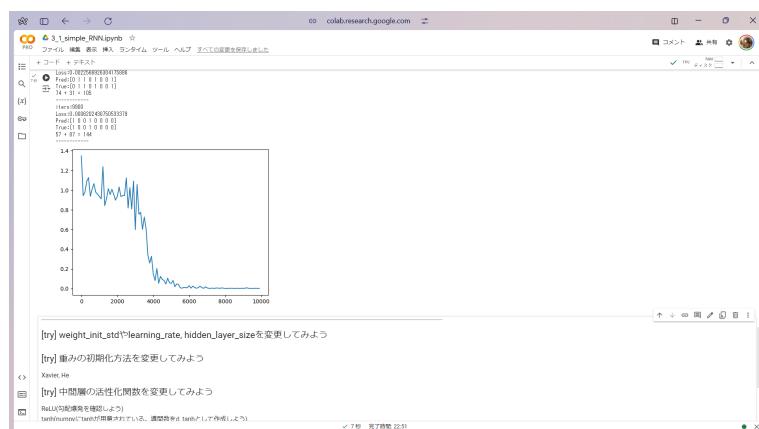


# 第1部

## 深層学習 day3 実装演習キャプチャ 1

## 1 3\_1\_simple\_RNN

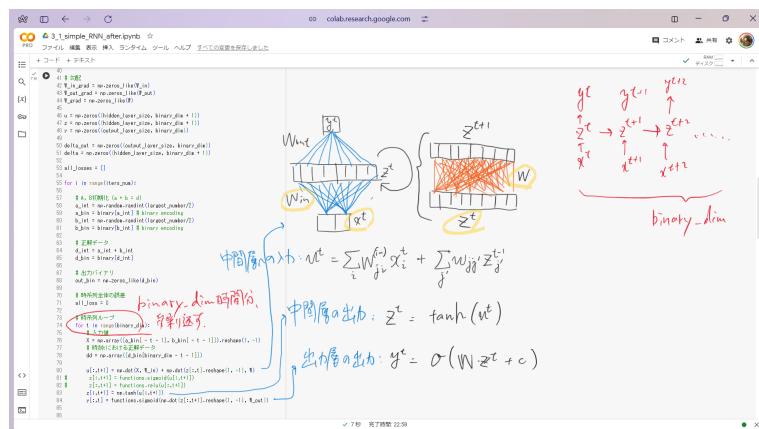
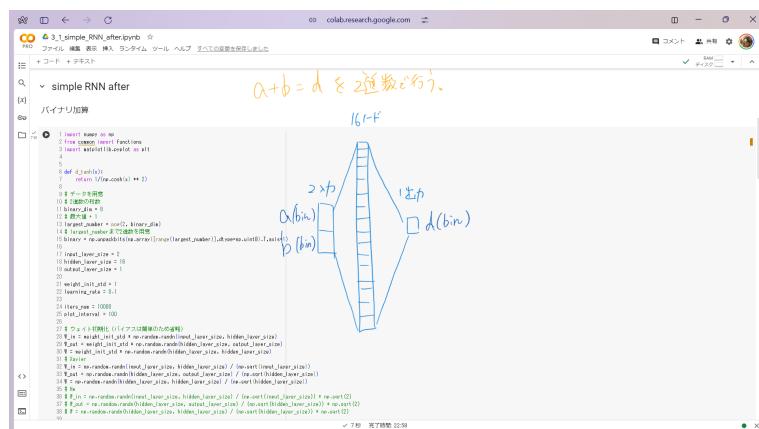


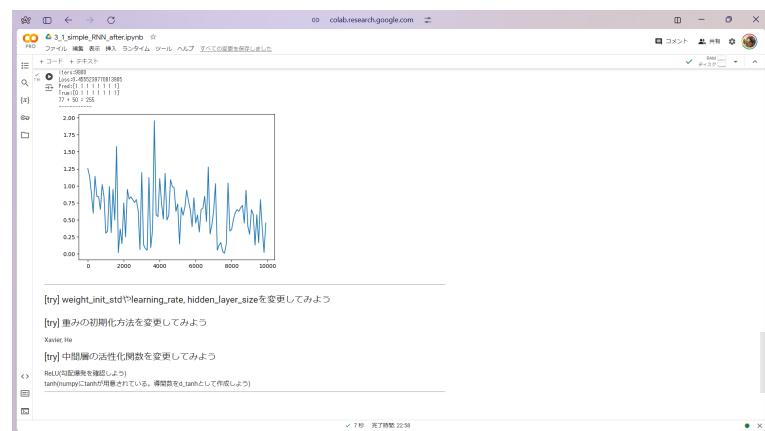
## 2 3\_1\_simple\_RNN\_after

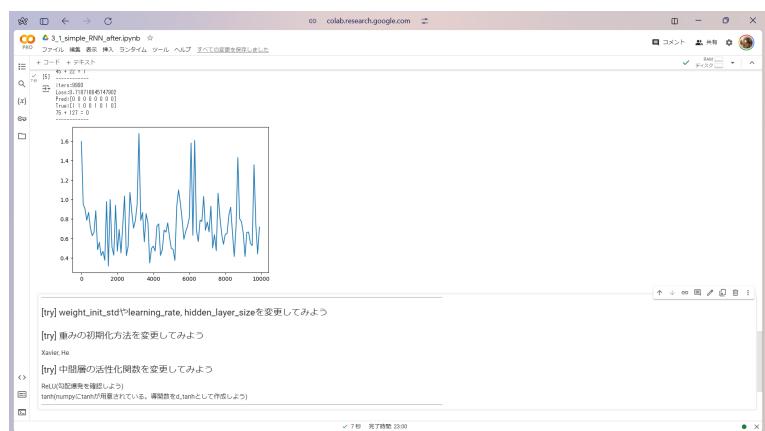
The screenshot shows the Google Colab interface with the following details:

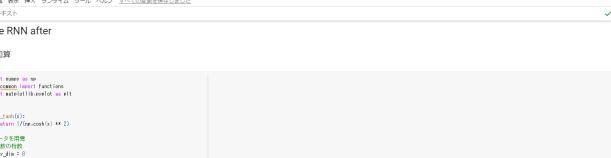
- File**, **Help**, **Logout** buttons are at the top.
- simple\_RNN\_after.pyrb** is the active notebook.
- コメント**, **新規**, **スクリプト** buttons are on the right.
- 検索** and **ナビゲーション** icons are on the left.
- 準備** section is expanded.
- Google Colab 用の処理** sub-section is expanded.
- 下記を実行します** heading.
- ドライブマウント
  - ノートブックファイルと同じフォルダへの移動
- Googleドライブのライセンス** note: ノートブックを単に単に `DNN_code/DNN_code_colab/day3` フォルダを置くことを推奨しています。必要に応じて、パスを変更してください。
- 実行** button with a play icon.
- コード** section shows the following code:

```
# Import Colab の関数を読み込む
# !pip install -q tensorflow
# !apt-get -qq update && apt-get -qq install -y libopenblas-dev
# !curl -O https://storage.googleapis.com/tensorflow/mac/tensorflow-1.13.1-py3-none-any.whl
# !pip install tensorflow-1.13.1-py3-none-any.whl
# !rm tensorflow-1.13.1-py3-none-any.whl
```
- 表示** at `/content`.
- sys.pathの設定** section.
- simple RNN after** section.
- バイナリ追加** button.



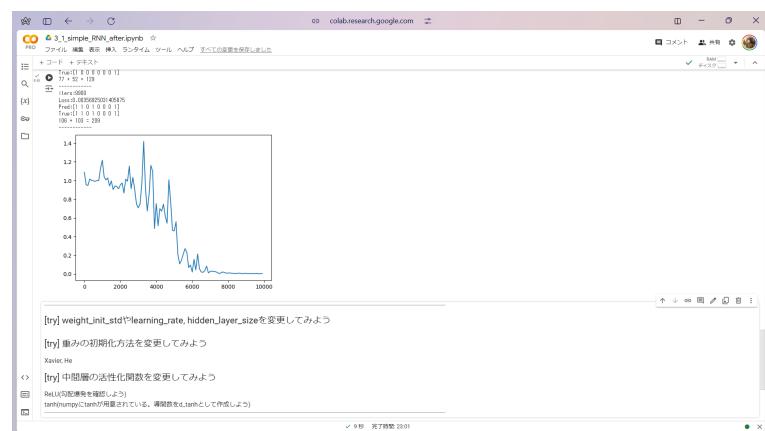






The screenshot shows a Google Colab notebook titled "simpleRNN after.ipynb". The code cell contains the following Python code:

```
1 import numpy as np
2 import tensorflow as tf
3 from tensorflow.keras import layers
4 from tensorflow.keras import models
5
6 def createModel():
7     return tf.keras.Sequential([
8         layers.Dense(10, activation='relu', input_shape=(10,)),
9         layers.Dense(1)
10    ])
11
12 # データの準備
13 # (訓練用データ, テスト用データ) のタプルを返す関数
14 def load_data():
15     (x_train, y_train), (x_test, y_test) = tf.keras.datasets.binarize_mnist.load_data()
16
17     x_train = x_train.reshape(-1, 10).astype('float32') / 255.0
18     y_train = y_train.astype('float32')
19
20     x_test = x_test.reshape(-1, 10).astype('float32') / 255.0
21     y_test = y_test.astype('float32')
22
23     return (x_train, y_train), (x_test, y_test)
24
25 # モデルの学習
26 def train(x_train, y_train, x_test, y_test):
27     model = createModel()
28
29     model.compile(optimizer='adam',
30                   loss='binary_crossentropy',
31                   metrics=['accuracy'])
32
33     history = model.fit(x_train, y_train,
34                          epochs=10,
35                          validation_data=(x_test, y_test))
36
37     return history
```



