Solve the following system of linear equations

$Ax = b$

where the n-by-n matrix A is symmetric (i.e. $A^T = A$), positive definite

(i.e. $x^TAx > 0$ for all non-zero vectors x in R^n), and real.

Conjugate vectors

$$uAv = 0$$

$$\langle u, v \rangle_A$$

$$= \langle v, u \rangle_A = 0$$

Suppose that $\{p_k\}$ is a sequence of n mutually conjugate directions.

Then the p_k form a basis of R_n,
so we can expand the solution of
$Ax = b$

in this basis:

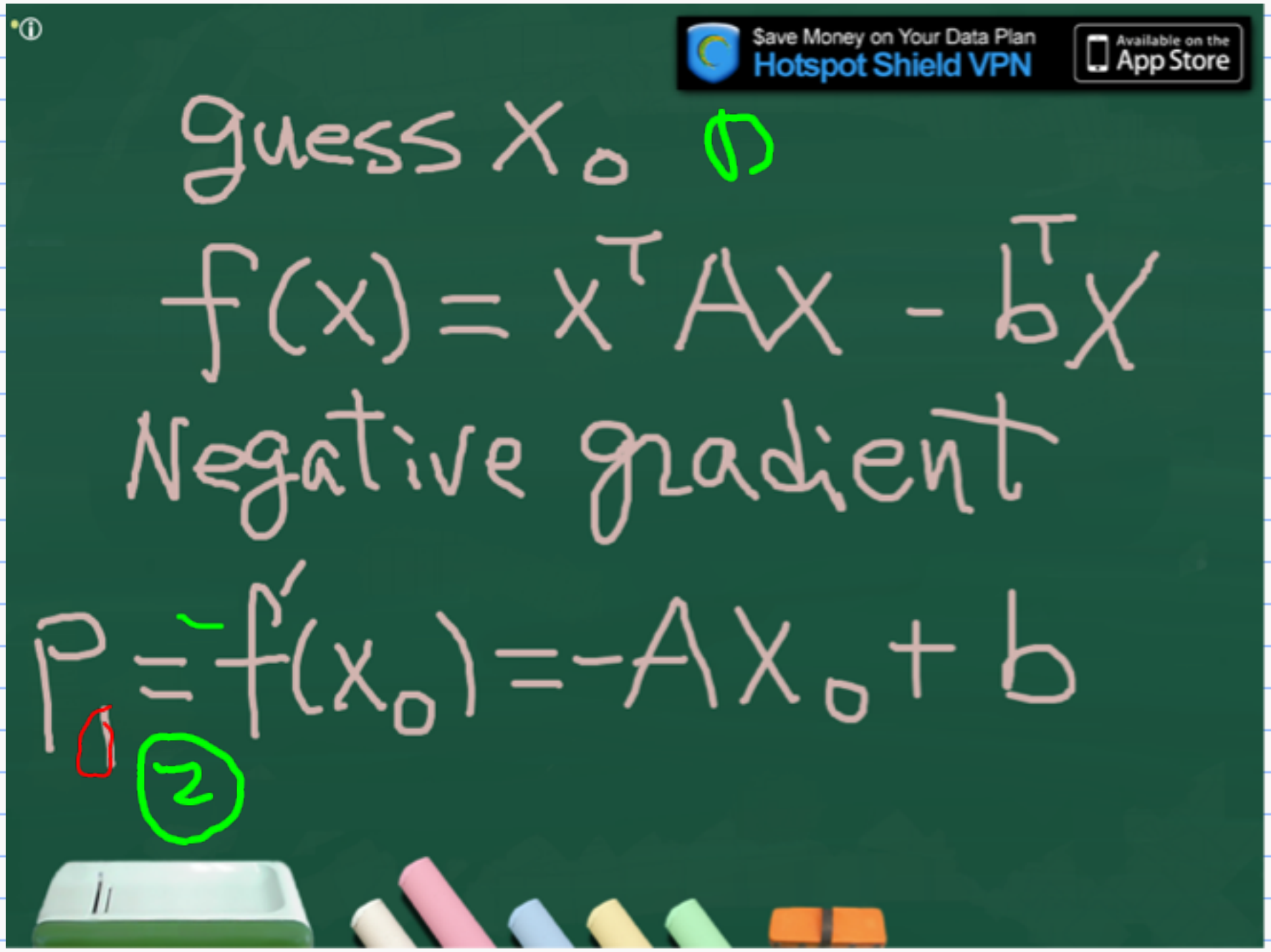
$$x_* = \sum_{k=0}^{n} \alpha_k p_k$$

$$b = Ax_* = \sum_{k=0}^{n} \alpha_k A p_k$$

$$P_k^T b = \sum_{i=b}^{n} P_k^T d_i A P_i$$

$$= d_k P_k^T A P_k$$

$$d_k = P_k^T b / \langle P_k, P_k \rangle_A$$



eg2

$$x_{k+1} = x_k + \alpha_{k+1} p_{k+1}$$

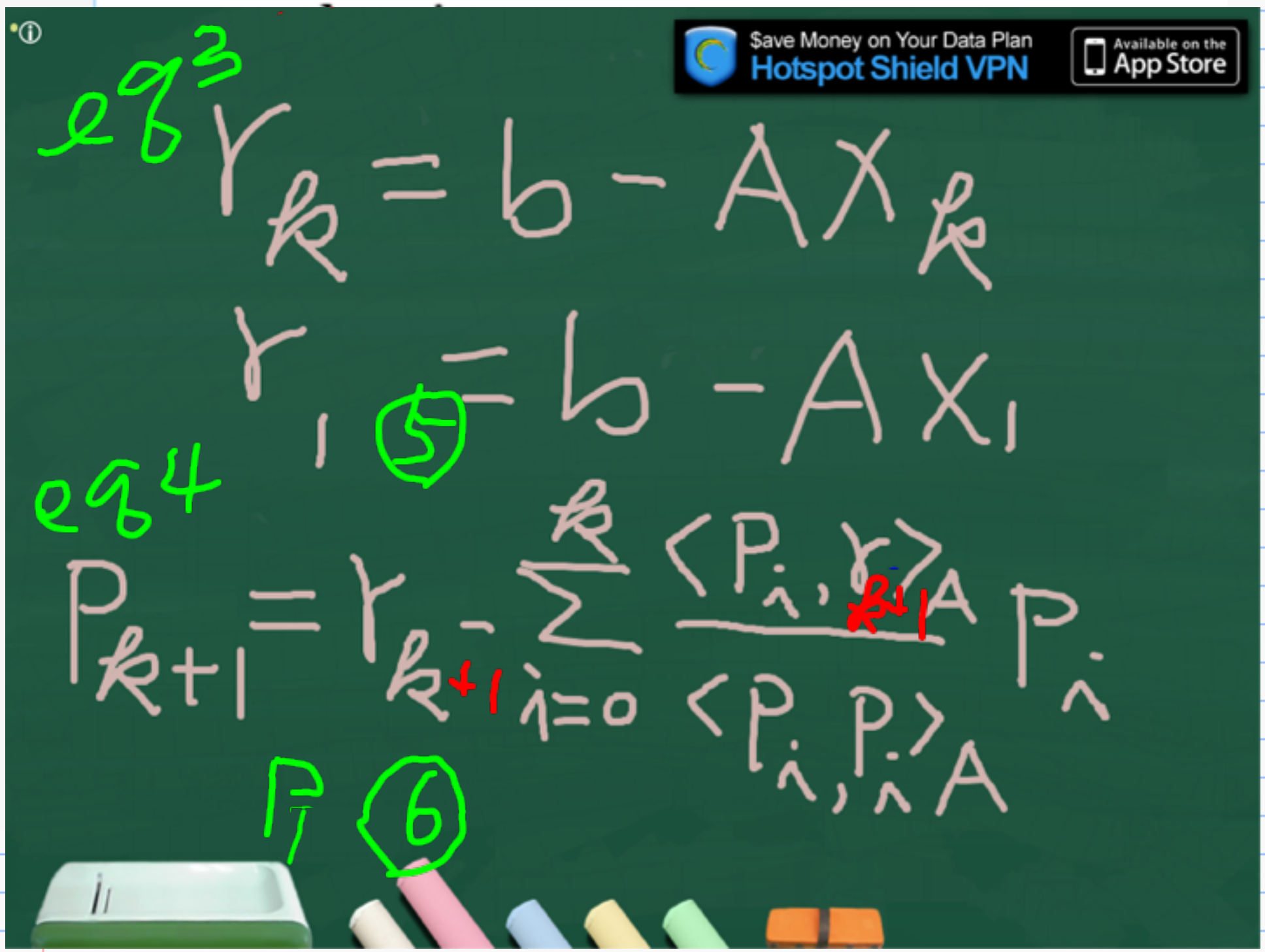
$$x_1 = x_0 + \alpha_0 p_0 \quad (k=0)$$

