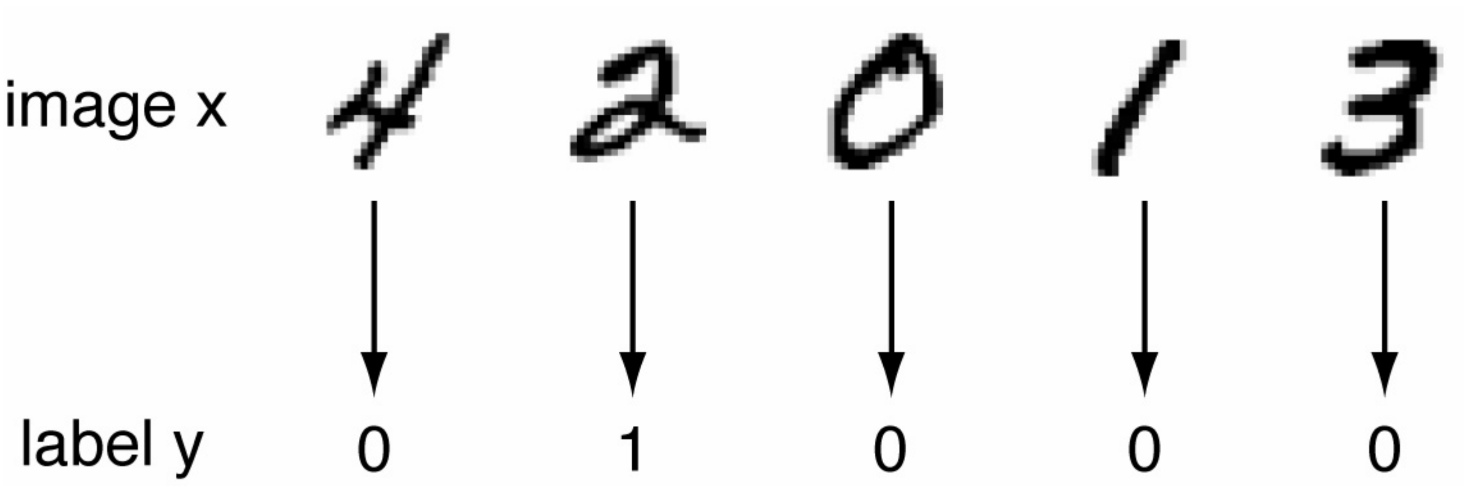
The delta rule

Sebastian Seung

Supervised learning



- How to learn this task with an LT neuron?

Delta rule

$$\Delta \mathbf{w} = \eta \left[y - H\left(\mathbf{w}^T \mathbf{x} \right) \right] \mathbf{x}$$

- Learning from mistakes.
- "delta": difference between desired and actual output.
- Also called "perceptron learning rule"

Training the bias/threshold

- The bias is like a synaptic strength for an extra input variable fixed at one.

$$\Delta b = \eta \left[y - H\left(\mathbf{w}^T \mathbf{x} + b \right) \right]$$

Two types of mistakes

- False positive $y = 0, \quad H\left(\mathbf{w}^T \mathbf{x}\right) = 1$
  - Make *w* less like *x*.

$$\Delta \mathbf{w} = -\eta \mathbf{x}$$

- False negative $y = 1, \quad H\left(\mathbf{w}^T \mathbf{x}\right) = 0$
  - Make *w* more like *x*.

$$\Delta \mathbf{w} = \eta \mathbf{x}$$

- The update is always proportional to *x.*

Online vs. batch update

- Update w after each example.

$$\Delta \mathbf{w} = \eta \left[y - H\left(\mathbf{w}^T \mathbf{x} \right) \right] \mathbf{x}$$

- Update w after the whole batch of examples.

$$\Delta \mathbf{w} = \eta \sum_a \left[y^a - H\left(\mathbf{w}^T \mathbf{x}^a \right) \right] \mathbf{x}^a$$

Margin

- The distance from an input vector **x** to the decision boundary
- For a weight vector **w** with unit norm
  - margin = **w•x**
- Large margin
  - Correct: “That was easy!”
  - Incorrect: “Not even close”

Perceptron convergence theorem

- If the examples are separable by a margin (“wiggle room”),
- Then the delta rule makes a finite number of mistakes.
  - I.e. the weight vector converges.

Proof sketch

- Let **w*** be a weight vector that separates the examples.
- Prove that **w•w*** increases faster than the norm of **w** as a function of the number of errors
- R = max length, M = min margin of input

$$N_{err} \leq \left(\frac{R}{M}\right)^2$$

Corollary

- If cycled through any finite set of examples,
- The delta rule converges to a weight vector with zero error on the set.

If examples are nonseparable

- The neuron cannot stop making errors.
- The delta rule does not converge.
- What can we say about the weight vector?

The delta rule is a gradient-based optimization algorithm.

The delta rule can be written in gradient form

$$\Delta \mathbf{w} = -\eta \frac{\partial e}{\partial \mathbf{w}}$$

$$e(\mathbf{w}, \mathbf{x}, y) = \left[y - H\left(\mathbf{w}^T \mathbf{x}\right) \right] \mathbf{w}^T \mathbf{x}$$

$$= \begin{cases} \mathbf{w}^T \mathbf{x}, & \text{false positive} \\ -\mathbf{w}^T \mathbf{x}, & \text{false negative} \\ 0, & \text{correct} \end{cases}$$

Proof: compute gradient

$$\frac{\partial e}{\partial \mathbf{w}} = \begin{cases} \mathbf{x}, & \text{false positive} \\ -\mathbf{x}, & \text{false negative} \\ 0, & \text{correct} \end{cases}$$

$$= -\left[y - H\left(\mathbf{w}^T \mathbf{x} \right) \right] \mathbf{x}$$

The delta rule is stochastic gradient descent

- For examples drawn at random from a probability distribution
- For examples drawn at random from a training set of *m* examples

$$E(\mathbf{w}) = \left\langle e(\mathbf{w}, \mathbf{x}, y) \right\rangle$$

$$E(\mathbf{w}) = \frac{1}{m} \sum_{a=1}^{m} e(\mathbf{w}, \mathbf{x}^a, y^a)$$

Closer look at the cost function

- The delta rule is a way of approximating the minimum of

$$\sum_a \left[y^a - H\left(\mathbf{w} \cdot \mathbf{x}^a\right) \right] \mathbf{w} \cdot \mathbf{x}^a$$

- The minimum is zero iff the examples are separable

Not all mistakes are equal

- The delta rule cost function penalizes mistakes by their margin

$$\sum_a \left[y^a - H\left(\mathbf{w}\cdot\mathbf{x}^a\right)\right]\mathbf{w}\cdot\mathbf{x}^a$$

- This is different from a cost function that penalizes all mistakes equally

$$\sum_a \left| y^a - H\left(\mathbf{w}\cdot\mathbf{x}^a\right)\right|$$

Batch learning

- Batch update is gradient descent on

$$\Delta \mathbf{w} = -\eta \frac{\partial E}{\partial \mathbf{w}} = -\eta \frac{1}{m}\left[y^a - H\left(\mathbf{w} \cdot \mathbf{x}^a \right) \right] \mathbf{x}^a$$

- Online learning is typically faster
- Minibatch learning (updating after every few examples) is a compromise

Summary

- Delta rule is error-driven learning
- Provably converges to zero error assuming nonzero margin
- Stochastic gradient descent
- Cost function is the average margin for erroneous examples