### 3 3\_2\_tf\_languagemodel

predict\_word.py

```
# coding: utf-8
# ファイル名: predict_word.py
# 用途: 単語の類似度を計算する
# ライセンス: MIT
# 版本: 1.0
# 作成者: 田中一也
# 最終更新日: 2023/09/10

# 実行環境: Python 3.8.12
# 環境変数: None
# ディレクトリ: /content/drive/MyDrive/Colab Notebooks/word2vec
# ファイル名: predict_word.py
# ファイル内容:
# 1. モデル読み込み
# 2. テスト用単語入力
# 3. モデル実行
# 4. 結果表示

# モデル読み込み
model = Word2Vec.load("word2vec.model")

# テスト用単語入力
target_word = "apple"

# モデル実行
similar_words = model.wv.most_similar(target_word)

# 結果表示
for word, similarity in similar_words:
    print(f'{word}: {similarity}')


# Prediction: I am looking at the book.
Prediction: I am looking at the book.
```

予測結果は、下記のコメント通り実行してもよい。(実行操作が長いため、直近の操作など誤認すること)

↑↑↑ 練習用モデルの構築  
↑↑↑ 実際にモデルを作成すると実際に用いた辞書ファイルがあるため必ずして操作しないでください  
↑↑↑ 実際にモデルを作成する場合、モデルファイルを保存  
↑↑↑ 実際にモデルを作成する場合、モデルファイルを保存  
↑↑↑ 実際にモデルを作成する場合、モデルファイルを保存  
↑↑↑ 実際にモデルを作成する場合、モデルファイルを保存



```
#predict_word.pyrb
# ファイル 読み込み 挿入 リンク ランタイム シール ヘルプ 全ての履歴を保存しました
+ コード + テキスト
[1] 1 insert_time
2 insert_datetime
3
4 class Language:
5     def __init__(self, self_vocabulary_size):
6         self.insert_time = 0
7         self.insert_datetime = 0
8         self.vocabulary_size = 30
9         self.output_size = self.vocabulary_size
10        self.hidden_size = 10
11        self._vocab = []
12        self.corpus = []
13        self.vocabulary_size = len(vocabulary)
14        self.vocabulary_size += len(insert_datetime) // 2
15        self.vocabulary_size += len(insert_time) // 2
16        self.hidden_layer_size = 10 // 隠れ層の数(2)で割る
17        self.output_layer_size = self.vocabulary_size // 出力層の数
18        self.output_size = self.output_layer_size
19        self.output_size += 1 // 末尾にEOS(End Of Sentence)を付ける
20        self.output_size += 1 // 開始記号の付加
21        self.output_size *= 2 // 全ての文字を大文字と小文字の組合せにする
22        self.output_size *= 2 // 全ての文字を大文字と小文字の組合せにする
23        self.output_size *= 2 // 全ての文字を大文字と小文字の組合せにする
24        self.output_size *= 2 // 全ての文字を大文字と小文字の組合せにする
25
26    # 1. 用語登録(登録する用語)
27    def add_vocab(self, word):
28        if word not in self._vocab:
29            self._vocab.append(word)
30
31    # 2. データ登録(登録するデータ)
32    def add_data(self, data):
33        if data not in self.corpus:
34            self.corpus.append(data)
35
36    # 3. モデル構築(モデルの構築)
37    def build_model(self, input_size, hidden_size, output_size, vocab_size):
38        self.model = nn.RNN(input_size, hidden_size, num_layers=1, batch_first=True)
39        self.model = nn.Sequential(self.model, nn.Linear(hidden_size, output_size))
40
41        self.trainable_variables = [self.model.parameters(), self.output_size]
42        self.optimizer = None
43
44    # 4. モデル学習(モデルの訓練)
45    def train(self, input, target):
46        input = torch.LongTensor(input)
47        target = torch.LongTensor(target)
48
49        self.model.train()
50
51        # 5. モデルの訓練(訓練の実行)
52        self.model.zero_grad()
53        self.model.zero_grad()
54        self.model.zero_grad()
55        self.model.zero_grad()
56
57        # 6. モデルの評価(評価の実行)
58        self.model.eval()
59
60        # 7. モデルの予測(予測の実行)
61        self.model.zero_grad()
62        self.model.zero_grad()
63        self.model.zero_grad()
64
65        # 8. モデルの学習(学習の実行)
66        self.model.zero_grad()
67
68        # 9. モデルの評価(評価の実行)
69        self.model.zero_grad()
70
71        # 10. モデルの予測(予測の実行)
72        self.model.zero_grad()
73
74        # 11. モデルの学習(学習の実行)
75
76        # 12. モデルの評価(評価の実行)
77
78        # 13. モデルの予測(予測の実行)
79
80        # 14. モデルの学習(学習の実行)
81
82        # 15. モデルの評価(評価の実行)
83
84        # 16. モデルの予測(予測の実行)
85
86        # 17. モデルの学習(学習の実行)
87
88        # 18. モデルの評価(評価の実行)
89
90        # 19. モデルの予測(予測の実行)
91
92        # 20. モデルの学習(学習の実行)
93
94        # 21. モデルの評価(評価の実行)
95
96        # 22. モデルの予測(予測の実行)
97
98        # 23. モデルの学習(学習の実行)
99
100       # 24. モデルの評価(評価の実行)
101
102       # 25. モデルの予測(予測の実行)
103
104       # 26. モデルの学習(学習の実行)
105
106       # 27. モデルの評価(評価の実行)
107
108       # 28. モデルの予測(予測の実行)
109
110       # 29. モデルの学習(学習の実行)
111
112       # 30. モデルの評価(評価の実行)
113
114       # 31. モデルの予測(予測の実行)
115
116       # 32. モデルの学習(学習の実行)
117
118       # 33. モデルの評価(評価の実行)
119
120       # 34. モデルの予測(予測の実行)
121
122       # 35. モデルの学習(学習の実行)
123
124       # 36. モデルの評価(評価の実行)
125
126       # 37. モデルの予測(予測の実行)
127
128       # 38. モデルの学習(学習の実行)
129
130       # 39. モデルの評価(評価の実行)
131
132       # 40. モデルの予測(予測の実行)
133
134       # 41. モデルの学習(学習の実行)
135
136       # 42. モデルの評価(評価の実行)
137
138       # 43. モデルの予測(予測の実行)
139
140       # 44. モデルの学習(学習の実行)
141
142       # 45. モデルの評価(評価の実行)
143
144       # 46. モデルの予測(予測の実行)
145
146       # 47. モデルの学習(学習の実行)
147
148       # 48. モデルの評価(評価の実行)
149
150       # 49. モデルの予測(予測の実行)
151
152       # 50. モデルの学習(学習の実行)
153
154       # 51. モデルの評価(評価の実行)
155
156       # 52. モデルの予測(予測の実行)
157
158       # 53. モデルの学習(学習の実行)
159
160       # 54. モデルの評価(評価の実行)
161
162       # 55. モデルの予測(予測の実行)
163
164       # 56. モデルの学習(学習の実行)
165
166       # 57. モデルの評価(評価の実行)
167
168       # 58. モデルの予測(予測の実行)
169
170       # 59. モデルの学習(学習の実行)
171
172       # 60. モデルの評価(評価の実行)
173
174       # 61. モデルの予測(予測の実行)
175
176       # 62. モデルの学習(学習の実行)
177
178       # 63. モデルの評価(評価の実行)
179
180       # 64. モデルの予測(予測の実行)
181
182       # 65. モデルの学習(学習の実行)
183
184       # 66. モデルの評価(評価の実行)
185
186       # 67. モデルの予測(予測の実行)
187
188       # 68. モデルの学習(学習の実行)
189
190       # 69. モデルの評価(評価の実行)
191
192       # 70. モデルの予測(予測の実行)
193
194       # 71. モデルの学習(学習の実行)
195
196       # 72. モデルの評価(評価の実行)
197
198       # 73. モデルの予測(予測の実行)
199
200       # 74. モデルの学習(学習の実行)
201
202       # 75. モデルの評価(評価の実行)
203
204       # 76. モデルの予測(予測の実行)
205
206       # 77. モデルの学習(学習の実行)
207
208       # 78. モデルの評価(評価の実行)
209
210       # 79. モデルの予測(予測の実行)
211
212       # 80. モデルの学習(学習の実行)
213
214       # 81. モデルの評価(評価の実行)
215
216       # 82. モデルの予測(予測の実行)
217
218       # 83. モデルの学習(学習の実行)
219
220       # 84. モデルの評価(評価の実行)
221
222       # 85. モデルの予測(予測の実行)
223
224       # 86. モデルの学習(学習の実行)
225
226       # 87. モデルの評価(評価の実行)
227
228       # 88. モデルの予測(予測の実行)
229
230       # 89. モデルの学習(学習の実行)
231
232       # 90. モデルの評価(評価の実行)
233
234       # 91. モデルの予測(予測の実行)
235
236       # 92. モデルの学習(学習の実行)
237
238       # 93. モデルの評価(評価の実行)
239
240       # 94. モデルの予測(予測の実行)
241
242       # 95. モデルの学習(学習の実行)
243
244       # 96. モデルの評価(評価の実行)
245
246       # 97. モデルの予測(予測の実行)
247
248       # 98. モデルの学習(学習の実行)
249
250       # 99. モデルの評価(評価の実行)
251
252       # 100. モデルの予測(予測の実行)
253
254       # 101. モデルの学習(学習の実行)
255
256       # 102. モデルの評価(評価の実行)
257
258       # 103. モデルの予測(予測の実行)
259
260       # 104. モデルの学習(学習の実行)
261
262       # 105. モデルの評価(評価の実行)
263
264       # 106. モデルの予測(予測の実行)
265
266       # 107. モデルの学習(学習の実行)
267
268       # 108. モデルの評価(評価の実行)
269
270       # 109. モデルの予測(予測の実行)
271
272       # 110. モデルの学習(学習の実行)
273
274       # 111. モデルの評価(評価の実行)
275
276       # 112. モデルの予測(予測の実行)
277
278       # 113. モデルの学習(学習の実行)
279
280       # 114. モデルの評価(評価の実行)
281
282       # 115. モデルの予測(予測の実行)
283
284       # 116. モデルの学習(学習の実行)
285
286       # 117. モデルの評価(評価の実行)
287
288       # 118. モデルの予測(予測の実行)
289
290       # 119. モデルの学習(学習の実行)
291
292       # 120. モデルの評価(評価の実行)
293
294       # 121. モデルの予測(予測の実行)
295
296       # 122. モデルの学習(学習の実行)
297
298       # 123. モデルの評価(評価の実行)
299
300       # 124. モデルの予測(予測の実行)
301
302       # 125. モデルの学習(学習の実行)
303
304       # 126. モデルの評価(評価の実行)
305
306       # 127. モデルの予測(予測の実行)
307
308       # 128. モデルの学習(学習の実行)
309
310       # 129. モデルの評価(評価の実行)
311
312       # 130. モデルの予測(予測の実行)
313
314       # 131. モデルの学習(学習の実行)
315
316       # 132. モデルの評価(評価の実行)
317
318       # 133. モデルの予測(予測の実行)
319
320       # 134. モデルの学習(学習の実行)
321
322       # 135. モデルの評価(評価の実行)
323
324       # 136. モデルの予測(予測の実行)
325
326       # 137. モデルの学習(学習の実行)
327
328       # 138. モデルの評価(評価の実行)
329
330       # 139. モデルの予測(予測の実行)
331
332       # 140. モデルの学習(学習の実行)
333
334       # 141. モデルの評価(評価の実行)
335
336       # 142. モデルの予測(予測の実行)
337
338       # 143. モデルの学習(学習の実行)
339
340       # 144. モデルの評価(評価の実行)
341
342       # 145. モデルの予測(予測の実行)
343
344       # 146. モデルの学習(学習の実行)
345
346       # 147. モデルの評価(評価の実行)
347
