



باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

علوم اعصاب: یادگیری، حافظه، شناخت

تمرین سری پنجم: شبکه‌های عصبی

موعد تحویل: جمعه ۳۰ فروردین ۱۳۹۸، ساعت ۲۳:۵۵

نحوه تحویل: فایل‌های مربوط به تمرین (گزارش و کد تمرین) را به صورت یک فایل با فرمت zip/.rar در محل مشخص شده در سامانه CW آپلود کنید.

نکات قابل توجه در این تمرین:

- کد تمرین را در قالب یک فایل با فرمت ipynb تحویل دهید. سعی کنید تا حد امکان مرتب و تمیز کد بنویسید و در موارد لازم، توضیحات تکمیلی را در کنار کد ضمیمه کنید.
- برای این تمرین، لازم است یک گزارش جداگانه با فرمت pdf. نیز تحویل دهید. گزارش شما باید حاوی پاسخ به پرسش‌های تئوری، تحلیل‌ها و نتیجه‌گیری‌ها، نمودارها و خروجی‌های نهائی مسائل، توضیحات لازم در مورد روال حل مسئله (که با تشخیص خودتان در موارد لازم ارائه می‌دهید)، و هر مورد دیگری که لازم به ذکر می‌دانید، باشد.
- گزارش و کد تمرین را در قالب یک فایل فشرده با فرمت zip/.rar. ذخیره کرده و از طریق سامانه CW تحویل دهید.
- مطابق با توضیحاتی که در فایل مشخصات درس در ابتدای ترم ارائه شد، برخی تمرین‌های کامپیوتری به دلیل اهمیت پایه‌ای و بالایی که دارند، بهتر است به صورت فردی (و نه گروهی) انجام شوند. این تمرین نیز یکی از این دسته تمرین‌هاست، بنابراین تحویل این تمرین به صورت انفرادی 10% نمره امتیازی دارد. (یعنی نمره شما در ضریب 1.1 ضرب می‌شود).
- محتوای این تمرین در راستای آشنایی و تسلط اولیه بر چگونگی کار با شبکه‌های عصبی مصنوعی و کتابخانه‌ی pytorch می‌باشد. نکات ابتدائی مربوط به این مباحث در دومین کارگاه پایتون درس بیان شده‌اند.
- دقت کنید که با توجه به مقدماتی بودن این تمرین و هدف آن (که آشنایی اولیه است، و نه پیاده‌سازی یک فرآیند پیشرفته)، نام این مجموعه سؤالات نیز «تمرین» در نظر گرفته شده است و نه «پروژه» (و به تبع آن انجام انفرادی تمرین نیز نمره‌ی امتیازی دارد، که در مورد پروژه‌ها چنین نخواهد بود).
- با این حال در نظر داشته باشید که تمرین‌های مختلف درس نمرات یکسانی ندارند، و سهم هر تمرین از مجموع کل نمرات متناسب با گستردگی و حجم تمرین خواهد بود؛ و این تمرین به وضوح نمره‌ی بیشتری از تمرین پیشین خواهد داشت.
- دقت کنید که موعد تحویل این تمرین از زمانی که صورت آن در اختیار شما قرار می‌گیرد فاصله‌ی زیادی دارد (حدود ۲۵ روز). علت این امر آن است که اولاً تمرین سری چهارم درس (که البته حجم کم و سبکی دارد) نیز به موازات این تمرین در اختیار شما قرار گرفته است، ثانیاً بتوانید بعد از تعطیلات نوروز به رفع اشکال حضوری بپردازید (و در صورت نیاز شما، یک جلسه‌ی رفع اشکال عمومی برگزار شود) و ثالثاً در صورتی که به هر دلیلی نمی‌توانید در تعطیلات عید زمان کافی صرف کنید، دچار مشکل نشوید. با این حال در نظر داشته باشید که علاوه بر این دو تمرین، ارائه‌های درس و آزمون تئوری اول درس نیز پس از تعطیلات وجود خواهند داشت، لذا در مدیریت زمان خود دقت لازم را داشته باشید.
- در پایان لازم به ذکر است که رعایت شرافت دانشجویی، دارای ارزشی به مراتب بالاتر از تعلقات دنیوی نظیر نمره است؛ از این رو به کسانی که در حل تمرین شرافت دانشجویی خود را - با اعمالی نظیر رونویسی تمرین - زیر پا می‌گذارند، نمره‌ای تعلق نخواهد گرفت.

