



آشنایی با اهداف کلی پروژه

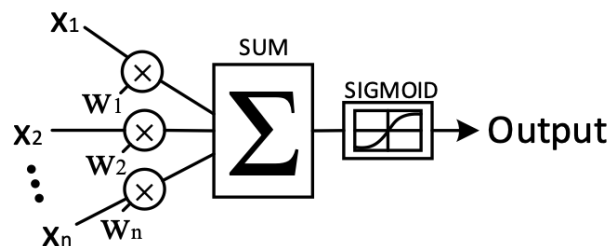
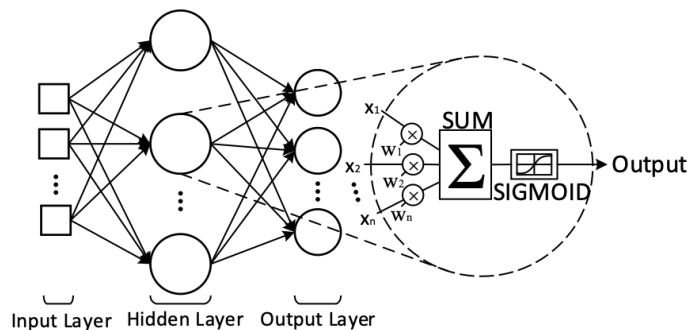
انواع زیادی از مدل‌های محاسباتی تحت عنوان کلی شبکه‌های عصبی معرفی شده‌اند که هر کدام از بخشی از قابلیت‌های سیستم عصبی الهام گرفته و برای دسته‌ای از کاربردها قابل استفاده هستند. یکی از پرکاربردترین این مدل‌ها برای سیستم‌های نهفته هوشمند (مثلاً در یک تشخیص‌دهنده تصویر) شبکه‌های عصبی پرسپترونی چند لایه (MLP) است که از یک لایه ورودی، n لایه میانی (و یا پنهان) و یک لایه خروجی تشکیل شده است. در بیشتر حالت‌های پیاده‌سازی MLPها در حل مسائل کوچک، این مدل‌ها دارای یک لایه ورودی، یک لایه میانی و یک لایه خروجی هستند. MLPها یکی از مهمترین رده‌های خانواده شبکه‌های عصبی پیش‌رسان^۱ به حساب می‌آیند. در معماری پیش‌رسان، جریان داده فقط در یک جهت و از لایه ورودی به سمت لایه‌های پنهان و از آنجا به لایه خروجی حرکت می‌نماید. به بیان دیگر، در این ساختار، ورودی نورون‌های لایه i ، خروجی نورون‌های لایه $i-1$ هستند (به غیر از اولین لایه ای که به ورودی‌های شبکه عصبی متصل است). هر نورون در لایه‌های میانی و خروجی که در شکل (۱) نشان داده شده است، خروجی تمام نورون‌های لایه قبل را گرفته و مجموع حاصل ضرب ورودی‌ها در وزن مخصوص به هر ورودی را به صورت رابطه‌ی (۱) محاسبه می‌کند.

$$y = \sum_j w_j x_j \quad \text{رابطه (۱)}$$

به بیان دیگر جواب نهایی از ضرب نقطه‌ای آرایه z عنصری وزن (w) و آرایه z عنصری ورودی (x) در یکدیگر به دست می‌آید. در نهایت هم یک تابع فعال‌سازی (2AF) جهت تولید نتیجه خروجی بر روی حاصل جمع نهایی (y) اعمال می‌شود. سپس حاصل، به تمام نورون‌های لایه بعد ارسال می‌گردد.

^۱ feed forward

^۲ Activation Function



شکل ۱. ساختار یک شبکه عصبی ۳ لایه و محاسبات داخلی هر نورون

مجموعه داده^۳ MNIST یک مجموعه داده نسبتاً بزرگ از ارقام دست نوشته است که معمولاً برای آموزش انواع شبکه‌های عصبی کاملاً متصل^۴ مورد استفاده قرار می‌گیرد. این مجموعه شامل تصاویر سیاه و سفید در ابعاد 28×28 است که از ۶۰۰۰ تصویر برای آموزش شبکه و ۱۰۰۰ تصویر برای تست شبکه تشکیل شده است. شکل (۲) تعدادی از تصاویر مورد استفاده در تست شبکه عصبی را نمایش می‌دهد.

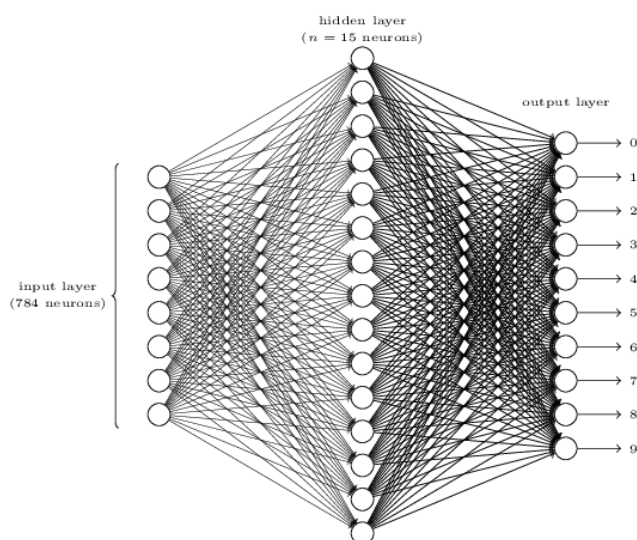


شکل ۲. تصاویر نمونه از مجموعه داده تست MNIST

^۳ dataset

^۴ Fully Connected

یک مدل شبکه‌ی عصبی مناسب برای این مجموعه داده در شکل (۳) نمایش داده شده است. همانطور که در شکل (۳) مشخص است، در لایه‌ی ورودی ۷۶۴ ورودی (28×28 پیکسل تصویر) وارد شبکه‌ی عصبی می‌شوند که هر ورودی به همه‌ی نورون‌های لایه‌ی میانی، با وزن‌های مختص به خود، متصل است. سپس، خروجی‌های نورون‌های لایه‌ی میانی به ۱۰ خروجی لایه‌ی خروجی، که نماد هر یک از اعداد ۰ تا ۹ هستند، متصل می‌شود.



