```
!pip3 install torch torchvision ipdb
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: torchvision in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: ipdb in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: pillow>=4.1.1 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: ipython>=5.1.0; python_version >= "3.4" in /usr/
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: olefile in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.6/di
Requirement already satisfied: pickleshare in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: pygments in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: decorator in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/p
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: pexpect; sys_platform != "win32" in /usr/local/l
Requirement already satisfied: wcwidth in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.6/dis
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.6/dist
```

```
 memory footprint support libraries/code
ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
pip install gputil
pip install psutil
pip install humanize
mport psutil
mport humanize
mport os
mport GPUtil as GPU
PUs = GPU.getGPUs()
 only one GPU on Colab and isn't guaranteed
pu = GPUs[0]
ef printm():
process = psutil.Process(os.getpid())
print("Gen RAM Free: " + humanize.naturalsize( psutil.virtual_memory().available ), "
print("GPU RAM Free: {0:.0f}MB | Used: {1:.0f}MB | Util {2:3.0f}% | Total {3:.0f}MB".f
rintm()
```

```
Collecting gputil
  Downloading https://files.pythonhosted.org/packages/ed/0e/5c61eedde9f6c87713e
Building wheels for collected packages: gputil
  Building wheel for gputil (setup.py) ... done
  Stored in directory: /root/.cache/pip/wheels/3d/77/07/80562de4bb0786e5ea18691
Successfully built gputil
Installing collected packages: gputil
Successfully installed gputil-1.4.0
Requirement already satisfied: psutil in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: humanize in /usr/local/lib/python3.6/dist-packag
Gen RAM Free: 12.9 GB  | Proc size: 119.7 MB
GPU RAM Free: 15079MB | Used: 0MB | Util   0% | Total 15079MB
```

```
!rm -r ./emergent-language
!git clone https://github.com/yoark/emergent-language.git
```

```
Cloning into 'emergent-language'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 385 (delta 12), reused 18 (delta 8), pack-reused 362
Receiving objects: 100% (385/385), 171.22 KiB | 908.00 KiB/s, done.
Resolving deltas: 100% (241/241), done.
```

```
!cd emergent-language && git checkout newtorch && git pull && git log -n1
```

```
Branch 'newtorch' set up to track remote branch 'newtorch' from 'origin'.
Switched to a new branch 'newtorch'
Already up to date.
commit 2b252e2559f2133348715d493759958c4fabdc37 (HEAD -> newtorch, origin/newto
Author: Yoark <yoark.yang@gmail.com>
Date:   Fri Apr 26 16:51:16 2019 -0600

    add print prob
```

```
!ls
!cd emergent-language && ls
```

```
emergent-language  sample_data
comp-graph.pdf  modules           num_per_timestamp.png  test.png
configs.py      notes.txt         playground.py          train.py
constants.py    num_per_epoch.png README.md              visualize.py
```

```
python
```

```
import os
if os.getcwd() != '/content/emergent-language':
  os.chdir('emergent-language')

# train.py

import numpy as np
import torch
from torch.optim import RMSprop
from torch.optim.lr_scheduler import ReduceLROnPlateau
import configs
from modules.agent import AgentModule
from modules.game import GameModule
from collections import defaultdict
import sys
import argparse


def create_parser():
  parser = argparse.ArgumentParser(description="Trains the agents for cooperative comm
  parser.add_argument('--no-utterances', action='store_true', help='if specified disab
  parser.add_argument('--penalize-words', action='store_true', help='if specified pena
  parser.add_argument('--n-epochs', '-e', type=int, help='if specified sets number of
  parser.add_argument('--learning-rate', type=float, help='if specified sets learning
  parser.add_argument('--batch-size', type=int, help='if specified sets batch size(def
  parser.add_argument('--n-timesteps', '-t', type=int, help='if specified sets timeste
  parser.add_argument('--num-shapes', '-s', type=int, help='if specified sets number c
  parser.add_argument('--num-colors', '-c', type=int, help='if specified sets number c
