

یادگیری ماشین با پایتون

فرزاد حسینیان

دانشگاه محقق اردبیلی

دانشکده علوم کامپیوتر

آبان 98

جلسه ششم

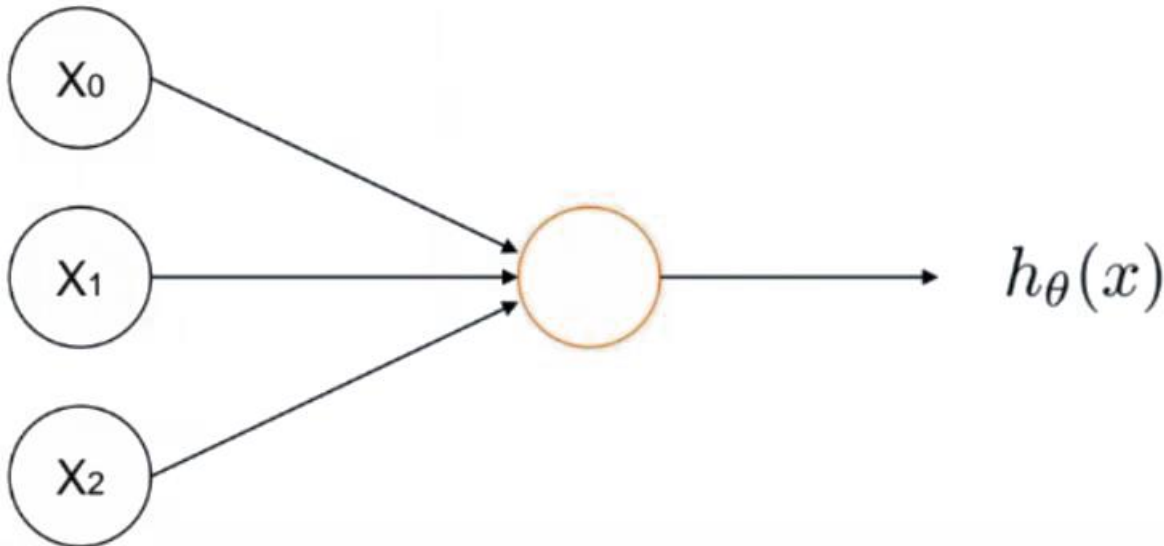
شبکه عصبی مصنوعی جلسه دوم

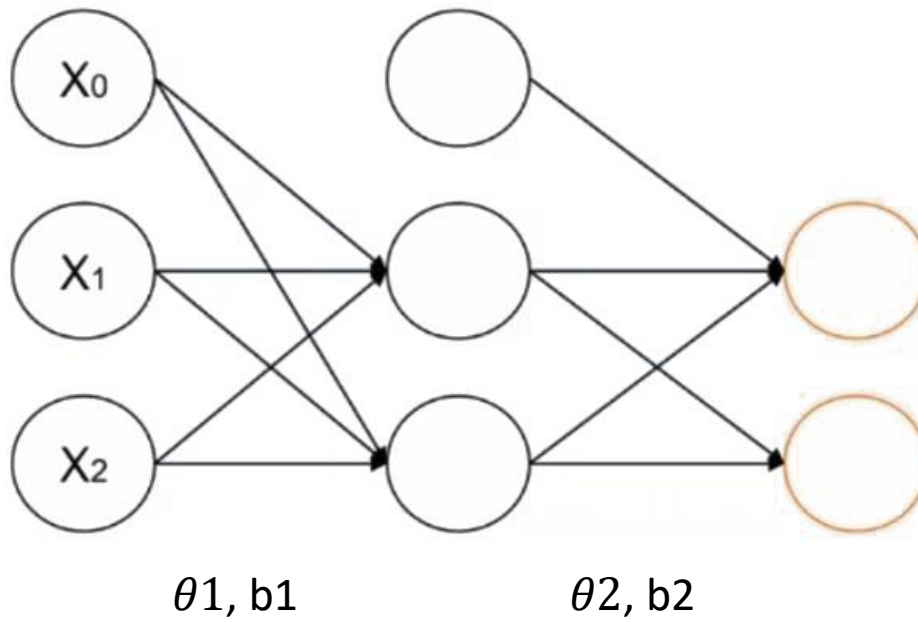
رند کار برای رگرسیون لجستیک

$$h_{\theta}(x) = g(\theta^T x)$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))$$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)}$$





