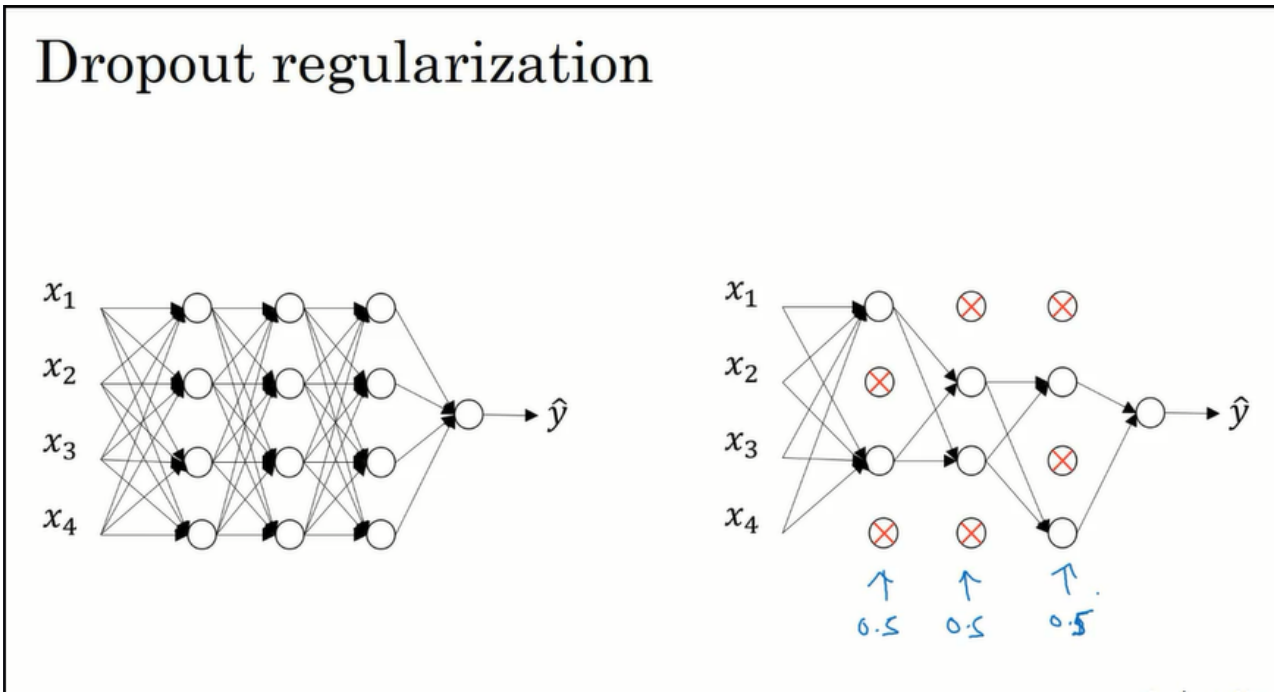
Dropout regularization

- 

- $Implementing\ dropout(\text{" } Inverted\ dropout \text{ "})$
  - *illustrate with layer l=3*
  - $d3 = np.random.rand(a3.shape[0], a3.shape[1]) < keep\_prob,$ $keep\_prob = 0.8$
  - $a3 = np.multiply(a3, d3)$
  - $a3 /= keep\_prob$
    - *Example*
    - $50\ units \rightarrow 10\ units\ shut\ off$
    - $z^{[4]} = w^{[4]} \underbrace{a^{[3]}}_{reduced\ by\ 20\%} + b^{[4]}$
    - $avoid\ getting\ smaller: /= 0.8$

Making predictions at test time

- **No dropout**

Why does dropout work

- Intuition: Can't rely on any one feature, so have to spread out weights
- Alternative: some layers apply dropout, some don't or varies *keep_prob*
- **Downside**: cost function no longer well-defined
  1. turn off dropout, plot cost function J to make sure J is decreasing

2. turn on dropout, to see if it works