348       # 149. モデルの予測(予測の実行)
349
350       # 150. モデルの学習(学習の実行)
351
352       # 151. モデルの評価(評価の実行)
353
354       # 152. モデルの予測(予測の実行)
355
356       # 153. モデルの学習(学習の実行)
357
358       # 154. モデルの評価(評価の実行)
359
360       # 155. モデルの予測(予測の実行)
361
362       # 156. モデルの学習(学習の実行)
363
364       # 157. モデルの評価(評価の実行)
365
366       # 158. モデルの予測(予測の実行)
367
368       # 159. モデルの学習(学習の実行)
369
370       # 160. モデルの評価(評価の実行)
371
372       # 161. モデルの予測(予測の実行)
373
374       # 162. モデルの学習(学習の実行)
375
376       # 163. モデルの評価(評価の実行)
377
378       # 164. モデルの予測(予測の実行)
379
380       # 165. モデルの学習(学習の実行)
381
382       # 166. モデルの評価(評価の実行)
383
384       # 167. モデルの予測(予測の実行)
385
386       # 168. モデルの学習(学習の実行)
387
388       # 169. モデルの評価(評価の実行)
389
390       # 170. モデルの予測(予測の実行)
391
392       # 171. モデルの学習(学習の実行)
393
394       # 172. モデルの評価(評価の実行)
395
396       # 173. モデルの予測(予測の実行)
397
398       # 174. モデルの学習(学習の実行)
399
400       # 175. モデルの評価(評価の実行)
401
402       # 176. モデルの予測(予測の実行)
403
404       # 177. モデルの学習(学習の実行)
405
406       # 178. モデルの評価(評価の実行)
407
408       # 179. モデルの予測(予測の実行)
409
410       # 180. モデルの学習(学習の実行)
411
412       # 181. モデルの評価(評価の実行)
413
414       # 182. モデルの予測(予測の実行)
415
416       # 183. モデルの学習(学習の実行)
417
418       # 184. モデルの評価(評価の実行)
419
420       # 185. モデルの予測(予測の実行)
421
422       # 186. モデルの学習(学習の実行)
423
424       # 187. モデルの評価(評価の実行)
425
426       # 188. モデルの予測(予測の実行)
427
428       # 189. モデルの学習(学習の実行)
429
430       # 190. モデルの評価(評価の実行)
431
432       # 191. モデルの予測(予測の実行)
433
434       # 192. モデルの学習(学習の実行)
435
436       # 193. モデルの評価(評価の実行)
437
438       # 194. モデルの予測(予測の実行)
439
440       # 195. モデルの学習(学習の実行)
441
442       # 196. モデルの評価(評価の実行)
443
444       # 197. モデルの予測(予測の実行)
445
446       # 198. モデルの学習(学習の実行)
447
448       # 199. モデルの評価(評価の実行)
449
450       # 200. モデルの予測(予測の実行)
451
452       # 201. モデルの学習(学習の実行)
453
454       # 202. モデルの評価(評価の実行)
455
456       # 203. モデルの予測(予測の実行)
457
458       # 204. モデルの学習(学習の実行)
459
460       # 205. モデルの評価(評価の実行)
461
462       # 206. モデルの予測(予測の実行)
463
464       # 207. モデルの学習(学習の実行)
465
466       # 208. モデルの評価(評価の実行)
467
468       # 209. モデルの予測(予測の実行)
469
470       # 210. モデルの学習(学習の実行)
471
472       # 211. モデルの評価(評価の実行)
473
474       # 212. モデルの予測(予測の実行)
475
476       # 213. モデルの学習(学習の実行)
477
478       # 214. モデルの評価(評価の実行)
479
480       # 215. モデルの予測(予測の実行)
481
482       # 216. モデルの学習(学習の実行)
483
484       # 217. モデルの評価(評価の実行)
485
486       # 218. モデルの予測(予測の実行)
487
488       # 219. モデルの学習(学習の実行)
489
490       # 220. モデルの評価(評価の実行)
491
492       # 221. モデルの予測(予測の実行)
493
494       # 222. モデルの学習(学習の実行)
495
496       # 223. モデルの評価(評価の実行)
497
498       # 224. モデルの予測(予測の実行)
499
500       # 225. モデルの学習(学習の実行)
501
502       # 226. モデルの評価(評価の実行)
503
504       # 227. モデルの予測(予測の実行)
505
506       # 228. モデルの学習(学習の実行)
507
508       # 229. モデルの評価(評価の実行)
509
510       # 230. モデルの予測(予測の実行)
511
512       # 231. モデルの学習(学習の実行)
513
514       # 232. モデルの評価(評価の実行)
515
516       # 233. モデルの予測(予測の実行)
517
518       # 234. モデルの学習(学習の実行)
519
520       # 235. モデルの評価(評価の実行)
521
522       # 236. モデルの予測(予測の実行)
523
524       # 237. モデルの学習(学習の実行)
525
526       # 238. モデルの評価(評価の実行)
527
528       # 239. モデルの予測(予測の実行)
529
530       # 240. モデルの学習(学習の実行)
531
532       # 241. モデルの評価(評価の実行)
533
534       # 242. モデルの予測(予測の実行)
535
536       # 243. モデルの学習(学習の実行)
537
538       # 244. モデルの評価(評価の実行)
539
540       # 245. モデルの予測(予測の実行)
541
542       # 246. モデルの学習(学習の実行)
543
544       # 247. モデルの評価(評価の実行)
545
546       # 248. モデルの予測(予測の実行)
547
548       # 249. モデルの学習(学習の実行)
549
550       # 250. モデルの評価(評価の実行)
551
552       # 251. モデルの予測(予測の実行)
553
554       # 252. モデルの学習(学習の実行)
555
556       # 253. モデルの評価(評価の実行)
557
558       # 254. モデルの予測(予測の実行)
559
560       # 255. モデルの学習(学習の実行)
561
562       # 256. モデルの評価(評価の実行)
563
564       # 257. モデルの予測(予測の実行)
565
566       # 258. モデルの学習(学習の実行)
567
568       # 259. モデルの評価(評価の実行)
569
570       # 260. モデルの予測(予測の実行)
571
572       # 261. モデルの学習(学習の実行)
573
574       # 262. モデルの評価(評価の実行)
575
576       # 263. モデルの予測(予測の実行)
577
578       # 264. モデルの学習(学習の実行)
579
580       # 265. モデルの評価(評価の実行)
581
582       # 266. モデルの予測(予測の実行)
583
584       # 267. モデルの学習(学習の実行)
585
586       # 268. モデルの評価(評価の実行)
587
588       # 269. モデルの予測(予測の実行)
589
590       # 270. モデルの学習(学習の実行)
591
592       # 271. モデルの評価(評価の実行)
593
594       # 272. モデルの予測(予測の実行)
595
596       # 273. モデルの学習(学習の実行)
597
598       # 274. モデルの評価(評価の実行)
599
600       # 275. モデルの予測(予測の実行)
601
602       # 276. モデルの学習(学習の実行)
603
604       # 277. モデルの評価(評価の実行)
605
606       # 278. モデルの予測(予測の実行)
607
608       # 279. モデルの学習(学習の実行)
609
610       # 280. モデルの評価(評価の実行)
611
612       # 281. モデルの予測(予測の実行)
613
614       # 282. モデルの学習(学習の実行)
615
616       # 283. モデルの評価(評価の実行)
617
618       # 284. モデルの予測(予測の実行)
619
620       # 285. モデルの学習(学習の実行)
621
622       # 286. モデルの評価(評価の実行)
623
624       # 287. モデルの予測(予測の実行)
625
626       # 288. モデルの学習(学習の実行)
627
628       # 289. モデルの評価(評価の実行)
629
630       # 290. モデルの予測(予測の実行)
631
632       # 291. モデルの学習(学習の実行)
633
634       # 292. モデルの評価(評価の実行)
635
636       # 293. モデルの予測(予測の実行)
637
638       # 294. モデルの学習(学習の実行)
639
640       # 295. モデルの評価(評価の実行)
641
642       # 296. モデルの予測(予測の実行)
643
644       # 297. モデルの学習(学習の実行)
645
646       # 298. モデルの評価(評価の実行)
647
648       # 299. モデルの予測(予測の実行)
649
650       # 300. モデルの学習(学習の実行)
651
652       # 301. モデルの評価(評価の実行)
653
654       # 302. モデルの予測(予測の実行)
655
656       # 303. モデルの学習(学習の実行)
657
658       # 304. モデルの評価(評価の実行)
659
660       # 305. モデルの予測(予測の実行)
661
662       # 306. モデルの学習(学習の実行)
663
664       # 307. モデルの評価(評価の実行)
665
666       # 308. モデルの予測(予測の実行)
667
668       # 309. モデルの学習(学習の実行)
669
670       # 310. モデルの評価(評価の実行)
671
672       # 311. モデルの予測(予測の実行)
673
674       # 312. モデルの学習(学習の実行)
675
676       # 313. モデルの評価(評価の実行)
677
678       # 314. モデルの予測(予測の実行)
679
680       # 315. モデルの学習(学習の実行)
681
682       # 316. モデルの評価(評価の実行)
683
684       # 317. モデルの予測(予測の実行)
685
686       # 318. モデルの学習(学習の実行)
687
688       # 319. モデルの評価(評価の実行)
689
690       # 320. モデルの予測(予測の実行)
691
692       # 321. モデルの学習(学習の実行)
693
694       # 322. モデルの評価(評価の実行)
695
696       # 323. モデルの予測(予測の実行)
697
698       # 324. モデルの学習(学習の実行)
699
700       # 325. モデルの評価(評価の実行)
701
702       # 326. モデルの予測(予測の実行)
703
704       # 327. モデルの学習(学習の実行)
705
706       # 328. モデルの評価(評価の実行)
707
708       # 329. モデルの予測(予測の実行)
709
710       # 330. モデルの学習(学習の実行)
711
712       # 331. モデルの評価(評価の実行)
713
714       # 332. モデルの予測(予測の実行)
715
716       # 333. モデルの学習(学習の実行)
717
718       # 334. モデルの評価(評価の実行)
719
720       # 335. モデルの予測(予測の実行)
721
722       # 336. モデルの学習(学習の実行)
723
724       # 337. モデルの評価(評価の実行)
725
726       # 338. モデルの予測(予測の実行)
727
728       # 339. モデルの学習(学習の実行)
729
730       # 340. モデルの評価(評価の実行)
731
732       # 341. モデルの予測(予測の実行)
733
734       # 342. モデルの学習(学習の実行)
735
736       # 343. モデルの評価(評価の実行)
737
738       # 344. モデルの予測(予測の実行)
739
740       # 345. モデルの学習(学習の実行)
741
742       # 346. モデルの評価(評価の実行)
743
744       # 347. モデルの予測(予測の実行)
745
746       # 348. モデルの学習(学習の実行)
747
748       # 349. モデルの評価(評価の実行)
749
750       # 350. モデルの予測(予測の実行)
751
752       # 351. モデルの学習(学習の実行)
753
754       # 352. モデルの評価(評価の実行)
755
756       # 353. モデルの予測(予測の実行)
757
758       # 354. モデルの学習(学習の実行)
759
760       # 355. モデルの評価(評価の実行)
761
762       # 356. モデルの予測(予測の実行)
763
764       # 357. モデルの学習(学習の実行)
765
766       # 358. モデルの評価(評価の実行)
767
768       # 359. モデルの予測(予測の実行)
769
770       # 360. モデルの学習(学習の実行)
771
772       # 361. モデルの評価(評価の実行)
773
774       # 362. モデルの予測(予測の実行)
775
776       # 363. モデルの学習(学習の実行)
777
778       # 364. モデルの評価(評価の実行)
779
780       # 365. モデルの予測(予測の実行)
781
782       # 366. モデルの学習(学習の実行)
783
784       # 367. モデルの評価(評価の実行)
785
786       # 368. モデルの予測(予測の実行)
787
788       # 369. モデルの学習(学習の実行)
789
790       # 370. モデルの評価(評価の実行)
791
792       # 371. モデルの予測(予測の実行)
793
794       # 372. モデルの学習(学習の実行)
795
796       # 373. モデルの評価(評価の実行)
797
798       # 374. モデルの予測(予測の実行)
799
800       # 375. モデルの学習(学習の実行)
801
802       # 376. モデルの評価(評価の実行)
803
804       # 377. モデルの予測(予測の実行)
805
806       # 378. モデルの学習(学習の実行)
807
808       # 379. モデルの評価(評価の実行)
809
810       # 380. モデルの予測(予測
```



