Νευρο-Ασαφής Υπολογιστική
Neuro-Fuzzy Computing

Διδάσκων –

Δημήτριος Κατσαρός

@ Τμ. ΗΜΜΥ

Πανεπιστήμιο Θεσσαλίας

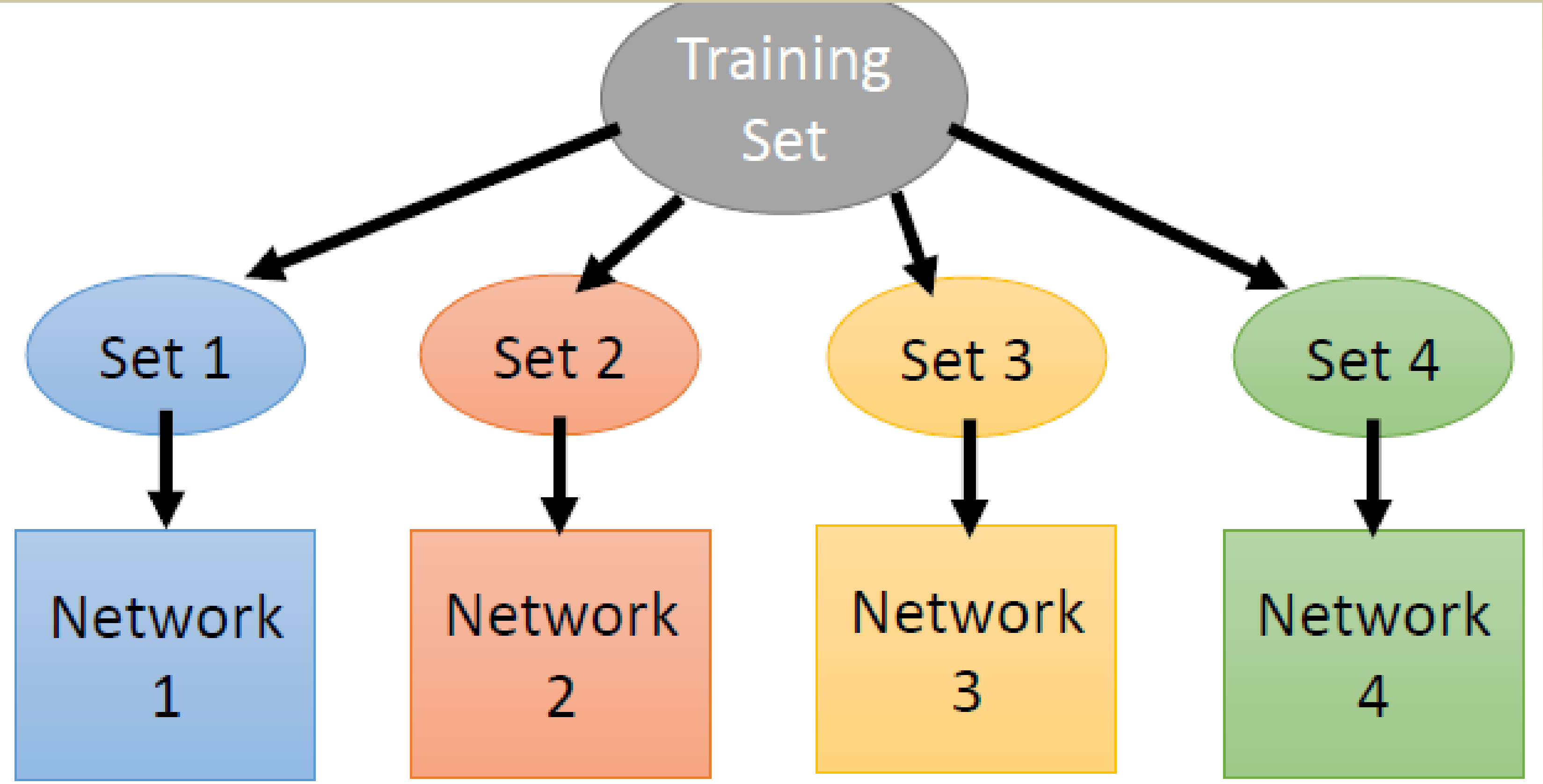
Διάλεξη 09η: Συμπληρωματική

Generalization for MLP: Dropout

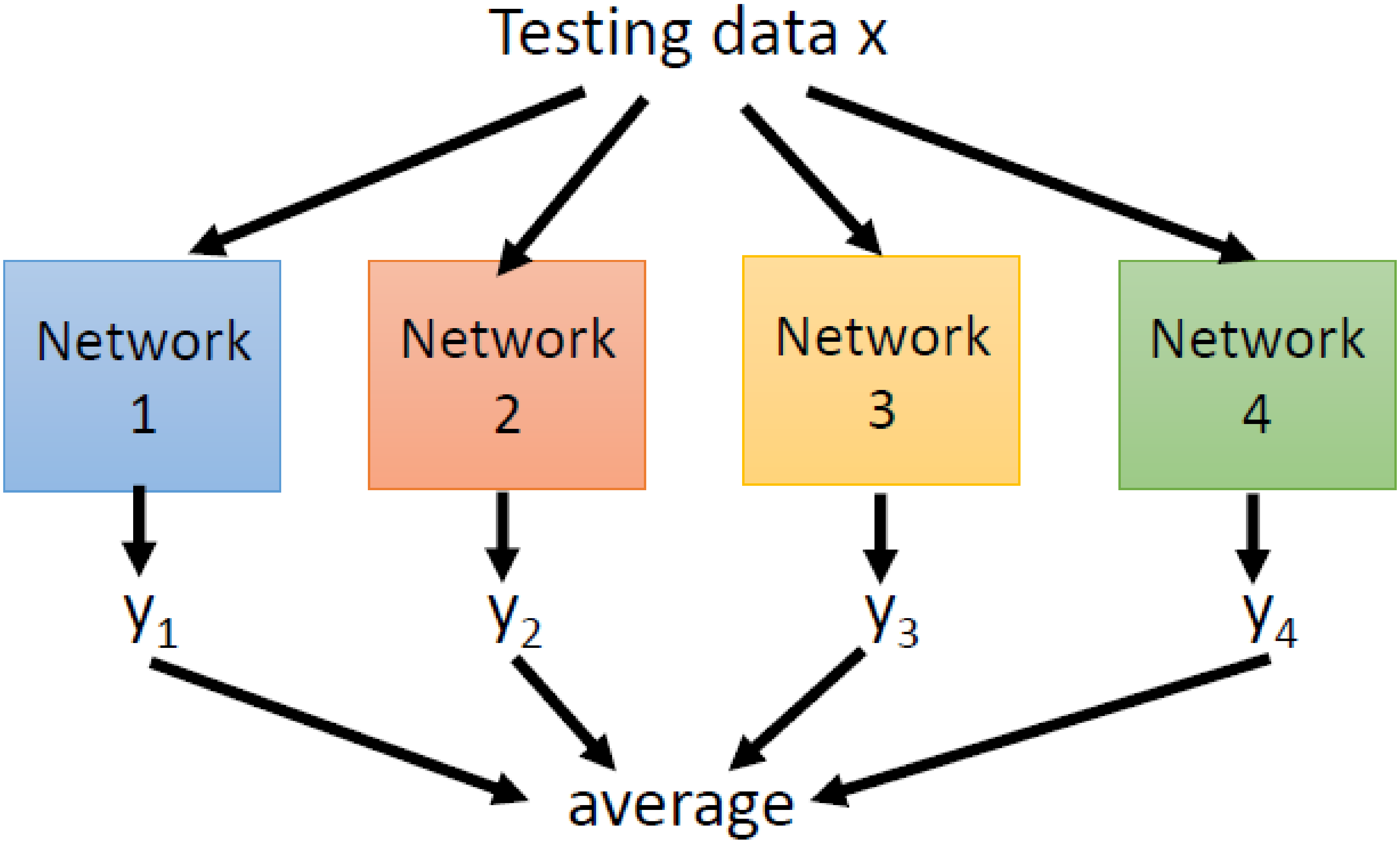
