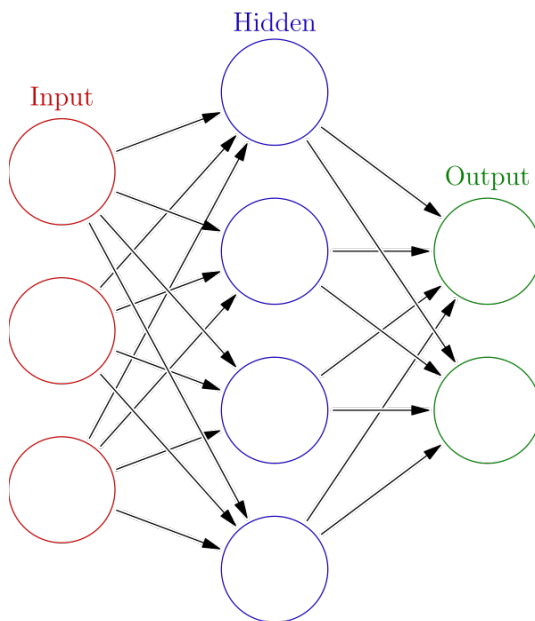




## آشنایی با اهداف کلی پروژه

در این پروژه قصد داریم یک شبکه‌ی عصبی مصنوعی<sup>۱</sup> را پیاده‌سازی کنیم. شبکه عصبی مصنوعی سیستمی محاسباتی است که از شبکه عصبی بیولوژیکی که مغز حیوانات را تشکیل می‌دهد الهام می‌گیرد. این شبکه عصبی میتواند رفتار سیستمی، که ورودی‌های سیستم و خروجی‌های متناظر با آن‌ها به آن داده شده است، را یاد بگیرد، به گونه ای که با در گرفتن ورودی یک سیستم بتواند خروجی آن را با تقریب مناسبی محاسبه کند.

شبکه عصبی از تعدادی واحد محاسباتی متصل بهم به نام نورن تشکیل شده است که هرکدام از این نورن‌ها بخشی از کارهای محاسباتی مربوط به یادگیری را انجام می‌دهند و به‌وسیله‌ی اتصالات خود به نورن‌های دیگر این محاسبات را به آن‌ها منتقل می‌کند. همچنین هر یال ارتباطی بین نورن‌ها یک وزن دارد که در طی فرآیند یادگیری این وزن‌ها به‌روز می‌شوند تا شبکه محاسباتی مناسبی برای شبیه سازی رفتار سیستم مذکور به‌دست آید. معمولاً این نورن‌ها در لایه‌هایی قرار می‌گیرند که هرلایه نوع متفاوتی از عملیات را روی ورودی‌های خود پیاده می‌کند. شکل زیر شبکه‌ی عصبی با سه لایه<sup>۲</sup> را به نمایش می‌گذارد:



<sup>1</sup>Artificial Neural Network

<sup>2</sup>Input Layer, Hidden Layer, Output Layer

پس از اینکه شبکه عصبی موفق به یادگیری رابطهی بین ورودی‌ها و خروجی‌های متناظر با آن‌ها شد، می‌تواند رفتار سیستماتیک آن‌ها را شبیه سازی کند.

در این پروژه قصد داریم رفتار یک تابع ریاضی سه متغیره<sup>3</sup> را که به وسیلهی شبکه عصبی شبیه سازی شده است، بررسی کنیم. به این منظور این تابع در نرم افزار متلب train شده است. حاصل این کار به دست آمدن وزن یال‌های متصل کنندهی نورون‌های هر لایه به نورون‌های لایه دیگر و میزان بایاس برای هریک از نورون‌های لایه‌ی پنهان و لایه‌ی خروجی شبکه است. فایلی شامل این مقادیر که حاصل train کردن شبکه است، در اختیار شما قرار میگیرد. لازم است، مطابق شکل بالا، شبکه‌ای با ۳ لایه ایجاد کنید. به صورتیکه نورون لایه‌ی ورودی با یال‌هایی به همه‌ی نورون‌های لایه‌ی پنهان متصل شود و همه‌ی نورون‌های لایه‌ی پنهان هم به نورون لایه‌ی خروجی متصل شوند. در قسمت‌های آینده توضیح بیشتری در این مورد داده می‌شود. شما همچنین می‌بایست این وزن‌ها و مقادیر بایاس را، که در فایل layersinfo.txt قرار داده شده است، به ترتیب به یال‌های بین نورون‌ها و خود نورون‌ها اختصاص دهید.

