Neural Networks

Foundations of Data Analysis

04/07/2020

Topic Outline

Basics of neural network theory and practice for supervised and unsupervised learning.

Most popular Neural Network models:

- architectures
- learning algorithms
- optimization

Neural Networks

- A NN is a machine learning approach inspired by the way in which the brain performs a particular learning task:

  - Inter neuron connection strengths (weights) are used to store the acquired information (the training examples).
  - During the learning process the weights are modified in order to correctly model the particular learning task on the training examples.

Learning

- Supervised Learning
  - Recognizing hand-written digits, pattern recognition, regression.
  - Labeled examples
    (input , desired output)
  - Neural Network models: perceptron, support vector machine.
- Unsupervised Learning
  - Find similar groups of documents in the web, content addressable memory, clustering.
  - Unlabeled examples
    (different realizations of the input alone)
  - Neural Network models: self organizing maps

Network architectures

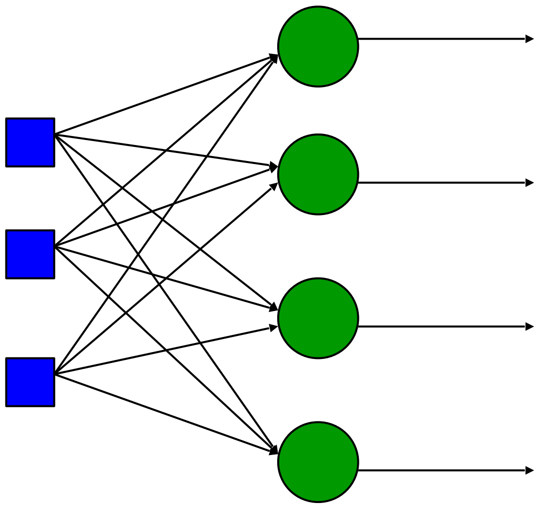
- Three different classes of network architectures

$$\left.\begin{array}{l}\text{– single-layer feed-forward}\\ \text{– multi-layer feed-forward}\\ \text{– recurrent}\end{array}\right\}\begin{array}{c}\text{neurons are organized}\\ \text{in acyclic layers}\end{array}$$

- The architecture of a neural network is linked with the learning algorithm used to train

