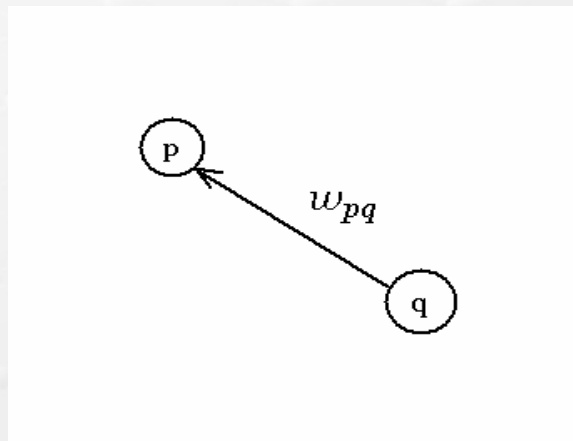


In general:



$$\Delta w_{pq} = h \sum_m d_p^m V_q^m$$

off - line

$$\Delta w_{pq} = h d_p^m V_q^m$$

on - line

Back – Propagation Algorithm

- Incremental update
- Consider a network with M layers and denote $(m = 0 \dots M)$

$V_i^m \equiv$ output of *i-th* unit of layer m

$w_{ij}^m \equiv$ weight on the connection between *j-th* neuron of layer m-1 and *i-th* neuron in layer m

Back – Propagation Algorithm

1. Initialize the weight to (small) random values
2. Choose a pattern \bar{x}^m and apply it to the input layer ($m=0$)

$$V_k^0 = x_k^m \quad \forall k$$

3. Propagate the signal forward:

$$V_i^m = g(h_i^m) = g\left(\sum_j w_{ij} V_j^{m-1}\right)$$

4. Compute the δ 's for the output layer:

$$d_i^M = g'(h_i^M)(y_i^M - V_i^M)$$

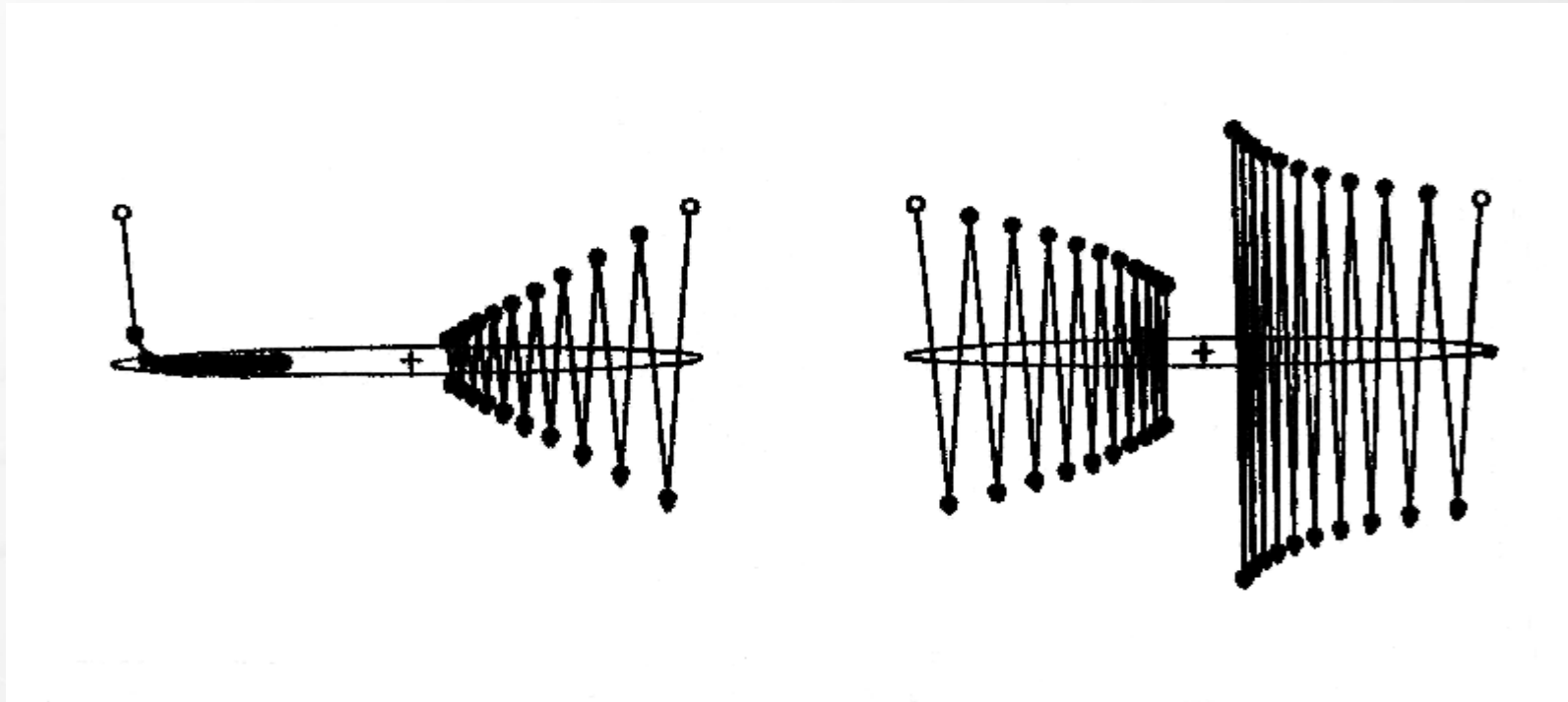
5. Compute the δ 's for all preceding layers:

$$d_i^{m-1} = g'(h_i^{m-1}) \sum_j w_{ji}^m d_j^m$$

6. Update connection weights:

$$w_{ij}^{NEW} = w_{ij}^{OLD} + \Delta w_{ij} \quad \text{where} \quad \Delta w_{ij} = h d_i^m V_j^{m-1}$$

7. Go back to step 2 until convergence



Gradient descent on a simple quadratic surface (the left and right parts are copies of the same surface). Four trajectories are shown, each for 20 steps from the open circle. The minimum is at the + and the ellipse shows a constant error contour. The only significant difference between the trajectories is the value of η , which was 0.02, 0.0476, 0.049, and 0.0505 from left to right.

Momentum Term

Gradient descent may :

- converge too slowly if η is small
- oscillate if η is too large

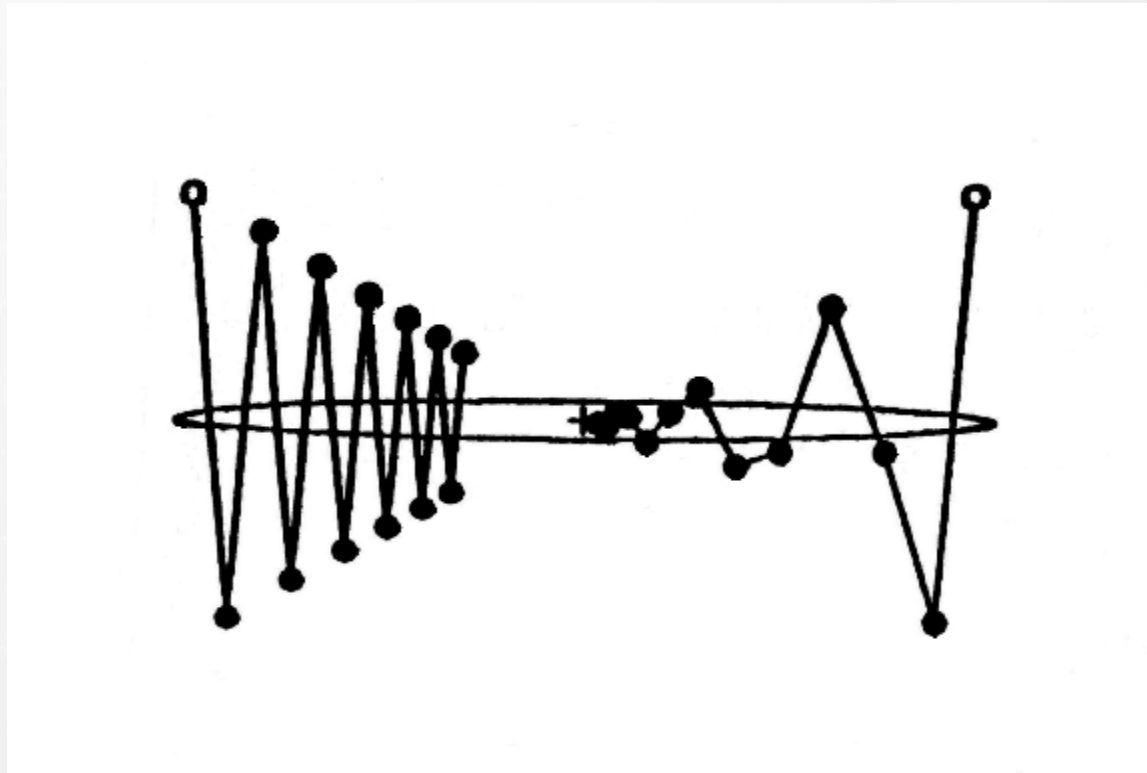
Simple remedy :

