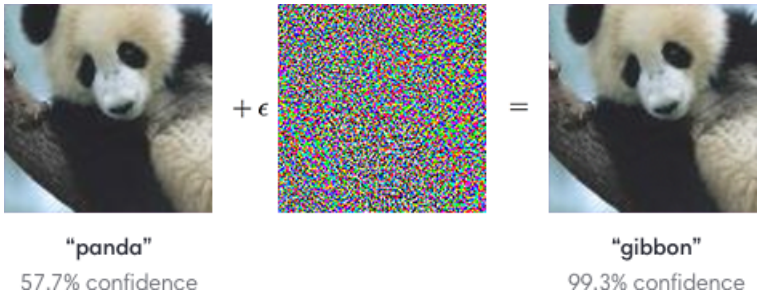
שיעור 8

5 ביוני 2019

1 Adversarial Examples

בשיעורים הקודמים עסקנו במגוון ארכיטקטורות של רשת נוירונים - החל מרשתות נוירונים פשוטות, fully connected, דרך רשתות קונבולוציה וכלה ברשתות RNN. למדנו כיצד רשתות נוירונים לומדות לבצע התאמה לבעיות, דרך back propagation ואלגוריתמי אופטימיזציה כדוגמת sgd. שיטות האופטימיזציה הללו הן פשוטות מאוד, אך מובילות לתוצאות מפתיעות לטובה. עם זאת, הפשטות הזו מסתירה כמה תופעות מעניינות ומפתיעות - אחת הגדולות בהן היא התופעה של adversarial attack: בהינתן רשת מאומנת, נוכל למצוא דוגמאות "דומות" מאוד לדוגמאות מסט האימון, אך שיסווגו באופן שונה לחלוטין. נכיר דרכים שונות ליצור דוגמאות כאלה. נפרמל את המשימה שלנו כך: בהינתן קלט של הרשת המסווג סיווג נכון, נרצה להוסיף רעש קטן לקלט, כך שהסיווג של הרשת יהיה שגוי. לתופעה זו השלכות מרחיקות לכת על השימוש של רשתות נוירונים ביום־יום, בתעשייה ובצבא: ככל שרשתות נוירונים תופסות מקום מרכזי יותר באלגוריתמיקה שסביבנו, כך הפגיעות שלנן נעשית מטרידה יותר. חשבו למשל מה יכול להתרחש בעולם בו ניתן לשטות בקלות באלגוריתמים של רכבים אוטונומיים או במזהי פנים בבידוק בטחוני בשדה התעופה.



איור 1:

1.1 תקיפות white box

תקיפת white box פירושו שאנו תוקפים את המודל בהינתן המודל. כלומר, אנו חשופים לרשת הנוירונים ולמשקולות שלה. נוכל להשתמש בתכונת הגזירות של הרשת כדי לייצר תקיפה. באימון באמצעות back propagation ראינו כי הרשת גזירה ביחס למשקולות שלה. אולם, הרשת גזירה גם ביחס לקלט שלה. לכן נוכל לחשב נגזרת של loss אותו נגדיר על

הרשת ביחס לקלט של הרשת. נפרמל זאת: נניח כי נתונה לנו הרשת f והדוגמא (x, y) אותה נרצה לסווג כ־$\tilde{y}$. הרשת מאומנת באמצעות ה־loss המסומן ב-L. נתבונן בביטוי

$$\tilde{L} := L(f(x), \tilde{y})$$

אנו רוצים למזער את הביטוי. בהינתן ש־L, f גזירות, נוכל לגזור את הביטוי ביחס ל־x ולקבל את $\frac{d\tilde{L}}{dx}$. כעת נוכל להתאים את x באמצעות תהליך של gradient descent:

$$x \mapsto x - \delta \cdot \frac{d\tilde{L}}{dx}$$

בתהליך זה אנו משנים בהדרגה את הקלט של הרשת עד להגעה לדוגמא תוקפת. אולם נשים לב כי אין מניעה בתהליך לשנות לחלוטין את x: אנו עשויים להגיע ל־x שאינו דומה כלל לקלט המקורי. היינו רוצים לאכוף אילוץ קשיח על הקלט של הרשת: בהינתן מטריקה על מרחב הקלטים ותחת סימון הקלט המקורי ב־x_0, נרצה לדאוג כי $d(x, x_0) < \varepsilon$. למשל, עבור בעיה של תמונות, נוכל להגדיר את המרחק בין תמונות כ־

$$d(x, x_0) = ||x - x_0||_{L^2}$$

כלומר, נחשוב על התמונות כעל וקטורים ונחשב את המרחק ביניהם.
כיצד נאכוף את האילוץ? דרך אחת לעשות זאת היא להבטיח באופן "קשיח" כי האילוץ מתקיים. כלומר, לא לאפשר לx "להתרחק" מ־x_0, ע"י קטיעה של תהליך ה־gradient descent או ע"י הקטנה של δ. עם זאת זהו אינו פתרון טוב - הקטיעה של תהליך האימון עשויה שלא לאפשר התכנסות. פתרון נפוץ יותר הוא שינוי של ה־loss באמצעות תוספת של אילוץ רך:

$$\tilde{L} := L(f(x), \tilde{y}) + C \cdot d(x, x_0)$$

למשל, עבור סיווג של תמונות,

$$\tilde{L} = binary\, cross\, entropy(f(x), \tilde{y}) + C \cdot ||x - x_0||^2$$

כאשר, כזכור,

$$binary\, cross\, entropy(f(x), \tilde{y}) = -\tilde{y} \cdot log(f(x)) - (1 - \tilde{y}) \cdot log(1 - f(x))$$

באופן זה אנו מבטיחים כי הרשת לא תלמד פתרונות "שונים מדי" מ־x_0. הביטוי $C \cdot ||x - x_0||^2$ משמש כרגולציה של האימון, ונוכל לשלוט ב־C כדי לשלוט בעוצמת הרגולציה.
בתהליך כולו נקבל קלט חדש, $\tilde{x}$, שיסווג כ־$\tilde{y}$, כאשר $d(\tilde{x}, x_0)$ הוא קטן. לכן קיבלנו "רעש" קטן, $(\tilde{x} - x_0)$, שכאשר אנו מוסיפים אותו לקלט אנו מקבלים דוגמא תוקפת. מה מבטיח לנו שנצליח לתקוף את הרשת? ובכן, הדבר לא מובטח. אולם מניסויים עולה כי ככל שהרשת מורכבת יותר, ולמעשה אין צורך ברשת מורכבת במיוחד, משימת התקיפה היא קלה יותר. למעשה, אין קונצנזוס בקהילה המחקרית על הסיבות לתופעת התקיפות של רשתות נוירונים. זאת עקב אבחנות נוספות על תקיפת רשתות נוירונים שנדון בהן בהמשך.

1.2 חלוקה לסוגי תקיפות

באופן כללי נהוג לחלק את התקיפות על מודלים לכמה סוגים, לרוב לפי כמה פרמטרים עיקריים:

1. מידת ההיכרות עם המודל אותו תוקפים (הארכיטקטורה שלו, המשקולות שלו, סט האימון שלו).

2. מידת החשיפה לפלט של המודל. למשל, עבור מודל המקבל תמונה וחוזה את האובייקט בתמונה, ישנו הבדל בין עבודה עם הפלט של המודל - וקטור התפלגות, softmax על כל הקטגוריות, לבין עבודה עם ה"תשובה" של המודל - האובייקט המופיע בתמונה.

3. אופן השינוי האפשרי בקלט. הדבר לרוב יתבטא באופן בו אנו מודדים את השינוי בקלט - למשל, נוכל לדרוש שינוי של ε במרחק L^2 בין המודלים, אך נוכל לדרוש גם שינוי של לכל היותר פיקסל יחיד בתמונה (one pixel attack, המתאימה לחלוטין לפורמליזם שפיתחנו קודם, כאשר השינוי נמדד במטריקת L^0, כמות תהאיברים בוקטור אותם אנו משנים).

