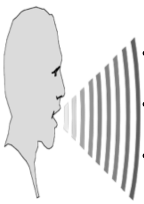
Rous

EEL5840: Elements of Machine Intelligence

Announcements



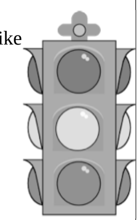
- Announcements:
  - > None
- Today's Handouts in www:
  - > Outline Class 10
- Web Site
  - > www.mil.ufl.edu/eel5840
  - > Software and Notes
  - > XLISP Documentation

Rous

EEL5840: Elements of Machine Intelligence

Today's Menu

- Introduction to ANNs (original material by Dr. Mike Nechyba)
  - > Slides 21-60
  - > Neural Nets in Practice - 3 steps
  - > ANN Training
  - > Gradient Descent
  - > Backprop Summary
  - > Basic Steps in Using ANNs
  - > Adjusting the Learning Rate



Rous

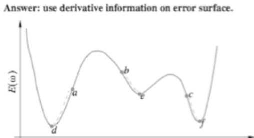
EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Neural network training

Key problem: How to adjust w to minimize E?

Answer: use derivative information on error surface.



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Gradient descent (one parameter)

1. Initialize ω to some random initial value.
2. Change ω iteratively at step t according to:

$$\omega(t+1) = \omega(t) - \eta\frac{dE}{d\omega(t)}$$

Implies local, not global minimum...

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

General gradient descent

1. Initialize $\mathbf{w}$ to some random initial value.
2. Change $\mathbf{w}$ iteratively at step t according to:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla E[\mathbf{w}(t)]$$

$$\nabla E[\mathbf{w}(t)] = \left[\frac{\partial E}{\partial \omega_1(t)} \; \frac{\partial E}{\partial \omega_2(t)} \; \cdots \; \frac{\partial E}{\partial \omega_q(t)}\right]^T$$

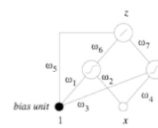
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Simple example of gradient computation

Compute $\frac{\partial E}{\partial \omega_4}$ for the neural network below:



Single training pattern

$\langle x, y \rangle$

$$E = \frac{1}{2}(y-z)^2$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Derivation

Generalization to multiple training patterns:

$$\frac{\partial E}{\partial \omega_j} = \frac{\partial}{\partial \omega_j}\left[\frac{1}{2}\sum_{i=1}^{p}(y_i - z_i)^2\right] = \sum_{i=1}^{p}\frac{\partial}{\partial \omega_j}\left[\frac{1}{2}(y_i - z_i)^2\right].$$

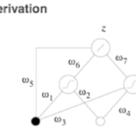
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Derivation

$$net_1 = \omega_1 + \omega_2 x$$
$$net_2 = \omega_3 + \omega_4 x$$
$$h_1 = \gamma(net_1)$$
$$h_2 = \gamma(net_2)$$
$$z = \omega_5 + \omega_6 h_1 + \omega_7 h_2$$
$$\frac{\partial E}{\partial \omega_4} = -(y-z)\frac{\partial z}{\partial \omega_4}$$



$$\frac{\partial E}{\partial \omega_4} = (z-y)\left(\frac{\partial z}{\partial h_2}\right)\left(\frac{\partial h_2}{\partial net_2}\right)\left(\frac{\partial net_2}{\partial \omega_4}\right)$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Derivation

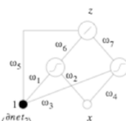
$net_1 = \omega_1 + \omega_2 x$

$net_2 = \omega_3 + \omega_4 x$

$h_1 = \gamma(net_1)$

$h_2 = \gamma(net_2)$

$z = \omega_5 + \omega_6 h_1 + \omega_7 h_2$



$$\frac{\partial E}{\partial \omega_4} = (z-y)\left(\frac{\partial z}{\partial h_2}\right)\left(\frac{\partial h_2}{\partial net_2}\right)\left(\frac{\partial net_2}{\partial \omega_4}\right)$$

$$\frac{\partial E}{\partial \omega_4} = (z-y)\omega_7\gamma'(net_2)x$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Generalization: Backpropagation

Key problem: Generalize specific result to compute derivatives in more general manner.

Answer: *Backpropagation algorithm* [Rumelhart and McClelland,1986].

- Efficient, algorithmic formulation for computing error derivatives
- Gradient computation without hardcoding derivatives (allows on-the-fly adjustment of NN architectures).

