دانشکده مهندسی کامپیوتر

دکتر محمدرضا محمدی

بهار ۱۴۰۰

تمرین سری یازدهم

یادگیری عمیق

مجتبی نافذ 96431335

مهلت تحویل : ۱۷ خرداد ۱۴۰۰ ساعت ۲۳:۵۹:۵۹

---

۱. سیاست (policy) و ارزش (value) در یادگیری تقویتی را تعریف کنید. سپس روش‌های مبتنی بر سیاست و مبتنی بر ارزش را به طور خلاصه توضیح دهید و تفاوت آن‌ها را بیان کنید.

تعریف policy : در واقع π policy مغز و هسته ی agent ماست. و تابعی است که در با گرفتن یک state به عنوان خروجی یک اکشن را میدهد و agent ما براساس آن action ادامه کار را می دهد. و هدف ما پیدا کردن π_* بهینه است.

تعریف value :ارزشمندی و سودمندی هر state را value آن state می نامیم.
به صورت واضح تر،ارزش یک state برابر است با expected discounted return ای که agent در صورت شروع از start sate و عمل کردن براساس سیاست maximum valuable state به دست می آورد.

روش policy based methods :
در این روش ما مستقیما تابع policy که با π نشان دادیم را train می کنیم.
این تابع به عنوان ورودی state را می گیرد و به عنوان خروجی action را تولید می کند: $\pi(state) = action$
البته بسته به deterministic و stochastic بودن در ممکن است خروجی مجموعه ای از احتمالات باشد.
actions = [Right, Left, Jump, Stop]
Deterministic: $\pi(state) = Right$
Stochastic: $\pi(state) = [Right:0.5, Left=0.3, Jump=0.15, Stop=0.05]$
به عنوان نمونه: یک شبکه ی عصبی طراحی میکنیم که بین ۱۰ اکشن خروجی انتخاب کند. و آن را train میکنیم.

روش value based methods :
در این روش ما به جای تابع policy، تابع value را train می کنیم.
تابع value، تابعی است که هر state را به به expected value ی آن state تبدیل می کند.
که بالاتر مفهوم value را بیان کردم که منظور از expected value همان value ی بالاست که توضیح داده شد.
در نهایت در روش value based methods از سیاست maximum value استفاده می کند و در هر state آن اکشنی را انتخاب می کند که تابع value برای آن state بیشترین value را خروجی می دهد.
$V\pi(state) = expected\ value\ of\ state$

تفاوت policy based methods و value based method در توضیحات کامل بیان شد.

و تفاوت اصلی در آن بود که اولی بر اساس آموزش خود policy اکشن بعدی را انتخاب می کرد.

و دومی بر اساس آموزش value اکشن بعدی خود را انتخاب می کرد.

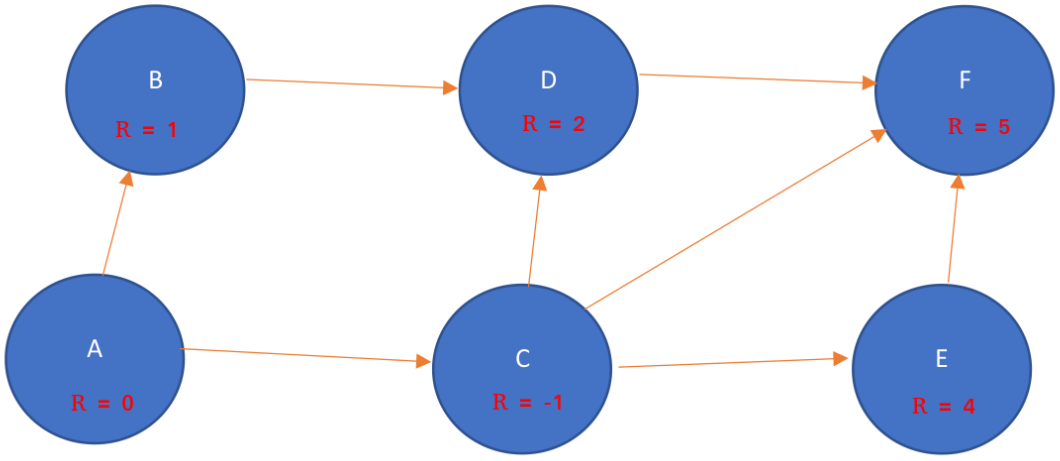
۲. گراف زیر چند state و actionهای مجاز بین آنها را نشان داده است (در یک اپیزود گذشتن از هر state تنها یک بار مجاز است). پاداش‌های دریافتی از stateهای A، B، C، D، E و F (که state F پایانی (Terminal) است) به ترتیب عبارتند از: 0، 1، 1-، 2، 4 و 5 (پارامتر گاما را ۱ در نظر بگیرید)

