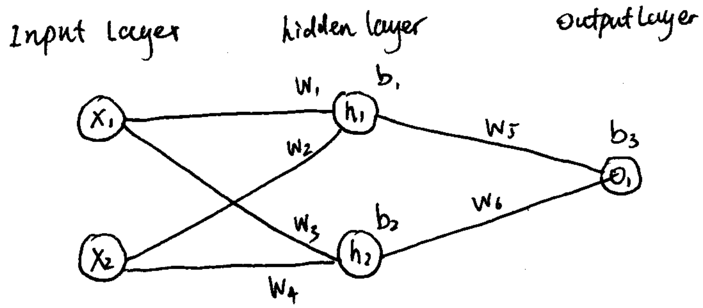
Implement a Neural Network From Scratch (Calc Process)

Provided by: Steve Yan (Ex10si0n) GitHub/Ex10si0n URL: www.aspires.cc

Network (Simplified)



$$MSE = \frac{\sum (y_{true} - y_{pred})^2}{n}$$

Minimize the loss

$$\underline{Loss} = J(\underline{w_1}, \underline{w_2}, \underline{w_3}, \underline{w_4}, \underline{w_5}, \underline{w_6}, \underline{b_1}, \underline{b_2}, \underline{b_3})$$

adjusting each params ......

by SGD

e.g. focus on w_1 brings Loss changes.

$\frac{\partial L}{\partial w_1}$ (need to determine).

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_1} \times \frac{\partial h_1}{\partial w_1}$$ (chain rule)

chain rule (proof).

consider $z = f(u,v)$ $u(x,y)$, $v(x,y)$ ($u(t), v(t)$ both derivable)

$z = f(u(t), v(t))$, then: $z = f(u(x,y), v(x,y))$

$$\frac{dz}{dx} = \frac{\frac{dz}{dt}}{\frac{dx}{dt}} \quad \Downarrow \quad \frac{dz}{dt} = \frac{dz}{dx} \cdot \frac{dx}{dt}.$$

$$\left\{ \begin{aligned} dz &= \frac{\partial f}{\partial u} du + \frac{\partial f}{\partial v} dv & \text{(total derivate)} \\ du &= \frac{\partial u}{\partial x} dx + \frac{\partial u}{\partial y} dy & \text{(total derivate)} \\ dv &= \frac{\partial v}{\partial x} dx + \frac{\partial v}{\partial y} dy & \text{(total derivate)} \end{aligned} \right.$$

→ see next page

$$dz = \frac{\partial f}{\partial u}\left(\frac{\partial u}{\partial x}dx + \frac{\partial u}{\partial y}dy\right) + \frac{\partial f}{\partial v}\left(\frac{\partial v}{\partial x}dx + \frac{\partial v}{\partial y}dy\right)$$

$$= \left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial x}\right)dx + \left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial y}\right)dy$$

$$\frac{\partial z}{\partial x} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial x} \qquad \frac{\partial z}{\partial y} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial f}{\partial v} \cdot \frac{\partial v}{\partial y}$$

⇒ chain rule.

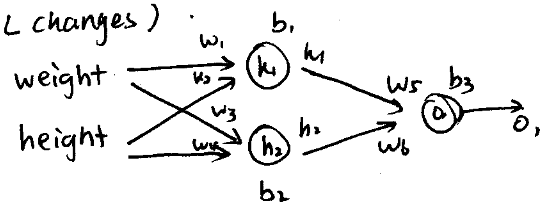
Lagrange's notation: (拉格朗日式)

$h'(x) = f'(g(x))g'(x)$ / $h' = (f \circ g)' = (f' \circ g) \cdot g'$

Leibniz's notation: (莱布尼茨式)

$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$ ✓

→ determine: (Focus on w_1 change makes L changes)



$$\frac{\partial L}{\partial w_1} = \underbrace{\frac{\partial L}{\partial y_{pred}}}_{①} \times \underbrace{\frac{\partial y_{pred}}{\partial h_1}}_{②} \times \underbrace{\frac{\partial h_1}{\partial w_1}}_{③}$$

① $$\frac{\partial L}{\partial y_{pred}} = \frac{\partial MSE}{\partial y_{pred}} = \frac{\partial\left[\frac{\Sigma(y_i - y_{pred})}{n}\right]}{\partial y_{pred}} = \frac{\partial\left[\frac{\Sigma(y_i - sigmoid(w_5h_1+w_6h_2+b_3))}{n}\right]}{\partial\, sigmoid(w_5h_1+w_6h_2+b_3)}$$

② $$\frac{\partial y_{pred}}{\partial h_1} = \frac{\partial\, sigmoid(w_5h_1+w_6h_2+b_3)}{\partial h_1} = w_5\, sigmoid'(w_5h_1+w_6h_2+b_3)$$

③ $$\frac{\partial h_1}{\partial w_1} = \frac{\partial\, sigmoid(w_1x_1+w_2x_2+b_1)}{\partial w_1} = x_1\, sigmoid'(w_1x_1+w_2x_2+b_1)$$

Compound function Derivation Law

$$\left(\frac{u}{v}\right)' = \frac{u'v-uv'}{v^2}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

$$sigmoid'(x) = \frac{(1+e^{-x})'}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) = sigmoid(x)\cdot[1-sigmoid(x)]$$

$\Rightarrow$ deriv_sigmoid(x) $\equiv$ sigmoid(x)[1-sigmoid(x)]