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

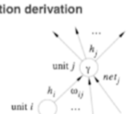
Backpropagation derivation

$h_j = \gamma(net_j)$

$net_j = \sum_k h_k \omega_{kj}$

$$\frac{\partial E}{\partial \omega_{ij}} = \left(\frac{\partial E}{\partial net_j}\right)\left(\frac{\partial net_j}{\partial \omega_{ij}}\right)$$

$$\frac{\partial net_j}{\partial \omega_{ij}} = h_i$$



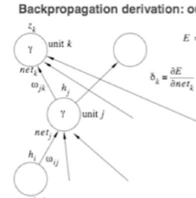
$$\delta_j = \frac{\partial E}{\partial net_j} \qquad \frac{\partial E}{\partial \omega_{ij}} = \delta_j h_i$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Backpropagation derivation: output units



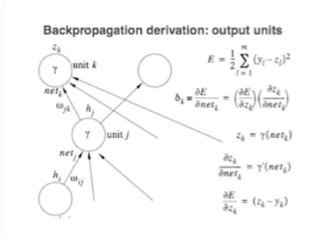
$$E = \frac{1}{2}\sum_{l=1}^{m}(y_l - z_l)^2$$

$$\delta_k = \frac{\partial E}{\partial net_k} = \left(\frac{\partial E}{\partial z_k}\right)\left(\frac{\partial z_k}{\partial net_k}\right)$$

Rous

EEL5840: Elements of Machine Intelligence

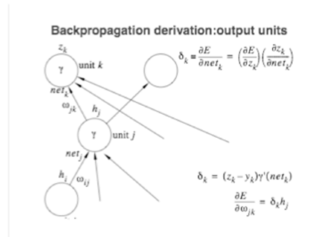
Introduction to Neural Networks



Rous

EEL5840: Elements of Machine Intelligence

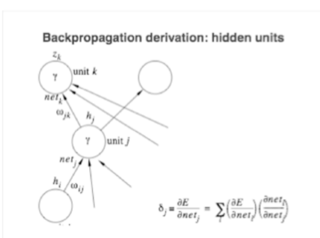
Introduction to Neural Networks



Rous

EEL5840: Elements of Machine Intelligence

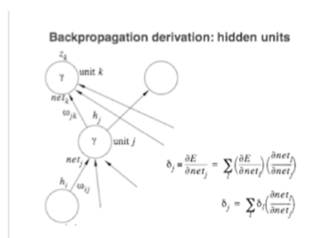
Introduction to Neural Networks



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

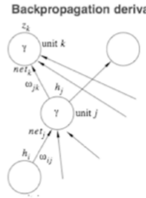


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Backpropagation derivation: hidden units



$$\delta_j = \sum_l \delta_l \left(\frac{\partial net_l}{\partial net_j}\right)$$

$$net_l = \sum_s \omega_{sl}\gamma(net_s)$$

$$\frac{\partial net_l}{\partial net_j} = \omega_{jl}\gamma'(net_j)$$

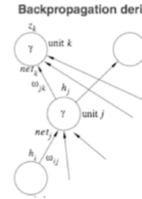
$$\delta_j = \sum_l \delta_l \omega_{jl}\gamma'(net_j)$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Backpropagation derivation: hidden units



$$\delta_j = \left(\sum_l \delta_l \omega_{jl}\right)\gamma'(net_j)$$

$$\frac{\partial E}{\partial \omega_{ij}} = \delta_j h_i$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Backpropagation summary

Output units:

$$\frac{\partial E}{\partial \omega_{jk}} = \delta_k h_j$$

$$\delta_k = (z_k - y_k)\gamma'(net_k)$$

Hidden units:

$$\frac{\partial E}{\partial \omega_{ij}} = \delta_j h_i$$

$$\delta_j = \left(\sum_l \delta_l \omega_{jl}\right)\gamma'(net_j)$$

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Basic steps in using neural networks

1. Collect training data
2. Preprocess training data
3. Select neural network architecture
4. Select learning algorithm
5. Weight initialization

6. Forward pass
7. Backward pass
8. Repeat steps 6 and 7 until satisfactory model is reached.

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

The Forward Pass

1. Apply an input vector $\mathbf{x}_i$ to network.
2. Compute the net input to each hidden unit (net_j).
3. Compute the hidden-unit outputs (h_j),
4. Compute the neural network outputs (z_k).