هدف شبکه عصبی مصنوعی: پیدا کردن وزن ها

a = output of last layer

$$e = \vec{y} - \vec{a}$$

$$J(\theta) = e^T e$$

$$e = \vec{y} - \vec{a} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 - a_1 \\ y_2 - a_2 \\ \vdots \\ y_n - a_n \end{bmatrix}$$

$$\begin{aligned} e^T e &= (y_1 - a_1)^2 + (y_2 - a_2)^2 + \dots + (y_n - a_n)^2 \\ &= \sum_{i=1}^n (y_i - a_i)^2 \end{aligned}$$

قبلی:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J$$

حالا:

$$\text{repeat until } \frac{\partial J}{\partial W^{layer}_{ij}} \rightarrow 0:$$

$$\{ W^{layer}_{ij} := W^{layer}_{ij} - \alpha \frac{\partial J}{\partial W^{layer}_{ij}}$$

آبدیت باياس به طول
مستقل از وزن هاست

$$W = W - \alpha \frac{\partial J}{\partial W}$$

$$b = b - \alpha \frac{\partial J}{\partial b}$$

الگوریتم گرادیان کاهش - قاعده زنجیره ای

$$z_i^{(k)} = \sum_{j=1}^{\#^{(k-1)}} \theta_{i,j}^{(k)} a_j^{(k-1)} + b_i^{(k)}$$

↔
Input to layer (k)

$$\frac{\partial}{\partial \theta_{i,j}^{(k)}} J(\Theta) = \underbrace{\frac{\partial J(\Theta)}{\partial z_i^{(k)}}}_{s_i^{(k)}} \underbrace{\frac{\partial z_i^{(k)}}{\partial \theta_{i,j}^{(k)}}}_{a_j^{(k-1)}}$$

$$\frac{\partial}{\partial b_i^{(k)}} J(\Theta) = \underbrace{\frac{\partial J(\Theta)}{\partial z_i^{(k)}}}_{s_i^{(k)}} \underbrace{\frac{\partial z_i^{(k)}}{\partial b_i^{(k)}}}_{1}$$

$s_i^{(k)}$ ≡ sensitivity of
ith neuron in layer
(k)

الگوریتم گرادیان کاهشی - قاعده زنجیره ای - پارامتر جدید

$$\frac{\partial}{\partial \theta_{i,j}^{(k)}} J(\Theta) = s_i^{(k)} * a_j^{(k-1)}$$

$$\frac{\partial}{\partial b_i^{(k)}} J(\Theta) = s_i^{(k)}$$

$$s_i^{(k)} = \frac{\partial J}{\partial z_i^{(k)}}$$

$$s_i^{(k)} = \frac{\partial J}{\partial z_i^{(k)}}$$

How to compute these?

$$s^{(k)} = \begin{bmatrix} s_1^{(k)} \\ s_2^{(k)} \\ \vdots \\ s_{\#}^{(k)} \end{bmatrix}$$

$$\dot{F}^{(k)}(z^{(k)}) = \begin{bmatrix} \dot{f}^{(k)}(z_1^{(k)}) & 0 & \dots & 0 \\ 0 & \dot{f}^{(k)}(z_2^{(k)}) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dot{f}^{(k)}(z_{\#^{(k)}}^{(k)}) \end{bmatrix}$$

$$\dot{F}^{(k)}(z^{(k)}) = \begin{bmatrix} \dot{f}^{(k)}(z_1^{(k)}) & 0 & \dots & 0 \\ 0 & \dot{f}^{(k)}(z_2^{(k)}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^{(k)}(z_{\#^{(k)}}^{(k)}) \end{bmatrix}$$

Activation Function of layer (k)

No of neurons of the k^{th} layer

$$g(z) = \frac{1}{1+e^{-z}}$$

$$\frac{dg}{dz} = -e^{-z} \times (-1) \frac{1}{(1+e^{-z})^2} = \frac{e^{-z}}{(1+e^{-z})^2}$$

$$= \underbrace{\frac{e^{-z}}{1+e^{-z}}}_{1-g(z)} \times \underbrace{\frac{1}{1+e^{-z}}}_{g(z)} = (1-g(z))g(z)$$

