**Assignment 4**

* שיר כהן, 315805168
* עמית שקרצ'י, 313278889

# DATASET

**תיאור ה-datasets בהם נעשה שימוש:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Number of classes** | **Number of Samples:** | **Name** |
| The MNIST database of handwritten digits. | 10 | 70,000 | **Mnist** |
| images of beans taken in the field using smartphone cameras. | 3 | 1,295 | **Beans** |
| Binary 20x16 digits of '0' through '9' and capital 'A' through 'Z'. | 36 | 1,404 | **binary\_alpha\_digits** |
| The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. | 10 | 60,000 | **cifar10** |
| Images of healthy and unhealthy citrus fruits and leaves. | 4 | 594 | **citrus\_leaves** |
| The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. | 120 | 20,580 | **stanford\_dogs** |
| Cassava consists of leaf images for the cassava plant depicting healthy and four disease conditions | 5 | 9,430 | **Cassava** |
| Images of hands playing rock, paper, scissor game. | 3 | 2,892 | **rock\_paper\_scissors** |
| A large set of images of horses and humans. | 2 | 1,283 | **horses\_or\_humans** |
| The Dmlab dataset contains frames observed by the agent acting in the DeepMind Lab environment, which are annotated by the distance between the agent and various objects present in the environment. The goal is to is to evaluate the ability of a visual model to reason about distances from the visual input in 3D environments. | 6 | 110,913 | **Dmlab** |
| This dataset contains images of - Handwritten Bangla numerals | 10 | 6,000 | **Cmaterdb** |
| Stanford Online Products Dataset | 12 | 120,053 | **stanford\_online\_products** |
| The STL-10 dataset is an image recognition dataset for developing unsupervised feature learning, deep learning, self-taught learning algorithms | 10 | 13,1000 | **stl10** |
| A large set of images of flowers | 5 | 3,670 | **tf\_flowers** |
| A large set of images of cats and dogs. | 2 | 23,262 | **cats\_vs\_dogs** |
| UC Merced is a 21 class land use remote sensing image dataset, with 100 images per class. | 21 | 2,100 | **uc\_merced** |
| Kuzushiji-MNIST is a drop-in replacement for the MNIST dataset (28x28 grayscale, 70,000 images), provided in the original MNIST format as well as a NumPy format. | 10 | 70,000 | **kmnist** |
| The Oxford Flowers 102 dataset is a consistent of 102 flower categories commonly occurring in the United Kingdom. | 102 | 8,189 | **oxford\_flowers102** |
| This dataset consists of 101 food categories | 101 | 101,000 | **food101** |
| The DeepWeeds dataset consists of 17,509 images capturing eight different weed species native to Australia in situ with neighbouring flora.The selected weed species are local to pastoral grasslands across the state of Queensland. | 9 | 17,509 | **deep\_weeds** |
| EuroSAT dataset is based on Sentinel-2 satellite images covering 13 spectral bands and consisting of 10 classes with 27000 labeled and geo-referenced samples. | 10 | 27,000 | **eurosat** |

# PAPER

**Masksembles for Uncertainty Estimation**

לחץ [כאן](https://arxiv.org/pdf/2012.08334.pdf) על מנת להגיע אל המאמר המקורי.

**IntroDUCTION**

הערכת האמינות ויכולת החיזוי של רשתות נוירונים עמוקות, היא נושא המחקר של המאמר. הכותבים מתייחסים ל-MC-Dropout ול-Deep Ensembles בתור פתרונות פופולריים לקבלת חיזוי אמין. שתי השיטות משתמשות ב-ensembles כדי להעריך את מידת אי הוודאות בחיזוי.

MC-Dropout- אימון של רשת אחת, בה מבצעים dropout בזמן האימון וגם בזמן הבדיקה של המודל. את החיזוי ניתן לעשות מספר פעמים, ומשום שאנו מבצעים dropout- נקבל פרדיקציות שונות בכל פעם. ביצועי השיטה ירודים יחסית במשימות של הערכת אי וודאות.

