# Semi Supervised Learning

*תקציר: במהלך ההכשרה, אסכם מספר נושאים "רוחביים" שאינם מתקשרים לפרק ספציפי. הנושא השני הוא semi supervised learning. במסמך זה אסביר מה זה semi supervised learning, מה ההנחות שמניחים על המידע ואילו גישות ואלגוריתמים קיימים.*

## רקע

שתי הגישות העיקריות בלמידת מכונה הן supervised learning ו unsupervised learning, כאשר ההבדל העיקרי ביניהם הוא שבראשונה יש תיוג למידע. הבעיה העיקרית בהשגת תיוג היא שלפעמים התהליך הינו יקר. מצד שני, בלי תיוג קשה יותר למצוא תוצאות משמעותיות יותר בעזרת המידע. גישה שמנסה לפתור את הבעיות הנ"ל היא semi-supervised learning. בגישה זו, המידע שלנו מתויג בחלקו (לרוב חלק קטן מאד יחסית מהדאטה).

בגישה זו מניחים כמה הנחות על המידע:

* Continuity – הסיכוי שנקודות שקרובות זו לזו יהיו בעלי אותו התיוג גבוה יותר מהסיכוי שנקודות שרחוקות זו מזו יהיו בעלי אותו תיוג.
* Cluster – המידע יכול להתחלק לקלאסטרים מופרדים כאשר התיוג בכל קלאסטר יהיה דומה והתיוג בין קלאסטרים יהיה שונה.
* Manifold – המידע כנראה נפרד על מרחב בעל מימד יחסית נמוך

## גישות

### Self-Training

עבור סט של מידע מתויג וסט של מידע לא מתויג, האימון מתבצע בצורה הבאה: בכל איטרציה מאמנים את המודל על המידע המתויג ומתייגים את המידע הלא מתויג. עבור תיוגים עם ביטחון גבוה (מעל סף מסוים), נוסיף אותם (יחד עם התיוג) למידע המתויג עליו מתאמנים באיטרציה הבאה. ממשיכים בתהליך מספר קבוע של פעמים או עד שאין תיוגים חדשים עם רמת ביטחון גבוהה מספיק. הבעיה העיקרית עם הגישה היא שבמידה ובשלב מסוים תייגנו עם ביטחון גבוה בצורה לא נכונה, אין לאלגוריתם דרך לתקן את עצמו באיטרציה מתקדמת.

### Multi-view Training

מאמנים מודלים שונים שמסתכלים על המידע בצורה אחרת (חלק שונה של הדאטה, מודל שונה, פיצ'רים שונים וכו')

Co-training – מחזיקים שני מודלים שונים כאשר כל אחד מהם מאומן על חלק מהמידע. בכל איטרציה, אם אחד מהמודלים (לא שניהם) מתייגים רשומה בביטחון גבוהה מערך סף אז מוסיפים את הרשומה והתיוג לסט האימון של המודל השני (שלא היה בטוח) לאיטרציה הבאה. ממשיכים כך עד שאין תוספות חדשות לסט האימונים של המודלים.

Democratic co-learning – מתחזקים קבוצת מודלים שונים ובכל איטרציה מאמנים אותם על כל הדאטה המתויג ומתייגים את המידע הלא מתויג. לכל רשומה ותיוג בודקים האם יש רוב של מודלים שתייגו אותה בתיוג מוסיים (רוב-> יותר מודלים תייגו את הרשומה בתיוג ספציפי ביחס לכמות המודלים שלא תייגו אותה בתיוג הזה). אם כן, התיוג ייקבע לפי ממוצע משוקלל ביחס לרמות הביטחון של המודלים.

Tri-training – נתחזק שלושה מודלים (רצוי שיהיו שונים – מודלים שונים, שימוש ב bootstrapping...). האיטרציות עצמן דומות לco training רק שנוסיף למודל 0 את הרשומה והתיוג אם מודל 1 ומודל 2 הסכימו על הרשומה והתיוג.

Tri-training with disagreements – כמו הרגיל אבל מוסיפים דרישה שמודל 0 לא יסכים עם מודל 1 ומודל 2 (בנוסף לכך שמודל 1 ו 2 יסכימו ביניהם).

Asymmetric tri-training – כמו הרגיל אבל המודל שמעדכנים לומד רק על הלייבלים שהמודלים האחרים הוסיפו לא (במקום עליהם וגם על המתויגים המקוריים). נועד להתמודד עם shiftים במידע.

