

## پروژه دوم – پیاده سازی شبکه عصبی با یک لایه مخفی

در پروژه قبل پیاده سازی شبکه لجستیک رگرسیون را آموختید. در این پروژه یک گام در شبکه عصبی جلو تر میرویم و شبکه عصبی را به همراه یک لایه مخفی ایجاد میکنیم.

### مرحله صفر: اضافه کردن کتابخانه های مورد نیاز

فایل های موجود در پوشه Requirements را در مسیر پروژه قرار دهید.

### مرحله اول: بارگذاری مجموعه دیتا

در این پروژه دیتاست به صورت یه مجموعه ای از ویژگی های  $X$  و  $Y$  است. این مجموعه داده، مجموعه ای از اطلاعات گل هاست که در دو کلاس دسته بندی شده است.

تمرین اول: ابعاد  $X$  و  $Y$  و تعداد داده های آموزشی ( $m$ ) را بدست آورید.

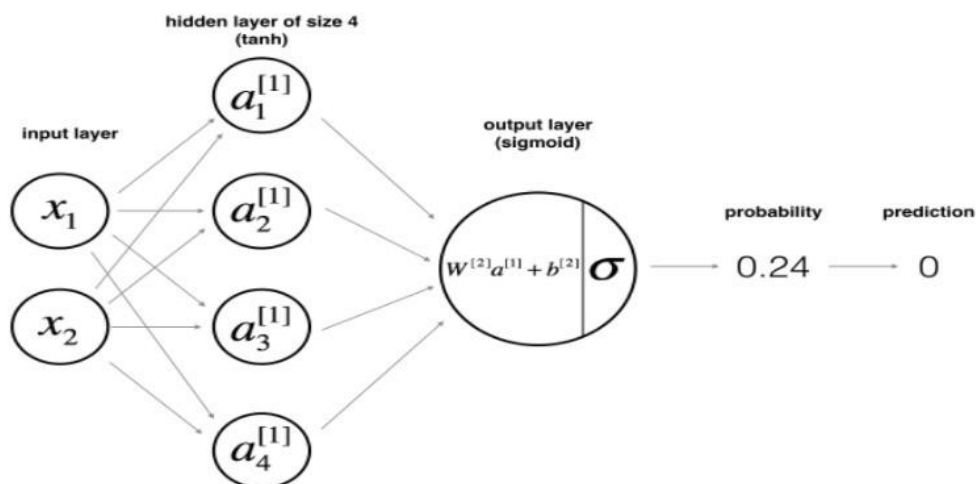
### مرحله دوم: مشاهده عملکرد لجستیک رگرسیون روی مجموعه داده ها

کد بخش لجستیک رگرسیون آورده شده است. کد آن را اجرا کنید و نتیجه دسته بندی را تفسیر کنید.

آیا دسته بندی به درستی انجام شده است؟

### مرحله سوم: استفاده از شبکه های عصبی

طبق آنچه که در ویدیو های درس مشاهده کردید ساختار شبکه عصبی به شکل زیر است:



یادآوری: ایجاد یک شبکه عصبی طی مراحل زیر انجام میشود:

- (۱) ساختار کلی تعریف میشود (پارامترهای ابعاد و تعداد نورون ها و لایه های مخفی)
- (۲) مقدار دهی اولیه وزن و بایاس انجام میشود
- (۳) انتشار رو به جلو
- (۴) محاسبه خطا
- (۵) پس انتشار
- (۶) آپدیت قوانین

تمرین دوم: ساختار کلی را تعریف کنید.  $n_x$  اندازه ورودی،  $n_h$  اندازه لایه مخفی (در اینجا ۴ در نظر بگیرید) و  $n_y$  اندازه خروجی است.

تمرین سوم: مقدار دهی اولیه را برای وزن ها با مقادیر تصادفی و برای بایاس ها با مقدار صفر انجام دهید. برای مقدار دهی تصادفی از تابع `np.random.randn()*0.01` کمک بگیرید.

تمرین چهارم: انتشار رو به جلو را با توابع فعالساز `tanh` برای `A1` و `sigmoid` برای `A2` پیاده سازی کنید.

تابع محاسبه هزینه را مطالعه کنید در مورد نحوه استفاده و کاربرد تابع `squeeze` تحقیق و بررسی کنید.

تمرین پنجم: تابع مربوط به پس انتشار را با توجه به فرمولهای زیر پیاده سازی و کامل کنید.

## Summary of gradient descent

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]}a^{[1]T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]})$$

$$dW^{[1]} = dz^{[1]}x^T$$

$$db^{[1]} = dz^{[1]}$$

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$$

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]'}(Z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$

Andrew Ng

تمرین ششم: به روز رسانی وزن ها را با توجه به فرمول های ویدیوها پیاده سازی و کامل کنید.

تمرین هفتم: تا اینجا اجزای شبکه عصبی را به صورت جدا جدا پیاده سازی کردید. در این تمرین تابع `nn_model` را با آنچه پیش از این پیاده سازی کردید تکمیل کنید.

تمرین هشتم: تابع پیش بینی را پیاده سازی کنید و حد آستانه پیش بینی کلاس ها را ۰/۵ در نظر بگیرید.

$$y_{prediction} = \mathbb{1}\{\text{activation} > 0.5\} = \begin{cases} 1 & \text{if } activation > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

مرحله چهارم: آموزش شبکه عصبی

در نهایت شبکه عصبی را آموزش دهید و نتایج دسته بندی را مشاهده و تفسیر کنید.