

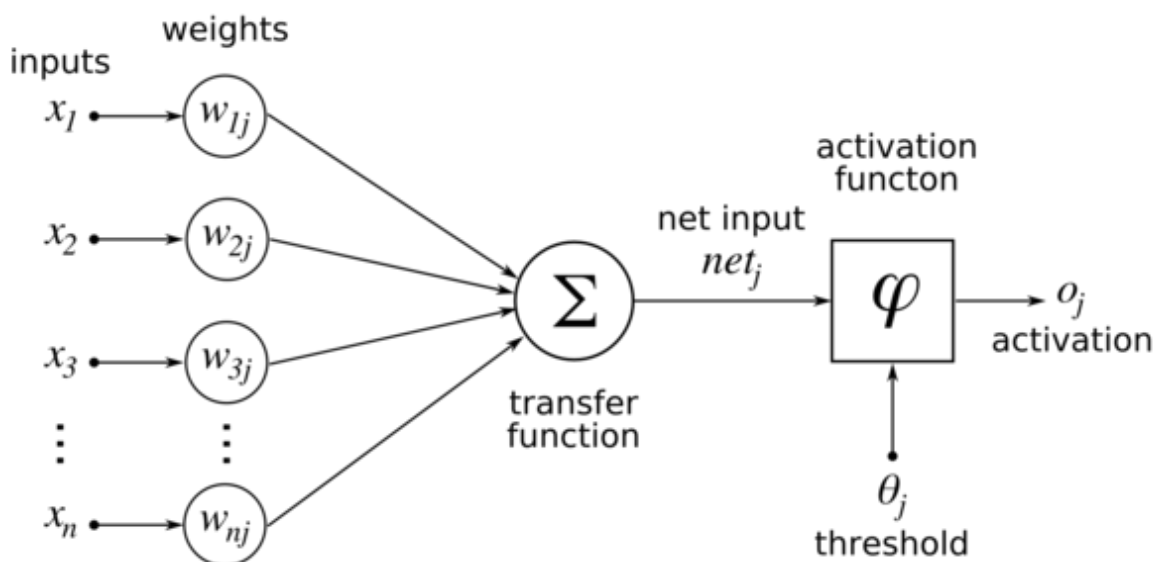


به نام خدا
درس سیستم عامل
پروژه سوم: نخ‌ها و همگام‌سازی



آشنایی با اهداف کلی پروژه

یکی از روش‌های محاسباتی نوین و پرکاربرد در زمینه یادگیری ماشین، نمایش دانش و اعمال دانش در جهت بیش‌بینی پاسخ‌های خروجی از سیستم‌های پیچیده، شبکه عصبی است. این شبکه از شبکه عصبی بیولوژیکی مغز حیوانات الهام می‌گیرد و می‌تواند رفتار سیستم‌های مختلف را شبیه‌سازی کند. شبکه عصبی از لایه‌های مختلفی تشکیل می‌شوند که در هر یک از این لایه‌ها تعدادی واحد محاسباتی متصل به هم به نام نورون وجود دارد. هرکدام از این نورون‌ها بخشی از کارهای محاسباتی مربوط به یادگیری را انجام می‌دهند و با ارتباط با نورن‌های دیگر خروجی خود را به آن‌ها منتقل می‌کنند. در این پروژه قصد داریم ساختار درونی یک نورون را پیاده‌سازی کنیم. شکل زیر ساختار داخلی یک نورون را نمایش می‌دهد.



همانطور که مشاهده می‌کنید در یک نورون، مقدار ورودی، که از لایه‌ی قبل دریافت شده، در وزن مربوط به آن ضرب می‌شود و سپس این مقادیر بدست‌آمده با یکدیگر و با بایاس مربوط به نورون جمع می‌شوند و در نهایت یک تابع فعال‌ساز روی این مقدار اعمال می‌شود که خروجی این تابع، خروجی نورون موردنظر است.

نحوه پیاده‌سازی ساختار یک نورون

فایل های ورودی:

در ابتدا دو فایل در اختیار شما قرار میگیرد که یکی از این فایل ها حاوی ورودی های نورون و دیگری حاوی وزن مربوط به هر ورودی نورون و بایاس نورون است.

ریسمان های دریافت کننده ورودی های نورون و وزن هر ورودی:

در هر تکرار از عملیات نورون یک ریسمان موظف به خواندن ورودی های نورون است و ریسمان دیگری وزن های ورودی ها را دریافت می کند. این نورون ۱۲۸ ورودی دارد.

ریسمان های میانی:

این ریسمان مانند یک واحد MAC^۱ عمل می کند و مقادیر ورودیش را یک به یک در هم ضرب کرده با مقادیر ضرب شده ی قبلی جمع می کند.

در این قسمت تعداد متغیری از ریسمان ها وظیفه ضرب مقادیر دریافت شده از دو ریسمان قبل را دارند. توجه کنید که تعداد ریسمان ها در زمان اجرا مشخص می شوند و ۱۲۸ عملیات ضرب بین این ریسمان ها تقسیم می شود. نحوه ی تخصیص منابع به ریسمان ها در این حالت ایستا است.

ریسمان تولید کننده خروجی:

این ریسمان مقادیر خروجی ریسمان های میانی را با هم جمع می کند تا بدین ترتیب جمع همه ی ۱۲۸ عملیات ضرب انجام شده بدست آید. سپس این مقدار را با بایاس نورون جمع کرده و وارد تابع فعال ساز می کند. در این قسمت از تابع $\arctan(x)$ به عنوان تابع فعالساز استفاده کنید. خروجی این تابع به عنوان خروجی نورون در یک فایل ذخیره می شود و محاسبات مربوط به یک سطر از ورودی ها خاتمه می یابد.

سمافورها:

نحوه ی استفاده از سمافورها به منظور دستیابی به حداکثر تسریع از اهمیت بسیاری برخوردار است.

محاسبه میزان تسریع:

میزان تسریع این عملیات را با پیاده سازی پروژه به صورت سری به دست آورید. این کار درکی از موازی سازی شبکه با استفاده از نخ ها بدست می دهد.

بخش امتیازی:

در بخش ریسمان های میانی، هر ریسمان پس از انجام یک عملیات ضرب، اختصاص یافتن ورودی های دیگری را از ریسمان های ورودی درخواست کند. بدیهی است این کار باید به کمک سمافورها انجام شود. به این ترتیب نحوه ی اختصاص یافتن ورودی ها به هر ریسمان پویا است و در زمان محاسبه صرفه جویی به عمل می آید.

¹ multiply-accumulator

سایر نکات

- کدهای شما باید به زبان C++ نوشته شوند و با کامپایلر g++ قابل اجرا باشند.
- حتماً در جلسه‌ی 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!