מסמך עיצוב

1. תיאור הארכיטקטורה:

**תיאור חומרה:** המערכת מורכבת משני חלקים עיקריים – אפליקציה על טלפון, ושרת.

**רכיבים שונים והקשרים ביניהם:** השרת מכיל את רשת הנוירונים והוא זה שיבצע את ניתוח התמונה, בעוד שהאפליקציה רק תשלח את התמונה לשרת ותקבל את הטקסט שנמצע בתמונה.

**מבנה בסיס הנתונים:** "בסיס הנתונים" בתוכנה יהיה רשת הנוירונים, שתיקח (לפי הערכות) כ 300 מגה בייט. הרשת שמורה בתור כמה קבצי טקסט (בשימוש במודול cPickle) ואין לה מבנה מיוחד.

1. תיאור הטכנולוגיה הרלוונטית:

**תיאור טכנולוגית המערכת:** ה"מוח" של המערכת היא רשת נוירונים, שיטה של למידת מכונה, אשר לומדת לזהות אותיות. באופן מופשט, הרשת מורכבת ממספר שכבות, אשר בכל אחת מהן יש מספר מסוים של "נוירונים". השכבה הראשונה של הרשת נקראת "שכבת הקלט" והיא פשוט 784 (28\*28) נוירונים, כאשר כל פיקסל מהתמונה משויך לנוירון קלט אחד. הנוירון מקבל מספר מ 0 ל 1 בתור הקלט שלו. בין כל שכבה ושכבה ישנן "סינפסות" – חיבורים בין נוירונים. כל נוירון בשכבה 1 מחובר לכל נוירון בשכבה 2, וכן הלאה. לכל סינפסה יש "משקל". המשקלים של כל סינפסה מתחילים אקראיים.

במהלך החישוב, כל נוירון בשכבה 1 יוציא קלט לכל נוירון בשכבה 2 שהוא המכפלה בין הקלט שלו והמשקל של הסינפסה. כל נוירון בשכבה 2 יקבל בתור הקלט שלו את הסכום של הפלטים של כל הנוירונים בשכבה 1, ואז יבצע על קלט זה פונקציה מתמטית שתהפוך אותו למספר בין 0 ל 1. הרשת תמשיך ככה בין כל השכבות עד שתגיע לשכבת הפלט, ובה יופיעו אחוזי הביטחון של הרשת בכל אחת מהאופציות. פעולה זו נקראת "Feedforward".

בתהליך הלמידה של הרשת, היא תבצע Feedforward על תמונה לדוגמא (שידוע מה התוצאה האמיתית שלה), תמצא את השגיאה שלה, ובעזרת חישוב מתמטי מסובך תוכל גם למצוא על כל משקל כמה הוא השפיע על השגיאה. היא תשנה בהתאם את המשקלים, ותעבור לדוגמא הבאה. כך למשך עשרות (או מאות) אלפי דוגמאות.

שאר המערכת הוא מאוד פשוט יחסית – אפליקציה שמצלמת תמונה ושלוחת לשרת, שרת שמקבל תמונה, מוצא בא אותיות נפרדות ומנתח כל אחת, ושולח את הטקסט הכולל חזרה לאפליקציה.

**הסיבה לבחירה בטכנולוגיה זו:** רשתות נוירונים הן סוג למידת המכונה המוביל כרגע בניתוח תמונה, וכנראה גם הפשוט ביותר ללמוד.

1. תיאור מודולים בהם נעשה שימוש:

**סקירת המודולים המרכיבים את המערכת ותיאורם:** המודולים המרכזיים שבהם רשת הנוירונים משתמשת הינם Theano ו Numpy.

Theano: מודול שעוזר ברשתות נוירונים. הוא מצויד במספר רב של פונקציות מתמטיות שצריכים בשביל רשת הנוירונים, מחלקות בשביל סוגים שונים של שכבות ונוירונים, אפשרות לבצע חישובים אינטנסיביים על ה GPU במקום על ה CPU, ועוד. השימוש בו הוא מאוד לא אינטואיטיבי בהתחלה, ולכן יש לקרוא עליו הסבר מפורט בשביל להבין את הקוד.

