Deep Neural Networks

**Torch Installation**

1. !pip3 install torch

Output: Requirement already satisfied: torch in /usr/local/lib/python3.6/dist-packages (1.0.0)

**Importing Relevant Libraries**

1. import torch
2. import numpy as np
3. import matplotlib.pyplot as plt
4. import torch.nn as nn
5. from sklearn import datasets

**Initializing Dataset**

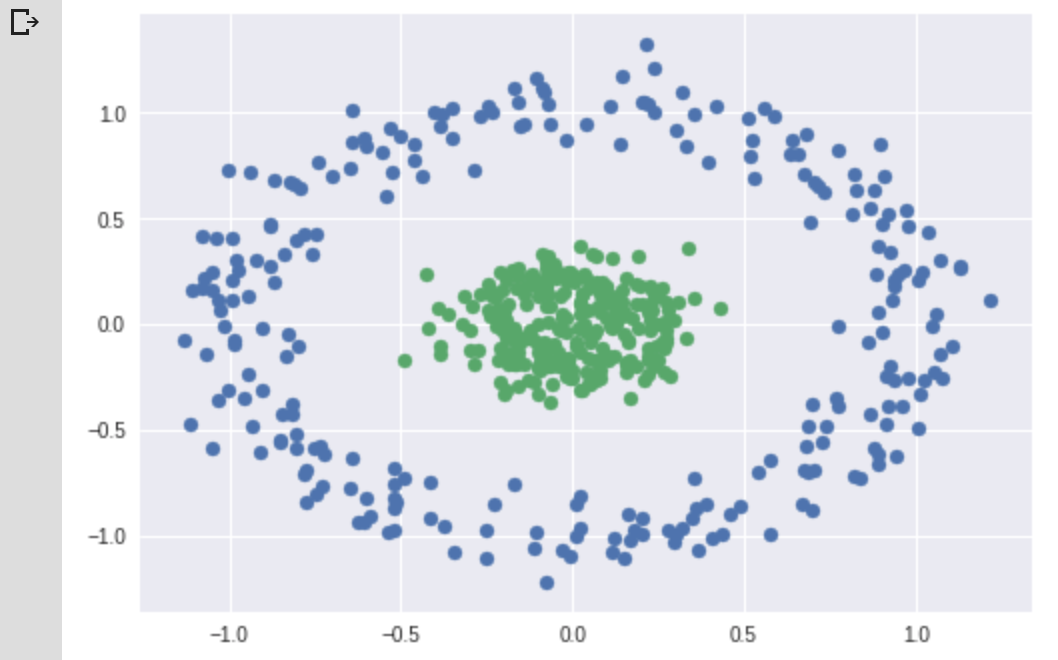
1. n\_pts = 500
2. X, y = datasets.make\_circles(n\_samples=n\_pts, random\_state=123, noise=0.1, factor=0.2)
3. x\_data = torch.Tensor(X)
4. y\_data = torch.Tensor(y.reshape(500, 1))

**Initializing Scatter Plot Function**

1. def scatter\_plot():
2. plt.scatter(X[y==0, 0], X[y==0, 1])
3. plt.scatter(X[y==1, 0], X[y==1, 1])

**Plotting Data**

1. scatter\_plot()



**Model Class Constructor**

1. class Model(nn.Module):
2. def \_\_init\_\_(self, input\_size, H1, output\_size):
3. super().\_\_init\_\_()
4. self.linear = nn.Linear(input\_size, H1)
5. self.linear2 = nn.Linear(H1, output\_size)
6. def forward(self, x):
7. x = torch.sigmoid(self.linear(x))
8. x = torch.sigmoid(self.linear2(x))
9. return x
10. def predict(self, x):
11. pred = self.forward(x)
12. if pred >= 0.5:
13. return 1
14. else:
15. return 0

**Initializing Model**

1. torch.manual\_seed(2)
2. model = Model(2, 4, 1)
3. print(list(model.parameters()))

Output: [Parameter containing: tensor([[ 0.1622, -0.1683],         [ 0.1939, -0.0361],         [ 0.3021,  0.1683],         [-0.0813, -0.5717]], requires\_grad=True), Parameter containing: tensor([ 0.1614, -0.6260,  0.0929,  0.0470], requires\_grad=True), Parameter containing: tensor([[-0.1099,  0.4088,  0.0334,  0.2073]], requires\_grad=True), Parameter containing: tensor([0.2116], requires\_grad=True)]

**Binary Cross Entropy Loss**

1. criterion = nn.BCELoss()
2. optimizer = torch.optim.Adam(model.parameters(), lr=0.1)

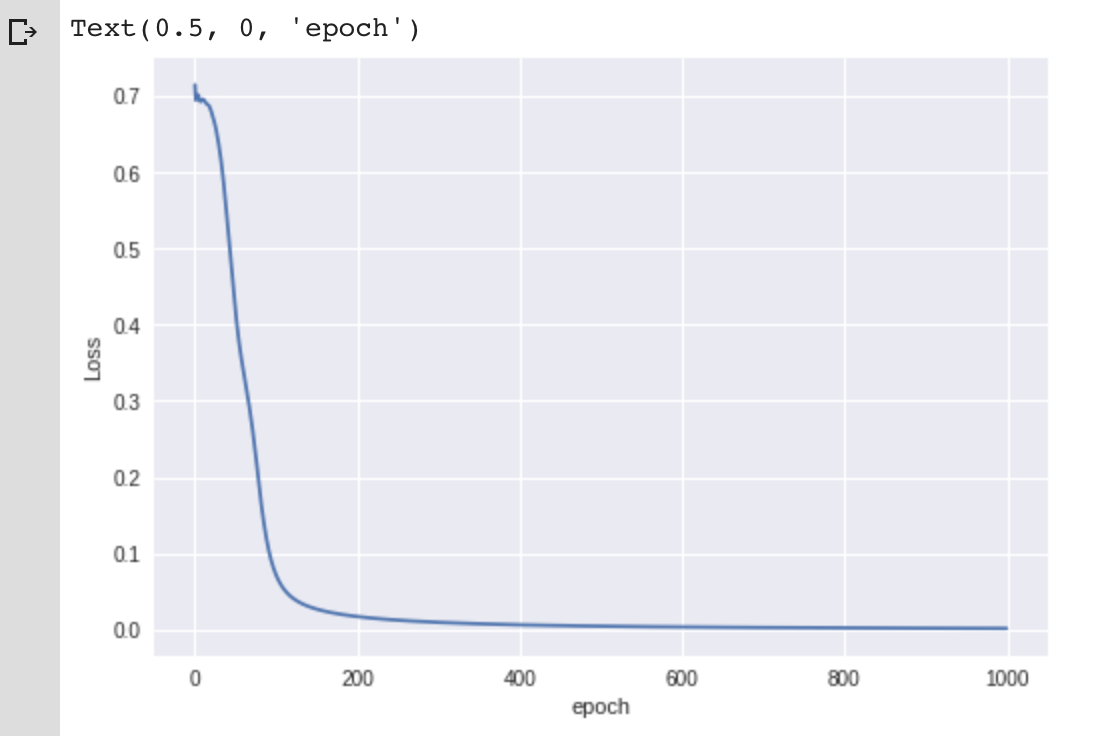
**Training Process**

1. epochs = 1000
2. losses = []
3. for i in range(epochs):
4. y\_pred = model.forward(x\_data)
5. loss = criterion(y\_pred, y\_data)
6. print("epoch:", i, "loss", loss.item())
7. losses.append(loss.item())
8. optimizer.zero\_grad()
9. loss.backward()
10. optimizer.step()

