[x_i \cdot [x_i > \alpha] + \alpha \cdot [x_i \leq \alpha] \right]_{i \in I}$$

$$ReLU(x_I) = \left[max(0, x_i) \right]_{i \in I}$$

$$PReLU(x_I) = \left[max(0, x_i) + a \cdot min(0, x_i) \right]_{i \in I}$$

註 ：激發函數通常寫成$\sigma(x_I)$

3. Dropout Layers

$$Dropout(x_I; p) = \left[0 \cdot [\#_i \leq p] + x_i \cdot [\#_i > p] \right]_{i \in I}$$ 其中$\#_I \sim uniform\left((0,1)^{|I|} \right)$為隨機向量

4. Sparse Layers

$$Embedding(z_J; I, D) := \left[w_{z_j d} \right]_{J \times D}$$ 其中$z_J \in I^{|J|}$, 參數矩陣為$w_{ID} := [w_{id}]_{(i,d) \in I \times D}$

註：在NLP(Natural Language Processing)領域

I代表詞種類(words)集合

$|D|$為 Word Embedding Dimension

不同的詞，可用$1, 2, 3, 4 \ldots |I|$編號，即 $I \underset{1-1}{\leftrightarrow} \{1, 2, 3, \ldots |I|\}$

$word_i$的詞向量$(word2vec)$即為$[\omega_{id}]_{d \in D}$

一句有$|J|$個詞的句子 $= z_J = [z_j] = [$第j個詞$]_{j=1,2,\ldots|J|}$

$Embedding($句子$) = [$詞實向量$] =$ 實矩陣

核心量化概念：詞 $\equiv$ 向量，句子 $\equiv$ 矩陣(有順序概念)，j代表位置，d代表詞特徵，詞特徵是演算法學來的!!

5.Distance Functions

$$CosineSimilarity(u_I, v_I) = \frac{u_I \cdot v_I}{max(||u_I||_2||v_I||_2, \epsilon)} = \frac{\sum_{i \in I} u_i v_i}{max\left(\sqrt{\sum_{i \in I} u_i^2 \sum_{i \in I} v_i^2}, \epsilon\right)}$$

$$PairwiseDistance(u_I; p) := ||u_I||_p = \left(\sum_{i \in I} |x_i|^p\right)^{\frac{1}{p}}$$

y_I^{pred}代表經由數學模型計算後的預測向量

y_I^{target}代表原始資料的目標向量(正確答案)

$|\mathcal{B}|$代表batchsize

$y_{ib}^{pred}, y_{ib}^{target}$代表樣本$b$的$y$值

註: 模型架構好以後 Pytorch 支援 Batch Input !!

$$Model([x_{Ib}]_{b \in \mathcal{B}}) := \Big[Model(x_{Ib})\Big]_{b \in \mathcal{B}}$$

6. Loss Functions (Sum Overall Batch Samples)

$$L1Loss\left(\left[(y_I^{pred}, y_I^{target})\right]_{b \in \mathcal{B}}\right) = \sum_{b \in \mathcal{B}} \left(\frac{1}{|I|} \sum_{i \in I} |y_{ib}^{pred} - y_{ib}^{target}|\right)$$

$$MSELoss\left(\left[(y_I^{pred}, y_I^{target})\right]_{b \in \mathcal{B}}\right) = \sum_{b \in \mathcal{B}} \left(\frac{1}{|I|} \sum_{i \in I} (y_{ib}^{pred} - y_{ib}^{target})^2\right)$$

$$CrossEntropyLoss\left(\left[(y_I^{pred}, y_I^{target})\right]_{b \in \mathcal{B}}\right) = H\left(y_I^{target}, Softmax(y_I^{pred})\right) = -\sum_{i \in I} y_i^{target} \ln\left(\frac{y_i^{pred}}{\sum_{i' \in I} y_{i'}^{pred}}\right)$$

註：y_I^{target} is one hot encoding , API use integer input

$$CRF(s_{IY}; \omega_{YY}) = -\left(\sum_{i=1}^{|I|} s_{iy_i} + \sum_{i=1}^{|I|-1} \omega_{y_i, y_{i+1}}\right)$$

註：$|I|$為句子長度，$|Y|$為 Label 種類集，s_{i,y_i}又稱為 emission score, y_I為 target label vector

註2：Bi-LSTM輸出為x_{ID}向量，$|D|$為 hidden dimension，需要再作線性轉換S_{IY}, $Linear(x_{ID}) = s_{IY}$

7.Recurrent Layers (Share Weight Matrix)

$$h_D \text{為接口}(\text{hidden dimension}), h_D^0 \text{可}\text{fixed}\text{或可加入一起學習}$$

$$RNNCell(x_I, h_D; \omega_{DI}, \omega_{DD}, b_D) = \tanh(\omega_{DI} x_I + \omega_{DD} h_D + b_D)$$

$$LSTMCell(x_I, c_D, h_D; \overbrace{\omega_{DI}^{x\to i}, \omega_{DD}^{h\to i}, \omega_{DI}^{x\to f}, \omega_{DD}^{h\to f}, \omega_{DI}^{x\to g}, \omega_{DD}^{h\to g}, \omega_{DI}^{x\to o}, \omega_{DD}^{h\to o}}^{\text{8 weight matrixs}}, \underbrace{b_D^i, b_D^f, b_D^g, b_D^o}_{\text{4 bias vectors}})$$

結構細節:

$$\begin{aligned}
f_D &:= \sigma(\omega_{DI}^{x\to f} x_I + \omega_{DD}^{h\to f} h_D + b_D^f) \\
i_D &:= \sigma(\omega_{DI}^{x\to i} x_I + \omega_{DD}^{h\to i} h_D + b_D^i) \\
o_D &:= \sigma(\omega_{DI}^{x\to o} x_I + \omega_{DD}^{h\to o} h_D + b_D^o) \\
g_D &:= tanh(\omega_{DI}^{x\to g} x_I + \omega_{DD}^{h\to g} h_D + b_D^g) \\
LSTMCell(x_I, c_D, h_D; \ldots) &= o_D \odot tanh(f_D \odot c_D + i_D \odot g_D)
\end{aligned}$$

$$GRUCell(x_I, h_D; \omega_{DI}^{x\to r}, \omega_{DD}^{h\to r}, \omega_{DD}^{x\to n}, \omega_{DI}^{x\to z}, \omega_{DD}^{h\to z}, \omega_{DD}^{h\to n}, b_D^r, b_D^n, b_D^z)$$

結構細節:

$$\begin{aligned}
r_D &= \sigma(\omega_{DI}^{x\to r} x_I + \omega_{DD}^{h\to r} h_D + b_D^r) \\
z_D &= \sigma(\omega_{DI}^{x\to z} x_I + \omega_{DD}^{h\to z} h_D + b_D^z) \\
n_D &= tanh(\omega_{DD}^{x\to n} x_I + r_D \odot (\omega_{DD}^{h\to n} h_D) + b_D^n) \\
GRUCell(x_I, h_D; \ldots) &= (1_D - z_D) \odot h_D + z_D \odot n_D
\end{aligned}$$