Bagging: Bootstrap aggregating

Ensemble

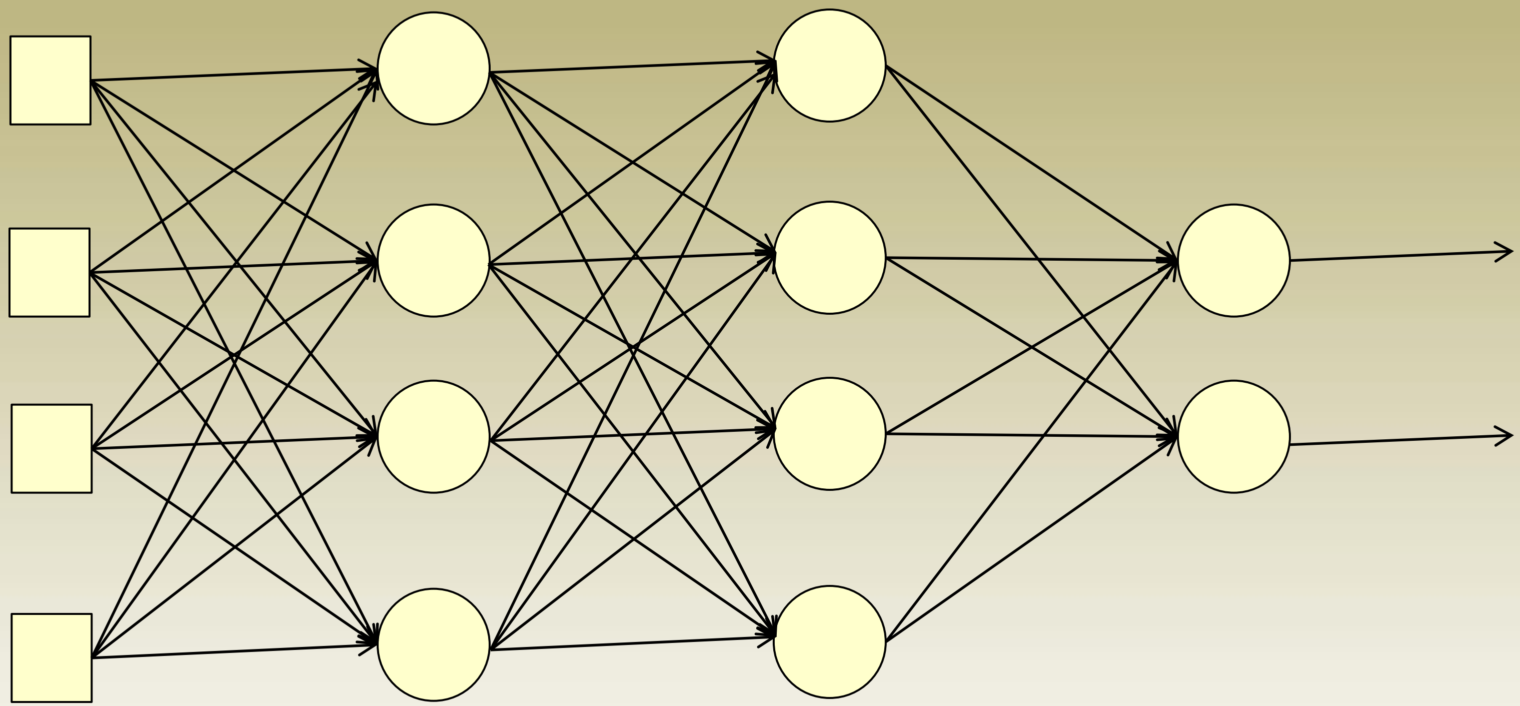
Train a bunch of networks with different structures