epoch: 0 loss 0.7148522138595581 epoch: 1 loss 0.6947312355041504 epoch: 2 loss 0.6955161094665527 epoch: 3 loss 0.7014897465705872 epoch: 4 loss 0.7013992667198181 epoch: 5 loss 0.697472095489502 epoch: 6 loss 0.6939275860786438 epoch: 7 loss 0.6928625106811523 epoch: 8 loss 0.693941593170166 epoch: 9 loss 0.6954426765441895 epoch: 10 loss 0.6958808302879333 epoch: 11 loss 0.6949241757392883 epoch: 12 loss 0.693157970905304 epoch: 13 loss 0.6914014220237732 epoch: 14 loss 0.6901755332946777 epoch: 15 loss 0.6894776225090027 epoch: 16 loss 0.6888664364814758 epoch: 17 loss 0.6877952814102173 epoch: 18 loss 0.6859349608421326 epoch: 19 loss 0.6832882165908813 epoch: 20 loss 0.6800878047943115 epoch: 21 loss 0.6766111850738525 epoch: 22 loss 0.6730255484580994 epoch: 23 loss 0.6693331003189087 epoch: 24 loss 0.6653968095779419 epoch: 25 loss 0.661038875579834 epoch: 26 loss 0.6561349034309387 epoch: 27 loss 0.6506633758544922 epoch: 28 loss 0.6446858048439026 epoch: 29 loss 0.6382841467857361 epoch: 30 loss 0.6314954161643982 epoch: 31 loss 0.6242801547050476 epoch: 32 loss 0.6165382266044617 epoch: 33 loss 0.6081583499908447 epoch: 34 loss 0.599071741104126 epoch: 35 loss 0.5892831683158875 epoch: 36 loss 0.5788646340370178 epoch: 37 loss 0.567934513092041 epoch: 38 loss 0.5566170811653137 epoch: 39 loss 0.5450131893157959 epoch: 40 loss 0.5331881046295166 epoch: 41 loss 0.5211832523345947 epoch: 42 loss 0.5090473890304565 epoch: 43 loss 0.4968593716621399 epoch: 44 loss 0.4847228527069092 epoch: 45 loss 0.47273701429367065 epoch: 46 loss 0.4609636068344116 epoch: 47 loss 0.4494229555130005 epoch: 48 loss 0.43812456727027893 epoch: 49 loss 0.4271085262298584 epoch: 50 loss 0.4164563715457916 epoch: 51 loss 0.40626832842826843 epoch: 52 loss 0.39661893248558044 epoch: 53 loss 0.3875318765640259 epoch: 54 loss 0.378994345664978 epoch: 55 loss 0.3709823787212372 epoch: 56 loss 0.36346206068992615 epoch: 57 loss 0.3563742935657501 epoch: 58 loss 0.3496249318122864 epoch: 59 loss 0.3431106209754944 epoch: 60 loss 0.3367486596107483 epoch: 61 loss 0.3304867148399353 epoch: 62 loss 0.3242776691913605 epoch: 63 loss 0.31806063652038574 epoch: 64 loss 0.31176477670669556 epoch: 65 loss 0.30532965064048767 epoch: 66 loss 0.2987109124660492 epoch: 67 loss 0.29187169671058655 epoch: 68 loss 0.2847749590873718 epoch: 69 loss 0.2773880362510681 epoch: 70 loss 0.26968953013420105 epoch: 71 loss 0.26166868209838867 epoch: 72 loss 0.25331950187683105 epoch: 73 loss 0.24464131891727448 epoch: 74 loss 0.23564566671848297 epoch: 75 loss 0.22636836767196655 epoch: 76 loss 0.2168763428926468 epoch: 77 loss 0.20726433396339417 epoch: 78 loss 0.19764815270900726 epoch: 79 loss 0.18815308809280396 epoch: 80 loss 0.17889809608459473 epoch: 81 loss 0.16998136043548584 epoch: 82 loss 0.161468505859375 epoch: 83 loss 0.15338830649852753 epoch: 84 loss 0.1457403600215912 epoch: 85 loss 0.13850761950016022 epoch: 86 loss 0.13167054951190948 epoch: 87 loss 0.12521560490131378 epoch: 88 loss 0.11913655698299408 epoch: 89 loss 0.11343125998973846 epoch: 90 loss 0.10809607803821564 epoch: 91 loss 0.10312210768461227 epoch: 92 loss 0.09849374741315842 epoch: 93 loss 0.09419038146734238 epoch: 94 loss 0.09018869698047638 epoch: 95 loss 0.08646497875452042 epoch: 96 loss 0.08299638330936432 epoch: 97 loss 0.0797617956995964 epoch: 98 loss 0.07674173265695572 epoch: 99 loss 0.07391861081123352 epoch: 100 loss 0.07127668708562851 epoch: 101 loss 0.06880202144384384 epoch: 102 loss 0.06648234277963638 epoch: 103 loss 0.06430692970752716 epoch: 104 loss 0.06226589158177376 epoch: 105 loss 0.060350000858306885 epoch: 106 loss 0.058550454676151276 epoch: 107 loss 0.05685851722955704 epoch: 108 loss 0.05526623874902725 epoch: 109 loss 0.053765829652547836 epoch: 110 loss 0.05235031992197037 epoch: 111 loss 0.05101332813501358 epoch: 112 loss 0.04974920675158501 epoch: 113 loss 0.04855281487107277 epoch: 114 loss 0.04741957038640976 epoch: 115 loss 0.04634511098265648 epoch: 116 loss 0.045325327664613724 epoch: 117 loss 0.044356316328048706 epoch: 118 loss 0.043434325605630875 epoch: 119 loss 0.04255584254860878 epoch: 120 loss 0.041717443615198135 epoch: 121 loss 0.040916189551353455 epoch: 122 loss 0.04014918580651283 epoch: 123 loss 0.03941398113965988 epoch: 124 loss 0.038708340376615524 epoch: 125 loss 0.03803040459752083 epoch: 126 loss 0.03737841546535492 epoch: 127 loss 0.03675095736980438 epoch: 128 loss 0.03614671528339386 epoch: 129 loss 0.03556445986032486 epoch: 130 loss 0.03500307723879814 epoch: 131 loss 0.03446146845817566 epoch: 132 loss 0.03393857553601265 epoch: 133 loss 0.033433444797992706 epoch: 134 loss 0.032945115119218826 epoch: 135 loss 0.032472606748342514 epoch: 136 loss 0.03201509639620781 epoch: 137 loss 0.03157180920243263 epoch: 138 loss 0.031141983345150948 epoch: 139 loss 0.030724940821528435 epoch: 140 loss 0.03032008372247219 epoch: 141 loss 0.029926858842372894 epoch: 142 loss 0.029544737190008163 epoch: 143 loss 0.02917322888970375 epoch: 144 loss 0.028811901807785034 epoch: 145 loss 0.02846027910709381 epoch: 146 loss 0.02811797335743904 epoch: 147 loss 0.02778458409011364 epoch: 148 loss 0.027459708973765373 epoch: 149 loss 0.027142977342009544 epoch: 150 loss 0.026834040880203247 epoch: 151 loss 0.02653254009783268 epoch: 152 loss 0.026238173246383667 epoch: 153 loss 0.025950629264116287 epoch: 154 loss 0.025669632479548454 epoch: 155 loss 0.025394927710294724 epoch: 156 loss 0.025126241147518158 epoch: 157 loss 0.02486337535083294 epoch: 158 loss 0.02460610494017601 epoch: 159 loss 0.02435426414012909 epoch: 160 loss 0.024107566103339195 epoch: 161 loss 0.023865947499871254 epoch: 162 loss 0.02362917549908161 epoch: 163 loss 0.02339707314968109 epoch: 164 loss 0.023169543594121933 epoch: 165 loss 0.022946393117308617 epoch: 166 loss 0.02272750623524189 epoch: 167 loss 0.02251272276043892 epoch: 168 loss 0.02230190299451351 epoch: 169 loss 0.02209496684372425 epoch: 170 loss 0.021891767159104347 epoch: 171 loss 0.02169221267104149 epoch: 172 loss 0.021496199071407318 epoch: 173 loss 0.02130361646413803 epoch: 174 loss 0.021114341914653778 epoch: 175 loss 0.02092834562063217 epoch: 176 loss 0.020745472982525826 epoch: 177 loss 0.020565692335367203 epoch: 178 loss 0.020388884469866753 epoch: 179 loss 0.020214980468153954 epoch: 180 loss 0.020043928176164627 epoch: 181 loss 0.01987561583518982 epoch: 182 loss 0.019709981977939606 epoch: 183 loss 0.01954697072505951 epoch: 184 loss 0.019386528059840202 epoch: 185 loss 0.01922856643795967 epoch: 186 loss 0.01907302625477314 epoch: 187 loss 0.018919849768280983 epoch: 188 loss 0.018768999725580215 epoch: 189 loss 0.018620390444993973 epoch: 190 loss 0.018473971635103226 epoch: 191 loss 0.018329724669456482 epoch: 192 loss 0.01818755455315113 epoch: 193 loss 0.018047457560896873 epoch: 194 loss 0.017909366637468338 epoch: 195 loss 0.017773212864995003 epoch: 196 loss 0.017638985067605972 epoch: 197 loss 0.017506632953882217 epoch: 198 loss 0.017376108095049858 epoch: 199 loss 0.01724737510085106 epoch: 200 loss 0.017120394855737686 epoch: 201 loss 0.016995126381516457 epoch: 202 loss 0.016871538013219833 epoch: 203 loss 0.016749590635299683 epoch: 204 loss 0.016629254445433617 epoch: 205 loss 0.016510486602783203 epoch: 206 loss 0.016393277794122696 epoch: 207 loss 0.016277581453323364 epoch: 208 loss 0.016163358464837074 epoch: 209 loss 0.016050592064857483 epoch: 210 loss 0.015939239412546158 epoch: 211 loss 0.0158293005079031 epoch: 212 loss 0.015720723196864128 epoch: 213 loss 0.01561348419636488 epoch: 214 loss 0.015507567673921585 epoch: 215 loss 0.015402935445308685 epoch: 216 loss 0.015299572609364986 epoch: 217 loss 0.015197459608316422 epoch: 218 loss 0.01509656012058258 epoch: 219 loss 0.014996864832937717 epoch: 220 loss 0.014898356050252914 epoch: 221 loss 0.014800972305238247 epoch: 222 loss 0.01470473688095808 epoch: 223 loss 0.01460962276905775 epoch: 224 loss 0.014515599235892296 epoch: 225 loss 0.014422631822526455 epoch: 226 loss 0.014330736361443996 epoch: 227 loss 0.01423988863825798 epoch: 228 loss 0.014150048606097698 epoch: 229 loss 0.014061207883059978 epoch: 230 loss 0.01397334598004818 epoch: 231 loss 0.013886465691030025 epoch: 232 loss 0.013800540938973427 epoch: 233 loss 0.013715534470975399 epoch: 234 loss 0.013631449081003666 epoch: 235 loss 0.013548275455832481 epoch: 236 loss 0.013465991243720055 epoch: 237 loss 0.01338458526879549 epoch: 238 loss 0.013304034247994423 epoch: 239 loss 0.013224335387349129 epoch: 240 loss 0.01314546912908554 epoch: 241 loss 0.013067418709397316 epoch: 242 loss 0.012990178540349007 epoch: 243 loss 0.01291373185813427 epoch: 244 loss 0.012838074006140232 epoch: 245 loss 0.012763177044689655 epoch: 246 loss 0.012689037248492241 epoch: 247 loss 0.012615654617547989 epoch: 248 loss 0.012542994692921638 epoch: 249 loss 0.012471077032387257 epoch: 250 loss 0.012399842031300068 epoch: 251 loss 0.012329336255788803 epoch: 252 loss 0.012259519658982754 epoch: 253 loss 0.012190369889140129 epoch: 254 loss 0.012121908366680145 epoch: 255 loss 0.01205409225076437 epoch: 256 loss 0.011986932717263699 epoch: 257 loss 0.01192041952162981 epoch: 258 loss 0.011854537762701511 epoch: 259 loss 0.011789293959736824 epoch: 260 loss 0.011724628508090973 epoch: 261 loss 0.011660599149763584 epoch: 262 loss 0.01159717421978712 epoch: 263 loss 0.0115343127399683 epoch: 264 loss 0.011472043581306934 epoch: 265 loss 0.011410360224545002 epoch: 266 loss 0.011349229142069817 epoch: 267 loss 0.011288655921816826 epoch: 268 loss 0.011228621937334538 epoch: 269 loss 0.011169148609042168 epoch: 270 loss 0.011110197752714157 epoch: 271 loss 0.011051773093640804 epoch: 272 loss 0.010993871837854385 epoch: 273 loss 0.01093649584800005 epoch: 274 loss 0.01087961345911026 epoch: 275 loss 0.010823245160281658 epoch: 276 loss 0.010767343454062939 epoch: 277 loss 0.010711945593357086 epoch: 278 loss 0.010657034814357758 epoch: 279 loss 0.010602590627968311 epoch: 280 loss 0.010548616759479046 epoch: 281 loss 0.010495091788470745 epoch: 282 loss 0.010442028753459454 epoch: 283 loss 0.010389419272542 epoch: 284 loss 0.010337254032492638 epoch: 285 loss 0.010285531170666218 epoch: 286 loss 0.010234236717224121 epoch: 287 loss 0.010183373466134071 epoch: 288 loss 0.010132933035492897 epoch: 289 loss 0.010082898661494255 epoch: 290 loss 0.010033289901912212 epoch: 291 loss 0.00998407881706953 epoch: 292 loss 0.009935273788869381 epoch: 293 loss 0.00988686177879572 epoch: 294 loss 0.009838852100074291 epoch: 295 loss 0.009791211225092411 epoch: 296 loss 0.009743969887495041 epoch: 297 loss 0.009697097353637218 epoch: 298 loss 0.009650609456002712 epoch: 299 loss 0.009604481048882008 epoch: 300 loss 0.009558715857565403 epoch: 301 loss 0.00951332040131092 epoch: 302 loss 0.009468273259699345 epoch: 303 loss 0.009423571638762951 epoch: 304 loss 0.009379226714372635 epoch: 305 loss 0.009335225448012352 epoch: 306 loss 0.009291558526456356 epoch: 307 loss 0.009248236194252968 epoch: 308 loss 0.009205232374370098 epoch: 309 loss 0.009162569418549538 epoch: 310 loss 0.009120220318436623 epoch: 311 loss 0.009078188799321651 epoch: 312 loss 0.009036477655172348 epoch: 313 loss 0.008995087817311287 epoch: 314 loss 0.008954005315899849 epoch: 315 loss 0.00891322921961546 epoch: 316 loss 0.008872750215232372 epoch: 317 loss 0.008832575753331184 epoch: 318 loss 0.008792692795395851 epoch: 319 loss 0.008753113448619843 epoch: 320 loss 0.00871380977332592 epoch: 321 loss 0.008674792014062405 epoch: 322 loss 0.008636065758764744 epoch: 323 loss 0.00859761517494917 epoch: 324 loss 0.00855944398790598 epoch: 325 loss 0.008521548472344875 epoch: 326 loss 0.008483917452394962 epoch: 327 loss 0.008446559309959412 epoch: 328 loss 0.008409467525780201 epoch: 329 loss 0.008372627198696136 epoch: 330 loss 0.008336056023836136 epoch: 331 loss 0.008299744687974453 epoch: 332 loss 0.008263678289949894 epoch: 333 loss 0.008227871730923653 epoch: 334 loss 0.008192315697669983 epoch: 335 loss 0.008156996220350266 epoch: 336 loss 0.008121922612190247 epoch: 337 loss 0.008087096735835075 epoch: 338 loss 0.008052508346736431 epoch: 339 loss 0.008018153719604015 epoch: 340 loss 0.007984041236341 epoch: 341 loss 0.007950142957270145 epoch: 342 loss 0.007916491478681564 epoch: 343 loss 0.00788305327296257 epoch: 344 loss 0.007849844172596931 epoch: 345 loss 0.007816866971552372 epoch: 346 loss 0.007784097921103239 epoch: 347 loss 0.007751546334475279 epoch: 348 loss 0.007719220127910376 epoch: 349 loss 0.007687097880989313 epoch: 350 loss 0.007655191235244274 epoch: 351 loss 0.007623497396707535 epoch: 352 loss 0.007592014968395233 epoch: 353 loss 0.007560729049146175 epoch: 354 loss 0.0075296503491699696 epoch: 355 loss 0.007498762104660273 epoch: 356 loss 0.007468090392649174 epoch: 357 loss 0.00743761146441102 epoch: 358 loss 0.0074073271825909615 epoch: 359 loss 0.007377236150205135 epoch: 360 loss 0.007347338367253542 epoch: 361 loss 0.007317631505429745 epoch: 362 loss 0.007288113236427307 epoch: 363 loss 0.007258783094584942 epoch: 364 loss 0.0072296373546123505 epoch: 365 loss 0.0072006783448159695 epoch: 366 loss 0.007171896751970053 epoch: 367 loss 0.007143298629671335 epoch: 368 loss 0.007114880718290806 epoch: 369 loss 0.00708663510158658 epoch: 370 loss 0.007058569695800543 epoch: 371 loss 0.007030674256384373 epoch: 372 loss 0.007002952042967081 epoch: 373 loss 0.006975407246500254 epoch: 374 loss 0.006948031950742006 epoch: 375 loss 0.006920812651515007 epoch: 376 loss 0.006893768906593323 epoch: 377 loss 0.006866890005767345 epoch: 378 loss 0.006840173155069351 epoch: 379 loss 0.006813624873757362 epoch: 380 loss 0.006787230260670185 epoch: 381 loss 0.006761001423001289 epoch: 382 loss 0.006734933238476515 epoch: 383 loss 0.006709013134241104 epoch: 384 loss 0.006683253217488527 epoch: 385 loss 0.006657651625573635 epoch: 386 loss 0.00663219578564167 epoch: 387 loss 0.006606899667531252 epoch: 388 loss 0.00658175116404891 epoch: 389 loss 0.006556747015565634 epoch: 390 loss 0.006531901191920042 epoch: 391 loss 0.0065071978606283665 epoch: 392 loss 0.006482640281319618 epoch: 393 loss 0.006458229850977659 epoch: 394 loss 0.006433961447328329 epoch: 395 loss 0.0064098346047103405 epoch: 396 loss 0.0063858479261398315 epoch: 397 loss 0.0063620032742619514 epoch: 398 loss 0.006338302977383137 epoch: 399 loss 0.006314740050584078 epoch: 400 loss 0.006291310302913189 epoch: 401 loss 0.0062680174596607685 epoch: 402 loss 0.006244867108762264 epoch: 403 loss 0.006221843417733908 epoch: 404 loss 0.0061989580281078815 epoch: 405 loss 0.006176200229674578 epoch: 406 loss 0.006153573282063007 epoch: 407 loss 0.006131080910563469 epoch: 408 loss 0.0061087156645953655 epoch: 409 loss 0.006086483597755432 epoch: 410 loss 0.0060643707402050495 epoch: 411 loss 0.0060423919931054115 epoch: 412 loss 0.006020532455295324 epoch: 413 loss 0.00599880563095212 epoch: 414 loss 0.005977187305688858 epoch: 415 loss 0.005955709144473076 epoch: 416 loss 0.005934345535933971 epoch: 417 loss 0.005913109984248877 epoch: 418 loss 0.00589198712259531 epoch: 419 loss 0.005870979279279709 epoch: 420 loss 0.00585010414943099 epoch: 421 loss 0.005829344503581524 epoch: 422 loss 0.005808691028505564 epoch: 423 loss 0.005788162350654602 epoch: 424 loss 0.0057677458971738815 epoch: 425 loss 0.005747445393353701 epoch: 426 loss 0.005727254785597324 epoch: 427 loss 0.005707186181098223 epoch: 428 loss 0.005687224678695202 epoch: 429 loss 0.005667377263307571 epoch: 430 loss 0.00564763555303216 epoch: 431 loss 0.005628006532788277 epoch: 432 loss 0.0056084864772856236 epoch: 433 loss 0.005589072126895189 epoch: 434 loss 0.005569774657487869 epoch: 435 loss 0.005550573579967022 epoch: 436 loss 0.005531479138880968 epoch: 437 loss 0.005512497387826443 epoch: 438 loss 0.005493616685271263 epoch: 439 loss 0.005474843550473452 epoch: 440 loss 0.005456168670207262 epoch: 441 loss 0.005437597166746855 epoch: 442 loss 0.005419131834059954 epoch: 443 loss 0.005400771275162697 epoch: 444 loss 0.005382504779845476 epoch: 445 loss 0.005364337470382452 epoch: 446 loss 0.005346277728676796 epoch: 447 loss 0.005328308325260878 epoch: 448 loss 0.005310437176376581 epoch: 449 loss 0.005292666610330343 epoch: 450 loss 0.0052749947644770145 epoch: 451 loss 0.005257411859929562 epoch: 452 loss 0.005239939317107201 epoch: 453 loss 0.00522254453971982 epoch: 454 loss 0.005205248016864061 epoch: 455 loss 0.005188052076846361 epoch: 456 loss 0.005170947872102261 epoch: 457 loss 0.005153931211680174 epoch: 458 loss 0.005137014668434858 epoch: 459 loss 0.005120173562318087 epoch: 460 loss 0.005103442817926407 epoch: 461 loss 0.005086786113679409 epoch: 462 loss 0.0050702267326414585 epoch: 463 loss 0.00505375349894166 epoch: 464 loss 0.005037365946918726 epoch: 465 loss 0.005021071992814541 epoch: 466 loss 0.005004859995096922 epoch: 467 loss 0.004988736007362604 epoch: 468 loss 0.004972696304321289 epoch: 469 loss 0.0049567436799407005 epoch: 470 loss 0.0049408781342208385 epoch: 471 loss 0.004925092216581106 epoch: 472 loss 0.004909399431198835 epoch: 473 loss 0.004893776960670948 epoch: 474 loss 0.004878244828432798 epoch: 475 loss 0.004862789064645767 epoch: 476 loss 0.004847422707825899 epoch: 477 loss 0.004832129925489426 epoch: 478 loss 0.0048169237561523914 epoch: 479 loss 0.004801790229976177 epoch: 480 loss 0.004786740988492966 epoch: 481 loss 0.004771769512444735 epoch: 482 loss 0.0047568767331540585 epoch: 483 loss 0.004742060787975788 epoch: 484 loss 0.0047273216769099236 epoch: 485 loss 0.004712662193924189 epoch: 486 loss 0.004698075819760561 epoch: 487 loss 0.004683563020080328 epoch: 488 loss 0.004669127054512501 epoch: 489 loss 0.004654772579669952 epoch: 490 loss 0.004640480969101191 epoch: 491 loss 0.004626270849257708 epoch: 492 loss 0.004612135235220194 epoch: 493 loss 0.0045980727300047874 epoch: 494 loss 0.004584071226418018 epoch: 495 loss 0.004570152144879103 epoch: 496 loss 0.004556303843855858 epoch: 497 loss 0.004542525392025709 epoch: 498 loss 0.004528812598437071 epoch: 499 loss 0.004515174310654402 epoch: 500 loss 0.004501603543758392 epoch: 501 loss 0.00448810588568449 epoch: 502 loss 0.004474671091884375 epoch: 503 loss 0.004461308009922504 epoch: 504 loss 0.0044480119831860065 epoch: 505 loss 0.004434778355062008 epoch: 506 loss 0.004421618301421404 epoch: 507 loss 0.004408523440361023 epoch: 508 