微積分 + Al Deep Learning + PyTorch

註記:

括號符號
$$\left(scalar\right), \left[tensor\right], ||norm||$$
如:向量可使用 $v_I = [v_i]_{i \in I}$
如:矩陣可使用 $a_{IJ} = [a_{ij}]_{(i,j) \in I \times J}$ 表示
$$單-大寫字母代表集合,如: $I,J,K,L\dots$ 代表指標集,如太多維可使用 I^1,I^2,\dots
對於取用同個指標集 I 子集的元素,可使用 $i,i',i'',$ 如太多個可使用 $i^{(2)},i^{(3)}\dots$
紅色代表為資料流,運算結果流
藍色代表模型參數 $(parameters)$,訓練時會受到最佳化演算法改變
綠色代表模型超參數 $(hyper-parameters)$,不會受訓練影響
紫色括號 $[P] \in \{0,1\}$,代表 $Iverson\ Bracket$$$

torch.nn

1. Linear Layers

$$Linear(oldsymbol{x}_J; oldsymbol{w}_{IJ}, oldsymbol{b}_I) := oldsymbol{w}_{IJ} \cdot oldsymbol{x}_J + oldsymbol{b}_I = igg[\sum_{j \in J} oldsymbol{w}_{ij} oldsymbol{x}_j + oldsymbol{b}_iigg]_{i \in I}$$

Gradient:

$$egin{aligned}
abla_{\omega_{IJ}} Linear &= [rac{oldsymbol{x}}{oldsymbol{x}}]_{(i,j) \in I imes J} \
abla_{b_I} Linear &= 1_I \end{aligned}$$

2. Nonlinear Activations

$$Softmax(oldsymbol{x}_I) = igg[rac{e^{oldsymbol{x}_i}}{||e^{oldsymbol{x}_I}||_1}igg]_{i \in I}$$
 其中分縣 $||e^{oldsymbol{x}_I}||_1 := \sum_{i' \in I} e^{oldsymbol{x}_{i'}}$
$$Tanh(oldsymbol{x}_I) := igg[rac{e^{oldsymbol{x}_i} - e^{-oldsymbol{x}_i}}{e^{oldsymbol{x}_i} + e^{-oldsymbol{x}_i}}igg]_{i \in I}$$

Gradient:

$$\nabla_{\boldsymbol{x}_I} Tanh(\boldsymbol{x}_I) = [1 - Tanh^2(\boldsymbol{x}_i)]_{i \in I}$$

$$Sigmoid(oldsymbol{x}_I) := igg[rac{1}{1 + e^{-oldsymbol{x}_i}}igg]_{i \in I}$$

Gradient:

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

$$LogSoftmax(oldsymbol{x}_I) := ln\circ Softmax(oldsymbol{x}_I) = \left[ln\left(rac{e^{oldsymbol{x}_i}}{||e^{oldsymbol{x}_I}||_1}
ight)
ight]_{i\in I}$$
 是 $Softplus(oldsymbol{x}_I;eta) = \left[rac{1}{eta}ln(1+e^{etaoldsymbol{x}_i})
ight]_{i\in I}$ 是 $Softsign(oldsymbol{x}_I) = \left[rac{oldsymbol{x}_i}{1+|oldsymbol{x}_i|}
ight]_{i\in I}$
$$Thershold(oldsymbol{x}_I,lpha) = \left[oldsymbol{x}_i\cdot[oldsymbol{x}_i>lpha]+lpha\cdot[oldsymbol{x}_i\leqlpha]
ight]_{i\in I}$$

$$ReLU(oldsymbol{x}_I) = \left[max(0,oldsymbol{x}_i)
ight]_{i\in I}$$

$$PReLU(oldsymbol{x}_I) = \left[max(0,oldsymbol{x}_i)+a\cdot min(0,oldsymbol{x}_i)
ight]_{i\in I}$$
 註:激發函數通常寫成 $\sigma(oldsymbol{x}_I)$

3. Dropout Layers

$$Dropout(oldsymbol{x}_I;p) = iggl[0\cdot [\#_i \leq p] + oldsymbol{x}_i\cdot [\#_i > p]iggr]_{i\in I}$$
 其中 $\#_I \sim uniformiggl((0,1)^{|I|}iggr)$ 為隨機向量

4. Sparse Layers

$$Embedding(\pmb{z}_J; \pmb{I}, D) := \left[\pmb{w}_{z_j d}
ight]_{\pmb{J} imes D}$$
 其中 $\pmb{z}_J \in I^{|J|}$,參數矩陣為 $\pmb{w}_{ID} := [w_{id}]_{(i,d) \in I imes D}$

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
$$1,2,3,4...|I|$$
編號,即 $I \underset{1-1}{\longleftrightarrow} \{1,2,3,\ldots |I|\}$

