Regularization

ISTD 50.035

Computer Vision

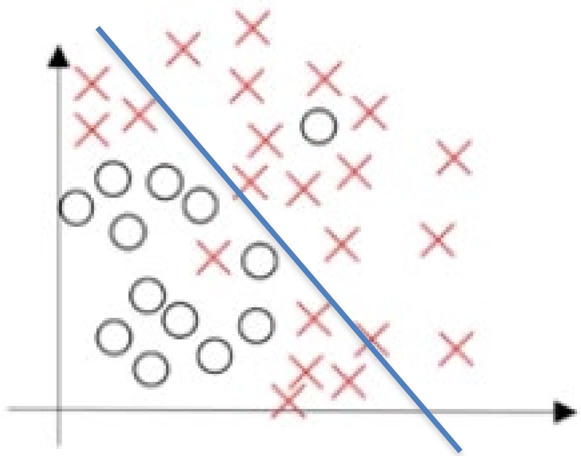
Acknowledgement: Some images are from various sources: UCF, Stanford cs231n, etc.

Overfitting

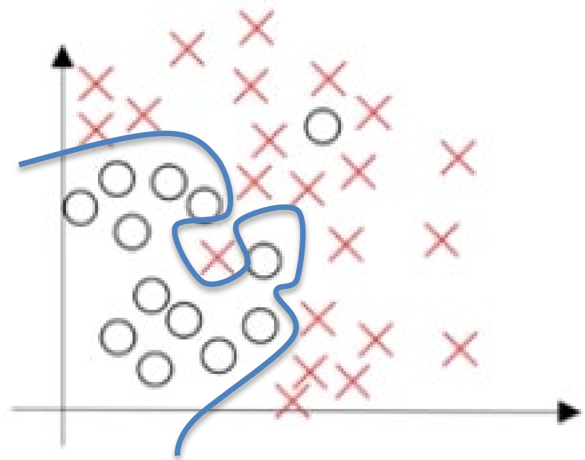
- Goal of model training: learn pattern from training data, generalize to new data of similar distribution

- Overfitting
  - Fit almost perfectly training data (small training error)
  - Perform poorly on new data (large validation error)
  - Learn specific pattern and noise of the training data, unable to extract the general / essential pattern

Overfitting



Underfitting



Overfitting

High bias: the method is unable to capture true pattern, large discrepancy
High variance: large difference in fits between datasets (train and test)

Regularization

- To reduce overfitting

- Regularization
  - Add a penalty to reduce the freedom of the model
  - Less likely to fit the noise
  - Improve generalization ability of the model

- Occam’s razor / Law of parsimony:
  - Simpler solutions are more likely to be correct than complex one

Regularization

$$L = \frac{1}{N} \sum_i L_i + \frac{\lambda}{2N} \|W\|_2^2$$

Sum of square

via back prop

$$\frac{\partial L}{\partial w_l} = \frac{1}{N} \sum_i \frac{\partial L_i}{\partial w_l} + \frac{\lambda}{N} w_l$$

$$w_l' = w_l - \gamma \frac{\partial L}{\partial w_l}$$

Dropout

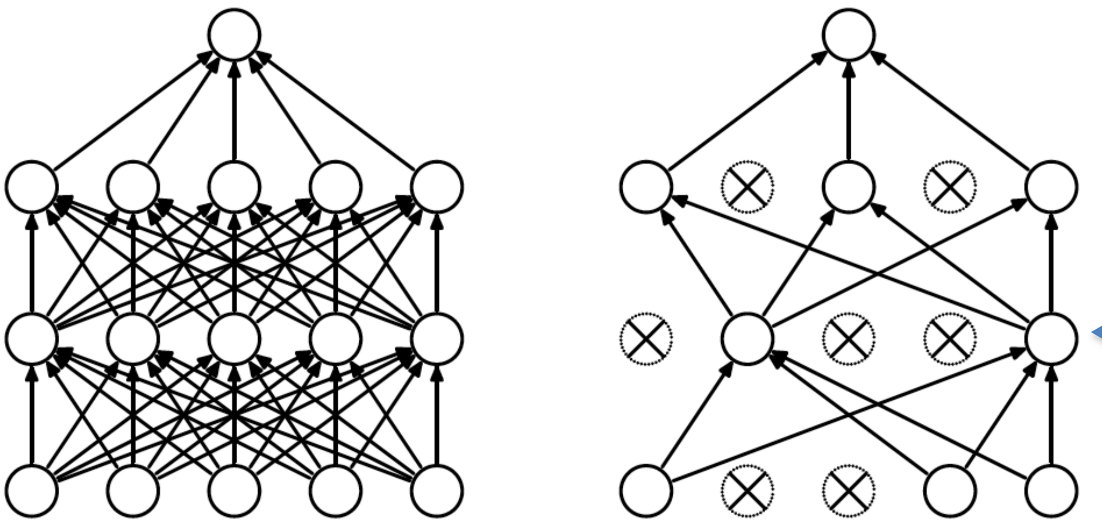
- DNN: a large number of parameters (freedom), prone to overfitting

- Dropout: during training, randomly ignore neurons in the network

- Each hidden unit is set to 0 with some probability (e.g. 0.5)
  – Forward pass: no contribution to downstream neurons
  – Backward pass: no weight update

- No dropout during testing

Dropout



(a) Standard Neural Net (b) After applying dropout.

Each layer: Probability p of keeping units

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

[Srivastava et al. 2014]

Dropout

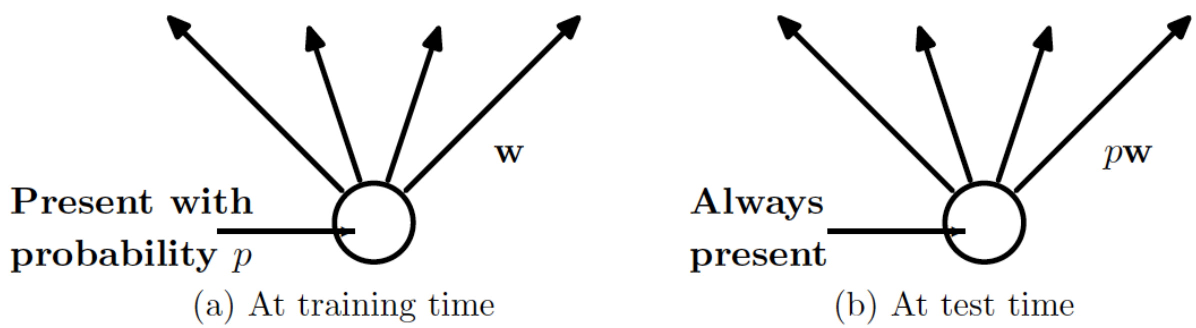


Figure 2: **Left**: A unit at training time that is present with probability p and is connected to units in the next layer with weights $\mathbf{w}$. **Right**: At test time, the unit is always present and the weights are multiplied by p. The output at test time is same as the expected output at training time.

[Srivastava et al. 2014]

Dropout

- Prevent co-adaption: hidden units work together to detect some complicated features specific to the training data
- With dropout, since some hidden units may not be available during training of some samples, cannot co-adapt
- Force hidden units to learn simpler and generally useful feature

Dropout

- Training with dropout: sampling "thinned" network
- A neural net with n units can be seen as a collection of 2^n possible thinned neural networks
- These networks all share weights
- Training a neural network with dropout can be seen as training a collection of 2^n thinned networks with extensive weight sharing
- At test time: approximately average the predictions from exponentially many thinned models