

画像実験課題 A, B

1029323422 天野岳洋

2022 年 12 月 23 日

1 概要

ここでは発展課題 A について取り組んだ内容について述べたのちに, コンテストに対して取り組んだ内容を述べる. 以後明記はしないが, 伝播層では (B, ?, 1) という形で伝播するものとし, 逆伝播層では畳み込み層, プーリング層を除き (?, B) という形でデータを扱っていることに注意したい.

2 AdvancedA

発展課題 A の実装について述べる. 実装が簡単であったり, 定義通りにしか実装していない場合は説明は省略もしくは非常に簡単に述べるものとする.

2.1 A1

活性化関数として Sigmoid 関数の代わりに RELU 関数を用いるという内容である. RELU 関数は次のように表せる関数である.

$$RELU(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (1)$$

また逆伝播は以下のようである

$$RELU'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

自分の実装を示す.

Listing 1 RELU

```
1  def __init__(self):
2      pass
3
4  def prop(self, x):
5      self.input = x
6      self.B, self.M, _ = x.shape
7      return np.where(x <= 0, 0, x)
8
9  def back(self, delta):
10     return delta * (np.where(self.input <= 0, 0, 1).reshape(self.B, self.M).T
    )
```

2.2 A2

Dropout 層を実装せよという内容である. Dropout 層の定義は以下のものである.

$$\text{Dropout}(x) = \begin{cases} x & (\text{ノードが無視されない場合}) \\ 0 & (\text{ノードが無視される場合}) \end{cases} \quad (3)$$

また逆伝播では,

$$\begin{aligned} \frac{En}{x} &= \frac{En}{y} \frac{y}{x} \\ &= \begin{cases} \frac{En}{y} & (\text{ノードが無視されない場合}) \\ 0 & (\text{ノードが無視された場合}) \end{cases} \end{aligned} \quad (4)$$

具体的な実装の説明に移る. Dropout 層のハイパーパラメータを ρ とする. このハイパーパラメータをもとにマスクされるノードの個数を定める. その後 `random.choice` によってどのノードがマスクされるかを選択し, 適切な処理をすればよい. またテストの際には定義にのっとり, マスクはせずに定数倍を行っている.

Listing 2 Dropout

```
1  def __init__(self, phi, M):
2      self.phi = phi
3      self.msk_num = int(M*phi)
4      self.M = M
5
6  def prop(self, x):
7      B = x.shape[0]
8      M = self.M
9      drop_random = np.repeat(np.random.choice(M, self.msk_num), B).reshape(-1,
10                                     B).T
11      mask_vector = np.ones((B, M))
12      mask_vector[np.repeat(np.arange(B), self.msk_num), drop_random.flatten()]
13                  = 0
14      self.msk = mask_vector.reshape(B, -1, 1)
15      return x * self.msk
16
17  def back(self, delta):
18      return delta * self.msk.reshape(self.B, self.M).transpose(1, 0)
19
20  def test(self, x):
21      return x * (1 - self.phi)
```

`prop` 層の `mask_vector` の作り方が少し複雑なので, 例を挙げて説明する. 例えば, `msk_num = 3`, `M = 5`, `B = 2` の時を考える. 今 `drop_random` は 0~4 から 3 個重複を許さずに選び, それを 2 回

ずつ繰り返し、次元を (3, 2) に変え転置をとるものだから、順番に追っていくと、例えば, 0, 1, 3 が選ばれたとすると、次のような形で処理が行われ、drop_random が得られることとなる。

$$[0, 1, 3] \rightarrow [0, 0, 1, 1, 3, 3] \rightarrow \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 3 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 3 \\ 0 & 1 & 3 \end{bmatrix}$$

続いて同様に mask_vector の推移を説明する。まず, (B, M) の形で全て 1 が入ったもので初期化され、その後 mask_vector[[0,0,0,1,1,1][0 1 3 0 1 3]] = 0 となっている。つまり

$$mask_vector \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

というような推移になっている。あとはこれを適切な形に変形させ、入力データとアダマール積をとればあるノードの出力が 0 になっていることが簡単にわかる。また逆伝播層でも入力の shape が違うことを除けば同様にアダマール積をとるだけである。

2.3 A3

Batch-Normalization を行えというものである。定義式は教科書のとおりである。

2.4 A4

様々な最適化手法を試すものとなっている。

2.5 A5

まだできてません。

2.6 A6

畳み込み層の実装を行う

2.7 A7

プーリングを行う。今回は MaxPooling を行うものとする。

3 contest

3.1 Data-argumentation

3.2 最終的な構成

3.3 不採用群

3.3.1 SAM

3.3.2 半教師あり学習

3.3.3 randomcrop

3.3.4 randomerasing

3.3.5 MixUp

3.3.6 cutout

3.3.7 label-smoothing