

## ۱ مقدمه

وقتی که با شکلهایی مانند شکل ۱ مواجه می شوید بلافاصله اعداد را تشخیص می دهید

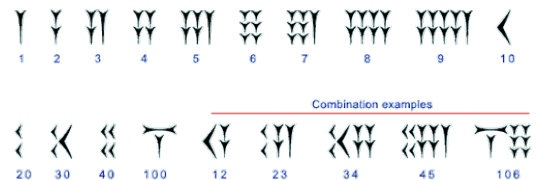


شکل ۱: پلاک فارسی

ولی وقتی با اعداد جدید (شکل ۲) برخورد می کنیم نمی دانیم که هر قسمت نشان دهنده چه چیزی است.



شکل ۲: اعداد میخی



شکل ۳: اعداد میخی

```

if PictureOfONE() and
PictureOfSingleCog() then
    return 2;
else
    if PictureOfONE() and
PictureOfDoubleCog() then
        return 3;
    end
end
end

```

Algorithm 1: first model

ولی همانطور که تا الان متوجه شدید این مدل مشکلاتی دارد، مانند:

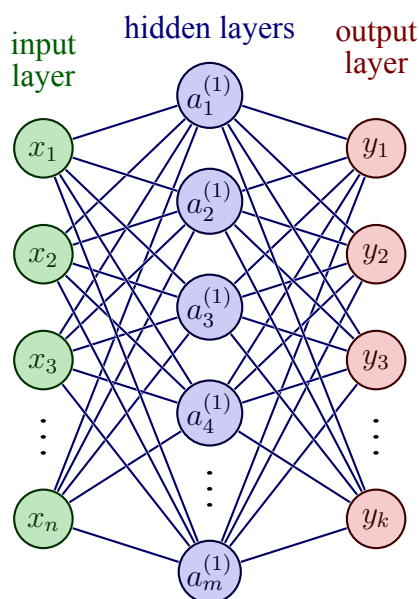
- سختی نوشتن کد
  - سختی پیدا کردن مشترکات توسط انسان
  - دقیق نبودن در بعضی مسائل
- پس نیاز است که مدل دیگری را بسازیم تا این مشکلات را برطرف کند.

تا زمانی که جواب های این شکل ها به ما نمایش می دهند (شکل ۲) و می توانیم روابط بین هر عدد را پیدا کنیم، به عنوان مثال عدد ۲۰ از دو عدد ۱۰ تشکیل شده و عدد ۳۰ از یک ۲۰ و یک ۱۰ تشکیل شده است؛ حال بیاید و این مشترکات را برای اعداد فارسی پیدا کنیم. اگر به سه عدد در شکل ۶ دقت کنید قسمت های مشترکی دارند، به عنوان مثال خمیدگی عدد ۱ در عدد ۲ و ۳ نیز قابل مشاهده است و می توان گفت که عدد دو از دو قسمت شکل ۵ تشکیل شده است. که قسمت اول آن از شکل عدد یک و قسمت دوم از یک دندان تشکیل شده و ما می توانیم با استفاده از

## ۲ شبکه عصبی

### ۲.۲ ساخت مدل عصبی برای کامپیوتر

حال که مقداری فهمیدیم که مغز انسان چگونه اعداد را تشخیص می‌دهد می‌توانیم مدلی مانند آن بسازیم تا با استفاده از آن بتوانیم اعداد را تشخیص دهیم. اگر بخواهیم ساختاری مانند ساختار عصبی را برای کامپیوتر بسازیم به شکل ۷ می‌رسیم.



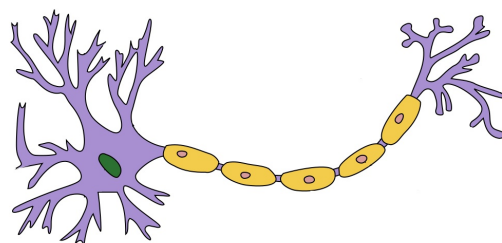
شکل ۷: شبکه عصبی با ۱ لایه پنهانی

هر لایه از تعدادی گره یا نود تشکیل شده است، که مقدار گره‌های  $x_i$  مقدارهای ورودی می‌شوند که اگر برای عکس بخواهیم در نظر بگیریم مشخصات پیکسل‌های عکس می‌شوند و تعداد آن به کیفیت و تعداد پیکسل‌های عکس بستگی دارد. لایه آخر یعنی لایه  $y_k$  لایه خروجی نامیده می‌شود و نشان دهنده تعداد خروجی‌هایی است که می‌توانیم داشته باشیم، به عنوان مثال برای تشخیص عدد تعداد لایه خروجی ۱۰ می‌باشد. بین لایه ورودی و لایه خروجی لایه پنهانی وجود دارد که هرچه تعداد آن بیشتر باشد دقت مدل ما بیشتر می‌شود.

