Logistic Regression

**Logistic Regression**

1. !pip3 install torch

Output: Collecting torch   Downloading<https://files.pythonhosted.org/packages/7e/60/66415660aa46b23b5e1b72bc762e816736ce8d7260213e22365af51e8f9c/torch-1.0.0-cp36-cp36m-manylinux1_x86_64.whl> (591.8MB)     100% |████████████████████████████████| 591.8MB 25kB/s  tcmalloc: large alloc 1073750016 bytes == 0x60eb4000 @  0x7ffa63ed32a4 0x591a07 0x5b5d56 0x502e9a 0x506859 0x502209 0x502f3d 0x506859 0x504c28 0x502540 0x502f3d 0x506859 0x504c28 0x502540 0x502f3d 0x506859 0x504c28 0x502540 0x502f3d 0x507641 0x502209 0x502f3d 0x506859 0x504c28 0x502540 0x502f3d 0x507641 0x504c28 0x502540 0x502f3d 0x507641 Installing collected packages: torch Successfully installed torch-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**

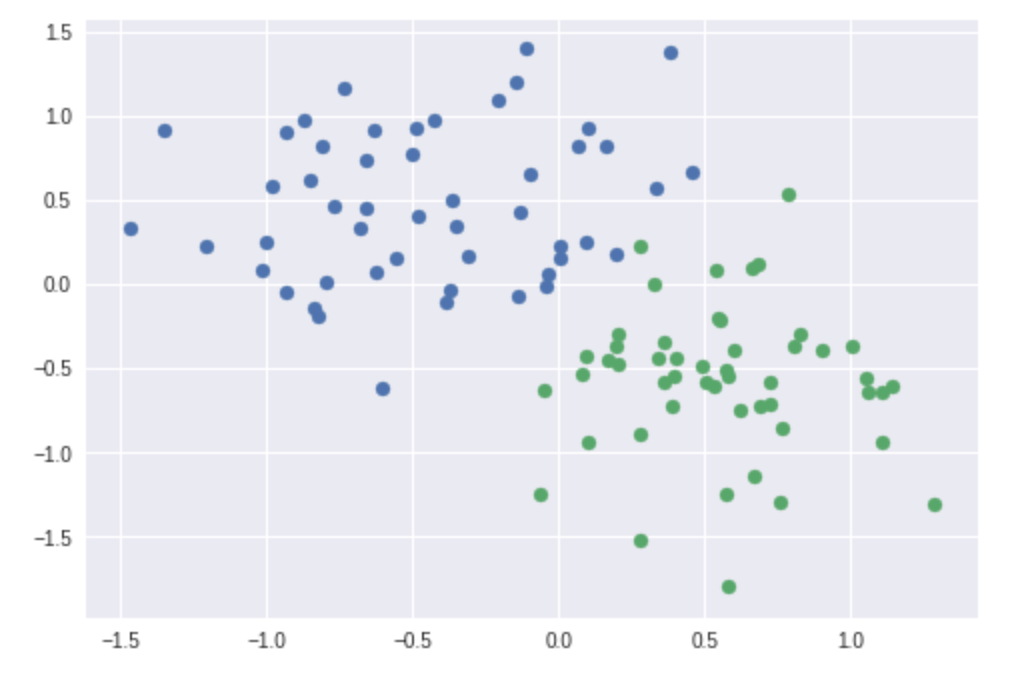
1. n\_pts = 100
2. centers = [[-0.5, 0.5], [0.5, -0.5]]
3. X, y = datasets.make\_blobs(n\_samples=n\_pts, random\_state=123, centers=centers, cluster\_std=0.4)
5. x\_data = torch.Tensor(X)
6. y\_data = torch.Tensor(y.reshape(100, 1))

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

**Instantiating Model Instance**

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

Output: [Parameter containing: tensor([[ 0.1622, -0.1683]], requires\_grad=True), Parameter containing: tensor([0.1939], requires\_grad=True)]

**Extracting Model Parameters**

1. [w, b] = model.parameters()
2. w1, w2 = w.view(2)
3. def get\_params():
4. return (w1.item(), w2.item(), b[0].item())

**Defining plot\_fit function**

1. def plot\_fit(title):
2. plt.title = title
3. w1, w2, b1 = get\_params()
4. x1 = np.array([-2, 2])
5. x2 = (w1\*x1 + b1)/(-w2)
6. plt.plot(x1, x2, 'r')
7. scatter\_plot()
8. plt.show()

**Plotting Initial Model**

1. plot\_fit('Initial Model')



**Binary Cross Entropy Loss and Optimizer Declaration**

1. criterion = nn.BCELoss()
2. optimizer = torch.optim.SGD(model.parameters(), lr=0.01)