Numpy: מודול מתמטי שמתמחה באלגברה לינארית – תחום חשוב מאוד ברשתות נוירונים. במקום להתייחס לכל נוירון ומשקל בתור int, ולאכסן אותם בתוך רשימות, Numpy מאפשר לנו ליצור אובייקטים של מטריצות, אשר מכילות את הנוירונים או את המשקלים, ולהפעיל עליהם חישובי מטריצות, מה שמייעל דרסטית את החישובים הדרושים בשביל ללמד רשת.

אחרי שהרשת למדה, היא נשמרת בעזרת מודול cPickle.

שנית, המערכת משתמשת במספר מודולים של ניתוחי תמונה, כמו OpenCV ו PIL. מודולים אלה יאפשרו לתוכנה לנקות רעש מתמונה, למצוא אותיות בה, לישר אותן, להפוך אותן לשחור-לבן, ולפרק אותן לאות-אות.

בנוסף, המערכת משתמשת במודול autocorrect, כדי למצוא שגיאות בטקסט שהיא הוציאה. אם היא מזהה שגיאה במילה, היא תנסה להחליף את האותיות בכאלה שהיא פחות בטוחה בהן, ובודקת האם הן מתאימות יותר. כך למשל, היא תוכל לדעת שהאות i היא לא 1.

לבסוף, המערכת משתמשת במודול socket בשביל התקשורת בין האפליקציה לשרת.

1. תיאור סביבת הפיתוח:

**שפת התכנות:** שפת התכנות היא Python, אך המודול Theano משתמש גם בשפת C בשביל יעילות. אני עוד לא בחרתי בסביבת עבודה בשביל פיתוח האפליקציה.

**כלי הפיתוח:** אין כלי פיתוח למעט המודולים.

**הסביבה והכלים לבדיקות:** סביבת העבודה למרבית המערכת היא Pycharm, אך אני אצטרך למצוא סביבת עבודה נוחה לכתיבת אפליקציה.

1. תיאור האלגוריתמים המרכזיים בפרויקט:

**ניסוח וניתוח של הבעיה האלגוריתמית:** זה בלתי אפשרי לתכנת באופן ידני תוכנה שמסוגלת לזיהוי דברים בתמונות, בגלל שפעולה זו היא מורכבת בהרבה מכל מערכת ידנית שיצרנו עד עתה, ואפילו אנחנו לא יודעים לאבחן *כיצד*, בדיוק, אנחנו מזהים דברים. המחשב רואה תמונה בתור רצף של ביטים, אין לא שום הבנה אינטואיטיבית של צורות או עצמים.

**אלגוריתמים קיימים לפתרון הבעיה:** ראה הסבר על רשתות נוירונים בתיאור טכנולוגיית המערכת.

הפנייה למקורות רלוונטיים: <http://neuralnetworksanddeeplearning.com>, <https://www.youtube.com/watch?v=aircAruvnKk>, <https://www.youtube.com/watch?v=i8D90DkCLhI&feature=youtu.be>

1. תיאור מבני הנתונים:

**פירוט מבני הנתונים:**

מולטי-רשת: מאגר של 5 רשתות נוירונים שמבצעות הצבעות בכדי להחליט.

רשת נוירונים: ראה הסבר על רשתות נוירונים בתיאור טכנולוגיית המערכת.

שכבת נוירונים: ראה הסבר על רשתות נוירונים בתיאור טכנולוגיית המערכת. ישנן מספר סוגי שכבות שונים, ולכל אחת מחלקה.

**פירוט מאגרי המידע של המערכת:** מאגרי המידע של המערכת הינם הדוגמאות מהן היא לומדת, והרשת הסופית, שנשמרת.

הדוגמאות ללמידה של המערכת הינן תמונות שעברו ניתוח – כל פיקסל קיבל ביית אחד, שנע מ 0 ל 1, כאשר 0 זה לבן, ו 1 זה שחור. בנוסף, לכל דוגמא מצורפת התשובה. את הדוגמאות מחלקים לשני חלקים, דוגמאות למידה, ודוגמאות בדיקה. המערכת משתמשת בדוגמאות הלמידה כדי ללמוד, ומדוגמאות הבדיקה היא לא לומדת, אלא רק בודקת את רמת הדיוק שלה עם דוגמאות שהיא עוד לא ראתה.

הרשת נשמרת פשוט בעזרת מודול cPickle