28 | 784 Pixels | training image.
28

$0 \le$ each pixel ≤ 255
black … white

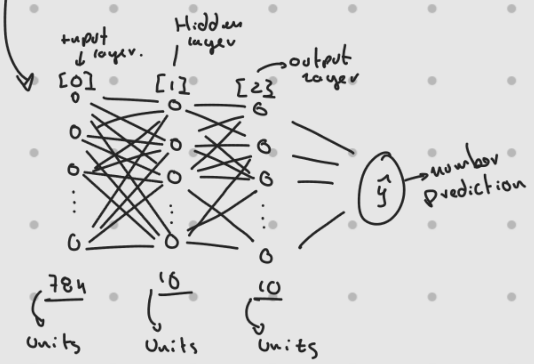
$\longrightarrow X = \begin{bmatrix} — x^1 — \\ — x^2 — \\ \vdots \\ — x^m — \end{bmatrix}^{T} = \begin{bmatrix} | & | & & | \\ x^1 & x^2 & \dots & x^m \\ | & | & & | \end{bmatrix}$ } 784 Rows.

Example (image)

784 colums → Pixels

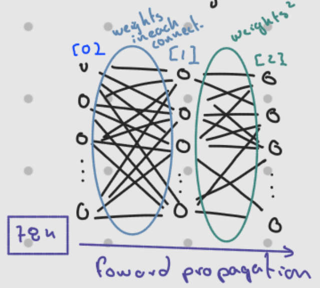
Goal:

784 (28 x 28) $\Longrightarrow$ prediction. $\underbrace{0, 1, 2, \dots, 9}_{10 \text{ classes}}$



step 1: Forward Propagation.

Take img and run it through network. and get Results.



Input Layer

① $A^{[0]} = X \quad [784 \times m]$

1st UnActivated Layer

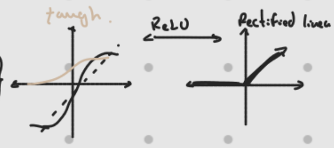
② $Z^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$ (weight, bias)
10×784, $784 \times m$, $10 \times 1 \Rightarrow 10 \times m$

step 2: Act. function

③ $A^{[1]} = g(Z^{[1]}) = \text{ReLU}(Z^{[1]})$

without it would only has linear connections → Linear Regre

ReLU(x) $\begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \le 0 \end{cases}$



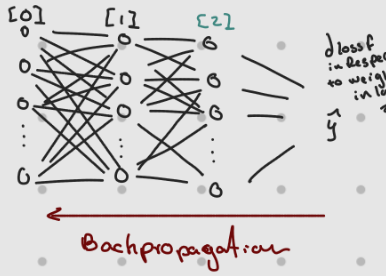
2nd UnActivated Layer.

UnAct. 2 layer ④ $Z^{[2]} = W^{[2]} A^{[0]} + b^{[1]}$
10×10, $10 \times m$, $10 \times 1 \Rightarrow 10 \times m$

⑤ $A^{[2]} = \text{Softmax}(Z^{[2]})$ → to have prob.

input layer $\begin{bmatrix} 1.3 \\ 5.1 \\ \vdots \\ 1.1 \end{bmatrix} \rightarrow \boxed{\dfrac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}} \rightarrow$ Probability $\begin{bmatrix} 0.02 \\ 0.90 \\ \vdots \\ 0.5 \end{bmatrix}$

But, how do we get good weights And biases? Backpropagation



error 2nd layer

Preds $\begin{bmatrix} 0.1 \\ 0.2 \\ 0.5 \\ 0.03 \\ \vdots \\ 0.1 \end{bmatrix}$

One-hot if Y = 4 $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ correct label

⑥ $dZ^{[2]} = A^{[2]} - Y$
$10 \times m$, $10 \times m$, $10 \times m$

d loss f in Respect to weight in layer 2

⑦ $dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$
10×10, $10 \times m$, $m \times 10$

⑧ $db^{[2]} = \frac{1}{m} \sum dZ^{[2]}$ → Avg. Abs error
10×1, 10×1

1st Layer.

⑨ $dZ^{[1]} = W^{[2]T} dZ^{[2]} \cdot g'(Z^{[1]})$
$10 \times m$, 10×10, $10 \times m$
in Reverse weights; error 2n; d Act. funct, to undo Act. funct

⑩ $dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$

⑪ $db^{[1]} = \frac{1}{m} \sum dZ^{[1]}$
10×1, 10×1

modify Params (weight/bias)

$W^{[1]} = W^{[1]} - \alpha dW^{[1]}$ (learning Rate)
$b^{[1]} = b^{[1]} - \alpha db^{[1]}$
$W^{[2]} = W^{[2]} - \alpha dW^{[2]}$
$b^{[2]} = b^{[2]} - \alpha db^{[2]}$
→ hyperparameter

Cycle.

Return to forward propagation