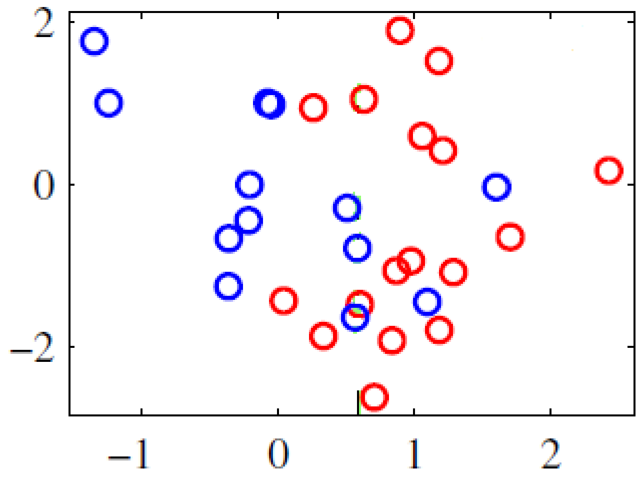
Boosting, Neural Nets and Learning Theory

- Building up intuition -

Recitation by Yang Xu

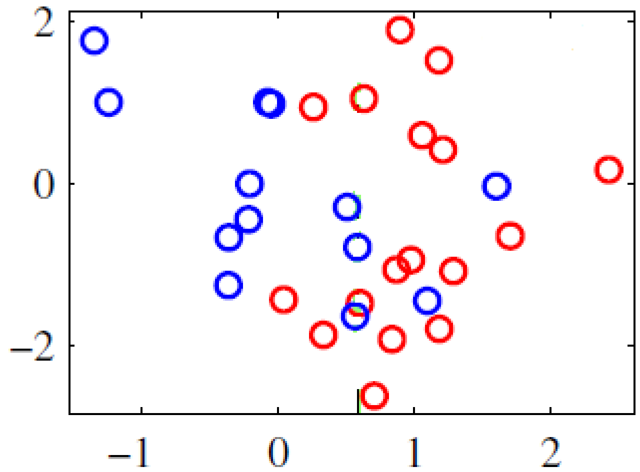
Puzzle 1

- How would you classify such data? Why does it seem hard to be solved perfectly?



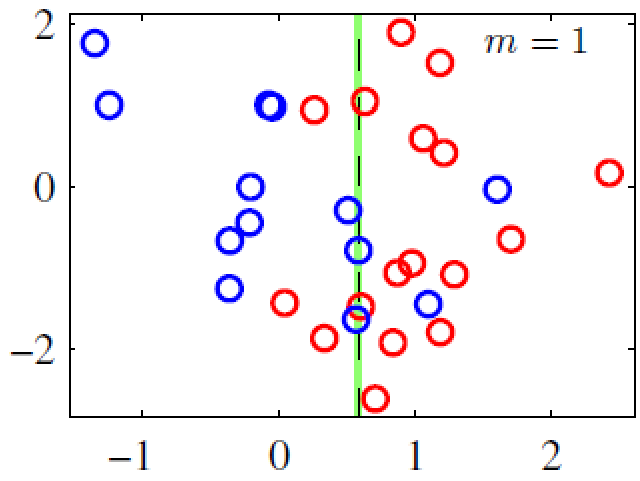
Boosting based on decision stumps

- At each iteration, greedily choose the dimension that minimizes errors based on thresholding.
- Q: which dimension would you choose at iter1?