۱ مقدمات؛ AND و XOR

هدف از این سؤال آشنایی مقدماتی با pytorch و شبکه‌های عصبی مصنوعی است که در آن می‌خواهیم کار با ساختارهای feed-forward را تمرین کنیم.

۱. در ابتدا از نصب و کارکرد درست pytorch بر روی سیستم خود اطمینان حاصل کنید. همچنین برای آشنایی به چگونگی عملکرد این کتابخانه (و مرور مفاهیمی که در دومین کارگاه عملی درس بیان شد) خوب است محتوای دو لینک زیر (و مشخصاً لینک دوم) را بررسی کنید:

- <https://pytorch.org/tutorials/>
- https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

۲. حال جدول صحت را برای دو مسأله‌ی منطقی AND و XOR تشکیل دهید. برای دستیابی به نتایج مطلوب به جای 0 و 1 از -1 و 1 استفاده کنید. بدین ترتیب شما برای هر مسأله ۴ سطر خواهید داشت که در هر یک، دو ورودی x_1 و x_2 و یک خروجی y وجود دارد.

۳. در این مرحله می‌خواهیم مسأله‌ی AND را با شبکه‌های عصبی مصنوعی حل کنیم. برای این کار با استفاده از pytorch یک شبکه‌ی عصبی یک‌لایه طراحی کنید که دو ورودی دریافت کرده و در خروجی، یک عنصر را تحویل می‌دهد. با استفاده از جدول صحت و تعداد epochهای متنوع، شبکه را آموزش دهید. برای این کار لازم است یک دیتاست آماده را به ترتیب به شبکه وارد کنید، به گونه‌ای که شبکه هر زوج ورودی را با خروجی متناظر خود دیده و تغییرات لازم برای آموزش را در خود نگه‌داری و در زمان مناسب اعمال کند. برای آن که شبکه بتواند به درستی آموزش ببیند، یک عنصر 1 به ورودی‌ها اضافه کنید:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

۴. توسط نمودار نشان دهید که چگونه loss در طول فرآیند آموزش کاهش پیدا می‌کند. تغییرات تابع هزینه در فرآیند آموزش را با تغییرات متغیرهای زیر رسم کرده و اثرات آن‌ها را بر روی فرآیند آموزش بررسی کنید:

- learning rate
- batch size
- number of epochs

در این جا باید چند نمودار بکشید که روند کاهش loss را بر حسب زمان آموزش نمایش دهد و برای چند مقدار مختلف از هر یک از متغیرهای فوق، این نمودارها را کشیده و با یکدیگر مقایسه کنید.

۵. فعالیت نورون‌ها را در شبکه رسم نمایید. برای این کار باید با استفاده از یک تصویر، ضرایب هر نورون را به صورت رنگی نمایش دهید که هر سطر نشان‌گر یک نورون است.

۶. حال قسمت ۳ را برای مسأله‌ی XOR پیاده‌سازی کنید. برای این مرحله لازم است از تعداد لایه‌های بیشتری در شبکه‌ی عصبی خود بهره ببرید. علت این امر (یعنی نیاز به لایه‌های بیشتر) را در گزارش توضیح دهید.

۷. در این گام، تعداد ورودی‌های مسأله‌ی XOR را به ۴ ورودی افزایش دهید. با دو ساختار متفاوت:

- زیاد کردن نورون‌های لایه‌ی میانی
- زیاد کردن تعداد لایه‌ها

شبکه‌ای بسازید که بتواند نتیجه را به درستی تعیین کند.

۸. در کدام حالت تعداد نورون کمتری استفاده شده است؟ دو حالت فوق را مقایسه کنید.

۲ Vanilla RNN and Parity Bit Problem

در این مسأله می‌خواهیم یک شبکه‌ی بازگشتی ساده (Vanilla RNN) را پیاده‌سازی کنیم و از آن در مسأله‌ی تشخیص بیت زوجیت^۱ استفاده کنیم. روابط یک عبور مستقیم^۲ در یک RNN ساده به صورت زیر است:

$$s_t = f(Ux_t + b_u + Ws_{t-1} + b_w)$$

$$o_t = f'(Vs_t + b_v)$$

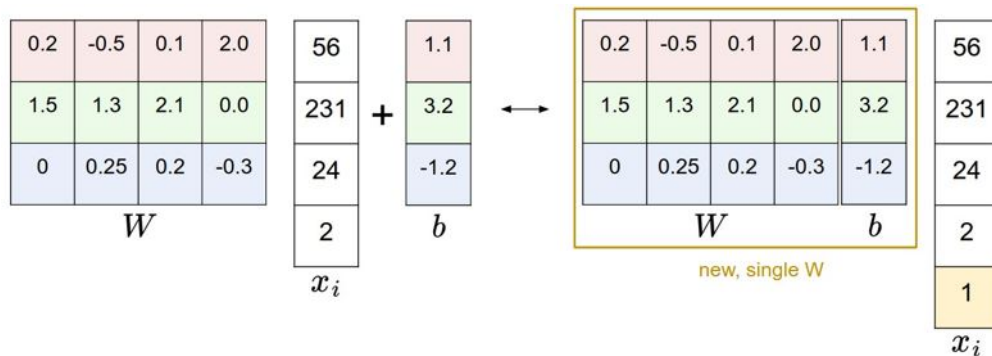
که در آن:

- s_t : Hidden state of RNN at time step t
- x_t : Input at time step t
- U : Learnable weight matrix for connections between the hidden state and input
- W : Learnable weight matrix for connections between the hidden states
- V : Learnable weight matrix for connections between the hidden state and output
- o_t : Output at time step t
- f and f' : Nonlinearity functions such as tanh, ReLU, & Softmax
- b_U, b_W, b_V : Bias terms

در صورتی که مقدار ضریب بایاس را در سیگنال ورودی ادغام کنیم (مطابق شکل ۱)، می‌توان روابط فوق را به صورت ساده‌تری بازنویسی کرد:

$$s_t = f(Ux_t + Ws_{t-1})$$

$$o_t = f'(Vs_t)$$



شکل ۱

۱. در ابتدا ۱۰۰۰ رشته‌ی دودویی تصادفی به طول ۱۰ تولید کنید. از این داده‌ها ۸۰۰ رشته را برای آموزش در نظر بگیرید و ۲۰۰ رشته‌ی دیگر را برای تست قرار دهید.
۲. حال یک شبکه‌ی عصبی بازگشتی طراحی کنید که در ورودی یک بیت را گرفته و مقدار بیت زوجیت دنباله‌ی دریافتی را از ابتدا تا آن لحظه محاسبه کند. از رشته‌های تولیدشده در گام ۱ برای آموزش این شبکه استفاده کنید.
۳. بررسی کنید که آیا این شبکه برای یک رشته‌ی بلند (مثلاً به طول ۱۰۰ عنصر) همچنان کار می‌کند یا خیر. این مسأله را تحلیل کنید.

^۱parity bit

^۲forward pass

۴. توسط نمودار نشان دهید که چگونه loss در طول فرآیند آموزش کاهش پیدا می‌کند. تغییرات تابع هزینه در فرآیند آموزش را با تغییرات متغیرهای زیر رسم کرده و اثرات آن‌ها را بر روی فرآیند آموزش بررسی کنید:

- learning rate
- batch size
- number of epochs
- hidden layer size

در این جا باید چند نمودار بکشید که روند کاهش loss را بر حسب زمان آموزش نمایش دهد و برای چند مقدار مختلف از هر یک از متغیرهای فوق، این نمودارها را کشیده و با یکدیگر مقایسه کنید.

۵. فعالیت نورون‌ها را در شبکه رسم نمایید. برای این کار باید با استفاده از یک تصویر، ضرایب هر نورون را به صورت رنگی نمایش دهید که هر سطر نشان‌گر یک نورون است.

۳ استفاده از شبکه‌های عصبی بازگشتی به منظور رفع نویز

۱. سه سیگنال سینوسی، مثلثی، و دندان‌اره‌ای با ۱۰ دوره‌ی تناوب تولید کنید. از این مجموعه دو دوره‌ی تناوب آخر را برای تست کنار بگذارید و ۸ تناوب اول را برای آموزش شبکه استفاده کنید.

۲. یک نویز گاوسی با میانگین صفر و واریانس 0.1 تولید کنید. از این پس این سیگنال را noise می‌نامیم.

۳. از سیگنال noise انتگرال بگیرید و آن را Brownian بنامید.

۴. سیگنال‌های اصلی را با noise و Brownian جمع کنید.

