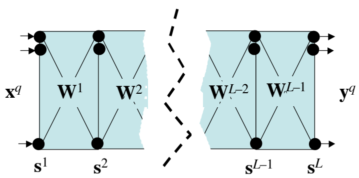
Multilayer Notation



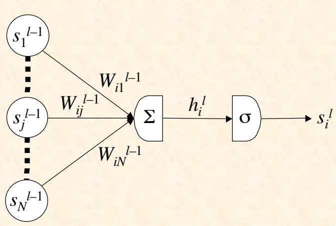
11/29/04 1

Notation

- L layers of neurons labeled 1, …, L
- N_l neurons in layer l
- $\mathbf{s}^l$ = vector of outputs from neurons in layer l
- input layer $\mathbf{s}^1 = \mathbf{x}^q$ (the input pattern)
- output layer $\mathbf{s}^L = \mathbf{y}^q$ (the actual output)
- $\mathbf{W}^l$ = weights between layers l and l+1
- Problem: find how outputs y_i^q vary with weights W_{jk}^l (l = 1, …, L–1)

11/29/04 2

Typical Neuron



11/29/04 3

Error Back-Propagation

We will compute $\dfrac{\partial E^q}{\partial W_{ij}^l}$ starting with last layer ($l = L - 1$)

and working back to earlier layers ($l = L - 2, \ldots, 1$)

11/29/04 4

Delta Values

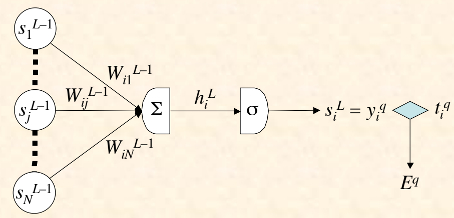
Convenient to break derivatives by chain rule:

$$\frac{\partial E^q}{\partial W_{ij}^{l-1}} = \frac{\partial E^q}{\partial h_i^l}\frac{\partial h_i^l}{\partial W_{ij}^{l-1}}$$

Let $\delta_i^l = \dfrac{\partial E^q}{\partial h_i^l}$

So $\dfrac{\partial E^q}{\partial W_{ij}^{l-1}} = \delta_i^l \dfrac{\partial h_i^l}{\partial W_{ij}^{l-1}}$

Output-Layer Neuron



Output-Layer Derivatives (1)

$$\delta_i^L = \frac{\partial E^q}{\partial h_i^L} = \frac{\partial}{\partial h_i^L}\sum_k \left(s_k^L - t_k^q\right)^2$$

$$= \frac{\mathrm{d}\left(s_i^L - t_i^q\right)^2}{\mathrm{d}\, h_i^L} = 2\left(s_i^L - t_i^q\right)\frac{\mathrm{d}\, s_i^L}{\mathrm{d}\, h_i^L}$$

$$= 2\left(s_i^L - t_i^q\right)\sigma'\left(h_i^L\right)$$

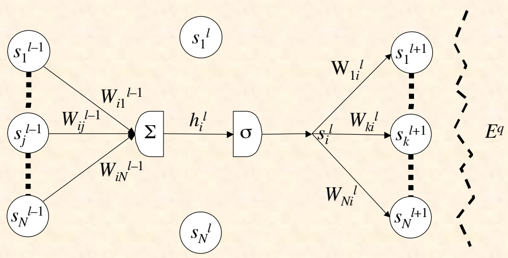
Output-Layer Derivatives (2)

$$\frac{\partial h_i^L}{\partial W_{ij}^{L-1}} = \frac{\partial}{\partial W_{ij}^{L-1}}\sum_k W_{ik}^{L-1} s_k^{L-1} = s_j^{L-1}$$

$$\therefore \frac{\partial E^q}{\partial W_{ij}^{L-1}} = \delta_i^L s_j^{L-1}$$

where $\delta_i^L = 2\left(s_i^L - t_i^q\right)\sigma'\left(h_i^L\right)$

Hidden-Layer Neuron



Hidden-Layer Derivatives (1)