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

-句有
$$|J|$$
個詞的句子 $=z_J=[z_j]=[$ 第 j 個詞 $]_{j=1,2,...|J|}$ $Embedding($ 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

$$Cosine Similarity(u_I,v_I) = rac{u_I \cdot v_I}{max(||u_I||_2||v_I||_2,\epsilon)} = rac{\sum_{i \in I} u_i v_i}{maxigg(\sqrt{\sum_{i \in I} u_i^2 \sum_{i \in I} v_i^2},\epsilonigg)}$$

$$Pairwise Distance(u_I;p) := ||u_I||_p = igg(\sum_{i \in I} |x_i|^pigg)^{rac{1}{p}}$$

$$y_I^{pred}$$
 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 $igg(\mathbb{E}$ 证確答案 $igg)$ | $oldsymbol{\mathcal{B}}$ |代表 batch size
$$y_{ib}^{pred}, y_{ib}^{target}$$
 代表樣本 $oldsymbol{b}$ 數值

註: 模型架構好以後 Pytorch 支援 Batch Input!!

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

6. Loss Functions (Sum Overall Batch Samples)

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

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

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

 $\ensuremath{\mathbbmm{}}\xspace: y_I^{target}$ is one hot encoding , API use integer input

$$CRF(oldsymbol{s}_{IY};oldsymbol{\omega}_{YY}) = -igg(\sum_{i=1}^{|I|}oldsymbol{s}_{iy_i} + \sum_{i=1}^{|I|-1}oldsymbol{\omega}_{y_i,y_{i+1}}igg)$$

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 為接口 (hidden dimension), h_D^0 可 fixed或可加入一起學習

$$RNNCell(oldsymbol{x}_I,h_D;\omega_{DI},\omega_{DD},b_D) = anh(oldsymbol{w}_{DI}oldsymbol{x}_I + \omega_{DD}oldsymbol{h}_D + oldsymbol{b}_D)$$

$$8 ext{ weight matrixs}$$
 $LSTMCell(oldsymbol{x}_I,c_D,h_D;oldsymbol{\omega}_{DI}^{x o i},\omega_{DD}^{h o i},\omega_{DI}^{x o f},\omega_{DD}^{h o f},\omega_{DI}^{x o g},\omega_{DD}^{h o g},\omega_{DI}^{h o g},\omega_{DD}^{h o g},\omega_{DD}^{h o o},b_D^i,b_D^g,b_D^o)$

$$4 ext{ bias vectors}$$

結構細節:

$$egin{aligned} f_D &:= \sigma(\omega_{DI}^{x o f} oldsymbol{x}_I + \omega_{DD}^{h o f} oldsymbol{h}_D + oldsymbol{b}_D^f) \ i_D &:= \sigma(\omega_{DI}^{x o i} oldsymbol{x}_I + \omega_{DD}^{h o i} oldsymbol{h}_D + oldsymbol{b}_D^i) \ o_D &:= \sigma(\omega_{DI}^{x o o} oldsymbol{x}_I + \omega_{DD}^{h o o} oldsymbol{h}_D + oldsymbol{b}_D^o) \ g_D &:= tanh(\omega_{DI}^{x o g} oldsymbol{x}_I + \omega_{DD}^{h o g} oldsymbol{h}_D + oldsymbol{b}_D^g) \ LSTMCell(oldsymbol{x}_I, c_D, h_D; \ldots) &= o_D \odot tanh(f_D \odot oldsymbol{c}_D + i_D \odot g_D) \end{aligned}$$

$$GRUCell(oldsymbol{x}_{I},oldsymbol{h}_{D};\omega_{DI}^{x
ightarrow r},\omega_{DD}^{h
ightarrow r},\omega_{DD}^{x
ightarrow r},\omega_{DI}^{x
ightarrow z},\omega_{DD}^{h
ightarrow z},\omega_{DD}^{h
ightarrow r},b_{D}^{r},b_{D}^{r},b_{D}^{z})$$

結構細節:

$$egin{aligned} r_D &= \sigma(oldsymbol{\omega}_{DI}^{x
ightarrow r} oldsymbol{x}_I + oldsymbol{\omega}_{DD}^{h
ightarrow r} oldsymbol{h}_D + oldsymbol{b}_D^r) \ z_D &= \sigma(oldsymbol{\omega}_{DI}^{x
ightarrow z} oldsymbol{x}_I + oldsymbol{\omega}_{DD}^{h
ightarrow z} oldsymbol{h}_D + oldsymbol{b}_D^n) \ n_D &= tanh(oldsymbol{\omega}_{DD}^{x
ightarrow n} oldsymbol{x}_I + r_D \odot (oldsymbol{\omega}_{DD}^{h
ightarrow n} oldsymbol{h}_D) + oldsymbol{b}_D^n) \ GRUCell(oldsymbol{z}_I, h_D; \dots) &= (1_D - z_D) \odot oldsymbol{h}_D + z_D \odot n_D \end{aligned}$$