The screenshot shows the Google Colab interface with a Python script named 'predict\_word.py'. The code implements a neural network for word prediction. It includes functions for data loading, model training, and prediction. A detailed docstring at the top explains the code's purpose and usage.

```
# coding: utf-8

# predict_word.py

# ファイル、編集、挿入、ラストタイム、ヘルプ
# <-- コード > テキスト

# 100 def train_step(model, inputs, labels, initial_state):
# 101     # ここで Adam の学習率を定めることで、Adamax, Adadelta などの他の最適化手法を適用することができる
# 102     # 例) tf.train.AdamOptimizer(model.learning_rate * learning_rate_decay)
# 103     # 104
# 105     # 106     def loss_fn():
# 107         # 108         # ここで loss の計算を定義する
# 109         # 110         # 111         # 112         # 113         # 114         # 115         # 116         # 117         # 118         # 119         # 120         # 121         # 122         # 123         # 124         # 125         # 126         # 127         # 128         # 129         # 130         # 131         # 132         # 133         # 134         # 135         # 136         # 137         # 138         # 139         # 140         # 141         # 142         # 143         # 144         # 145         # 146         # 147         # 148         # 149         # 150         # 151         # 152         # 153         # 154         # 155         # 156         # 157         # 158         # 159         # 160         # 161         # 162         # 163         # 164         # 165         # 166         # 167         # 168         # 169         # 170         # 171         # 172         # 173         # 174         # 175         # 176         # 177         # 178         # 179         # 180         # 181         # 182         # 183         # 184         # 185         # 186         # 187         # 188         # 189         # 190         # 191         # 192         # 193         # 194         # 195         # 196         # 197         # 198         # 199         # 200         # 201         # 202         # 203         # 204         # 205         # 206         # 207         # 208         # 209         # 210         # 211         # 212         # 213         # 214         # 215         # 216         # 217         # 218         # 219         # 220         # 221         # 222         # 223         # 224         # 225         # 226         # 227         # 228         # 229         # 230         # 231         # 232         # 233         # 234         # 235         # 236         # 237         # 238         # 239         # 240         # 241         # 242         # 243         # 244         # 245         # 246         # 247         # 248         # 249         # 250         # 251         # 252         # 253         # 254         # 255         # 256         # 257         # 258         # 259         # 260         # 261         # 262         # 263         # 264         # 265         # 266         # 267         # 268         # 269         # 270         # 271         # 272         # 273         # 274         # 275         # 276         # 277         # 278         # 279         # 280         # 281         # 282         # 283         # 284         # 285         # 286         # 287         # 288         # 289         # 290         # 291         # 292         # 293         # 294         # 295         # 296         # 297         # 298         # 299         # 300         # 301         # 302         # 303         # 304         # 305         # 306         # 307         # 308         # 309         # 310         # 311         # 312         # 313         # 314         # 315         # 316         # 317         # 318         # 319         # 320         # 321         # 322         # 323         # 324         # 325         # 326         # 327         # 328         # 329         # 330         # 331         # 332         # 333         # 334         # 335         # 336         # 337         # 338         # 339         # 340         # 341         # 342         # 343         # 344         # 345         # 346         # 347         # 348         # 349         # 350         # 351         # 352         # 353         # 354         # 355         # 356         # 357         # 358         # 359         # 360         # 361         # 362         # 363         # 364         # 365         # 366         # 367         # 368         # 369         # 370         # 371         # 372         # 373         # 374         # 375         # 376         # 377         # 378         # 379         # 380         # 381         # 382         # 383         # 384         # 385         # 386         # 387         # 388         # 389         # 390         # 391         # 392         # 393         # 394         # 395         # 396         # 397         # 398         # 399         # 400         # 401         # 402         # 403         # 404         # 405         # 406         # 407         # 408         # 409         # 410         # 411         # 412         # 413         # 414         # 415         # 416         # 417         # 418         # 419         # 420         # 421         # 422         # 423         # 424         # 425         # 426         # 427         # 428         # 429         # 430         # 431         # 432         # 433         # 434         # 435         # 436         # 437         # 438         # 439         # 440         # 441         # 442         # 443         # 444         # 445         # 446         # 447         # 448         # 449         # 450         # 451         # 452         # 453         # 454         # 455         # 456         # 457         # 458         # 459         # 460         # 461         # 462         # 463         # 464         # 465         # 466         # 467         # 468         # 469         # 470         # 471         # 472         # 473         # 474         # 475         # 476         # 477         # 478         # 479         # 480         # 481         # 482         # 483         # 484         # 485         # 486         # 487         # 488         # 489         # 490         # 491         # 492         # 493         # 494         # 495         # 496         # 497         # 498         # 499         # 500         # 501         # 502         # 503         # 504         # 505         # 506         # 507         # 508         # 509         # 510         # 511         # 512         # 513         # 514         # 515         # 516         # 517         # 518         # 519         # 520         # 521         # 522         # 523         # 524         # 525         # 526         # 527         # 528         # 529         # 530         # 531         # 532         # 533         # 534         # 535         # 536         # 537         # 538         # 539         # 540         # 541         # 542         # 543         # 544         # 545         # 546         # 547         # 548         # 549         # 550         # 551         # 552         # 553         # 554         # 555         # 556         # 557         # 558         # 559         # 5510         # 5511         # 5512         # 5513         # 5514         # 5515         # 5516         # 5517         # 5518         # 5519         # 5520         # 5521         # 5522         # 5523         # 5524         # 5525         # 5526         # 5527         # 5528         # 5529         # 5530         # 5531         # 5532         # 5533         # 5534         # 5535         # 5536         # 5537         # 5538         # 5539         # 5540         # 5541         # 5542         # 5543         # 5544         # 5545         # 5546         # 5547         # 5548         # 5549         # 5550         # 5551         # 5552         # 5553         # 5554         # 5555         # 5556         # 5557         # 5558         # 5559         # 55510         # 55511         # 55512         # 55513         # 55514         # 55515         # 55516         # 55517         # 55518         # 55519         # 55520         # 55521         # 55522         # 55523         # 55524         # 55525         # 55526         # 55527         # 55528         # 55529         # 55530         # 55531         # 55532         # 55533         # 55534         # 55535         # 55536         # 55537         # 55538         # 55539         # 55540         # 55541         # 55542         # 55543         # 55544         # 55545         # 55546         # 55547         # 55548         # 55549         # 55550         # 55551         # 55552         # 55553         # 55554         # 55555         # 55556         # 55557         # 55558         # 55559         # 55560         # 55561         # 55562         # 55563         # 55564         # 55565         # 55566         # 55567         # 55568         # 55569         # 55570         # 55571         # 55572         # 55573         # 55574         # 55575         # 55576         # 55577         # 55578         # 55579         # 55580         # 55581         # 55582         # 55583         # 55584         # 55585         # 55586         # 55587         # 55588         # 55589         # 55590         # 55591         # 55592         # 55593         # 55594         # 55595         # 55596         # 55597         # 55598         # 55599         # 555100         # 555101         # 555102         # 555103         # 555104         # 555105         # 555106         # 555107         # 555108         # 555109         # 555110         # 555111         # 555112         # 555113         # 555114         # 555115         # 555116         # 555117         # 555118         # 555119         # 555120         # 555121         # 555122         # 555123         # 555124         # 555125         # 555126         # 555127         # 555128         # 555129         # 555130         # 555131         # 555132         # 555133         # 555134         # 555135         # 555136         # 555137         # 555138         # 555139         # 555140         # 555141         # 555142         # 555143         # 555144         # 555145         # 555146         # 555147         # 555148         # 555149         # 555150         # 555151         # 555152         # 555153         # 555154         # 555155         # 555156         # 555157         # 555158         # 555159         # 555160         # 555161         # 555162         # 555163         # 555164         # 555165         # 555166         # 555167         # 555168         # 555169         # 555170         # 555171         # 555172         # 555173         # 555174         # 555175         # 555176         # 555177         # 555178         # 555179         # 555180         # 555181         # 555182         # 555183         # 555184         # 555185         # 555186         # 555187         # 555188         # 555189         # 555190         # 555191         # 555192         # 555193         # 555194         # 555195         # 555196         # 555197         # 555198         # 555199         # 555200         # 555201         # 555202         # 555203         # 555204         # 555205         # 555206         # 555207         # 555208         # 555209         # 555210         # 555211         # 555212         # 555213         # 555214         # 555215         # 555216         # 555217         # 555218         # 555219         # 555220         # 555221         # 555222         # 555223         # 555224         # 555225         # 555226         # 555227         # 555228         # 555229         # 555230         # 555231         # 555232         # 555233         # 555234         # 555235         # 555236         # 555237         # 555238         # 555239         # 555240         # 555241         # 555242         # 555243         # 555244         # 555245         # 555246         # 555247         # 555248         # 555249         # 555250         # 555251         # 555252         # 555253         # 555254         # 555255         # 555256         # 555257         # 555258         # 555259         # 555260         # 555261         # 555262         # 555263         # 555264         # 555265         # 555266         # 555267         # 555268         # 555269         # 555270         # 555271         # 555272         # 555273         # 555274         # 555275         # 555276         # 555277         # 555278         # 555279         # 555280         # 555281         # 555282         # 555283         # 555284         # 555285         # 555286         # 555287         # 555288         # 555289         # 555290         # 555291         # 555292         # 555293         # 555294         # 555295         # 555296         # 555297         # 555298         # 555299         # 555300         # 555301         # 555302         # 555303         # 555304         # 555305         # 555306         # 555307         # 555308         # 555309         # 555310         # 555311         # 555312         # 555313         # 555314         # 555315         # 555316         # 555317         # 555318         # 555319         # 555320         # 555321         # 555322         # 555323         # 555324         # 555325         # 555326         # 555327         # 555328         # 555329         # 555330         # 555331         # 555332         # 555333         # 555334         # 555335         # 555336         # 555337         # 555338         # 555339         # 555340         # 555341         # 555342         # 555343         # 555344         # 555345         # 555346         # 555347         # 555348         # 555349         # 555350         # 555351         # 555352         # 555353         # 555354         # 555355         # 555356         # 555357         # 555358         # 555359         # 555360         # 555361         # 555362         # 555363         # 555364         # 555365         # 555366         # 555367         # 555368         # 555369         # 555370         # 555371         # 555372         # 555373         # 555374         # 555375         # 555376         # 555377         # 555378         # 555379         # 555380         # 555381         # 555382         # 555383         # 555384         # 555385         # 555386         # 555387         # 555388         # 555389         # 555390         # 555391         # 555392         # 555393         # 555394         # 555395         # 555396         # 555397         # 555398         # 555399         # 555400         # 555401         # 555402         # 555403         # 555404         # 555405         # 555406         # 555407         # 555408         # 555409         # 555410         # 555411         # 555412         # 555413         # 555414         # 555415         # 555416         # 555417         # 555418         # 555419         # 555420         # 555421         # 555422         # 555423         # 555424         # 555425         # 555426         # 555427         # 555428         # 555429         # 555430         # 555431         # 555432         # 555433         # 555434         # 555435         # 555436         # 555437         # 555438         # 555439         # 555440         # 555441         # 555442         # 555443         # 555444         # 555445         # 555446         # 555447         # 555448         # 555449         # 555450         # 555451         # 555452         # 555453         # 555454         # 555455         # 555456         # 555457         # 555458         # 555459         # 555460         # 555461         # 555462         # 555463         # 555464         # 555465         # 555466         # 555467         # 555468         # 555469         # 555470         # 555471         # 555472         # 555473         # 555474         # 555475         # 555476         # 555477         # 555478         # 555479         # 555480         # 555481         # 555482         # 555483         # 555484         # 555485         # 555486         # 555487         # 555488         # 555489         # 555490         # 555491         # 555492         # 555493         # 555494         # 555495         # 555496         # 555497         # 555498         # 555499         # 555500         # 555501         # 555502         # 555503         # 555504         # 555505         # 555506         # 555507         # 555508         # 555509         # 555510         # 555511         # 555512         # 555513         # 555514         # 555515         # 555516         # 555517         # 555518         # 555519         # 555520         # 555521         # 555522         # 555523         # 555524         # 555525         # 555526         # 555527         # 555528         # 555529         # 555530         # 555531         # 555532         # 555533         # 555534         # 555535         # 555536         # 555537         # 555538         # 555539         # 555540         # 555541         # 555542         # 555543         # 555544         # 555545         # 555546         # 555547         # 555548         # 555549         # 555550         # 555551         # 555552         # 555553         # 555554         # 555555         # 555556         # 555557         # 555558         # 555559         # 555560         # 555561         # 555562         # 555563         # 555564         # 555565         # 555566         # 555567         # 555568         # 555569         # 555570         # 555571         # 555572         # 555573         # 555574         # 555575         # 555576         # 555577         # 555578         # 555579         # 555580         # 555581         # 555582         # 555583         # 555584         # 555585         # 555586         # 555587         # 555588         # 555589         # 555590         # 555591         # 555592         # 555593         # 555594         # 555595         # 555596         # 555597         # 555598         # 555599         # 5555100         # 5555101         # 5555102         # 5555103         # 5555104         # 5555105         # 5555106         # 5555107         # 5555108         # 5555109         # 5555110         # 5555111         # 5555112         # 5555113         # 5555114         # 5555115         # 5555116         # 5555117         # 5555118         # 5555119         # 5555120         # 5555121         # 5555122         # 5555123         # 5555124         # 5555125         # 5555126         # 5555127         # 5555128         # 5555129         # 5555130         # 5555131         # 5555132         # 5555133         # 5555134         # 5555135         # 5555136         # 5555137         # 5555138         # 5555139         # 5555140         # 5555141         # 5555142         # 5555143         # 5555144         # 5555145         # 5555146         # 5555147         # 5555148         # 5555149         # 5555150         # 5555151         # 5555152         # 5555153         # 5555154         # 5555155         # 5555156         # 5555157         # 5555158         # 5555159         # 5555160         # 5555161         # 5555162         # 5555163         # 5555164         # 5555165         # 5555166         # 5555167         # 5555168         # 5555169         # 5555170         # 5555171         # 5555172         # 5555173         # 5555174         # 5555175         # 5555176         # 5555177         # 5555178         # 5555179         # 5555180         # 5555181         # 5555182         # 5555183         # 5555184         # 5555185         # 5555186         # 5555187         # 5555188         # 5555189         # 5555190         # 5555191         # 5555192         # 5555193         # 5555194         # 5555195         # 5555196         # 5555197         # 5555198         # 5555199         # 5555200         # 5555201         # 5555202         # 5555203         # 5555204         # 5555205         # 5555206         # 5555207         # 5555208         # 5555209         # 5555210         # 5555211         # 5555212         # 5555213         # 5555214         # 5555215         # 5555216         # 5555217         # 5555218         # 5555219         # 5555220         # 5555221         # 5555222         # 5555223         # 5555224         # 5555225         # 5555226         # 5555227         # 5555228         # 5555229         # 5555230         # 5555231         # 5555232         # 5555233         # 5555234         # 5555235         # 5555236         # 5555237         # 5555238         # 5555239         # 5555240         # 5555241         # 5555242         # 5555243         # 5555244         # 5555245         # 5555246         # 5555247         # 5555248         # 5555249         # 5555250         # 5555251         # 5555252         # 5555253         # 5555254         # 5555255         # 5555256         # 5555257         # 5555258         # 5555259         # 5555260         # 5555261         # 5555262         # 5555263         # 5555264         # 5555265         # 5555266         # 5555267         # 5555268         # 5555269         # 5555270         # 5555271         # 5555272         # 5555273         # 5555274         # 5555275         # 5555276         # 5555277         # 5555278         # 5555279         # 5555280         # 5555281         # 5555282         # 5555283         # 5555284         # 5555285         # 5555286         # 5555287         # 5555288         # 5555289         # 5555290         # 5555291         # 5555292         # 5555293         # 5555294         # 5555295         # 5555296         # 5555297         # 5555298         # 5555299         # 5555300         # 5555301         # 5555302         # 5555303         # 5555304         # 5555305         # 5555306         # 5555307         # 5555308         # 5555309         # 5555310         # 5555311         # 5555312         # 5555313         # 5555314         # 5555315         # 5555316         # 5555317         # 5555318         # 5555319         # 5555320         # 5555321         # 5555322         # 5555323         # 5555324         # 5555325         # 5555326         # 5555327         # 5555328         # 5555329         # 5555330         # 5555331         # 5555332         # 5555333         # 5555334         # 5555335         # 5555336         # 5555337         # 5555338         # 5555339         # 5555340         # 5555341         # 5555342         # 5555343         # 5555344         # 5555345         # 5555346         # 5555347         # 5555348         # 5555349         # 5555350         # 5555351         # 5555352         # 5555353         # 5555354         # 5555355         # 5555356         # 5555357         # 5555358         # 5555359         # 5555360         # 5555361         # 5555362         # 5555363         # 5555364         # 5555365         # 5555366         # 5555367         # 5555368         # 5555369         # 5555370         # 5555371         # 5555372         # 5555373         # 5555374         # 5555375         # 5555376         # 5555377         # 5555378         # 5555379         # 5555380         # 5555381         # 5555382         # 5555383         # 5555384         # 5555385         # 5555386         # 5555387         # 5555388         # 5555389         # 5555390         # 5555391         # 5555392         # 5555393         # 5555394         # 5555395         # 5555396         # 5555397         # 5555398         # 5555399         # 5555400         # 5555401         # 5555402         # 5555403         # 5555404         # 5555405         # 5555406         # 5555407         # 5555408         # 5555409         # 5555410         # 5555411         # 5555412         # 5555413         # 5555414         # 5555415         # 5555416         # 5555417         # 5555418         # 5555419         # 5555420         # 5555421         # 5555422         # 5555423         # 5555424         # 5555425         # 5555426         # 5555427         # 5555428         # 5555429         # 5555430         # 5555431         # 5555432         # 5555433         # 5555434         # 5555435         # 5555436         # 5555437         # 5555438         # 5555439         # 5555440         # 5555441         # 5555442         # 5555443         # 5555444         # 5555445         # 5555446         # 5555447         # 5555448         # 5555449         # 5555450         # 5555451         # 5555452         # 5555453         # 5555454         # 5555455         # 5555456         # 5555457         # 5555458         # 5555459         # 5555460         # 5555461         # 5555462         # 5555463         # 5555464         # 5555465         # 5555466         # 5555467         # 5555468         # 5555469         # 5555470         # 5555471         # 5555472         # 5555473         # 5555474         # 5555475         # 5555476         # 5555477         # 5555478         # 5555479         # 5555480
```



