# מטלה 3 (מטלה 2 חלק ב)

* מומלץ לעבוד עם google colab זהו תוסף פשוט שמסיפים לגוגל דרייב שלכם ומאפשר להריץ פייטון ונותן לכם גישה חינמית לGPU. להתקנה [הוראות התקנה](https://cs231n.github.io/setup-instructions)  
  יש לשמור את התיקיה עם הקבצים המסופקים בדרייב (שמות באנגלית בלבד גם בתיקיות הביניים)   
  יש להשלים את הקוד במחברות היופיטר ובעיקר בקבצים המסופקים של המסווגים.   
  ריבוע הקוד הראשון בכל מחברת נותן לcollab גישה לדרייב שלכם שיוכל לזהות את הקבצים. יש להזין לתוך המשתנה FOLDERNAME את התיקיה בא שמרתם את המטלה.  
  שימו לב שתצטרכו עוד כ200 מגה בייט בשביל הדאטה סט שיורד לכם אוטומטית.  
  לאחר סיום המחברות יש להריץ את המחברת collect\_submitions.ipynb מה שיצור לכם קובץ זיפ וPDFים עם התוצאות שלכם (יש לוודא שהמחברות שמורות לאחר הרצה מוצלחת כי אנו בודקים את התוצאות) להגיש את הזיפ שנוצר ואת הPDFים
* ניתן לעבוד לוקאלית למרות שזה פחות מומלץ. במקרה זה הריבוע הראשון במחברות מיותר. יש לעבוד על לינוקס בלבד. כדי להוריד את הדאטה סט יש להריץ את הקובץ get\_dataset.sh  
  כדי להגיש יש להריץ את collectSubmissions.sh

במטלה זו תממשו שכבות רגולריזציה (batch norm, layer norm, dropout)

תתנסו לשיפורי הביצועים ששכבות אלו מציעות.

תממשו אלגוריתמי אופטימיזציה (adam,sgd momentum ,rmsprop)

בנוסף תתנסו לראשונה בפייטוץ. סביבת העבודה של פייסבוק לכתיבת רשתות עמוקות.

עליכם להשלים את המחברות שבתרגיל

1. FullyConectedNets.ipynb
2. Dropout.ipynb
3. BatchNormalization.ipynb
4. PyTorch.ipynb

חלק מהשכבות שתשתמשו בהם כבר מיששתם במטלה הקודמת לכן אין צורך לממש אותם שוב. המימוש שלהם כבר נמצא במקומות הרלוונטיים. (מישהו לב למי שעוד לא הגיש את מטלה 2 חלק א אין להעתיק את הפתרון הנתון כאן למטלה הקודמת)

כדי להשלים את המחברות עליכם להוסיף מימושים לפונקציות שנמצאות בקבצים:

* CV7062610
  + Layers.py (9 functions)
  + Optim.py (3 function)

**שימוש לב**: זהו איננו קורס תכנות לכן מימשנו עבורכם את כל המעטפת הדרושה כדי להריץ את התרגילים. אתם נדרשים לממש רק את מה שקשור ישירות ללמידה עמוקה כמו שלמדנו בכיתה.

**המימושים של הפונקציות קצרים מאוד**.במימוש נכון רוב הפונקציות דורשות לא יותר מ10 שורות קוד (לרוב פחות) למרות שיכולות להיות 2-3 פונקציות שידרשו 20-30 שורות. אין צורך (למרות שמותר לכם) להוסיף פונקציות חדשות.

**אם אתם רואים שאתם כותבים בלוקים ארוכים של קוד אז כנראה הסתבכתם ויש לחזור על התאוריה עם דף העט**. הפתרון הרשמי מכיל פחות מ200 שורות קוד חדשות כולל הערות ושורות ריקות.

סדר העבודה אינו בדיוק לינארי.

1. עליכם להתחיל במחברת FullyConectedNets ולהשלים אותה כמעט עד הסוף פרט לבלוק האחרון בו אתם נדרשים לממש רשת כאוות נפשכם.
2. להשלים את מחברות הbatchnorm dropout בכל סדר שתרצו
3. לסיים את מחברת הfullyConected
4. להשלים את מחברת הpytorch

# פרויקט זיהוי אורז:

עליכם להוריד את הדאטה סט מכאן

https://drive.google.com/file/d/1eSp5f5ih17blcqjgxJQ1IKx9a7QXTqJT/view?usp=sharing

לפי מה שלמדנו בתרגול עליכם תכתוב pytorch dataset (אתם מורידים פשוט תמונות בתיקיות) ומודל קלסיפיקציה לבחירתכם שמזהה את 5 סוגי האורז. עליכם להגיע למעל 90% דיוק.

עליכם להגיש את:

* כל קבצי הקוד כקובץ זיפ (ללא הדאטה סט)
* קובץ לוג שמראה את אימון וריצה בזמן אמת של הקוד שלכם
* קובץ עם המודל המאומן שלכם שמור
* סקריפט קטן שיודע לקחת תמונה אחת של אורז (משורת ריצה) ולכתוב למסך פלט של איזה סוג אורז זה.

Script example:

Python myScript.py ../dataset/some\_imageof\_rise.jpg

Output:

Basmati

בכל שאלה אתם מוזמנים לפנות לפורום של קורס במודל של התרגול או למייל שלי neiterman21@gmail.com