(in the code)

$$sigmoid'(x) = sigmoid(x)\,[1-sigmoid(x)]$$

Total derivate Relationship (cont. last pg)

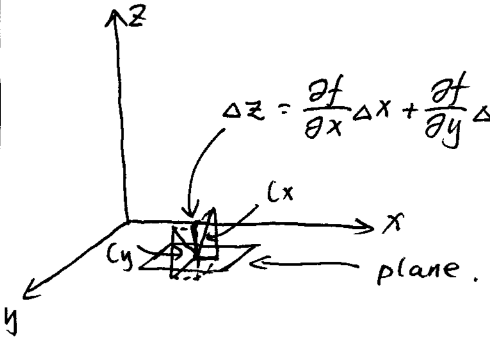
$$dz = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy$$ ← CN: 全微分关系

let:

$z = f(x,y)$ is smooth on (x_0, y_0)

then for a plane (平面) is:

$$z = C_0 + C_x(x-x_0) + C_y(y-y_0)$$



when $x = x_0$, $y = y_0$

$$C_0 = f(x_0, y_0) \Rightarrow C_x = \frac{\partial f}{\partial x}, \quad C_y = \frac{\partial f}{\partial y}$$ (consider C's as slopes)

let $\Delta x = x - x_0$ $\Delta y = y - y_0$ $\Delta z = z - C_0$

$$\Delta z = \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y$$

$$\lim_{\Delta x,y,z\to 0}(\Delta z) = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy.$$
$$= dz$$

then. $dz = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy$.

⇒ Neuron o1 :

$$\frac{\partial y_{pred}}{\partial w_5} = h_1 \cdot sigmoid'(w_5 h_1 + w_6 h_2 + b_3)$$

$$\frac{\partial y_{pred}}{\partial w_6} = h_2 \cdot sigmoid'(w_5 h_1 + w_6 h_2 + b_3)$$

$$\frac{\partial y_{pred}}{\partial b_3} = sigmoid'(w_5 h_1 + w_6 h_2 + b_3)$$

⇒ Neuron h1 :

$$\frac{\partial h_1}{\partial w_1} = x_1 \cdot sigmoid'(w_1 x_1 + w_2 x_2 + b_1)$$

$$\frac{\partial h_1}{\partial w_2} = x_2 \cdot sigmoid'(w_1 x_1 + w_2 x_2 + b_1)$$

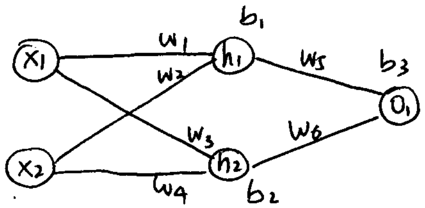
$$\frac{\partial h_1}{\partial b_1} = sigmoid'(w_1 x_1 + w_2 x_2 + b_1)$$

⇒ Neuron h2 :

$$\frac{\partial h_1}{\partial w_3} = x_1 \cdot sigmoid'(w_3 x_1 + w_4 x_2 + b_2)$$

$$\frac{\partial h_1}{\partial w_4} = x_2 \cdot sigmoid'(w_3 x_1 + w_4 x_2 + b_2)$$

$$\frac{\partial h_1}{\partial b_2} = sigmoid'(w_3 x_1 + w_4 x_2 + b_2)$$



$$\frac{\partial y_{pred}}{\partial h_1} = w_5 \cdot sigmoid'(w_5 h_1 + w_6 h_2 + b_3)$$

$$\frac{\partial y_{pred}}{\partial h_2} = w_6 \cdot sigmoid'(w_5 h_1 + w_6 h_2 + b_3)$$

Gradient descend (in back-prop)

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$$ (θ represents w_x, b_x) .

$$w_1 \leftarrow w_1 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_1} \times \frac{\partial h_1}{\partial w_1}$$

$$w_2 \leftarrow w_2 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_1} \times \frac{\partial h_1}{\partial w_2}$$

$$w_3 \leftarrow w_3 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_2} \times \frac{\partial h_2}{\partial w_3}$$

$$w_4 \leftarrow w_4 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_2} \times \frac{\partial h_2}{\partial w_4}$$

$$w_5 \leftarrow w_5 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial w_5}$$

$$w_6 \leftarrow w_6 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial w_5}$$

Feed forward

$$h_1 = sigmoid(x_1 \cdot w_1 + x_2 \cdot w_2 + b_1)$$

$$h_2 = sigmoid(x_1 \cdot w_3 + x_2 \cdot w_4 + b_2)$$

$$o_1 = sigmoid(h_1' \cdot w_5 + h_2 \cdot w_6 + b_3)$$

$\uparrow$
y_{pred}

Gradient descend (cont.)

updates biases. .

$$b_1 \leftarrow b_1 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_1} \times \frac{\partial h_1}{\partial b_1}$$

$$b_2 \leftarrow b_2 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\partial h_2} \times \frac{\partial h_2}{\partial b_2}$$

$$b_3 \leftarrow b_3 - \eta \frac{\partial L}{\partial y_{pred}} \times \frac{\partial y_{pred}}{\theta b_3}$$

Process of training a Neural Network

① Feed forward (x_1 x_2) → y_{pred}

② MSE (y_{pred})

③ Back propagation (model, y_{pred} , y_{true} , x_1 x_2)
(w1+b1)

until convergence