loss 0.004395491909235716 epoch: 509 loss 0.004382522776722908 epoch: 510 loss 0.004369627218693495 epoch: 511 loss 0.004356789868324995 epoch: 512 loss 0.00434401910752058 epoch: 513 loss 0.004331309348344803 epoch: 514 loss 0.004318661522120237 epoch: 515 loss 0.00430608494207263 epoch: 516 loss 0.004293565638363361 epoch: 517 loss 0.00428110733628273 epoch: 518 loss 0.00426871282979846 epoch: 519 loss 0.0042563797906041145 epoch: 520 loss 0.004244105890393257 epoch: 521 loss 0.004231888800859451 epoch: 522 loss 0.004219742026180029 epoch: 523 loss 0.004207647405564785 epoch: 524 loss 0.0041956123895943165 epoch: 525 loss 0.004183639772236347 epoch: 526 loss 0.0041717179119586945 epoch: 527 loss 0.004159861244261265 epoch: 528 loss 0.004148063249886036 epoch: 529 loss 0.004136316478252411 epoch: 530 loss 0.004124634433537722 epoch: 531 loss 0.004113004542887211 epoch: 532 loss 0.004101428668946028 epoch: 533 loss 0.004089910537004471 epoch: 534 loss 0.0040784478187561035 epoch: 535 loss 0.0040670400485396385 epoch: 536 loss 0.0040556928142905235 epoch: 537 loss 0.004044391680508852 epoch: 538 loss 0.004033150151371956 epoch: 539 loss 0.004021953791379929 epoch: 540 loss 0.004010817501693964 epoch: 541 loss 0.003999733831733465 epoch: 542 loss 0.003988706041127443 epoch: 543 loss 0.003977724350988865 epoch: 544 loss 0.003966798074543476 epoch: 545 loss 0.003955926280468702 epoch: 546 loss 0.00394509918987751 epoch: 547 loss 0.003934327978640795 epoch: 548 loss 0.003923606127500534 epoch: 549 loss 0.003912931773811579 epoch: 550 loss 0.0039023079443722963 epoch: 551 loss 0.0038917353376746178 epoch: 552 loss 0.0038812123239040375 epoch: 553 loss 0.003870743792504072 epoch: 554 loss 0.0038603132124990225 epoch: 555 loss 0.003849932923913002 epoch: 556 loss 0.0038396099116653204 epoch: 557 loss 0.0038293295074254274 epoch: 558 loss 0.003819093806669116 epoch: 559 loss 0.0038089058361947536 epoch: 560 loss 0.0037987639661878347 epoch: 561 loss 0.0037886686623096466 epoch: 562 loss 0.003778627375140786 epoch: 563 loss 0.003768624970689416 epoch: 564 loss 0.0037586695980280638 epoch: 565 loss 0.003748757764697075 epoch: 566 loss 0.0037388931959867477 epoch: 567 loss 0.003729073563590646 epoch: 568 loss 0.0037192991003394127 epoch: 569 loss 0.003709565382450819 epoch: 570 loss 0.0036998779978603125 epoch: 571 loss 0.0036902313586324453 epoch: 572 loss 0.0036806308198720217 epoch: 573 loss 0.0036710731219500303 epoch: 574 loss 0.003661559661850333 epoch: 575 loss 0.003652090672403574 epoch: 576 loss 0.0036426580045372248 epoch: 577 loss 0.0036332698073238134 epoch: 578 loss 0.0036239258479326963 epoch: 579 loss 0.003614617045968771 epoch: 580 loss 0.003605354344472289 epoch: 581 loss 0.0035961326211690903 epoch: 582 loss 0.0035869537387043238 epoch: 583 loss 0.0035778095480054617 epoch: 584 loss 0.00356870936229825 epoch: 585 loss 0.0035596489906311035 epoch: 586 loss 0.0035506244748830795 epoch: 587 loss 0.0035416423343122005 epoch: 588 loss 0.0035326990764588118 epoch: 589 loss 0.003523795632645488 epoch: 590 loss 0.0035149301402270794 epoch: 591 loss 0.003506102366372943 epoch: 592 loss 0.003497316036373377 epoch: 593 loss 0.003488565795123577 epoch: 594 loss 0.003479853505268693 epoch: 595 loss 0.0034711784683167934 epoch: 596 loss 0.0034625413827598095 epoch: 597 loss 0.003453941782936454 epoch: 598 loss 0.0034453750122338533 epoch: 599 loss 0.003436845261603594 epoch: 600 loss 0.003428352065384388 epoch: 601 loss 0.0034199003130197525 epoch: 602 loss 0.003411481622606516 epoch: 603 loss 0.0034030962269753218 epoch: 604 loss 0.0033947466872632504 epoch: 605 loss 0.0033864369615912437 epoch: 606 loss 0.0033781572710722685 epoch: 607 loss 0.0033699176274240017 epoch: 608 loss 0.0033617098815739155 epoch: 609 loss 0.003353532636538148 epoch: 610 loss 0.0033453963696956635 epoch: 611 loss 0.003337287809699774 epoch: 612 loss 0.0033292167354375124 epoch: 613 loss 0.0033211777918040752 epoch: 614 loss 0.00331317400559783 epoch: 615 loss 0.003305202815681696 epoch: 616 loss 0.0032972649205476046 epoch: 617 loss 0.003289362182840705 epoch: 618 loss 0.0032814843580126762 epoch: 619 loss 0.003273643786087632 epoch: 620 loss 0.003265836276113987 epoch: 621 loss 0.003258058335632086 epoch: 622 loss 0.0032503113616257906 epoch: 623 loss 0.003242598380893469 epoch: 624 loss 0.003234916366636753 epoch: 625 loss 0.0032272629905492067 epoch: 626 loss 0.003219646168872714 epoch: 627 loss 0.003212055191397667 epoch: 628 loss 0.003204498440027237 epoch: 629 loss 0.003196968697011471 epoch: 630 loss 0.003189471550285816 epoch: 631 loss 0.003182003740221262 epoch: 632 loss 0.0031745685264468193 epoch: 633 loss 0.0031671596225351095 epoch: 634 loss 0.0031597805209457874 epoch: 635 loss 0.0031524335499852896 epoch: 636 loss 0.003145110560581088 epoch: 637 loss 0.0031378206331282854 epoch: 638 loss 0.0031305616721510887 epoch: 639 loss 0.003123326925560832 epoch: 640 loss 0.0031161257065832615 epoch: 641 loss 0.003108948701992631 epoch: 642 loss 0.0031018031295388937 epoch: 643 loss 0.0030946782790124416 epoch: 644 loss 0.003087588818743825 epoch: 645 loss 0.003080527065321803 epoch: 646 loss 0.003073487663641572 epoch: 647 loss 0.003066479926928878 epoch: 648 loss 0.003059498965740204 epoch: 649 loss 0.0030525443144142628 epoch: 650 loss 0.0030456145759671926 epoch: 651 loss 0.003038711380213499 epoch: 652 loss 0.003031840780749917 epoch: 653 loss 0.00302499090321362 epoch: 654 loss 0.0030181691981852055 epoch: 655 loss 0.003011376829817891 epoch: 656 loss 0.0030046047177165747 epoch: 657 loss 0.002997864270582795 epoch: 658 loss 0.0029911480378359556 epoch: 659 loss 0.0029844525270164013 epoch: 660 loss 0.0029777868185192347 epoch: 661 loss 0.002971145324409008 epoch: 662 loss 0.0029645299073308706 epoch: 663 loss 0.002957937540486455 epoch: 664 loss 0.0029513719491660595 epoch: 665 loss 0.0029448296409100294 epoch: 666 loss 0.0029383155051618814 epoch: 667 loss 0.0029318188317120075 epoch: 668 loss 0.0029253519605845213 epoch: 669 loss 0.0029189065098762512 epoch: 670 loss 0.002912485273554921 epoch: 671 loss 0.002906091744080186 epoch: 672 loss 0.0028997198678553104 epoch: 673 loss 0.0028933731373399496 epoch: 674 loss 0.002887043636292219 epoch: 675 loss 0.0028807425405830145 epoch: 676 loss 0.002874462166801095 epoch: 677 loss 0.0028682081028819084 epoch: 678 loss 0.002861978253349662 epoch: 679 loss 0.0028557665646076202 epoch: 680 loss 0.002849582815542817 epoch: 681 loss 0.002843417227268219 epoch: 682 loss 0.00283727515488863 epoch: 683 loss 0.0028311549685895443 epoch: 684 loss 0.002825058065354824 epoch: 685 loss 0.0028189809527248144 epoch: 686 loss 0.0028129287529736757 epoch: 687 loss 0.0028068991377949715 epoch: 688 loss 0.002800889080390334 epoch: 689 loss 0.0027949009090662003 epoch: 690 loss 0.0027889367192983627 epoch: 691 loss 0.002782992785796523 epoch: 692 loss 0.0027770681772381067 epoch: 693 loss 0.0027711607981473207 epoch: 694 loss 0.002765284851193428 epoch: 695 loss 0.002759425202384591 epoch: 696 loss 0.002753583714365959 epoch: 697 loss 0.0027477657422423363 epoch: 698 loss 0.0027419677935540676 epoch: 699 loss 0.002736190101131797 epoch: 700 loss 0.0027304321993142366 epoch: 701 loss 0.0027246945537626743 epoch: 702 loss 0.00271897972561419 epoch: 703 loss 0.0027132821269333363 epoch: 704 loss 0.0027076059486716986 epoch: 705 loss 0.002701947931200266 epoch: 706 loss 0.0026963127311319113 epoch: 707 loss 0.002690694062039256 epoch: 708 loss 0.0026850972790271044 epoch: 709 loss 0.002679518423974514 epoch: 710 loss 0.002673960058018565 epoch: 711 loss 0.0026684210170060396 epoch: 712 loss 0.0026629045605659485 epoch: 713 loss 0.002657396951690316 epoch: 714 loss 0.002651917515322566 epoch: 715 loss 0.0026464529801160097 epoch: 716 loss 0.0026410087011754513 epoch: 717 loss 0.0026355793233960867 epoch: 718 loss 0.002630174858495593 epoch: 719 loss 0.0026247850619256496 epoch: 720 loss 0.0026194120291620493 epoch: 721 loss 0.002614059019833803 epoch: 722 loss 0.0026087246369570494 epoch: 723 loss 0.002603408182039857 epoch: 724 loss 0.002598108956590295 epoch: 725 loss 0.0025928313843905926 epoch: 726 loss 0.002587565453723073 epoch: 727 loss 0.0025823204778134823 epoch: 728 loss 0.002577098086476326 epoch: 729 loss 0.002571886172518134 epoch: 730 loss 0.002566695213317871 epoch: 731 loss 0.0025615200866013765 epoch: 732 loss 0.002556363819167018 epoch: 733 loss 0.002551223151385784 epoch: 734 loss 0.0025460992474108934 epoch: 735 loss 0.0025409930385649204 epoch: 736 loss 0.0025359017308801413 epoch: 737 loss 0.0025308318436145782 epoch: 738 loss 0.00252577499486506 epoch: 739 loss 0.00252073653973639 epoch: 740 loss 0.002515715081244707 epoch: 741 loss 0.0025107089895755053 epoch: 742 loss 0.0025057231541723013 epoch: 743 loss 0.002500746166333556 epoch: 744 loss 0.0024957924615591764 epoch: 745 loss 0.0024908508639782667 epoch: 746 loss 0.002485924167558551 epoch: 747 loss 0.00248101563192904 epoch: 748 loss 0.002476125257089734 epoch: 749 loss 0.0024712500162422657 epoch: 750 loss 0.00246638897806406 epoch: 751 loss 0.002461543772369623 epoch: 752 loss 0.002456717425957322 epoch: 753 loss 0.002451901091262698 epoch: 754 loss 0.0024471029173582792 epoch: 755 loss 0.0024423201102763414 epoch: 756 loss 0.0024375522043555975 epoch: 757 loss 0.0024328024592250586 epoch: 758 loss 0.002428062492981553 epoch: 759 loss 0.0024233409203588963 epoch: 760 loss 0.0024186361115425825 epoch: 761 loss 0.0024139429442584515 epoch: 762 loss 0.0024092679377645254 epoch: 763 loss 0.0024046064354479313 epoch: 764 loss 0.0023999582044780254 epoch: 765 loss 0.0023953246418386698 epoch: 766 loss 0.002390708774328232 epoch: 767 loss 0.00238610222004354 epoch: 768 loss 0.0023815161548554897 epoch: 769 loss 0.002376943128183484 epoch: 770 loss 0.0023723836056888103 epoch: 771 loss 0.0023678371217101812 epoch: 772 loss 0.00236330833286047 epoch: 773 loss 0.0023587895557284355 epoch: 774 loss 0.0023542875424027443 epoch: 775 loss 0.0023497992660850286 epoch: 776 loss 0.0023453214671462774 epoch: 777 loss 0.002340861828997731 epoch: 778 loss 0.0023364133667200804 epoch: 779 loss 0.0023319825995713472 epoch: 780 loss 0.002327563241124153 epoch: 781 loss 0.0023231550585478544 epoch: 782 loss 0.002318763639777899 epoch: 783 loss 0.0023143833968788385 epoch: 784 loss 0.0023100159596651793 epoch: 785 loss 0.0023056655190885067 epoch: 786 loss 0.002301326021552086 epoch: 787 loss 0.002297003287822008 epoch: 788 loss 0.0022926891688257456 epoch: 789 loss 0.0022883897181600332 epoch: 790 loss 0.002284102374687791 epoch: 791 loss 0.002279829466715455 epoch: 792 loss 0.002275572158396244 epoch: 793 loss 0.002271323697641492 epoch: 794 loss 0.0022670880425721407 epoch: 795 loss 0.0022628677543252707 epoch: 796 loss 0.0022586605045944452 epoch: 797 loss 0.0022544634994119406 epoch: 798 loss 0.002250279765576124 epoch: 799 loss 0.0022461102344095707 epoch: 800 loss 0.0022419504821300507 epoch: 801 loss 0.002237802604213357 epoch: 802 loss 0.0022336712572723627 epoch: 803 loss 0.0022295513190329075 epoch: 804 loss 0.0022254453506320715 epoch: 805 loss 0.0022213486954569817 epoch: 806 loss 0.002217263448983431 epoch: 807 loss 0.0022131914738565683 epoch: 808 loss 0.002209131605923176 epoch: 809 loss 0.002205083379521966 epoch: 810 loss 0.002201046794652939 epoch: 811 loss 0.0021970218513160944 epoch: 812 loss 0.0021930101793259382 epoch: 813 loss 0.002189008519053459 epoch: 814 loss 0.0021850208286195993 epoch: 815 loss 0.002181044779717922 epoch: 816 loss 0.0021770792081952095 epoch: 817 loss 0.0021731252782046795 epoch: 818 loss 0.0021691848523914814 epoch: 819 loss 0.0021652523428201675 epoch: 820 loss 0.002161332406103611 epoch: 821 loss 0.00215742364525795 epoch: 822 loss 0.0021535283885896206 epoch: 823 loss 0.0021496443077921867 epoch: 824 loss 0.002145772334188223 epoch: 825 loss 0.002141907112672925 epoch: 826 loss 0.002138057490810752 epoch: 827 loss 0.0021342141553759575 epoch: 828 loss 0.0021303861867636442 epoch: 829 loss 0.0021265738178044558 epoch: 830 loss 0.0021227598190307617 epoch: 831 loss 0.0021189660765230656 epoch: 832 loss 0.002115183509886265 epoch: 833 loss 0.002111406298354268 epoch: 834 loss 0.002107646083459258 epoch: 835 loss 0.0021038923878222704 epoch: 836 loss 0.002100147772580385 epoch: 837 loss 0.0020964154973626137 epoch: 838 loss 0.0020926971919834614 epoch: 839 loss 0.0020889861043542624 epoch: 840 loss 0.002085287356749177 epoch: 841 loss 0.0020815988536924124 epoch: 842 loss 0.002077919663861394 epoch: 843 loss 0.0020742518827319145 epoch: 844 loss 0.0020705920178443193 epoch: 845 loss 0.002066947054117918 epoch: 846 loss 0.0020633067470043898 epoch: 847 loss 0.002059679012745619 epoch: 848 loss 0.0020560643170028925 epoch: 849 loss 0.0020524554420262575 epoch: 850 loss 0.0020488582085818052 epoch: 851 loss 0.0020452735479921103 epoch: 852 loss 0.0020416947081685066 epoch: 853 loss 0.002038131467998028 epoch: 854 loss 0.0020345752127468586 epoch: 855 loss 0.0020310254767537117 epoch: 856 loss 0.0020274927373975515 epoch: 857 loss 0.002023963490501046 epoch: 858 loss 0.0020204440224915743 epoch: 859 loss 0.0020169371273368597 epoch: 860 loss 0.002013435820117593 epoch: 861 loss 0.0020099508110433817 epoch: 862 loss 0.002006473718211055 epoch: 863 loss 0.002003003377467394 epoch: 864 loss 0.0019995425827801228 epoch: 865 loss 0.0019960927311331034 epoch: 866 loss 0.001992653589695692 epoch: 867 loss 0.0019892235286533833 epoch: 868 loss 0.0019858013838529587 epoch: 869 loss 0.0019823857583105564 epoch: 870 loss 0.0019789808429777622 epoch: 871 loss 0.001975588733330369 epoch: 872 loss 0.001972203841432929 epoch: 873 loss 0.001968828495591879 epoch: 874 loss 0.0019654619973152876 epoch: 875 loss 0.0019621059764176607 epoch: 876 loss 0.0019587590359151363 epoch: 877 loss 0.0019554204773157835 epoch: 878 loss 0.001952089834958315 epoch: 879 loss 0.0019487675745040178 epoch: 880 loss 0.0019454591674730182 epoch: 881 loss 0.0019421535544097424 epoch: 882 loss 0.001938859117217362 epoch: 883 loss 0.0019355752738192677 epoch: 884 loss 0.0019322963198646903 epoch: 885 loss 0.0019290311029180884 epoch: 886 loss 0.0019257678650319576 epoch: 887 loss 0.0019225215073674917 epoch: 888 loss 0.0019192807376384735 epoch: 889 loss 0.001916045555844903 epoch: 890 loss 0.0019128205021843314 epoch: 891 loss 0.0019096052274107933 epoch: 892 loss 0.0019063985673710704 epoch: 893 loss 0.0019032000564038754 epoch: 894 loss 0.0019000101601704955 epoch: 895 loss 0.0018968278309330344 epoch: 896 loss 0.001893653767183423 epoch: 897 loss 0.0018904866883531213 epoch: 898 loss 0.0018873290391638875 epoch: 899 loss 0.001884182682260871 epoch: 900 loss 0.0018810407491400838 epoch: 901 loss 0.0018779117381200194 epoch: 902 loss 0.0018747885478660464 epoch: 903 loss 0.0018716711783781648 epoch: 904 loss 0.001868565334007144 epoch: 905 loss 0.0018654647283256054 epoch: 906 loss 0.001862370758317411 epoch: 907 loss 0.0018592911073938012 epoch: 908 loss 0.0018562162294983864 epoch: 909 loss 0.0018531479872763157 epoch: 910 loss 0.0018500882433727384 epoch: 911 loss 0.001847035950049758 epoch: 912 loss 0.0018439912237226963 epoch: 913 loss 0.0018409577896818519 epoch: 914 loss 0.0018379288958385587 epoch: 915 loss 0.00183491047937423 epoch: 916 loss 0.001831896835938096 epoch: 917 loss 0.0018288949504494667 epoch: 918 loss 0.001825896673835814 epoch: 919 loss 0.001822908059693873 epoch: 920 loss 0.0018199268961325288 epoch: 921 loss 0.0018169552786275744 epoch: 922 loss 0.0018139892490580678 epoch: 923 loss 0.0018110257806256413 epoch: 924 loss 0.0018080800073221326 epoch: 925 loss 0.0018051357474178076 epoch: 926 loss 0.0018021995201706886 epoch: 927 loss 0.0017992716748267412 epoch: 928 loss 0.001796351047232747 epoch: 929 loss 0.0017934388015419245 epoch: 930 loss 0.0017905301647260785 epoch: 931 loss 0.0017876315396279097 epoch: 932 loss 0.0017847403651103377 epoch: 933 loss 0.0017818582709878683 epoch: 934 loss 0.0017789840931072831 epoch: 935 loss 0.0017761148046702147 epoch: 936 loss 0.0017732486594468355 epoch: 937 loss 0.001770398230291903 epoch: 938 loss 0.0017675487324595451 epoch: 939 loss 0.0017647084314376116 epoch: 940 loss 0.0017618763959035277 epoch: 941 loss 0.001759049715474248 epoch: 942 loss 0.001756231882609427 epoch: 943 loss 0.0017534163780510426 epoch: 944 loss 0.0017506106523796916 epoch: 945 loss 0.001747814705595374 epoch: 946 loss 0.0017450221348553896 epoch: 947 loss 0.0017422375967726111 epoch: 948 loss 0.0017394589958712459 epoch: 949 loss 0.001736690872348845 epoch: 950 loss 0.0017339267069473863 epoch: 951 loss 0.001731172320432961 epoch: 952 loss 0.0017284232890233397 epoch: 953 loss 0.0017256784485653043 epoch: 954 loss 0.0017229446675628424 epoch: 955 loss 0.0017202149610966444 epoch: 956 loss 0.0017174900276586413 epoch: 957 loss 0.0017147751059383154 epoch: 958 loss 0.001712063909508288 epoch: 959 loss 0.001709361094981432 epoch: 960 loss 0.0017066631698980927 epoch: 961 loss 0.001703976420685649 epoch: 962 loss 0.0017012932803481817 epoch: 963 loss 0.0016986160771921277 epoch: 964 loss 0.0016959485365077853 epoch: 965 loss 0.0016932835569605231 epoch: 966 loss 0.0016906268429011106 epoch: 967 loss 0.001687975600361824 epoch: 968 loss 0.001685332041233778 epoch: 969 loss 0.0016826954670250416 epoch: 970 loss 0.0016800632001832128 epoch: 971 loss 0.0016774411778897047 epoch: 972 loss 0.0016748220659792423 epoch: 973 loss 0.0016722100554034114 epoch: 974 loss 0.00166960374917835 epoch: 975 loss 0.0016670055920258164 epoch: 976 loss 0.0016644116258248687 epoch: 977 loss 0.0016618247609585524 epoch: 978 loss 0.001659245346672833 epoch: 979 loss 0.0016566700069233775 epoch: 980 loss 0.001654103398323059 epoch: 981 loss 0.0016515376046299934 epoch: 982 loss 0.0016489846166223288 epoch: 983 loss 0.0016464322106912732 epoch: 984 loss 0.0016438895836472511 epoch: 985 loss 0.001641349750570953 epoch: 986 loss 0.001638819114305079 epoch: 987 loss 0.0016362936003133655 epoch: 988 loss 0.001633774721994996 epoch: 989 loss 0.001631261082366109 epoch: 990 loss 0.0016287550097331405 epoch: 991 loss 0.0016262511489912868 epoch: 992 loss 0.001623757416382432 epoch: 993 loss 0.0016212720656767488 epoch: 994 loss 0.0016187846194952726 epoch: 995 loss 0.0016163060208782554 epoch: 996 loss 0.001613831496797502 epoch: 997 loss 0.0016113673336803913 epoch: 998 loss 0.001608904218301177 epoch: 999 loss 0.0016064513474702835