Deep Ensemble- בניית אנסמבל של מספר מודלים בלתי תלויים, וחישוב הפרדיקציה על ידי ממוצע של כל התוצאות מהמודלים. ביצועי השיטה טובים יחסית במשימות של הערכת אי וודאות, אך זמני האימון וה-inference גדולים והשימוש בזכרון רחב מאוד.

כותבי המאמר מציגים את Masksembles, גישה להערכת אי ודאות עם ביצועים זהים ל- Deep Ensemble, אך בעלויות חישוב נמוכות יותר. בשיטה זו, מבצעים dropout לפרמטרים של המודל בצורה מובנית (ולא רנדומלית כמו ב-MC-Dropout).

בגישה זו, משתמשים במספר מוגדר מראש של binary masks, לפיהם נבחרים הנוירונים להם יבוצע drop. בזמן האימון- בוחריםmask רנדומלית, בדומה ל-dropout רגיל. במהלך ה- inference, מריצים את המודל מספר פעמים- כמספר ה-mask-ים כדי לייצר סט של פרדיקציות והערכת אי וודאות.

השיטה מקבלת מספר פרמטרים:

* מספר ה-mask-ים הכולל.
* ממוצע של חפיפה בין mask-ים שונים
* מס' ה-1 וה-0 בכל mask.

הגדרת הפרמטרים יוצרת מודל שנע בין Deep Ensemble לבין MC-Dropout.

**METHOD**

יצירת ה-mask-ים:

* N- מספר ה-mask-ים.
* M- מס' האחדות בכל mask. ערך זה שווה ל-model capacity, מספר פונקציות האקטיבציה שנעשה בהן שימוש במהלך forward pass אחד.
* S- פרמטר לקביעת רמת החפיפה בין mask-ים שונים.

1. ראשית, מייצרים N וקטורים מלאים באפסים, בגודל .
2. הופכים M ערכים רנדומליים בכל וקטור ל-1.
   1. אם בוקטור החדש ישנו מצב של רצף אפסים- מצב בו אף אחד מה-mask-ים אינו משתמש בפיצ'ר, נסיר את הפיצ'רים הללו.

Diagram

Description automatically generated

מגבלות האלגוריתם:

* בכל פעם שנרצה לשנות את M, N או S, נצטרך לבצע התאמה למספר השכבות ברשת, על מנת למנוע מצב בו בפועל אין dropout בכלל.

יתרונות האלגוריתם:

הגדרת הפרמטרים S,M,N ממקמת את Masksembles על הרצף בין MC-Dropout לבין Deep Ensembles. כאשר S שואף לאינסוף, אין לנו חפיפה כלל ב-mask-ים והתנהגות הרשת היא כשל Deep Ensembles. כאשר S=1, יש לנו חפיפה מלאה והתנהגות הרשת היא כשל MC-Dropout. הדבר נותן למתכנן הרשת את היכולת לייצר הערכת אמינות טובה יותר מזו ש-MC-Dropout מספק, בבזבוז של פחות מקום וזמן חישוב מאשר Deep Ensembles.

**INTRODUCTION**

אנסמבל היא שיטה שמשתמשת במספר מודלים כך שהוא מאפשר לקבל תוצאה טובה מכל אחד מהמודלים לבד. המצב האידיאלי שכל מודל יעשה טעויות בלתי תלויות במודלים אחרים וכך הדיוק יעלה.

ניתן לאמן רשתות נוירונים שונות אשר כל אחת מהן תתכנס אך כאשר הן יקבלו דוגמאות מה-data הן יחזו תחזיות שונות. ולכן משתמשים באנסמבל על מנת לבצע ממוצע של ההצבעות.