شکل ۳. داخل یک شبکه‌ی عصبی طراحی شده برای MNIST

در این پروژه قصد داریم شبکه‌ی عصبی برای آزمایش داده‌های تست مجموعه MNIST پیاده‌سازی کنیم و میزان دقت شبکه مورد نظر را به دست آوریم. برای این منظور، از نخ‌های به عنوان واحدهای عملیاتی برای محاسبه خروجی نورون‌های لایه‌های مختلف استفاده خواهیم کرد.

ایجاد شبکه عصبی

شبکه‌ی عصبی از ۱ نخ برای لایه‌ی ورودی، ۸ نخ برای لایه‌ی میانی اول، ۱۰ نخ برای لایه‌ی خروجی و ۱ نخ برای ارائه‌ی عدد خروجی شبکه، که عدد پیش‌بینی شده توسط شبکه را به دست می‌دهد، تشکیل می‌شود. وزن‌ها و بایاس مربوط به هر نورون در لایه‌های میانی در فلدر `net params` قرار دارد.

لایه ورودی

در این لایه، یک نخ مسئول خواندن پیکسل‌های یک تصویر از فایل t10k-images-idx3-ubyte ، که در داخل فلدر data قرار دارد، می‌شود. از انجایی که تعداد کل تصاویر ۱۰۰۰۰ است، این نخ ۱۰۰۰۰ بار این کار را تکرار میکند.

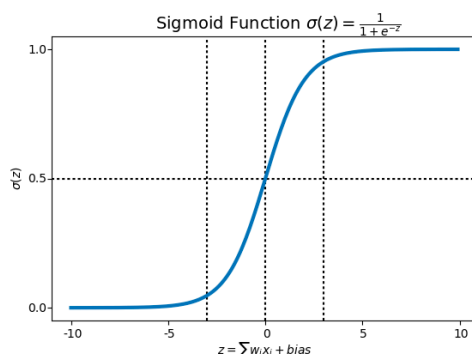
لایه میانی

در این لایه ۲۵۶ نورون وجود دارد که هر نورون دارای پارامترهای مخصوص به خود (وزن‌های متناظر با هر پیکسل تصویر و بایاس نورون) است. از ۸ نخ برای انجام محاسبات مربوط به هر نورون استفاده می‌کنیم که به این ترتیب هر نخ، وظیفه‌ی انجام محاسبات مربوط به ۳۲ نورون را برعهده دارد. تابع فعال‌ساز هر نورون را در این لایه Relu در نظر بگیرید.

$$RELU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

لایه خروجی

در این لایه نیز ۱۰ نورون، متناظر با اعداد ۰ تا ۹، وجود دارد که از ۱۰ نخ، به ازای هر نورون یک نخ، برای پیاده سازی آن استفاده می‌کنیم. فعال‌ساز نورون‌های این لایه را تابع سیگموید در نظر بگیرید.



عدد خروجی شبکه

در انتها، یک نخ لایه‌ی خروجی عدد پیش بینی شده توسط شبکه را به دست می‌آورد. برای این کار، مقدار ماکزیمم خروجی‌های لایه‌ی قبل را محاسبه کرده و عدد متناظر با آن مقدار را به عنوان عدد

پیش بینی شبکه در نظر بگیرید. سپس این عدد با عدد اصلی متناظر با آن تصویر، که در فایل t10k-labels-idx1-ubyte قرار دارد، مقایسه می‌کند و نرخ موفقیت شبکه را به روز می‌کند.

همگام سازی

برای انجام صحیح محاسبات لازم است سمافورهای در بین لایه‌های این شبکه تعبیه شوند که انتقال داده‌ها را در میان این لایه‌ها را مدیریت کنند. نحوه پیاده‌سازی سمافورها از تنوع بسیاری برخوردار است و استفاده از کمترین تعداد سمافورها مبنای کار است.

بخش امتیازی

تعداد نورون‌های لایه میانی به صورت دلخواه و در زمان اجرا تعیین شود.

سایر نکات

- در فلدر پروژه فایلی با نام Serial.cpp وجود دارد که در آن این شبکه عصبی به صورت سری پیاده سازی شده است. حتماً قبل از حضور در جلسه توجیهی، این فایل را مطالعه کنید.
- کدهای شما باید به زبان C++ نوشته شوند و با کامپایلر g++ قابل اجرا باشند.
- حتماً در جلسه توجیهی شرکت داشته باشید. نکاتی که در کلاس درس و فروم مطرح می‌شوند جزء پروژه هستند.
- این پروژه انفرادی است.
- کدهای خود را به صورت یک فایل zip. آپلود کنید.
- کد کسی را کپی نکنید. حتی یک تابع!