Bagging: Bootstrap aggregating

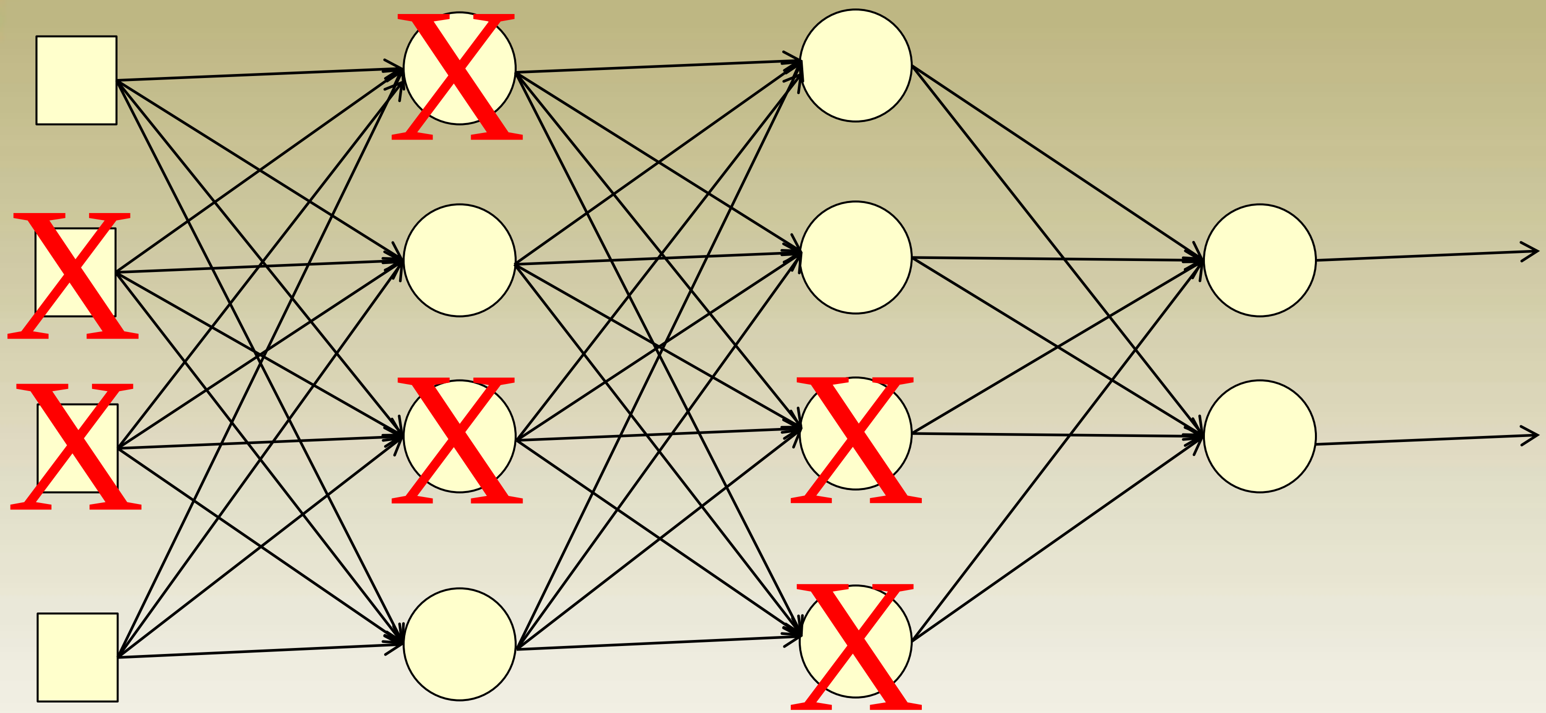


Dropout



