

شبکه نورونی سه لایه

۱.۱ لود کردن دیتاست

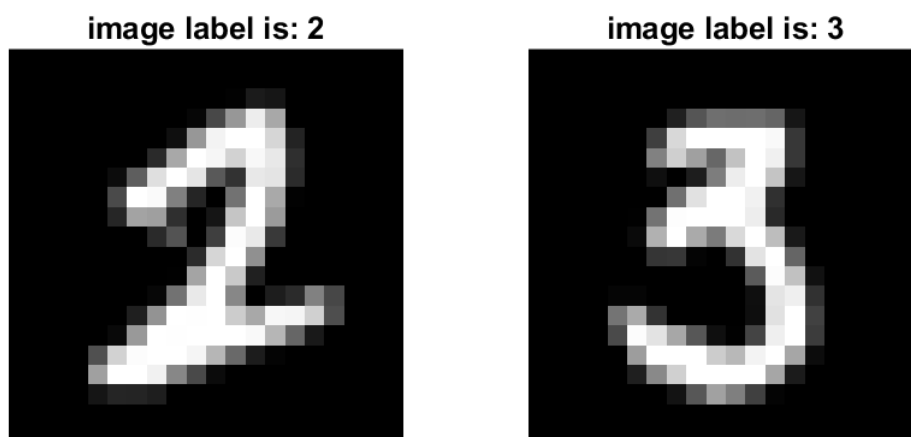
پس از لود کردن دیتاست، ۱۰۰ تصویر به صورت تصادفی انتخاب گردید و در شکل زیر نمایش داده شد.



شکل ۱.۱: صد تصویر از اعداد دست نویس

۲.۱ جداسازی داده های train و test

در این بخش مطابق با آنچه که گفته شده است، ۳۰۰ تای اول داده ها به عنوان داده های train و ۲۰۰ تای بعد به عنوان تست ذخیره گردید و متناظر با این تصاویر لیبل مناسب نیز انتخاب گردید و ماتریس های مورد نظر ساخته شد. دو نمونه از داده های تست با لیبل متناظر در زیر نشان داده شده است.



شکل ۲.۱: تصویر و لیبل متناظر

۳.۱ ساختن ساختار شبکه عصبی

ابتدا مطابق فرمول داده شده در زیر، وزن ها را به صورت رندم در بازه ی $[-\epsilon, \epsilon]$ مقدار دهی می کنیم، علت این کار که بجای صفر از مقادیر رندم برای مقدار دهی اولیه استفاده می شود آن است که در فرآیند backpropagation با صفر بودن همه ی وزن ها نورون های مختلف در فرآیند یادگیری وزن های یکسان دریافت می کنند و در نهایت همه ی نورون های یک لایه یک ویژگی را استخراج خواهند کرد که این امر باعث ناکارآمدی شبکه خواهد شد.

$$W = rand(L_{out}, 1 + L_{in}) * 2 * \epsilon - \epsilon, \quad \epsilon = 0.12$$

علت آنکه لینک نورون های ورودی یکی بیشتر از تعداد مد نظر است وجود بایاس در ماتریس وزن های نورونی است.

۴.۱ الگوریتم تئوری محاسبه ی تابع هزینه و مشتق آن

در این قسمت ابتدا مشتق تابع هزینه را نسبت به هر یک از وزن های هر یک از لایه ها محاسبه می کنیم:

$$a^{(1)} = [1 \ X] \quad z^{(2)} = a^{(1)}\theta^{(1)T} \quad a^{(1)} = \sigma(z^{(2)}) \quad a^{(3)} = [1 \ a^{(2)}] \quad z^{(3)} = a^{(3)}\theta^{(2)T} \quad a^{(4)} = \sigma(z^{(3)})$$