Single Layer Feed-forward

Input nodes layer

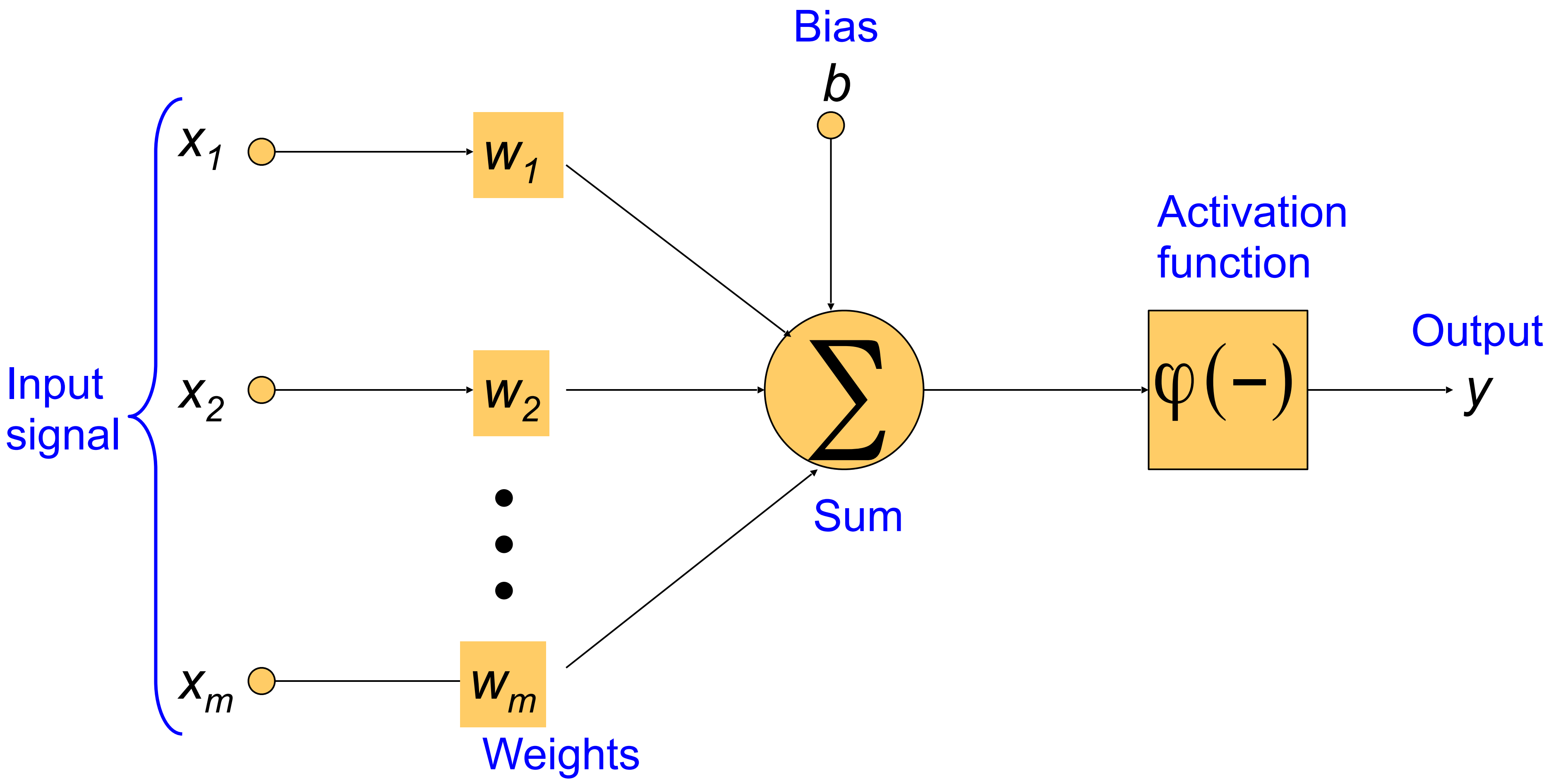


Output nodes layer

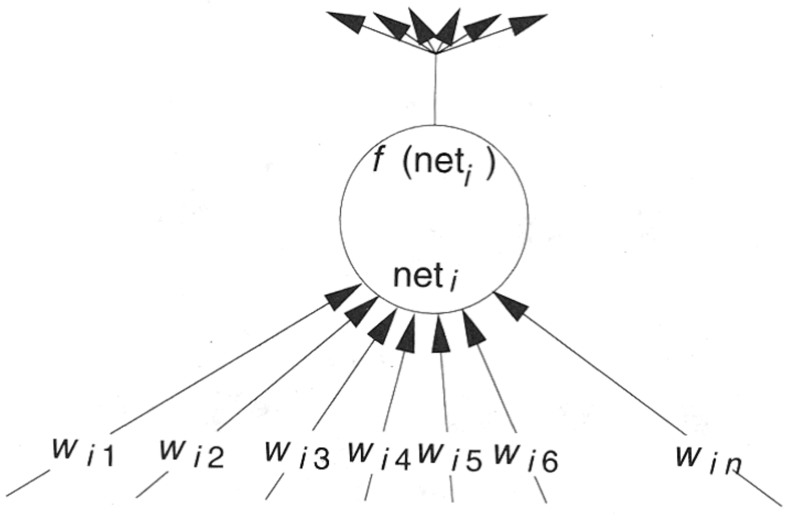
The Neuron

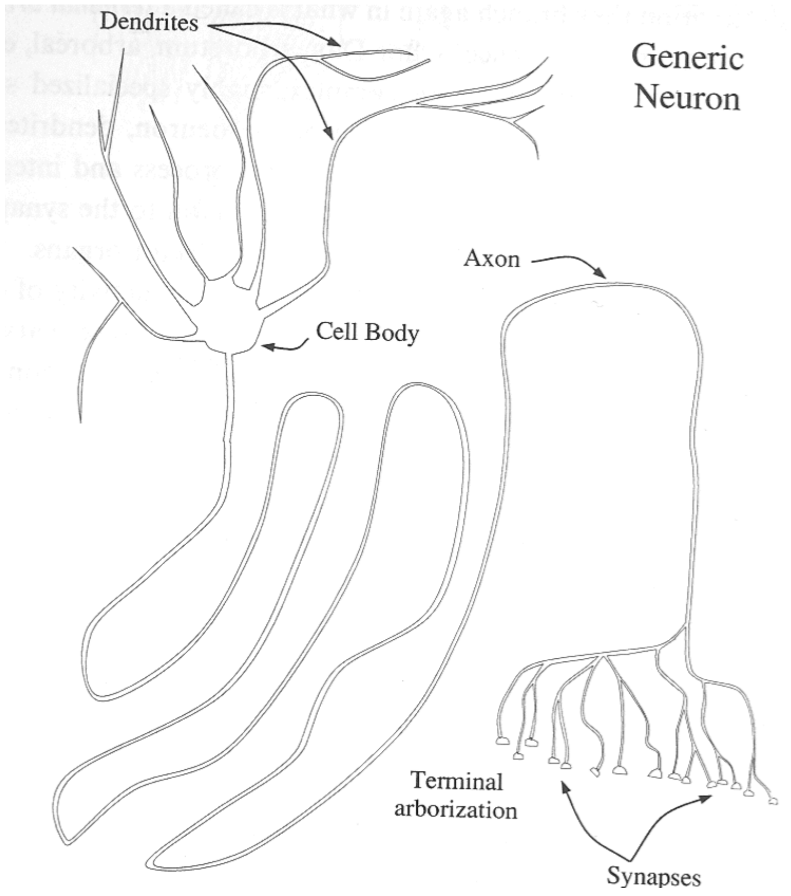
- The neuron is the basic information processing unit of a NN. It consists of :
  - A set of connecting links, each link characterized by a weight: $W_1, W_2, \ldots, W_m$
  - An add function (linear combiner) computes the weighted sum of the inputs: $u = \sum_{j=1}^{m} w_j x_j$
  - Activation function for limiting the amplitude of the output of the neuron. $y = \varphi(u + b)$

The Neuron


Bias
b
Input signal
x_1
x_2
x_m
w_1
w_2
w_m
Weights
$\sum$
Sum
Activation function
$\varphi(-)$
Output
y