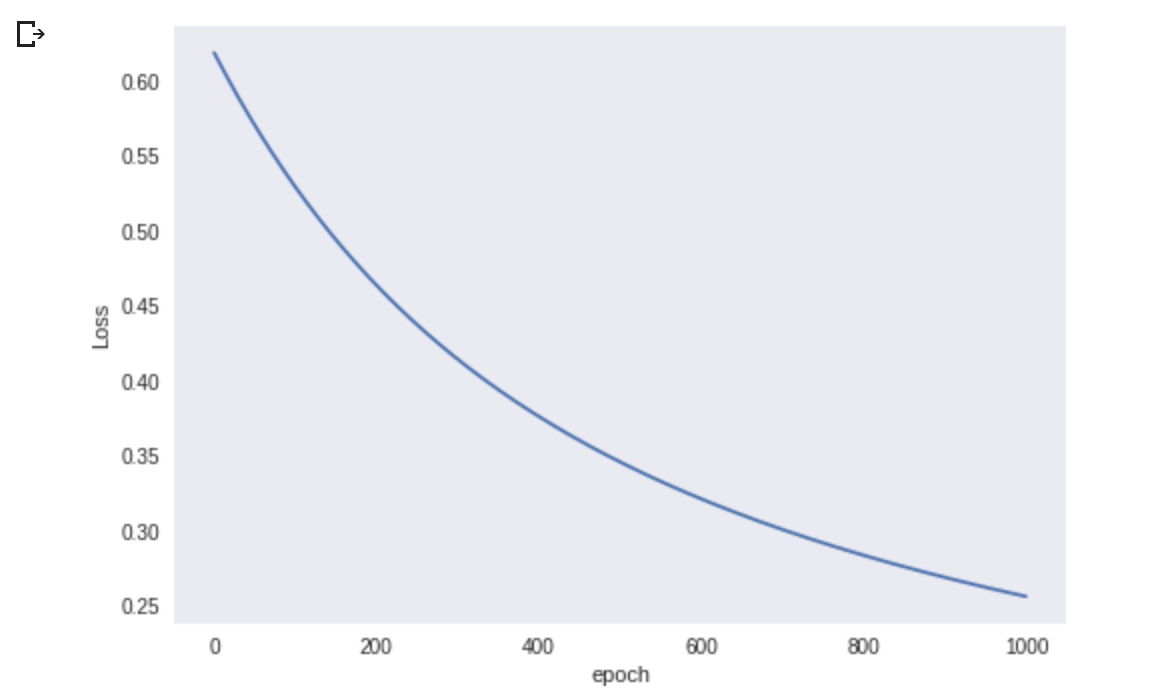
**Training Model**

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

epoch: 0 loss: 0.6185115575790405 epoch: 1 loss: 0.6174639463424683 epoch: 2 loss: 0.6164201498031616 epoch: 3 loss: 0.6153794527053833 epoch: 4 loss: 0.614342451095581 epoch: 5 loss: 0.6133086681365967 epoch: 6 loss: 0.6122783422470093 epoch: 7 loss: 0.6112515330314636 epoch: 8 loss: 0.6102280616760254 epoch: 9 loss: 0.6092080473899841 epoch: 10 loss: 0.6081910729408264 epoch: 11 loss: 0.6071775555610657 epoch: 12 loss: 0.606167733669281 epoch: 13 loss: 0.6051608324050903 epoch: 14 loss: 0.604157567024231 epoch: 15 loss: 0.6031574606895447 epoch: 16 loss: 0.602160632610321 epoch: 17 loss: 0.6011670827865601 epoch: 18 loss: 0.6001767516136169 epoch: 19 loss: 0.599189817905426 epoch: 20 loss: 0.5982059836387634 epoch: 21 loss: 0.5972254276275635 epoch: 22 loss: 0.5962479114532471 epoch: 23 loss: 0.5952739119529724 epoch: 24 loss: 0.5943028926849365 epoch: 25 loss: 0.5933352112770081 epoch: 26 loss: 0.5923705101013184 epoch: 27 loss: 0.5914090275764465 epoch: 28 loss: 0.5904508829116821 epoch: 29 loss: 0.5894957184791565 epoch: 30 loss: 0.588543713092804 epoch: 31 loss: 0.5875945687294006 epoch: 32 loss: 0.5866488218307495 epoch: 33 loss: 0.5857059955596924 epoch: 34 loss: 0.5847664475440979 epoch: 35 loss: 0.5838297009468079 epoch: 36 loss: 0.5828962922096252 epoch: 37 loss: 0.5819657444953918 epoch: 38 loss: 0.581038236618042 epoch: 39 loss: 0.5801137685775757 epoch: 40 loss: 0.5791923403739929 epoch: 41 loss: 0.5782737135887146 epoch: 42 loss: 0.577358067035675 epoch: 43 loss: 0.5764455199241638 epoch: 44 loss: 0.5755358934402466 epoch: 45 loss: 0.5746294260025024 epoch: 46 loss: 0.5737255811691284 epoch: 47 loss: 0.5728249549865723 epoch: 48 loss: 0.5719269514083862 epoch: 49 loss: 0.5710318088531494 epoch: 50 loss: 0.5701398849487305 epoch: 51 loss: 0.5692505240440369 epoch: 52 loss: 0.5683643221855164 epoch: 53 loss: 0.5674808621406555 epoch: 54 loss: 0.5666001439094543 epoch: 55 loss: 0.5657223463058472 epoch: 56 loss: 0.5648473501205444 epoch: 57 loss: 0.5639750957489014 epoch: 58 loss: 0.5631057620048523 epoch: 59 loss: 0.5622392892837524 epoch: 60 loss: 0.561375617980957 epoch: 61 loss: 0.5605144500732422 epoch: 62 loss: 0.5596560835838318 epoch: 63 loss: 0.5588006973266602 epoch: 64 loss: 0.5579478144645691 epoch: 65 loss: 0.5570977330207825 epoch: 66 loss: 0.5562503933906555 epoch: 67 loss: 0.5554057955741882 epoch: 68 loss: 0.5545637011528015 epoch: 69 loss: 0.5537243485450745 epoch: 70 loss: 0.5528878569602966 epoch: 71 loss: 0.5520538687705994 epoch: 72 loss: 0.5512227416038513 epoch: 73 loss: 0.5503939986228943 epoch: 74 loss: 0.5495679378509521 epoch: 75 loss: 0.5487444996833801 epoch: 76 loss: 0.5479238629341125 epoch: 77 loss: 0.5471054911613464 epoch: 78 loss: 0.5462900400161743 epoch: 79 loss: 0.5454771518707275 epoch: 80 loss: 0.5446667671203613 epoch: 81 loss: 0.5438589453697205 epoch: 82 loss: 0.5430535674095154 epoch: 83 loss: 0.5422506928443909 epoch: 84 loss: 0.5414506196975708 epoch: 85 loss: 0.5406527519226074 epoch: 86 loss: 0.5398576855659485 epoch: 87 loss: 0.5390648245811462 epoch: 88 loss: 0.5382746458053589 epoch: 89 loss: 0.5374867916107178 epoch: 90 loss: 0.5367016792297363 epoch: 91 loss: 0.535918653011322 epoch: 92 loss: 0.5351383686065674 epoch: 93 loss: 0.5343605279922485 epoch: 94 loss: 0.5335849523544312 epoch: 95 loss: 0.5328119993209839 epoch: 96 loss: 0.5320413112640381 epoch: 97 loss: 0.5312730669975281 epoch: 98 loss: 0.5305072665214539 epoch: 99 loss: 0.5297438502311707 epoch: 100 loss: 0.5289824604988098 epoch: 101 loss: 0.5282240509986877 epoch: 102 loss: 0.5274677872657776 epoch: 103 loss: 0.5267136693000793 epoch: 104 loss: 0.5259618759155273 epoch: 105 loss: 0.5252127647399902 epoch: 106 loss: 0.524465799331665 epoch: 107 loss: 0.5237209796905518 epoch: 108 loss: 0.5229786038398743 epoch: 109 loss: 0.5222384929656982 epoch: 110 loss: 0.5215004682540894 epoch: 111 loss: 0.5207651257514954 epoch: 112 loss: 0.5200319290161133 epoch: 113 loss: 0.5193008184432983 epoch: 114 loss: 0.5185720324516296 epoch: 115 loss: 0.5178455114364624 epoch: 116 loss: 0.5171211361885071 epoch: 117 loss: 0.5163991451263428 epoch: 118 loss: 0.5156794190406799 epoch: 119 loss: 0.5149617791175842 epoch: 120 loss: 0.5142462253570557 epoch: 121 loss: 0.5135331153869629 epoch: 122 loss: 0.5128219723701477 epoch: 123 loss: 0.5121132135391235 epoch: 124 loss: 0.5114063024520874 epoch: 125 loss: 0.5107016563415527 epoch: 126 loss: 0.5099992752075195 epoch: 127 loss: 0.5092989206314087 epoch: 128 loss: 0.5086007714271545 epoch: 129 loss: 0.5079045295715332 epoch: 130 loss: 0.5072106122970581 epoch: 131 loss: 0.5065188407897949 epoch: 132 loss: 0.5058292150497437 epoch: 133 loss: 0.5051414966583252 epoch: 134 loss: 0.5044559240341187 epoch: 135 loss: 0.5037723779678345 epoch: 136 loss: 0.5030909180641174 epoch: 137 loss: 0.5024116039276123 epoch: 138 loss: 0.5017342567443848 epoch: 139 loss: 0.50105881690979 epoch: 140 loss: 0.5003856420516968 epoch: 141 loss: 0.4997144341468811 epoch: 142 loss: 0.4990452826023102 epoch: 143 loss: 0.49837803840637207 epoch: 144 loss: 0.4977127015590668 epoch: 145 loss: 0.497049480676651 epoch: 146 loss: 0.4963882863521576 epoch: 147 loss: 0.49572890996932983 epoch: 148 loss: 0.49507153034210205 epoch: 149 loss: 0.49441614747047424 epoch: 150 loss: 0.4937628209590912 epoch: 151 loss: 0.49311134219169617 epoch: 152 loss: 0.49246203899383545 epoch: 153 loss: 0.49181443452835083 epoch: 154 loss: 0.4911685883998871 epoch: 155 loss: 0.49052485823631287 epoch: 156 loss: 0.48988306522369385 epoch: 157 loss: 0.48924314975738525 epoch: 158 loss: 0.4886051118373871 epoch: 159 loss: 0.48796898126602173 epoch: 160 loss: 0.487334668636322 epoch: 161 loss: 0.4867023527622223 epoch: 162 loss: 0.4860716760158539 epoch: 163 loss: 0.4854430854320526 epoch: 164 loss: 0.4848161041736603 epoch: 165 loss: 0.4841911196708679 epoch: 166 loss: 0.4835679233074188 epoch: 167 loss: 0.4829466640949249 epoch: 