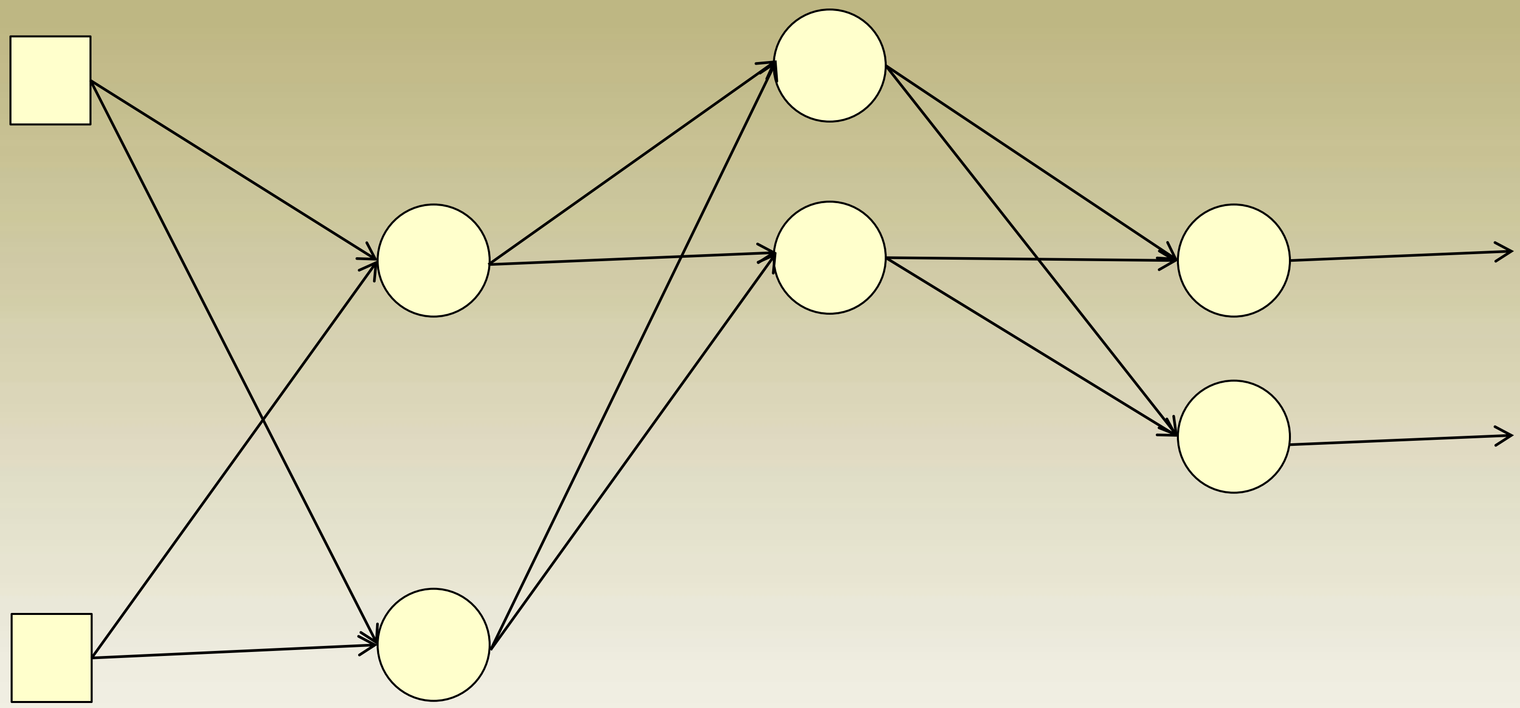
- **Each time before forward and backward phase**
  - Each neuron has p% to dropout