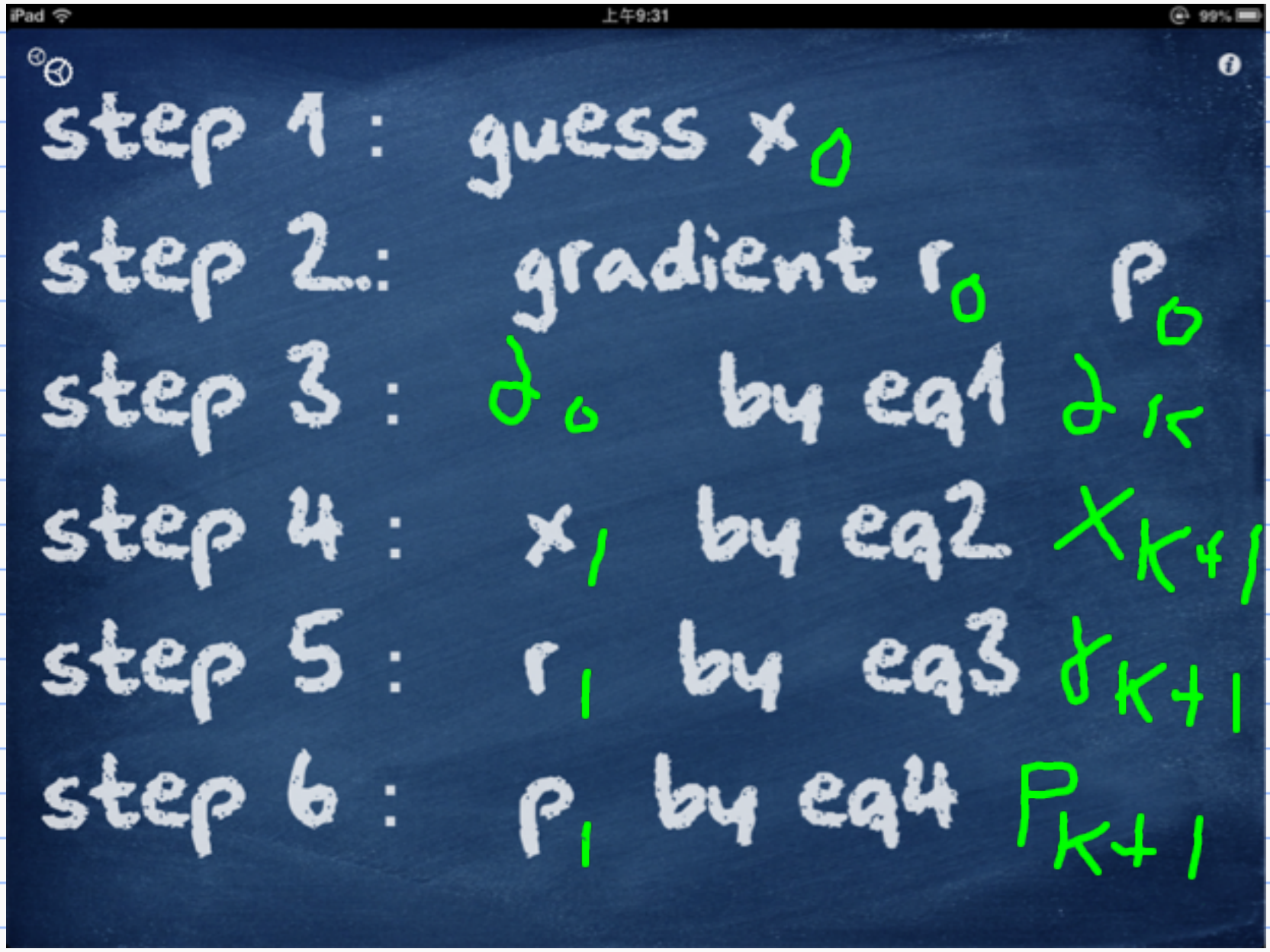
(4)

