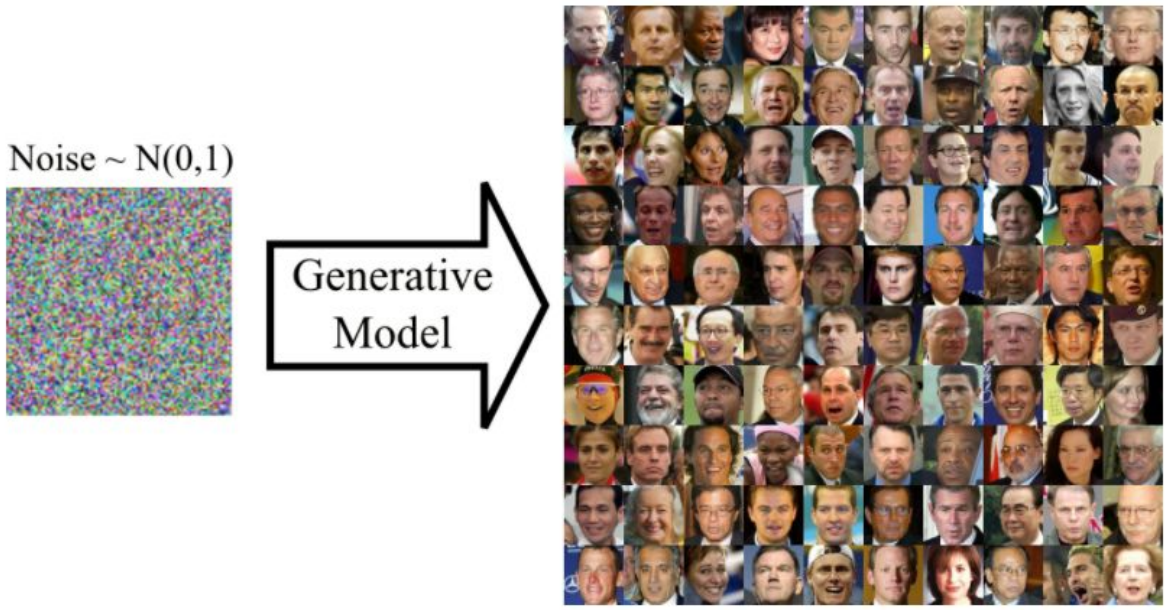
מימוש GAN ליצירת פרצופים

ברכותיי! זהו התרגיל המסכם את הפרק שלנו על רשתות נוירונים. בהתאם, הוא יהיה מעניין ומאתגר יותר מהקודמים. בתרגיל זה נממש מודל שמקבל רעש אקראי ומייצר ממנו פרצופים אנושיים באמצעות ארכיטקטורת GAN.



את ה- dataset לצורך העבודה ניתן למצוא בתיקיית התרגיל- ברשותך כ- 13,000 תמונות פרצופים שונות, מחולקות בתיקיות. בהמשך המסמך אציע ארכיטקטורות לרשתות המרכיבות את ה- GAN. מדובר בהמלצה בלבד, ואתה כמובן רשאי להתייחס אליהן כך.

בניית ה- discriminator:

אבן הבניין הבסיסית שלנו ביצירת ה- discriminator היא שכבת הקונבולוציה. בנוסף אליה, מומלץ להשתמש גם בשכבת Max Pooling, כדי להקטין את מימדי התמונות. אחת מפונקציות האקטיבציה המומלצות לעבודה עם תמונות היא פונקציית ה- ReLu, ובה נשתמש גם כאן. יהיה לנו מאוד חשוב להשתמש בטכניקת Dropout במימוש שלנו, כדי לאפשר אימון מוצלח (עוד על כך בהמשך).

על- כן, נשתמש למעשה באבני הבניין הבאות:

- Convolutional layer
- Max Pooling layer
- ReLu activatoin
- Dropout layer

נשתמש ב"בלוק" שיורכב מארבע השכבות האלו, אחת אחרי השנייה. הרשת תכיל שלושה בלוקים כאלה, שיבואו אחד אחרי השני. החלק האחרון ברשת יהיה מעבר לחיזוי פשוט של True/False באמצעות שיטוח התמונות לכדי וקטור אחד והוספת כמה שכבות קשירות לחלוטין (גם כאן מומלץ להוסיף שכבות Dropout).

