Foundations of Machine Learning

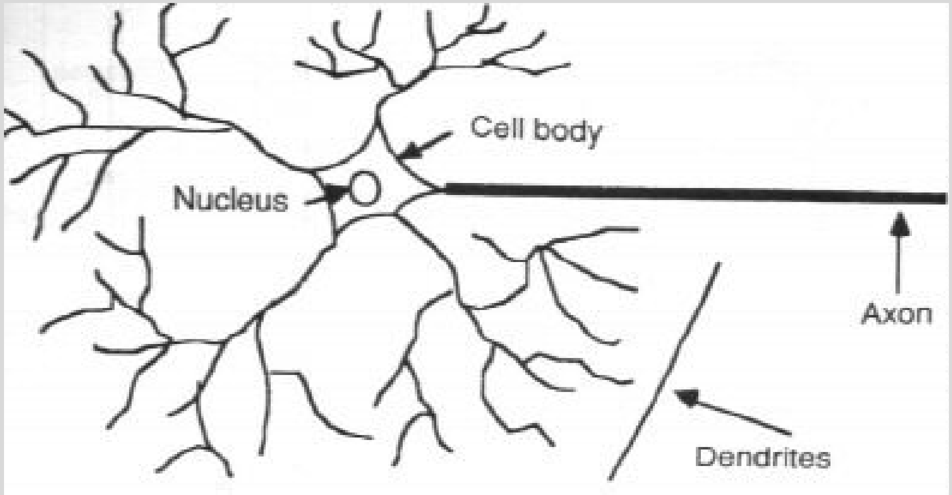
Module 6: Neural Network
Part A: Introduction

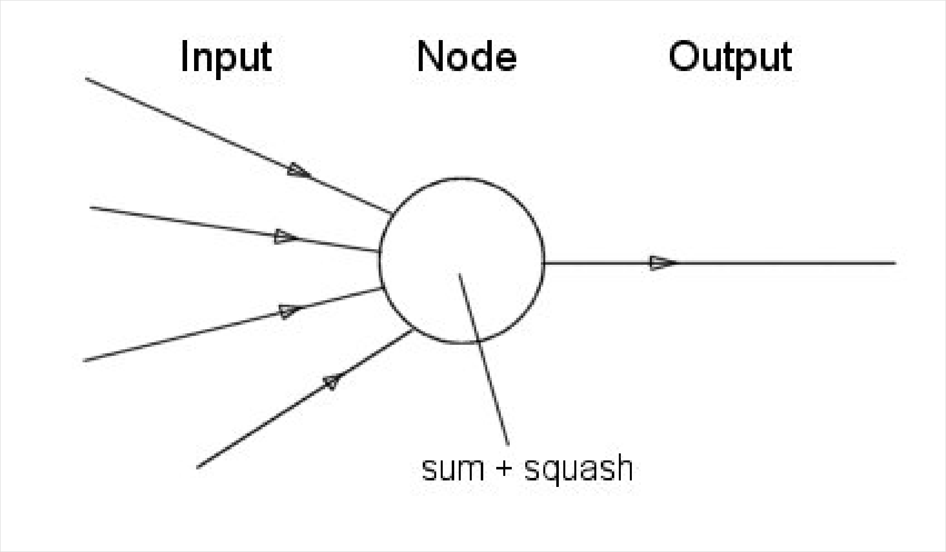
Sudeshna Sarkar
IIT Kharagpur

Introduction

- Inspired by the human brain.
- Some NNs are models of biological neural networks
- Human brain contains a massively interconnected net of 10^{10}-10^{11} (10 billion) neurons (cortical cells)
  - Massive parallelism – large number of simple processing units
  - Connectionism – highly interconnected
  - Associative distributed memory
    - Pattern and strength of synaptic connections