همانطور که مشخص است در پیاده‌سازی این شبکه نخ‌ها نقش نورون‌های لایه‌های مختلف شبکه را ایفا می‌کنند.

## لایه ورودی

در ابتدا می‌بایست به ازای هر متغیر اعدادی رندوم، این اعداد فرم اعشاری دارند و در بازه [0, 50] قرار می‌گیرند، تولید کرده و آن‌ها را در فایلی ذخیره کنید<sup>4</sup>. این لایه وظیفه دارد ورودی‌ها را از این فایل خوانده و برای پردازش به لایه‌ی بعدی ارسال کند. پس لازم است عملیات خواندن یک مقدار از متغیرها توسط نخ‌ی انجام شده و مجدداً نخ برای خواندن سری بعدی مقادیر این ۳ متغیر آماده شود.

## لایه پنهان

این لایه از ۱۰ نورن تشکیل شده است. هر یک از نورن‌های این لایه مقادیر فرستاده شده توسط لایه قبل را در وزن متناظر با آن‌ها ضرب کرده و مقادیر ضرب شده را باهم و با بایاس مربوط به خود جمع می‌کنند. سپس یک تابع فعالساز<sup>5</sup>، که برای نگاشتن این مقادیر به بازه‌ی [1, -1] به کار می‌رود، روی این مقادیر اعمال می‌شود و نتیجه به عنوان مقدار خروجی این نورن به لایه بعدی فرستاده می‌شود.

---

<sup>3</sup> تابع ریاضی train شده در فایل func.txt قرار گرفته است.

<sup>4</sup> نمونه‌ای از این فایل با نام InputFile.txt در پوشه پروژه قرار گرفته است.

<sup>5</sup> Activation Function; برای پیاده‌سازی آن از تابع tanh استفاده شود.

## لایه خروجی

این لایه نیز ورودی‌های خود را که از لایه پنهان دریافت کرده است در وزن یال‌های متناظر ضرب کرده و حاصل مقادیر را با هم و با بایاس خود جمع میکند و به عنوان خروجی در یک فایل می‌نویسد. در این لایه تابع فعال‌ساز تابع خطی  $x = y$  است که به هر مقدار خودش را می‌نگارد پس نیازی به پیاده‌سازی آن نمی‌باشد. در این لایه نیز از یک نخ به عنوان نورن خروجی استفاده کنید.

## همگام سازی

برای اینکه هر سری از مقادیر متغیرها به صورت صحیح محاسبه شده و به لایه‌های بعدی منتقل شوند لازم است سمافورهای در بین لایه‌ها تعبیه شوند که انتقال داده‌ها را در میان لایه‌ها مدیریت کنند. نحوه‌ی پیاده‌سازی سمافورها از تنوع بسیاری برخوردار است و استفاده از کمترین تعداد سمافورها مبنای کار است. مشخص است که پیاده‌سازی این پروژه از تنوع بالایی برخوردار است و نحوه پیاده‌سازی بسیار حائز اهمیت است و پیاده‌سازی بهینه در نمردهی تاثیر بسیاری دارد.

## بخش امتیازی

۱. یک نخ جدید پس از لایه خروجی قرار دهید که پس از دریافت مقادیر خروجی، میزان انحراف معیار را به‌ازای تمامی این مقادیر محاسبه کند.
۲. میزان تسریع این عملیات را با پیاده‌سازی سریال پروژه به دست آورید. این کار درکی از موازی سازی شبکه، به‌کمک نخ‌ها، بدست می‌دهد.

## 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!