**Plotting Losses at Various Epochs**

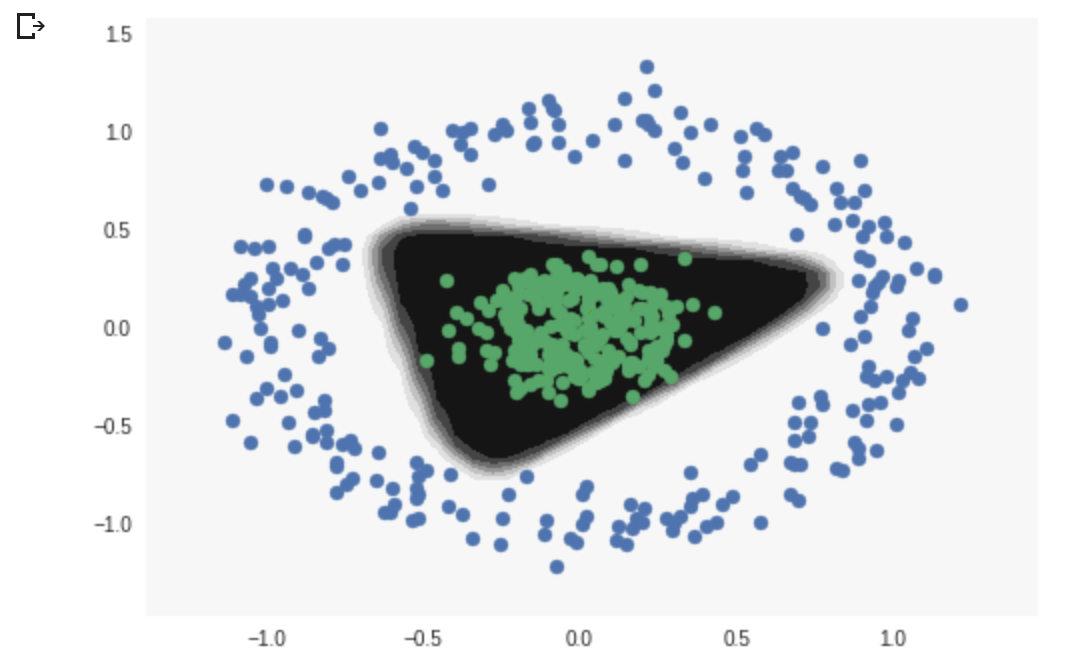
1. plt.plot(range(epochs), losses)
2. plt.ylabel('Loss')
3. plt.xlabel('epoch')

**Defining** **Plot Decision Boundary Function**

1. def plot\_decision\_boundary(X, y):
2. x\_span = np.linspace(min(X[:, 0]) -0.25, max(X[:, 0])+0.25)
3. y\_span = np.linspace(min(X[:, 1]) -0.25, max(X[:, 1])+0.25)
4. xx, yy = np.meshgrid(x\_span, y\_span)
5. grid = torch.Tensor(np.c\_[xx.ravel(), yy.ravel()])
6. pred\_func = model.forward(grid)
7. z = pred\_func.view(xx.shape).detach().numpy()
8. plt.contourf(xx, yy, z)