$$\Delta w_{pq}(t+1) = -\eta \frac{\partial E}{\partial w_{pq}} + \alpha \Delta w_{pq}(t)$$

momentum

The momentum term allows us to use large values for the “learning rate” η thereby avoiding oscillatory phenomena

Usually: $\alpha = 0.9$ $\eta = 0.5$



Gradient descent on the simple quadratic surface. Both trajectories are for 12 steps with $\eta = 0.0476$, the best value in the absence of momentum. On the left there is no momentum ($\alpha = 0$), while $\alpha = 0.5$ on the right.

Local Minima

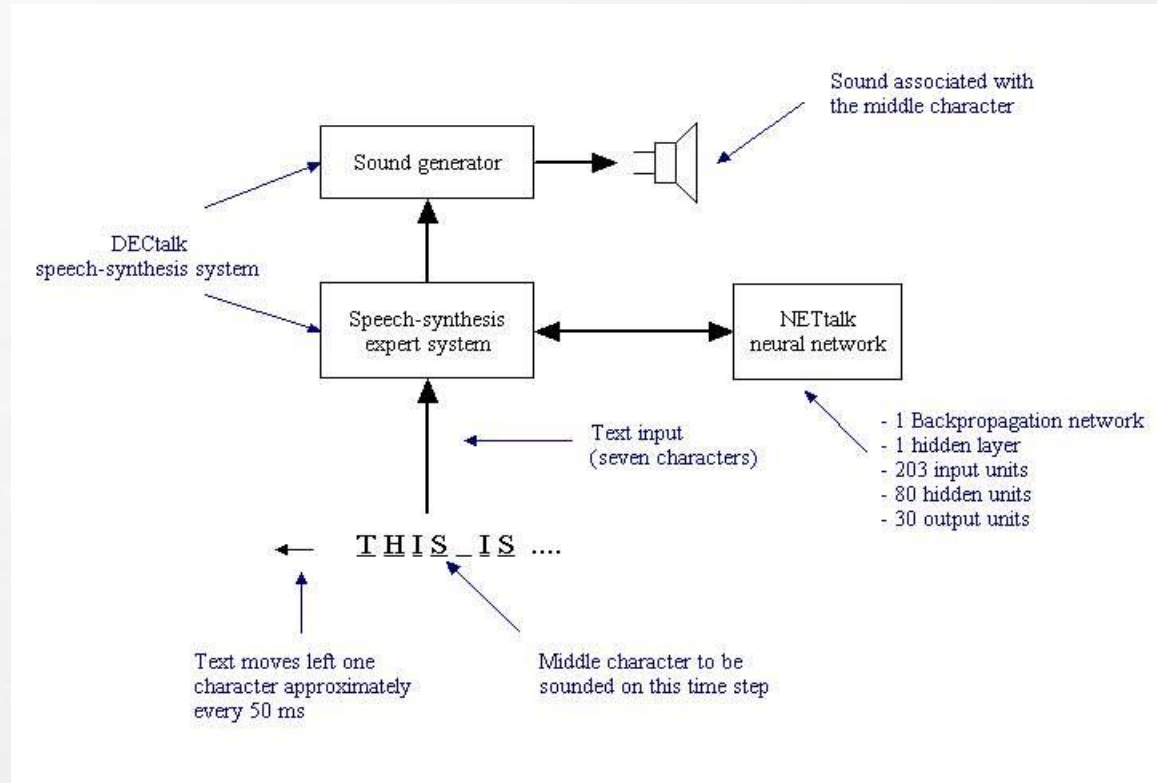
Back-prop cannot avoid local minima.

Choice of initial weights it's also important to avoid this problem. If they are too large, the sigomoids tend to saturate since the beginning of the learning process.

