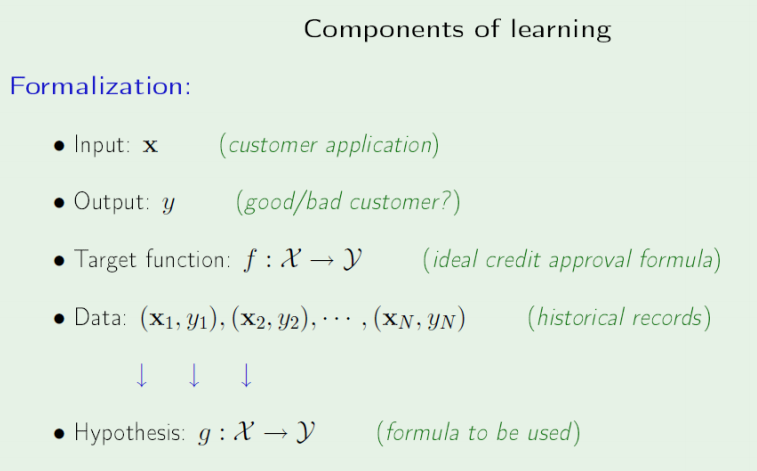
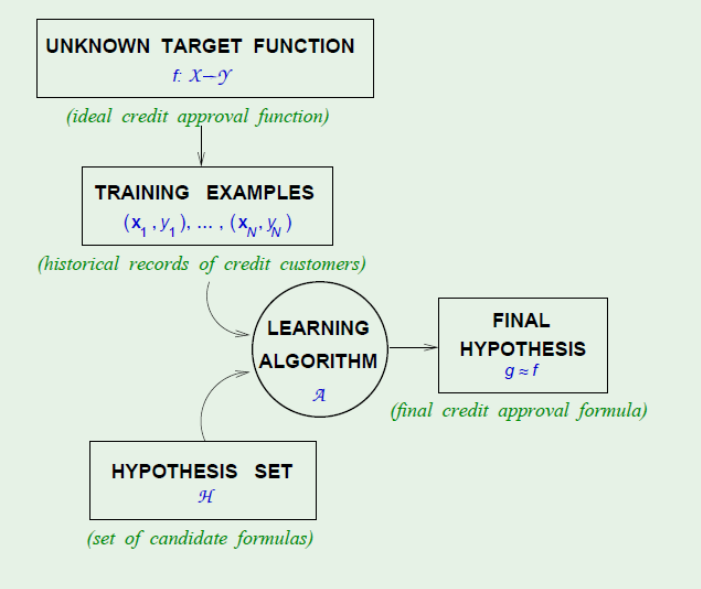
**למידה מנתונים: רגרסיה לינארית - Learning from Data: Linear Regression**

**רגרסיה לינארית** היא שיטה [סטטיסטית](http://he.wikipedia.org/wiki/%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99%D7%A7%D7%94), המשמשת לבדיקה וניצול של קשרים לינאריים בין שני משתנים או יותר. שיטה זו, על הגרסאות השונות שלה, נמצאת בשימוש נפוץ ביותר במחקרים כמותיים, בכל תחומי המדע.



רכיבים של למידה  
מתן צורה:   
קלט: x (יישום לקוח)  
פלט: y (לקוח טוב/רע?)  
פונקציית המטרה: f:x->y (נוסחת אישור אשראי אידיאלית)  
נתונים: ... (רשומות היסטוריות)  
השערה: g: x->y (נוסחה לשימוש

