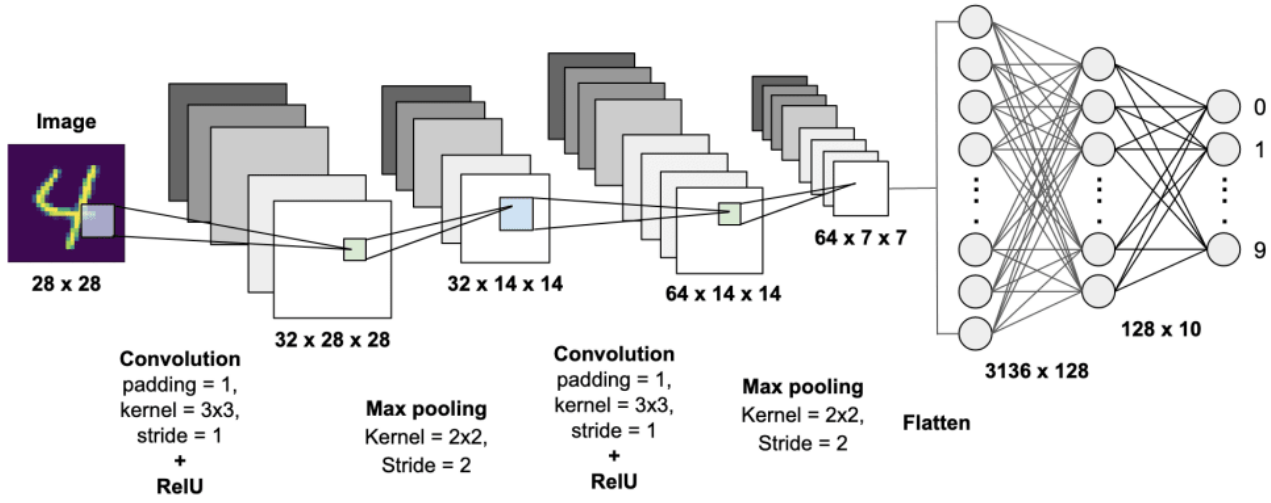
[open AI]

[Torch]