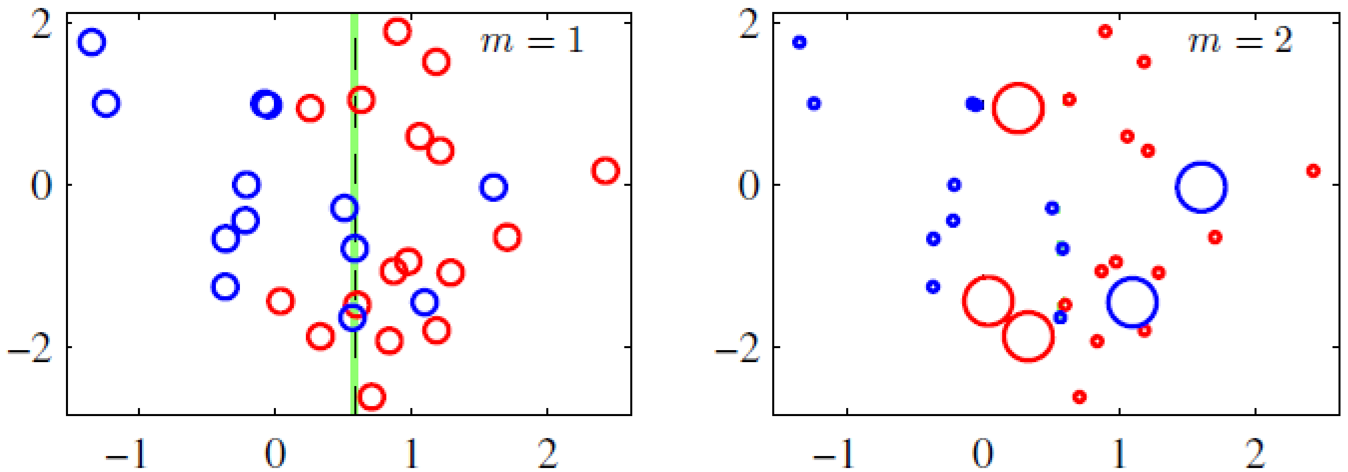
Boosting based on decision stumps

- At each iteration, greedily choose the dimension that minimizes errors based on thresholding.



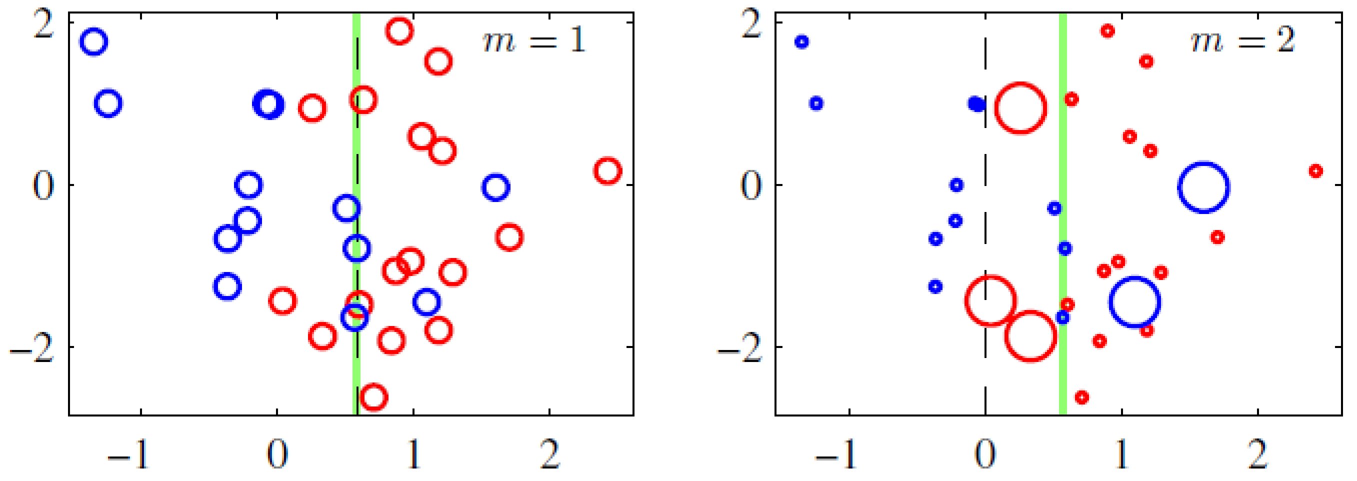
More iterations

- What's next?
- How would you threshold now at iter2?