Recall $\frac{\partial E^q}{\partial W_{ij}^{l-1}} = \delta_i^l \frac{\partial h_i^l}{\partial W_{ij}^{l-1}}$

$$\delta_i^l = \frac{\partial E^q}{\partial h_i^l} = \sum_k \frac{\partial E^q}{\partial h_k^{l+1}} \frac{\partial h_k^{l+1}}{\partial h_i^l} = \sum_k \delta_k^{l+1} \frac{\partial h_k^{l+1}}{\partial h_i^l}$$

$$\frac{\partial h_k^{l+1}}{\partial h_i^l} = \frac{\partial \sum_m W_{km}^l s_m^l}{\partial h_i^l} = \frac{\partial W_{ki}^l s_i^l}{\partial h_i^l} = W_{ki}^l \frac{\mathrm{d}\sigma(h_i^l)}{\mathrm{d}h_i^l} = W_{ki}^l \sigma'(h_i^l)$$

$$\therefore \delta_i^l = \sum_k \delta_k^{l+1} W_{ki}^l \sigma'(h_i^l) = \sigma'(h_i^l) \sum_k \delta_k^{l+1} W_{ki}^l$$

Hidden-Layer Derivatives (2)

$$\frac{\partial h_i^l}{\partial W_{ij}^{l-1}} = \frac{\partial}{\partial W_{ij}^{l-1}} \sum_k W_{ik}^{l-1} s_k^{l-1} = \frac{\mathrm{d} W_{ij}^{l-1} s_j^{l-1}}{\mathrm{d} W_{ij}^{l-1}} = s_j^{l-1}$$

$$\therefore \frac{\partial E^q}{\partial W_{ij}^{l-1}} = \delta_i^l s_j^{l-1}$$

where $\delta_i^l = \sigma'(h_i^l) \sum_k \delta_k^{l+1} W_{ki}^l$

Derivative of Sigmoid

Suppose $s = \sigma(h) = \frac{1}{1 + \exp(-\alpha h)}$ (logistic sigmoid)

$$\mathrm{D}_h\, s = \mathrm{D}_h \left[1 + \exp(-\alpha h)\right]^{-1} = -\left[1 + \exp(-\alpha h)\right]^{-2} \mathrm{D}_h \left(1 + e^{-\alpha h}\right)$$

$$= -\left(1 + e^{-\alpha h}\right)^{-2} \left(-\alpha e^{-\alpha h}\right) = \alpha \frac{e^{-\alpha h}}{\left(1 + e^{-\alpha h}\right)^2}$$

$$= \alpha \frac{1}{1 + e^{-\alpha h}} \frac{e^{-\alpha h}}{1 + e^{-\alpha h}} = \alpha s \left(\frac{1 + e^{-\alpha h}}{1 + e^{-\alpha h}} - \frac{1}{1 + e^{-\alpha h}}\right)$$

$$= \alpha s(1 - s)$$

Summary of Back-Propagation Algorithm

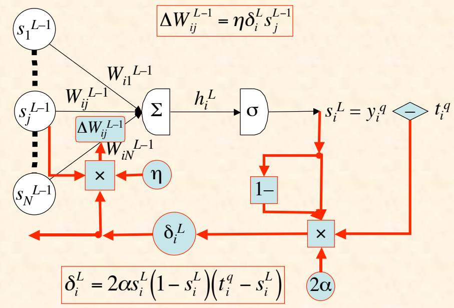
Output layer: $\delta_i^L = 2\alpha s_i^L\left(1 - s_i^L\right)\left(s_i^L - t_i^q\right)$

$$\frac{\partial E^q}{\partial W_{ij}^{L-1}} = \delta_i^L s_j^{L-1}$$

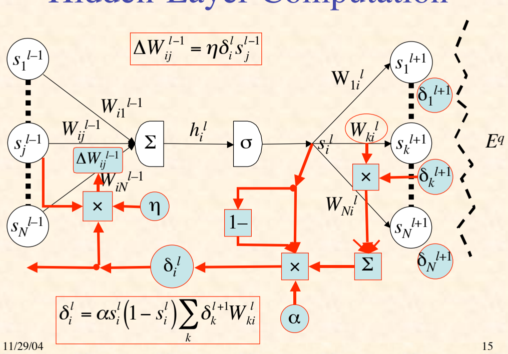
Hidden layers: $\delta_i^l = \alpha s_i^l\left(1 - s_i^l\right)\sum_k \delta_k^{l+1} W_{ki}^l$