eg1

$$\alpha_0 = \frac{p_0^T b}{\langle p_0, p_0 \rangle_A}$$

(3)




iPad
上午9:31
99%
step 1 : guess x_0
step 2.: gradient r_0 p_0
step 3 : α_0 by eq1 α_{k+1}
step 4 : x_1 by eq2 x_{k+1}
step 5 : r_1 by eq3 β_{k+1}
step 6 : p_1 by eq4 p_{k+1}

EQ1

$$\alpha_k = \frac{\langle p_k, r_k \rangle}{\langle p_k, p_k \rangle_A}$$

EQ2

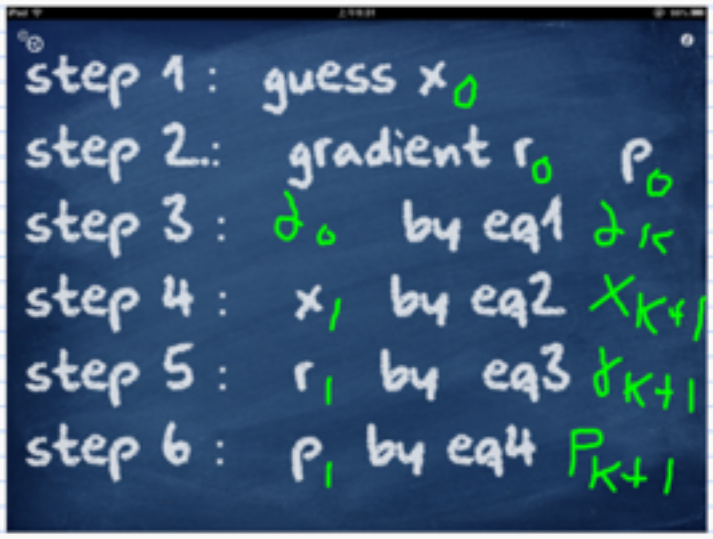
$$x_{k+1} = x_k + \alpha_k p_k$$

EQ3

$$r_{k+1} = b - A x_{k+1}$$
$$= r_k - A \alpha_k p_k$$

EQ4

$$p_{k+1} = r_{k+1} - \sum_{i \le k} \frac{\langle p_i, r_{k+1} \rangle_A}{\langle p_i, p_i \rangle_A} p_i$$



3G 下午2:40

en.wikipedia.org/wiki/Conjugate_gradient_method

indbook > 商品簡介 > 戰後臺灣政治案件—湯守仁案史料彙編 × Conjugate gradien

$$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$$
$$\mathbf{p}_0 := \mathbf{r}_0$$
$$k := 0$$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}$$
$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p}_k$$
$$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$$

if $\mathbf{r}_{k+1}$ is sufficiently small **then** exit

$$\beta_k := \frac{\mathbf{r}_{k+1}^{\mathrm{T}}\mathbf{r}_{k+1}}{\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k}$$
$$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$$
$$k := k + 1$$

end repeat

questions

Are all ρ_k conjugate?

Does the algorithm exactly find all ρ_k defined by eq1-4?

Flow chart?

