به نام خدا

تمرین سری اول درس رباتیک اجتماعی و شناختی

(شبکه های عصبی مصنوعی)

مدرس: علیرضا طاهری

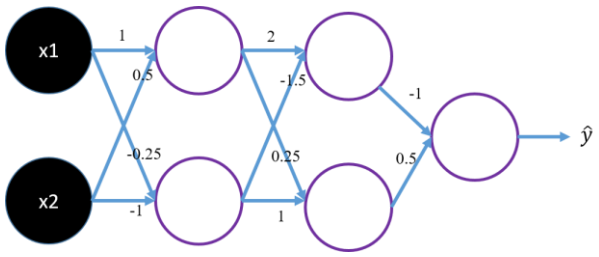
دانشکده مهندسی مکانیک، دانشگاه صنعتی شریف

۱- برای انجام یک طبقه بندی دو کلاسه با داده های آموزش داده شده، می خواهیم از شبکه عصبی ساده سه لایه زیر استفاده کنیم. با درنظر گرفتن تابع فعال سازی tanh برای نرون های لایه های (مخفی) اول و دوم و تابع فعال سازی sigmoid برای نرون لایه سوم (خروجی) و همچنین تابع هزینه میانگین مربعات خطا (MSE)، مطلوبست:

الف) محاسبه عبارت های $\frac{\partial Loss}{\partial w_{11}^{[3]}}$ ، $\frac{\partial Loss}{\partial w_{11}^{[2]}}$ و $\frac{\partial Loss}{\partial b_{1}^{[1]}}$ به صورت پارامتری.

ب) با وزن های تصادفی داده شده و با نرخ آموزش $\eta = 0.01$، الگوریتم انتشار به عقب (BP) را به میزان یک دوره (epoch) اجرا کرده و تمامی وزن ها (و بایاس ها) را به روزرسانی کنید (مقادیر تمامی بایاس ها در حالت اول صفر فرض شده است).

Sample #	x1	x2	y
1	1	0	1
2	-1	0	1
3	0	1	1
4	0	-1	1
5	2	0	0
6	-2	0	0
7	0	2	0
8	0	-2	0



۲- به صورت کیفی به پرسش های زیر پاسخ دهید:

الف) اگر به جای توابع فعال سازی غیر خطی معرفی شده، برای تمامی نرون های یک شبکه MLP از تابع همانی (یا هر تابع خطی دیگری) استفاده کنیم، خروجی نهایی بر حسب ورودی چگونه خواهد بود؟ این موضوع، چه مشکلی را به وجود می آورد؟

ب) در آموزش شبکه های عصبی به هیچ عنوان مناسب نیست که وزن های تصادفی اولیه همگی یکسان (مثلا صفر) باشند. به نظر شما دلیل این امر چیست؟

پ) اگرچه الگوریتم انتشار به عقب به صورت سیستماتیک برای هر تعداد لایه در شبکه های MLP قابل اجراست، با در نظرگرفتن توابع سیگموییید یا تانژانت هذلولوی به عنوان توابع فعال سازی نرون ها، چرا ما تاکنون در عمل، (عمدتاً) تعداد لایه های مخفی را محدود به یک یا دو در نظر می گیریم؟

۳- یک شبکه عصبی چندلایه پیشنهاد دهید که با آن بتوانیم تابع $y = 2\sin(x) + \sin(2x)$ را بازتولید کنیم (ورودی شبکه: اسکالر x؛ و خروجی شبکه: اسکالر y). ساختار و وزن های شبکه و خروجی آن را برای بازه $x\epsilon[0,3\pi]$ ارائه نمایید. در نظر داشته باشید که داده های آموزش فقط از مجموعه $x\epsilon\{0,\frac{\pi}{20},\frac{2\pi}{20},\frac{3\pi}{20},\dots,2\pi\}$ (و y های مربوطه به عنوان برچسب) به عنوان ورودی به شبکه داده شوند. آیا شبکه شما در درون یابی و برون یابی تابع مورد نظر، قدرت های یکسانی دارد؟ (اگرچه مجازید از هر بستر/بسته نرم افزاری برای آموزش شبکه خود استفاده کنید، اما پیشنهاد می کنم تلاش کنید که خودتان کدهای مربوط به الگوریتم انتشار به عقب را نوشته و اجرا کنید.)

۴- یکی از قابلیت های مورد انتظار از یک ربات اجتماعی، توانمندی تشخیص حالت چهره افراد (نظیر شادی، غم، خشم و ...) می باشد. در این مسئله، مجموعه عکس هایی مستخرج از پایگاه داده CK+ به عنوان داده های آموزش در اختیار شما قرار می گیرد. شما می بایست یک شبکه عصبی چندلایه طراحی کنید که تا حد امکان، طبقه بندی ۸ کلاسه شناسایی حالت های چهره کاربران را به خوبی انجام دهد. عکس ها در ابعاد ۴۹۰×۶۴۰ پیکسل (عمدتاً در محدوده طیف خاکستری) می باشند و در هشت پوشه مختلف قرار داده شده اند؛ به عبارتی، نام پوشه ها، معادل برچسب عکس موجود در آن ها است. ما در نهایت کدهای شما را روی مجموعه دادگانی به عنوان داده های تست (که هم اکنون در اختیار شما نیست) نیز پیاده سازی خواهیم کرد تا میزان توانمندی شبکه پیشنهادی شما را در طبقه بندی داده های جدید محک بزنیم.

لینک دانلود داده های آموزش:

https://drive.google.com/file/d/1XL0W5YdJjdE4WVfo9YHjddH4V6CJ6H10/view?usp=sharing

توضیح تکمیلی ۱: در صورتی که عکس رنگی در مجموعه دادگان وجود داشت، با دستورات نمونه زیر از کتابخانه opencv می توانید آن را به طیف خاکستری تبدیل نمایید:

```
import cv2
image= cv2.imread("picture.jpg",0)
```

توضیح تکمیلی ۲: با استفاده از مجموعه دستورات نمونه زیر می توان با مشخص کردن مسیر پوشه مورد نظر، تمامی تصاویر موجود در آن پوشه با فرمت مشخص را خواند و در یک آرایه ذخیره نمود:

```
import cv2
import glob

image_dir = "./image/"
cv_img = []
for img in glob.glob(image_dir+"*.jpg"):
    image= cv2.imread(img)
    cv_img.append(image)
```