168 loss: 0.4823271632194519 epoch: 169 loss: 0.4817095100879669 epoch: 170 loss: 0.48109352588653564 epoch: 171 loss: 0.48047950863838196 epoch: 172 loss: 0.47986698150634766 epoch: 173 loss: 0.47925662994384766 epoch: 174 loss: 0.47864776849746704 epoch: 175 loss: 0.478040874004364 epoch: 176 loss: 0.4774356782436371 epoch: 177 loss: 0.47683224081993103 epoch: 178 loss: 0.47623059153556824 epoch: 179 loss: 0.4756303131580353 epoch: 180 loss: 0.4750322997570038 epoch: 181 loss: 0.47443583607673645 epoch: 182 loss: 0.4738411605358124 epoch: 183 loss: 0.47324806451797485 epoch: 184 loss: 0.4726565480232239 epoch: 185 loss: 0.47206705808639526 epoch: 186 loss: 0.4714791476726532 epoch: 187 loss: 0.47089287638664246 epoch: 188 loss: 0.47030842304229736 epoch: 189 loss: 0.4697253704071045 epoch: 190 loss: 0.4691442549228668 epoch: 191 loss: 0.4685647487640381 epoch: 192 loss: 0.46798667311668396 epoch: 193 loss: 0.46741050481796265 epoch: 194 loss: 0.4668358266353607 epoch: 195 loss: 0.46626290678977966 epoch: 196 loss: 0.4656916558742523 epoch: 197 loss: 0.4651219844818115 epoch: 198 loss: 0.4645538330078125 epoch: 199 loss: 0.46398746967315674 epoch: 200 loss: 0.46342262625694275 epoch: 201 loss: 0.46285927295684814 epoch: 202 loss: 0.4622977077960968 epoch: 203 loss: 0.46173766255378723 epoch: 204 loss: 0.4611791968345642 epoch: 205 loss: 0.46062225103378296 epoch: 206 loss: 0.46006691455841064 epoch: 207 loss: 0.45951324701309204 epoch: 208 loss: 0.4589608311653137 epoch: 209 loss: 0.45841044187545776 epoch: 210 loss: 0.45786118507385254 epoch: 211 loss: 0.4573136270046234 epoch: 212 loss: 0.45676758885383606 epoch: 213 loss: 0.45622318983078003 epoch: 214 loss: 0.45568016171455383 epoch: 215 loss: 0.4551387131214142 epoch: 216 loss: 0.45459869503974915 epoch: 217 loss: 0.45406028628349304 epoch: 218 loss: 0.4535233974456787 epoch: 219 loss: 0.4529879093170166 epoch: 220 loss: 0.4524538516998291 epoch: 221 loss: 0.4519214332103729 epoch: 222 loss: 0.4513905346393585 epoch: 223 loss: 0.4508609473705292 epoch: 224 loss: 0.4503328800201416 epoch: 225 loss: 0.4498063027858734 epoch: 226 loss: 0.44928112626075745 epoch: 227 loss: 0.44875743985176086 epoch: 228 loss: 0.44823524355888367 epoch: 229 loss: 0.4477144181728363 epoch: 230 loss: 0.4471950829029083 epoch: 231 loss: 0.44667720794677734 epoch: 232 loss: 0.4461606740951538 epoch: 233 loss: 0.4456455707550049 epoch: 234 loss: 0.4451318681240082 epoch: 235 loss: 0.4446195662021637 epoch: 236 loss: 0.4441088140010834 epoch: 237 loss: 0.4435991942882538 epoch: 238 loss: 0.4430912137031555 epoch: 239 loss: 0.44258439540863037 epoch: 240 loss: 0.44207894802093506 epoch: 241 loss: 0.44157522916793823 epoch: 242 loss: 0.4410724639892578 epoch: 243 loss: 0.4405713379383087 epoch: 244 loss: 0.4400714933872223 epoch: 245 loss: 0.43957290053367615 epoch: 246 loss: 0.439075767993927 epoch: 247 loss: 0.43858012557029724 epoch: 248 loss: 0.438085675239563 epoch: 249 loss: 0.437592476606369 epoch: 250 loss: 0.43710049986839294 epoch: 251 loss: 0.4366101324558258 epoch: 252 loss: 0.4361210763454437 epoch: 253 loss: 0.4356331527233124 epoch: 254 loss: 0.4351467490196228 epoch: 255 loss: 0.4346614480018616 epoch: 256 loss: 0.4341774880886078 epoch: 257 loss: 0.43369483947753906 epoch: 258 loss: 0.43321359157562256 epoch: 259 loss: 0.43273353576660156 epoch: 260 loss: 0.43225476145744324 epoch: 261 loss: 0.43177711963653564 epoch: 262 loss: 0.43130093812942505 epoch: 263 loss: 0.4308260381221771 epoch: 264 loss: 0.4303523004055023 epoch: 265 loss: 0.4298797845840454 epoch: 266 loss: 0.4294084906578064 epoch: 267 loss: 0.42893868684768677 epoch: 268 loss: 0.42847001552581787 epoch: 269 loss: 0.4280024766921997 epoch: 270 loss: 0.4275362491607666 epoch: 271 loss: 0.42707115411758423 epoch: 272 loss: 0.42660748958587646 epoch: 273 loss: 0.4261450171470642 epoch: 274 loss: 0.42568355798721313 epoch: 275 loss: 0.42522352933883667 epoch: 276 loss: 0.42476463317871094 epoch: 277 loss: 0.42430704832077026 epoch: 278 loss: 0.42385023832321167 epoch: 279 loss: 0.4233950078487396 epoch: 280 loss: 0.42294082045555115 epoch: 281 loss: 0.42248794436454773 epoch: 282 loss: 0.42203617095947266 epoch: 283 loss: 0.4215855896472931 epoch: 284 loss: 0.42113614082336426 epoch: 285 loss: 0.42068779468536377 epoch: 286 loss: 0.4202407896518707 epoch: 287 loss: 0.4197949469089508 epoch: 288 loss: 0.4193502366542816 epoch: 289 loss: 0.4189063608646393 epoch: 290 loss: 0.41846397519111633 epoch: 291 loss: 0.41802260279655457 epoch: 292 loss: 0.417582631111145 epoch: 293 loss: 0.4171435236930847 epoch: 294 loss: 0.4167054295539856 epoch: 295 loss: 0.4162687063217163 epoch: 296 loss: 0.41583287715911865 epoch: 297 loss: 0.4153984785079956 epoch: 298 loss: 0.4149649739265442 epoch: 299 loss: 0.4145326316356659 epoch: 300 loss: 0.41410133242607117 epoch: 301 loss: 0.4136713147163391 epoch: 302 loss: 0.4132422208786011 epoch: 303 loss: 0.4128142297267914 epoch: 304 loss: 0.4123874604701996 epoch: 305 loss: 0.4119616448879242 epoch: 306 loss: 0.4115370213985443 epoch: 307 loss: 0.41111335158348083 epoch: 308 loss: 0.4106908440589905 epoch: 309 loss: 0.4102693796157837 epoch: 310 loss: 0.40984901785850525 epoch: 311 loss: 0.4094295799732208 epoch: 312 loss: 0.4090113937854767 epoch: 313 loss: 0.4085940420627594 epoch: 314 loss: 0.40817803144454956 epoch: 315 loss: 0.4077627956867218 epoch: 316 loss: 0.40734875202178955 epoch: 317 loss: 0.40693560242652893 epoch: 318 loss: 0.40652385354042053 epoch: 319 loss: 0.40611279010772705 epoch: 320 loss: 0.40570276975631714 epoch: 321 loss: 0.40529385209083557 epoch: 322 loss: 0.40488606691360474 epoch: 323 loss: 0.40447911620140076 epoch: 324 loss: 0.4040732681751251 epoch: 325 loss: 0.4036683738231659 epoch: 326 loss: 0.40326446294784546 epoch: 327 loss: 0.4028615951538086 epoch: 328 loss: 0.40245985984802246 epoch: 329 loss: 0.40205898880958557 epoch: 330 loss: 0.40165916085243225 epoch: 331 loss: 0.40126025676727295 epoch: 332 loss: 0.400862455368042 epoch: 333 loss: 0.4004655182361603 epoch: 