



بسم الله الرحمن الرحيم

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

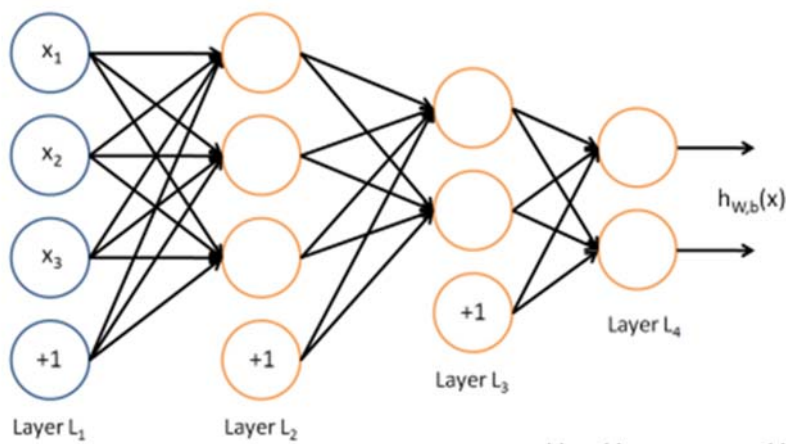
تمرین نهایی - پاییز 1396



پیاده سازی یک شبکه‌ی عصبی مصنوعی MLP برای طبقه بندی مجموعه داده MNIST

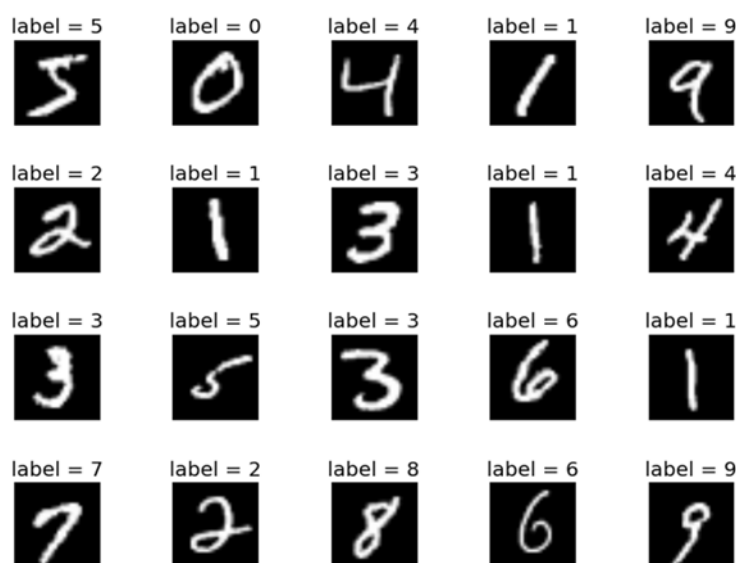
شرح تمرین

در دو تمرین قبل با ساختار یک مدل ساده از یک نرون مصنوعی آشنا شدید. در این تمرین که به عنوان پروژه پایانی به شما داده می-شود هدف پیاده سازی یک شبکه مصنوعی به نام MLP است که ساختار کلی آن مطابق شکل 1 است. شبکه ی MLP تشکیل شده از سه قسمت لایه ی ورودی، لایه ی پنهان و لایه ی خروجی است. در شکل 1 L1 به عنوان لایه ی ورودی، L2 و L3 به عنوان لایه ی پنهان و L4 به عنوان لایه ی خروجی تعریف می شود.

