Noisy Deep Q-Networks

```
!apt-get update && apt-get install -y xvfb
```

```
Unpacking libxkbfile1:amd64 (1:1.1.0-1build3) ...
Selecting previously unselected package x11-xkb-utils.
Preparing to unpack .../3-x11-xkb-utils_7.7+5build4_amd64.deb ...
Unpacking x11-xkb-utils (7.7+5build4) ...
Selecting previously unselected package xfonts-encodings.
Preparing to unpack .../4-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb ...
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) ...
Selecting previously unselected package xfonts-utils.
Preparing to unpack .../5-xfonts-utils_1%3a7.7+6build2_amd64.deb ...
Unpacking xfonts-utils (1:7.7+6build2) ...
Selecting previously unselected package xfonts-base.
Preparing to unpack .../6-xfonts-base_1%3a1.0.5_all.deb ...
Unpacking xfonts-base (1:1.0.5) ...
Selecting previously unselected package xserver-common.
Preparing to unpack .../7-xserver-common_2%3a21.1.4-2ubuntu1.7~22.04.12_all.deb ...
Unpacking xserver-common (2:21.1.4-2ubuntu1.7~22.04.12) ...
Selecting previously unselected package xvfb.
Preparing to unpack .../8-xvfb_2%3a21.1.4-2ubuntu1.7~22.04.12_amd64.deb ...
Unpacking xvfb (2:21.1.4-2ubuntu1.7~22.04.12) ...
Setting up libfontenc1:amd64 (1:1.1.4-1build3) ...
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) ...
Setting up libxkbfile1:amd64 (1:1.1.0-1build3) ...
Setting up libxfont2:amd64 (1:2.0.5-1build1) ...
Setting up x11-xkb-utils (7.7+5build4) ...
Setting up xfonts-utils (1:7.7+6build2) ...
Setting up xfonts-base (1:1.0.5) ...
Setting up xserver-common (2:21.1.4-2ubuntu1.7~22.04.12) ...
Setting up xvfb (2:21.1.4-2ubuntu1.7~22.04.12) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link
```

```
!pip install gym==0.17.3
```

```
Collecting gym==0.17.3
  Downloading gym-0.17.3.tar.gz (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 54.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from gym==0.17.3) (1.13.1)
Requirement already satisfied: numpy>=1.10.4 in /usr/local/lib/python3.10/dist-packages (from gym==0.17.3) (1.26.4)
Collecting pyglet<=1.5.0,>=1.4.0 (from gym==0.17.3)
  Downloading pyglet-1.5.0-py2.py3-none-any.whl.metadata (7.6 kB)
Collecting cloudpickle<1.7.0,>=1.2.0 (from gym==0.17.3)
  Downloading cloudpickle-1.6.0-py3-none-any.whl.metadata (4.3 kB)
Requirement already satisfied: future in /usr/local/lib/python3.10/dist-packages (from pyglet<=1.5.0,>=1.4.0->gym==0.17.3) (1.0.0)
Downloading cloudpickle-1.6.0-py3-none-any.whl (23 kB)
Downloading pyglet-1.5.0-py2.py3-none-any.whl (1.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 54.4 MB/s eta 0:00:00
Building wheels for collected packages: gym
  Building wheel for gym (setup.py) ... done
  Created wheel for gym: filename=gym-0.17.3-py3-none-any.whl size=1654616 sha256=c7769093c17a7c8112c8705c6caf65e58e14868838613c0335
  Stored in directory: /root/.cache/pip/wheels/af/4b/74/fcfc8238472c34d7f96508a63c962ff3ac9485a9a4137afd4e
Successfully built gym
Installing collected packages: pyglet, cloudpickle, gym
```

```
  Attempting uninstall: cloudpickle
    Found existing installation: cloudpickle 3.1.0
    Uninstalling cloudpickle-3.1.0:
      Successfully uninstalled cloudpickle-3.1.0
  Attempting uninstall: gym
    Found existing installation: gym 0.25.2
    Uninstalling gym-0.25.2:
      Successfully uninstalled gym-0.25.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
bigframes 1.27.0 requires cloudpickle>=2.0.0, but you have cloudpickle 1.6.0 which is incompatible.
dask 2024.10.0 requires cloudpickle>=3.0.0, but you have cloudpickle 1.6.0 which is incompatible.