Neuron

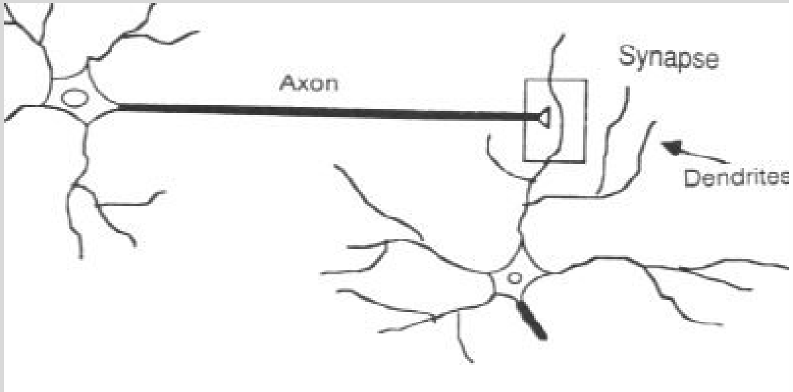


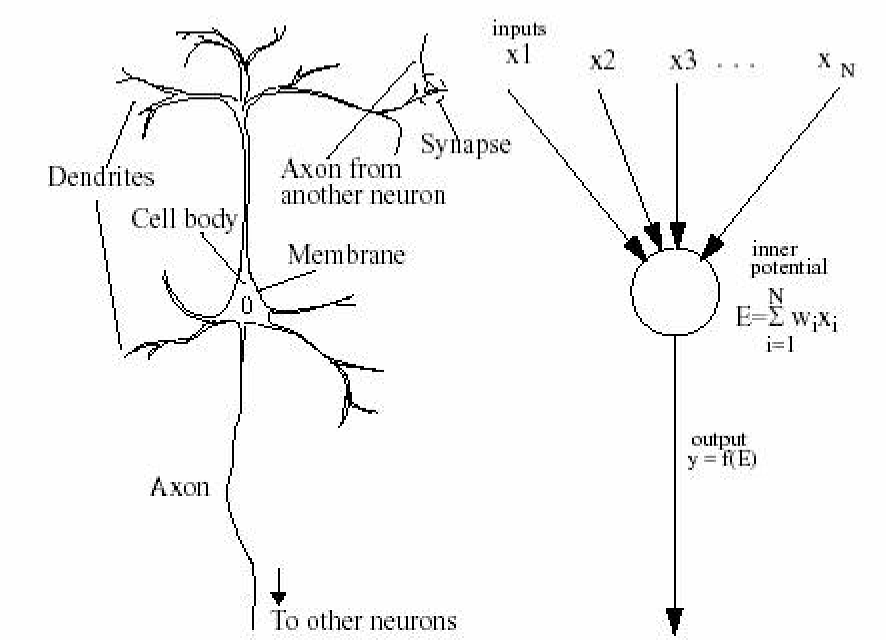


Neural Unit

ANNs

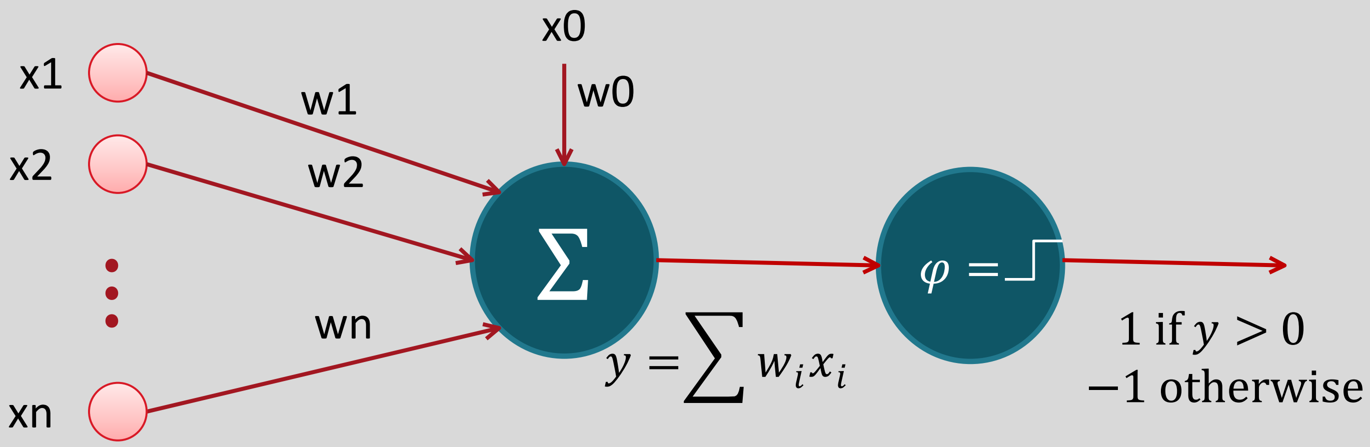
- ANNs incorporate the two fundamental components of biological neural nets:
  1. Nodes - Neurones
  2. Weights - Synapses