334 loss: 0.4000696539878845 epoch: 335 loss: 0.399674654006958 epoch: 336 loss: 0.3992806673049927 epoch: 337 loss: 0.39888766407966614 epoch: 338 loss: 0.39849549531936646 epoch: 339 loss: 0.39810463786125183 epoch: 340 loss: 0.3977143168449402 epoch: 341 loss: 0.3973252475261688 epoch: 342 loss: 0.3969371020793915 epoch: 343 loss: 0.396549791097641 epoch: 344 loss: 0.3961635231971741 epoch: 345 loss: 0.3957781493663788 epoch: 346 loss: 0.39539361000061035 epoch: 347 loss: 0.39501017332077026 epoch: 348 loss: 0.39462757110595703 epoch: 349 loss: 0.3942459523677826 epoch: 350 loss: 0.3938651978969574 epoch: 351 loss: 0.3934853672981262 epoch: 352 loss: 0.39310649037361145 epoch: 353 loss: 0.3927285373210907 epoch: 354 loss: 0.3923514187335968 epoch: 355 loss: 0.39197540283203125 epoch: 356 loss: 0.39160001277923584 epoch: 357 loss: 0.39122581481933594 epoch: 358 loss: 0.3908523619174957 epoch: 359 loss: 0.3904797434806824 epoch: 360 loss: 0.3901081383228302 epoch: 361 loss: 0.3897373676300049 epoch: 362 loss: 0.38936755061149597 epoch: 363 loss: 0.38899853825569153 epoch: 364 loss: 0.3886304199695587 epoch: 365 loss: 0.38826319575309753 epoch: 366 loss: 0.38789692521095276 epoch: 367 loss: 0.38753142952919006 epoch: 368 loss: 0.3871667981147766 epoch: 369 loss: 0.38680294156074524 epoch: 370 loss: 0.386440247297287 epoch: 371 loss: 0.3860781490802765 epoch: 372 loss: 0.38571697473526 epoch: 373 loss: 0.38535675406455994 epoch: 374 loss: 0.3849973678588867 epoch: 375 loss: 0.3846386969089508 epoch: 376 loss: 0.3842810094356537 epoch: 377 loss: 0.38392409682273865 epoch: 378 loss: 0.3835679590702057 epoch: 379 loss: 0.3832126259803772 epoch: 380 loss: 0.38285836577415466 epoch: 381 loss: 0.3825048506259918 epoch: 382 loss: 0.3821520507335663 epoch: 383 loss: 0.3818000853061676 epoch: 384 loss: 0.3814489245414734 epoch: 385 loss: 0.3810987174510956 epoch: 386 loss: 0.3807491958141327 epoch: 387 loss: 0.3804006576538086 epoch: 388 loss: 0.3800526559352875 epoch: 389 loss: 0.3797057271003723 epoch: 390 loss: 0.37935951352119446 epoch: 391 loss: 0.37901413440704346 epoch: 392 loss: 0.37866947054862976 epoch: 393 loss: 0.3783257305622101 epoch: 394 loss: 0.37798264622688293 epoch: 395 loss: 0.37764036655426025 epoch: 396 loss: 0.37729892134666443 epoch: 397 loss: 0.37695831060409546 epoch: 398 loss: 0.3766184151172638 epoch: 399 loss: 0.3762793242931366 epoch: 400 loss: 0.37594103813171387 epoch: 401 loss: 0.37560349702835083 epoch: 402 loss: 0.37526679039001465 epoch: 403 loss: 0.37493079900741577 epoch: 404 loss: 0.3745954930782318 epoch: 405 loss: 0.3742610812187195 epoch: 406 loss: 0.37392738461494446 epoch: 407 loss: 0.37359458208084106 epoch: 408 loss: 0.37326228618621826 epoch: 409 loss: 0.3729308843612671 epoch: 410 loss: 0.3726001977920532 epoch: 411 loss: 0.372270405292511 epoch: 412 loss: 0.3719412684440613 epoch: 413 loss: 0.3716128170490265 epoch: 414 loss: 0.371285080909729 epoch: 415 loss: 0.37095823884010315 epoch: 416 loss: 0.37063202261924744 epoch: 417 loss: 0.3703065514564514 epoch: 418 loss: 0.36998191475868225 epoch: 419 loss: 0.3696579039096832 epoch: 420 loss: 0.36933472752571106 epoch: 421 loss: 0.36901217699050903 epoch: 422 loss: 0.3686903715133667 epoch: 423 loss: 0.36836928129196167 epoch: 424 loss: 0.36804890632629395 epoch: 425 loss: 0.3677292764186859 epoch: 426 loss: 0.3674103915691376 epoch: 427 loss: 0.3670922517776489 epoch: 428 loss: 0.3667745590209961 epoch: 429 loss: 0.36645781993865967 epoch: 430 loss: 0.3661418557167053 epoch: 431 loss: 0.36582642793655396 epoch: 432 loss: 0.3655116558074951 epoch: 433 loss: 0.36519771814346313 epoch: 434 loss: 0.3648844063282013 epoch: 435 loss: 0.3645719587802887 epoch: 436 loss: 0.36425986886024475 epoch: 437 loss: 0.36394867300987244 epoch: 438 loss: 0.36363810300827026 epoch: 439 loss: 0.36332833766937256 epoch: 440 loss: 0.3630192279815674 epoch: 441 loss: 0.3627106845378876 epoch: 442 loss: 0.36240288615226746 epoch: 443 loss: 0.36209580302238464 epoch: 444 loss: 0.36178919672966003 epoch: 445 loss: 0.3614836037158966 epoch: 446 loss: 0.3611784875392914 epoch: 447 loss: 0.3608740270137787 epoch: 448 loss: 0.36057013273239136 epoch: 449 loss: 0.36026710271835327 epoch: 450 loss: 0.3599647879600525 epoch: 451 loss: 0.3596627712249756 epoch: 452 loss: 0.3593617379665375 epoch: 453 loss: 0.35906124114990234 epoch: 454 loss: 0.35876142978668213 epoch: 455 loss: 0.35846224427223206 epoch: 456 loss: 0.3581637144088745 epoch: 457 loss: 0.35786575078964233 epoch: 458 loss: 0.3575686514377594 epoch: 459 loss: 0.3572719991207123 epoch: 460 loss: 0.3569760024547577 epoch: 461 loss: 0.3566807508468628 epoch: 462 loss: 0.3563860356807709 epoch: 463 loss: 0.35609206557273865 epoch: 464 loss: 0.35579854249954224 epoch: 465 loss: 0.35550570487976074 epoch: 466 loss: 0.35521361231803894 epoch: 467 loss: 0.35492220520973206 epoch: 468 loss: 0.35463112592697144 epoch: 469 loss: 0.35434094071388245 epoch: 470 loss: 0.35405123233795166 epoch: 471 loss: 0.3537622392177582 epoch: 472 loss: 0.3534739315509796 epoch: 473 loss: 0.3531860113143921 epoch: 474 loss: 0.3528987765312195 epoch: 475 loss: 0.3526122272014618 epoch: 476 loss: 0.3523262143135071 epoch: 477 loss: 0.3520407974720001 epoch: 478 loss: 0.35175594687461853 epoch: 479 loss: 0.351471871137619 epoch: 480 loss: 0.3511882722377777 epoch: 481 loss: 0.35090532898902893 epoch: 482 loss: 0.3506229519844055 epoch: 483 loss: 0.35034123063087463 epoch: 484 loss: 0.35006001591682434 epoch: 485 loss: 0.34977948665618896 epoch: 486 loss: 0.34949952363967896 epoch: 487 loss: 0.3492198884487152 epoch: 488 loss: 0.34894120693206787 epoch: 489 loss: 0.34866294264793396 epoch: 490 loss: 0.3483852446079254 epoch: 491 loss: 0.348108172416687 epoch: 492 loss: 0.34783169627189636 epoch: 493 loss: 0.3475557267665863 epoch: 494 loss: 0.3472804129123688 epoch: 495 loss: 0.34700560569763184 epoch: 496 loss: 0.3467313051223755 epoch: 497 loss: 0.3464578688144684 epoch: 498 loss: 0.346184641122818 epoch: 499 loss: 0.3459121286869049 epoch: 500 loss: 