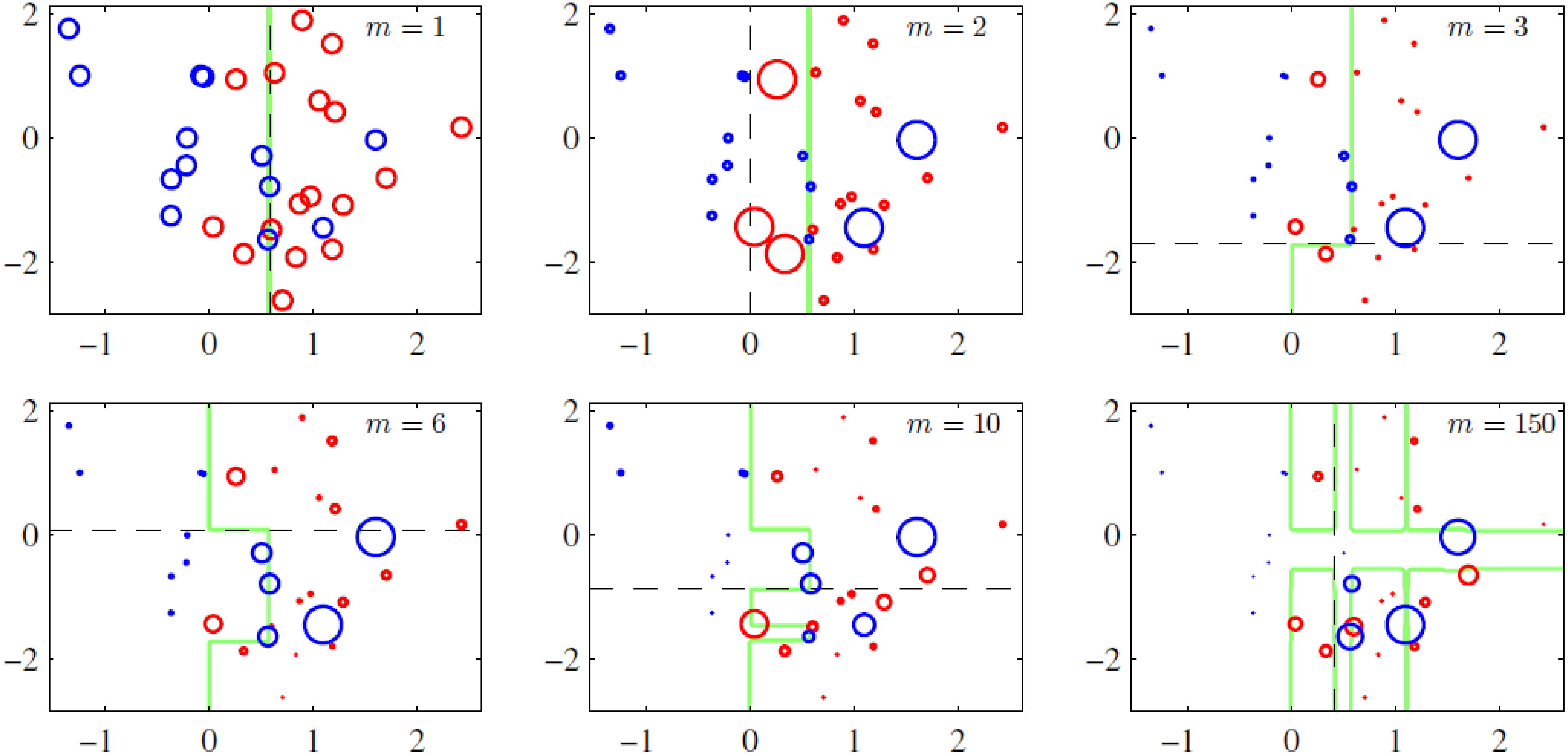


Boosting in action

- Why did the decision boundary shift (slightly towards the left) in the second iteration?



What does Boosting learn eventually?



From Bishop (2006)

Summarizing AdaBoost

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$.
2. For $m = 1, \ldots, M$:
   (a) Fit a classifier $y_m(\mathbf{x})$ to the training data by minimizing the weighted error function
   $$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$
   where $I(y_m(\mathbf{x}_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(\mathbf{x}_n) \neq t_n$ and 0 otherwise.
   (b) Evaluate the quantities
   $$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$
   and then use these to evaluate
   $$\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}.$$
   Where does alpha come from? Any clue?
   (c) Update the data weighting coefficients
   $$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right).$$

From Bishop (2006)

Why doesn’t Boosting overfit?

- Recall there is usually a trade-off between model complexity and predictability.
- What makes Boosting ‘special’ that seems escapable from this trade-off?

Puzzle 2

- What algorithm would you use to automatically recognize these written digits?