## 4 3\_4\_spoken\_digit

```
# 3_4_spoken_digits.ipynb ☆
# ファイル ノートブック 表示 ヘルプ ヘルプ [エラーの表示を停止しました]
# コメント リポート ダッシュボード
# フォルダ + ドライブ
# 検索 [x]
# ハードウェア
# spoken_digitsデータセットを例にした再帰型ニューラルネットワークのいろいろ
# spoken_digitsデータセットを例にした再帰型ニューラルネットワークの構築方法を学びます。
# そのため、このコードを実行するためには必要なモジュールをインストール・インポートします。
# ! pip install pydub
# ! git clone https://github.com/eduardo-fernandes/pydub.git
# ! cd pydub
# ! python setup.py install
# ! rm -rf pydub
# ! pip install librosa
# ! pip install numpy
# ! pip install tensorflow
# ! pip install librosa
# ! pip install pydub
# ! pip install scikit-learn
# ! pip install keras
# ! pip install tensorflow

# データセットの読み込みと中の確認

spoken_digits データセットは, emit の重複です。データの特徴は,


- 音声が約10秒間の音波の波形が記録されています。
- 各音波は、各音波の音を表すラベルで分類されています。
- 人の歌が、各自を自分で読み上げています。
- 各自のファイルには、対応する音を記述しています。



データセットの読み込みは, assertIsDatasetで行っています。assertIsDatasetは手始めに確認の段階を行なうデータを手軽にグンフロードデータを使えるようにできるライブラリです。

ライブラリの機能で、ダウンロードしたデータセットを、シャッフルすることや、分割することができます。
ここでは、データセット全体の,


- 70%トレーニング用
- 15%検証用
- 15%テスト用

