Some mathematical observations on the Naive Discriminative Learner: Rescorla-Wagner vs. single-layer perceptron vs. least-squares regression

Stefan Evert
FAU Erlangen-Nürnberg

17 August 2015

Contents

1 Introduction

Naive Discriminative Learning (NDL, Baayen 2011) performs linguistic classification on the basis of direct associations between cues and outcomes, which are learned incrementally according to the Rescorla-Wagner (R-W) equations (Rescorla and Wagner 1972). Danks (2003) argued that if the R-W equations successfully acquire the true associations between cues and outcomes, they should approximate an equilibrium state in which the expected change in association values is zero if a cue-outcome event is randomly presented to the learner. This equilibrium state can be computed directly by solving a matrix equation, without carrying out many iterations of R-W updates, making NDL attractive as an efficient learning technique for quantitative linguistics. Use of the Danks equilibrium also circumvents the problem that a simulation of the R-W model does not converge to a single state unless the learning rate is gradually reduced.

It is well known that the R-W model is closely related to neural networks (through the "delta rule" for gradient-descent training of a single-layer perceptron) and to linear least-squares regression (e.g. Danks 2003; Baayen 2011). However, most authors do not seem to be aware of the true depth of these similarities and of their implications.

In this paper, we show that the R-W equations are identical to gradient-descent training of a single-layer feed-forward neural network, which we refer to as a single layer perceptron (SLP[1]) here (Sec. 3). Based on this result, we present a new, simpler derivation of the

[1] The term SLP is often reserved for a particular form of such a single-layer network using a Heavyside activation function (cf. `https://en.wikipedia.org/wiki/Perceptron`). Here, we use it more generally to refer to any single-layer feed-forward network.

equilibrium conditions (Danks 2003) and prove that they correspond to the solution of a linear least-squares regression problem (Sec. 4). In Sec. 5 we discuss some consequences of these new insights.

2 The Rescorla-Wagner equations

This section gives a mathematically precise definition of the R-W model, following the notation of Danks (2003). The purpose of the R-W equations is to determine associations between a set of cues $C_1, \dots, C_n$ and a single outcome O in a population of event tokens $(\mathbf{c}^{(t)}, o^{(t)})$, where $\mathbf{c}^{(t)} = (c_1^{(t)}, \dots, c_n^{(t)})$ is a vector of cue indicators for event t and $o^{(t)}$ an indicator for the outcome O. Formally,

$$c_i^{(t)} = \begin{cases} 1 & \text{if } C_i \text{ is present} \\ 0 & \text{otherwise} \end{cases} \qquad o^{(t)} = \begin{cases} 1 & \text{if } O \text{ results} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for $t = 1, \dots, m$ (and $m = \infty$ can be allowed).

When presented with an event $(\mathbf{c}, o)$, the R-W equations update the associations V_i between cues and the outcome according to Eq. (2), which is a more formal notation of Eq. (1) from Danks (2003, 111).

$$\Delta V_i = \begin{cases} 0 & \text{if } c_i = 0 \\ \alpha_i \beta_1 \big(\lambda - \sum_{j=1}^n c_j V_j\big) & \text{if } c_i = 1 \wedge o = 1 \\ \alpha_i \beta_2 \big(0 - \sum_{j=1}^n c_j V_j\big) & \text{if } c_i = 1 \wedge o = 0 \end{cases} \tag{2}$$

Here, α_i is a measure of the salience of cue C_i, β_1 and β_2 are learning rates for positive ($o = 1$) and negative ($o = 0$) events, and λ is the maximal activation level of the outcome O. A simplified form of the R-W equations proposed by (Widrow and Hoff 1960) assumes that $a_1 = \cdots = a_n = 1$, $\beta_1 = \beta_2 = \beta$ and $\lambda = 1$ (known as the W-H rule).

Danks (2003) argues that a successful R-W model should approach an equilibrium state of the association vector $\mathbf{V} = (V_1, \dots, V_n)$ where the expected update $E[\Delta V_i] = 0$ if a random event token is sampled from the population. If we make the simplifying assumption that $\beta_1 = \beta_2 = \beta$, this condition corresponds to the equality

$$\lambda \frac{1}{m} \sum_{t=1}^{m} c_i^{(t)} o_i^{(t)} - \sum_{j=1}^{n} V_j \frac{1}{m} \sum_{t=1}^{m} c_i^{(t)} c_j^{(t)} = 0 \tag{3}$$

In Danks's notation, Eq. (3) can be written as

$$\lambda P(O, C_i) - \sum_{j=1}^{n} V_j P(C_i, C_j) = 0 \tag{4}$$

and is equal to his Eq. (11) (Danks 2003, 113) multiplied by $P(C_i)$.