Heuristic  Choose initial weights as $w_{ij} \cong 1 / \sqrt{k_i}$

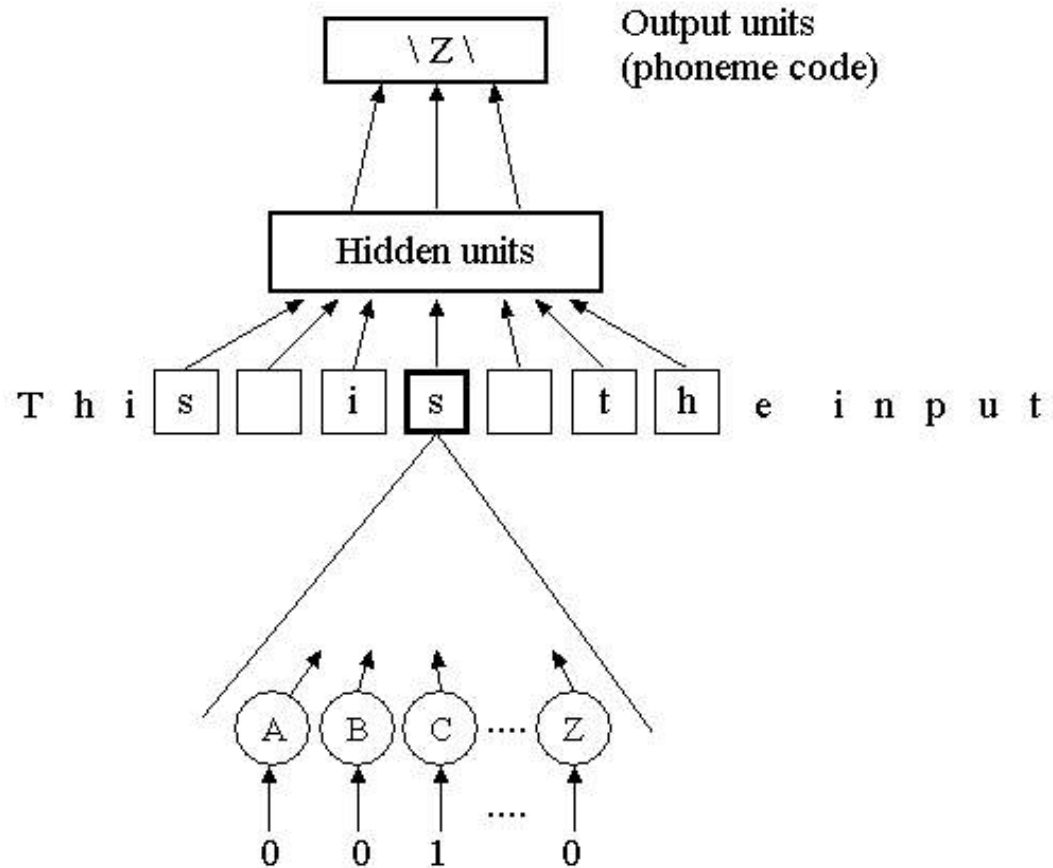
where k_i is the number of units that feed unit i (the “fan-in” of i)

NETtalk System



NETtalk neural network speech synthesizer. The NETtalk backpropagation network is trained by a rule-based expert system element of the DECTalk commercial speech synthesis system. NETtalk is then used to replace that element. The result is a new speech synthesis system that has approximately the same overall performance as the original. In other words, the NETtalk neural network becomes functionally equivalent to an expert system with hundreds of rules. The question then becomes: how are these rules represented within the NETtalk neural network? The answer is: nobody really knows.

The NETtalk architecture



NETtalk

- A network to pronounce English text
- 7 x 29 input units
- 1 hidden layer with 80 hidden units
- 26 output units encoding phonemes
- Trained by 1024 words with context
- Produce intelligible speech after 10 training epochs
- Functionally equivalent to DEC-talk
- Rule-based DEC-talk is the result of a decade of efforts by many linguists
- NETtalk learns from examples, and requires no linguistic knowledge

