

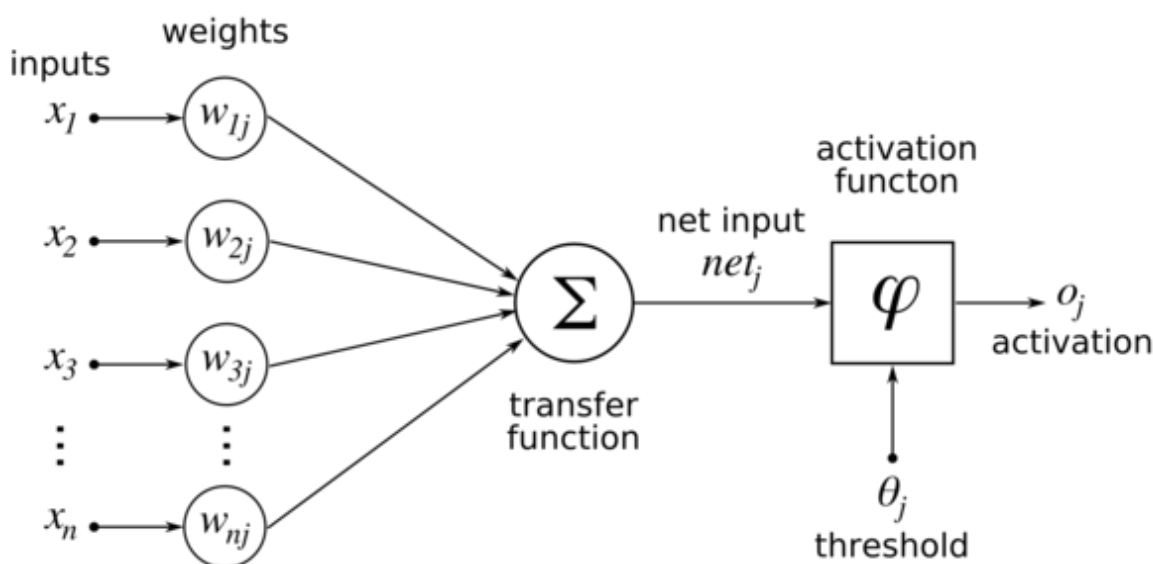


به نام خدا
درس سیستم عامل
پروژه سوم: نخ‌ها و همگام‌سازی



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 فایل‌ها حاوی ورودی‌های نورون و دیگری حاوی وزن مربوط به هر ورودی نورون و بایاس نورون است.

ریسمان‌های دریافت کننده ورودی‌های نورون و وزن هر ورودی:

در هر تکرار از عملیات نورون یک ریسمان موظف به خواندن ورودی‌های نورون است و ریسمان دیگری وزن‌های ورودی‌ها را دریافت می‌کند. این نورون ۱۲۸ ورودی دارد.

ریسمان‌های میانی:

این ریسمان مانند یک واحد MAC^۱ عمل می‌کند و مقادیر ورودیش را یک به یک در هم ضرب کرده با مقادیر ضرب‌شده‌ی قبلی جمع می‌کند.

در این قسمت تعداد متغیری از ریسمان‌ها وظیفه ضرب مقادیر دریافت شده از دو ریسمان قبل را دارند. توجه کنید که تعداد ریسمان‌ها در زمان اجرا مشخص می‌شوند و ۱۲۸ عملیات ضرب بین این ریسمان‌ها تقسیم می‌شود. نحوه‌ی تخصیص منابع به ریسمان‌ها در این حالت ایستا است.

ریسمان تولید کننده خروجی:

این ریسمان مقادیر خروجی ریسمان‌های میانی را با هم جمع می‌کند تا بدین ترتیب جمع همه‌ی ۱۲۸ عملیات ضرب انجام شده بدست آید. سپس این مقدار را با بایاس نورون جمع کرده و وارد تابع فعال‌ساز می‌کند. در این قسمت از تابع $\arctan(x)$ به عنوان تابع فعال‌ساز استفاده کنید. خروجی این تابع به عنوان خروجی نورون در یک فایل ذخیره می‌شود و محاسبات مربوط به یک سطر از ورودی‌ها خاتمه می‌یابد.

سمافورها:

نحوه‌ی 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 نخ‌ها بدست می‌دهد.

بخش امتیازی:

در بخش ریسمان‌های میانی، هر ریسمان پس از انجام یک عملیات ضرب، اختصاص یافتن ورودی‌های دیگری را از ریسمان‌های ورودی درخواست کند. بدیهی است این کار باید به کمک سمافورها انجام شود. به این ترتیب نحوه‌ی اختصاص یافتن ورودی‌ها به هر ریسمان پویا است و در زمان محاسبه صرفه جویی به عمل می‌آید.

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!