0.34564027190208435 epoch: 501 loss: 0.345368891954422 epoch: 502 loss: 0.3450981378555298 epoch: 503 loss: 0.34482792019844055 epoch: 504 loss: 0.34455811977386475 epoch: 505 loss: 0.34428897500038147 epoch: 506 loss: 0.34402039647102356 epoch: 507 loss: 0.34375235438346863 epoch: 508 loss: 0.34348490834236145 epoch: 509 loss: 0.3432179391384125 epoch: 510 loss: 0.34295162558555603 epoch: 511 loss: 0.3426858186721802 epoch: 512 loss: 0.3424203395843506 epoch: 513 loss: 0.3421556353569031 epoch: 514 loss: 0.34189143776893616 epoch: 515 loss: 0.3416278064250946 epoch: 516 loss: 0.3413645625114441 epoch: 517 loss: 0.3411019444465637 epoch: 518 loss: 0.34084004163742065 epoch: 519 loss: 0.3405783772468567 epoch: 520 loss: 0.3403174877166748 epoch: 521 loss: 0.34005698561668396 epoch: 522 loss: 0.3397969901561737 epoch: 523 loss: 0.3395375907421112 epoch: 524 loss: 0.3392787575721741 epoch: 525 loss: 0.33902034163475037 epoch: 526 loss: 0.33876246213912964 epoch: 527 loss: 0.33850517868995667 epoch: 528 loss: 0.33824828267097473 epoch: 529 loss: 0.3379921019077301 epoch: 530 loss: 0.3377362787723541 epoch: 531 loss: 0.33748099207878113 epoch: 532 loss: 0.33722636103630066 epoch: 533 loss: 0.3369719982147217 epoch: 534 loss: 0.3367183208465576 epoch: 535 loss: 0.3364649713039398 epoch: 536 loss: 0.33621230721473694 epoch: 537 loss: 0.33596011996269226 epoch: 538 loss: 0.3357083797454834 epoch: 539 loss: 0.3354572355747223 epoch: 540 loss: 0.3352064788341522 epoch: 541 loss: 0.33495619893074036 epoch: 542 loss: 0.33470645546913147 epoch: 543 loss: 0.33445727825164795 epoch: 544 loss: 0.3342086374759674 epoch: 545 loss: 0.3339603841304779 epoch: 546 loss: 0.33371248841285706 epoch: 547 loss: 0.3334653079509735 epoch: 548 loss: 0.3332185745239258 epoch: 549 loss: 0.3329722583293915 epoch: 550 loss: 0.332726389169693 epoch: 551 loss: 0.33248114585876465 epoch: 552 loss: 0.33223628997802734 epoch: 553 loss: 0.331991970539093 epoch: 554 loss: 0.3317481279373169 epoch: 555 loss: 0.33150479197502136 epoch: 556 loss: 0.3312618136405945 epoch: 557 loss: 0.33101943135261536 epoch: 558 loss: 0.33077743649482727 epoch: 559 loss: 0.3305359184741974 epoch: 560 loss: 0.3302949070930481 epoch: 561 loss: 0.33005425333976746 epoch: 562 loss: 0.32981425523757935 epoch: 563 loss: 0.3295746147632599 epoch: 564 loss: 0.32933560013771057 epoch: 565 loss: 0.32909679412841797 epoch: 566 loss: 0.3288586437702179 epoch: 567 loss: 0.3286207616329193 epoch: 568 loss: 0.32838356494903564 epoch: 569 loss: 0.32814663648605347 epoch: 570 loss: 0.32791030406951904 epoch: 571 loss: 0.32767432928085327 epoch: 572 loss: 0.3274388611316681 epoch: 573 loss: 0.32720378041267395 epoch: 574 loss: 0.3269692659378052 epoch: 575 loss: 0.32673510909080505 epoch: 576 loss: 0.3265014588832855 epoch: 577 loss: 0.3262682855129242 epoch: 578 loss: 0.3260354995727539 epoch: 579 loss: 0.32580316066741943 epoch: 580 loss: 0.32557129859924316 epoch: 581 loss: 0.3253398537635803 epoch: 582 loss: 0.3251088261604309 epoch: 583 loss: 0.3248782753944397 epoch: 584 loss: 0.3246481716632843 epoch: 585 loss: 0.32441848516464233 epoch: 586 loss: 0.3241891860961914 epoch: 587 loss: 0.32396042346954346 epoch: 588 loss: 0.32373207807540894 epoch: 589 loss: 0.32350409030914307 epoch: 590 loss: 0.3232766091823578 epoch: 591 loss: 0.32304951548576355 epoch: 592 loss: 0.32282280921936035 epoch: 593 loss: 0.32259663939476013 epoch: 594 loss: 0.3223707973957062 epoch: 595 loss: 0.3221454620361328 epoch: 596 loss: 0.3219205439090729 epoch: 597 loss: 0.32169604301452637 epoch: 598 loss: 0.3214719295501709 epoch: 599 loss: 0.32124829292297363 epoch: 600 loss: 0.32102492451667786 epoch: 601 loss: 0.3208022713661194 epoch: 602 loss: 0.3205796778202057 epoch: 603 loss: 0.3203578293323517 epoch: 604 loss: 0.3201361894607544 epoch: 605 loss: 0.3199150562286377 epoch: 606 loss: 0.3196941316127777 epoch: 607 loss: 0.3194738030433655 epoch: 608 loss: 0.3192538917064667 epoch: 609 loss: 0.3190343379974365 epoch: 610 loss: 0.318815141916275 epoch: 611 loss: 0.3185964822769165 epoch: 612 loss: 0.3183782398700714 epoch: 613 loss: 0.3181602656841278 epoch: 614 loss: 0.31794273853302 epoch: 615 loss: 0.31772562861442566 epoch: 616 loss: 0.3175089955329895 epoch: 617 loss: 0.317292720079422 epoch: 618 loss: 0.3170766532421112 epoch: 619 loss: 0.31686127185821533 epoch: 620 loss: 0.31664618849754333 epoch: 621 loss: 0.3164314329624176 epoch: 622 loss: 0.3162171244621277 epoch: 623 loss: 0.3160032331943512 epoch: 624 loss: 0.3157895505428314 epoch: 625 loss: 0.315576434135437 epoch: 626 loss: 0.31536373496055603 epoch: 627 loss: 0.3151513338088989 epoch: 628 loss: 0.3149394094944 epoch: 629 loss: 0.3147277235984802 epoch: 630 loss: 0.3145165741443634 epoch: 631 loss: 0.31430575251579285 epoch: 632 loss: 0.31409531831741333 epoch: 633 loss: 0.3138852119445801 epoch: 634 loss: 0.3136756420135498 epoch: 635 loss: 0.31346625089645386 epoch: 636 loss: 0.3132573366165161 epoch: 637 loss: 0.313048779964447 epoch: 638 loss: 0.31284067034721375 epoch: 639 loss: 0.3126329183578491 epoch: 640 loss: 0.312425434589386 epoch: 641 loss: 0.31221848726272583 epoch: 642 loss: 0.3120117485523224 epoch: 643 loss: 0.3118055462837219 epoch: 644 loss: 0.3115994930267334 epoch: 645 loss: 0.31139397621154785 epoch: 646 loss: 0.31118881702423096 epoch: 647 loss: 0.3109840452671051 epoch: 648 loss: 0.3107796013355255 epoch: 649 loss: 0.3105754554271698 epoch: 650 loss: 0.31037193536758423 epoch: 651 loss: 0.31016844511032104 epoch: 652 loss: 0.30996549129486084 epoch: 653 loss: 0.3097628355026245 epoch: 654 loss: 0.3095605969429016 epoch: 655 loss: 0.30935874581336975 epoch: 656 loss: 0.30915719270706177 epoch: 657 loss: 0.30895599722862244 epoch: 658 loss: 0.308755099773407 epoch: 659 loss: 0.30855467915534973 epoch: 660 loss: 0.30835455656051636 epoch: 661 loss: 0.3081548511981964 epoch: 662 loss: 0.30795541405677795 epoch: 663 loss: 0.30775636434555054 epoch: 664 loss: 0.3075575828552246 epoch: 665 loss: 0.3073593080043793 epoch: 666 loss: 0.3071613311767578 epoch: 