3 R-W and the single layer perceptron

We will now formulate a single-layer feed-forward neural network (SLP) whose learning behaviour – with gradient-descent training, which corresponds to the backprop algorithm for a SLP and is also known as the "delta rule" in this case – is identical to the R-W equations with equal positive and negative learning rates $\beta_1 = \beta_2$ (but no other restrictions). The SLP requires a slightly different representation of events as pairs $(\mathbf{x}^{(t)}, z^{(t)})$ with

$$x_i^{(t)} = \begin{cases} a_i & \text{if } C_i \text{ is present} \\ 0 & \text{otherwise} \end{cases} \qquad z^{(t)} = \begin{cases} \lambda & \text{if } O \text{ results} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Here, $a_i > 0$ is a (different) measure of the salience of cue C_i and $\lambda > 0$ the maximum activation of outcome O. Note that the event representation $(\mathbf{x}, z)$ is connected to the representation $(\mathbf{c}, o)$ through the equivalences $x_i = a_i c_i$ and $z = \lambda o$. In the W-H case, the two representations are identical.

The SLP computes the activation of the outcome as a linear combination $y = \sum_{i=1}^n w_i x_i$ of the input variables, where $\mathbf{w} = (w_1, \dots, w_n)$ is the weight vector of the network. It uses a linear activation function $h(y) = y$ and a Euclidean cost function for the difference between y and the desired activation level z. The cost associated with a given event token $(\mathbf{x}, z)$ is thus

$$E(\mathbf{w}, \mathbf{x}, z) = (z - y)^2 = \left(z - \sum_{i=1}^n w_i x_i \right)^2 . \tag{6}$$

For batch updates based on the full population of event tokens, the corresponding batch cost is

$$E(\mathbf{w}) = \sum_{t=1}^m E(\mathbf{w}, \mathbf{x}^{(t)}, z^{(t)}). \tag{7}$$

If smaller batches are used, the sum j ranges over a subset of the population for each update step.

Presented with an event token $(\mathbf{x}, z)$, gradient-descent training of this SLP updates the weight vector by

$$\Delta w_i = -\delta \frac{\partial E(\mathbf{w}, \mathbf{x}, z)}{\partial w_i} \tag{8}$$

where $\delta > 0$ is the learning rate and the gradient $\partial E / \partial w_i$ is given by

$$\frac{\partial E(\mathbf{w}, \mathbf{x}, z)}{\partial w_i} = 2(z - y)(-x_i) = -2 \left(z - \sum_{j=1}^n w_j x_j \right) x_i . \tag{9}$$

Inserting the equalities $x_i = a_i c_i$ and $z = \lambda o$, we obtain

$$\Delta w_i = \begin{cases} 0 & \text{if } c_i = 0 \\ 2\delta a_i \big(\lambda - \sum_{j=1}^n c_j a_j w_j \big) & \text{if } c_i = 1 \wedge o = 1 \\ 2\delta a_i \big(0 - \sum_{j=1}^n c_j a_j w_j \big) & \text{if } c_i = 1 \wedge o = 0 \end{cases} \tag{10}$$

Comparing this with Eq. (2), we can set $V_j = a_j w_j$, i.e. we interpret the weight vector $\mathbf{w}$ of the SLP as salience-adjusted cue-outcome associations. With $\Delta V_i = a_i \Delta w_i$, we obtain

$$\Delta V_i = \begin{cases} 0 & \text{if } c_i = 0 \\ 2\delta a_i^2 \big(\lambda - \sum_{j=1}^n c_j V_j \big) & \text{if } c_i = 1 \wedge o = 1 \\ 2\delta a_i^2 \big(0 - \sum_{j=1}^n c_j V_j \big) & \text{if } c_i = 1 \wedge o = 0 \end{cases} \tag{11}$$

which is identical to the R-W equations for $\beta_1 = \beta_2 = 2\delta$ and $\alpha_i = a_i^2$.

The assumption $\beta_1 = \beta_2$ can be relaxed if we change the representation of events to

$$x_i^{(t)} = \begin{cases} a_i & \text{if } c_i = 1 \wedge o = 1 \\ a_i \sqrt{\frac{\beta_2}{\beta_1}} & \text{if } c_i = 1 \wedge o = 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

i.e. if we allow the salience of cues to differ between positive ($o = 1$) and negative ($o = 0$) events; the scaling factor β_2/β_1 is the same for all cues C_i. We do not pursue this extension further here because it affects the equilibrium state in an unpredictable way. As Danks (2003) has already observed, the cue saliences α_i have no impact at all on the equilibrium and the maximum activation level λ merely results in a linear scaling.

4 R-W and least-squares regression

We have shown in Sec. 3 that the R-W equations describe the gradient-descent training of a SLP for the linear regression problem

$$\min_{\mathbf{w}} E(\mathbf{w}) = \min_{\mathbf{w}} \sum_{t=1}^{m} E(\mathbf{w}, \mathbf{x}^{(t)}, z^{(t)}). \tag{13}$$

This equivalence holds generally, not only in the case of the simplified W-H rule. Thus, both R-W and our SLP aim to solve the same regression.