Successfully installed cloudpickle-1.6.0 gym-0.17.3 pyglet-1.5.0
```

```
!pip install pygame
```

```
Requirement already satisfied: pygame in /usr/local/lib/python3.10/dist-packages (2.6.1)
```

```
!pip install stable-baselines3==1.4.0
```

```
Collecting stable-baselines3==1.4.0
  Downloading stable_baselines3-1.4.0-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: gym<0.20,>=0.17 in /usr/local/lib/python3.10/dist-packages (from stable-baselines3==1.4.0) (0.17.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from stable-baselines3==1.4.0) (1.26.4)
Requirement already satisfied: torch>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from stable-baselines3==1.4.0) (2.5.1+cu121
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from stable-baselines3==1.4.0) (1.6.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from stable-baselines3==1.4.0) (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from stable-baselines3==1.4.0) (3.8.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from gym<0.20,>=0.17->stable-baselines3==1.4.0) (1
Requirement already satisfied: pyglet<=1.5.0,>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from gym<0.20,>=0.17->stable-baseli
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.1->stable-baselines3==1.4.0) (3
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.1->stable-baseli
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.1->stable-baselines3==1.4.0) (3
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.1->stable-baselines3==1.4.0) (3.1
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.1->stable-baselines3==1.4.0) (2024
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.1->stable-baselines3==1.4.0
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch>=1.8.1->stab
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4.0
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==1.4
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->stable-baselines3==
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->stable-baselines3==1.4.0) (2024
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas->stable-baselines3==1.4.0) (20
Requirement already satisfied: future in /usr/local/lib/python3.10/dist-packages (from pyglet<=1.5.0,>=1.4.0->gym<0.20,>=0.17->stabl
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->stable-ba
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.8.1->stable-baselin
Downloading stable_baselines3-1.4.0-py3-none-any.whl (176 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 176.9/176.9 kB 11.5 MB/s eta 0:00:00
Installing collected packages: stable-baselines3
Successfully installed stable-baselines3-1.4.0
```

```
!pip install pytorch-lightning
```

```
Collecting pytorch-lightning
  Downloading pytorch_lightning-2.4.0-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: torch>=2.1.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning) (2.5.1+cu121)
Requirement already satisfied: tqdm>=4.57.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning) (4.66.6)
Requirement already satisfied: PyYAML>=5.4 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning) (6.0.2)
Requirement already satisfied: fsspec>=2022.5.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0->pytorch-lig