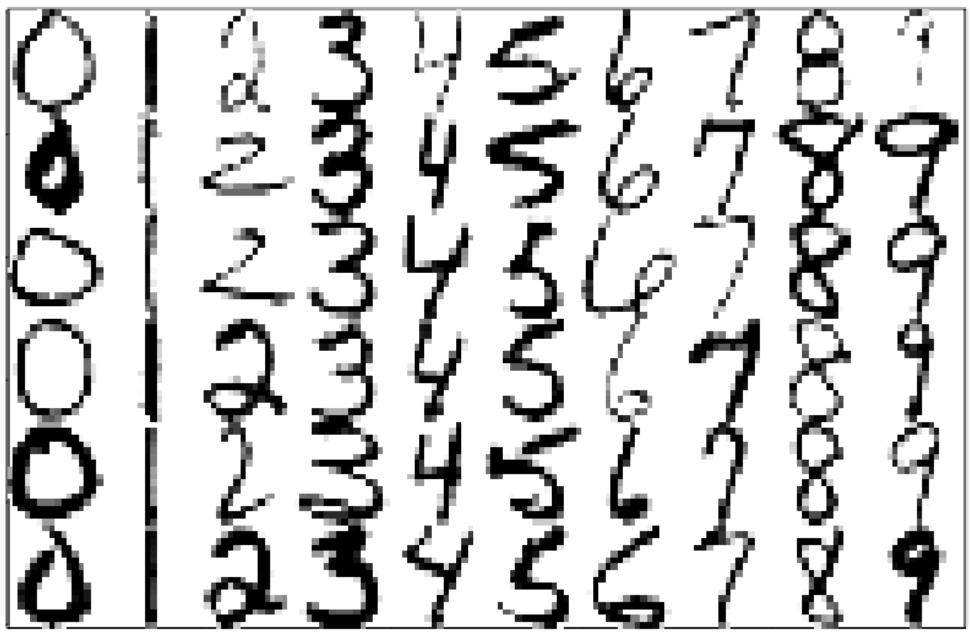


Figure from Google search

Puzzle simplified

- Suppose we use Neural Networks to classify these digits.
- What would be the input and output?

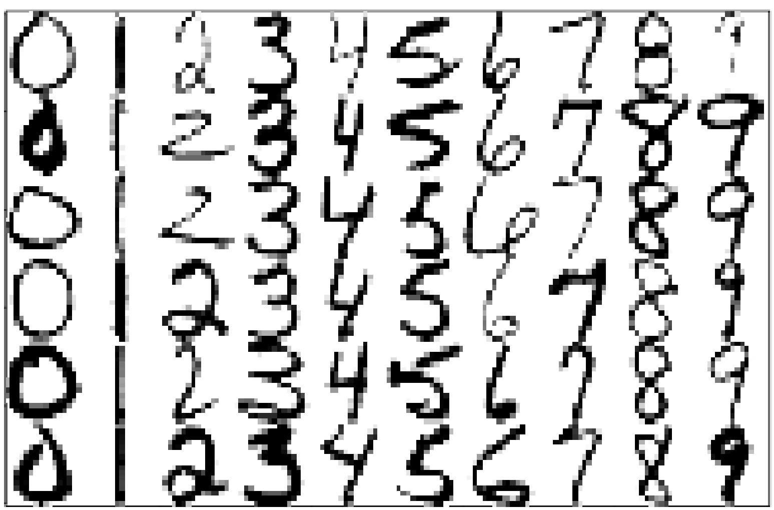
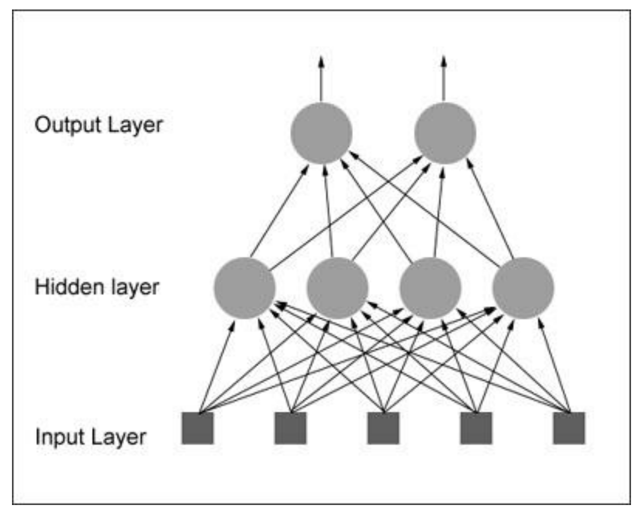
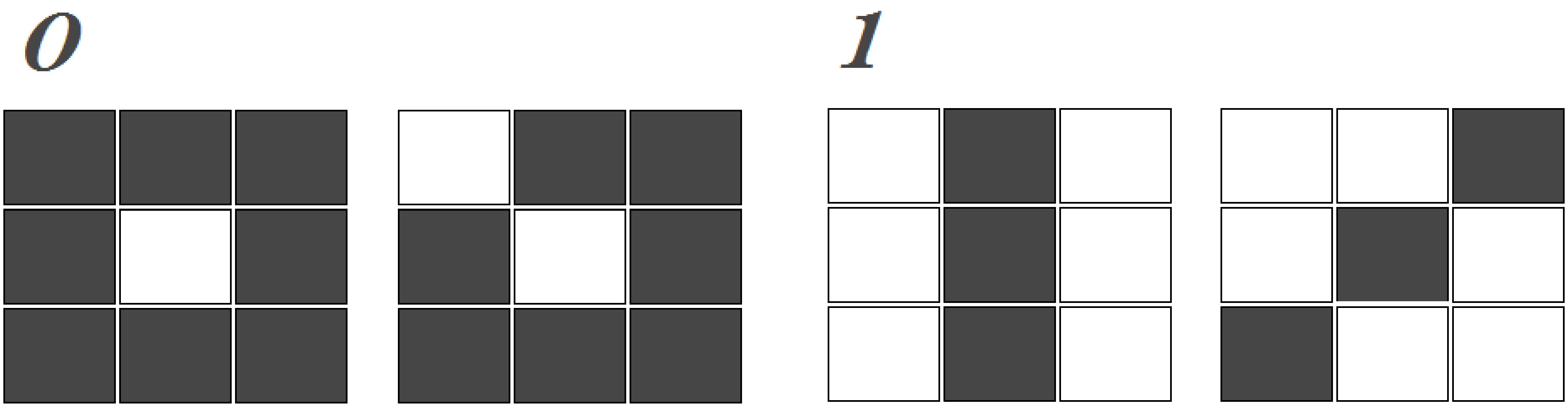