<u>בניית ה- generator:</u>

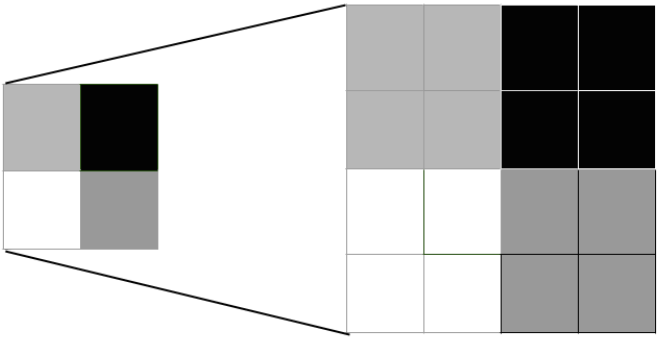
ה- generator יורכב באופן דומה ל- discriminator, אלא שניתקל בבעיה- פונקציות כדוגמת קונבולוציה או pooling אינן פונקציות הפיכות, כך שאין מושג ברור לגבי מהי deconvolution או מהו unpooling.

<u>שכבת deconvolution:</u>

נשתמש פשוט בשכבת convolution, כאשר אנו מגדילים את התמונה (נקבל את השוליים, אותם בשכבת קונבולוציה רגילה אנו לרוב מסירים).

<u>שכבת unpooling:</u>

נשתמש ב- unpooling לפי העיקרון הבא:



למרבה המזל, כבר יש לנו שכבה כזו בארגז הכלים שלנו- מימשת אותה ממש לאחרונה.

ב- generator נשתמש באבן בניין נוספת, שלאו דווקא נרצה להשתמש בה ב- discriminator: שכבת batch normalization. עוד עליה בהמשך המסמך.

בהתאם לכך, נשתמש בבניית ה- generator באבני הבניין הבאות:

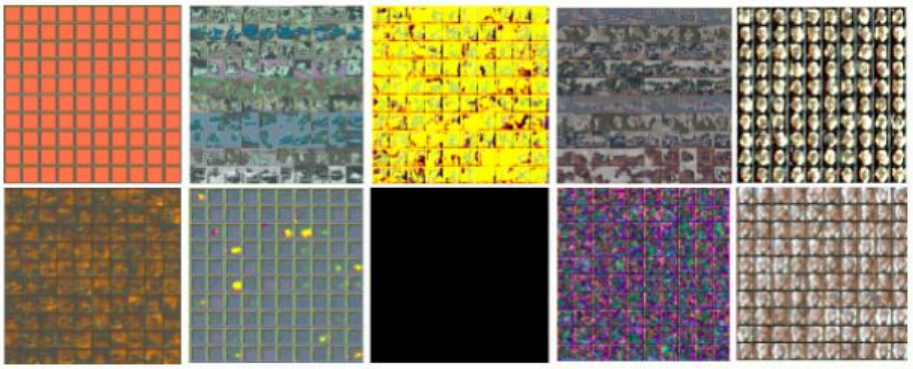
- Unpooling layer
- Deconvolutional layer
- ReLu activation

- Dropout layer
- Batch Normalizatoin layer

גם כאן נשתמש ב"בלוק" שיורכב מהשכבות הללו בסדר הזה. כמובן שלפניו יבואו כמה שכבות שתפקידן לקחת את וקטור הרעש שנקבל ולהעבירו לתמונה.

<u>סוגיות באימון הרשתות</u>

כמו שכבר שמת לב, האימון של GAN הוא לא טריוויאלי ודורש זהירות.



בתמונה ניתן לראות דוגמאות למצבים בהם ה- generator נופל למקרה קצה ספציפי, ולא מצליח ללמוד בצורה טובה, לרוב כיוון שה- discriminator לא מצליח להבדיל את התמונות הללו מפרצופים.