فرمول بدست آوردن حساسیت لایه ها

$$s^{(k)} = \dot{F}^{(k)}(z^{(k)})(\theta^{(k+1)})^T s^{(k+1)}$$

حساسیت آخرین لایه (لایه خروجی)

$$s^{(M)} = -2\dot{F}^{(M)}(z^{(k)})(y - a)$$

Repeat until convergence:

$$\theta_{i,j}^{(k)} := \theta_{i,j}^{(k)} - \alpha s_i^{(k)} a_j^{(k-1)}$$

$$b_i^{(k)} := b_i^{(k)} - \alpha s_i^{(k)}$$

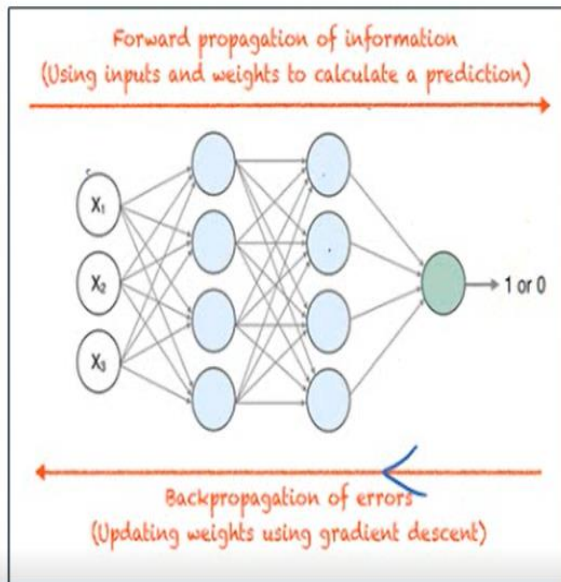
بردار ساز شده: Repeat until convergence:

$$\theta^{(k)} := \theta^{(k)} - \alpha s^{(k)} a^{(k-1)T}$$

$$b^{(k)} := b^{(k)} - \alpha s^{(k)}$$

الگوریتم پس انتشار خطا

روشی برای کاهش گرادیان کاهش جهت حداقل کردن کل مربعات خطای خروجی است.



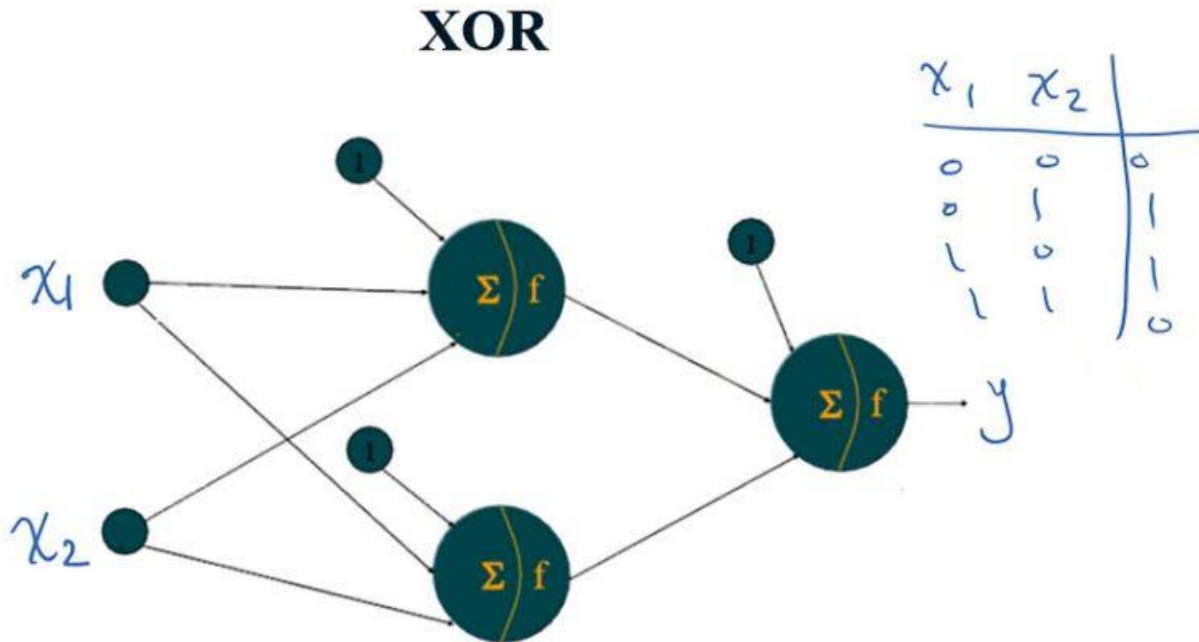
هنگامی که خطایی در خروجی شبکه‌های پیش‌خور چند لایه‌ای رخ می‌دهد، این خطا باید به سمت ورودی برگشت داده شود و باعث اصلاح وزن‌ها گردد.