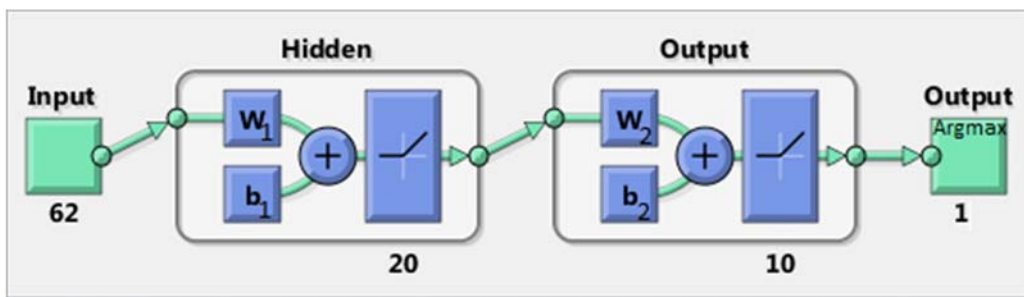


شکل 1 ساختار کلی یک شبکه ی مصنوعی MLP

از شبکه های عصبی مصنوعی به عنوان کاربرهای مختلف از جمله طبقه بندی، حافظه سپاری، خوشه بندی و ... استفاده می شود. در این تمرین هدف پیاده سازی یک شبکه ی MLP جهت طبقه بندی یک سری داده تشکیل شده از عکس های دست نوشته ی اعداد بین 0 تا 9 می باشد. این عکس ها از مجموعه داده ی MNIST که چند نمونه ی آن در شکل 2 نشان داده شده است انتخاب شده است. عکس های مجموعه داده ی MNIST ابعاد 28×28 پیکسل دارد. جهت سادگی و کوچک شدن ابعاد شبکه ی مصنوعی مورد نظر این عکس ها که شامل 786 پیکسل می باشد را با استفاده از الگوریتم های feature conditioning که در مسائل الگوییابی استفاده می شود به 62 ویژگی کاهش داده شده است. این مجموعه همانطور که مشخص است دارای 10 کلاس متناظر با اعداد 0 تا 9 است. لذا شبکه مصنوعی مورد نظر طراحی شده در لایه ی ورودی دارای 62 نود ورودی و در لایه خروجی دارای 10 نرون خروجی است. شبکه طراحی شده برای این مسئله دارای یک لایه ی پنهان با 20 نرون در نظر گرفته شده است. توابع فعال ساز در هر دو لایه تابع Relu در نظر گرفته شده است. تابع Relu در صورتی که ورودی کمتر از صفر باشد خروجی را صفر در غیر این صورت خود ورودی را در خروجی عبور می دهد. در نتیجه شبکه ی MLP مورد نظر مطابق شکل 3 می باشد.



شکل 2 چند نمونه از مجموعه داده MNIST



شکل 3 شبکه عصبی مصنوعی MLP مورد نیاز در این پروژه

در خروجی برای اینکه برچسب عکس ورودی مشخص شود، باید بین خروجی 10 نرون بیشترین مقدار انتخاب شود، در صورتی که نرون i بیشترین مقدار خروجی را داشته باشد، برچسب عکس ورودی، عدد i خواهد بود.

برای پیاده سازی این شبکه نیاز به وزن های مربوطه به لایه ی اول (w_1) و دوم (w_2) و مقادیر بایاس لایه ی اول (b_1) و دوم (b_2) می-باشد. w_1 و w_2 به ترتیب یک ماتریس با ابعاد 62 در 20 و 20 در 10 است که مجموعاً شامل 1404 وزن می-شود. B_1 و b_2 به ترتیب یک ماتریس با ابعاد 20 در یک و 10 در یک است که مجموعاً شامل 30 مقدار بایاس خواهد شد.

نکات پیاده سازی

- ✓ تمام محاسبات به صورت **fixed point** علامت دار و با عرض دیتای 16 بیت که 6 بیت برای قسمت اعشار و 10 بیت برای قسمت صحیح انجام شود. همچنین برای ورودی ها و مقادیر وزن و بایاس این مشخصات را در نظر بگیرید.
- ✓ مقادیر وزن ها و بایاس ها را مطابق طراحی خود در حافظه در نظر بگیرید.
- ✓ ورودی را می توانید فرض کنید در حافظه قرار دارد یا به صورت سریال مطابق تمرین اول دریافت شود.
- ✓ در انتخاب نحوه ی استفاده از حافظه و تعداد نرون استفاده شده آزاد هستید ولی طراحی شما نباید غیر منطقی باشد.
- ✓ مقادیر وزن ها و بایاس ها در فایل های جداگانه در سایت بارگذاری شده است.
- ✓ کد متلب برای مدل کردن شبکه ی مورد نظر با استفاده از محاسبات **fixed point** نیز در سایت بارگذاری شده است.
- ✓ مآژول های جمع کننده و ضرب کننده ی محاسبات **fixed point** در سات بارگذاری شده است. گروه هایی که تمرین قبلی را نتوانسته اند انجام بدهند می توانند از این مآژول ها استفاده کنند.
- ✓ 100 داده ورودی به عنوان تست نیز در سایت بارگذاری شده است. داده با ابعاد واقعی برای مشاهده ی عکس ها در صورت نیاز نیز قرار داده شده است.

نکات جانبی

این پروژه باید در گروه های دو نفری انجام گیرد. هر دو عضو گروه باید در پیاده سازی شرکت داشته باشند. درصد نمره هر فرد بستگی به میزان مشارکت هر کدام یک از اعضای گروه خواهد داشت.

گزارش کاملی از نحوه ی طراحی و نکات پیاده سازی به همراه بلاک دیاگرام طراحی خود و همچنین نتایج سنتز و Place&Route به صورت مرتب به همراه کدهای خود ارسال نمایید.

هنگام تحویل پروژه هر دو اعضای گروه باید حضور داشته باشند.

شبیه سازی در ابزار modelsim قسمت اجباری پروژه است.

گروه هایی که بتوانند بر روی برد جواب بگیرند تا 25 درصد نمره ی امتیازی می گیرند

موفق باشید