تابع سیگموید و مشتق آن به صورت زیر تعریف می شود:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

$$\frac{d\sigma(x)}{dx} = \sigma(x) * (1 - \sigma(x))$$

برای وزن های لایه دوم داریم:

$$J(\theta) = \sum_{k=1}^K [-y_k * \log(a_K^{(4)}) - (1 - y_k) * \log(1 - a_K^{(4)})]$$

$$\frac{\partial J(\theta)}{\partial \theta_{k,j}^{(2)}} = \frac{\partial J(\theta)}{\partial a_K^{(4)}} * \frac{\partial a_K^{(4)}}{\partial z_K^{(3)}} * \frac{\partial z_K^{(3)}}{\partial \theta_{k,j}^{(2)}}$$

$$\frac{\partial J(\theta)}{\partial a_K^{(4)}} = \frac{-y_k}{a_k^{(4)}} + \frac{1 - y_k}{1 - a_k^{(4)}} = \frac{a_k^{(4)} - y_k}{a_k^{(4)}(1 - a_k^{(4)})}$$

$$\frac{\partial a_K^{(4)}}{\partial z_K^{(3)}} = \sigma(z_K^{(3)}) * (1 - \sigma(z_K^{(3)})) = a_k^{(4)}(1 - a_k^{(4)})$$

$$\frac{\partial z_K^{(3)}}{\partial \theta_{k,j}^{(2)}} = a_j^{(3)}$$

با تجميع روابط فوق خواهیم داشت:

$$\frac{\partial J(\theta)}{\partial \theta_{k,j}^{(2)}} = \begin{cases} \frac{1}{m} \sum_{i=1}^m (a_k^{(4)} - y_k) a_j^{(3)} & \text{if } j = 0 \\ \frac{1}{m} \sum_{i=1}^m (a_k^{(4)} - y_k) a_j^{(3)} + \frac{\lambda}{m} \theta_{k,j} & \text{if } j \neq 0 \end{cases}$$

برای وزن های لایه ی اول نیز داریم:

$$\begin{aligned}
J(\theta) &= \sum_{k=1}^K [-y_k * \log(a_K^{(3)}) - (1 - y_k) * \log(1 - a_K^{(3)})] \\
\frac{\partial J(\theta)}{\partial \theta_{k,j}^{(2)}} &= \frac{\partial J(\theta)}{\partial a_K^{(4)}} * \frac{\partial a_K^{(4)}}{\partial z_K^{(3)}} * \frac{\partial z_K^{(3)}}{\partial a_j^{(3)}} * \frac{\partial a_j^{(2)}}{\partial z_j^{(2)}} * \frac{\partial z_j^{(2)}}{\partial \theta_{j,i}^{(1)}} \\
\frac{\partial J(\theta)}{\partial a_K^{(4)}} &= \frac{-y_k}{a_k^{(4)}} + \frac{1 - y_k}{1 - a_k^{(4)}} = \frac{a_k^{(4)} - y_k}{a_k^{(4)}(1 - a_k^{(4)})} \\
\frac{\partial a_K^{(4)}}{\partial z_K^{(3)}} &= \sigma(z_K^{(3)}) * (1 - \sigma(z_K^{(3)})) = a_k^{(4)}(1 - a_k^{(4)}) \\
\frac{\partial z_K^{(3)}}{\partial a_j^{(3)}} &= \theta_{k,j}^{(2)} \\
\frac{\partial a_j^{(2)}}{\partial z_j^{(2)}} &= \sigma(z_j^{(2)}) * (1 - \sigma(z_j^{(2)})) \\
\frac{\partial z_j^{(2)}}{\partial \theta_{j,i}^{(1)}} &= a_i^{(1)}
\end{aligned}$$

با جمع روابط داریم:

$$\frac{\partial J(\theta)}{\partial \theta_{j,i}^{(1)}} = \begin{cases} \frac{1}{m} \sum_{i=1}^m (a_k^{(4)} - y_k) * \theta_{k,j}^{(2)} * \sigma(z_j^{(2)}) * (1 - \sigma(z_j^{(2)})) * a_i^{(1)} & \text{if } j = 0 \\ \frac{1}{m} \sum_{i=1}^m (a_k^{(4)} - y_k) * \theta_{k,j}^{(2)} * \sigma(z_j^{(2)}) * (1 - \sigma(z_j^{(2)})) * a_i^{(1)} + \frac{\lambda}{m} \theta_{j,i} & \text{if } j \neq 0 \end{cases}$$