Dropout: Some neurons are eliminated



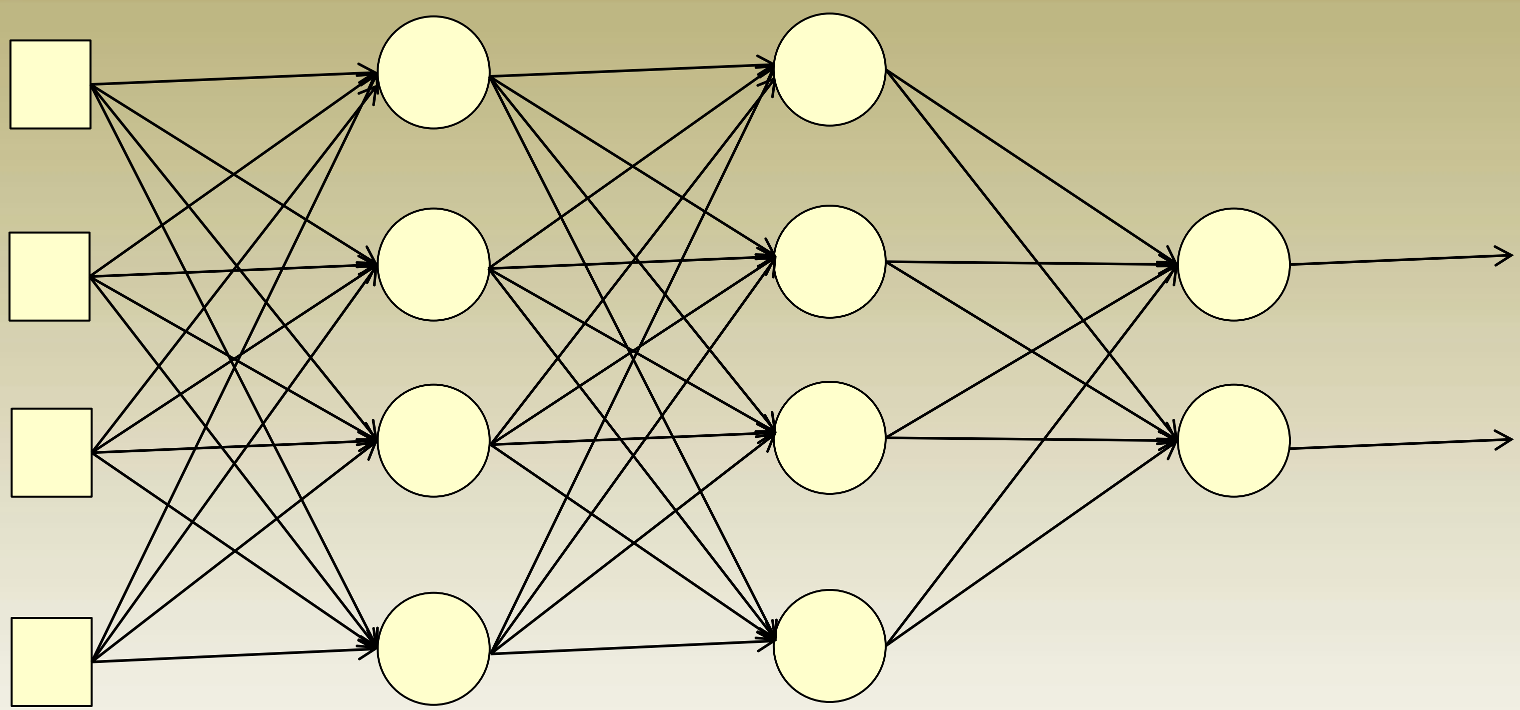
- The marked neurons (and their associated connections) are deleted

Dropout: Changed topology



- **The structure of the network is changed**
  - Using the new network for training