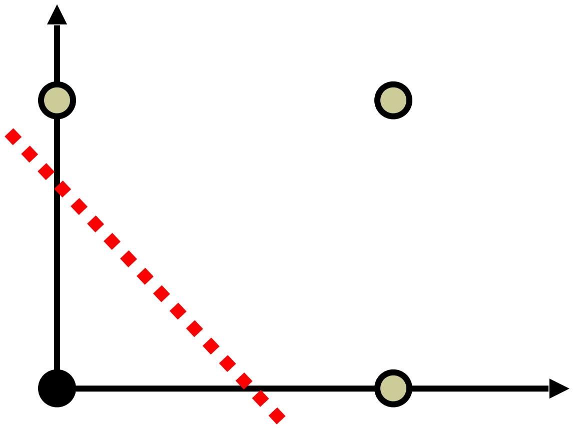
Perceptron as a Single Layer Neuron

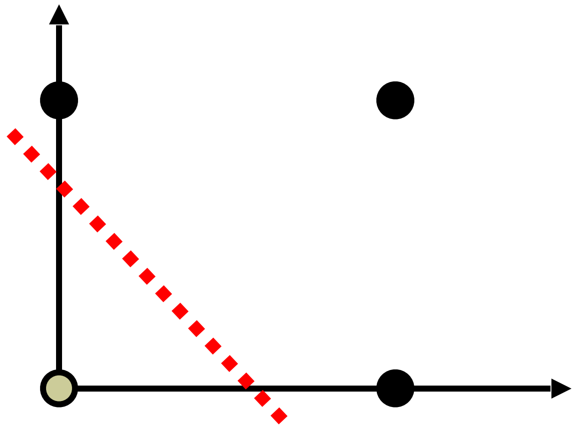




Why Multi-layer?

- Linear separable:

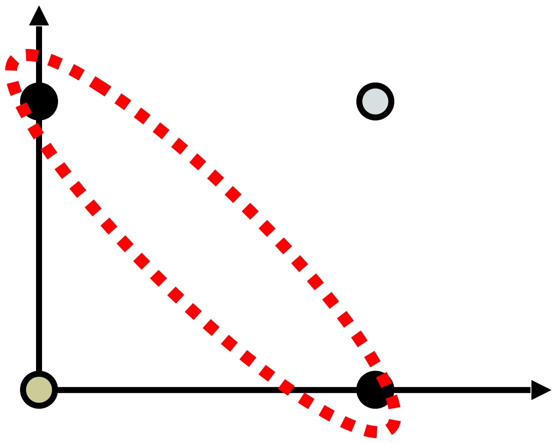




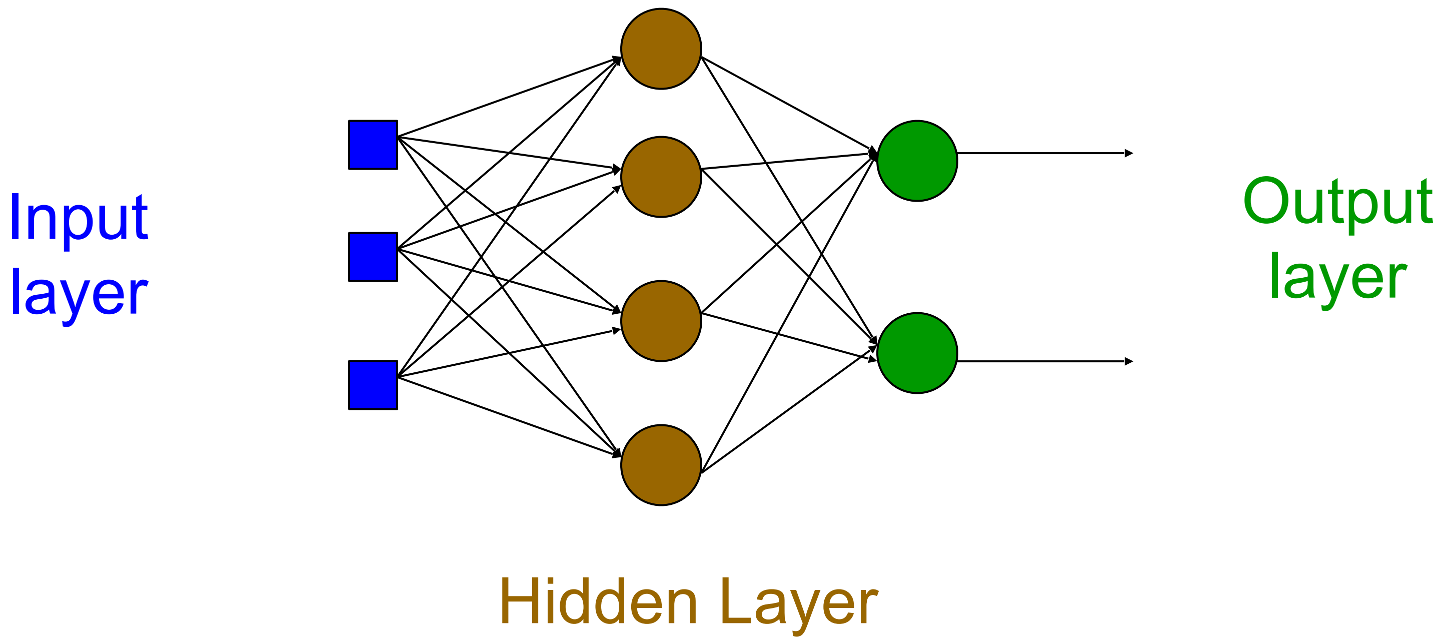
$x \lor y$

- Linear inseparable:

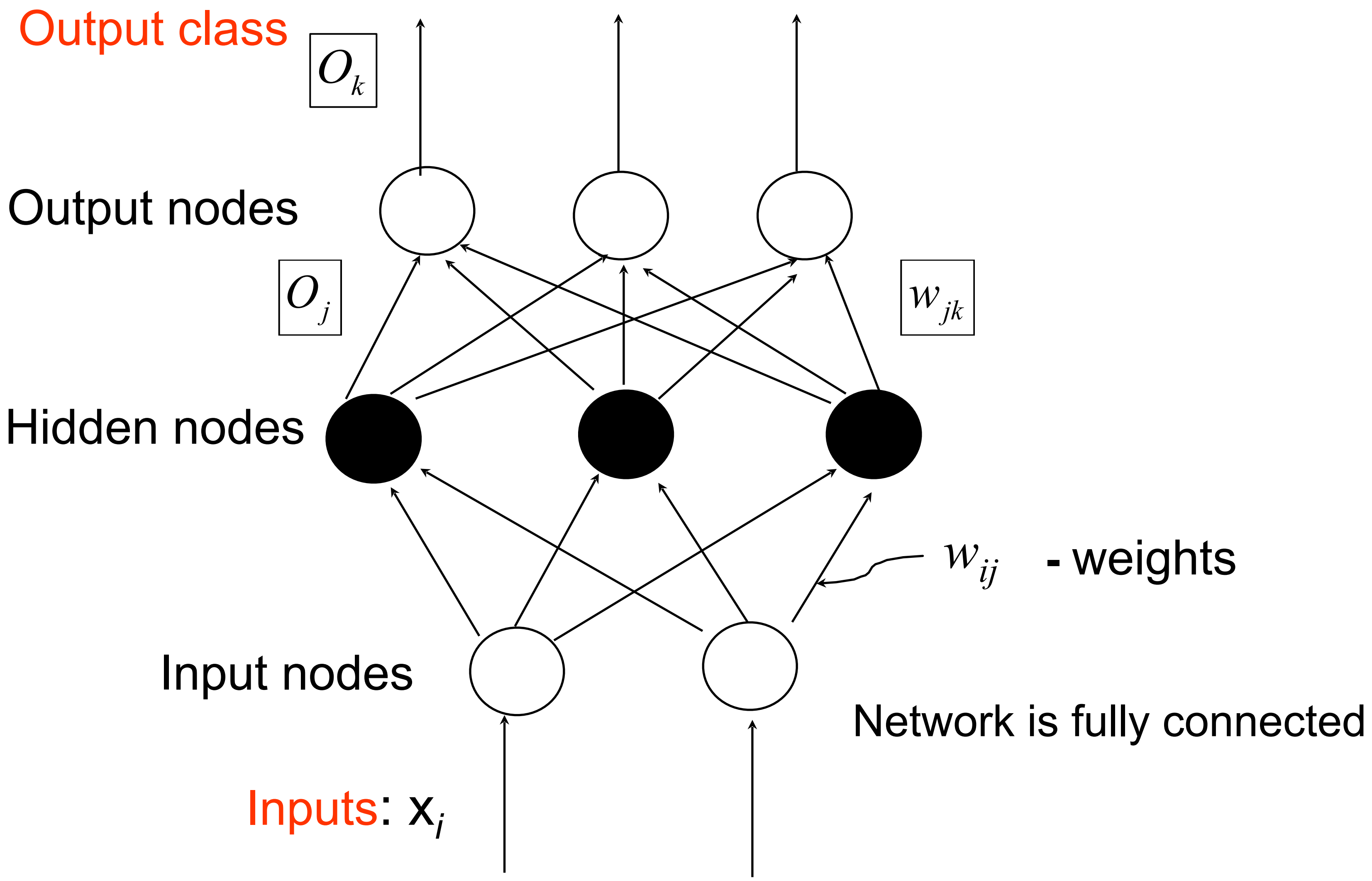
- Solution?