How to implement the algorithm by Matlab codes?

Are your codes correct?

How to extend for solving nonlinear systems?

Training a neural network with conjugate gradient methods

This paper appears in:
Neural Networks, 1995. Proceedings., IEEE International Conference on
Date of Conference: Nov/Dec 1995
Author(s): Towsey, M.
Dept. of Biophys., United Arab Emirates Univ., Al-Ain
Alpsan, D. ; Szirtha, L.
Volume: 1
Page(s): 373 - 378 vol.1
Product Type: Conference Publications

ABSTRACT

This study investigates the use of several variants of conjugate gradient (CG) optimisation and line search methods to accelerate the convergence of an MLP neural network learning two medical signal classification problems. Much of the previous work has been done with artificial problems which have little relevance to real

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 5, NO. 6, NOVEMBER 1994

Brief Papers

Training Feedforward Networks with the Marquardt Algorithm

Martin T. Hagan and Mohammad B. Menhaj

***Abstract*— The Marquardt algorithm for nonlinear least squares is presented and is incorporated into the backpropagation algorithm for training feedforward neural networks. The algorithm is tested on several function approximation problems, and is compared with a conjugate gradient algorithm and a variable learning rate algorithm. It is found that the Marquardt algorithm is much more efficient than either of the other techniques when the network contains no more than a few hundred weights.**

I. Introduction

SINCE the backpropagation learning algorithm [1] was first popularized, there has been considerable research on methods to accelerate the convergence of the algorithm. This research falls roughly into two categories. The first category involves the development of ad hoc techniques (e.g., [2]–[5]). These techniques include such ideas as varying the learning rate, using momentum and rescaling variables. Another category of research has focused on standard numerical optimization techniques (e.g., [6]–[9]).

The most popular approaches from the second category have used conjugate gradient or quasi-Newton (secant) methods. The quasi-Newton methods are considered to be more efficient, but their storage and computational requirements go up as the square of the size of the network. There have been some limited memory quasi-Newton (one step secant) algorithms that speed up convergence while limiting memory requirements [8,10]. If exact line searches are used, the one step secant methods produce conjugate directions.

Another area of numerical optimization that has been applied to neural networks is nonlinear least squares [11]–[13]. The more general optimization methods were designed to work effectively on all sufficiently smooth objective functions. However, when the form of the objective function is known it is often possible to design more efficient algorithms. One particular form of objective function that is of interest for neural networks is a sum of squares of other nonlinear functions. The minimization of objective functions of this type is called

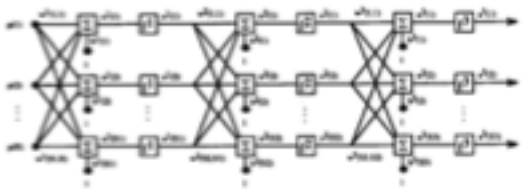


Fig. 1. Three-layer feedforward network.

This paper presents the application of a nonlinear least squares algorithm to the batch training of multi-layer perceptrons. For very large networks the memory requirements of the algorithm make it impractical for most current machines (as is the case for the quasi-Newton methods). However, for networks with a few hundred weights the algorithm is very efficient when compared with conjugate gradient techniques. Section II briefly presents the basic backpropagation algorithm. The main purpose of this section is to introduce notation and concepts which are needed to describe the Marquardt algorithm. The Marquardt algorithm is then presented in Section III. In Section IV the Marquardt algorithm is compared with the conjugate gradient algorithm and with a variable learning rate variation of backpropagation. Section V contains a summary and conclusions.

II. Backpropagation Algorithm

Consider a multilayer feedforward network, such as the three-layer network of Fig. 1.

The net input to unit i in layer $k+1$ is

$$n^{k+1}(i) = \sum_{j=1}^{Sk} w^{k+1}(i,j)a^k(j) + b^{k+1}(i). \quad (1)$$

The output of unit i will be

$$a^{k+1}(i) = f^{k+1}\left(n^{k+1}(i)\right). \quad (2)$$

For an M layer network the system equations in matrix form