כדי לייצר אנסמבל עבור רשתות נוירונים –

1. מבחינה חישובית -backpropagation לכל מודל
2. מבחינת זיכרון – פרמטרים ומשקולות שונים לכל מודל

המטרה של המאמר הוא להתמודד עם הבעיה של החישוב והזיכרון כאשר מייצרים אנסמבל של רשתות נוירונים.

מראים כי לשיטה שלהם יש את הפיזור טוב ביותר בין דיוק, זמן ריצה וזיכרון בארכיטקטורות שונות של למידה עמוקה - CIFAR-10/100 classification with ResNet32 and WMT14 EN-DE/EN-FR machine translation with Transformer. בנוסף, מראים כי השיטה מתאימה ל- out-of-distribution datasets uncertainty evaluation on contextual bandits.

**METHOD**

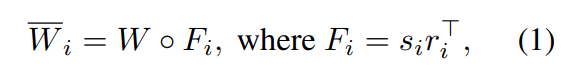
W– משקלים ברשת נוירונים, משקל איטי, משקל משותף.

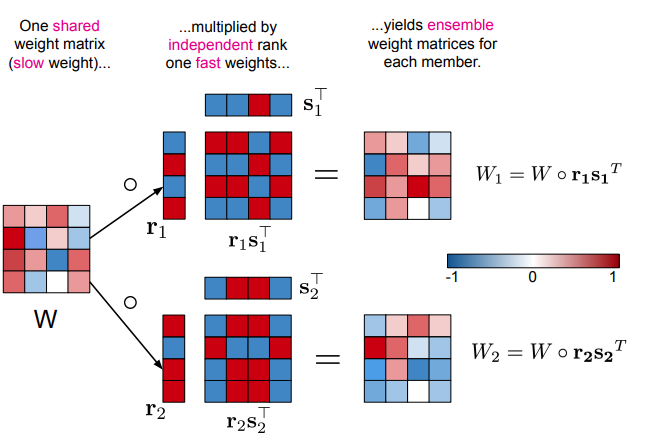
m – ממד הקלט

n – ממד הפלט

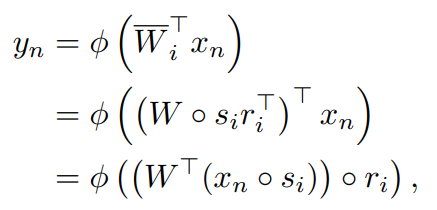
F – משקל מהיר

נניח כי יש M מודלים ולכל מודל יש מטריצת משקל Wi. לכל אחד מהמודלים יש שני וקטורים של הקלט והפלט (r ו- s) בעלי אותו גודל עבור המודל. הם יוצרים עבור כל מודל את המשקלים לפי הנוסחה הבאה –





מראים כיצד להפוך את מנגנון ייצור המשקולות למקבילי על ידי חישובים שעושים ב-batch. נגדיר את X כאקטיבציה ולכן האקטיבציה של השכבה הבאה תוגדר על ידי -



כאשר φ מייצג את האקטיבציה. הפלט מייצג את האקטיבציה לשכבה הבאה. כדי לייעל את החישובים מגדירים מטריצת R ו-S עבור כל דוגמה ב-minibatch –



X הקלט. וכך ניתן להשיג את האקטיבציה של כל מודל באופן מהיר יותר.

הגדרת הפלט – לוקחים את ממוצע התחזיות של כל מודל. נניח כי גודל ה-batch הוא B ויש M מודלים. ולכן אנחנו מבצעים B\*M חישובים וזה מאפשר לחשב את ה-forward במעבר יחיד- באמצעות המטריצה שמייצגת את המשקולות של כל המודלים באנסמבל.

OMPUTATIONAL COST

במקום לבצע כפל מטריצות הם משתמשים ב- Hadamard product שהוא יעיל יותר.

מגבלה של השיטה batchEnsamble – בחלק הזה (לא ברורה המגבלה) משהו עם ה-batch size.

צריך לשמור בזכרון שני סטים של וקטורים בלבד (S ו-R), יותר חסכוני מלשמור מטריצת משקולות.