If the training procedure is successful, the weight vector $\mathbf{w}$ should approach the least-squares solution of the regression problem. With single updates (corresponding to the R-W model), convergence cannot be achieved unless the learning rate is gradually reduced. With batch updates treating the entire population as a single batch, the cost $E(\mathbf{w})$ is a convex function of $\mathbf{w}$ and the gradient descent procedure converges to its unique minimum after a sufficient number of iterations.[2]

In order to express Eq. (13) more concisely, we define an $m \times n$ matrix $\mathbf{X} = (x_i^{(t)}) = (x_{ti})$ of cues for all event tokens in the population. The rows of this matrix correspond to event tokens t, the columns to cues i; i.e. row number t contains the input vector $\mathbf{x}^{(t)}$. We also define the column vector $\mathbf{z} = (z^{(1)}, \dots, z^{(m)})$ of outcomes and recall that $\mathbf{w} = (w_1, \dots, w_n)$ is a column vector of SLP weights. The batch cost can now be written as a dot product

$$E(\mathbf{w}) = \big(\mathbf{z} - \mathbf{X}\mathbf{w}\big)^T \big(\mathbf{z} - \mathbf{X}\mathbf{w}\big). \tag{14}$$

The least-squares solution must satisfy the condition $\nabla E(\mathbf{w}) = \mathbf{0}$, which leads to the so-called normal equations

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{z}. \tag{15}$$

In the case of the W-H rule, $\mathbf{X}$ is a coincidence matrix between cues and events, with $x_{ti} \in \{0, 1\}$. A straightforward calculation shows that $\mathbf{X}^T\mathbf{X}$ is a square matrix of co-occurrence counts $f(C_i, C_j)$ between cues, and $\mathbf{X}^T\mathbf{z}$ is a vector of co-occurrence counts $f(C_i, O)$ between the cues and the outcome O. Dividing Eq. (15) by m, we obtain Eq. (4) with $\lambda = 1$, i.e.

$$P(O, C_i) - \sum_{j=1}^{n} V_j P(C_i, C_j) = 0$$

[2] In fact, the minimum of $E(\mathbf{w})$ might not be unique under certain circumstances, viz. if the correlation matrix of the cues is not positive definite; cf. **?**, 115–116 for the special case of "coextensive" cues. In order to keep the discussion straightforward, we assume the general case of a unique minimum in the present paper.

which is the same as Eq. (3) of Danks (2003) with rows multiplied by $P(C_i)$. Since linear regression is invariant wrt. the salience factors a_i (the weights are simply adjusted by reciprocal factors $1/a_i$ to achieve the same regression values) and scales linearly with λ, equivalence to the equilibrium conditions (Danks 2003, 112–114) also holds for arbitrary values of a_i and λ.

In the general case where the regression problem has a unique solution, $\mathbf{X}^T\mathbf{X}$ is symmetric and positive definite. It can therefore be inverted and the least-squares solution is given by

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z}. \tag{16}$$

Standard statistical software such as R (R Development Core Team 2010) can be used to compute $\mathbf{w}^*$ reliably and efficiently. It is not necessary to carry out the iterative training procedure of the R-W model or the neural network, and there is no need to worry about convergence of the iterative training.

5 Consequences

- We have shown that R-W association learning, a linear SLP neural network and linear regression are fully equivalent and should ideally lead to the same least-squares solution. As long as a researcher is only interested in the result of association learning, not in the iterative process, it is sufficient to calculate the least-squares solution directly from Eq. (16).
- The R-W salience factors α_i have no effect on the learning result – because linear regression is not sensitive to such a scaling of the input variables – but only on the learning process: associations for cues with high salience α_i are learnt faster than for other cues. The parameter λ leads to a (trivial) linear scaling of the learning result, but has not effect on the learning process. Only different learning rates $\beta_1 \neq \beta_2$ affect the learning result, because they modify $\mathbf{X}^T\mathbf{X}$ in a complex way.
- If R-W association learning or SLP training does not approximate the least-squares solution, it can arguably be considered to have failed. The only research question of interest that requires R-W iteration or application of the delta rule is thus: Under which circumstances and for which parameter settings does the R-W iteration converge or at least approximate the linear regresson? This is particularly relevant for single-event updates (as specified for the R-W model), which are much less robust and lead to larger fluctuations then batch updates. We plan to work on these issues with the help of simulation experiments.
- Having established NDL as linear regression with its well-known drawbacks (e.g. a propensity for overfitting the training data, especially if there is a large number n of cues), it will be interesting to contrast it with more state-of-the-art machine learning techniques. We plan to carry out a mathematical analysis and empirical study of (i) logistic regression, which is more appropriate for dichotomous data than linear least-squares regression, and (ii) regularization techniques, which control overfitting and encourage sparse solutions.

References

Baayen, R. Harald (2011). Corpus linguistics and naive discriminative learning. *Brazilian Journal of Applied Linguistics*, **11**, 295–328.

Danks, David (2003). Equilibria of the Rescorla-Wagner model. *Journal of Mathematical Psychology*, **47**, 109–121.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. See also `http://www.r-project.org/`.

Rescorla, Robert A. and Wagner, Allen R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black and W. F. Prokasy (eds.), *Classical Conditioning II: Current Research and Theory*, chapter 3, pages 64–99. Appleton-Century-Crofts, New York.

Widrow, Bernard and Hoff, Marcian E. (1960). Adaptive switching circuits. In *IRE WESCON Convention Record*, pages 96–104, New York. IRE.