```



Q 3\_4\_spoken\_digitz.ipynb

PRO ファイル 表示 リンク ライブセル ヘルプ プラットフォームの画面を保存しました  
+ コード テキスト

Q いろいろな再構型ニューラルネットワーク

(a) データが複数あるときに、様々なモデルデータに対して適用できることを確認しましょう。ここでは、TensorFlowのKeras APIを使って、簡単なモデルを開発しています。

✓ 小規模な畳み込みネットワーク

最初のセルでは、次のように畳み込み層を用います。このセルでは1層の畳み込み層とMaxPooling層を持ちます。

このセルの実行結果は、4枚並んで、2枚のフルカートンで検出を指出されます。

また、畠み込み層からの出力層は、完全接続層であります。GlobalAveragePoolingを行っています。この層は、Flatten層と同様に、零数のフィルターで戻すデータを平均化し、全結合層への接続を可能にします。

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical

(x_train, y_train), (x_test, y_test) = cifar10.load_data()
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)

model_1 = Sequential([
    Conv2D(16, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    MaxPooling2D((2, 2), strides=2, activation='relu'),
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D((2, 2), strides=2, activation='relu'),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2), strides=2, activation='relu'),
    Flatten(),
    Dense(512, activation='relu'),
    Dense(10, activation='softmax')
])

model_1.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model_1.summary()
```



```
3_4_spoken_digit.ipynb
```

GRUを使用したモデル

RNN(Recurrent Neural Network)は時間とともに計算量を増すために工夫されたGNNを使ってモデルを定義します。単純RNNのときたく同時に、直後の結合を考慮する。

```
1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import ...
4
5
6 # tf.keras.Sequentialクラスを継承
7 class RNNModel(tf.keras.Model):
8     def __init__(self, vocab_size=10, embedding_dim=16, hidden_dim=32, output_dim=10):
9         super().__init__()
10        self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
11        self.rnn = tf.keras.layers.SimpleRNN(hidden_dim, return_sequences=True)
12        self.dense = tf.keras.layers.Dense(output_dim, activation='softmax')
13
14    def call(self, inputs):
15        x = self.embedding(inputs)
16        x = self.rnn(x)
17        x = self.dense(x)
18
19        return x
20
21 model = RNNModel()
22
23 dataset = tf.data.Dataset.from_tensor_slices((train_x, train_y))
24 dataset = dataset.shuffle(len(train_x)).batch(32).prefetch(tf.data.AUTOTUNE)
25
26 epochs = 10
27
28 optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
29 loss_fn = tf.keras.losses.SparseCategoricalCrossentropy()
30 accuracy_fn = tf.keras.metrics.Accuracy()
31
32
33 @tf.function
34 def train_step(model, inputs, targets, optimizer, loss_fn, accuracy_fn):
35     with tf.GradientTape() as tape:
36         predictions = model(inputs)
37         loss = loss_fn(targets, predictions)
38
39         accuracy_fn.update_state(targets, predictions)
40
41     tape.gradient(loss, model.trainable_variables)
42     optimizer.apply_gradients(zip(tape.gradient(loss, model.trainable_variables), model.trainable_variables))
43
44     return loss, accuracy_fn.result()
45
46
47 @tf.function
48 def test_step(model, inputs, targets, accuracy_fn):
49     predictions = model(inputs)
50     accuracy_fn.update_state(targets, predictions)
51
52
53 @tf.function
54 def evaluate(model, dataset, accuracy_fn):
55     accuracy_fn.reset_states()
56
57     for inputs, targets in dataset:
58         accuracy_fn.update_state(targets, model(inputs))
59
60     return accuracy_fn.result()
61
62
63 epochs = 10
64
65 for epoch in range(epochs):
66     for step, (inputs, targets) in enumerate(dataset):
67         loss, accuracy = train_step(model, inputs, targets, optimizer, loss_fn, accuracy_fn)
68
69         if step % 100 == 0:
70             print(f'Epoch {epoch}, Step {step} - Loss: {loss} - Accuracy: {accuracy}')
71
72
73 test_accuracy = evaluate(model, test_dataset, accuracy_fn)
74
75 print(f'Test Accuracy: {test_accuracy}'
```