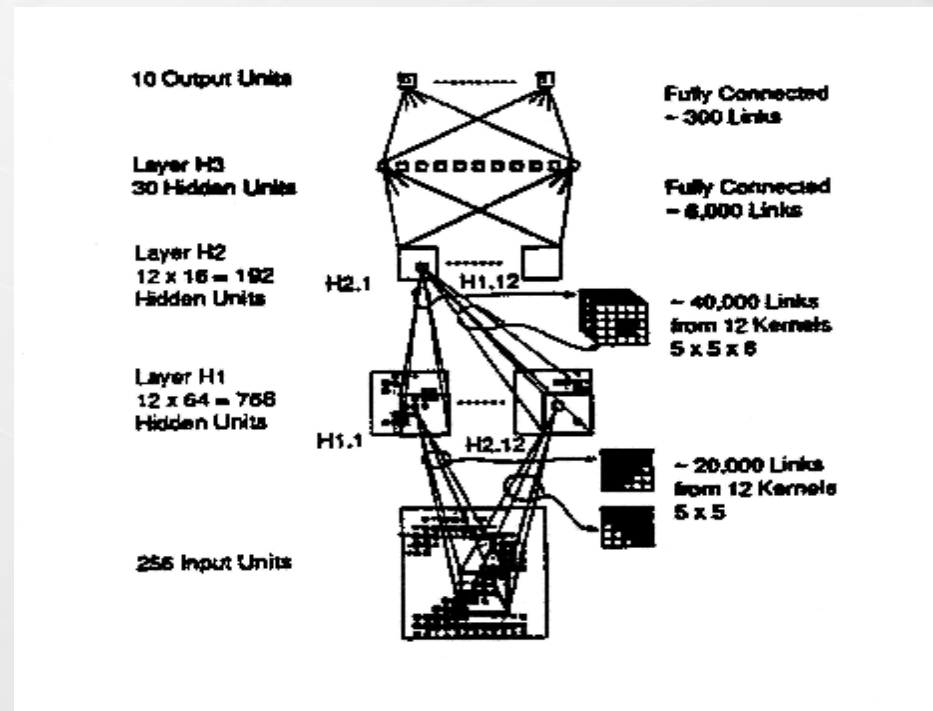
Character Recognition

2601446357146371037314497
1105711129991102160028870
3301033010290602840029012
9405290670980129550299055
510129201803270124431064
1161176057188600158701899
1157557212570688327499516
9950572001536272203292372
3507271272315393053880314
1371914119129192531917014
1011913485726803224414186
6359720299299722510046701
3084141591010615406103631
1064111030475262009979966
8912086708557131427955460
2017730187112993089970984
0109707597331972015519055
1073318255182814358010163
1787521655460554603546055
18255108503047520439401


Isolated handwritten digits taken from postal zip codes

Character Recognition: Approach

- Preprocessing
Scaling → 16 x 16 gray level images
- Network
256 input units, 3 hidden layers, 10 output units
Gray level image is fed into the network



First Hidden Layer: H1

- 12 groups of 64 units arranged in 12 independent 8 x 8 feature maps.
- Each unit in a feature map takes input from a 5 x 5 neighborhood in the input image.
- Units that are 1 pixel apart in the feature map are 2 pixel apart in the input image.
- Each unit in a group performs the same operation  25 equal weights and 1 bias.

$$12 \times 8 \times 8 = 768 \text{ units}$$

$$768 \times 25 + 768 = 19968 \text{ connections}$$

$$25 \times 12 + 768 = 1068 \text{ independent parameters}$$

Second Hidden Layer: H2

- 12 groups of 16 units arranged in 12 independent 4 x 4 feature maps.
- Each unit combines local information coming from 8 of the 12 feature maps in H1 (200 inputs to each unit).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------|---|---|---|---|---|---|---|---|---|----|----|----|
| H1.1 | x | x | x | | | | | | | x | x | x |
| H1.2 | x | x | x | | | | | | | x | x | x |
| H1.3 | x | x | x | x | x | x | | | | | | |
| H1.4 | x | x | x | x | x | x | | | | | | |
| H1.5 | | | | x | x | x | x | x | x | | | |
| H1.6 | | | | x | x | x | x | x | x | | | |
| H1.7 | | | | | | | x | x | x | x | x | x |
| H1.8 | | | | | | | x | x | x | x | x | x |
| H1.9 | x | x | x | x | x | x | x | x | x | x | x | x |
| H1.10 | x | x | x | x | x | x | x | x | x | x | x | x |
| H1.11 | x | x | x | x | x | x | x | x | x | x | x | x |
| H1.12 | x | x | x | x | x | x | x | x | x | x | x | x |

$12 \times 4 \times 4 = 192$ units

$192 \times 200 + 192 = 38592$ connections

$12 \times 200 + 192 = 2592$ independent parameters

Rest of the Network

- Third Hidden Layer: H3
30 units fully connected to H2, 5790 connections
- Output Units
10 units fully connected to H3, 310 connections
- Total specification
1256 units
64660 connections
9760 independent parameters