Preventing Overfitting

- After each epoch, find the accuracy of the model on the validation data, which has labels. If the accuracy has plateaued, you can stop training.
    - We use validation data accuracy b/c the model is training on train data, so it has never seen the val data before
    - If it memorizes and overfits for train data, then accuracy may keep increasing, even though it is not learning general features.
- Have LOTS OF TRAINING DATA
- Reduce network size
- REGULARIZATION
    - See regularization notes
- Weights initialization
    - But why not random gaussian?
        - Example:
        - 1000 input neurons, half start with weight of 0, half start with weight of 1 (simple example)
        - First hidden layer has just one neuron(for example)
        - Weighted sum output will have 501 terms, since only the neurons with weight > 0 contribute, and plus the 1 bias term
        - The weight sum is distributed as a gaussian curve with mean zero and standard deviation sqrt(501)
        - This means that it is highly likely that |z| can be very large, which will saturate the neuron after it goes through the sigmoid
            - Remember that is x value is too big above 1, then the sigmoid function will always output ~1, and gradients can't flow since slope there is ~0
    - What's a better way?

- Initialize weights as gaussian, but with mean 0 and standard deviation 1 / sqrt(n) where n is the number of inputs that the neuron has.
    - In the earlier example, the standard deviation of the weighted sum will be sqrt(3/2), which is small, so curve has a sharp peak
- This will speed up learning, but probably won't increase final accuracy