<u>ניטור האימון וביצוע התאמות:</u>

לאורך האימון נדפיס גרפים של מידת ההצלחה של ה- discriminator בחיזוי תשובה חיובית לדוגמאות אמת ובחיזוי תשובה שלילית לדוגמאות מזויפות,בנוסף לגרף , של מידת ההצלחה של ה- generator . נקפיד על שוויון בין הצלחת ה- generator והצלחת ה- discriminator. אם אחת הרשתות טובה מדי, לא נאמן אותה בתורה. מומלץ לגבש לכך כללי אצבע ולדבוק בהם.

<u>שימוש ב- Dropout:</u>

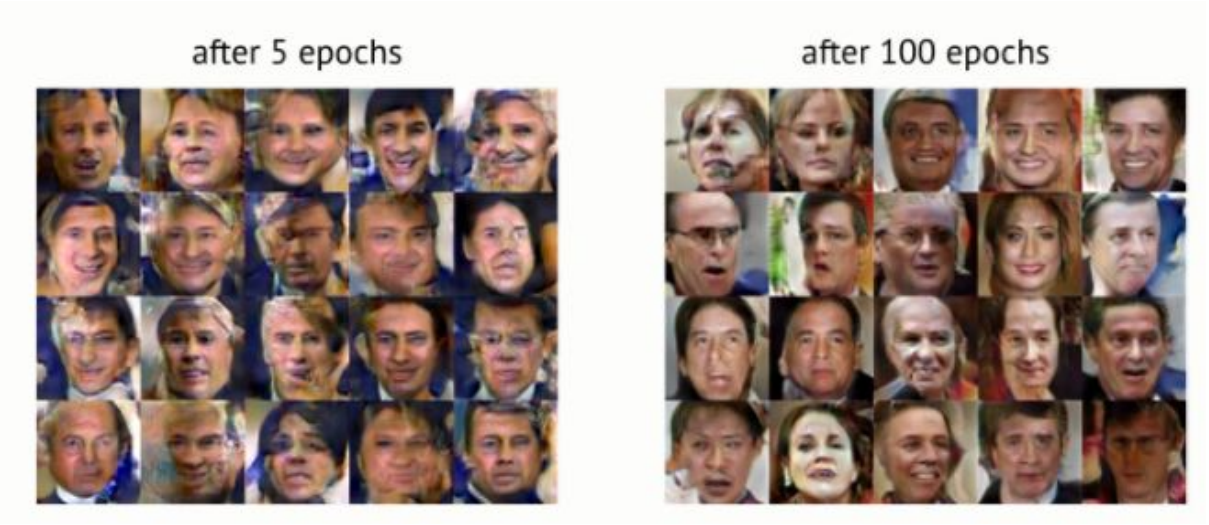
כדי להימנע ממצבים כמו בתמונה לעיל מומלץ לעשות שימוש נרחב ב- dropout, במיוחד ב- discriminator.

<u>שימוש ב- Batch Normalizatoin:</u>

קרא את המאמר המצורף בתיקיית התרגיל על Batch Normalizatoin. ל- Batch Normalizatoin יש יכולת טובה להאיץ ולשפר את האימון. מומלץ להשתמש בו במסגרת ה- generator (שהוא

מלכתחילה החוליה החלשה יותר בארכיטקטורה). לאו דווקא נרצה להשתמש בו במסגרת ה-discriminator, כדי לא להגיע ל- discriminator טוב מדי.

במהלך האימון אתה צפוי להגיע לתמונות שידמו יותר ויותר לפרצופים אנושיים.



מצורף כאן קישור לבלוג ממנו נלקח התרגיל. מומלץ לא להשתמש בו בטרם כלו כל הקיצים- מהות התרגיל היא מימוש עצמי.

שימוש בארכיטקטורת WGAN

אחרי שהתנסית עם ה- GAN, קרא את המאמר על WGAN. זו הרחבה של GAN שהוכיחה ביצועים טובים בהרבה מה- GAN המקורי. המאמר יפנה אותך לקוד שמממש את המאמר. הפעל אותו על הבעיה שלך והבן כיצד הוא עובד.

בהצלחה!