```
3_4_spoken_digit.ipynb
```

モデルの学習にかかる時間は計算量が少ないため、単純RNNに比べるとずっと短くなります。

```
1 (23) 1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import ...
4
5
6 # tf.keras.Sequentialクラスを継承
7 class SimpleRNNModel(tf.keras.Model):
8     def __init__(self, vocab_size=10, embedding_dim=16, hidden_dim=32, output_dim=10):
9         super().__init__()
10        self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
11        self.rnn = tf.keras.layers.SimpleRNN(hidden_dim, return_sequences=True)
12        self.dense = tf.keras.layers.Dense(output_dim, activation='softmax')
13
14    def call(self, inputs):
15        x = self.embedding(inputs)
16        x = self.rnn(x)
17        x = self.dense(x)
18
19        return x
20
21 model = SimpleRNNModel()
22
23 dataset = tf.data.Dataset.from_tensor_slices((train_x, train_y))
24 dataset = dataset.shuffle(len(train_x)).batch(32).prefetch(tf.data.AUTOTUNE)
25
26 epochs = 10
27
28 optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
29 loss_fn = tf.keras.losses.SparseCategoricalCrossentropy()
30 accuracy_fn = tf.keras.metrics.Accuracy()
31
32
33 @tf.function
34 def train_step(model, inputs, targets, optimizer, loss_fn, accuracy_fn):
35     with tf.GradientTape() as tape:
36         predictions = model(inputs)
37         loss = loss_fn(targets, predictions)
38
39         accuracy_fn.update_state(targets, predictions)
40
41     tape.gradient(loss, model.trainable_variables)
42     optimizer.apply_gradients(zip(tape.gradient(loss, model.trainable_variables), model.trainable_variables))
43
44     return loss, accuracy_fn.result()
45
46
47 @tf.function
48 def test_step(model, inputs, targets, accuracy_fn):
49     predictions = model(inputs)
50     accuracy_fn.update_state(targets, predictions)
51
52
53 @tf.function
54 def evaluate(model, dataset, accuracy_fn):
55     accuracy_fn.reset_states()
56
57     for inputs, targets in dataset:
58         accuracy_fn.update_state(targets, model(inputs))
59
60     return accuracy_fn.result()
61
62
63 epochs = 10
64
65 for epoch in range(epochs):
66     for step, (inputs, targets) in enumerate(dataset):
67         loss, accuracy = train_step(model, inputs, targets, optimizer, loss_fn, accuracy_fn)
68
69         if step % 100 == 0:
70             print(f'Epoch {epoch}, Step {step} - Loss: {loss} - Accuracy: {accuracy}')
71
72
73 test_accuracy = evaluate(model, test_dataset, accuracy_fn)
74
75 print(f'Test Accuracy: {test_accuracy}'
```

```
3_4_spoken_digit.ipynb
```

双方に向RNN(LSTM)を使用したモデル

最初に、双方に向LSTMを用いたモデルを紹介します。Kerasで双方に向接続されたネットワークを定義するためには、系列データに対するレイヤーを双向(Bidirectional)レイヤーで構成することで、定義することができます。

```
1 (23) 1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import ...
4
5
6 # tf.keras.Sequentialクラスを継承
7 class BidirectionalRNNModel(tf.keras.Model):
8     def __init__(self, vocab_size=10, embedding_dim=16, hidden_dim=32, output_dim=10):
9         super().__init__()
10        self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
11        self.rnn = tf.keras.layers.Bidirectional(tf.keras.layers.SimpleRNN(hidden_dim, return_sequences=True))
12        self.dense = tf.keras.layers.Dense(output_dim, activation='softmax')
13
14    def call(self, inputs):
15        x = self.embedding(inputs)
16        x = self.rnn(x)
17        x = self.dense(x)
18
19        return x
20
21 model = BidirectionalRNNModel()
22
23 dataset = tf.data.Dataset.from_tensor_slices((train_x, train_y))
24 dataset = dataset.shuffle(len(train_x)).batch(32).prefetch(tf.data.AUTOTUNE)
25
26 epochs = 10
27
28 optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
29 loss_fn = tf.keras.losses.SparseCategoricalCrossentropy()
30 accuracy_fn = tf.keras.metrics.Accuracy()
31
32
33 @tf.function
34 def train_step(model, inputs, targets, optimizer, loss_fn, accuracy_fn):
35     with tf.GradientTape() as tape:
36         predictions = model(inputs)
37         loss = loss_fn(targets, predictions)
38
39         accuracy_fn.update_state(targets, predictions)
40
41     tape.gradient(loss, model.trainable_variables)
42     optimizer.apply_gradients(zip(tape.gradient(loss, model.trainable_variables), model.trainable_variables))
43
44     return loss, accuracy_fn.result()
45
46
47 @tf.function
48 def test_step(model, inputs, targets, accuracy_fn):
49     predictions = model(inputs)
50     accuracy_fn.update_state(targets, predictions)
51
52
53 @tf.function
54 def evaluate(model, dataset, accuracy_fn):
55     accuracy_fn.reset_states()
56
57     for inputs, targets in dataset:
58         accuracy_fn.update_state(targets, model(inputs))
59
60     return accuracy_fn.result()
61
62
63 epochs = 10
64
65 for epoch in range(epochs):
66     for step, (inputs, targets) in enumerate(dataset):
67         loss, accuracy = train_step(model, inputs, targets, optimizer, loss_fn, accuracy_fn)
68
69         if step % 100 == 0:
70             print(f'Epoch {epoch}, Step {step} - Loss: {loss} - Accuracy: {accuracy}')
71
72
73 test_accuracy = evaluate(model, test_dataset, accuracy_fn)
74
75 print(f'Test Accuracy: {test_accuracy}'
```

3\_4\_spoken\_digital.ipynb

```
[4]: model = Sequential()
[5]: model.add(Dense(1))
[6]: dataset_train = tensorflow.data.Dataset.from_tensor_slices((x_train, y_train))
[7]: dataset_train = dataset_train.map(lambda x, y: (x, tf.one_hot(y, 10)))
[8]: dataset_train = dataset_train.shuffle(1000).batch(32)
[9]: dataset_train = dataset_train.prefetch(1)
[10]: dataset_val = tensorflow.data.Dataset.from_tensor_slices((x_val, y_val))
[11]: dataset_val = dataset_val.map(lambda x, y: (x, tf.one_hot(y, 10)))
[12]: dataset_val = dataset_val.shuffle(1000).batch(32)
[13]: dataset_val = dataset_val.prefetch(1)
[14]: dataset_val = dataset_val.cache()
[15]: val_accuraccy = []
[16]: for epoch in range(10):
[17]:   for step, (x_batch, y_batch) in enumerate(dataset_train):
[18]:     with tf.GradientTape() as tape:
[19]:       y_pred = model(x_batch)
[20]:       loss = tensorflow.reduce_mean(tensorflow.square(y_batch - y_pred))
[21]:     gradients = tape.gradient(loss, model.trainable_variables)
[22]:     optimizer.apply_gradients(zip(gradients, model.trainable_variables))
[23]:   if epoch % 1 == 0:
[24]:     val_loss, val_accuracy = validate(model, dataset_val)
[25]:     print('Epoch %d: loss %.4f, accuracy %.4f' % (epoch, loss, accuracy))
[26]:     val_accuraccy.append(val_accuracy)
[27]:     print('Validation loss: %.4f, validation accuracy: %.4f' % (val_loss, val_accuracy))
[28]:     print('Validation accuracy: %.4f' % val_accuracy)
[29]:     print('Validation accuracy: %.4f' % val_accuracy)
```

このノートブックでは、これまでのモデルと同様に、`compile()`を行い、`fit()`で学習データを渡し実行します。

Kerasを用いたモデルで、勾配クリッピングを行なう。

```
[3]: ! tensorboard --logdir ./logs/tensorboard --port 6006
```

この表示では、いろいろな要素を含む各手順が学習できるところまでを確認しています。経験的練習や学習の調整は行われていません。実際のモデル開発では、学習のためやモデルの特徴をより工夫することで、経験練習を向上させます。

3\_4\_spoken\_digital.ipynb

```
[4]: model = Sequential()
[5]: model.add(Dense(1))
[6]: dataset_train = tensorflow.data.Dataset.from_tensor_slices((x_train, y_train))
[7]: dataset_train = dataset_train.map(lambda x, y: (x, tf.one_hot(y, 10)))
[8]: dataset_train = dataset_train.shuffle(1000).batch(32)
[9]: dataset_train = dataset_train.prefetch(1)
[10]: dataset_val = tensorflow.data.Dataset.from_tensorlices((x_val, y_val))
[11]: dataset_val = dataset_val.map(lambda x, y: (x, tf.one_hot(y, 10)))
[12]: dataset_val = dataset_val.shuffle(1000).batch(32)
[13]: dataset_val = dataset_val.prefetch(1)
[14]: dataset_val = dataset_val.cache()
[15]: val_accuraccy = []
[16]: for epoch in range(10):
[17]:   for step, (x_batch, y_batch) in enumerate(dataset_train):
[18]:     with tf.GradientTape() as tape:
[19]:       y_pred = model(x_batch)
[20]:       loss = tensorflow.reduce_mean(tensorflow.square(y_batch - y_pred))
[21]:     gradients = tape.gradient(loss, model.trainable_variables)
[22]:     gradients, global_norm = tensorflow.clip_by_norm(gradients, 1.0, 1.0)
[23]:     optimizer.apply_gradients(zip(gradients, model.trainable_variables))
[24]:   if epoch % 1 == 0:
[25]:     val_loss, val_accuracy = validate(model, dataset_val)
[26]:     print('Epoch %d: loss %.4f, accuracy %.4f' % (epoch, loss, accuracy))
[27]:     val_accuraccy.append(val_accuracy)
[28]:     print('Validation loss: %.4f, validation accuracy: %.4f' % (val_loss, val_accuracy))
[29]:     print('Validation accuracy: %.4f' % val_accuracy)
[30]:     print('Validation accuracy: %.4f' % val_accuracy)
```

この表示では、いろいろな要素を含む各手順が学習できるところまでを確認しています。経験的練習や学習の調整は行われていません。実際のモデル開発では、学習のためやモデルの特徴をより工夫することで、経験練習を向上させます。

勾配クリッピング

Kerasを用いたモデルで、勾配クリッピングを行なう。