BATCHENSEMBLE AS AN APPROACH TO LIFELONG LEARNING

עלות הזיכרון הינו החלק המשמעותי באנסמבל ו-bacthEnsmble פותר את הבעיה. ב- bacthEnsmble אין צורך לשמור דאטא ממשימות קודמות. בנוסף, bacthEnsmble צורך הרבה פחות זיכרון מאשר PNN משום שרק "משקולות מהירות" מתאמנות על המשימה החדשה שניתנה (בשיטה זו כל מודל לומד משימה אחרת, כך שהמטריצה W מכילה משקולות לכלל המשימות שנלמדו)

**מגבלות** –

במאמר ציינו שתי מגבלות:

1. בשיטה זו כל משימת lifelong learning מקבלת בסופו של דבר ייצוג במטריצה בדרגה 1. הגבלה של הייצוג של משימה למטריצה בדרגה 1 מקשה על ביצועים כאשר מדובר במשימות שונות מאוד אחת מהשנייה. (יכול להיותשאם נייצג את המשימה שנלמדה במטריצה גדולה יותר, נחזיק ידע רחב יותר לגבי המשימה ונוכל להגיע לפתרונות טובים גם עבור משימות שונות מאוד).

2. המשקל המשות, נוצר מאימון על המשימה הראשונה בלבד, כך שרק המידע מהמשימה הראשונה מועבר למשימו עוקבות (כאשר ככל הנראה אין תלות בין המשימות האחרות ולא ניתן "ללמוד" מהן). שהוצע הוא שימוש ב- lateral connections, כמו שמבוצע ב-PNN. (עמית- בדוגמא שראיתי הוסיפו connections בין נוירונים בשכבה החבויה, כמו בתמונה הבאה:

Diagram

Description automatically generated)

**DIVERSITY ANALYSIS**

מטרת חלק זה הייתה לבדוק מהי רמת ה-diversity שניתן להגיע אליה באמצעות שימוש במטריצה בדרגה 1, שכן ככל שרמת ה-diversity גבוהה יותר כך ביצועי האנסמבל טובים יותר.

השוו את השיטה dropout ensemble ו- naive ensemble לפי ממד diversity שבודק את המחלוקות בין המודלים במערת ה-test.

המודל הנאיבי מאתחל את המשקולות בצורה רנדומלית. ב- dropout יש את אותם פרמטרי אתחול. השיטה שלהם מאותחל בצורה רנדומלית.

ולכן, כאשר יש מעט נתוני אימון אז השיטה שלהם והנאיבית עובדות טוב כי יש מגוון רב בין המודלים.

ולכן מצאו כי השיטה מתאימה כאשר יש מעט נתוני אימון (CIFRA100 לעומת CIFRA10).

**CONCLUSION**

ניתן להשתמש בשיטה לשיפור הדיוק וחוסר הוודאות כמו כל שיטת אנסמבל רגילה. היתרון בשיטה שהיא מסירה את צוואר הבקבוק של הזיכרון.

[**Papers with code**](https://paperswithcode.com/paper/batchensemble-an-alternative-approach-to-1#code)

[**Git google wraps model**](https://github.com/google/uncertainty-baselines)

[**Git papers**](https://github.com/google/edward2)

**דו"ח**

**שלב 1 – בחירת אלגוריתם מאמר**

1. תיאור האלגוריתם
2. יתרונות בשימוש באלגוריתם הזה לעומת אחרים
3. חסרונות של האלגוריתם

**שלב 2 – הצעת שיפור**

להציע שיפור לאלגוריתם.

**שלב 3 – בחירת אלגוריתם מוכר להשוואה**

לבחור אלגוריתם שיהווה base line שאליו נשווה.