Collecting torchmetrics>=0.7.0 (from pytorch-lightning)
  Downloading torchmetrics-1.6.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning) (24.2)
Requirement already satisfied: typing-extensions>=4.4.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning) (4.12.2
Collecting lightning-utilities>=0.10.0 (from pytorch-lightning)
  Downloading lightning_utilities-0.11.9-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0->p
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from lightning-utilities>=0.10.0->pytorch-ligh
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-lightning) (3.16.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-lightning) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-lightning) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-lightning) (1.13
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch>=2.1.0->pytc
Requirement already satisfied: numpy>1.20.0 in /usr/local/lib/python3.10/dist-packages (from torchmetrics>=0.7.0->pytorch-lightning
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1-
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[htt
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fssp
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec
Requirement already satisfied: async-timeout<6.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1-
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=2.1.0->pytorch-lightr
```

```
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.17.0->aiohttp!=4.0.0a0,!=4.0
Downloading pytorch_lightning-2.4.0-py3-none-any.whl (815 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 815.2/815.2 kB 27.1 MB/s eta 0:00:00
Downloading lightning_utilities-0.11.9-py3-none-any.whl (28 kB)
Downloading torchmetrics-1.6.0-py3-none-any.whl (926 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 926.4/926.4 kB 35.0 MB/s eta 0:00:00
Installing collected packages: lightning-utilities, torchmetrics, pytorch-lightning
Successfully installed lightning-utilities-0.11.9 pytorch-lightning-2.4.0 torchmetrics-1.6.0
```

```
!pip install pyvirtualdisplay
```

```
Collecting pyvirtualdisplay
  Downloading PyVirtualDisplay-3.0-py3-none-any.whl.metadata (943 bytes)
Downloading PyVirtualDisplay-3.0-py3-none-any.whl (15 kB)
Installing collected packages: pyvirtualdisplay
Successfully installed pyvirtualdisplay-3.0
```

```
!pip install git+https://github.com/GrupoTuring/PyGame-Learning-Environment
```

```
Collecting git+https://github.com/GrupoTuring/PyGame-Learning-Environment
  Cloning https://github.com/GrupoTuring/PyGame-Learning-Environment to /tmp/pip-req-build-gur90g1x
  Running command git clone --filter=blob:none --quiet https://github.com/GrupoTuring/PyGame-Learning-Environment /tmp/pip-req-buil
  Resolved https://github.com/GrupoTuring/PyGame-Learning-Environment to commit 52ace013e3ea2fe5df08df98ec4dda902801e9df
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from ple==0.0.2) (1.26.4)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from ple==0.0.2) (11.0.0)
Building wheels for collected packages: ple
  Building wheel for ple (setup.py) ... done
  Created wheel for ple: filename=ple-0.0.2-py3-none-any.whl size=722357 sha256=ecf48e351a0d3e31bb62fe0b8f45aa7991c4d08322e58b8e684c
