**Multilayer Notation**

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

---

**Typical Neuron**

**Error Back-Propagation**

We will compute $\frac{\partial E^q}{\partial W_{jk}^l}$ starting with last layer ($l = L - 1$) and working back to earlier layers ($l = L - 2, ..., 1$).
Delta Values

Convenient to break derivatives by chain rule:

\[
\frac{\partial E^q}{\partial W^E} = \frac{\partial E^q}{\partial h^L} \frac{\partial h^L}{\partial W^E}
\]

Let \( \delta_l^i = \frac{\partial E^q}{\partial h^L} \)

So \( \frac{\partial E^q}{\partial W^E} = \delta_l^i \frac{\partial h^L}{\partial W^E} \)

Output-Layer Neuron

\[
\delta^L = \frac{\partial E^q}{\partial h^L} = \sum_k \left( s_k^L - t_k^L \right) \frac{d}{dh^L} \left( s_k^L \right) = 2(s_i^L - t_i^L) \sigma'(h_i^L)
\]

Output-Layer Derivatives (1)

\[
\frac{\partial h_i^L}{\partial W_0^{L-1}} = \frac{\partial}{\partial W_0^{L-1}} \sum_k W_{ik}^{L-1} s_k^{L-1} = s_i^{L-1}
\]

Output-Layer Derivatives (2)

where \( \delta^L = 2(s_i^L - t_i^L) \sigma'(h_i^L) \)
Hidden-Layer Neuron

Hidden-Layer Derivatives (1)
Recall
\[ \frac{\partial E'}{\partial W_{ij}^{l-1}} = \delta'_j \frac{\partial h_i}{\partial W_{ij}^{l-1}} \]
\[ \delta'_j = \frac{\partial E'}{\partial h_i} = \sum_{k} \frac{\partial E'}{\partial s_k} \frac{\partial s_k}{\partial h_i} + \sum_{k} \delta_k^{l+1} \frac{\partial h_i}{\partial s_k} \]
\[ \frac{\partial h_i}{\partial W_{ij}^{l-1}} = -\sum_{k} W_{kj}^{l} \frac{d}{dh_i} \log(1 + e^{-h_i}) = \theta'(h_i) \sum_k W_{kj}^{l} \delta'_k \]
\[ \therefore \delta'_j = \sum_k \delta_k^{l+1} W_{kj}^{l} \theta'(h_i) = \theta'(h_i) \sum_k \delta_k^{l+1} W_{kj}^{l} \]

Hidden-Layer Derivatives (2)
\[ \frac{\partial h_i}{\partial W_{ij}^{l-1}} = \frac{\partial}{\partial W_{ij}^{l-1}} \sum_{k} W_{kj}^{l} s_k^{l-1} = \frac{dW_{ij}^{l-1} s_k^{l-1}}{dW_{ij}^{l-1}} = s_j^{l-1} \]
\[ \therefore \frac{\partial E'}{\partial W_{ij}^{l-1}} = \delta'_j s_j^{l-1} \]
where \[ \delta'_j = \theta'(h_i) \sum_k \delta_k^{l+1} W_{kj}^{l} \]

Derivative of Sigmoid
Suppose \[ s = \theta(h) = \frac{1}{1 + e^{-h}} \] (logistic sigmoid)
\[ D_s = D_s \left[ 1 + \exp(-ah) \right] = \left[ 1 + \exp(-ah) \right]^2 D_a (1 + e^{-ah}) \]
\[ = \left[ 1 + e^{-ah} \right] \left[ -e^{-ah} \right] = \theta'(h) \frac{e^{-ah}}{1 + e^{-ah}} \]
\[ = \frac{1}{1 + e^{-ah}} - \frac{e^{-ah}}{1 + e^{-ah}} = \theta(h) \left( 1 + e^{-ah} \right) - \frac{1}{1 + e^{-ah}} \]
\[ = \theta(h) (1 - \theta(h)) \]
Summary of Back-Propagation Algorithm

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

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

Hidden layers: $\delta_i^l = \alpha(1 - s_i^l)\sum \delta_j^{l+1} W_{ij}^l$

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

Output-Layer Computation

$\delta_i^L = \eta \delta_i^{L+1}$

Hidden-Layer Computation

$\delta_i^l = \alpha(1 - s_i^l)\sum \delta_j^{l+1} W_{ij}^l$

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

Gradient Computation in Batch Learning

Gradient Computation in Online Learning

The Golden Rule of Neural Nets

Neural Networks are the second-best way to do everything!
VIII. Review of Key Concepts

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

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

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