667 loss: 0.3069636821746826 epoch: 668 loss: 0.3067662715911865 epoch: 669 loss: 0.30656933784484863 epoch: 670 loss: 0.30637282133102417 epoch: 671 loss: 0.30617648363113403 epoch: 672 loss: 0.3059805929660797 epoch: 673 loss: 0.3057849407196045 epoch: 674 loss: 0.3055897355079651 epoch: 675 loss: 0.3053949177265167 epoch: 676 loss: 0.30520033836364746 epoch: 677 loss: 0.30500611662864685 epoch: 678 loss: 0.3048121929168701 epoch: 679 loss: 0.30461862683296204 epoch: 680 loss: 0.3044254183769226 epoch: 681 loss: 0.3042326867580414 epoch: 682 loss: 0.3040400445461273 epoch: 683 loss: 0.30384787917137146 epoch: 684 loss: 0.3036560118198395 epoch: 685 loss: 0.30346444249153137 epoch: 686 loss: 0.30327317118644714 epoch: 687 loss: 0.30308234691619873 epoch: 688 loss: 0.3028918206691742 epoch: 689 loss: 0.30270150303840637 epoch: 690 loss: 0.30251166224479675 epoch: 691 loss: 0.302322119474411 epoch: 692 loss: 0.30213287472724915 epoch: 693 loss: 0.30194395780563354 epoch: 694 loss: 0.3017553389072418 epoch: 695 loss: 0.3015671670436859 epoch: 696 loss: 0.3013792037963867 epoch: 697 loss: 0.3011915683746338 epoch: 698 loss: 0.3010042905807495 epoch: 699 loss: 0.3008173108100891 epoch: 700 loss: 0.30063068866729736 epoch: 701 loss: 0.3004443645477295 epoch: 702 loss: 0.3002582788467407 epoch: 703 loss: 0.300072580575943 epoch: 704 loss: 0.2998872697353363 epoch: 705 loss: 0.29970210790634155 epoch: 706 loss: 0.299517422914505 epoch: 707 loss: 0.2993329167366028 epoch: 708 loss: 0.299148827791214 epoch: 709 loss: 0.2989649176597595 epoch: 710 loss: 0.29878148436546326 epoch: 711 loss: 0.2985983192920685 epoch: 712 loss: 0.2984153926372528 epoch: 713 loss: 0.2982327938079834 epoch: 714 loss: 0.29805052280426025 epoch: 715 loss: 0.29786860942840576 epoch: 716 loss: 0.29768693447113037 epoch: 717 loss: 0.29750561714172363 epoch: 718 loss: 0.2973245084285736 epoch: 719 loss: 0.297143816947937 epoch: 720 loss: 0.29696333408355713 epoch: 721 loss: 0.2967832088470459 epoch: 722 loss: 0.29660338163375854 epoch: 723 loss: 0.2964238226413727 epoch: 724 loss: 0.29624462127685547 epoch: 725 loss: 0.29606562852859497 epoch: 726 loss: 0.29588696360588074 epoch: 727 loss: 0.29570862650871277 epoch: 728 loss: 0.29553067684173584 epoch: 729 loss: 0.2953529357910156 epoch: 730 loss: 0.2951754331588745 epoch: 731 loss: 0.29499825835227966 epoch: 732 loss: 0.29482144117355347 epoch: 733 loss: 0.29464492201805115 epoch: 734 loss: 0.2944685220718384 epoch: 735 loss: 0.29429250955581665 epoch: 736 loss: 0.29411688446998596 epoch: 737 loss: 0.2939414978027344 epoch: 738 loss: 0.29376640915870667 epoch: 739 loss: 0.29359155893325806 epoch: 740 loss: 0.2934170067310333 epoch: 741 loss: 0.29324275255203247 epoch: 742 loss: 0.2930687665939331 epoch: 743 loss: 0.29289510846138 epoch: 744 loss: 0.2927217185497284 epoch: 745 loss: 0.29254862666130066 epoch: 746 loss: 0.29237574338912964 epoch: 747 loss: 0.2922031879425049 epoch: 748 loss: 0.2920309901237488 epoch: 749 loss: 0.2918590009212494 epoch: 750 loss: 0.2916873097419739 epoch: 751 loss: 0.29151585698127747 epoch: 752 loss: 0.2913447618484497 epoch: 753 loss: 0.29117393493652344 epoch: 754 loss: 0.2910033166408539 epoch: 755 loss: 0.2908329665660858 epoch: 756 loss: 0.2906630039215088 epoch: 757 loss: 0.29049327969551086 epoch: 758 loss: 0.29032376408576965 epoch: 759 loss: 0.2901546359062195 epoch: 760 loss: 0.2899855971336365 epoch: 761 loss: 0.28981703519821167 epoch: 762 loss: 0.2896486520767212 epoch: 763 loss: 0.2894805669784546 epoch: 764 loss: 0.2893126308917999 epoch: 765 loss: 0.2891451120376587 epoch: 766 loss: 0.28897780179977417 epoch: 767 loss: 0.2888107895851135 epoch: 768 loss: 0.2886439859867096 epoch: 769 loss: 0.2884775698184967 epoch: 770 loss: 0.28831130266189575 epoch: 771 loss: 0.28814536333084106 epoch: 772 loss: 0.28797975182533264 epoch: 773 loss: 0.2878141701221466 epoch: 774 loss: 0.28764912486076355 epoch: 775 loss: 0.28748419880867004 epoch: 776 loss: 0.2873196005821228 epoch: 777 loss: 0.28715527057647705 epoch: 778 loss: 0.28699102997779846 epoch: 779 loss: 0.2868272364139557 epoch: 780 loss: 0.2866637110710144 epoch: 781 loss: 0.28650033473968506 epoch: 782 loss: 0.2863372564315796 epoch: 783 loss: 0.2861744463443756 epoch: 784 loss: 0.28601181507110596 epoch: 785 loss: 0.28584954142570496 epoch: 786 loss: 0.28568753600120544 epoch: 787 loss: 0.28552570939064026 epoch: 788 loss: 0.28536418080329895 epoch: 789 loss: 0.2852029502391815 epoch: 790 loss: 0.2850418984889984 epoch: 791 loss: 0.2848811447620392 epoch: 792 loss: 0.2847207188606262 epoch: 793 loss: 0.2845603823661804 epoch: 794 loss: 0.28440049290657043 epoch: 795 loss: 0.28424060344696045 epoch: 796 loss: 0.2840811610221863 epoch: 797 loss: 0.28392189741134644 epoch: 798 loss: 0.28376293182373047 epoch: 799 loss: 0.28360411524772644 epoch: 800 loss: 0.28344565629959106 epoch: 801 loss: 0.28328737616539 epoch: 802 loss: 0.2831293046474457 epoch: 803 loss: 0.2829716205596924 epoch: 804 loss: 0.2828141152858734 epoch: 805 loss: 0.28265678882598877 epoch: 806 loss: 0.2824997901916504 epoch: 807 loss: 0.2823430299758911 epoch: 808 loss: 0.2821864187717438 epoch: 809 loss: 0.2820301651954651 epoch: 810 loss: 0.2818740904331207 epoch: 811 loss: 0.28171828389167786 epoch: 812 loss: 0.28156277537345886 epoch: 813 loss: 0.2814074158668518 epoch: 814 loss: 0.2812522351741791 epoch: 815 loss: 0.2810974419116974 epoch: 816 loss: 0.28094279766082764 epoch: 817 loss: 0.2807884216308594 epoch: 818 loss: 0.2806342840194702 epoch: 819 loss: 0.2804804742336273 epoch: 820 loss: 0.2803266942501068 epoch: 821 loss: 0.28017333149909973 epoch: 822 loss: 0.2800200581550598 epoch: 823 loss: 0.2798670530319214 epoch: 824 loss: 0.279714435338974 epoch: 825 loss: 0.2795618772506714 epoch: 826 loss: 0.2794097661972046 epoch: 827 loss: 0.2792576849460602 epoch: 828 loss: 0.2791058421134949 epoch: 829 loss: 0.27895432710647583 epoch: 830 loss: 0.2788029611110687 epoch: 831 loss: 0.2786518931388855 epoch: 832 loss: 0.2785010039806366 epoch: 833 loss: 