  Stored in directory: /tmp/pip-ephem-wheel-cache-evig_fhv/wheels/10/b1/df/d464fcb2796fd6bc3bcc8bfb63243b9a007492378ae4204806
Successfully built ple
Installing collected packages: ple
Successfully installed ple-0.0.2
```

```
!pip install git+https://github.com/lusob/gym-ple
```

```
Collecting git+https://github.com/lusob/gym-ple
  Cloning https://github.com/lusob/gym-ple to /tmp/pip-req-build-lne70jg_
  Running command git clone --filter=blob:none --quiet https://github.com/lusob/gym-ple /tmp/pip-req-build-lne70jg_
  Resolved https://github.com/lusob/gym-ple to commit 7cedbf4e31be86f5ca2aae5c0dfd9d38825af64e
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: gym_ple
  Building wheel for gym_ple (setup.py) ... done
  Created wheel for gym_ple: filename=gym_ple-0.3-py3-none-any.whl size=5321 sha256=b4cc15e6f5716e480274b04effd71f78ef663ddee6fe464a
  Stored in directory: /tmp/pip-ephem-wheel-cache-mcrrmxml/wheels/ba/e1/35/46d7b0fc0e941e9cf345d94283f45aa090b7e634ee15876cb5
Successfully built gym_ple
Installing collected packages: gym_ple
Successfully installed gym_ple-0.3
```

Setup virtual display

```
from pyvirtualdisplay import Display
Display(visible=False, size=(1400, 900)).start()
```

```
<pyvirtualdisplay.display.Display at 0x7926e4a0eef0>
```

Import the necessary code libraries

```
import copy
import torch
import random
import gym
import gym_ple
import matplotlib

import numpy as np
import torch.nn.functional as F

import matplotlib.pyplot as plt
import matplotlib.animation as animation

from collections import deque, namedtuple
from IPython.display import HTML
from base64 import b64encode

from torch import nn
```

```
from torch.utils.data import DataLoader
from torch.utils.data.dataset import IterableDataset
from torch.optim import AdamW

from pytorch_lightning import LightningModule, Trainer

from gym.wrappers import TransformObservation

from stable_baselines3.common.atari_wrappers import MaxAndSkipEnv, WarpFrame


device = 'cuda:0' if torch.cuda.is_available() else 'cpu'
num_gpus = torch.cuda.device_count()
```

```
pygame 2.6.1 (SDL 2.28.4, Python 3.10.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
couldn't import doomish
Couldn't import doom
```

```
# Copied from: https://colab.research.google.com/github/deepmind/dm_control/blob/master/tutorial.ipynb#scrollTo=gKc1FNhKiVJX

def display_video(frames, framerate=30):
  height, width, _ = frames[0].shape
  dpi = 70
  orig_backend = matplotlib.get_backend()
  matplotlib.use('Agg')
  fig, ax = plt.subplots(1, 1, figsize=(width / dpi, height / dpi), dpi=dpi)
  matplotlib.use(orig_backend)
  ax.set_axis_off()
  ax.set_aspect('equal')
  ax.set_position([0, 0, 1, 1])
  im = ax.imshow(frames[0])
  def update(frame):
    im.set_data(frame)
    return [im]
  interval = 1000/framerate
  anim = animation.FuncAnimation(fig=fig, func=update, frames=frames,
                                 interval=interval, blit=True, repeat=False)
  return HTML(anim.to_html5_video())
```

Create the Deep Q-Network

```
import math
from torch.nn.init import kaiming_uniform_, zeros_

class NoisyLinear(nn.Module):

  def __init__(self, in_features, out_features, sigma):
    super(NoisyLinear, self).__init__()
    self.w_mu = nn.Parameter(torch.empty((out_features, in_features)))