Multi-layer Feed-forward



Multi-layer Feed-forward



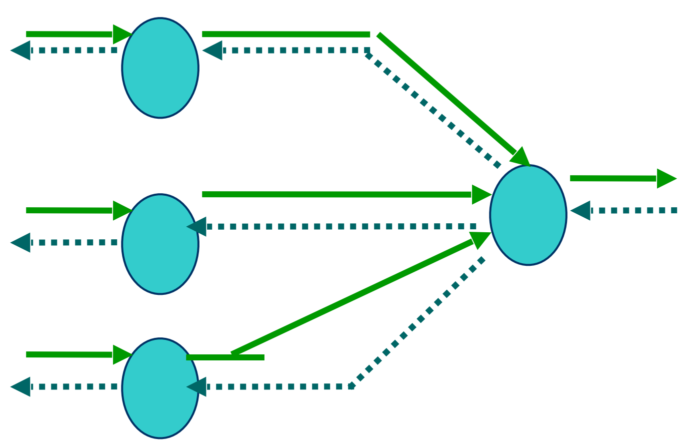
Training: Back Propagation

The back propagation algorithm

- searches for weight values that minimize the total error of the network over the set of training examples.
- consists of the repeated application of the following two passes:
  - **Forward pass**: in this step the network is activated on one example and the error of (each neuron of) the output layer is computed.
  - **Backward pass**: in this step the network error is used for updating the weights. Starting at the output layer, the error is propagated backwards through the network, layer by layer. This is done by recursively computing the local gradient of each neuron.

Training: Back Propagation

- Back-propagation training algorithm



→ *Network activation Forward Step*

⇠ *Error propagation Backward Step*

- Back propagation adjusts the weights of the NN in order to minimize the network total mean squared error

Stochastic Gradient Descent

Stochastic Gradient Descent

- Goal of machine learning :
  - Minimize expected loss (error)

$$\min_h L(h) = \mathbf{E}\left[\text{loss}(h(x), y)\right]$$

given samples

$$(x_i, y_i) \; i = 1, 2 \ldots m$$

Gradient Descent

- Process all examples together in each step

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \left(\frac{1}{n} \sum_{i=1}^{n} \frac{\partial L(w, x_i, y_i)}{\partial w} \right)$$

where L is the regularized loss function

- Entire training set examined at each step
- Very slow when *n* is very large

Stochastic Gradient Descent

- "Optimize" one example at a time
- Choose examples randomly (or reorder and choose in order)
  - Learning representative of example distribution

```
for $i = 1$ to $n$:
    $w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$
```

where L is the regularized loss function

Stochastic Gradient Descent

for $i = 1$ to n:

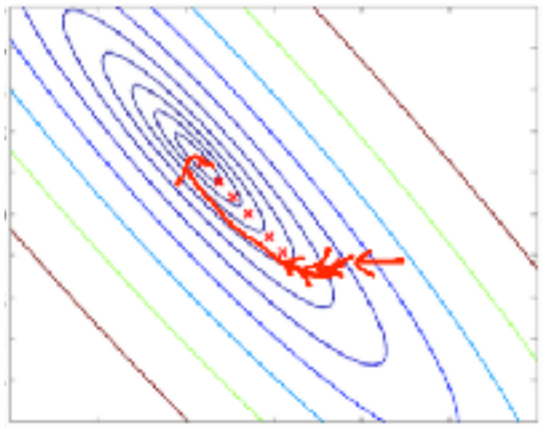
$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

