CISC/CMPE452/COGS400 Engineering Multilayer Backpropagation NN

Ch. 3 Text book

Farhana Zulkernine

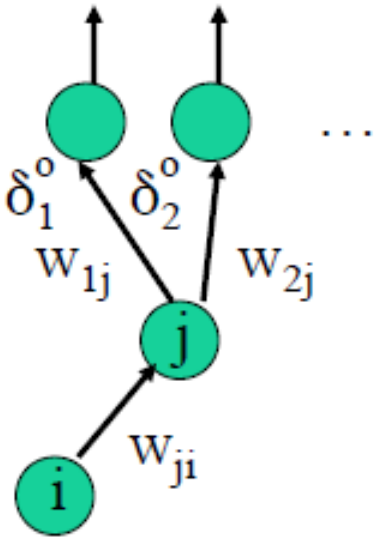
Engineering ANN

- Specific design and implementation choices can require adaptation of the BP algorithm.
  - For example, Sigmoid activation functions will not generate 1 and 0 like TLU.
  - When classes are not linearly separable, multiple output nodes can have values > 0
  - So, for a data point without the class label, which class should the data point be classified into?
  - How long should you iterate for training?
  - How to test that model is accurate enough?

Problems with Backpropagation

1. It is not generally possible to know whether or not there are any *local minima* in the error surface.
   - Even for very simple problems, it has been shown that Backpropagation will sometimes not find the "correct" solution.
2. The kind of simple error function that is used in the basic Backpropagation equations is usually not appropriate.
   - The derivation using a more robust error function can be very difficult.

Problems (cont…)



$$\delta_j^o = (d_j - y_j^2).f'(a_j^2)$$

$$\delta_i^h = \{\sum_j^m \delta_j^o . w_{jh}\}.f'(a_h^1)$$

3. The **accumulated error at the hidden nodes is a sum of signed values**, and so two very large output node errors (one negative and one positive) *could cancel each other out*, leaving the impression that there is no error that was contributed to by the hidden node.

Problems (cont…)

4. The *training set*, the *test set*, and the *validation set* should all be statistically indistinguishable from one another.
   - One cannot control whether the data/situation will change after the network is trained → performance evaluation helps.
   - When selecting the test trials, one should ensure that they are statistically indistinguishable from the training trials and not just randomly selected.
   - It is **difficult to know how many validation trials to use**. More *training* trials is best, but insufficient validation trials could lead to over-generalization.

Problems (cont…)

5. Training times can be very long.
   - May require tens of thousands of training trials.
     - Larger the network →more weights→ longer training time required.
6. No exact methods to configure ANN (number of hidden nodes), and trial-and-error tests may be required to find the best possible performance.
7. Backpropagation is **"unbiological"**.
   - There is no evidence of an ability to use synaptic connections "backwards", as is required to develop the error values at the hidden nodes. There are models which attempt to make Backpropagation more plausible by having a set of "satellite" nodes around each hidden node to backpropagate the signals in a way that conforms to the usual neuron models.
   - Even the strictly forward layering is unlike the usual structures of the brain.

Problems (cont…)

8. Backpropagation cannot usually have its operation enhanced easily with, say, an additional training trial.
   - The relearning that is required can result in a complete restructuring of the values of the weights.
9. Backpropagation may be able to solve a problem, but it is generally not useful in understanding the solution that is being taken.
   - Because features are grouped at the multiple hidden layers making it difficult to know which features are important in classification.

K-Class Classification Problem

- Use K output nodes and transform given desired output to a **1-hot vector**, i.e., 1 at k th position or output node, and 0 at all other positions.

$$\boldsymbol{d_k} = (0^1, 0^2, .., 1^k, .., 0^K)$$

May not happen if not linearly separable

- Training set T_k represents class C_k and has data points x_{kp}, transformed desired outputs d_{kp}.
- The complete training set is $T = T_1 \cup \ldots \cup T_K$.
- ANN should be trained such that for input x_{kp} output d_{kp} is generated.

Error Calculation: Change Desired Output

- Also due to the sigmoid output function, it is hard to reach y=1

$$f(\text{net}) = 1 \text{ when net} \rightarrow \infty \text{ and}$$

$$f(\text{net}) = 0 \text{ when net} \rightarrow -\infty$$

- Because of the shallow slope of the sigmoid function at extreme net inputs, even approaching these values would be very slow.
- Solution: Transform $d_k = (0.1^1, 0.1^2, \ldots, 0.9^k, \ldots, 0.1^K)$

  Using typical values for ε range between 0.01 and 0.1.

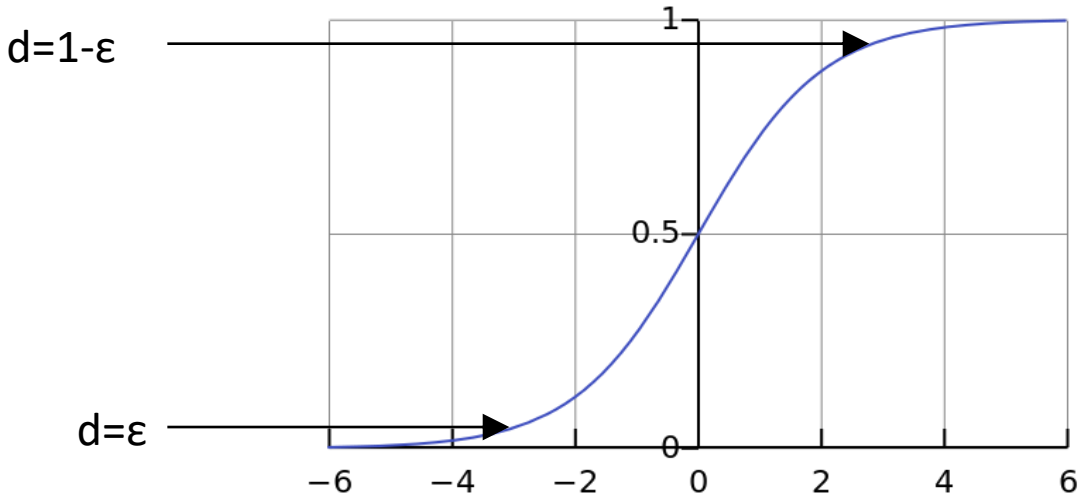
Error Calculation (cont…)

Error $e_{p,j} = 0$ **when** $|(d_j - y_j)| <= \varepsilon \rightarrow \Delta w = 0$

i.e., $d_j = (1- \varepsilon)$ and $y_j = f(\text{net}) >= (1- \varepsilon)$ as net $\rightarrow \infty$

$d_j = \varepsilon$ and $y_j = f(\text{net}) <= \varepsilon$ as net $\rightarrow -\infty$

Otherwise, $e_{p,j} = |d_{p,j} - y_{p,j}| \rightarrow \Delta w \propto |d_{p,j} - y_{p,j}|$



Classification of Unknown Pattern

- **After a network is trained**, assign an unknown input to the class for which $\|d - y\|$ is the lowest

  For data point x_p, if $\|d_j - y_j\| \leq \|d_k - y_k\|$,
  where $j \neq k$ for all k, then assign x_p to class j.
  - $\|d - y\|$ represents Euclidean norm or distance between d and y
  - $\|d - y\| => \sqrt{(d_1 - y_1)^2 + (d_2 - y_2)^2 + \cdots + (d_m - y_m)^2}$
- If two classes are equidistant, assign the pattern randomly to one of the two classes.

  If d= 1, and $y_j > y_k$ for $j \neq k$, assign x_p to class j.
  - For a two-class problem if y>=0.5 assign it to class C_1 and otherwise to class C_2.