4. מידת השליטה שלנו על הקלט. ניתן להבדיל בין התקפות בהן אנו שולטים במדויק בכל ערך בקלט הנכנס למודל, כדוגמת התקפת ה־white box שתיארנו קודם, ובין התקפות real world, בהן הקלט שאנו מכניסים עובר כמה שלבים עד לכניסה שלו לרשת - למשל, כאשר הקלט של הרשת בהכרח עובר דרך מצלמה שמצלמת תמונות ושולחת אותן לרשת, כפי שקורה במודלים של רכבים אוטונומיים.

כעת משתיארנו תקיפה בסיסית וחילקנו את התקיפות האפשריות לפי אוסף תכונות שלהן, נתעניין בתקיפות במסגרת מאתגרת יותר.

1.3 תקיפות black box

תקיפות black box הן שם כולל לתקיפה בה אין לנו ידע על המודל. נעבוד תחת ההנחה כי אין לנו כלל מידע על הארכיטקטורה של המודל, וודאי שלא על המשקולות שלו, אך כאשר יש לנו את סט האימון עליו המודל התאמן. התרחיש אינו מופרך - למשל, אנו יודעים כי מודלים רבים של עיבוד תמונה משתמשים באוסף מאגרי מידע מוכרים. למעשה, נעיר כי במציאות הבעיה חמורה מכך - מודלים רבים של עיבוד תמונה עושים שימוש נרחב ב־transfer learning, כך שמתקבלת רשת שאת רובה אנו מכירים גם ברמת המשקולות.
כעת, ללא מידע על המודל, לא נוכל לגזור אותו לפי הקלט, כך שהשיטה שהתגנו קודם לכן לכאורה איננה שימושית. אולם מסתבר כי ניתן לתקוף את הבעיה מכיוון אחר. אם המודל אותו אנו תוקפים הוא f, נוכל לבנות מודל חדש, g. נאמן אותו "ללמוד" את f. כלומר, בהינתן קלט x נרצה שיחזה את $f(x)$. נעיר כי אנו לא יודעים מהי הארכיטקטורה של f, כך שאת הארכיטקטורה של g נאלץ לקבוע בעצמנו. נקבל מודל מאומן, g. כעת נתקוף אותו ונייצר דוגמא תוקפת, $\tilde{x}$. מסתבר כי הדוגמא שייצרנו היא דוגמא תוקפת גם עבור המודל המקורי f. בשלב זה אני מציע לעצור כדי להעריך את העובדה הכלל־לא־טריוויאלית הזו - דוגמא תוקפת עבור המודל g, שאומן בהשראת המודל f, היא דוגמא תוקפת גם עבור המודל f. למעשה, התופעה הזו היא מקרה פרטי של תופעה רחבה יותר שניגע בה בהמשך, תופעה שמציבה את האתגר הגדול ביותר בפני הבנה של מנגנון התקיפה כולו.
נצלול אל הפרטים של התקיפה שתיארנו. ישנן שתי שאלות עיקריות העולות מתיאור הפתרון שהצענו: כמה אנו רגישים לאריכטקטורה של המודל g? לכמה דוגמאות $(x, f(x))$ נזדקק באימון המודל g? באופן מפתיע, אין חשיבות רבה לארכיטקטורה של המודל g. אנו יודעים מראש מה צריכים להיות הגדלים של הקלט ושל הפלט של המודל f מהיכרות שלנו עם אופן הפעולה של רשתות נוירונים, כך שנוכל לקבוע אותם גם עבור g. נוכל להשתמש גם בידע שלנו על ה־domain של הקלט: למשל, עבור חיזוי בתמונות, נוכל להשתמש ברשתות קונבולוציה. אולם מעבר לכך, שינויים בארכיטקטורה לא משפיעים מהותית על פעולת הפתרון שלנו.

