Regularization

- $J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(y^i, y^i) + \frac{\lambda}{2m}|w|^2$
- λ is regularization parameter (hyper parameter to be tuned)

- L2 regularization
- $J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(y^i, y^i) + \frac{\lambda}{2m}|w|^2$

- L1 regularization -> make w sparse -> model compressing
- $J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(y^i, y^i) + \frac{\lambda}{2m}|w|$

Regularization back-propagation

- $dw = (before\ backprop) + \frac{\lambda}{m} w$
- $w = w\ - \alpha dw$
- $w = \mathrm{w} - \alpha \left(before\ backprop + \frac{\lambda}{m} w \right)$
- Make w small (called weight decay)

Why regularization reduces overfitting

- Make λ big, make some w ~ 0.
- Make contribution from individual neuron small.
- Big network -> logistic regression
- With good λ, high variance -> high bias

Dropout regularization

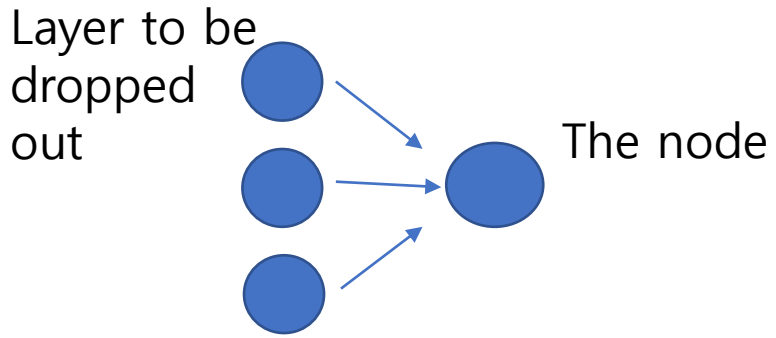
- Give probability to remove a node
- Node to be removed will be random.
- Node removal should be performed training sample by sample.
- No dropout in test

Inverted dropout

- Ex) l = 3 keep. Prob = 0.8
- d3 = np.random.rand(a3.shape[0], a3.shape[1]) < keep. Prob
- a3 = np.multiply(a3,d3)

- This process will remove about 20% of output.
- $z^4 = w^4 a^3 + b^4$ -> a^3 will be reduced by 20%
- z^4 expectation also be reduced by 20%.
- So, (a3 = a3/keep. Prob) to maintain the z^4 expectation value.

Understanding dropout



- Dropout remove nodes randomly. the node cannot rely on any one feature.

->Spreading of weights

- Do not use dropout in not overfitted data

- But the cost function J is not well defined. First, you need to check the J decrease, and then use dropout.

Other regularization methods

- Data augmentation

- Early stopping

-> Stop iterations when J of test sample starts increase.

Other regularization methods

- Early stopping has a problem.

- Progress of machine learning
  - First, optimize cost function J (momentum , Adam, RMSprop)
  - Avoid overfit (regularization, augmentation)

  - Optimizing cost function -> Find w, b -> J small
  - Reduce overfitting is other issue.

  orthogonalization

  - Early stopping mix two tasks. (stop J being smaller)
  - Ng says L2 regularization can be a option to not use early stopping.