الف) فرض کنید اگر بر اساس سیاست ۱ پیش برویم، مسیر ACEF انتخاب شود. بازده (return) این مسیر را به دست آورید.

ب) فرض کنید اگر بر اساس سیاست ۲ پیش برویم، مسیر ABDF انتخاب شود. بازده (return) این مسیر را به دست آورید.

ج) بازده کدام یک از سیاست‌های ذکر شده در الف یا ب بهتر بود؟ اگر سیاست با بازده بهتری برای انتخاب مسیر وجود دارد آن را بیان کنید و بازده آن را محاسبه کنید.

د) موارد الف تا ج را با در نظر گرفتن پارامتر گاما برابر با ۰.۵ محاسبه کنید.



gama=1

الف)

$$R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = -1 + 4 + 5 = 8$$

ب)

$$R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = 1 + 2 + 5 = 8$$

ج) بازده هر دو برابر است و تفاوتی ندارد و در این حالت سیاست بهتری وجود ندارد و سیاست های بهینه همین دو مورد هستند.

د)سیاست ۱ :

$$R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = -1 + 0.5*4 + 0.25*5 = 2.25$$

سیاست ۲ :

$$R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = 1 + 0.5*2 + 0.25*5 = 3.25$$

سیاست ۲ بازده بیشتری دارد. و در بین این دو بهتر است.

در کل هم سیاست ۲ بهترین است و سیاست بهتری نداریم.

۳. در این تمرین قصد این است که یک شبکه Siamese برای یافتن embeddingهای تفکیک‌کننده برای اعداد موجود در مجموعه داده MNIST طراحی شود. در نوتبوک تهیه‌شده پیاده‌سازی‌های لازم را انجام دهید و خروجی خود را که embeddingها را با استفاده از PCA در فضای دو بعدی نمایش می‌دهد، تحلیل کنید (از PCA استفاده می‌شود تا ابعاد بردار ورودی را به ۲ کاهش دهیم تا در صفحه قابل نمایش باشد و این کار را هم برای embeddingهای استخراج شده و هم برای خود تصاویر خام انجام می‌دهیم). همچنین، در رابطه با محتوای هر cell که در ابتدای آن عبارت `# describe this cell` نوشته شده است هم به طور دقیق توضیح دهید.

بلاک :

```
def create_batch(batch_size=256):
```

یک داده را به عنوان anchor انتخاب می کند

و یک batch ، از ۲۵۶ داده را به عنوان positive و هم label با batch را انتخاب می کند