  parser.add_argument('--max-agents', type=int, help='if specified sets maximum number
  parser.add_argument('--min-agents', type=int, help='if specified sets minimum number
  parser.add_argument('--max-landmarks', type=int, help='if specified sets maximum num
  parser.add_argument('--min-landmarks', type=int, help='if specified sets minimum num
  parser.add_argument('--vocab-size', '-v', type=int, help='if specified sets maximum
  parser.add_argument('--world-dim', '-w', type=int, help='if specified sets the side
  parser.add_argument('--oov-prob', '-o', type=int, help='higher value penalize uncomm
  parser.add_argument('--load-model-weights', type=str, help='if specified start with
  parser.add_argument('--save-model-weights', type=str, help='if specified save the mo
  parser.add_argument('--use-cuda', action='store_true', help='if specified enables tr
  return parser

def print_losses(epoch, losses, dists, game_config):
    for a in range(game_config.min_agents, game_config.max_agents + 1):
        for l in range(game_config.min_landmarks,
                       game_config.max_landmarks + 1):
            loss_al = losses[a][l]
            loss = loss_al[-1] if loss_al else 0
            min_loss = min(loss_al) if loss_al else 0

            dist_al = dists[a][l]
            dist = dist_al[-1] if dist_al else 0
            min_dist = min(dist_al) if dist_al else 0

            print(
                "[epoch %d][%d agents, %d landmarks][%d batches][last loss: %f][min lo
                % (epoch, a, l, len(loss_al), loss, min_loss, dist, min_dist))
    print("_________________________")



#     args_list = sys.argv[1:]
args_list = ['-e', '1000', '-t', '20', '--penalize-words', '--use-cuda']
parser = create_parser()
args = vars(parser.parse_args(args_list))
agent_config = configs.get_agent_config(args)
game_config = configs.get_game_config(args)
training_config = configs.get_training_config(args)
print("Training with config:")
print(training_config)
print(game_config)
print(agent_config)
agent = AgentModule(agent_config)
if training_config.use_cuda:
    agent.cuda()
optimizer = RMSprop(agent.parameters(), lr=training_config.learning_rate)
scheduler = ReduceLROnPlateau(optimizer, 'min', verbose=True, cooldown=5)
losses = defaultdict(lambda:defaultdict(list))
dists = defaultdict(lambda:defaultdict(list))
probs = defaultdict(list)
for epoch in range(training_config.num_epochs):
    num_agents = np.random.randint(game_config.min_agents, game_config.max_agents+1)
    num_agents = 2
    num_landmarks = np.random.randint(game_config.min_landmarks, game_config.max_landm
    agent.reset()
    game = GameModule(game_config, num_agents, num_landmarks)
    if training_config.use_cuda:
        game.cuda()
    optimizer.zero_grad()

    total_loss, _, num_utter, utter_num_t, prob = agent(game)
    if epoch % 10 == 0:
      print(prob)
#     if num_agents == 2:
#       probs[num_agents].append(prob)
#       print(prob)
#     else:
#       probs[num_agents].append(prob)
    #print(utter_num_t)
    per_agent_loss = total_loss.data / num_agents / game_config.batch_size
    losses[num_agents][num_landmarks].append(per_agent_loss)
    dist = game.get_avg_agent_to_goal_distance()
    avg_dist = dist.data / num_agents / game_config.batch_size
    dists[num_agents][num_landmarks].append(avg_dist)

    print_losses(epoch, losses, dists, game_config)

    total_loss.backward()
    optimizer.step()

    if num_agents == game_config.max_agents and num_landmarks == game_config.max_landm
        scheduler.step(losses[game_config.max_agents][game_config.max_landmarks][-1])

if training_config.save_model:
    torch.save(agent, training_config.save_model_file)
    print("Saved agent model weights at %s" % training_config.save_model_file)
    """
    import code
    code.interact(local=locals())
    """
# torch.cuda.empty_cache()
```

```
[epoch 970][2 agents, 3 landmarks][971 batches][last loss: 340.874084][min loss
[epoch 970][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 971][2 agents, 3 landmarks][972 batches][last loss: 313.638062][min loss
[epoch 971][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 972][2 agents, 3 landmarks][973 batches][last loss: 319.968109][min loss
[epoch 972][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 973][2 agents, 3 landmarks][974 batches][last loss: 331.355286][min loss