$$\frac{\partial E^q}{\partial W_{ij}^{l-1}} = \delta_i^l s_j^{l-1}$$

Output-Layer Computation



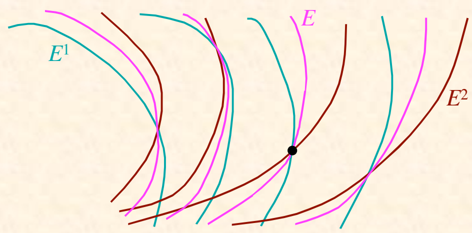
Hidden-Layer Computation



Training Procedures

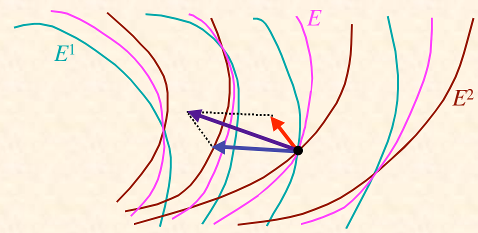
- Batch Learning
  - on each *epoch* (pass through all the training pairs),
  - weight changes for all patterns accumulated
  - weight matrices updated at end of epoch
  - accurate computation of gradient
- Online Learning
  - weight are updated after back-prop of each training pair
  - usually randomize order for each epoch
  - approximation of gradient
- Doesn't make much difference

Summation of Error Surfaces



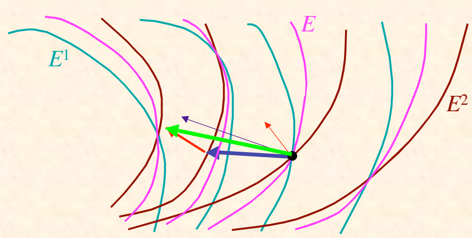
11/29/04 17

Gradient Computation in Batch Learning



11/29/04 18

Gradient Computation in Online Learning



11/29/04 19

The Golden Rule of Neural Nets

Neural Networks are the *second-best* way to do *everything*!

11/29/04 20

VIII. Review of Key Concepts

11/29/04 21

Complex Systems

- Many interacting elements
- Local vs. global order: entropy
- Scale (space, time)
- Phase space
- Difficult to understand
- Open systems

11/29/04 22

Many Interacting Elements

- Massively parallel
- Distributed information storage & processing
- Diversity
  - avoids premature convergence
  - avoids inflexibility

11/29/04 23

Complementary Interactions

- Positive feedback / negative feedback
- Amplification / stabilization
- Activation / inhibition
- Cooperation / competition
- Positive / negative correlation

11/29/04 24

Emergence & Self-Organization

- Microdecisions lead to macrobehavior
- Circular causality (macro / micro feedback)
- Coevolution
  - predator/prey, Red Queen effect
  - gene/culture, niche construction, Baldwin effect

Pattern Formation

- Excitable media
- Amplification of random fluctuations
- Symmetry breaking
- Specific difference vs. generic identity
- Automatically adaptive

Stigmergy

- Continuous (quantitative)
- Discrete (qualitative)
- Coordinated algorithm
  - non-conflicting
  - sequentially linked

Emergent Control

- Stigmergy
- Entrainment (distributed synchronization)
- Coordinated movement
  - through attraction, repulsion, local alignment
  - in concrete or abstract space
- Cooperative strategies
  - nice & forgiving, but reciprocal
  - evolutionarily stable strategy

Attractors

- Classes
  - point attractor
  - cyclic attractor
  - chaotic attractor
- Basin of attraction
- Imprinted patterns as attractors
  - pattern restoration, completion, generalization, association

Wolfram's Classes

- Class I: point
- Class II: cyclic
- Class III: chaotic
- Class IV: complex (edge of chaos)
  - persistent state maintenance
  - bounded cyclic activity
  - global coordination of control & information
  - order for free

Energy / Fitness Surface

- Descent on energy surface / ascent on fitness surface
- Lyapunov theorem to prove asymptotic stability / convergence
- Soft constraint satisfaction / relaxation
- Gradient (steepest) ascent / descent
- Adaptation & credit assignment

Biased Randomness

- Exploration vs. exploitation
- Blind variation & selective retention
- Innovation vs. incremental improvement
- Pseudo-temperature
- Diffusion
- Mixed strategies

Natural Computation

- Tolerance to noise, error, faults, damage
- Generality of response
- Flexible response to novelty
- Adaptability
- Real-time response
- Optimality is secondary

Student Course Evaluation! (Do it online)