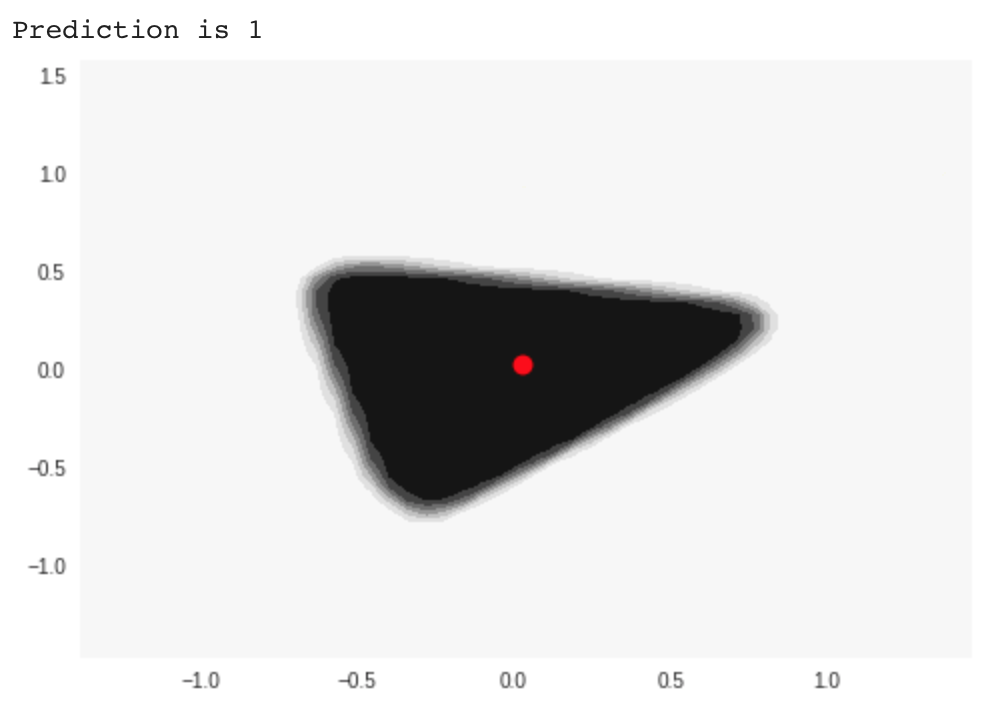
**Plot Decision Boundary Function**

1. plot\_decision\_boundary(X, y)
2. scatter\_plot()



**Making Prediction on Datapoint**

1. x = 0.025
2. y = 0.025
3. point = torch.Tensor([x, y])
4. prediction = model.predict(point)
5. plt.plot([x], [y], marker='o', markersize=10, color="red")
6. print("Prediction is", prediction)
7. plot\_decision\_boundary(X, y)