**שלב 4 – הערכה**

1. Data – בחירת 20 dataset מתוך רשימה. ניתן להוסיף data שייתכן שיהיה יותר טוב עבור השיפור שהצענו.
2. הערכה –
   1. לעשות 10 cv
      1. מריצים לפחות 50 ניסיונות –
         1. 3 cv לאופטימיזציה של שלושת האלגוריתמים (לפי מטריקה למשל cross entropy) עבור 2 פרמטרים **לפחות**. מומלץ להשתמש ב- Bayesian Optimization.
         2. נבחר את הפרמטר שהיה הכי טוב לפי מיצוע של ה-3 הרצות.
   2. לוקחים את האופטימיזציה הכי טובה ובודקים על ה-train.
3. מדדים

**שלב 5 – ניתוח סטטיסטי**

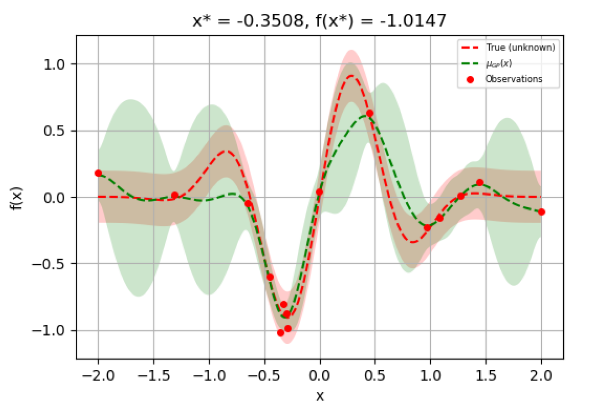
להשתמש במבחן פרידמן (anova).

[**הסבר על השיטה Bayesian Optimization –**](https://github.com/mardani72/Hyper-Parameter_optimization/blob/master/Hyper_Param_Facies_tf_final.ipynb)

נרצה לבצע אופטימיזציה לפרמטרים של הרשת נוירונים. ניתן לחלק את האופטימיזציה לשני חלקים –

1. פרמטרי אימון – למשל המשקולות של רשת הנוירונים אשר נלמדים בתהליך האימון של האלגוריתם.
2. Hyper-parameters – פרמטרים שנין לקבוע לפני האימון כמו מקדם הלמידה ומספר השכבות. הבחירה של הפרמטרים האלו היא מתישה וקשה מאוד למצוא את הפרמטרים הטובים ביותר.

שיטת ה-grid search מבצע לולאה אשר עוברת על כל טווח הערכים האפשריים של הפרמטרים שהגדרנו. הבעיה בשיטה הנ"ל הוא זמן הריצה הארוך. למשל עבור 3 פרמטרים שלכל אחד מהם יש 10 אפשרויות השיטה תרוץ 10^3 פעמים. שיטה אחרת היא random search אשר לא עוברת על קומבינציות אלא מבצעת חיפוש אקראי אחרי האופטימיזציה האופטימלית. כאשר יש הרבה פרמטרים לכוונן שיטה זאת פחות יעילה.

השיטה שבה נשתמש היא Bayesian Optimization אשר מנסה למדל את ההשפעה של ערכי הפרמטרים על התוצאה שלנו [בהמשך להשלים איזה מדד בחרנו למקסם]. השיטה מניחה התפלגות גאוסיאנית. ציר ה-X מציין טווח של פרמטר שעליו עושים את תהליך ה-tuning, ציר ה-Y מציין את המדד. נרצה שהמודל שלנו יצליח להעריך את הפונקציה האדומה (הפונקציה האמיתית שבה הפרמטר שלנו מתנהג). הקו הירוק הירוק מייצג את הפונקציה שכרגע אנחנו מצליחים לייצר והשטח הירוק מראה את האי וודאות שלנו. ככל שהשטח גדול כך אנחנו פחות יודעים על האזור הזה. לאחר שנבדוק מספר נקודות ננסה להעריך מה הנקודה הבאה שמשתלם לבדוק.

החלטנו לאמוד את הפרמטרים הבאים – [להשלים]

הסיבה שבחרנו לאמוד את הפרמטרים היא [להשלים]