۵. یک شبکه‌ی عصبی بازگشتی طراحی کنید که سیگنال آلوده به نویز را رفع نویز کند؛ یعنی مقادیر سیگنال نویزی را به عنوان ورودی به شبکه بدهید و به ازای هر نقطه از سیگنال، مقدار دقیق و بدون نویز آن را نیز به عنوان تارگت به شبکه وارد کنید. برای این کار ابتدا سیگنال‌های جمع شده با noise را به شبکه بدهید، و سپس سیگنال‌های جمع شده با Brownian را وارد شبکه کنید.

نکته: برای آن که شبکه بهتر آموزش ببیند، می‌توانید نویزهای متفاوت تولید کنید و داده‌های آموزش خود را افزایش دهید.

۶. اگر خود سیگنال Brownian را به شبکه بدهیم، خروجی چگونه خواهد بود؟

۷. تحلیل کنید که حساسیت این ابزار به آستانه‌ی نویز چگونه است. برای این مسأله فقط از سیگنال noise استفاده کنید و نیازی به تحلیل نسبت به نویز Brownian نیست. برای این کار مقدار واریانس نویز را افزایش دهید و SNR خروجی نهایی را محاسبه کنید.

۸. توسط نمودار نشان دهید که چگونه loss در طول فرآیند آموزش کاهش پیدا می‌کند. تغییرات تابع هزینه در فرآیند آموزش را با تغییرات متغیرهای زیر رسم کرده و اثرات آن‌ها را بر روی فرآیند آموزش بررسی کنید:

- learning rate
- batch size
- number of epochs
- hidden layer size

در این جا باید چند نمودار بکشید که روند کاهش loss را بر حسب زمان آموزش نمایش دهد و برای چند مقدار مختلف از هر یک از متغیرهای فوق، این نمودارها را کشیده و با یکدیگر مقایسه کنید.

۹. فعالیت نورون‌ها را در شبکه رسم نمایید. برای این کار باید با استفاده از یک تصویر، ضرایب هر نورون را به صورت رنگی نمایش دهید که هر سطر نشان‌گر یک نورون است.

۴ رفع نویز در فضای دوبعدی

در این مسأله می‌خواهیم رفتار یک شبکه‌ی عصبی بازگشتی را در فضای دوبعدی مشاهده کنیم.
۱. توسط قطعه کد زیر، یک خم مارپیچ را در فضای دوبعدی تعریف کنید:

```
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(1000)
u = 0.3
r0 = 10

r = r0 + u * t
omega = 2 * np.pi * 0.01
phi0 = np.pi
phi = -omega * t + phi0

x = r * np.cos(phi)
y = r * np.sin(phi)

plt.plot(x, y)
plt.show()
```

از قطعه کد بالا یک بار ۲۰۰ نقطه‌ی آخر را برای تست در نظر بگیرید و بار دیگر ۲۰۰ نقطه‌ی ابتدایی را به عنوان داده‌ی تست لحاظ کنید. در پایان این دو حالت را با یکدیگر مقایسه کنید.

۲. حال یک نویز گوسی با میانگین صفر و واریانس 0.1 تولید کنید و با سیگنال اصلی جمع کنید.

۳. مشابه مسأله‌ی قبلی، هر نقطه از خم نویزی را به ورودی یک شبکه‌ی عصبی بازگشتی بدهید و به عنوان خروجی، خم اصلی را باز پس بگیرید.

۴. خم ورودی و خم خروجی تولیدشده را در یک شکل نمایش دهید و مقدار SNR خروجی را گزارش کنید.

۵. توسط نمودار نشان دهید که چگونه loss در طول فرآیند آموزش کاهش پیدا می‌کند. تغییرات تابع هزینه در فرآیند آموزش را با تغییرات متغیرهای زیر رسم کرده و اثرات آن‌ها را بر روی فرآیند آموزش بررسی کنید:

- learning rate
- batch size
- number of epochs
- hidden layer size

در این جا باید چند نمودار بکشید که روند کاهش loss را بر حسب زمان آموزش نمایش دهد و برای چند مقدار مختلف از هر یک از متغیرهای فوق، این نمودارها را کشیده و با یکدیگر مقایسه کنید.

۶. فعالیت نورون‌ها را در شبکه رسم نمایید. برای این کار باید با استفاده از یک تصویر، ضرایب هر نورون را به صورت رنگی نمایش دهید که هر سطر نشان‌گر یک نورون است.