0.2783503532409668 epoch: 834 loss: 0.2781999409198761 epoch: 835 loss: 0.2780497968196869 epoch: 836 loss: 0.277899831533432 epoch: 837 loss: 0.27775004506111145 epoch: 838 loss: 0.27760064601898193 epoch: 839 loss: 0.2774512767791748 epoch: 840 loss: 0.27730220556259155 epoch: 841 loss: 0.2771534025669098 epoch: 842 loss: 0.27700483798980713 epoch: 843 loss: 0.27685636281967163 epoch: 844 loss: 0.27670830488204956 epoch: 845 loss: 0.27656036615371704 epoch: 846 loss: 0.2764126658439636 epoch: 847 loss: 0.2762651741504669 epoch: 848 loss: 0.27611789107322693 epoch: 849 loss: 0.27597084641456604 epoch: 850 loss: 0.2758239805698395 epoch: 851 loss: 0.2756774425506592 epoch: 852 loss: 0.2755310535430908 epoch: 853 loss: 0.2753848731517792 epoch: 854 loss: 0.27523890137672424 epoch: 855 loss: 0.2750932276248932 epoch: 856 loss: 0.27494776248931885 epoch: 857 loss: 0.27480238676071167 epoch: 858 loss: 0.27465736865997314 epoch: 859 loss: 0.27451249957084656 epoch: 860 loss: 0.2743678390979767 epoch: 861 loss: 0.2742234170436859 epoch: 862 loss: 0.27407917380332947 epoch: 863 loss: 0.2739351689815521 epoch: 864 loss: 0.2737914025783539 epoch: 865 loss: 0.27364787459373474 epoch: 866 loss: 0.2735045254230499 epoch: 867 loss: 0.27336129546165466 epoch: 868 loss: 0.27321842312812805 epoch: 869 loss: 0.273075670003891 epoch: 870 loss: 0.27293312549591064 epoch: 871 loss: 0.2727908194065094 epoch: 872 loss: 0.27264878153800964 epoch: 873 loss: 0.27250680327415466 epoch: 874 loss: 0.27236512303352356 epoch: 875 loss: 0.2722237706184387 epoch: 876 loss: 0.2720823884010315 epoch: 877 loss: 0.2719413638114929 epoch: 878 loss: 0.2718004882335663 epoch: 879 loss: 0.27165988087654114 epoch: 880 loss: 0.27151942253112793 epoch: 881 loss: 0.2713792622089386 epoch: 882 loss: 0.2712392210960388 epoch: 883 loss: 0.271099328994751 epoch: 884 loss: 0.2709597647190094 epoch: 885 loss: 0.27082037925720215 epoch: 886 loss: 0.27068111300468445 epoch: 887 loss: 0.2705421447753906 epoch: 888 loss: 0.27040329575538635 epoch: 889 loss: 0.2702646255493164 epoch: 890 loss: 0.27012625336647034 epoch: 891 loss: 0.2699880301952362 epoch: 892 loss: 0.26985010504722595 epoch: 893 loss: 0.2697122395038605 epoch: 894 loss: 0.26957467198371887 epoch: 895 loss: 0.26943716406822205 epoch: 896 loss: 0.2692999839782715 epoch: 897 loss: 0.26916295289993286 epoch: 898 loss: 0.2690262198448181 epoch: 899 loss: 0.26888954639434814 epoch: 900 loss: 0.26875317096710205 epoch: 901 loss: 0.26861685514450073 epoch: 902 loss: 0.2684808671474457 epoch: 903 loss: 0.26834505796432495 epoch: 904 loss: 0.26820939779281616 epoch: 905 loss: 0.2680738866329193 epoch: 906 loss: 0.2679387331008911 epoch: 907 loss: 0.2678036391735077 epoch: 908 loss: 0.26766881346702576 epoch: 909 loss: 0.26753419637680054 epoch: 910 loss: 0.2673996090888977 epoch: 911 loss: 0.26726531982421875 epoch: 912 loss: 0.2671312093734741 epoch: 913 loss: 0.266997367143631 epoch: 914 loss: 0.266863614320755 epoch: 915 loss: 0.2667301297187805 epoch: 916 loss: 0.26659682393074036 epoch: 917 loss: 0.26646360754966736 epoch: 918 loss: 0.26633065938949585 epoch: 919 loss: 0.26619789004325867 epoch: 920 loss: 0.2660652995109558 epoch: 921 loss: 0.2659328877925873 epoch: 922 loss: 0.26580071449279785 epoch: 923 loss: 0.26566869020462036 epoch: 924 loss: 0.2655368745326996 epoch: 925 loss: 0.26540520787239075 epoch: 926 loss: 0.2652737498283386 epoch: 927 loss: 0.26514241099357605 epoch: 928 loss: 0.26501142978668213 epoch: 929 loss: 0.2648804783821106 epoch: 930 loss: 0.26474976539611816 epoch: 931 loss: 0.26461923122406006 epoch: 932 loss: 0.2644888162612915 epoch: 933 loss: 0.2643587291240692 epoch: 934 loss: 0.2642287015914917 epoch: 935 loss: 0.2640989422798157 epoch: 936 loss: 0.2639692723751068 epoch: 937 loss: 0.26383981108665466 epoch: 938 loss: 0.26371052861213684 epoch: 939 loss: 0.26358145475387573 epoch: 940 loss: 0.26345258951187134 epoch: 941 loss: 0.2633238434791565 epoch: 942 loss: 0.263195276260376 epoch: 943 loss: 0.2630668878555298 epoch: 944 loss: 0.2629387676715851 epoch: 945 loss: 0.26281073689460754 epoch: 946 loss: 0.26268288493156433 epoch: 947 loss: 0.2625552713871002 epoch: 948 loss: 0.2624277174472809 epoch: 949 loss: 0.2623004615306854 epoch: 950 loss: 0.2621733546257019 epoch: 951 loss: 0.26204633712768555 epoch: 952 loss: 0.26191961765289307 epoch: 953 loss: 0.2617930471897125 epoch: 954 loss: 0.26166656613349915 epoch: 955 loss: 0.26154032349586487 epoch: 956 loss: 0.2614142596721649 epoch: 957 loss: 0.2612883746623993 epoch: 958 loss: 0.2611626386642456 epoch: 959 loss: 0.26103711128234863 epoch: 960 loss: 0.2609117329120636 epoch: 961 loss: 0.2607864737510681 epoch: 962 loss: 0.26066136360168457 epoch: 963 loss: 0.2605365514755249 epoch: 964 loss: 0.2604118585586548 epoch: 965 loss: 0.2602873146533966 epoch: 966 loss: 0.2601628601551056 epoch: 967 loss: 0.26003870368003845 epoch: 968 loss: 0.259914755821228 epoch: 969 loss: 0.2597907781600952 epoch: 970 loss: 0.2596670985221863 epoch: 971 loss: 0.259543776512146 epoch: 972 loss: 0.2594203054904938 epoch: 973 loss: 0.2592971920967102 epoch: 974 loss: 0.2591741681098938 epoch: 975 loss: 0.2590513527393341 epoch: 976 loss: 0.2589285969734192 epoch: 977 loss: 0.25880616903305054 epoch: 978 loss: 0.2586837708950043 epoch: 979 loss: 0.2585615813732147 epoch: 980 loss: 0.2584395706653595 epoch: 981 loss: 0.258317768573761 epoch: 982 loss: 0.25819605588912964 epoch: 983 loss: 0.25807446241378784 epoch: 984 loss: 0.25795313715934753 epoch: 985 loss: 0.25783199071884155 epoch: 986 loss: 0.25771090388298035 epoch: 987 loss: 0.25759005546569824 epoch: 988 loss: 0.2574693560600281 epoch: 989 loss: 0.25734880566596985 epoch: 990 loss: 0.25722837448120117 epoch: 991 loss: 0.2571081817150116 epoch: 992 loss: 0.25698813796043396 epoch: 993 loss: 0.2568681538105011 epoch: 994 loss: 0.2567485272884369 epoch: 995 loss: 0.2566289007663727 epoch: 996 loss: 0.2565094828605652 epoch: 997 loss: 0.25639018416404724 epoch: 998 loss: 0.2562710642814636 epoch: 999 loss: 0.2561522126197815