**Feel free to ignore the boiler plate code below. It was only there for debugging purposes.**

##   NO\_CAFFE2\_OPS#     disable Caffe2 operators build##   USE\_GLOO\_IBVERBS#     toggle features related to distributed support##   USE\_OPENCV#     enables use of OpenCV for additional operators##   USE\_FFMPEG#     enables use of ffmpeg for additional operators##   USE\_LEVELDB#     enables use of LevelDB for storage##   USE\_LMDB#     enables use of LMDB for storage##   BUILD\_BINARY#     enables the additional binaries/ build##   PYTORCH\_BUILD\_VERSION#   PYTORCH\_BUILD\_NUMBER#     specify the version of PyTorch, rather than the hard-coded version#     in this file; used when we're building binaries for distribution##   TORCH\_CUDA\_ARCH\_LIST#     specify which CUDA architectures to build for.#     ie `TORCH\_CUDA\_ARCH\_LIST="6.0;7.0"`#     These are not CUDA versions, instead, they specify what#     classes of NVIDIA hardware we should generate PTX for.##   ONNX\_NAMESPACE#     specify a namespace for ONNX built here rather than the hard-coded#     one in this file; needed to build with other frameworks that share ONNX.##   BLAS#     BLAS to be used by Caffe2. Can be MKL, Eigen, ATLAS, or OpenBLAS. If set#     then the build will fail if the requested BLAS is not found, otherwise#     the BLAS will be chosen based on what is found on your system.##   USE\_FBGEMM#     Enables use of FBGEMM##   USE\_REDIS#     Whether to use Redis for distributed workflows (Linux only)##   USE\_ZSTD#     Enables use of ZSTD, if the libraries are found## Environment variables we respect (these environment variables are# conventional and are often understood/set by other software.)##   CUDA\_HOME (Linux/OS X)#   CUDA\_PATH (Windows)#     specify where CUDA is installed; usually /usr/local/cuda or#     /usr/local/cuda-x.y#   CUDAHOSTCXX#     specify a different compiler than the system one to use as the CUDA#     host compiler for nvcc.##   CUDA\_NVCC\_EXECUTABLE#     Specify a NVCC to use. This is used in our CI to point to a cached nvcc##   CUDNN\_LIB\_DIR#   CUDNN\_INCLUDE\_DIR#   CUDNN\_LIBRARY#     specify where cuDNN is installed##   MIOPEN\_LIB\_DIR#   MIOPEN\_INCLUDE\_DIR#   MIOPEN\_LIBRARY#     specify where MIOpen is installed##   NCCL\_ROOT\_DIR#   NCCL\_LIB\_DIR#   NCCL\_INCLUDE\_DIR#     specify where nccl is installed##   NVTOOLSEXT\_PATH (Windows only)#     specify where nvtoolsext is installed##   LIBRARY\_PATH#   LD\_LIBRARY\_PATH#     we will search for libraries in these paths from \_\_future\_\_ import print\_functionfrom setuptools import setup, Extension, distutils, Command, find\_packagesimport setuptools.command.build\_extimport setuptools.command.installimport setuptools.command.developimport setuptools.command.build\_pyimport distutils.unixccompilerimport distutils.command.buildimport distutils.command.cleanimport distutils.sysconfigimport filecmpimport platformimport subprocessimport shutilimport multiprocessingimport sysimport osimport jsonimport globimport importlib # If you want to modify flags or environmental variables that is set when# building torch, you should do it in tools/setup\_helpers/configure.py.# Please don't add it here unless it's only used in PyTorch.from tools.setup\_helpers.configure import \*from tools.setup\_helpers.generate\_code import generate\_codefrom tools.setup\_helpers.ninja\_builder import NinjaBuilder, ninja\_build\_extimport tools.setup\_helpers.configure ################################################################################# Parameters parsed from environment################################################################################ VERBOSE\_SCRIPT = True# see if the user passed a quiet flag to setup.py arguments and respect# that in our parts of the buildfor arg in sys.argv:    if arg == "--":        break    if arg == '-q' or arg == '--quiet':        VERBOSE\_SCRIPT = False if VERBOSE\_SCRIPT:    def report(\*args):        print(\*args)else:    def report(\*args):        pass # Constant known variables used throughout this filecwd = os.path.dirname(os.path.abspath(\_\_file\_\_))lib\_path = os.path.join(cwd, "torch", "lib")third\_party\_path = os.path.join(cwd, "third\_party")tmp\_install\_path = lib\_path + "/tmp\_install"caffe2\_build\_dir = os.path.join(cwd, "build")# lib/pythonx.x/site-packagesrel\_site\_packages = distutils.sysconfig.get\_python\_lib(prefix='')# full absolute path to the dir abovefull\_site\_packages = distutils.sysconfig.get\_python\_lib()# CMAKE: full path to python libraryif IS\_WINDOWS:    cmake\_python\_library = "{}/libs/python{}.lib".format(        distutils.sysconfig.get\_config\_var("prefix"),        distutils.sysconfig.get\_config\_var("VERSION"))else:    cmake\_python\_library = "{}/{}".format(        distutils.sysconfig.get\_config\_var("LIBDIR"),        distutils.sysconfig.get\_config\_var("INSTSONAME"))cmake\_python\_include\_dir = distutils.sysconfig.get\_python\_inc()  class PytorchCommand(setuptools.Command):    """    Base Pytorch command to avoid implementing initialize/finalize\_options in    every subclass    """    user\_options = []     def initialize\_options(self):        pass     def finalize\_options(self):        pass  ################################################################################# Version, create\_version\_file, and package\_name################################################################################package\_name = os.getenv('TORCH\_PACKAGE\_NAME', 'torch')version = '1.1.0a0'if os.getenv('PYTORCH\_BUILD\_VERSION'):    assert os.getenv('PYTORCH\_BUILD\_NUMBER') is not None    build\_number = int(os.getenv('PYTORCH\_BUILD\_NUMBER'))    version = os.getenv('PYTORCH\_BUILD\_VERSION')    if build\_number > 1:        version += '.post' + str(build\_number)else:    try:        sha = subprocess.check\_output(['git', 'rev-parse', 'HEAD'], cwd=cwd).decode('ascii').strip()        version += '+' + sha[:7]    except Exception:        passreport("Building wheel {}-{}".format(package\_name, version))  class create\_version\_file(PytorchCommand):    def run(self):        global version, cwd        report('-- Building version ' + version)        version\_path = os.path.join(cwd, 'torch', 'version.py')        with open(version\_path, 'w') as f:            f.write("\_\_version\_\_ = '{}'\n".format(version))            # NB: This is not 100% accurate, because you could have built the            # library code with DEBUG, but csrc without DEBUG (in which case            # this would claim to be a release build when it's not.)            f.write("debug = {}\n".format(repr(DEBUG)))            f.write("cuda = {}\n".format(repr(CUDA\_VERSION)))  ################################################################################# Building dependent libraries################################################################################ # All libraries that torch could depend ondep\_libs = ['caffe2'] missing\_pydep = '''Missing build dependency: Unable to `import {importname}`.Please install it via `conda install {module}` or `pip install {module}`'''.strip()  def check\_pydep(importname, module):    try:        importlib.import\_module(importname)    except ImportError:        raise RuntimeError(missing\_pydep.format(importname=importname, module=module))  # Calls build\_pytorch\_libs.sh/bat with the correct env variablesdef build\_libs(libs):    for lib in libs:        assert lib in dep\_libs, 'invalid lib: {}'.format(lib)    if IS\_WINDOWS:        build\_libs\_cmd = ['tools\\build\_pytorch\_libs.bat']    else:        build\_libs\_cmd = ['bash', os.path.join('..', 'tools', 'build\_pytorch\_libs.sh')]     my\_env, extra\_flags = get\_pytorch\_env\_with\_flags()    build\_libs\_cmd.extend(extra\_flags)    my\_env["PYTORCH\_PYTHON\_LIBRARY"] = cmake\_python\_library    my\_env["PYTORCH\_PYTHON\_INCLUDE\_DIR"] = cmake\_python\_include\_dir    my\_env["PYTORCH\_BUILD\_VERSION"] = version     cmake\_prefix\_path = full\_site\_packages    if "CMAKE\_PREFIX\_PATH" in my\_env:        cmake\_prefix\_path = my\_env["CMAKE\_PREFIX\_PATH"] + ";" + cmake\_prefix\_path    my\_env["CMAKE\_PREFIX\_PATH"] = cmake\_prefix\_path     if VERBOSE\_SCRIPT:        my\_env['VERBOSE\_SCRIPT'] = '1'    try:        os.mkdir('build')    except OSError:        pass     kwargs = {'cwd': 'build'} if not IS\_WINDOWS else {}     if subprocess.call(build\_libs\_cmd + libs, env=my\_env, \*\*kwargs) != 0:        report("Failed to run '{}'".format(' '.join(build\_libs\_cmd + libs)))        sys.exit(1)  # Build all dependent librariesclass build\_deps(PytorchCommand):    def run(self):        report('setup.py::build\_deps::run()')        # Check if you remembered to check out submodules         def check\_file(f):            if not os.path.exists(f):                report("Could not find {}".format(f))                report("Did you run 'git submodule update --init --recursive'?")                sys.exit(1)         check\_file(os.path.join(third\_party\_path, "gloo", "CMakeLists.txt"))        check\_file(os.path.join(third\_party\_path, "pybind11", "CMakeLists.txt"))        check\_file(os.path.join(third\_party\_path, 'cpuinfo', 'CMakeLists.txt'))        check\_file(os.path.join(third\_party\_path, 'onnx', 'CMakeLists.txt'))        check\_file(os.path.join(third\_party\_path, 'QNNPACK', 'CMakeLists.txt'))        check\_file(os.path.join(third\_party\_path, 'fbgemm', 'CMakeLists.txt'))         check\_pydep('yaml', 'pyyaml')        check\_pydep('typing', 'typing')         libs = []        libs += ['caffe2']        build\_libs(libs)         # Use copies instead of symbolic files.        # Windows has very poor support for them.        sym\_files = ['tools/shared/cwrap\_common.py', 'tools/shared/\_utils\_internal.py']        orig\_files = ['aten/src/ATen/common\_with\_cwrap.py', 'torch/\_utils\_internal.py']        for sym\_file, orig\_file in zip(sym\_files, orig\_files):            same = False            if os.path.exists(sym\_file):                if filecmp.cmp(sym\_file, orig\_file):                    same = True                else:                    os.remove(sym\_file)            if not same:                shutil.copyfile(orig\_file, sym\_file)         self.copy\_tree('torch/lib/tmp\_install/share', 'torch/share')        self.copy\_tree('third\_party/pybind11/include/pybind11/',                       'torch/lib/include/pybind11')  build\_dep\_cmds = {}rebuild\_dep\_cmds = {} for lib in dep\_libs:    # wrap in function to capture lib    class build\_dep(build\_deps):        description = 'Build {} external library'.format(lib)         def run(self):            build\_libs([self.lib])    build\_dep.lib = lib    build\_dep\_cmds['build\_' + lib.lower()] = build\_dep     class rebuild\_dep(build\_deps):        description = 'Rebuild {} external library'.format(lib)         def run(self):            tools.setup\_helpers.configure.RERUN\_CMAKE = False            build\_libs([self.lib])    rebuild\_dep.lib = lib    rebuild\_dep\_cmds['rebuild\_' + lib.lower()] = rebuild\_dep  class build\_module(PytorchCommand):    def run(self):        report('setup.py::build\_module::run()')        self.run\_command('build\_py')        self.run\_command('build\_ext')  class build\_py(setuptools.command.build\_py.build\_py):     def run(self):        report('setup.py::build\_py::run()')        self.run\_command('create\_version\_file')        setuptools.command.build\_py.build\_py.run(self)  class develop(setuptools.command.develop.develop):     def run(self):        report('setup.py::develop::run()')        self.run\_command('create\_version\_file')        setuptools.command.develop.develop.run(self)        self.create\_compile\_commands()     def create\_compile\_commands(self):        def load(filename):            with open(filename) as f:                return json.load(f)        ninja\_files = glob.glob('build/\*compile\_commands.json')        cmake\_files = glob.glob('torch/lib/build/\*/compile\_commands.json')        all\_commands = [entry                        for f in ninja\_files + cmake\_files                        for entry in load(f)]         # cquery does not like c++ compiles that start with gcc.        # It forgets to include the c++ header directories.        # We can work around this by replacing the gcc calls that python        # setup.py generates with g++ calls instead        for command in all\_commands:            if command['command'].startswith("gcc "):                command['command'] = "g++ " + command['command'][4:]         new\_contents = json.dumps(all\_commands, indent=2)        contents = ''        if os.path.exists('compile\_commands.json'):            with open('compile\_commands.json', 'r') as f:                contents = f.read()        if contents != new\_contents:            with open('compile\_commands.json', 'w') as f:                f.write(new\_contents)         if not USE\_NINJA:            report("WARNING: 'develop' is not building C++ code incrementally")            report("because ninja is not installed. Run this to enable it:")            report(" > pip install ninja")  build\_ext\_parent = ninja\_build\_ext if USE\_NINJA \    else setuptools.command.build\_ext.build\_ext  class build\_ext(build\_ext\_parent):     def run(self):        # report build options        if USE\_NUMPY:            report('-- Building with NumPy bindings')        else:            report('-- NumPy not found')        if USE\_CUDNN:            report('-- Detected cuDNN at ' + CUDNN\_LIBRARY + ', ' + CUDNN\_INCLUDE\_DIR)        else:            report('-- Not using cuDNN')        if USE\_MIOPEN:            report('-- Detected MIOpen at ' + MIOPEN\_LIBRARY + ', ' + MIOPEN\_INCLUDE\_DIR)        else:            report('-- Not using MIOpen')        if USE\_CUDA:            report('-- Detected CUDA at ' + CUDA\_HOME)        else:            report('-- Not using CUDA')        if USE\_MKLDNN:            report('-- Using MKLDNN')        else:            report('-- Not using MKLDNN')        if USE\_NCCL and USE\_SYSTEM\_NCCL:            report('-- Using system provided NCCL library at ' + NCCL\_SYSTEM\_LIB + ', ' + NCCL\_INCLUDE\_DIR)        elif USE\_NCCL:            report('-- Building NCCL library')        else:            report('-- Not using NCCL')        if USE\_DISTRIBUTED:            report('-- Building with THD distributed package ')            if IS\_LINUX:                report('-- Building with c10d distributed package ')            else:                report('-- Building without c10d distributed package')        else:            report('-- Building without distributed package')         # It's an old-style class in Python 2.7...        setuptools.command.build\_ext.build\_ext.run(self)         # Copy the essential export library to compile C++ extensions.        if IS\_WINDOWS:            build\_temp = self.build\_temp             ext\_filename = self.get\_ext\_filename('\_C')            lib\_filename = '.'.join(ext\_filename.split('.')[:-1]) + '.lib'             export\_lib = os.path.join(                build\_temp, 'torch', 'csrc', lib\_filename).replace('\\', '/')             build\_lib = self.build\_lib             target\_lib = os.path.join(                build\_lib, 'torch', 'lib', '\_C.lib').replace('\\', '/')             self.copy\_file(export\_lib, target\_lib)     def build\_extensions(self):        # The caffe2 extensions are created in        # tmp\_install/lib/pythonM.m/site-packages/caffe2/python/        # and need to be copied to build/lib.linux.... , which will be a        # platform dependent build folder created by the "build" command of        # setuptools. Only the contents of this folder are installed in the        # "install" command by default.        # We only make this copy for Caffe2's pybind extensions        caffe2\_pybind\_exts = [            'caffe2.python.caffe2\_pybind11\_state',            'caffe2.python.caffe2\_pybind11\_state\_gpu',            'caffe2.python.caffe2\_pybind11\_state\_hip',        ]        i = 0        while i < len(self.extensions):            ext = self.extensions[i]            if ext.name not in caffe2\_pybind\_exts:                i += 1                continue            fullname = self.get\_ext\_fullname(ext.name)            filename = self.get\_ext\_filename(fullname)            report("\nCopying extension {}".format(ext.name))             src = os.path.join(tmp\_install\_path, rel\_site\_packages, filename)            if not os.path.exists(src):                report("{} does not exist".format(src))                del self.extensions[i]            else:                dst = os.path.join(os.path.realpath(self.build\_lib), filename)                report("Copying {} from {} to {}".format(ext.name, src, dst))                dst\_dir = os.path.dirname(dst)                if not os.path.exists(dst\_dir):                    os.makedirs(dst\_dir)                self.copy\_file(src, dst)                i += 1        distutils.command.build\_ext.build\_ext.build\_extensions(self)     def get\_outputs(self):        outputs = distutils.command.build\_ext.build\_ext.get\_outputs(self)        outputs.append(os.path.join(self.build\_lib, "caffe2"))        report("setup.py::get\_outputs returning {}".format(outputs))        return outputs  class build(distutils.command.build.build):    sub\_commands = [        ('build\_deps', lambda self: True),    ] + distutils.command.build.build.sub\_commands  class rebuild(distutils.command.build.build):    sub\_commands = [        ('build\_deps', lambda self: True),    ] + distutils.command.build.build.sub\_commands     def run(self):        tools.setup\_helpers.configure.RERUN\_CMAKE = False        distutils.command.build.build.run(self)  class install(setuptools.command.install.install):     def run(self):        report('setup.py::run()')        if not self.skip\_build:            self.run\_command('build\_deps')         setuptools.command.install.install.run(self)  class clean(distutils.command.clean.clean):    def run(self):        import glob        import re        with open('.gitignore', 'r') as f:            ignores = f.read()            pat = re.compile(r'^#( BEGIN NOT-CLEAN-FILES )?')            for wildcard in filter(None, ignores.split('\n')):                match = pat.match(wildcard)                if match:                    if match.group(1):                        # Marker is found and stop reading .gitignore.                        break                    # Ignore lines which begin with '#'.                else:                    for filename in glob.glob(wildcard):                        try:                            os.remove(filename)                        except OSError:                            shutil.rmtree(filename, ignore\_errors=True)         # It's an old-style class in Python 2.7...        distutils.command.clean.clean.run(self)  ################################################################################# Configure compile flags################################################################################ library\_dirs = [] if IS\_WINDOWS:    # /NODEFAULTLIB makes sure we only link to DLL runtime    # and matches the flags set for protobuf and ONNX    extra\_link\_args = ['/NODEFAULTLIB:LIBCMT.LIB']    # /MD links against DLL runtime    # and matches the flags set for protobuf and ONNX    # /Z7 turns on symbolic debugging information in .obj files    # /EHa is about native C++ catch support for asynchronous    # structured exception handling (SEH)    # /DNOMINMAX removes builtin min/max functions    # /wdXXXX disables warning no. XXXX    extra\_compile\_args = ['/MD', '/Z7',                          '/EHa', '/DNOMINMAX',                          '/wd4267', '/wd4251', '/wd4522', '/wd4522', '/wd4838',                          '/wd4305', '/wd4244', '/wd4190', '/wd4101', '/wd4996',                          '/wd4275']    if sys.version\_info[0] == 2:        if not check\_env\_flag('FORCE\_PY27\_BUILD'):            report('The support for PyTorch with Python 2.7 on Windows is very experimental.')            report('Please set the flag `FORCE\_PY27\_BUILD` to 1 to continue build.')            sys.exit(1)        # /bigobj increases number of sections in .obj file, which is needed to link        # against libaries in Python 2.7 under Windows        extra\_compile\_args.append('/bigobj')else:    extra\_link\_args = []    extra\_compile\_args = [        '-std=c++11',        '-Wall',        '-Wextra',        '-Wno-strict-overflow',        '-Wno-unused-parameter',        '-Wno-missing-field-initializers',        '-Wno-write-strings',        '-Wno-unknown-pragmas',        # This is required for Python 2 declarations that are deprecated in 3.        '-Wno-deprecated-declarations',        # Python 2.6 requires -fno-strict-aliasing, see        # http://legacy.python.org/dev/peps/pep-3123/        # We also depend on it in our code (even Python 3).        '-fno-strict-aliasing',        # Clang has an unfixed bug leading to spurious missing        # braces warnings, see        # https://bugs.llvm.org/show\_bug.cgi?id=21629        '-Wno-missing-braces',    ]    if check\_env\_flag('WERROR'):        extra\_compile\_args.append('-Werror') library\_dirs.append(lib\_path) # we specify exact lib names to avoid conflict with lua-torch installsCAFFE2\_LIBS = []if USE\_CUDA:    CAFFE2\_LIBS.extend(['-Wl,--no-as-needed', os.path.join(lib\_path, 'libcaffe2\_gpu.so'), '-Wl,--as-needed'])if USE\_ROCM:    CAFFE2\_LIBS.extend(['-Wl,--no-as-needed', os.path.join(lib\_path, 'libcaffe2\_hip.so'), '-Wl,--as-needed']) # static library onlyif IS\_DARWIN:    CAFFE2\_LIBS = []    if USE\_CUDA:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'libcaffe2\_gpu.dylib'))    if USE\_ROCM:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'libcaffe2\_hip.dylib')) if IS\_WINDOWS:    CAFFE2\_LIBS = []    if USE\_CUDA:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'caffe2\_gpu.lib'))    if USE\_ROCM:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'caffe2\_hip.lib')) main\_compile\_args = ['-D\_THP\_CORE', '-DONNX\_NAMESPACE=' + ONNX\_NAMESPACE]main\_libraries = ['shm', 'torch\_python']main\_link\_args = []main\_sources = ["torch/csrc/stub.cpp"] # Before the introduction of stub.cpp, \_C.so and libcaffe2.so defined# some of the same symbols, and it was important for \_C.so to be# loaded before libcaffe2.so so that the versions in \_C.so got# used. This happened automatically because we loaded \_C.so directly,# and libcaffe2.so was brought in as a dependency (though I suspect it# may have been possible to break by importing caffe2 first in the# same process).## Now, libtorch\_python.so and libcaffe2.so define some of the same# symbols. We directly load the \_C.so stub, which brings both of these# in as dependencies. We have to make sure that symbols continue to be# looked up in libtorch\_python.so first, by making sure it comes# before libcaffe2.so in the linker command.main\_link\_args.extend(CAFFE2\_LIBS) try:    import numpy as np    NUMPY\_INCLUDE\_DIR = np.get\_include()    USE\_NUMPY = Trueexcept ImportError:    USE\_NUMPY = False if USE\_CUDA:    if IS\_WINDOWS:        cuda\_lib\_path = CUDA\_HOME + '/lib/x64/'    else:        cuda\_lib\_dirs = ['lib64', 'lib']        for lib\_dir in cuda\_lib\_dirs:            cuda\_lib\_path = os.path.join(CUDA\_HOME, lib\_dir)            if os.path.exists(cuda\_lib\_path):                break    library\_dirs.append(cuda\_lib\_path) if DEBUG:    if IS\_WINDOWS:        extra\_link\_args.append('/DEBUG:FULL')    else:        extra\_compile\_args += ['-O0', '-g']        extra\_link\_args += ['-O0', '-g'] if REL\_WITH\_DEB\_INFO:    if IS\_WINDOWS:        extra\_link\_args.append('/DEBUG:FULL')    else:        extra\_compile\_args += ['-g']        extra\_link\_args += ['-g']  def make\_relative\_rpath(path):    if IS\_DARWIN:        return '-Wl,-rpath,@loader\_path/' + path    elif IS\_WINDOWS:        return ''    else:        return '-Wl,-rpath,$ORIGIN/' + path ################################################################################# Declare extensions and package################################################################################ extensions = []packages = find\_packages(exclude=('tools', 'tools.\*'))C = Extension("torch.\_C",              libraries=main\_libraries,              sources=main\_sources,              language='c++',              extra\_compile\_args=main\_compile\_args + extra\_compile\_args,              include\_dirs=[],              library\_dirs=library\_dirs,              extra\_link\_args=extra\_link\_args + main\_link\_args + [make\_relative\_rpath('lib')],              )extensions.append(C) if not IS\_WINDOWS:    DL = Extension("torch.\_dl",                   sources=["torch/csrc/dl.c"],                   language='c'                   )    extensions.append(DL)  if USE\_CUDA:    thnvrtc\_link\_flags = extra\_link\_args + [make\_relative\_rpath('lib')]    if IS\_LINUX:        thnvrtc\_link\_flags = thnvrtc\_link\_flags + ['-Wl,--no-as-needed']    # these have to be specified as -lcuda in link\_flags because they    # have to come right after the `no-as-needed` option    if IS\_WINDOWS:        thnvrtc\_link\_flags += ['cuda.lib', 'nvrtc.lib']    else:        thnvrtc\_link\_flags += ['-lcuda', '-lnvrtc']    cuda\_stub\_path = [cuda\_lib\_path + '/stubs']    if IS\_DARWIN:        # on macOS this is where the CUDA stub is installed according to the manual        cuda\_stub\_path = ["/usr/local/cuda/lib"]    THNVRTC = Extension("torch.\_nvrtc",                        sources=['torch/csrc/nvrtc.cpp'],                        language='c++',                        extra\_compile\_args=main\_compile\_args + extra\_compile\_args,                        include\_dirs=[cwd],                        library\_dirs=library\_dirs + cuda\_stub\_path,                        extra\_link\_args=thnvrtc\_link\_flags,                        )    extensions.append(THNVRTC) # These extensions are built by cmake and copied manually in build\_extensions()# inside the build\_ext implementaitonextensions.append(    Extension(        name=str('caffe2.python.caffe2\_pybind11\_state'),        sources=[]),)if USE\_CUDA:    extensions.append(        Extension(            name=str('caffe2.python.caffe2\_pybind11\_state\_gpu'),            sources=[]),    )if USE\_ROCM:    extensions.append(        Extension(            name=str('caffe2.python.caffe2\_pybind11\_state\_hip'),            sources=[]),    ) cmdclass = {    'create\_version\_file': create\_version\_file,    'build': build,    'build\_py': build\_py,    'build\_ext': build\_ext,    'build\_deps': build\_deps,    'build\_module': build\_module,    'rebuild': rebuild,    'develop': develop,    'install': install,    'clean': clean,}cmdclass.update(build\_dep\_cmds)cmdclass.update(rebuild\_dep\_cmds) entry\_points = {    'console\_scripts': [        'convert-caffe2-to-onnx = caffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'caffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inccaffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',