در صنعت بسیاری از محصولاتی که ساخته می‌شود برگرفته از حیوانات یا اشیاء درون طبیعت می‌باشد، به عنوان مثال ایده سنسورهای التراسونیک که از خفاش‌ها برگرفته شده. پس برای ساخت مدل جدید می‌توانیم از انسان‌ها ایده بگیریم.

### ۱.۲ تشخیص اعداد توسط انسان

در انسان، سیستم بینایی تا ۸۰ درصد از تمام داده‌های حسی دریافتی از محیط را جمع‌آوری می‌کند. این سیل اطلاعات نوری، ورودی‌های بصری که توسط تقریباً ۱۳۰ میلیون سلول حساس به نور در شبکه چشم گرفته شده و به سیگنال‌های الکتروشیمیایی تبدیل می‌شوند، به شبکه پیچیده‌ای از سلول‌های عصبی وارد شده و پردازش می‌شوند. [۱]



شکل ۶: سلول عصبی

هنگامی که شما به شکل ۲ نگاه کردید مانند نوزادی که تازه شروع به یادگیری اعداد کرده است نمی‌دانستید هر شکل نشان دهنده چیست و در اصل سلول‌های عصبی نمی‌دانستند که با سیگنال‌های الکتروشیمیایی هر شکل چه خروجی دهند و به محض اینکه جواب هر شکل را در شکل ۳ دیدید با تغییراتی که در این سلول‌های عصبی به وجود آمد دیگر جواب‌های هر شکل را می‌دانید و در اصل یاد گرفتید.

### ۳.۲ چگونه انجام عملیات برای خروجی گرفتن (فاز ۲ پروژه)

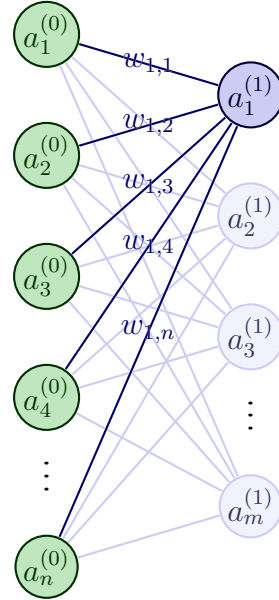
اگر معادله ۱ را برای تمام گره‌ها در نظر بگیرید به ضرب ماتریس فرمول ۲ می‌رسید.

$$A^{(1)} = W^{(0)}A^{(0)} + b^{(0)} \quad (۲)$$

که در آن

$$A^{(1)} = \begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} \quad A^{(0)} = \begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \\ \vdots \\ a_m^{(0)} \end{pmatrix} \quad b^{(0)} = \begin{pmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{pmatrix}$$

$$W^{(0)} = \begin{pmatrix} w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ w_{2,0} & w_{2,1} & \dots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,0} & w_{m,1} & \dots & w_{m,n} \end{pmatrix} \quad (۳)$$



شکل ۸: قسمتی از شبکه عصبی

همانطور که در شکل ۸ هر اتصال دارای وزن  $w_{i,j}$  که  $i$  نشان دهنده شماره گره مقصد و  $j$  نشان دهنده گره قبلی می‌باشد. هر گره دارای مقدار عددی بین ۰ و ۱ است و هر  $w_{i,j}$  دارای مقدار بین -۱ و ۱ می‌باشد و نحوه محاسبه گره مقصد به صورت معادله ۱ می‌باشد.

$$\begin{aligned} a_1^{(1)} &= \sigma(w_{1,0}a_0^{(0)} + w_{1,1}a_1^{(0)} \\ &\quad + \dots + w_{1,n}a_n^{(0)} + b_1^{(0)}) \\ &= \sigma\left(\sum_{i=1}^n w_{1,i}a_i^{(0)} + b_1^{(0)}\right) \end{aligned} \quad (1)$$

که در این معادله  $\sigma$  تابع فعال‌سازی است و کار آن تبدیل مقدار معادله ۱ به بازه ۰ و ۱ می‌باشد. و همچنین مقدار  $b_1^{(1)}$  را داریم که برای کم کردن خطا می‌باشد.