Perceptrons

- Basic unit in a neural network: Linear separator
  - N inputs, x1 ... xn
  - Weights for each input, w1 ... wn
  - A bias input x0 (constant) and associated weight w0
  - Weighted sum of inputs, $y = \sum_{i=0}^{n} w_i x_i$
  - A threshold function, i.e., 1 if y > 0, −1 if y <= 0



Perceptron training rule

Updates perceptron weights for a training ex as follows:

$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = \eta\,(y - \hat{y})\,x_i$$

- If the data is linearly separable and η is sufficiently small, it will converge to a hypothesis that classifies all training data correctly in a finite number of iterations

Gradient Descent

- Perceptron training rule may not converge if points are not linearly separable
- Gradient descent by changing the weights by the total error for all training points.
  - If the data is not linearly separable, then it will converge to the best fit

Linear neurons

- The neuron has a real-valued output which is a weighted sum of its inputs

$$\hat{y} = \sum_i w_i x_i = \mathbf{w}^T \mathbf{x}$$

- Define the error as the squared residuals summed over all training cases:

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

- Differentiate to get error derivatives for weights

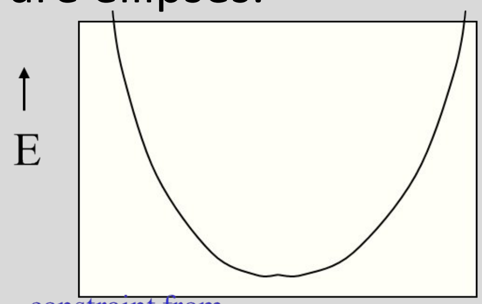
$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{j=1..m} \frac{\partial \widehat{y_j}}{\partial w_i} \frac{\partial E_j}{\partial \widehat{y_j}} = -\sum_{j=1..m} x_{i,j} (y_j - \widehat{y_j})$$

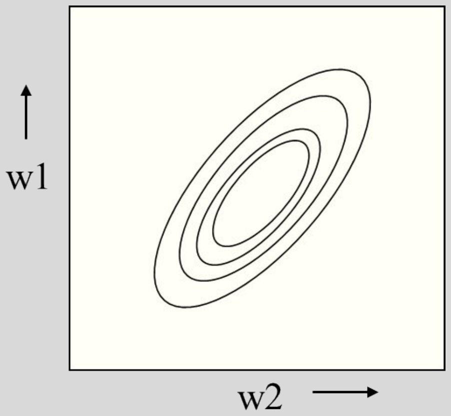
- The batch delta rule changes the weights in proportion to their error derivatives summed over all training cases

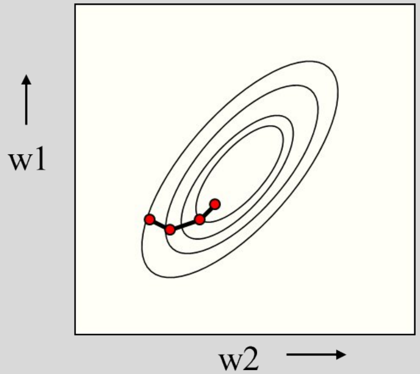
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