۱- پیش‌خور کردن (feed-forward) ورودی

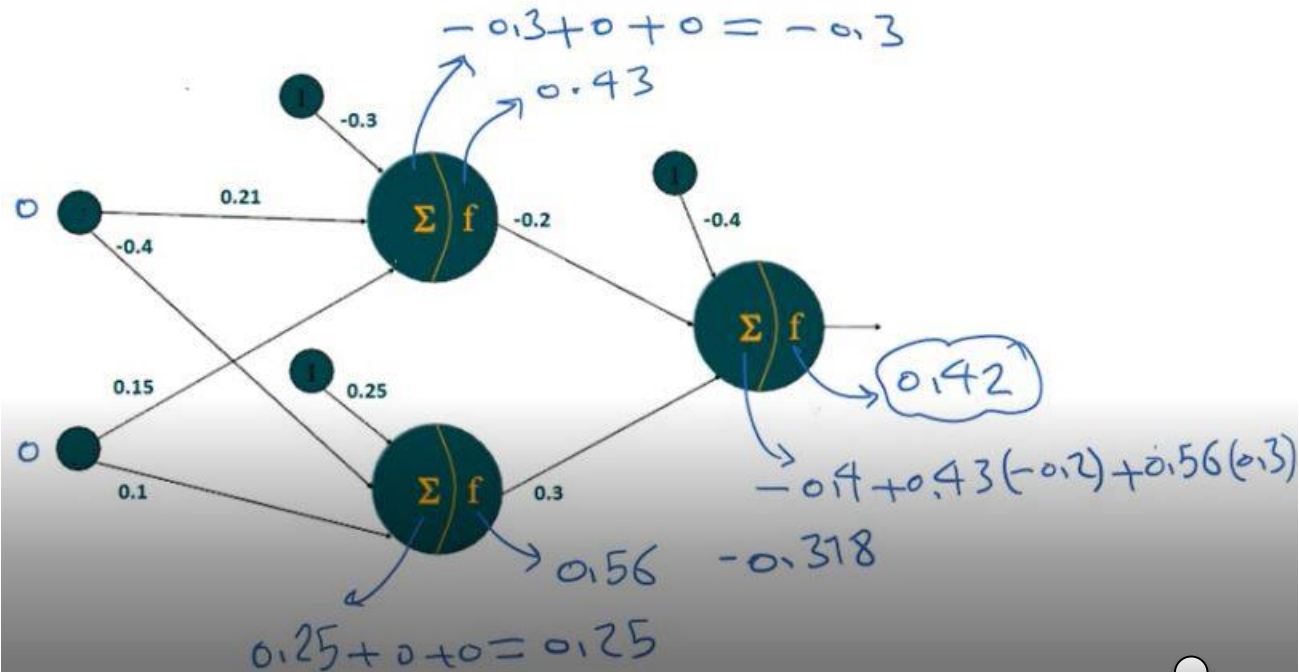
۲- محاسبه و پس انتشار خطای مربوطه

۳- تنظیم وزن‌ها

مثال



پیش خور کردن ورودی



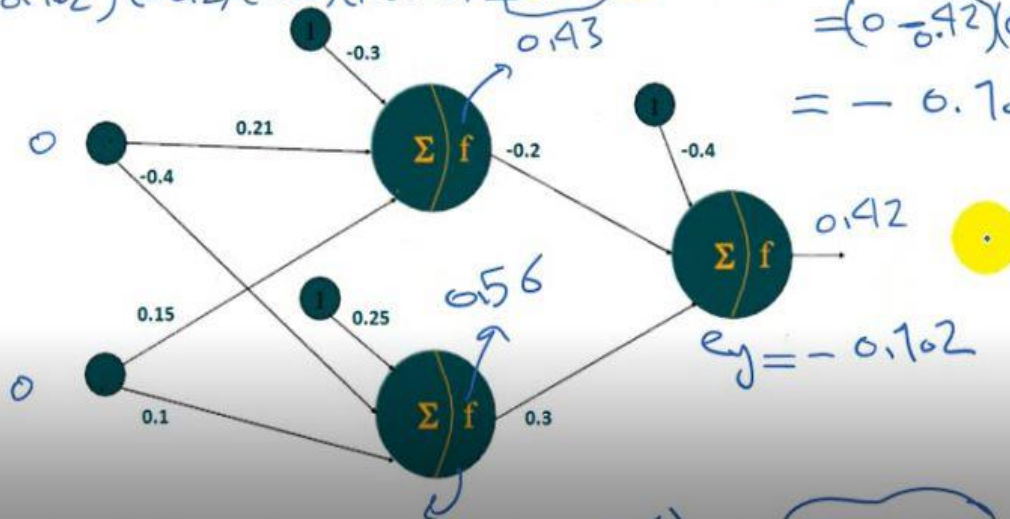
پس انتشار خطا

$$e_j = e_y \cdot w_{jy} \cdot h_j (1 - h_j)$$

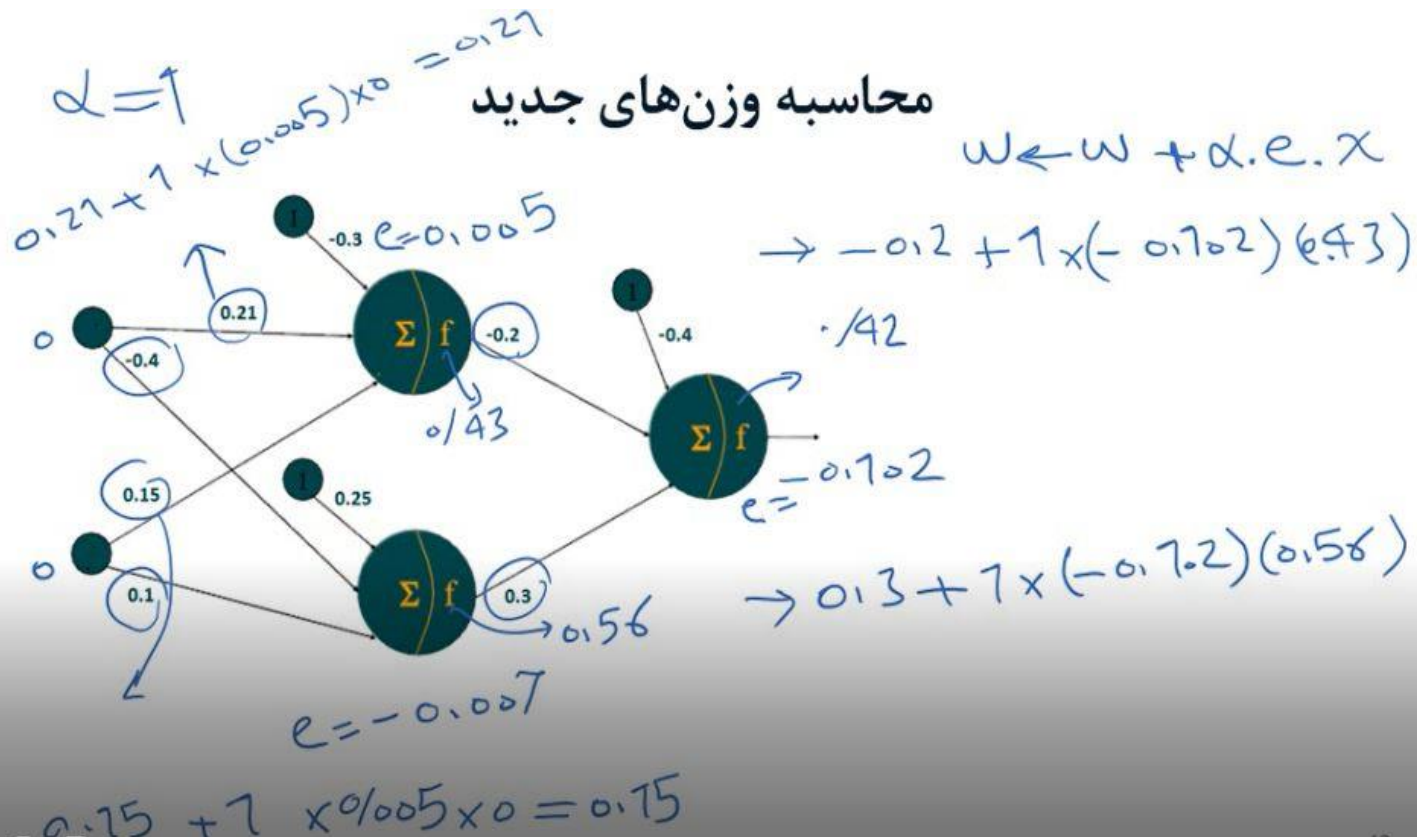
$$= (-0.102) (-0.2) (0.43) (1 - 0.43) = 0.005$$

$$e_y = (t - y) y (1 - y)$$

$$= (0 - 0.42) (0.42) (1 - 0.42) = -0.102$$



$$e = (-0.102) (0.3) (0.56) (1 - 0.56) = -0.007$$



محاسبه بایاس های جدید

$$b = b + \alpha e$$

$$-0.3 + 1(0.055)$$

$$-0.4 + 1 \times (-0.102)$$



$$0.25 + 1 \times (-0.007)$$