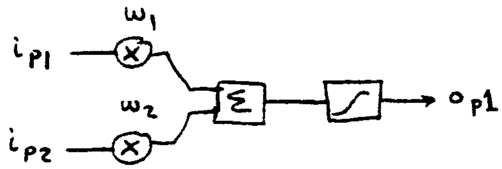
12 May 1989 **Gradient Descent** – Example – Single neuron network

Neil E Cotter

This is a general purpose technique for finding the minimum of a function. It can be summed up by saying "to find the minimum of a function, head downhill in the steepest direction." For example, the quickest way to get down from the top of a mountain is to step off a cliff.

We want to minimize error E_p by changing synaptic weights w_{ji}, w_{kj}, etc.

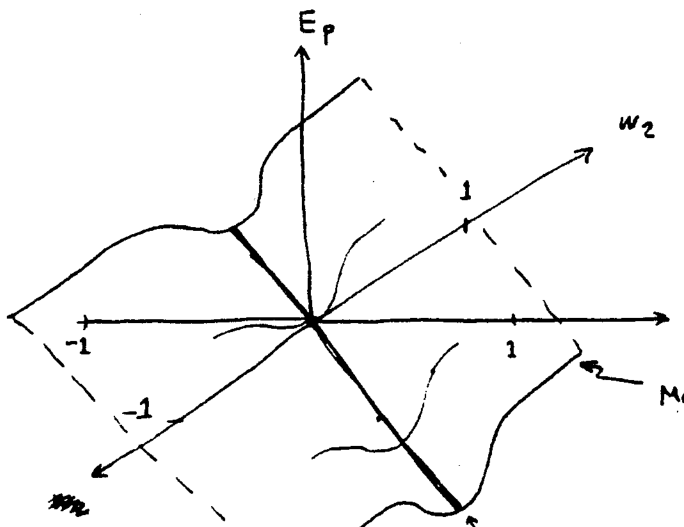
ex:



Suppose we want $i_{p1} = 1$, $i_{p2} = 1$, $o_{p1} = \frac{1}{2} \equiv t_{p1}$

We can plot $E_p = \frac{(t_{p1} - o_{p1})^2}{2} = \frac{\left(\frac{1}{2} - \frac{1}{1+e^{-w_1 i_{p1} - w_2 i_{p2}}}\right)^2}{2}$

versus w_1 and w_2:



Max possible E_p is $\frac{{\frac{1}{2}}^2}{2} = \frac{1}{8}$

$E_p = 0$ when $w_1 = -w_2$

Steepest direction is – direction of gradient $= -\underbrace{\left(\frac{\partial E_p}{\partial w_1}, \frac{\partial E_p}{\partial w_2}\right)}_{\text{gradient}}$

- *Trust me or ... think about it*

12 May 1989
Neil E Cotter

Gradient Descent - Example (cont.) - Single neuron network

Learning Rule $\Delta w_i = -\eta \dfrac{\partial E_p}{\partial w}$

For two ~~neurons~~ synapses we have $\Delta w_1 = -\eta \dfrac{\partial E_p}{\partial w_1}$

$$\Delta w_2 = -\eta \frac{\partial E_p}{\partial w_2}$$

In vector notation: $\Delta \vec{w} = -\eta \, \nabla E_p(\vec{w})$

change in weights step size gradient

- This rule steps up down the Error surface toward a minimum value of E_p.

- We use a small step size (e.g. $.05 = \eta$) to avoid getting trapped in local minima.