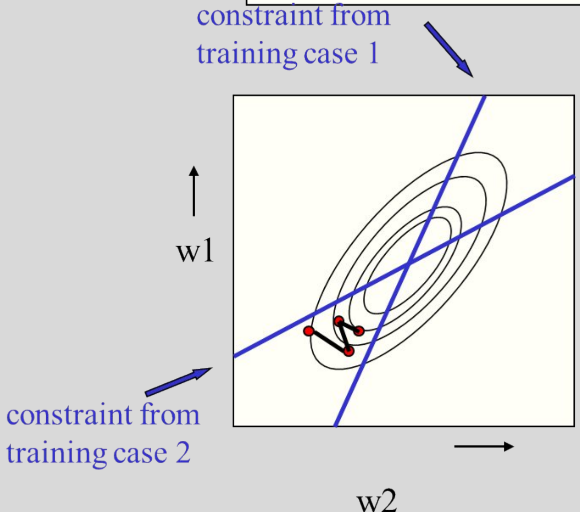
Error Surface

- The error surface lies in a space with a horizontal axis for each weight and one vertical axis for the error.
  - For a linear neuron, it is a quadratic bowl.
  - Vertical cross-sections are parabolas.
  - Horizontal cross-sections are ellipses.









Batch Line and Stochastic Learning

Batch Learning

- Steepest descent on the error surface

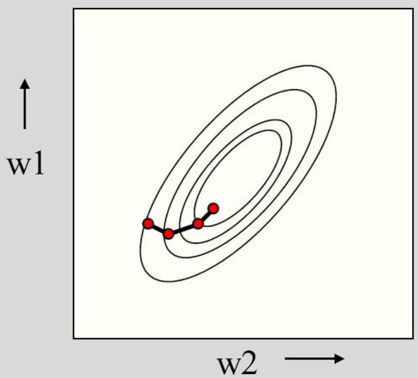
Stochastic/ Online Learning

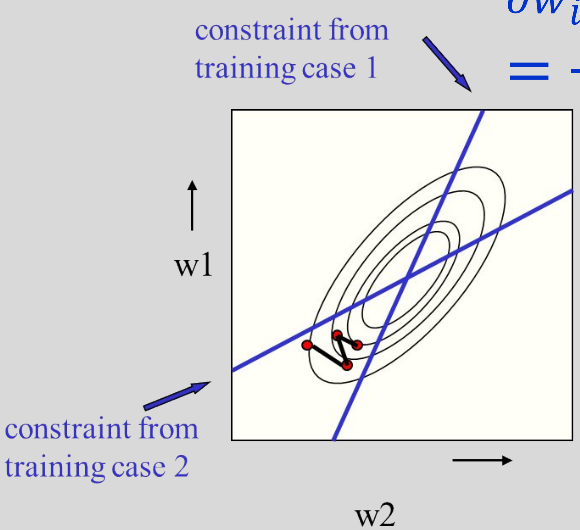
For each example compute the gradient.

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

$$\frac{\partial E}{\partial w_i} = \frac{1}{2}\frac{\partial \hat{y}}{\partial w_i}\frac{\partial E_j}{\partial \hat{y}}$$

