Perceptrons

Jack Higgins

12 Jan 2016

Contents

Neural Networks

Information processing architecture loosely modelled on the brain, consisting of a large number of interconnected processing units (neurons). Generally used to model relationships between inputs and outputs or find patterns in data.

Example: SethBling's MarI/O SethBling explains how he used a Neural Network to learn to play Super Mario World.

Layers

Three types of layers

Input Layer takes inputs and passes them to other layers
Hidden Layers intermediate layers of neurons between input and output layers
Output Layer output

Perceptrons

A perceptron is a *supervised learning* algorithm for learning a binary classifier: a function that maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value):

$$f(x) = \begin{cases} 1 & if\ w \cdot x + b > 0 \\ 0 & otherwise \end{cases}$$

Multilayer Perceptrons

Back Propagation

Backpropagation [Rumelhart et al., 1986] is a special case of reverse-mode AD used to compute the gradient of a multi-layer perceptron to minimize an error function when training the weights.[1]

[1] http://www.bcl.hamilton.ie/~barak/papers/toplas-reverse.pdf

Stochastic Gradient Descent

Linear Regression

Plot a line of greatest fit for the function given a set of ‘correct’ answers.

Notation

$$\begin{aligned} m &= \text{Number of training examples} \\ x's &= \text{"input" variable /features} \\ y's &= \text{"output" variable / "target" variable} \\ (x, y) &= \text{single training example} \\ (x^{(i)}, y^{(i)}) &= i^{th} \text{ training example} \end{aligned}$$

$$\begin{aligned} h &= \text{hypothesis function} :: x's \mapsto y's \\ h_\theta(x) &= \theta_0 + \theta_1 x \end{aligned}$$

$h_\theta(x)$ predicts if `y` is a linear function of `x`.

Cost Function

How to fit the best possible line to an equation. To do this, we

$$\begin{aligned} &minimise_{\theta_0\theta_1} J(\theta_0, \theta_1) \\ &\qquad \text{where } J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} h_\theta \left(x^{(i)}) - y^{(i)} \right)^2 \end{aligned}$$

$J(\theta_0, \theta_1)$ is the **cost function** or **squared error function**

Exam Questions

2015 Autumn

You implement a simple batch gradient descent optimization routine. You have tested the gradient calculation against numeric differences, and are positive that you're getting the right gradient. But when you run the optimizer, after a while you notice that the training set error (which you print every iteration) has stopped going down and started to go up! What will you do, and (briefly) why?

2015 Autumn

List three advantages of a multilayer perceptron trained using a perceptron trained using back propagation vs a Support Vector Machine for a two-class binary discrimination task. Describe in particular the memory requirements of a fielded system, and the difficulty of adapting the classifier to new labelled data.