Figure from Google search

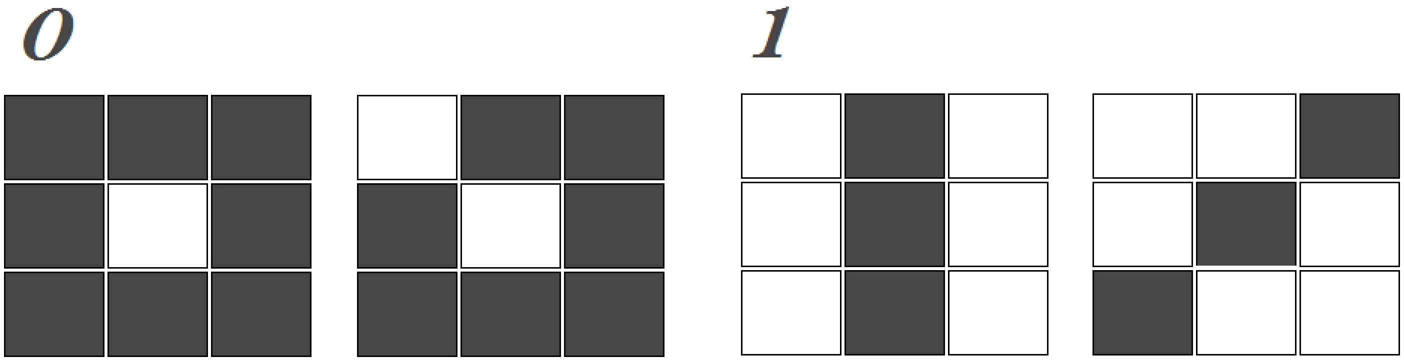


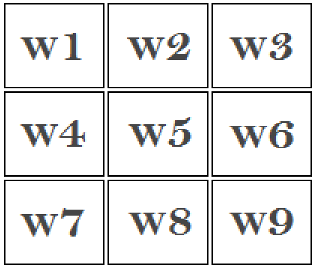
Puzzle simplified further

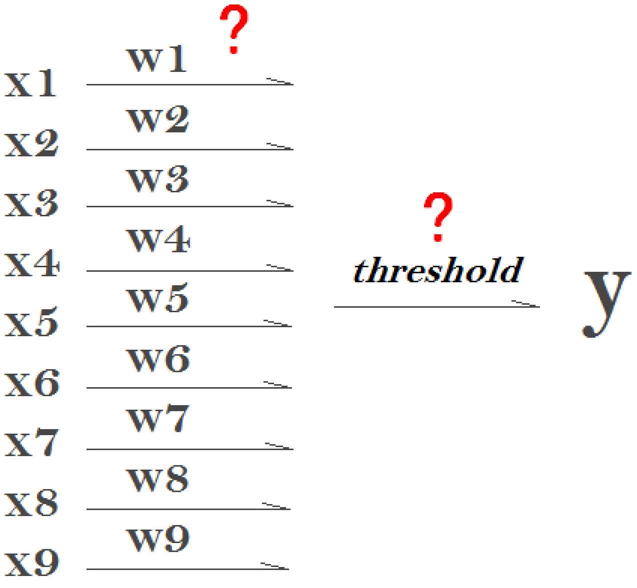
- Suppose we have 2 digits with two examples of each as follows. How might the network look like?



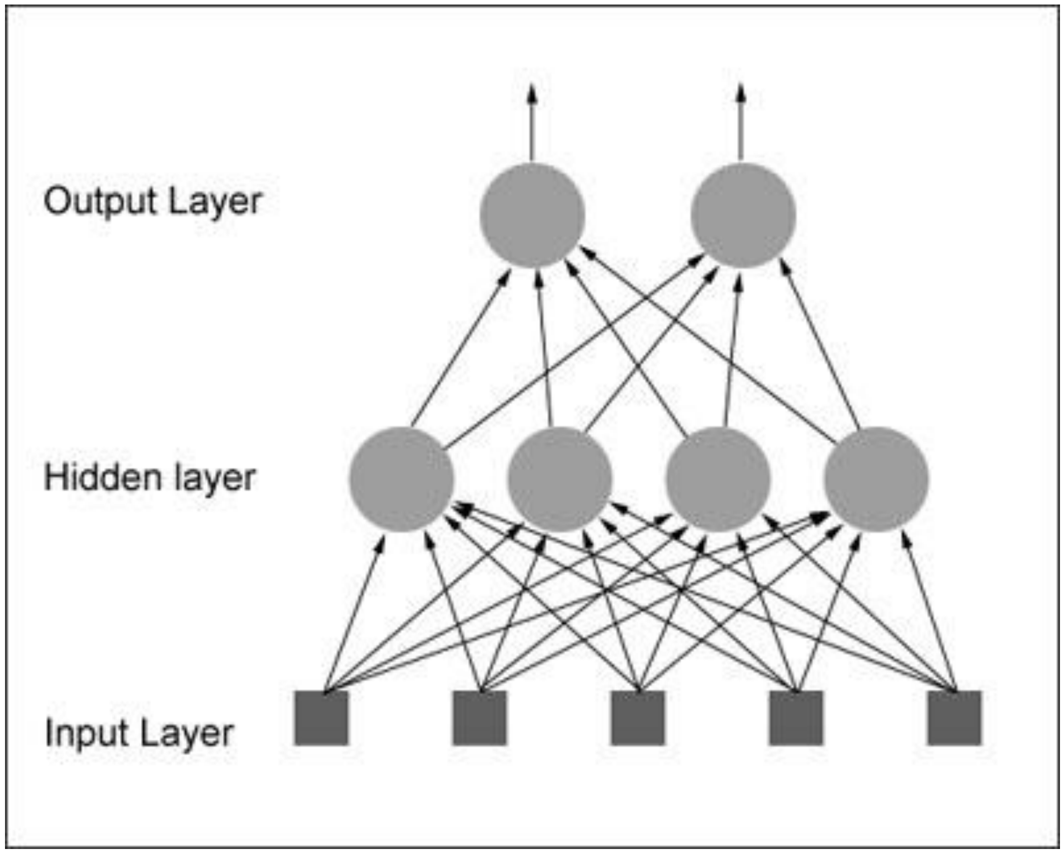
Predicting the weights and threshold







Multi-layer perceptrons and backpropagation



Learning theory

- Some basic definitions
  - Instance
  - Attribute
  - Distribution
  - Concept
  - Target function
  - Hypothesis space
  - Version space
  - Consistent learner

Similarities and differences

- PAC
- Agnostic
- VC dimension

PAC

- Assumptions
- Interpreting the bound
- Limitations

VC dimension

- Shattering
- Interpreting the bound

Mistake bound

- Homework problem