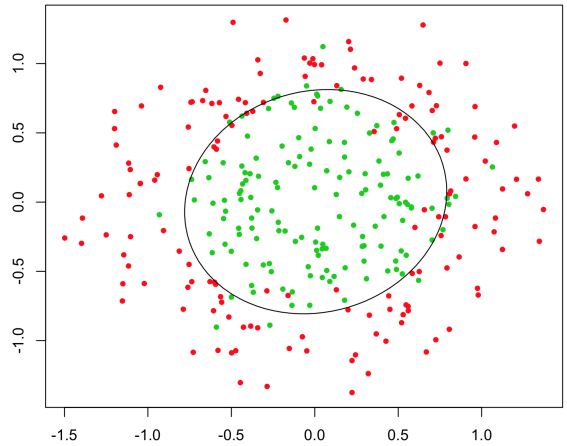
באשר לכמות הדוגמאות שאנו זקוקים לה באימון - לא נוכל לתת לכך הערכה מפורשת, אך נוכל לתאר שיטות לתייג באמצעות f כמה שפחות דוגמאות x באמצעות שיטות מעולם ה־active learning, בו אנו מתייגים קלטים לרשת על סמך הביצועים שלה על קלטים אחרים. באופן היוריסטי, נוכל לבחון את האזורים בקלט בהם הרשת "לא בטוחה", ולדגום מהם יותר.
שיטה זו פשוטה, אך אפקטיבית. למודל g נהוג לקרוא surrogate model עבור המודל f.

1.4 עקרון הטרנספרביליות

תיארנו קודם לכן תופעה מדהימה למדי: עבור מודל g שאומן לקרב את מודל f, דוגמא תוקפת עבור מודל g תתקוף גם את מודל f, ללא תלות רבה בארכיטקטורה של g וללא כל ידע על הארכיטקטורה של f או על המשקולות שלו. למעשה התופעה היא רחבה יותר: עבור שני מודלים f_1, f_2, דוגמא תוקפת עבור המודל f_1 היא גם דוגמא תוקפת עבור מודל f_2. כלומר, "דוגמא תוקפת" היא תכונה מובנית של תהליך הלמידה שלנו (!). תופעה זו מוכרת כ"עקרון הטרנספרביליות". תופעה זו מסקרנת מאוד את החוקרים בתחום, ומקשה על ההסבר של תופעת התקיפה של רשתות נוירונים. נציג בקצרה כמה הסברים לתופעת התקיפה:

1. רשתות מבצעות over fit. הן מתאימות לקלט מסוים, ושינויים קטנים בקלט גורמים לרשתות לתחזית שגויה.

2. התופעה קשורה למימד הגבוה של הקלט ברשת. למשל, עבור רשת המסווגת קלט לאחת משתי קטגוריות, היא למעשה בונה "משטח החלטה", שמצדו האחד היא חוזה ערך אחד, ומצדו השני את הערך השני. במימד גבוה למשטח זה יש גיאומטריה מורכבת, ושינויים קלים בדוגמא "חוצים את המשטח".



איור 2:

3. המבנה של רשת נוירונים הוא ליניארי למקוטעין (עבור שימוש ב־$ReLu$) או קרוב להיות ליניארי למקוטעין. עבור מבנה שכזה ישנן הוכחות קונסטרוקטיביות לבנייה של דוגמאות תוקפות.

4. רשת רואה "מיקרו־פיצ'רים" ומשתמשת בהם לחיזוי, אך הם אינם רובוסטיים להפרעות מסוגים שונים. למשל, עבור תמונות, מדובר בפיצ'רים בהם העין האנושית אינה מבחינה.

כאמור, אין הסכמה באקדמיה על הסבר לשלל התופעות שמערבות דוגמאות תוקפות.

2 מודלים גנרטיביים

נציג כעת תחום אחר ברשתות נוירונים, אך נתמקד בגישה המתקשרת לתופעת הדוגמאות התוקפות. נעסוק במודלים גנרטיביים. מטרתם היא, באופן כללי, "לייצר מידע". הדבר יכול לאפשר אימון טוב יותר של מודלים המצריכים מידע רב. בשפה פורמלית, נחשוב על קלט כעל דוגמאות x הנדגמות מהתפלגות $p(x)$. למשל, עבור אוסף התמונות של פרצופים, $p(x)$ היא התפלגות על מרחב התמונות האפשריות (וקטורים בגודל $n \times n$), כך שאם נדגום מתוכה נקבל תמונה של פרצוף. המטרה שלנו במודלים גנרטיביים היא "לקרב את $p(x)$" , כלומר לאשר לנו לדגום מתוך $p(x)$, בקירוב טוב ובקלות.