    self.w_sigma = nn.Parameter(torch.empty((out_features, in_features)))
    self.b_mu = nn.Parameter(torch.empty((out_features)))
    self.b_sigma = nn.Parameter(torch.empty((out_features)))

    kaiming_uniform_(self.w_mu, a=math.sqrt(5))
    kaiming_uniform_(self.w_sigma, a=math.sqrt(5))
    zeros_(self.b_mu)
    zeros_(self.b_sigma)

  def forward(self, x, sigma=0.5):
    if self.training:
      w_noise = torch.normal(0, sigma, size=self.w_mu.size()).to(device)
      b_noise = torch.normal(0, sigma, size=self.b_mu.size()).to(device)
      return F.linear(x, self.w_mu + self.w_sigma * w_noise, self.b_mu + self.b_sigma * b_noise)
    else:
      return F.linear(x, self.W_mu, self.b_mu)
```

```
class DQN(nn.Module):

  def __init__(self, hidden_size, obs_shape, n_actions, sigma=0.5):
    super().__init__()

    self.conv = nn.Sequential(
      nn.Conv2d(obs_shape[0], 64, kernel_size=3),
      nn.MaxPool2d(kernel_size=4),
      nn.ReLU(),
      nn.Conv2d(64, 16, kernel_size=3),
      nn.MaxPool2d(kernel_size=4),
      nn.ReLU()
```

```
    )
    conv_out_size = self._get_conv_out(obs_shape)
    print(conv_out_size)
    self.head = nn.Sequential(
      NoisyLinear(conv_out_size, hidden_size, sigma=sigma),
      nn.ReLU(),
    )

    self.fc_adv = NoisyLinear(hidden_size, n_actions, sigma=sigma)
    self.fc_value = NoisyLinear(hidden_size, 1, sigma=sigma)

  def _get_conv_out(self, shape):
    conv_out = self.conv(torch.zeros(1, *shape))
    return int(np.prod(conv_out.size()))

  def forward(self, x):
    x = self.conv(x.float()).view(x.size()[0], -1)
    x = self.head(x)
    adv = self.fc_adv(x)
    value = self.fc_value(x)
    return value + adv - torch.mean(adv, dim=1, keepdim=True)
```

Create the policy

```
def greedy(state, net):
  state = torch.tensor([state]).to(device)
  q_values = net(state)
  _, action = torch.max(q_values, dim=1)
  action = int(action.item())
  return action
```

Create the replay buffer

```
class ReplayBuffer:

  def __init__(self, capacity):
    self.buffer = deque(maxlen=capacity)
    self.priorities = deque(maxlen=capacity)
    self.capacity = capacity
    self.alpha = 0.0  # anneal.
    self.beta = 1.0  # anneal.
    self.max_priority = 0.0

  def __len__(self):
    return len(self.buffer)

  def append(self, experience):
    self.buffer.append(experience)
    self.priorities.append(self.max_priority)

  def update(self, index, priority):
    if priority > self.max_priority:
      self.max_priority = priority
    self.priorities[index] = priority

  def sample(self, batch_size):
    prios = np.array(self.priorities, dtype=np.float64) + 1e-4 # Stability constant.
    prios = prios ** self.alpha
    probs = prios / prios.sum()

    weights = (self.__len__() * probs) ** -self.beta
    weights = weights / weights.max()

    idx = random.choices(range(self.__len__()), weights=probs, k=batch_size)
    sample = [(i, weights[i], *self.buffer[i]) for i in idx]
    return sample
```

```
class RLDataset(IterableDataset):

  def __init__(self, buffer, sample_size=400):
    self.buffer = buffer
    self.sample_size = sample_size

  def __iter__(self):
    for experience in self.buffer.sample(self.sample_size):
      yield experience
```

Create the environment

```
class RunningMeanStd:
    # https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance#Parallel_algorithm
    def __init__(self, epsilon=1e-4, shape=()):
        self.mean = np.zeros(shape, "float64")
        self.var = np.ones(shape, "float64")
        self.count = epsilon

    def update(self, x):
        batch_mean = np.mean(x, axis=0)
        batch_var = np.var(x, axis=0)
        batch_count = x.shape[0]
        self.update_from_moments(batch_mean, batch_var, batch_count)

    def update_from_moments(self, batch_mean, batch_var, batch_count):
        self.mean, self.var, self.count = update_mean_var_count_from_moments(
            self.mean, self.var, self.count, batch_mean, batch_var, batch_count
        )


def update_mean_var_count_from_moments(
    mean, var, count, batch_mean, batch_var, batch_count
):
    delta = batch_mean - mean
    tot_count = count + batch_count

    new_mean = mean + delta * batch_count / tot_count
    m_a = var * count
    m_b = batch_var * batch_count
    M2 = m_a + m_b + np.square(delta) * count * batch_count / tot_count
    new_var = M2 / tot_count
    new_count = tot_count

    return new_mean, new_var, new_count


class NormalizeObservation(gym.core.Wrapper):
    def __init__(
        self,
        env,
        epsilon=1e-8,
    ):
        super().__init__(env)
        self.num_envs = getattr(env, "num_envs", 1)
        self.is_vector_env = getattr(env, "is_vector_env", False)
        if self.is_vector_env:
            self.obs_rms = RunningMeanStd(shape=self.single_observation_space.shape)
        else:
            self.obs_rms = RunningMeanStd(shape=self.observation_space.shape)
        self.epsilon = epsilon

    def step(self, action):
        obs, rews, dones, infos = self.env.step(action)
        if self.is_vector_env:
            obs = self.normalize(obs)
        else:
            obs = self.normalize(np.array([obs]))[0]
        return obs, rews, dones, infos

    def reset(self, **kwargs):
        return_info = kwargs.get("return_info", False)
        if return_info:
            obs, info = self.env.reset(**kwargs)
        else:
            obs = self.env.reset(**kwargs)
        if self.is_vector_env:
            obs = self.normalize(obs)
        else:
            obs = self.normalize(np.array([obs]))[0]
        if not return_info:
            return obs
        else:
            return obs, info

    def normalize(self, obs):
        self.obs_rms.update(obs)
        return (obs - self.obs_rms.mean) / np.sqrt(self.obs_rms.var + self.epsilon)
```

```
class NormalizeReward(gym.core.Wrapper):
    def __init__(
        self,
        env,
        gamma=0.99,
        epsilon=1e-8,
    ):
        super().__init__(env)
        self.num_envs = getattr(env, "num_envs", 1)
        self.is_vector_env = getattr(env, "is_vector_env", False)
        self.return_rms = RunningMeanStd(shape=())
        self.returns = np.zeros(self.num_envs)
        self.gamma = gamma
        self.epsilon = epsilon

    def step(self, action):
        obs, rews, dones, infos = self.env.step(action)
        if not self.is_vector_env:
            rews = np.array([rews])
        self.returns = self.returns * self.gamma + rews
        rews = self.normalize(rews)
        self.returns[dones] = 0.0
        if not self.is_vector_env:
            rews = rews[0]