[epoch 973][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 974][2 agents, 3 landmarks][975 batches][last loss: 326.982452][min loss
[epoch 974][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 975][2 agents, 3 landmarks][976 batches][last loss: 305.964722][min loss
[epoch 975][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 976][2 agents, 3 landmarks][977 batches][last loss: 321.468964][min loss
[epoch 976][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 977][2 agents, 3 landmarks][978 batches][last loss: 327.328369][min loss
[epoch 977][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 978][2 agents, 3 landmarks][979 batches][last loss: 322.294189][min loss
[epoch 978][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 979][2 agents, 3 landmarks][980 batches][last loss: 342.181305][min loss
[epoch 979][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
tensor([2.9297e-04, 4.9292e-01, 2.4414e-04, 2.5552e-01, 1.1230e-03, 1.4160e-03,
        2.3252e-01, 3.9063e-04, 4.3945e-04, 8.7891e-04, 1.6602e-03, 4.3945e-04,
        1.1719e-03, 1.0254e-03, 1.7090e-03, 6.3477e-04, 5.8594e-04, 2.9297e-03,
        3.4180e-03, 6.8359e-04], device='cuda:0')
[epoch 980][2 agents, 3 landmarks][981 batches][last loss: 319.047424][min loss
[epoch 980][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 981][2 agents, 3 landmarks][982 batches][last loss: 338.268097][min loss
[epoch 981][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 982][2 agents, 3 landmarks][983 batches][last loss: 326.054108][min loss
[epoch 982][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 983][2 agents, 3 landmarks][984 batches][last loss: 321.895782][min loss
[epoch 983][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 984][2 agents, 3 landmarks][985 batches][last loss: 326.725586][min loss
[epoch 984][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 985][2 agents, 3 landmarks][986 batches][last loss: 318.650696][min loss
[epoch 985][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 986][2 agents, 3 landmarks][987 batches][last loss: 326.537506][min loss
[epoch 986][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
[epoch 987][2 agents, 3 landmarks][988 batches][last loss: 321.610443][min loss
[epoch 987][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
________________________
```

```
[epoch 988][2 agents, 3 landmarks][989 batches][last loss: 319.538544][min loss
[epoch 988][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 989][2 agents, 3 landmarks][990 batches][last loss: 324.691223][min loss
[epoch 989][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
tensor([0.0007, 0.4581, 0.0010, 0.3267, 0.0017, 0.0020, 0.1837, 0.0005, 0.0009,
        0.0009, 0.0026, 0.0008, 0.0016, 0.0014, 0.0037, 0.0010, 0.0014, 0.0035,
        0.0067, 0.0013], device='cuda:0')
[epoch 990][2 agents, 3 landmarks][991 batches][last loss: 329.528381][min loss
[epoch 990][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 991][2 agents, 3 landmarks][992 batches][last loss: 322.662537][min loss
[epoch 991][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 992][2 agents, 3 landmarks][993 batches][last loss: 317.860260][min loss
[epoch 992][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 993][2 agents, 3 landmarks][994 batches][last loss: 318.638611][min loss
[epoch 993][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 994][2 agents, 3 landmarks][995 batches][last loss: 330.204224][min loss
[epoch 994][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 995][2 agents, 3 landmarks][996 batches][last loss: 317.149536][min loss
[epoch 995][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 996][2 agents, 3 landmarks][997 batches][last loss: 323.999603][min loss
[epoch 996][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 997][2 agents, 3 landmarks][998 batches][last loss: 334.917389][min loss
[epoch 997][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 998][2 agents, 3 landmarks][999 batches][last loss: 320.668121][min loss
[epoch 998][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
[epoch 999][2 agents, 3 landmarks][1000 batches][last loss: 330.824188][min los
[epoch 999][3 agents, 3 landmarks][0 batches][last loss: 0.000000][min loss: 0.
_________________________
Saved agent model weights at latest.pt
```