דיאגרמת הלמידה – גרסה 1



פונקציית יעד לא ידועה  
f: x->y

סדרה של השערות

סט השערה  
H

אלגוריתם למידה

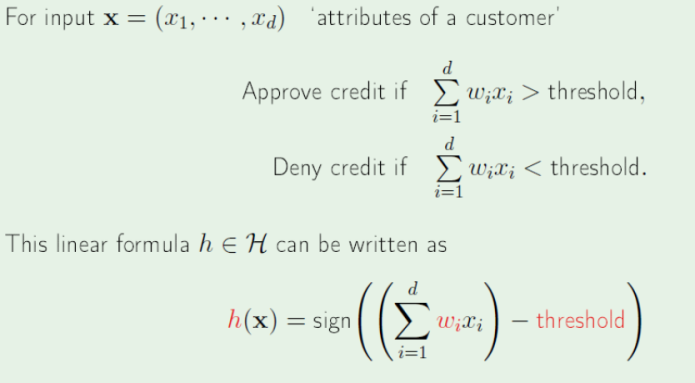
נוסחה סופית לאישור אשראי

השערה סופית  
g = f

רשומות היסטוריות של לקוחות אשראי

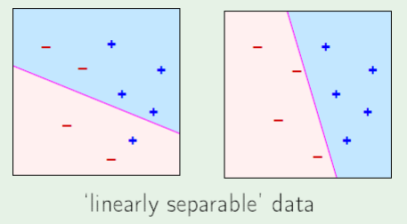
דוגמאות לאימון   
(x1,y1),…,(xn,yn)

פונקציית אישור אשראי אידיאלית

**Perceptron** - הפרספטרון הוא אלגוריתם סיווג, היינו, מטרתו היא להבדיל בין סוגים שונים של [דגימות](http://he.wikipedia.org/wiki/%D7%93%D7%92%D7%99%D7%9E%D7%94_(%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99%D7%A7%D7%94)) אותן הוא מקבל.

עבור קלט X = (X1,....,Xd) 'תכונות של לקוח'  
  
אשר את האשראי אם הנוסחה > גבול  
דחה את האשראי אם הנוסחה < גבול

יכולה להירשם גם כנוסחה הבאה:



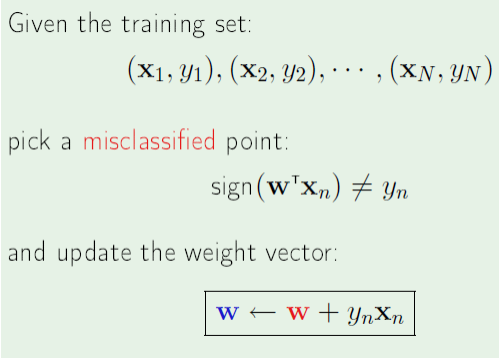
דוגמאות לנתונים מופרדים באופן ליניארי

Perceptron learning algorithm (PLA) – אלגוריתם הלמידה פרספטרון

בהינתן סט אימון: (x1,y1), (x2,y2),...,(xn,yn)

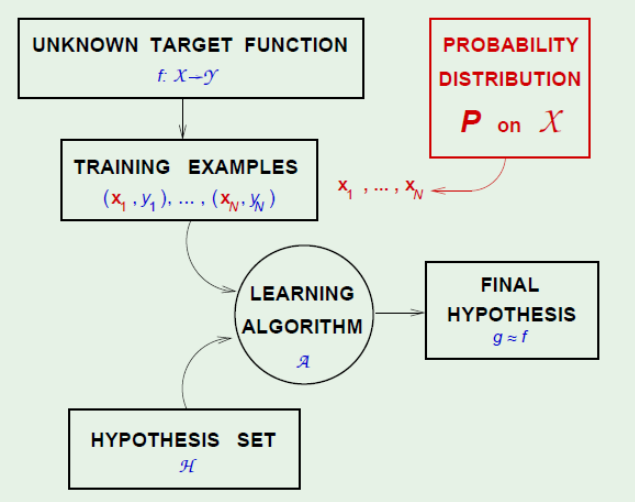
שינוי של המשקל לפני הנתון.

בעזרת האלגוריתם, הפרספטרון תמיד לומד על עצמו מחדש ומעדכן את המשקלות בהתאם.