'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inccaffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inclib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/include/pybind11/\*.h',                'lib/include/pybind11/detail/\*.h',                'lib/include/TH/\*.h\*',                'lib/include/TH/generic/\*.h\*',                'lib/include/THC/\*.cuh',                'lib/include/THC/\*.h\*',                'lib/include/THC/generic/\*.h',                'lib/include/THCUNN/\*.cuh',                'lib/include/THNN/\*.h',                'share/cmake/ATen/\*.cmake',                'share/cmake/Caffe2/\*.cmake',                'share/cmake/Caffe2/public/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/FindCUDA/\*.cmake',                'share/cmake/Gloo/\*.cmake',                'share/cmake/Torch/\*.cmake',            ],            'caffe2': [                'cpp\_test/\*',                'python/serialized\_test/data/operator\_test/\*.zip',            ]        },    )

'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inclib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/include/pybind11/\*.h',                'lib/include/pybind11/detail/\*.h',                'lib/include/TH/\*.h\*',                'lib/include/TH/generic/\*.h\*',                'lib/include/THC/\*.cuh',                'lib/include/THC/\*.h\*',                'lib/include/THC/generic/\*.h',                'lib/include/THCUNN/\*.cuh',                'lib/include/THNN/\*.h',                'share/cmake/ATen/\*.cmake',                'share/cmake/Caffe2/\*.cmake',                'share/cmake/Caffe2/public/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/FindCUDA/\*.cmake',                'share/cmake/Gloo/\*.cmake',                'share/cmake/Torch/\*.cmake',            ],            'caffe2': [                'cpp\_test/\*',                'python/serialized\_test/data/operator\_test/\*.zip',            ]        },    )

VERBOSE\_SCRIPT:    def report(\*args):        print(\*args)else:    def report(\*args):        pass # Constant known variables used throughout this filecwd = os.path.dirname(os.path.abspath(\_\_file\_\_))lib\_path = os.path.join(cwd, "torch", "lib")third\_party\_path = os.path.join(cwd, "third\_party")tmp\_install\_path = lib\_path + "/tmp\_install"caffe2\_build\_dir = os.path.join(cwd, "build")# lib/pythonx.x/site-packagesrel\_site\_packages = distutils.sysconfig.get\_python\_lib(prefix='')# full absolute path to the dir abovefull\_site\_packages = distutils.sysconfig.get\_python\_lib()# CMAKE: full path to python libraryif IS\_WINDOWS:    cmake\_python\_library = "{}/libs/python{}.lib".format(        distutils.sysconfig.get\_config\_var("prefix"),        distutils.sysconfig.get\_config\_var("VERSION"))else:    cmake\_python\_library = "{}/{}".format(        distutils.sysconfig.get\_config\_var("LIBDIR"),        distutils.sysconfig.get\_config\_var("INSTSONAME"))cmake\_python\_include\_dir = distutils.sysconfig.get\_python\_inc()  class PytorchCommand(setuptools.Command):    """    Base Pytorch command to avoid implementing initialize/finalize\_options in    every subclass    """    user\_options = []     def initialize\_options(self):        pass     def finalize\_options(self):        pass  ################################################################################# Version, create\_version\_file, and package\_name################################################################################package\_name = os.getenv('TORCH\_PACKAGE\_NAME', 'torch')version = '1.1.0a0'if os.getenv('PYTORCH\_BUILD\_VERSION'):    assert os.getenv('PYTORCH\_BUILD\_NUMBER') is not None    build\_number = int(os.getenv('PYTORCH\_BUILD\_NUMBER'))    version = os.getenv('PYTORCH\_BUILD\_VERSION')    if build\_number > 1:        version += '.post' + str(build\_number)else:    try:        sha = subprocess.check\_output(['git', 'rev-parse', 'HEAD'], cwd=cwd).decode('ascii').strip()        version += '+' + sha[:7]    except Exception:        passreport("Building wheel {}-{}".format(package\_name, version))  class create\_version\_file(PytorchCommand):    def run(self):        global version, cwd        report('-- Building version ' + version)        version\_path = os.path.join(cwd, 'torch', 'version.py')        with open(version\_path, 'w') as f:            f.write("\_\_version\_\_ = '{}'\n".format(version))            # NB: This is not 100% accurate, because you could have built the            # library code with DEBUG, but csrc without DEBUG (in which case            # this would claim to be a release build when it's not.)            f.write("debug = {}\n".format(repr(DEBUG)))            f.write("cuda = {}\n".format(repr(CUDA\_VERSION)))  ################################################################################# Building dependent libraries################################################################################ # All libraries that torch could depend ondep\_libs = ['caffe2'] missing\_pydep = '''Missing build dependency: Unable to `import {importname}`.Please install it via `conda install {module}` or `pip install {module}`'''.strip()  def check\_pydep(importname, module):    try:        importlib.import\_module(importname)    except ImportError:        raise RuntimeError(missing\_pydep.format(importname=importname, module=module))  # Calls build\_pytorch\_libs.sh/bat with the correct env variablesdef build\_libs(libs):    for lib in libs:        assert lib in dep\_libs, 'invalid lib: {}'.format(lib)    if IS\_WINDOWS:        build\_libs\_cmd = ['tools\\build\_pytorch\_libs.bat']    else:        build\_libs\_cmd = ['bash', os.path.join('..', 'tools', 'build\_pytorch\_libs.sh')]     my\_env, extra\_flags = get\_pytorch\_env\_with\_flags()    build\_libs\_cmd.extend(extra\_flags)    my\_env["PYTORCH\_PYTHON\_LIBRARY"] = cmake\_python\_library    my\_env["PYTORCH\_PYTHON\_INCLUDE\_DIR"] = cmake\_python\_include\_dir    my\_env["PYTORCH\_BUILD\_VERSION"] = version     cmake\_prefix\_path = full\_site\_packages    if "CMAKE\_PREFIX\_PATH" in my\_env:        cmake\_prefix\_path = my\_env["CMAKE\_PREFIX\_PATH"] + ";" + cmake\_prefix\_path    my\_env["CMAKE\_PREFIX\_PATH"] = cmake\_prefix\_path     if VERBOSE\_SCRIPT:        my\_env['VERBOSE\_SCRIPT'] = '1'    try:        os.mkdir('build')    except OSError:        pass     kwargs = {'cwd': 'build'} if not IS\_WINDOWS else {}     if subprocess.call(build\_libs\_cmd + libs, env=my\_env, \*\*kwargs) != 0:        report("Failed to run '{}'".format(' '.join(build\_libs\_cmd + libs)))        sys.exit(1)  # Build all dependent librariesclass build\_deps(PytorchCommand):    def run(self):        report('setup.py::build\_deps::run()')        # Check if you remembered to check out submodules         def check\_file(f):            if not os.path.exists(f):                report("Could not find {}".format(f))                report("Did you run 'git submodule update --init --recursive'?")                sys.exit(1)         check\_file(os.path.join(third\_party\_path, "gloo", "CMakeLists.txt"))        check\_file(os.path.join(third\_party\_path, "pybind11", "CMakeLists.txt"))        check\_file(os.path.join(third\_party\_path, 'cpuinfo', 'CMakeLists.txt'))        check\_file(os.path.join(third\_party\_path, 'onnx', 'CMakeLists.txt'))        check\_file(os.path.join(third\_party\_path, 'QNNPACK', 'CMakeLists.txt'))        check\_file(os.path.join(third\_party\_path, 'fbgemm', 'CMakeLists.txt'))         check\_pydep('yaml', 'pyyaml')        check\_pydep('typing', 'typing')         libs = []        libs += ['caffe2']        build\_libs(libs)         # Use copies instead of symbolic files.        # Windows has very poor support for them.        sym\_files = ['tools/shared/cwrap\_common.py', 'tools/shared/\_utils\_internal.py']        orig\_files = ['aten/src/ATen/common\_with\_cwrap.py', 'torch/\_utils\_internal.py']        for sym\_file, orig\_file in zip(sym\_files, orig\_files):            same = False            if os.path.exists(sym\_file):                if filecmp.cmp(sym\_file, orig\_file):                    same = True                else:                    os.remove(sym\_file)            if not same:                shutil.copyfile(orig\_file, sym\_file)         self.copy\_tree('torch/lib/tmp\_install/share', 'torch/share')        self.copy\_tree('third\_party/pybind11/include/pybind11/',                       'torch/lib/include/pybind11')  build\_dep\_cmds = {}rebuild\_dep\_cmds = {} for lib in dep\_libs:    # wrap in function to capture lib    class build\_dep(build\_deps):        description = 'Build {} external library'.format(lib)         def run(self):            build\_libs([self.lib])    build\_dep.lib = lib    build\_dep\_cmds['build\_' + lib.lower()] = build\_dep     class rebuild\_dep(build\_deps):        description = 'Rebuild {} external library'.format(lib)         def run(self):            tools.setup\_helpers.configure.RERUN\_CMAKE = False            build\_libs([self.lib])    rebuild\_dep.lib = lib    rebuild\_dep\_cmds['rebuild\_' + lib.lower()] = rebuild\_dep  class build\_module(PytorchCommand):    def run(self):        report('setup.py::build\_module::run()')        self.run\_command('build\_py')        self.run\_command('build\_ext')  class build\_py(setuptools.command.build\_py.build\_py):     def run(self):        report('setup.py::build\_py::run()')        self.run\_command('create\_version\_file')        setuptools.command.build\_py.build\_py.run(self)  class develop(setuptools.command.develop.develop):     def run(self):        report('setup.py::develop::run()')        self.run\_command('create\_version\_file')        setuptools.command.develop.develop.run(self)        self.create\_compile\_commands()     def create\_compile\_commands(self):        def load(filename):            with open(filename) as f:                return json.load(f)        ninja\_files = glob.glob('build/\*compile\_commands.json')        cmake\_files = glob.glob('torch/lib/build/\*/compile\_commands.json')        all\_commands = [entry                        for f in ninja\_files + cmake\_files                        for entry in load(f)]         # cquery does not like c++ compiles that start with gcc.        # It forgets to include the c++ header directories.        # We can work around this by replacing the gcc calls that python        # setup.py generates with g++ calls instead        for command in all\_commands:            if command['command'].startswith("gcc "):                command['command'] = "g++ " + command['command'][4:]         new\_contents = json.dumps(all\_commands, indent=2)        contents = ''        if os.path.exists('compile\_commands.json'):            with open('compile\_commands.json', 'r') as f:                contents = f.read()        if contents != new\_contents:            with open('compile\_commands.json', 'w') as f:                f.write(new\_contents)         if not USE\_NINJA:            report("WARNING: 'develop' is not building C++ code incrementally")            report("because ninja is not installed. Run this to enable it:")            report(" > pip install ninja")  build\_ext\_parent = ninja\_build\_ext if USE\_NINJA \    else setuptools.command.build\_ext.build\_ext  class build\_ext(build\_ext\_parent):     def run(self):        # report build options        if USE\_NUMPY:            report('-- Building with NumPy bindings')        else:            report('-- NumPy not found')        if USE\_CUDNN:            report('-- Detected cuDNN at ' + CUDNN\_LIBRARY + ', ' + CUDNN\_INCLUDE\_DIR)        else:            report('-- Not using cuDNN')        if USE\_MIOPEN:            report('-- Detected MIOpen at ' + MIOPEN\_LIBRARY + ', ' + MIOPEN\_INCLUDE\_DIR)        else:            report('-- Not using MIOpen')        if USE\_CUDA:            report('-- Detected CUDA at ' + CUDA\_HOME)        else:            report('-- Not using CUDA')        if USE\_MKLDNN:            report('-- Using MKLDNN')        else:            report('-- Not using MKLDNN')        if USE\_NCCL and USE\_SYSTEM\_NCCL:            report('-- Using system provided NCCL library at ' + NCCL\_SYSTEM\_LIB + ', ' + NCCL\_INCLUDE\_DIR)        elif USE\_NCCL:            report('-- Building NCCL library')        else:            report('-- Not using NCCL')        if USE\_DISTRIBUTED:            report('-- Building with THD distributed package ')            if IS\_LINUX:                report('-- Building with c10d distributed package ')            else:                report('-- Building without c10d distributed package')        else:            report('-- Building without distributed package')         # It's an old-style class in Python 2.7...        setuptools.command.build\_ext.build\_ext.run(self)         # Copy the essential export library to compile C++ extensions.        if IS\_WINDOWS:            build\_temp = self.build\_temp             ext\_filename = self.get\_ext\_filename('\_C')            lib\_filename = '.'.join(ext\_filename.split('.')[:-1]) + '.lib'             export\_lib = os.path.join(                build\_temp, 'torch', 'csrc', lib\_filename).replace('\\', '/')             build\_lib = self.build\_lib             target\_lib = os.path.join(                build\_lib, 'torch', 'lib', '\_C.lib').replace('\\', '/')             self.copy\_file(export\_lib, target\_lib)     def build\_extensions(self):        # The caffe2 extensions are created in        # tmp\_install/lib/pythonM.m/site-packages/caffe2/python/        # and need to be copied to build/lib.linux.... , which will be a        # platform dependent build folder created by the "build" command of        # setuptools. Only the contents of this folder are installed in the        # "install" command by default.        # We only make this copy for Caffe2's pybind extensions        caffe2\_pybind\_exts = [            'caffe2.python.caffe2\_pybind11\_state',            'caffe2.python.caffe2\_pybind11\_state\_gpu',            'caffe2.python.caffe2\_pybind11\_state\_hip',        ]        i = 0        while i < len(self.extensions):            ext = self.extensions[i]            if ext.name not in caffe2\_pybind\_exts:                i += 1                continue            fullname = self.get\_ext\_fullname(ext.name)            filename = self.get\_ext\_filename(fullname)            report("\nCopying extension {}".format(ext.name))             src = os.path.join(tmp\_install\_path, rel\_site\_packages, filename)            if not os.path.exists(src):                report("{} does not exist".format(src))                del self.extensions[i]            else:                dst = os.path.join(os.path.realpath(self.build\_lib), filename)                report("Copying {} from {} to {}".format(ext.name, src, dst))                dst\_dir = os.path.dirname(dst)                if not os.path.exists(dst\_dir):                    os.makedirs(dst\_dir)                self.copy\_file(src, dst)                i += 1        distutils.command.build\_ext.build\_ext.build\_extensions(self)     def get\_outputs(self):        outputs = distutils.command.build\_ext.build\_ext.get\_outputs(self)        outputs.append(os.path.join(self.build\_lib, "caffe2"))        report("setup.py::get\_outputs returning {}".format(outputs))        return outputs  class build(distutils.command.build.build):    sub\_commands = [        ('build\_deps', lambda self: True),    ] + distutils.command.build.build.sub\_commands  class rebuild(distutils.command.build.build):    sub\_commands = [        ('build\_deps', lambda self: True),    ] + distutils.command.build.build.sub\_commands     def run(self):        tools.setup\_helpers.configure.RERUN\_CMAKE = False        distutils.command.build.build.run(self)  class install(setuptools.command.install.install):     def run(self):        report('setup.py::run()')        if not self.skip\_build:            self.run\_command('build\_deps')         setuptools.command.install.install.run(self)  class clean(distutils.command.clean.clean):    def run(self):        import glob        import re        with open('.gitignore', 'r') as f:            ignores = f.read()            pat = re.compile(r'^#( BEGIN NOT-CLEAN-FILES )?')            for wildcard in filter(None, ignores.split('\n')):                match = pat.match(wildcard)                if match:                    if match.group(1):                        # Marker is found and stop reading .gitignore.                        break                    # Ignore lines which begin with '#'.                else:                    for filename in glob.glob(wildcard):                        try:                            os.remove(filename)                        except OSError:                            shutil.rmtree(filename, ignore\_errors=True)         # It's an old-style class in Python 2.7...        distutils.command.clean.clean.run(self)  ################################################################################# Configure compile flags################################################################################ library\_dirs = [] if IS\_WINDOWS:    # /NODEFAULTLIB makes sure we only link to DLL runtime    # and matches the flags set for protobuf and ONNX    extra\_link\_args = ['/NODEFAULTLIB:LIBCMT.LIB']    # /MD links against DLL runtime    # and matches the flags set for protobuf and ONNX    # /Z7 turns on symbolic debugging information in .obj files    # /EHa is about native C++ catch support for asynchronous    # structured exception handling (SEH)    # /DNOMINMAX removes builtin min/max functions    # /wdXXXX disables warning no. XXXX    extra\_compile\_args = ['/MD', '/Z7',                          '/EHa', '/DNOMINMAX',                          '/wd4267', '/wd4251', '/wd4522', '/wd4522', '/wd4838',                          '/wd4305', '/wd4244', '/wd4190', '/wd4101', '/wd4996',                          '/wd4275']    if sys.version\_info[0] == 2:        if not check\_env\_flag('FORCE\_PY27\_BUILD'):            report('The support for PyTorch with Python 2.7 on Windows is very experimental.')            report('Please set the flag `FORCE\_PY27\_BUILD` to 1 to continue build.')            sys.exit(1)        # /bigobj increases number of sections in .obj file, which is needed to link        # against libaries in Python 2.7 under Windows        extra\_compile\_args.append('/bigobj')else:    extra\_link\_args = []    extra\_compile\_args = [        '-std=c++11',        '-Wall',        '-Wextra',        '-Wno-strict-overflow',        '-Wno-unused-parameter',        '-Wno-missing-field-initializers',        '-Wno-write-strings',        '-Wno-unknown-pragmas',        # This is required for Python 2 declarations that are deprecated in 3.        '-Wno-deprecated-declarations',        # Python 2.6 requires -fno-strict-aliasing, see        # http://legacy.python.org/dev/peps/pep-3123/        # We also depend on it in our code (even Python 3).        '-fno-strict-aliasing',        # Clang has an unfixed bug leading to spurious missing        # braces warnings, see        # https://bugs.llvm.org/show\_bug.cgi?id=21629        '-Wno-missing-braces',    ]    if check\_env\_flag('WERROR'):        extra\_compile\_args.append('-Werror') library\_dirs.append(lib\_path) # we specify exact lib names to avoid conflict with lua-torch installsCAFFE2\_LIBS = []if USE\_CUDA:    CAFFE2\_LIBS.extend(['-Wl,--no-as-needed', os.path.join(lib\_path, 'libcaffe2\_gpu.so'), '-Wl,--as-needed'])if USE\_ROCM:    CAFFE2\_LIBS.extend(['-Wl,--no-as-needed', os.path.join(lib\_path, 'libcaffe2\_hip.so'), '-Wl,--as-needed']) # static library onlyif IS\_DARWIN:    CAFFE2\_LIBS = []    if USE\_CUDA:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'libcaffe2\_gpu.dylib'))    if USE\_ROCM:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'libcaffe2\_hip.dylib')) if IS\_WINDOWS:    CAFFE2\_LIBS = []    if USE\_CUDA:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'caffe2\_gpu.lib'))    if USE\_ROCM:        CAFFE2\_LIBS.append(os.path.join(lib\_path, 'caffe2\_hip.lib')) main\_compile\_args = ['-D\_THP\_CORE', '-DONNX\_NAMESPACE=' + ONNX\_NAMESPACE]main\_libraries = ['shm', 'torch\_python']main\_link\_args = []main\_sources = ["torch/csrc/stub.cpp"] # Before the introduction of stub.cpp, \_C.so and libcaffe2.so defined# some of the same symbols, and it was important for \_C.so to be# loaded before libcaffe2.so so that the versions in \_C.so got# used. This happened automatically because we loaded \_C.so directly,# and libcaffe2.so was brought in as a dependency (though I suspect it# may have been possible to break by importing caffe2 first in the# same process).## Now, libtorch\_python.so and libcaffe2.so define some of the same# symbols. We directly load the \_C.so stub, which brings both of these# in as dependencies. We have to make sure that symbols continue to be# looked up in libtorch\_python.so first, by making sure it comes# before libcaffe2.so in the linker command.main\_link\_args.extend(CAFFE2\_LIBS) try:    import numpy as np    NUMPY\_INCLUDE\_DIR = np.get\_include()    USE\_NUMPY = Trueexcept ImportError:    USE\_NUMPY = False if USE\_CUDA:    if IS\_WINDOWS:        cuda\_lib\_path = CUDA\_HOME + '/lib/x64/'    else:        cuda\_lib\_dirs = ['lib64', 'lib']        for lib\_dir in cuda\_lib\_dirs:            cuda\_lib\_path = os.path.join(CUDA\_HOME, lib\_dir)            if os.path.exists(cuda\_lib\_path):                break    library\_dirs.append(cuda\_lib\_path) if DEBUG:    if IS\_WINDOWS:        extra\_link\_args.append('/DEBUG:FULL')    else:        extra\_compile\_args += ['-O0', '-g']        extra\_link\_args += ['-O0', '-g'] if REL\_WITH\_DEB\_INFO:    if IS\_WINDOWS:        extra\_link\_args.append('/DEBUG:FULL')    else:        extra\_compile\_args += ['-g']        extra\_link\_args += ['-g']  def make\_relative\_rpath(path):    if IS\_DARWIN:        return '-Wl,-rpath,@loader\_path/' + path    elif IS\_WINDOWS:        return ''    else:        return '-Wl,-rpath,$ORIGIN/' + path ################################################################################# Declare extensions and package################################################################################ extensions = []packages = find\_packages(exclude=('tools', 'tools.\*'))C = Extension("torch.\_C",              libraries=main\_libraries,              sources=main\_sources,              language='c++',              extra\_compile\_args=main\_compile\_args + extra\_compile\_args,              include\_dirs=[],              library\_dirs=library\_dirs,              extra\_link\_args=extra\_link\_args + main\_link\_args + [make\_relative\_rpath('lib')],              )extensions.append(C) if not IS\_WINDOWS:    DL = Extension("torch.\_dl",                   sources=["torch/csrc/dl.c"],                   language='c'                   )    extensions.append(DL)  if USE\_CUDA:    thnvrtc\_link\_flags = extra\_link\_args + [make\_relative\_rpath('lib')]    if IS\_LINUX:        thnvrtc\_link\_flags = thnvrtc\_link\_flags + ['-Wl,--no-as-needed']    # these have to be specified as -lcuda in link\_flags because they    # have to come right after the `no-as-needed` option    if IS\_WINDOWS:        thnvrtc\_link\_flags += ['cuda.lib', 'nvrtc.lib']    else:        thnvrtc\_link\_flags += ['-lcuda', '-lnvrtc']    cuda\_stub\_path = [cuda\_lib\_path + '/stubs']    if IS\_DARWIN:        # on macOS this is where the CUDA stub is installed according to the manual        cuda\_stub\_path = ["/usr/local/cuda/lib"]    THNVRTC = Extension("torch.\_nvrtc",                        sources=['torch/csrc/nvrtc.cpp'],                        language='c++',                        extra\_compile\_args=main\_compile\_args + extra\_compile\_args,                        include\_dirs=[cwd],                        library\_dirs=library\_dirs + cuda\_stub\_path,                        extra\_link\_args=thnvrtc\_link\_flags,                        )    extensions.append(THNVRTC) # These extensions are built by cmake and copied manually in build\_extensions()# inside the build\_ext implementaitonextensions.append(    Extension(        name=str('caffe2.python.caffe2\_pybind11\_state'),        sources=[]),)if USE\_CUDA:    extensions.append(        Extension(            name=str('caffe2.python.caffe2\_pybind11\_state\_gpu'),            sources=[]),    )if USE\_ROCM:    extensions.append(        Extension(            name=str('caffe2.python.caffe2\_pybind11\_state\_hip'),            sources=[]),    ) cmdclass = {    'create\_version\_file': create\_version\_file,    'build': build,    'build\_py': build\_py,    'build\_ext': build\_ext,    'build\_deps': build\_deps,    'build\_module': build\_module,    'rebuild': rebuild,    'develop': develop,    'install': install,    'clean': clean,}cmdclass.update(build\_dep\_cmds)cmdclass.update(rebuild\_dep\_cmds) entry\_points = {    'console\_scripts': [        'convert-caffe2-to-onnx = caffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'caffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inccaffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',