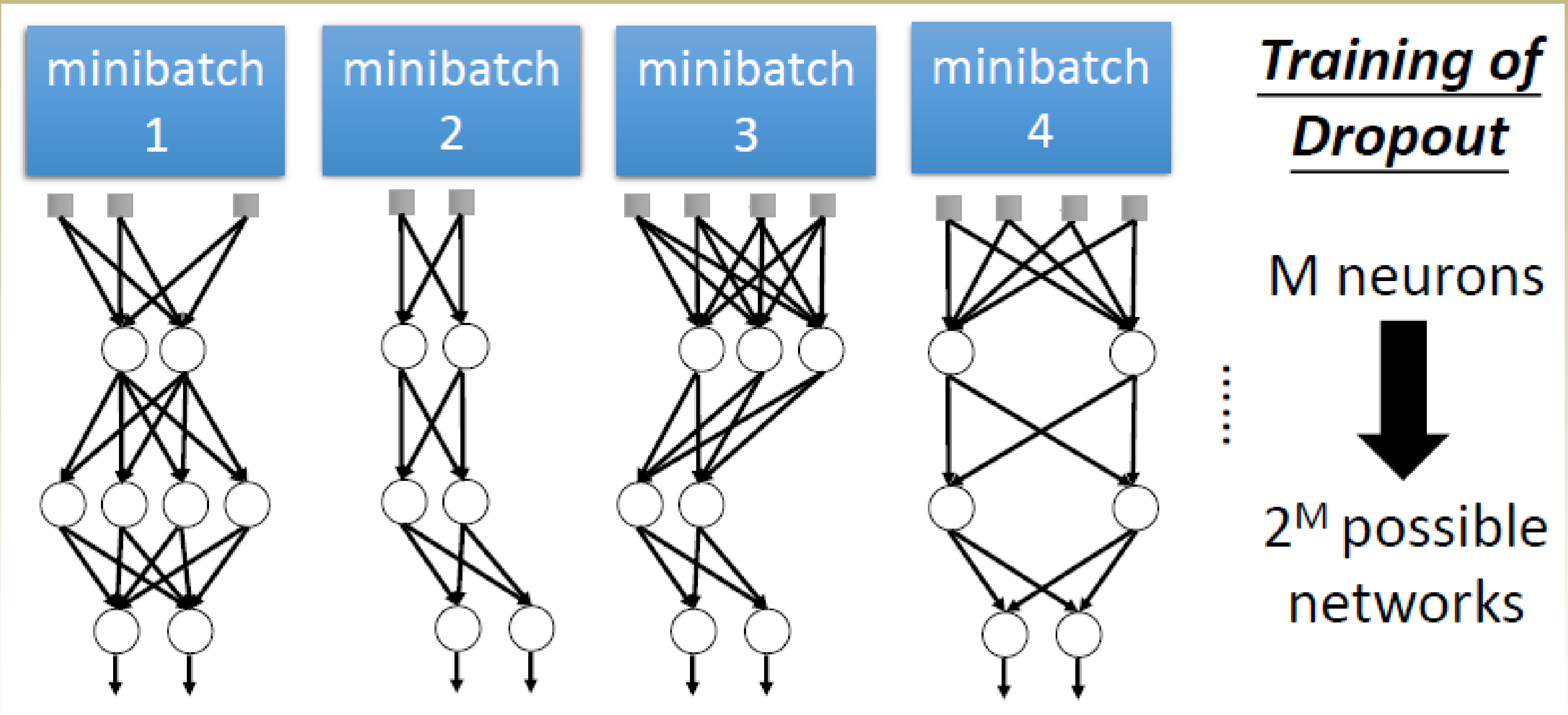
Dropout: Testing



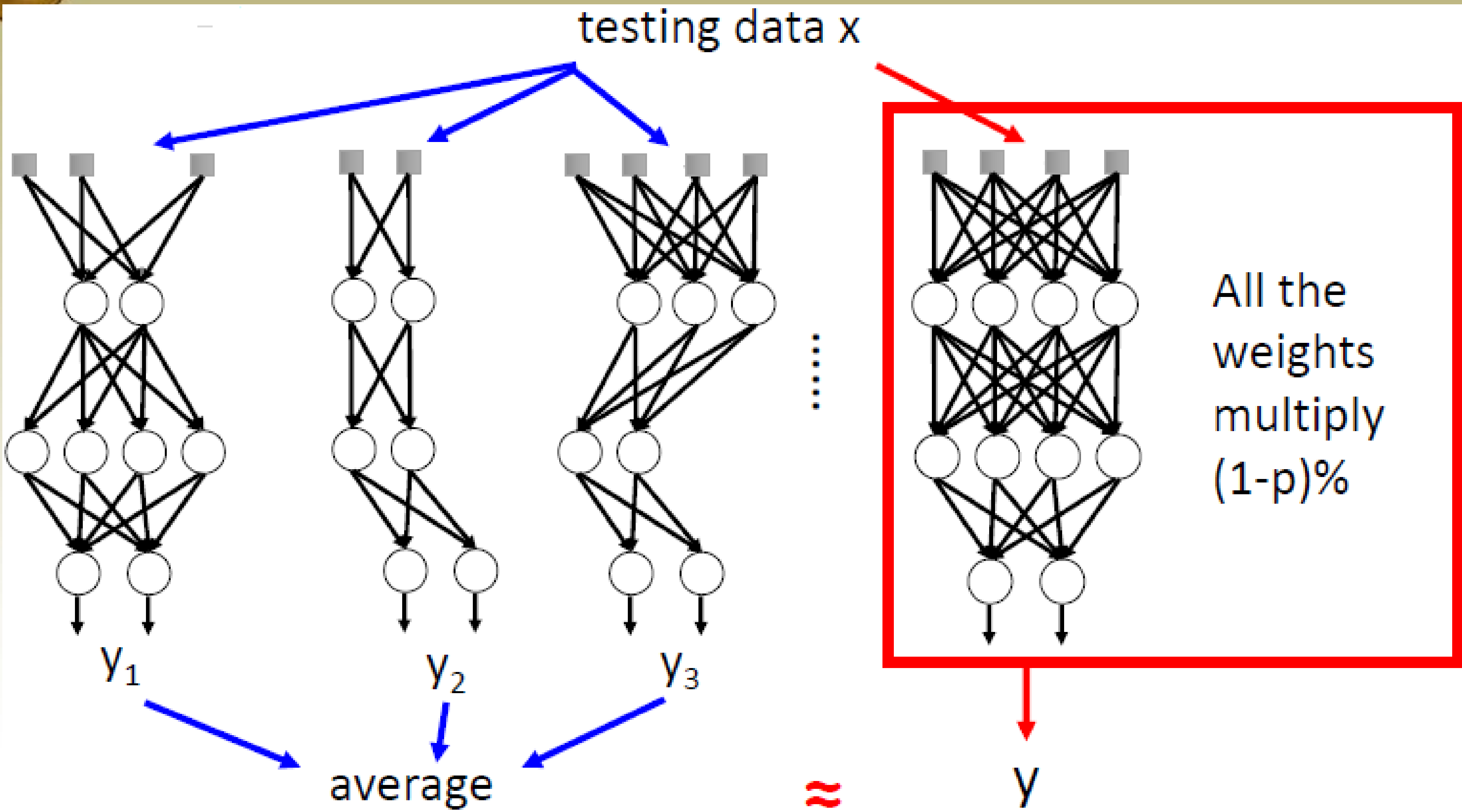
No dropout

- If the dropout rate at training is p%, all the weights times (1-p)%
- Assume that the dropout rate is 50%. If a weight w=1 by training, set w=0.5 for testing

Dropout: A kind of ensemble



Dropout: A kind of ensemble



Dropout for a single linear unit

A unit computing a weighted sum of n inputs of the form:

$$S = S(I) = \sum_{i=1}^{n} w_i I_i$$

where $I=(I_1,I_2,\ldots,I_n)$ is the input vector.

If we delete inputs with a uniform distribution over all possible subsets of inputs, or equivalently with a probability q=0.5 of deletion, then there are 2^n possible networks, including the empty network. For a fixed I, the average output over all these networks can be written as:

$$E(S) = \frac{1}{2^n} \sum_{\mathcal{N}} S(\mathcal{N}, I)$$

where N is used to index all possible sub-networks, i.e., all possible edge deletions. In this simple case, deletion of input units or of edges are the same thing.

Dropout for a single linear unit

This sum can be expanded using networks of size 0,1,2,…,n in the form:

$$E(S) = \frac{1}{2^n}\left[0 + \left(\sum_{i=1}^{n} w_i I_i\right) + \left(\sum_{1\le i<j\le n} w_i I_i + w_j I_j\right) + \dots\right]$$

In this expansion, the term $w_i I_i$ occurs

$$1 + \binom{n-1}{1} + \binom{n-2}{2} + \cdots + \binom{n-1}{n-1} = 2^{n-1}$$

times. So, finally the average output is:

$$E(S) = \frac{2^{n-1}}{2^n}\left(\sum_{i=1}^{n} w_i I_i\right) = \sum_{i=1}^{n} \frac{w_i}{2} I_i$$

Thus in the case of a single linear unit, for any fixed input I the output obtained by halving all the weights is equal to the arithmetic mean of the outputs produced by all the possible sub-networks. This combinatorial approach can be applied to other cases (e.g., p != 0.5)