למידה לעומת יישום התיאוריה

* עם הפרספטרון קיים פתרון מדויק למשימות שהן לא מופרדות ליניארית
* בעולם האמיתי משימות מסוג זה אינן מעניינות – יש לנו מודלים תיאורטיים/מתמטיים עבור מקרים כאלו.
* הלמידה בשימוש כאשר: כאשר יש לנו דפוס קיים, אי אפשר ליישם זאת באופן מתמטי, יש לנו מספיק נתונים על כך.



דיאגרמת הלמידה – גרסה 2  
  
נוסף למודל הזה גם התפלגות ההסתברות

מה ההסתברות ש – x יקרה (באדום)  
  
בנוסף לכל המודל שקיים נוספה עוד נוסחה כדי שתעזור לנו למצוא את ההשערה הסופית.

סימונים ללמידה:

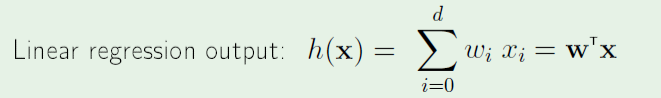
* עבור h(x) פונקציית ההשערה שלנו אנו מאמנים את w (משקל) כך שהשגיאה ב- y על הנתונים של סט התצפיות תהיה מינימלית.
* מפני שהנתונים שלי מסט הנתונים הוא "מדגם" מכל הנתונים האפשריים שקיימים, שגיאה זו נקראת  
   “in sample” error Ein(h). שגיאה אשר נוצרה מנתונים שיש לי ולא מכל הנתונים שיכולים להיות בעולם.
* כאשר אנחנו משתמשים בהשערה שלנו על בנקודות נתונים אחרות אז השגירה אשר נוצרה על נתונים אחרים נקראת “out of sample” error Eout(h). שגיאות מחוץ למדגם.
* המטרה היא להשיג Eout קטן, קרוב לאפס.

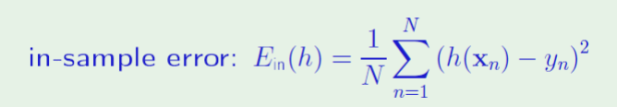
The ‘pocket’ algorithm

אלגוריתם הלמידה פרספטרון לא מסיים אם סט הלמידה לא מופרד באופן ליניארי. בהרבה מקרים בעולם האמיתי, לעומת זאת, אנו רוצים למצוא את ההפרדה הלינארית הטובה ביותר אפילו שסט הלמידה לא אידיאלי. אלגוריתם "הכיס" הוא שינוי חוק הפרספטרון. האלגוריתם מאחסן את וקטור המשקלות הטוב ביותר עד כה ב"כיס" בזמן שהוא ממשיך עדיין ללמוד. המשקלים בפועל משתנים רק אם נמצא וקטור משקלות טוב יותר.

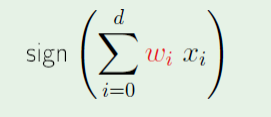
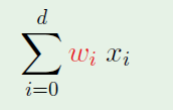
Linear regression (LR) vs. classification – רגרסיה לינארית לעומת סיווג

Classification – אישור אשראי (כן/לא)  
regression – מסגרת אשראי (סכום הדולר)

הפלט של רגרסיה לינארית.

שגיאה ברגרסיה לינארית: משתמשים בשגיאה בריבוע:  
  
שגיאת in sample, בתוך המדגם

הרגרסיה הלינארית היא מוגבלת. לדוגמא: קו האשראי מושפע מהתכונה "מס' שנים בבית המגורים", אבל לא בצורה לינארית. האם אנו יכולים לעשות את זה עם מודלים ליניאריים?

כלים של רגרסיה לינארית: כלים של classification:

האלגוריתם עובד בגלל **הלינאריות במשקלות**.

Transform the data nonlinear – שינוי הנתונים הלא לינאריים