۵.۱ پیاده سازی شبکه عصبی با استفاده از الگوریتم Backpropagation

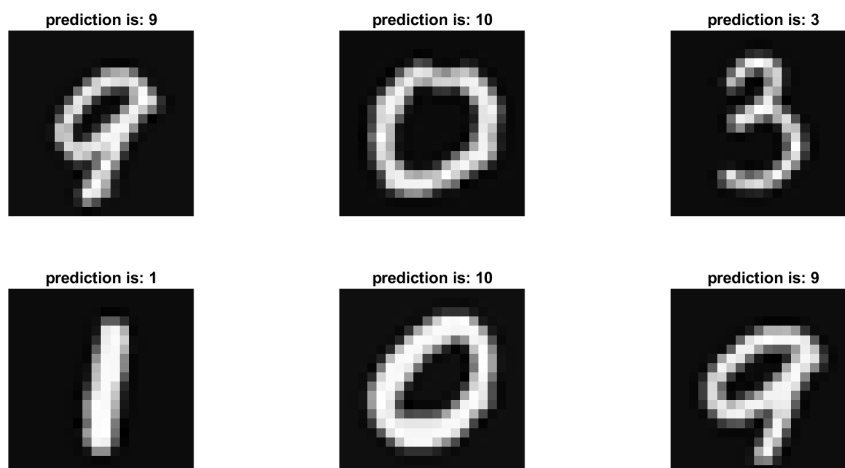
در این بخش با توجه به فرمول های بدست آمده در بالا و نکات گفته شده در صورت سوال، یک تابع نوشته شد که پارامترهای شبکه، تعداد نورون هر لایه، ورودی و خروجی را دریافت می کند و در نهایت مقدار تابع هزینه و مشتق وزن ها را به خروجی توسط الگوریتم backpropagation اعلام می کند.

۶.۱ پردازش شبکه عصبی با استفاده از تابع قبل

در این مرحله وزن های اولیه ی شبکه و تابع محاسبه ی مقدار تابع هزینه و مشتق وزن ها را با بهینه ساز داده شده می دهیم و در نهایت این بهینه ساز، وزن های بهینه را در خروجی ایجاد می کند که با داشتن این وزن ها می توان خروجی را با توجه به فرمول های بالا ایجاد نمود.

۷.۱ مشاهده نتایج لایه آخر شبکه عصبی

پس از انجام روندهای بالا برای چند نمونه از داده های تست، تصویر ورودی و خروجی پیش بینی شده توسط شبکه را نشان می دهیم.

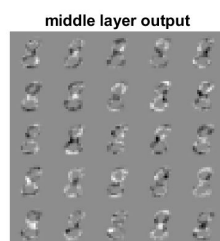


شکل ۳.۱: تصویر و خروجی شبکه

همانطور که در تمرین نیز اشاره شده است شماره ی ۱۰ متناظر با عدد صفر است.

۸.۱ مشاهده نتایج لایه میانی و محاسبه ی دقت شبکه عصبی

برای نمایش نتایج لایه ی میانی برای یک تصویر برداری شده تمامی وزن ها به صورت المانی ضرب گردید اما جمع نهایی انجام نشد. به این طریق ۲۵ تصویر با ابعاد تصویر اصلی بدست آمد که برای یک مورد به صورت زیر نشان داده شد.



شکل ۴.۱: نتایج لایه ی میانی

برای محاسبه ی دقت شبکه نیز تعداد پیش بینی های درست شبکه بر تعداد کل داده های تست تقسیم گردید و دقت 87.6 درصد بدست آمد.