איור 3:

2.1 GAN - Generative Adversarial Network

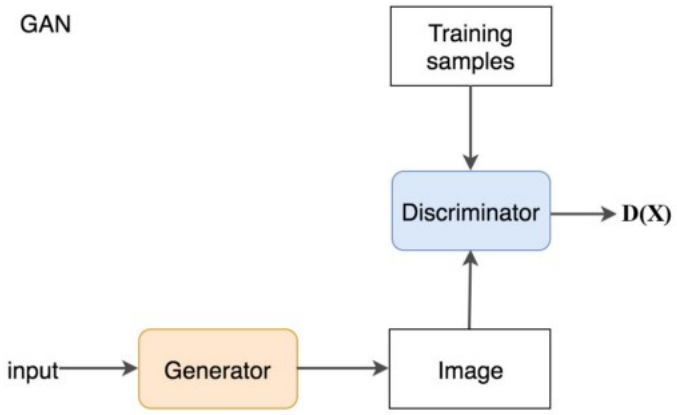
מודל ה־GAN היה מודל פורץ דרך בתחום הייצור של מידע, והפך את התחום למתוחכם בהרבה. נציג את המודל הנאיבי. לאחר מכן נדון בבעיות שעולות בו ובשיפוריים אפשריים שלו. המודל מורכב משתי רשתות נוירונים - $d\,(Discriminator)$ ו־$g\,(Generator)$. הן מאומנות האחת כנגד השנייה. נתאר את פעולתם דרך דוגמא: נניח כי אנו מעוניינים לייצר תמונות של פנים. המודל g יקבל כקלט רעש אקראי, למשל רעש מההתפלגות $\mathcal{N}(0,1)^k$. הוא יחזיר כפלט תמונה בגודל $n \times n$. המודל d יקבל תמונה בגודל $n \times n$ ויחזיר 1 אם היא "תמונה אמיתית" ו־0 אם היא תמונה שנוצרה באמצעות g. נאמן אותם לסירוגין:

1. נקפיא את המודל g. נייצר סט אימון המורכב מדוגמאות אמיתיות ומדוגמאות שנוצרו באמצעות g. נאמן את המודל d להחזיר 1 על התמונות האמיתיות ו־0 על התמונות ש־g ייצר.

2. נקפיא את המודל d. נשתמש ב־g כדי "לתקוף" את המודל d. בפרט, נאמן את g באמצעות ה־loss הבא: $binary\,cross\,entropy(d(g(x)), 1)$. כלומר, נאמן את g כך שאם נפעיל על התוצרים שלו את d, נקבל כי d מחזיר 1.

3. נחזור על השלבים הקודמים.

איור 4:



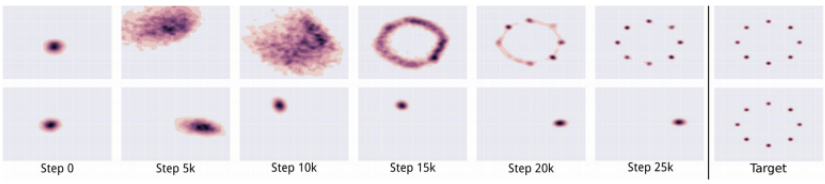
נקבל כי לשני המודלים יש "מטרות סותרות": האחד, g, מנסה לייצר תמונות מציאותיות ככל האפשר, במובן זה שיצליחו "לתקוף" את המודל d, ואילו השני, d, מנסה לזהות את התמונות של g. אנו מאמנים אותם לסירוגין. בפרט, לא נאמן אותם עד שנגיע למיצוי של הלמידה: היוריסטיקה סבירה יכולה להיות אימון של epoch אחד עבור כל מודל. זו טכניקת אימון מסובכת, הדורשת מאיתנו "יד על הדופק" בכל שלב באימון. תחת אימון מוצלח נקבל ירידה הדדית של ה־loss־ים של שני המודלים, כאשר שניהם נעשים מתוחכמים יותר. נעיר כי התוצאה הסופית אליה אנו חותרים - מודל גנרטיבי חזק - לאו דווקא קורלטיבית עם ה־loss שנקבל באימון בתהליך שתיארנו, ונדרש לבחון את ההישגים שלה באופן אחר.
באימון של GAN עשויות לצוץ כמה בעיות:

1. חוסר שיווי משקל בין המודלים. יתכן מצב בו בכל שלב באימון, המודל אותו אנו מאמנים ברגע מסוים "מנצח" את השני, ומגיע ל־loss נמוך מדי. הדבר מוביל לחוסר יציבות של האימון ולחוסר התכנסות לפתרון.

2. בעיית mode collapse: אנו עשויים לקבל מצב בו הגנרטור מתכנס בכל פעם לפתרון יחיד, במובן זה שאינו מייצר תמונות שונות כתוצאה מקבלת רעש אקראי, אלא את אותה התמונה ה"תוקפת" בשינויים קלים. ניתן לראות המחשה לכך בתמונה המצורפת. בשורה העליונה ניתן לראות מודל g שהצליח ללמוד לייצר דוגמאות מכל

ההתפלגות. בשורה השנייה ניתן לראות מודל ש"קופץ" לאזור אחר בהתפלגות בכל פעם שהמודל d מצליח לזהות את הדוגמאות שלו.

איור 5:



3. בעיה נוספת שעשויה להיווצר היא היות המודל d מוצלח כל כך עד כי המודל g לא מצליח להתגבר עליו ולקבל גרדיאנטים כדי להשתפר. באופן כללי האימון של המודל d בכל שלב באיטרציה הוא פשוט יותר מאימון המודל g.

בעיות אלו הן מסובכות מאוד, וגורמות לכך שאימון של GAN הוא משימה מורכבת ולא יציבה.

2.2 WGAN

נעשו בשנים האחרונות מספר ניסיונות לתאר ארכיטקטורות יציבות יותר. אחד הניסיונות המשמעותיים בתחום הוא ה־WGAN. מהותה היא באחבנה על טבע ה־loss הסטנדרטי של GAN - *binary cross entropy*.
הארכיטקטורה של WGAN דומה מאוד לזו של ה־GAN. היא עושה שימוש ב־loss אחר במהלך העדכון של משקולות הגנרטור, כאשר נוסיף רכיב קטן לארכיטקטורה (בדמות שינוי של הגרדיאנטים אחרי חישובם - קטימה שלהם):

	Discriminator	*Generator*
GAN	$\nabla_\theta \sum_i log(d(x_i)) + log(1 - d(g(z_i)))$	$\nabla_\theta \sum_i log(d(g(z_i)))$
$WGAN$	$\nabla_\theta \sum_i d(x_i) - d(g(z_i))$	$\nabla_\theta \sum d(g(z_i))$

השינוי ב־loss הוא מזערי, ויחד עם שינוי קטן בגרדיאנטים של d שנועד להבטיח תכונות מסוימות שנדרשות מהתיאוריה של פיתוח WGAN, הוא מביא לתוצאות טובות בהרבה. למעשה, ה־WGAN מגיע מתוך האבחנה לפיה ה־loss שהשתמשנו בו ב־GAN מאפשר למדוד מרחק בין התפלגויות, אך הוא לא מצטיין בהעברת גרדיאנטים לרשת: גם כאשר לנו נדמה כי פעולה מסוימת של הרשת צריכה להקטין את ה־loss, בפועל היא אינה עושה זאת. ה־loss המעודכן ב־WGAN הוא יותר "רך" ומתחשב גם בשינויים כאלה. הגברת הפידבק הזו פירושה התכנסות בטוחה יותר, שמתגברת על רוב התקלות שנגענו בהן קודם לכן.