        return obs, rews, dones, infos

    def normalize(self, rews):
        self.return_rms.update(self.returns)
        return rews / np.sqrt(self.return_rms.var + self.epsilon)
```

```
env = gym.make('Catcher-v0')
```

```
/usr/local/lib/python3.10/dist-packages/gym/logger.py:30: UserWarning: WARN: Environment '<class 'gym_ple.ple_env.PLEEnv'>' has depr
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
```

```
obs = env.reset()
obs.shape
```

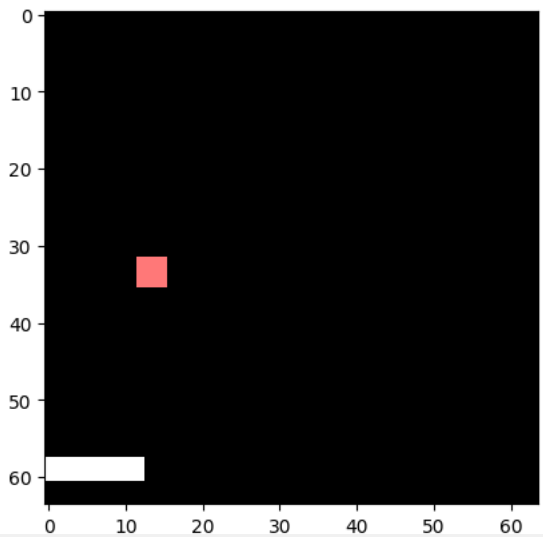
```
(64, 64, 3)
```

```
env.observation_space, env.action_space
```

```
(Box(0, 255, (64, 64, 3), uint8), Discrete(3))
```

```
for i in range(20):
  obs, rew, done, info = env.step(env.action_space.sample())

plt.imshow(obs)
```

```
<matplotlib.image.AxesImage at 0x7925aee0b160>
```




```
env = MaxAndSkipEnv(env, skip=2)
```

```
obs = env.reset()
```

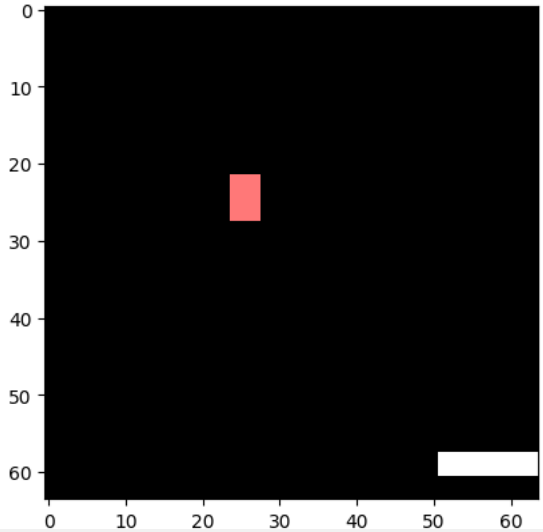
```
for i in range(10):
  obs, _, _, _ = env.step(env.action_space.sample())
```

```
type(obs), obs.shape
```

```
(numpy.ndarray, (64, 64, 3))
```

```
plt.imshow(obs)
```

```
<matplotlib.image.AxesImage at 0x7925ab55c2b0>
```




```
env = WarpFrame(env, height=42, width=42)
```

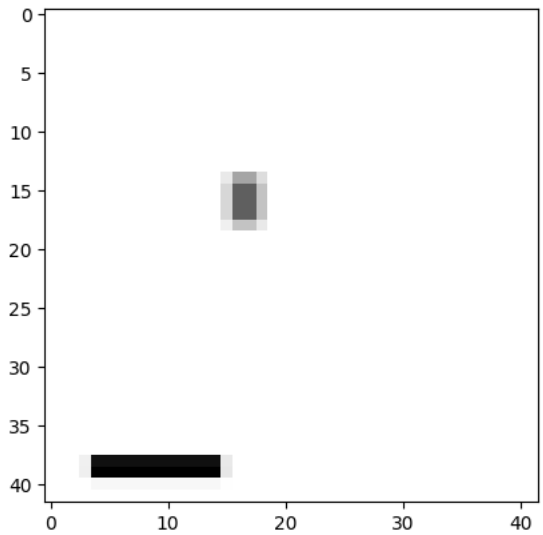
```
obs = env.reset()

for i in range(10):
  obs, _, _, _ = env.step(env.action_space.sample())
```

```
type(obs), obs.shape
```

```
(numpy.ndarray, (42, 42, 1))
```

```
plt.imshow(obs.squeeze(), cmap='gray_r')
```

```
<matplotlib.image.AxesImage at 0x7925ab578520>
```




```
obs.min(), obs.max()
```

```
(0, 255)
```

```
env = TransformObservation(env, lambda x: x.swapaxes(-1, 0))
env.observation_space = gym.spaces.Box(low=0, high=255, shape=(1, 42, 42), dtype=np.float32)
```

```
obs = env.reset()

for i in range(10):
  obs, _, _, _ = env.step(env.action_space.sample())
```

```
obs.shape
```

```
(1, 42, 42)
```

```
def create_environment(env_name):
  env = gym_ple.make(env_name)
  env = MaxAndSkipEnv(env, skip=2)
  env = WarpFrame(env, height=42, width=42)
  env = TransformObservation(env, lambda x: x.swapaxes(-1, 0))
  env.observation_space = gym.spaces.Box(low=0, high=255, shape=(1, 42, 42), dtype=np.float32)
  env = NormalizeObservation(env)
  env = NormalizeReward(env)
  return env
```

Create the Deep Q-Learning algorithm

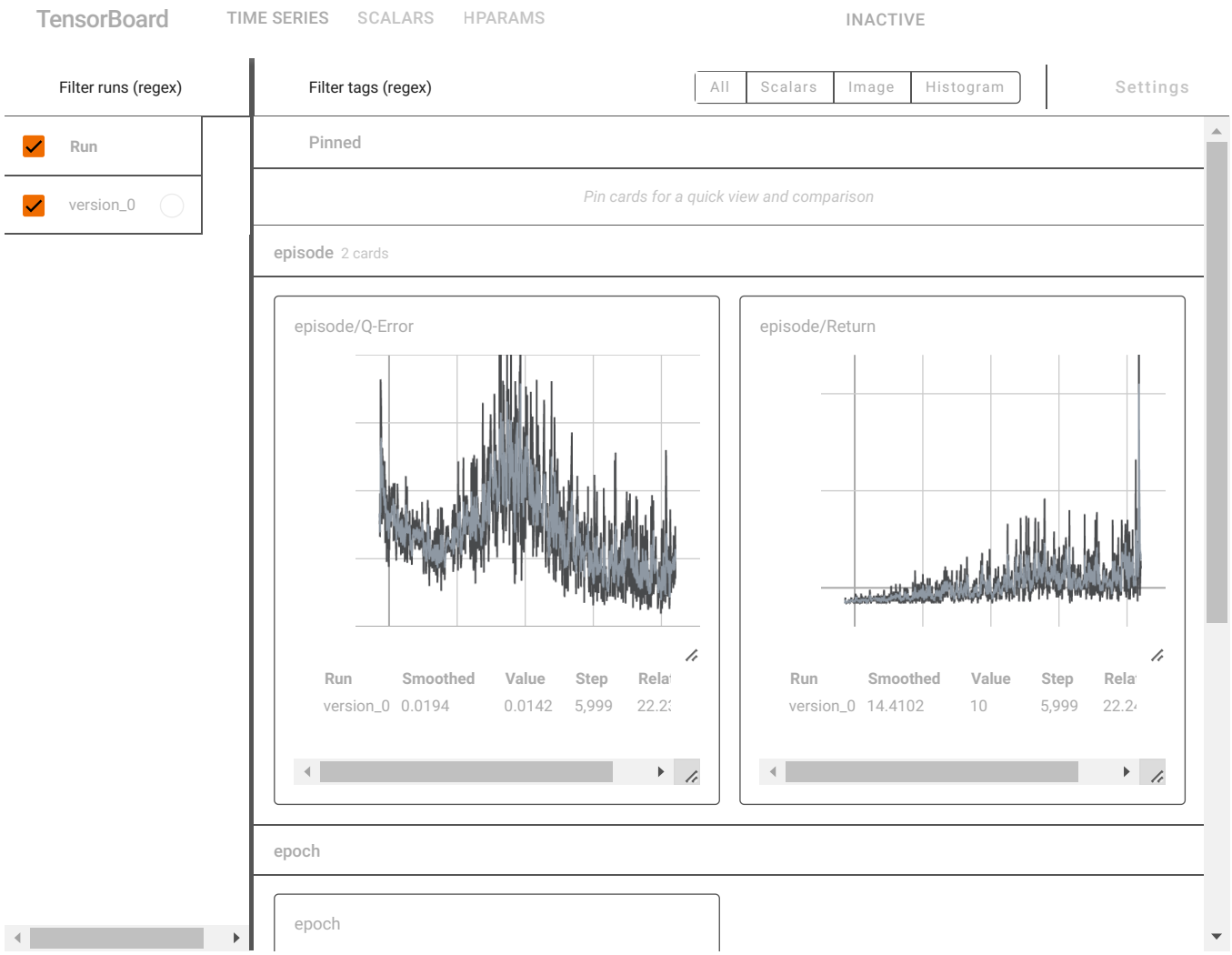
```
class DeepQLearning(LightningModule):

  # Initialize.
  def __init__(self, env_name, policy=greedy, capacity=100_000,
               batch_size=256, lr=1e-3, hidden_size=128, gamma=0.99,
               loss_fn=F.smooth_l1_loss, optim=AdamW, samples_per_epoch=1_000,
               sync_rate=10, sigma=0.5, a_start=0.5, a_end=0.0, a_last_episode=100,
               b_start=0.4, b_end=1.0, b_last_episode=100):

    super().__init__()
    self.env = create_environment(env_name)

    obs_size = self.env.observation_space.shape
    n_actions = self.env.action_space.n

    self.q_net = DQN(hidden_size, obs_size, n_actions, sigma=sigma)

    self.target_q_net = copy.deepcopy(self.q_net)

    self.policy = policy
    self.buffer = ReplayBuffer(capacity=capacity)

    self.save_hyperparameters()

    while len(self.buffer) < self.hparams.samples_per_epoch:
      print(f"{len(self.buffer)} samples in experience buffer. Filling...")