**Plotting Losses**

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



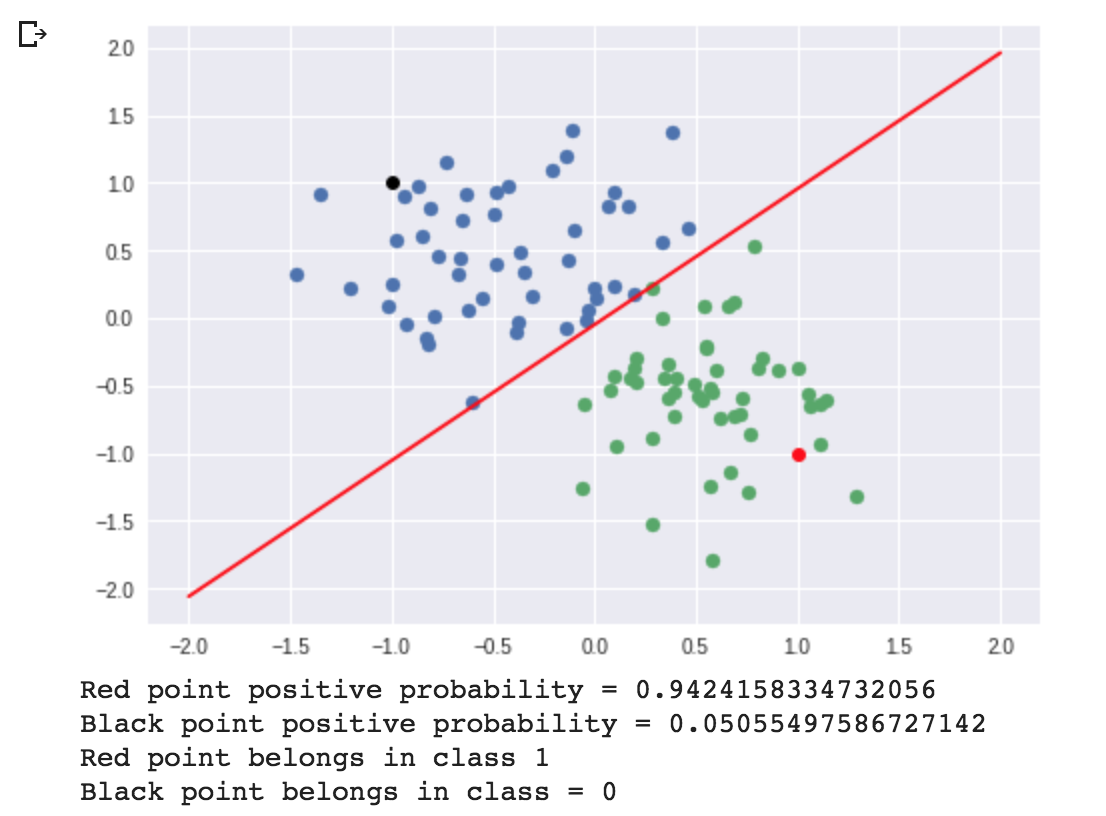
**Plotting Trained Model**

1. plot\_fit("Trained Model")



**Making Predictions**

1. point1 = torch.Tensor([1.0, -1.0])
2. point2 = torch.Tensor([-1.0, 1.0])
3. plt.plot(point1.numpy()[0], point1.numpy()[1], 'ro')
4. plt.plot(point2.numpy()[0], point2.numpy()[1], 'ko')
5. plot\_fit("Trained Model")
6. print("Red point positive probability = {}".format(model.forward(point1).item()))
7. print("Black point positive probability = {}".format(model.forward(point2).item()))
8. print("Red point belongs in class {}".format(model.predict(point1)))
9. print("Black point belongs in class = {}".format(model.predict(point2)))



**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',            ]        },    )