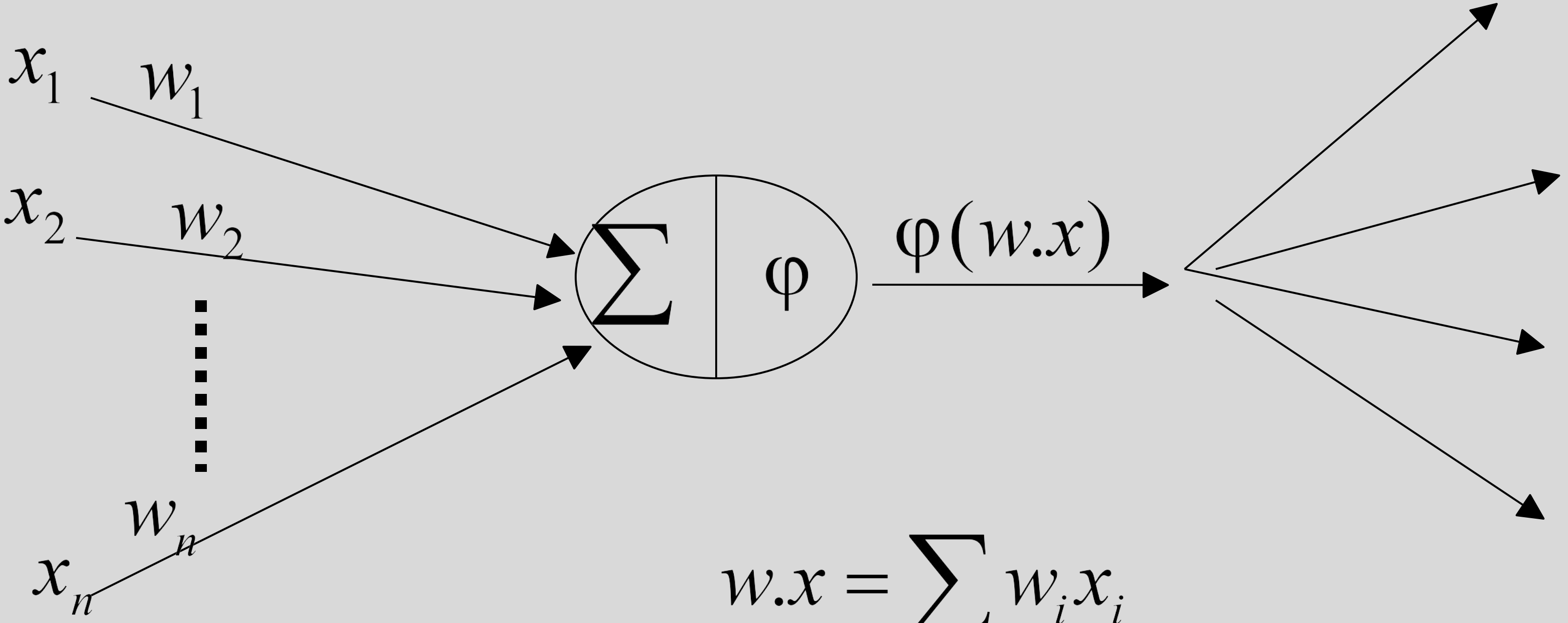
$$= -x_i(y - \hat{y})$$





Computation at Units

- Compute a 0-1 or a *graded* function of the weighted sum of the inputs
- $\varphi()$ is the *activation* function



$$w.x = \sum w_i x_i$$

Neuron Model: Logistic Unit

$$\varphi(\mathrm{z}) = \frac{1}{1+e^{-\mathrm{z}}} = \frac{1}{1+e^{-\mathrm{w.x}}}$$

$\phi'(z) = \varphi(z)(1-\varphi(z))$

$E = \frac{1}{2}\sum_d (y-\hat{y})^2 = \frac{1}{2}\sum_d (y-\varphi(w.x_d))^2$

$\frac{\partial E}{\partial w_i} = \sum_d \frac{1}{2}\frac{\partial E_d}{\partial \widehat{y_d}}\frac{\partial \widehat{y_d}}{\partial w_i}$

$= \sum_d (y_d-\widehat{y_d})\frac{\partial y}{\partial w_i}(y_d-\varphi(w.x_d))$

$= -\sum_d (y_d-\widehat{y_d})\,\varphi'(w.x_d)x_{i,d}$

$= -\sum_d \left(y_d-\widehat{y_d}\right)\widehat{y_d}(1-\widehat{y_d})x_{i,d}$

Training Rule: $\Delta w_i = \eta\sum_d \left(y_d-\widehat{y_d}\right)\widehat{y_d}(1-\widehat{y_d})x_{i,d}$

Thank You