

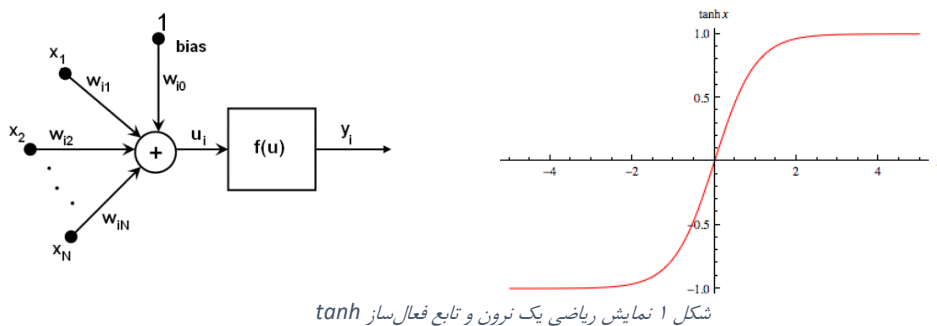


تمرین کامپیوتری شماره ۱ برنامه نویسی موازی

نیمسال اول ۹۶-۹۷

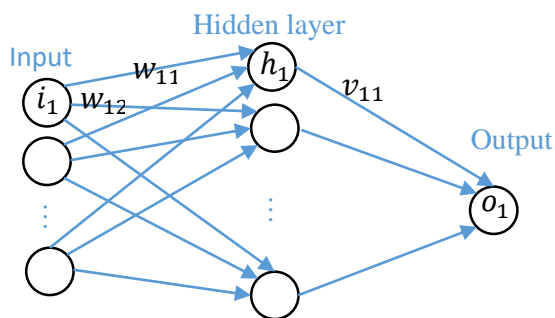
زمان آپلود: روز دوشنبه ۸ آبان (تا ساعت ۱۲ شب)

پایه‌ای‌ترین قسمت یک شبکه عصبی، نرون (Neuron) می‌باشد. نرون زیر را در نظر بگیرید، ورودی‌ها در وزن‌های نرون (Weight) ضرب می‌شوند و با بایاس جمع می‌شوند و در خروجی نرون یک تابع فعال‌ساز (Activation Function) روی آن اعمال می‌شود و خروجی نرون از این تابع به دست می‌آید.



### شبکه چندلایه پرسپترون

شبکه پرسپترون از چندین نرون در هر لایه تشکیل می‌شود که در شکل ۲ مشخص است.



شکل ۲ گراف یک شبکه با یک لایه مخفی

هدف این تمرین پیاده سازی و آموزش یک شبکه با استفاده از دستورات SIMD اینتل است.

### مجموعه داده:

این مجموعه شامل ۱۰۳۰ نمونه داده جمع آوری شده از میزان استحکام تراکم بتون می‌باشد. میزان استحکام بتون تابعی به شدت غیر خطی است.

هر نمونه داده ۹ ویژگی دارد و هدف این مساله پیش‌بینی پارامتر نهم با استفاده از سایر ویژگی‌ها در همان لحظه است. پس لایه ورودی ۸ نرون دارد و لایه خروجی ۱ نرون خواهد داشت. برای یک لایه مخفی به منظور پیاده سازی بهتر SIMD ۱۶ نرون استفاده کنید.

برای سادگی، پیش پردازش‌های لازم روی همه نمونه داده‌ها انجام شده است و نمونه داده‌ها آماده برای استفاده هستند. برای آشنایی بیشتر با مجموعه داده می‌توانید به آدرس زیر مراجعه نمایید.

<http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>

## آموزش شبکه:

پایه‌ای ترین روش آموزش شبکه‌های عصبی، پس انتشار خطا است که در اینجا استفاده خواهیم کرد:

مرحله ۰. مقداردهی اولیه وزن‌ها و بایاس مقادیر تصادفی بسیار کوچک است. نرخ یادگیری آلفا را ۰,۰۱ در نظر می‌گیریم.  
مرحله ۱. در این مرحله یک ورودی را انتخاب کرده و با ضرب ماتریس ورودی (که یک ماتریس ۸ عضوی است) در ماتریس وزن‌های لایه اول (که یک ماتریس ۸ در ۱۶ است) و جمع بایاس‌ها، مقادیر ورودی به لایه بعد را به دست می‌آوریم.  
همین کار را برای لایه آخر هم انجام می‌دهیم (ماتریس وزن ۱۶ عضوی است) تا خروجی شبکه حاصل گردد.

$$y = \tanh(b + \sum x_i w_i)$$

معادله ۱ محاسبه خروجی نرون

مرحله ۲. برای محاسبه خطای خروجی حاصل با خروجی مطلوب آن نمونه داده آموزشی از مجموع مربعات خطا استفاده می‌کنیم.

$$E_{total} = \sum \frac{1}{2} (target - out)^2$$

معادله ۲ محاسبه خطای خروجی

مرحله ۳. محاسبه گرادیان وزن‌های لایه آخر

$$\begin{aligned} \frac{\partial E_{total}}{\partial v_{11}} &= \frac{\partial E_{total}}{\partial out\ o1} * \frac{\partial out\ o1}{\partial net\ o1} * \frac{\partial net\ o1}{\partial v_{11}} \\ &= (out\ o1 - target\ o1) * [1 - \tanh^2(out\ o1)] * out\ h1 \\ &= \delta o_1 * out\ h_1 \end{aligned}$$

برای محاسبه  $\frac{\partial E_{total}}{\partial bias}$  بجای ورودی  $out\ h_1$  مقدار ۱ را استفاده میکنیم و مشابه وزن به روزرسانی می‌شود.

مرحله ۴. محاسبه گرادیان وزن‌های لایه نخست

$$\frac{\partial E_{total}}{\partial w_{11}} = \frac{\partial E_{total}}{\partial out\ h1} * \frac{\partial out\ h1}{\partial net\ h1} * \frac{\partial net\ h1}{\partial w_{11}}$$

$$= (\delta o_1 * v_{11}) * [1 - \tanh^2(out\ h1)] * i_1$$

برای محاسبه  $\frac{\partial E_{total}}{\partial bias}$  بجای ورودی  $i_1$  مقدار ۱ را استفاده میکنیم و مشابه وزن به روزرسانی می‌شود.

مرحله ۵. پس از محاسبه گرادیان همه وزن‌ها، به روزرسانی انجام می‌شود

$$weight^+ = weight^- - \alpha * \frac{\partial E_{total}}{\partial weight} \quad bias^+ = bias^- - \alpha * \frac{\partial E_{total}}{\partial bias}$$

*معادله ۳ به روزرسانی وزن‌ها*

مرحله ۶. دوباره به مرحله ۱ می‌رویم. این کار را تا جایی که خطا میزان کمی داشته باشد ادامه می‌دهیم.

### نکته‌ها:

- این روش آموزش گرادیان نزولی تصادفی نام دارد که ما گرادیان نزولی دسته‌ای را استفاده می‌کنیم. در هر بار همه داده‌ها را اعمال می‌کنیم و گرادیان همه وزن‌ها را برای همه این داده‌ها محاسبه می‌کنیم و با میانگین گرادیان وزن‌ها به روزرسانی انجام می‌دهیم.
- کد سریال ضمیمه شده است و باید میزان تسریع حاصل خود را گزارش کنید.
- پایاده سازی SIMD برای قسمت آموزش ضروری است اما با موازی سازی همه قسمت‌ها و بستن شبکه با تعداد لایه بیشتر تا ۱۵ درصد نمره بیشتر در نظر گرفته شده است.