      self.play_episode()

  @torch.no_grad()
  def play_episode(self, policy=None):
    state = self.env.reset()
    done = False

    while not done:
      if policy:
        action = policy(state, self.q_net)
      else:
        action = self.env.action_space.sample()
      next_state, reward, done, info = self.env.step(action)
      exp = (state, action, reward, done, next_state)
      self.buffer.append(exp)
      state = next_state

  # Forward.
  def forward(self, x):
    return self.q_net(x)

  # Configure optimizers.
  def configure_optimizers(self):
    q_net_optimizer = self.hparams.optim(self.q_net.parameters(), lr=self.hparams.lr)
    return [q_net_optimizer]

  # Create dataloader.
  def train_dataloader(self):
    dataset = RLDataset(self.buffer, self.hparams.samples_per_epoch)
    dataloader = DataLoader(
        dataset=dataset,
        batch_size=self.hparams.batch_size
```

```
    )
    return dataloader

  # Training step.
  def training_step(self, batch, batch_idx):
    indices, weights, states, actions, rewards, dones, next_states = batch
    weights = weights.unsqueeze(1)
    actions = actions.unsqueeze(1)
    rewards = rewards.unsqueeze(1)
    dones = dones.unsqueeze(1)

    state_action_values = self.q_net(states).gather(1, actions)

    with torch.no_grad():
      _, next_actions = self.q_net(next_states).max(dim=1, keepdim=True)
      next_action_values = self.target_q_net(next_states).gather(1, next_actions)
      next_action_values[dones] = 0.0

    expected_state_action_values = rewards + self.hparams.gamma * next_action_values

    td_errors = (state_action_values - expected_state_action_values).abs().detach()

    for idx, e in zip(indices, td_errors):
      self.buffer.update(idx, e.item())

    loss = weights * self.hparams.loss_fn(state_action_values, expected_state_action_values, reduction='none')
    loss = loss.mean()

    self.log('episode/Q-Error', loss)
    return loss

  # Training epoch end.
  def on_train_epoch_end(self):
    alpha = max(
        self.hparams.a_end,
        self.hparams.a_start - self.current_epoch / self.hparams.a_last_episode
    )
    beta = min(
        self.hparams.b_end,
        self.hparams.b_start + self.current_epoch / self.hparams.b_last_episode
    )
    self.buffer.alpha = alpha
    self.buffer.beta = beta

    self.play_episode(policy=self.policy)
    self.log('episode/Return', self.env.unwrapped.game_state.score())

    if self.current_epoch % self.hparams.sync_rate == 0:
      self.target_q_net.load_state_dict(self.q_net.state_dict())
```

Purge logs and run the visualization tool (Tensorboard)

```
!rm -r /content/lightning_logs/
!rm -r /content/videos/
%load_ext tensorboard
%tensorboard --logdir /content/lightning_logs/
```

```
rm: cannot remove '/content/lightning_logs/': No such file or directory
rm: cannot remove '/content/videos/': No such file or directory
```



Train the policy

```
import pytorch_lightning as pl
import warnings
warnings.filterwarnings('ignore')
```

```
algo = DeepQLearning(
  'Catcher-v0',
  lr=0.0005,
  sigma=0.5,
  hidden_size=512,
  samples_per_epoch=1_000,
  a_last_episode=1_200,
  b_last_episode=1_200
)

trainer = pl.Trainer(
    accelerator="gpu" if num_gpus else "cpu", # Use 'gpu' if num_gpus is greater than 0, otherwise use 'cpu'
    devices=1, # Specify the number of GPUs or 'auto' for automatic detection
    max_epochs=1500,
    log_every_n_steps=1
)

trainer.fit(algo)
```

```
64
0 samples in experience buffer. Filling...
77 samples in experience buffer. Filling...
157 samples in experience buffer. Filling...
260 samples in experience buffer. Filling...
320 samples in experience buffer. Filling...
475 samples in experience buffer. Filling...
557 samples in experience buffer. Filling...
705 samples in experience buffer. Filling...
771 samples in experience buffer. Filling...
855 samples in experience buffer. Filling...
934 samples in experience buffer. Filling...
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type | Params | Mode
-----------------------------------------------
0 | q_net        | DQN  | 80.5 K | train
1 | target_q_net | DQN  | 80.5 K | train
```

Check the resulting policy

```
161 K     Total params
```

```
env = algo.env
policy = algo.policy
q_net = algo.q_net.cuda()
frames = []

for episode in range(10):
  done = False
  obs = env.reset()
  while not done:
    frames.append(env.render(mode='rgb_array'))
    action = policy(obs, q_net)
    obs, _, done, _ = env.step(action)
```

```
display_video(frames)
```

```
__
```