#### Algorithm 2: code for model with one hidden layer

//Initialize weight matrix

$W^{(0)} \leftarrow$  random  $m \times d$  matrix with value's in  $(-1,1)$

$W^{(1)} \leftarrow$  random  $p \times m$  matrix with value's in  $(-1,1)$

//Initialize bias vectors

$b^{(0)} \leftarrow$  random  $m \times 1$  matrix

$b^{(1)} \leftarrow$  random  $p \times 1$  matrix

**while**  $end_{of\_TrainSet}$  **do**

$X \leftarrow$  Picture  $d \times 1$  matrix

$Z^{(0)} \leftarrow W^{(0)} * X + b^{(0)}$

$A^{(0)} \leftarrow \sigma(Z^{(0)})$

$Z^{(1)} \leftarrow W^{(1)} * A^{(0)} + b^{(1)}$

$A^{(1)} \leftarrow \sigma(Z^{(1)})$

**end**

## ۴.۲ نحوه یادگیری (فاز ۲ پروژه)

در اصل خروجی که این شبکه باید به ما می‌داد باید به ماتریس زیر نزدیک باشد:

$$\hat{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (5)$$

برای اینکه این مسئله را حل کنیم نیاز است که بدانیم چقدر با جواب فاصله داریم پس  $y - \hat{y}$  را محاسبه می‌کنیم تا این اختلاف را پیدا کنیم حال بر اساس خروجی این معادله فاصله  $w, b$  ها را پیدا کرده و از جواب قبلی کم می‌کنیم و مقدار جدید را بدست می‌آوریم و اینکار را برای تمام لایه‌ها انجام می‌دهیم.

تا به الان ما با نحوه عملکرد این مدل آشنا شدیم، حال می‌خواهیم با نحوه یادگیری این مدل آشنا شویم.

همانطور که در بخش ۳.۱ گفتیم هر گره دارای مقدار بین ۰ تا ۱ است که این مقدار توسط ورودی ما مشخص می‌شود و ما نمی‌توانیم برای یادگیری مدل از این گره‌ها استفاده کنیم ولی مقدار  $w_{i,j}$  و  $b_i^{(0)}$  ها به ازای هر ورودی ثابت می‌ماند. پس می‌توانیم برای یادگیری این مقادیر را تغییر دهیم، ولی چگونه؟

زمانی که برنامه برای اولین بار اجرا می‌شود مقدار تمام متغیرهای  $b, w$  های تمام لایه‌ها بصورت تصادفی انتخاب می‌شوند. فرض کنید که به ازای ورودی عدد ۱ خروجی ما بر اساس این مقادیر تصادفی ماتریس زیر شد و چون ورودی ما عدد ۱ بود می‌بینیم که جواب اشتباه است.

$$y = \begin{pmatrix} 0.2134 \\ 0.13353 \\ 0.7234 \\ \vdots \\ 0.6675 \end{pmatrix} \quad (4)$$

---

### Algorithm 3: code for model with one hidden layer

---

```

OutPut  $\leftarrow A^{(1)}(y)$ 
desireOutput  $\leftarrow$  True output of
picture ( $\hat{y}$ )

/* start Backpropagation phase
*/

dZ(1)  $\leftarrow$  Output - desireOutput
dW(1)  $\leftarrow$  dZ(1) * A(0)T
db(1)  $\leftarrow$  mean(dZ(1))
dZ(0)  $\leftarrow$  (W(1)T * dZ(1)) *  $\sigma'(Z^{(0)})$ 

dW(0)  $\leftarrow$  dZ(0) * XT
db(0)  $\leftarrow$  mean(dZ(0))
/* Update parameters */

W(1)  $\leftarrow$  W(1) - dW(1)
b(1)  $\leftarrow$  b(1) - db(1)
W(0)  $\leftarrow$  W(0) - dW(0)
b(0)  $\leftarrow$  b(0) - db(0)

```

---

برای اینکه مدل ما توانایی تشخیص تمام اعداد را داشته باشد نیاز است که تعداد زیادی عکس از اعداد مختلف را به آن بدهیم برای یادگیری و به ازای هر ورودی که می‌دهیم مدل ما دقیق تر میشود. برای کامل شدن الگوریتم، الگوریتم ۳ را درون حلقه الگوریتم ۲ قرار دهید. در صورت علاقه به یادگیری بیشتر میتوانید به رفرنس [۲] مراجعه کنید.

## References

- [1] Miroslav Román Rosón, Yannik Bauer, Ann H Kotkat, Philipp Berens, Thomas Euler, and Laura Busse. Mouse dlgn receives functional input from a diverse population of retinal ganglion cells with limited convergence. *Neuron*, 102(2):462–476, 2019.
- [2] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.