

# DROPOUT

- randomly (and temporarily) deletes half of the hidden layer neurons in the network
  - do not touch the input/output neurons
- Forward and backpropagate through the network, and update the weights and biases like normal
- repeat the process by restoring the deleted neurons, and then randomly deleting more neurons
- When this process is repeated, the network will learn weights and biases for when only half of the hidden neurons are visible
  - so when we run the full network at the end, we can half the weights to compensate
- Dropout is kind of like averaging the effects of many neural networks, which prevents overfitting.
  - Theoretically, the “different” networks will overfit in different ways, so averaging the outputs will reduce overfitting in general
- Since neurons are randomly dropping out, each neuron learns to be independent, and not rely too much on the output of other neurons
  - Networks overfit when neurons are too co-dependent, and it reduces their ability to generalize
  - Dropout also tends to reduce weights, slight regularization
- Uses a KEEP\_PROP parameter, which has the probability that a neuron will be dropped in one iteration