Multi-task tri-training –בהתחלה מאמנים בלי bootstrapping (על כל המידע). בכל איטרציה מוסיפים לסט האימונים דוגמה אם שני המודלים האחרים מסכימים עליה. אחד מהמודלים מאמנים רק על המידע החדש ואת האחרים מאמנים על החדש+הישן.

### Self-Ensemble

לרוב משתמשים בסוג אחד של מודל עם היפר-פרמטרים.

Ladder networks – בכל איטרציה, כאשר מתייגים בביטחון גבוה רשומה כלשהי, לא נוסיף אותה כמות שהיא למידע לאימונים באיטרציה הבאה אלא נייצר דוגמאות שעברו שינוי כלשהו (פרטובציה) על ידי אגמנטציה (במקרה של תמונות: לסובב, לחתוך, להגדיל וכו') עם התיוג של הרשומה המקורית ונוסיף אותם לדאטה לאיטרציה הבאה. נועד לפתח מודל שעמיד בפני רעש.

Virtual Adversarial Learning – כמו ladders אבל הופכים את הרשומה לאדברסיריאלית (באה להרוס את המודל). דוגמה אדברסיריאלית אמיתית תדרוש גם את התיוג האמיתי של הרשומה אבל בגלל שאנחנו עובדים על virtual adversarial, אפשר להסתדר בלי (פירוט על איך - במאמר). המטרה היא להפוך את המודל לעמיד תוך מתן דגש לפיצ'רים שאיתם המודל מתמודד הכי פחות טוב (על ידי יצירת דוגמאות ש"מנצלות" את החולשה של המודל לפיצ'רים האלה ובכך לשפרו באיטרציה הבאה).

Pi model – בכל איטרציה, חוזים עבור כל רשומה את התיוג לאחר שמפעילים על הרשומה שתי פרטובציות שונות. כעת נקבע את ה loss את ריבוע ההפרש בין התיוגים (ונכפיל בקבוע שיקבע כמה החישוב הזה ישפיע בסוף). לאחר מכן, אם יש לנו את התיוג האמיתי של הרשומה, נוסיף ל loss במידה והתיוג (תחת אחת מהאוגמנטציות. אחת קבועה. תמיד אותה אחת.) אינו נכון ונעדכן את המודל בהתאם ל loss שקיבלנו בעזרת אופטימייזר כלשהו (ADAM וכו'). בנוסף, בכל איטרציה משקל ה loss (כלומר, משקל האיטרציה) נקבע על ידי משקל שעולה בכל איטרציה. איור של העסק מופיע בחלק העליון של איור 1.

הרעיון הוא שבכל איטרציה נקבל תיוג עמיד (שנבחן על ידי שתי פרטובציות שונות) ובנוסף ניתן דגש רב יותר לאיטרציות מאוחרות.

Temporal Ensembling – כמו pi model אבל הפעם במקום להשוות בין הביצועים על שתי פרטובציות שונות אנחנו משווים את התיוג של פרטובציה אחת לתיוג על אותה הפרטובציה באיטרציה הקודמת. יתרון יחסי אחד לעומת pi הוא שהתהליך מהיר יותר (דורש תיוג יחיד לעומת תיוג כפול). יתרון נוסף הוא שתיוג המטרה (שנקבע על ידי איטרציות קודמות) יהיה פחות רועש מהתיוג שמשתמשים בו ב pi (פרטובציה אחרת). איור של העסק מופיע בחלק התחתון של איור 1.

Mean Teacher – במקום לבצע ממוצע משוקלל של התיוגים לאורך זמן, עושים ממוצע משוקלל של המשקלים של המודלים לאורך זמן. השיטה רלוונטית רק לרשתות (בגלל כל העניין שעובדים עם משקלים).

SWA – כמו קודם אבל את הממוצע המשוקלל מבצעים בצורה שונה. עד עכשיו עבדנו בצורה שאיטרציות מאוחרות מקבלות יותר משקל מאיטרציות מוקדמות. ב SWA המשקל של האיטרציות בא בצורה מעגלית (כלומר, משקל האיטרציה גדל עד שלב מסויים ואז הוא חוזר להיות קטן, שוב גדל וכו'). האופטימיזציה מתבצעת בעזרת stochastic gradient descent. בגלל ש SGD לא תמיד מגיע לנקודת מקסימום יחידה, ממשקלים בצורה "מעגלית"- על מנת לבדוק כמה נקודות מקסימום לוקאליות.

|  |
| --- |
|  |

איור 1 : מבנה של pi model ושל temporal ensembling