What is CNN

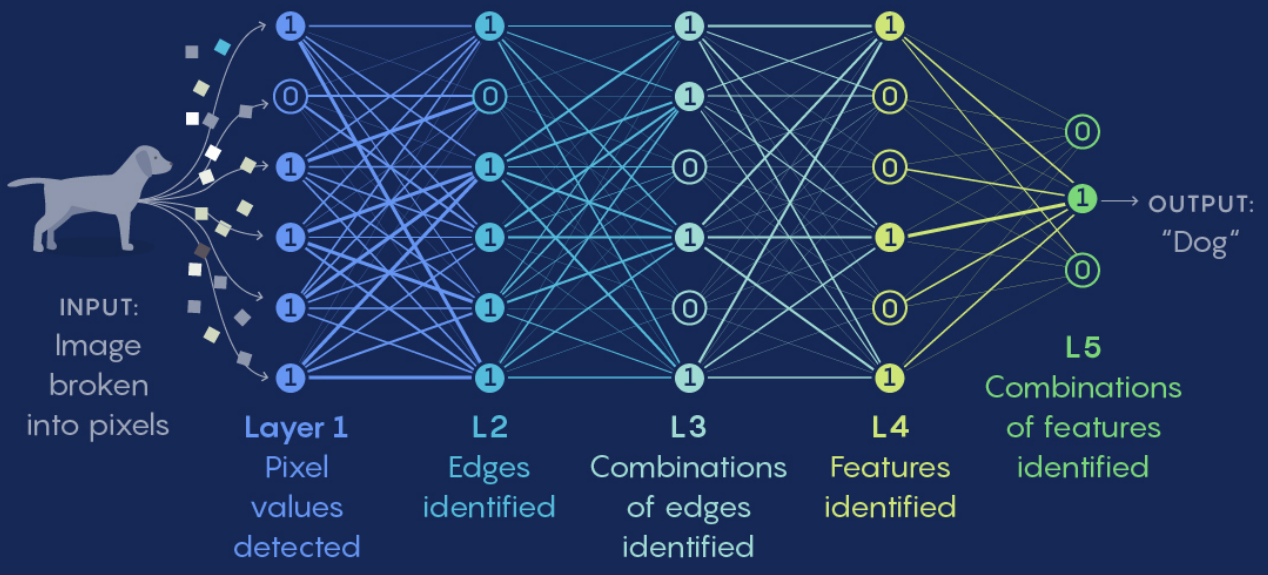
- CNN convolutional -neural network



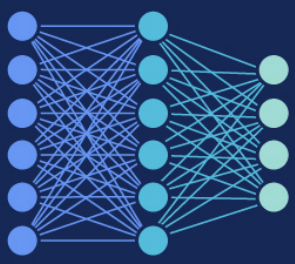
- neural network

How to Design a Neural Network

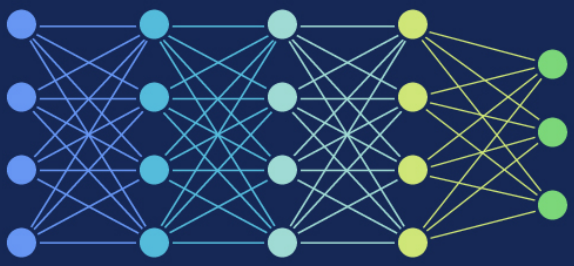
Neural networks pass an input, like an image, through multiple layers of digital neurons. Each layer reveals additional features of the input. Mathematicians are revealing how a network's architecture — how many neurons and layers it has and how they're connected — determines the kinds of tasks that the neural network will be good at.



When data is fed into a network, each artificial neuron that fires (labeled "1") transmits signals to certain neurons in the next layer, which are likely to fire if multiple signals are received. This process reveals abstract information about the input.



A **SHALLOW NETWORK** has few layers but many neurons per layer. These "expressive" networks are computationally intensive.



A **DEEP NETWORK** has many layers and relatively few neurons per layer. It can achieve high levels of abstraction using relatively few neurons.

MNIST

plugin to Q Learning

- Q learning

$$Q^* : State \times Action \to \mathbb{R} \tag{7}$$

Reward take from prevoius experience, recent experience make bigger effect.

discount :γ

$$R_{t_0} = \sum_{t=t_0}^{\infty} \gamma^{t-t_0} r_t \tag{2}$$

The π fuction get best rewards from Learnig exprience

in this state this action get best result

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\ Q^*(s,a) \tag{3}$$

new `Q Value= Bellman`

$Q^{\pi} = reward + discount\cdot$ Best act for next state

$$Q^{\pi}(s,a) = r + \gamma Q^{\pi}(s', \pi(s')) \tag{4}$$

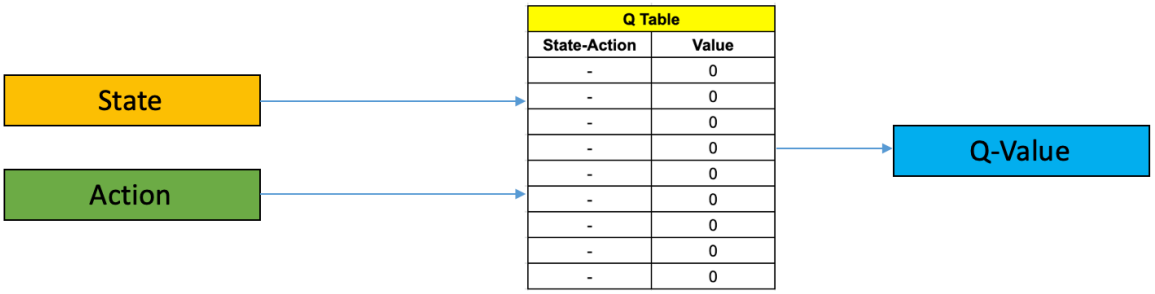
temporal difference:

the value need to make from current to best state

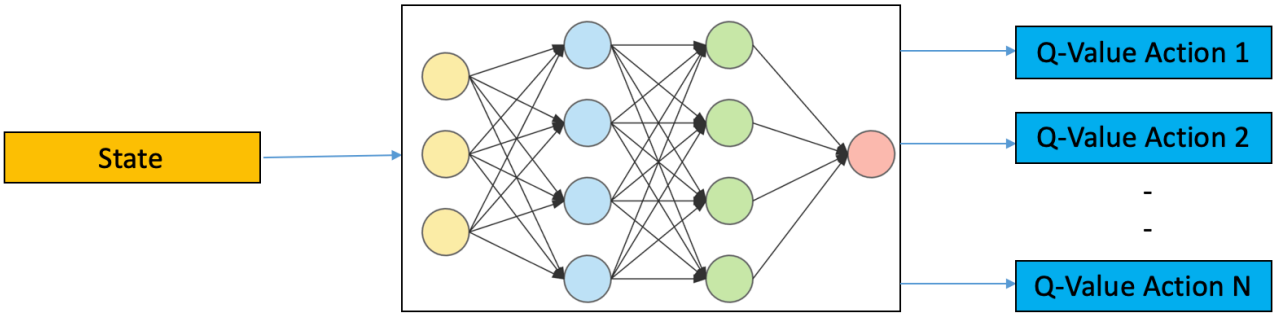
$$\delta = Q(s,a) - (r + \gamma \max_a Q(s_{t+1}, a)) \tag{5}$$

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{new value (temporal difference target)}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \Bigg)}^{\text{temporal difference}}$$

$$Q^{new} = Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot Q^{\pi} - Q(s_t, a_t)) \tag{6}$$



Q Learning



Deep Q Learning

After Q learn from percitular State(image) and relation ship between action and result

i can update itself.