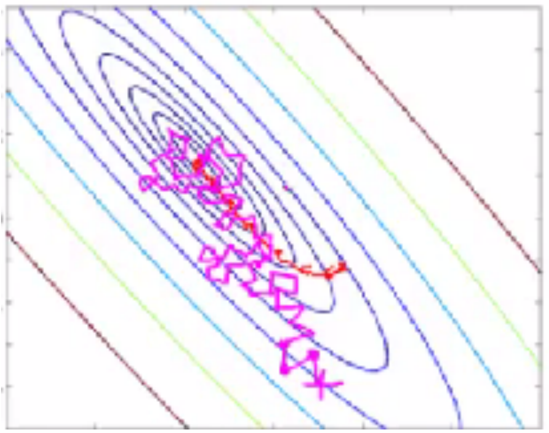
where L is the regularized loss function

- Equivalent to online learning (the weight vector *w* changes with every example)
- Convergence guaranteed for convex functions (to local minimum)

Issues of Stochastic Gradient Descent

- Convergence very sensitive to the learning rate (oscillations near solution due to probabilistic nature of sampling)
  - Might need to decrease with time to ensure the algorithm converges eventually
- Basically – SGD is good for machine learning with large data sets!


GD


SGD

Network Parameters

- How are the weights initialized?
- How many hidden layers and how many neurons?
- How many examples in the training set?

Weights

- In general, initial weights are randomly chosen, with typical values between -1.0 and 1.0 or -0.5 and 0.5.

- There are two types of NNs. The first type is known as
  - Fixed Networks – where the weights are fixed
  - Adaptive Networks – where the weights are changed to reduce prediction error.

Size of Training Data

- Rule of thumb:
  - the number of training examples should be at least five to ten times the number of weights of the network.

- Other rule:

$$\boxed{N > \frac{|W|}{(1 - a)}}$$

|W|= number of weights

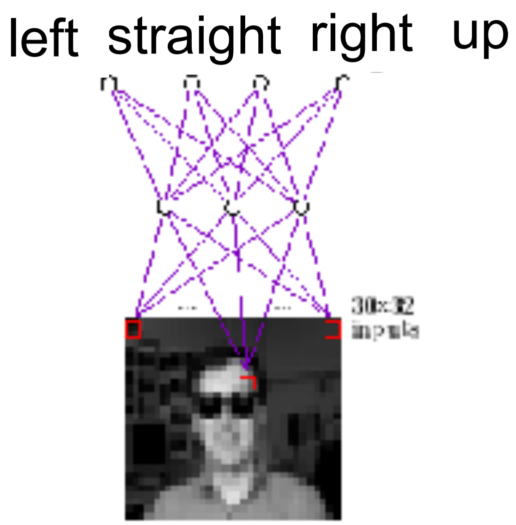
a = expected accuracy on test set

Network In Use

- Neural networks are best at identifying data patterns or trends, they are well suited for predicting:
  - sales forecasting
  - industrial process control
  - customer research
  - risk management

- NN are also used in the following specific paradigms:
  - recognition of speakers in communications
  - texture analysis
  - object / hand-written words / face recognition
  - detecting diseases from various image scans

Face Recognition



Input images

90% accurate learning head pose, and recognizing 1-of-20 faces

Disadvantages of Network

- The individual relations between the input variables and the output variables are not developed by engineering judgment so that the model tends to be a black box or input/output table without analytical basis.
- The sample size has to be large.
- Requires lot of computation so training can be time consuming.