```
[3]: ! tensorboard --logdir ./logs/tensorboard --port 6006
```

この表示では、いろいろな要素を含む各手順が学習できるところまでを確認しています。経験的練習や学習の調整は行われていません。実際のモデル開発では、学習のためやモデルの特徴をより工夫することで、経験練習を向上させます。

3\_4\_spoken\_digital.ipynb

```
[4]: model = Sequential()
[5]: model.add(RNN(100, return_sequences=True))
[6]: model.add(Dropout(0.2))
[7]: model.add(RNN(100, return_sequences=False))
[8]: model.add(Dropout(0.2))
[9]: model.add(Dense(1))
[10]: dataset_train = tensorflow.data.Dataset.from_tensor_slices((x_train, y_train))
[11]: dataset_train = dataset_train.map(lambda x, y: (x, tf.one_hot(y, 10)))
[12]: dataset_train = dataset_train.shuffle(1000).batch(32)
[13]: dataset_train = dataset_train.prefetch(1)
[14]: dataset_val = tensorflow.data.Dataset.from_tensorlices((x_val, y_val))
[15]: dataset_val = dataset_val.map(lambda x, y: (x, tf.one_hot(y, 10)))
[16]: dataset_val = dataset_val.shuffle(1000).batch(32)
[17]: dataset_val = dataset_val.prefetch(1)
[18]: dataset_val = dataset_val.cache()
[19]: val_accuraccy = []
[20]: for epoch in range(10):
[21]:   for step, (x_batch, y_batch) in enumerate(dataset_train):
[22]:     with tf.GradientTape() as tape:
[23]:       y_pred = model(x_batch)
[24]:       loss = tensorflow.reduce_mean(tensorflow.square(y_batch - y_pred))
[25]:     gradients = tape.gradient(loss, model.trainable_variables)
[26]:     optimizer.apply_gradients(zip(gradients, model.trainable_variables))
[27]:   if epoch % 1 == 0:
[28]:     val_loss, val_accuracy = validate(model, dataset_val)
[29]:     print('Epoch %d: loss %.4f, accuracy %.4f' % (epoch, loss, accuracy))
[30]:     val_accuraccy.append(val_accuracy)
[31]:     print('Validation loss: %.4f, validation accuracy: %.4f' % (val_loss, val_accuracy))
[32]:     print('Validation accuracy: %.4f' % val_accuracy)
[33]:     print('Validation accuracy: %.4f' % val_accuracy)
```

このノートブックではRNNを用いたモデルでは、まったく学習ができていないため、工夫する必要があるが、時間がないため一旦廃止。

## 5 3\_5\_Seq2Seq(Encoder-Decoder)\_sin-cos

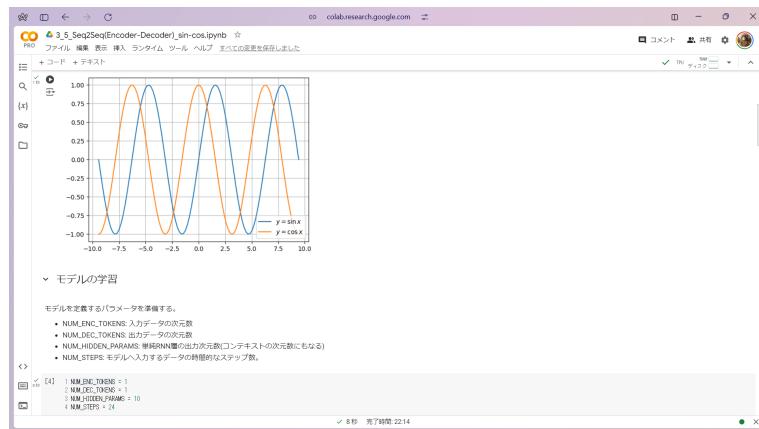
```

3_5_Seq2Seq(Encoder-Decoder)_sin-cos.ipynb
[1]: import tensorflow as tf
[2]: import numpy as np
[3]: import matplotlib.pyplot as plt

データの準備

ここでは、sin関数の値からcos関数の値をSeq2Seqモデルに予測させることを試みる。
入力データとして、seq_inと、seq_outを用意する。
xは、np.linspace(-10, 10, 100)で生成される。
seq_inは、np.sin(x)で生成される。
seq_outは、np.cos(x)で生成される。
sin関数の値が入力値、cos関数の値が推論したい値となる。
yは、sin(x)とcos(x)の値を格納する。
plt.plot(x, seq_in, label='sin(x)')
plt.plot(x, seq_out, label='cos(x)')
plt.legend()
plt.show()

```



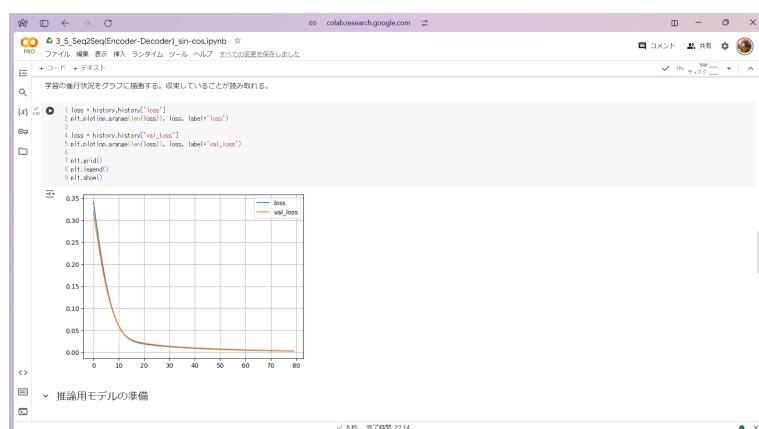
```

3_5_Seq2Seq(Encoder-Decoder)_sin-cos.ipynb
モデルを定義する(パラメータを準備する)。
• NUM_ENC_TOKENS: 入力データの次元数
• NUM_DEC_TOKENS: 出力データの次元数
• NUM_HIDDEN_PARAMS: モデルの隠れ層の出力次元数(コンテキストの次元数にもなる)
• NUM_STEPS: ディテール化するデータの時間的なステップ数。
[5]: tf.keras.backend.clear_session()

e_inout = tf.keras.layers.Input(shape=(NUM_STEPS, NUM_ENC_TOKENS), name="e_inout")
e_out = tf.keras.layers.SpatialDropout1D(0.999, return_sequences=True, name="e_out")
d_inout = tf.keras.layers.Input(shape=(NUM_STEPS, NUM_DEC_TOKENS), name="d_inout")
d_out = tf.keras.layers.SpatialDropout1D(0.999, return_sequences=True, name="d_out")
d_inout_0 = tf.keras.layers.Input(shape=(1, NUM_DEC_TOKENS), initial_state=True, name="d_inout_0")
d_out_0 = tf.keras.layers.Dense(NUM_DEC_TOKENS, activation="linear", name="d_out_0")
d_out_0 = tf.keras.layers.Dropout(0.999, name="d_out_0")

model_train = tf.keras.models.Model([e_inout, d_inout], [e_out, d_out])
model_train.compile(optimizer="adam", loss="mean_squared_error")
model_train.summary()
Model: "model"

```



```
[x] 検証を行うためのモデルを構築する。
デコーラーは二つのステップの力を有するのに注意。Seq2Seqの推論段階には遅く、それまでに推論した結果を次の段階に用いる。ステートコンテキストは、このときモデルの外に保持しておき、デコーラーの入力として渡される。
[1]
  1 model_pred_d = tf.keras.models.Model([inputs[e_inout], outputs[e_state]])
  2 pred_d_inout = tf.keras.layers.Input(shape=(1,))
  3 pred_d_outin = tf.keras.layers.Input(shape=(1, 1))
  4 pred_d_outin.set_shape([None, None, None, None])
  5 pred_d_outin, pred_d_state = L.inference_d(pred_d_inout, initial_state=[pred_d_state_in])
  6 pred_d_outin = d.dense_1d(pred_d_outin)
  7 pred_d_outin = d.dense_1d(pred_d_outin)
  8
  9 pred_d_model = tf.keras.Model([inputs[e_inout], pred_d_state_in], outputs=pred_d_outin, name='pred_d')
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
  38
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
  56
  57
  58
  59
  60
  61
  62
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
  75
  76
  77
  78
  79
  80
  81
  82
  83
  84
  85
  86
  87
  88
  89
  90
  91
  92
  93
  94
  95
  96
  97
  98
  99
  100
  101
  102
  103
  104
  105
  106
  107
  108
  109
  110
  111
  112
  113
  114
  115
  116
  117
  118
  119
  120
  121
  122
  123
  124
  125
  126
  127
  128
  129
  130
  131
  132
  133
  134
  135
  136
  137
  138
  139
  140
  141
  142
  143
  144
  145
  146
  147
  148
  149
  150
  151
  152
  153
  154
  155
  156
  157
  158
  159
  160
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1018
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1038
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1096
  1097
  1098
  1099
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1195
  1196
  1197
  1198
  1199
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1295
  1296
  1297
  1298
  1299
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1395
  1396
  1397
  1398
  1399
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1495
  1496
  1497
  1498
  1499
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1595
  1596
  1597
  1598
  1599
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1609
  1610
  1611
  161
```