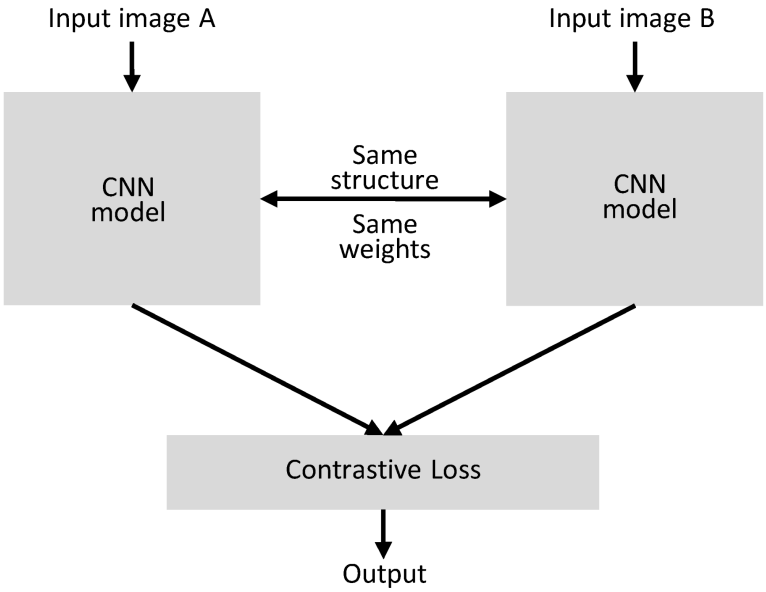
و یک batch ، از ۲۵۶ داده را به عنوان negative و مخالف label با batch را انتخاب می کند(از ۹ عدد دیگر)

بلاک:

```
Implement Siamese network

input_anchor = tf.keras.layers.Input(shape=(28, 28, 1), name="anchor")
Input    ……
```

سعی شد که شبکه ی siamese شبیه زیر را پیاده سازی کنم که وزن ها مشترک می باشند:



در واقع برای ما سه شبکه ی dense را که با هم concatenate می کند.

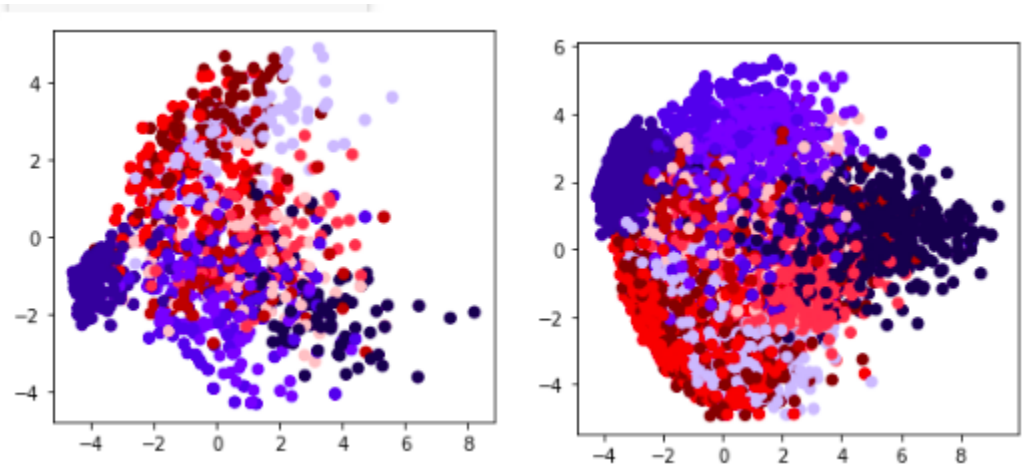
بلاک:

```
class PCAPlotter(tf.keras.callbacks.Callback):
      …….
```

یک customize callback function را پیاده سازی می کند.

از PCA استفاده می کنیم تا ابعاد بردار ورودی را به ۲ کاهش داده تا در صفحه قابل نمایش باشد و این کار را هم برای embedding های استخراج شده و هم برای خود تصاویر خام انجام میدهیم

تحلیل قسمت آخر:



در این قسمت می بینیم که **embedding** هایی که در کلاس یکسان بوده اند و همرنگ هستند با هم کاملا نزدیک به هم می باشند و این نشانگر قدرت شبکه است

و همچنین می بینیم که **embedding** های کلاس **negetive** هم به هم دور اند

و این قدرت آموزش سه نوع داده با وزن شبکه ی مشترک است.

ر