'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inccaffe2.python.onnx.bin.conversion:caffe2\_to\_onnx',        'convert-onnx-to-caffe2 = caffe2.python.onnx.bin.conversion:onnx\_to\_caffe2',    ]} if \_\_name\_\_ == '\_\_main\_\_':    setup(        name=package\_name,        version=version,        description=("Tensors and Dynamic neural networks in "                     "Python with strong GPU acceleration"),        ext\_modules=extensions,        cmdclass=cmdclass,        packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inclib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/include/pybind11/\*.h',                'lib/include/pybind11/detail/\*.h',                'lib/include/TH/\*.h\*',                'lib/include/TH/generic/\*.h\*',                'lib/include/THC/\*.cuh',                'lib/include/THC/\*.h\*',                'lib/include/THC/generic/\*.h',                'lib/include/THCUNN/\*.cuh',                'lib/include/THNN/\*.h',                'share/cmake/ATen/\*.cmake',                'share/cmake/Caffe2/\*.cmake',                'share/cmake/Caffe2/public/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/FindCUDA/\*.cmake',                'share/cmake/Gloo/\*.cmake',                'share/cmake/Torch/\*.cmake',            ],            'caffe2': [                'cpp\_test/\*',                'python/serialized\_test/data/operator\_test/\*.zip',            ]        },    )

'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h', packages=packages,        entry\_points=entry\_points,        package\_data={            'torch': [                'lib/\*.so\*',                'lib/\*.dylib\*',                'lib/\*.dll',                'lib/\*.lib',                'lib/\*.pdb',                'lib/torch\_shm\_manager',                'lib/\*.h',                'lib/include/ATen/\*.h',                'lib/include/ATen/cpu/\*.h',                'lib/include/ATen/core/\*.h',                'lib/include/ATen/cuda/\*.cuh',                'lib/include/ATen/cuda/\*.h',                'lib/include/ATen/cuda/detail/\*.cuh',                'lib/include/ATen/cuda/detail/\*.h',                'lib/include/ATen/cudnn/\*.h',                'lib/include/ATen/detail/\*.h',                'lib/include/caffe2/utils/\*.h',                'lib/include/c10/\*.h',                'lib/include/c10/macros/\*.h',                'lib/include/c10/core/\*.h',                'lib/include/ATen/core/dispatch/\*.h',                'lib/include/c10/core/impl/\*.h',                'lib/include/ATen/core/opschema/\*.h',                'lib/include/c10/util/\*.h',                'lib/include/c10/cuda/\*.h',                'lib/include/c10/cuda/impl/\*.h',                'lib/include/c10/hip/\*.h',                'lib/include/c10/hip/impl/\*.h',                'lib/include/caffe2/\*\*/\*.h',                'lib/include/torch/\*.h',                'lib/include/torch/csrc/\*.h',                'lib/include/torch/csrc/api/include/torch/\*.h',                'lib/include/torch/csrc/api/include/torch/data/\*.h',                'lib/include/torch/csrc/api/include/torch/data/dataloader/\*.h',                'lib/include/torch/csrc/api/include/torch/data/datasets/\*.h',                'lib/include/torch/csrc/api/include/torch/data/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/data/samplers/\*.h',                'lib/include/torch/csrc/api/include/torch/data/transforms/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/\*.h',                'lib/include/torch/csrc/api/include/torch/detail/ordered\_dict.h',                'lib/include/torch/csrc/api/include/torch/nn/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/inclib/include/torch/csrc/api/include/torch/nn/modules/\*.h',                'lib/include/torch/csrc/api/include/torch/nn/parallel/\*.h',                'lib/include/torch/csrc/api/include/torch/optim/\*.h',                'lib/include/torch/csrc/api/include/torch/serialize/\*.h',                'lib/include/torch/csrc/autograd/\*.h',                'lib/include/torch/csrc/autograd/generated/\*.h',                'lib/include/torch/csrc/cuda/\*.h',                'lib/include/torch/csrc/jit/\*.h',                'lib/include/torch/csrc/jit/generated/\*.h',                'lib/include/torch/csrc/jit/passes/\*.h',                'lib/include/torch/csrc/jit/script/\*.h',                'lib/include/torch/csrc/utils/\*.h',                'lib/include/pybind11/\*.h',                'lib/include/pybind11/detail/\*.h',                'lib/include/TH/\*.h\*',                'lib/include/TH/generic/\*.h\*',                'lib/include/THC/\*.cuh',                'lib/include/THC/\*.h\*',                'lib/include/THC/generic/\*.h',                'lib/include/THCUNN/\*.cuh',                'lib/include/THNN/\*.h',                'share/cmake/ATen/\*.cmake',                'share/cmake/Caffe2/\*.cmake',                'share/cmake/Caffe2/public/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/\*.cmake',                'share/cmake/Caffe2/Modules\_CUDA\_fix/upstream/FindCUDA/\*.cmake',                'share/cmake/Gloo/\*.cmake',                'share/cmake/Torch/\*.cmake',            ],            'caffe2': [                'cpp\_test/\*',                'python/serialized\_test/data/operator\_test/\*.zip',            ]        },    )