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

The Backward Pass

1. Evaluate δ_k at the outputs, where,

   $\delta_k = \partial E / \partial net_k$

   for each output unit k.
2. Backpropagate the δ values from the outputs backwards through the neural network.
3. Compute $\partial E / \partial \omega_i$.
4. Update weights based on the computed gradient,

   $\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla E[\mathbf{w}(t)]$.

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Practical issues

1. What should your training data be?
   - Sufficient training data?
   - Biased training data?
   - Deterministic/stochastic task?
   - Stationary/non-stationary?
2. What should your neural network architecture be?
3. Preprocessing of data.
4. Weight initialization — why small, random values?

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Practical issues (continued)

5. Selecting the learning parameter

In gradient descent:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla E[\mathbf{w}(t)]$$

what should η be?

Difficult question to answer...

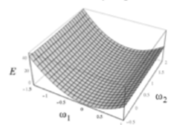
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Selecting the learning parameter: an example

Sample error surface: $E = 20\omega_1^2 + \omega_2^2$ **(realistic?)**



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Selecting the learning parameter: an example

Where is the minimum of this "error surface?"

$E = 20\omega_1^2 + \omega_2^2$

How many steps to convergence? ($\sqrt{E} < 10^{-6}$)

- Different initial weights
- Different learning rates

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Deriving the gradient descent equations

$$E = 20\omega_1^2 + \omega_2^2$$

Gradient?

$$\frac{\partial E}{\partial \omega_1} = 40\omega_1 \qquad \frac{\partial E}{\partial \omega_2} = 2\omega_2$$

Gradient descent?

$$\omega_1(t+1) = \omega_1(t) - \eta \frac{\partial E}{\partial \omega_1(t)}$$

$$\omega_1(t+1) = \omega_1(t)(1 - 40\eta)$$

$$\omega_2(t+1) = \omega_2(t)(1 - 2\eta)$$

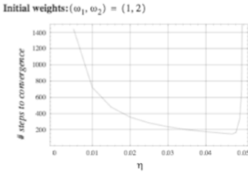
Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Convergence experiments

Initial weights: $(\omega_1, \omega_2) = (1, 2)$

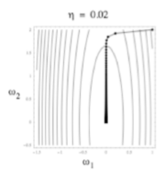


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

A closer look

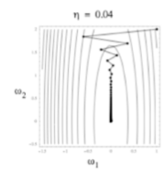


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

A closer look

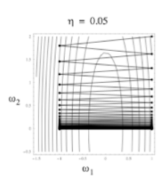


Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

A closer look



Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

What happens at $\eta > 0.5$?

Gradient descent equations:

$$\omega_1(t+1) = \omega_1(t)(1-40\eta)$$

$$\omega_2(t+1) = \omega_2(t)(1-2\eta)$$

Similar to fixed-point iteration:

$$\omega(t+1) = c\omega(t)$$

- diverges for $\|c\| > 1$, $\omega(0) \neq 0$
- converges for $\|c\| < 1$.

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Convergence of gradient descent equations

$$\omega_1(t+1) = \omega_1(t)(1-40\eta)$$

$$\omega_2(t+1) = \omega_2(t)(1-2\eta)$$

require that:

$\|1-40\eta\| < 1$

$-1 < 1-40\eta < 1$

$0 < \eta < 0.05$

Why not $\|1-2\eta\| < 1$ **?**

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Learning rate discussion

- Problematic error surfaces: "long, steep-sided valleys"
- If learning rate is too small, slow convergence. If learning rate is too large, possible divergence.
- Theoretical bounds not possible in general case (only for specific, trivial example).

Motivation for looking at more advanced training algorithms — doing more with the gradient information. Any thoughts?

Rous

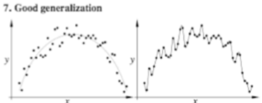
EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Practical issues (continued)

6. Pattern vs. batch training

7. Good generalization



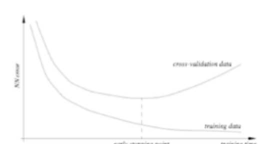
- Sufficiently constrained neural network architecture.
- Cross validation.

Rous

EEL5840: Elements of Machine Intelligence

Introduction to Neural Networks

Good generalization: Two data sets



Rous

EEL5840: Elements of Machine Intelligence

The End!