

Contents

1. Introduction	6
1.1 What is Machine Learning?	6
1.1.1 The Basic Concept.....	6
1.1.2 Data, Tasks and Learning.....	7
1.2 Applied Math	8
1.2.1 Linear Algebra.....	8
1.2.2 Calculus.....	13
1.2.3 Probability	15
1. References.....	25
2. Machine Learning.....	26
2.1 Supervised Learning Algorithms.....	26
2.1.1 Support Vector Machines (SVM).....	26
2.1.2 Naive Bayes	29
2.1.3 K-Nearest Neighbors (K-NN)	31
2.1.4 Quadratic\Linear Discriminant Analysis (QDA\LDA)	33
2.1.5 Decision Trees.....	35
2.2 Unsupervised Learning Algorithms.....	49
2.2.1 K-means.....	49
2.2.2 Mixture Models.....	50
2.2.3 Expectation–maximization (EM)	52
2.2.4 Hierarchical Clustering.....	54
2.2.5 Local Outlier Factor (LOF).....	55
2.3 Dimensionally Reduction.....	57
2.3.1 Principal Components Analysis (PCA)	58
2.3.2 t-distributed Stochastic Neighbors Embedding (t-SNE).....	62
2.4 Ensemble Learning.....	65
2.4.1 Introduction to Ensemble Learning.....	65
2.4.2 Bootstrap aggregating (Bagging).....	65
2.4.3 Boosting.....	67
2. References.....	70
3. Linear Neural Networks	71
3.1 Linear Regression	71
3.1.1 The Basic Concept.....	71
3.1.2 Gradient Descent	73

3.1.3 Regularization and Cross Validation	73
3.1.4 Linear Regression as Classifier	74
3.2 Softmax Regression.....	76
3.2.1 Logistic Regression.....	76
3.2.2 Cross Entropy and Gradient descent.....	77
3.2.3 Optimization	78
3.2.4 SoftMax Regression – Multi Class Logistic Regression	79
3.2.5 SoftMax Regression as Neural Network.....	80
3. References.....	81
4. Deep Neural Networks	82
4.1 Multilayer Perceptron (MLP).....	82
4.1.1 From a Single Neuron to Deep Neural Network.....	82
4.1.2 Activation Function	83
4.1.3 Xor	85
4.2 Computational Graphs and propagation.....	86
4.2.1 Computational Graphs.....	86
4.2.2 Forward and Backward propagation.....	86
4.3 Optimization	88
4.3.1 Data Normalization	88
4.3.2 Weight Initialization	88
4.3.3 Batch Normalization.....	89
4.3.4 Mini Batch	89
4.3.5 Gradient Descent Optimization Algorithms	90
4.4 Generalization.....	92
4.4.1 Regularization.....	92
4.4.2 Weight Decay.....	93
4.4.3 Model Ensembles and Drop Out.....	93
4.4.4 Data Augmentation.....	94
4. References.....	95
5. Convolutional Neural Networks (CNNs).....	96
5.1 Convolutional Layers.....	96
5.1.1 From Fully-Connected Layers to Convolutions	96
5.1.2 Padding, Stride and Dilatation.....	97
5.1.3 Pooling.....	98
5.1.4 Training.....	99
5.1.5 Convolutional Neural Networks (LeNet).....	99

5.2 CNN Architectures	100
5.2.1 AlexNet.....	100
5.2.2 VGG	101
5.2.3 GoogleNet.....	101
5.2.4 Residual Networks (ResNet).....	102
5.2.5 Densely Connected Networks (DenseNet).....	103
5.2.6 U-Net.....	103
5.2.7 Transfer Learning	104
5. References.....	105
6. Recurrent Neural Networks	106
6.1 Sequence Models	106
6.1.1 Vanilla Recurrent Neural Networks	106
6.1.2 Learning Parameters	107
6.2 RNN Architectures.....	108
6.2.1 Long Short-Term Memory (LSTM)	108
6.2.2 Gated Recurrent Units (GRU).....	109
6.2.3 Deep RNN	110
6.2.4 Bidirectional RNN	111
6.2.5 Sequence to Sequence Learning.....	111
6. References.....	111
7. Deep Generative Models.....	113
7.1 Variational AutoEncoder (VAE)	113
7.1.1 Dimensionality Reduction	113
7.1.2 Autoencoders (AE)	114
7.1.3 Variational AutoEncoders (VAE)	115
7.2 Generative Adversarial Networks (GANs)	118
7.2.1 Generator and Discriminator	118
7.2.2 Deep Convolutional GAN (DCGAN).....	121
7.2.3 Conditional GAN (cGAN)	121
7.2.4 Pix2Pix	121
7.2.5 CycleGAN	122
7.2.6 Progressively Growing GAN (ProGAN).....	122
7.2.7 StyleGAN	123
7.2.8 Wasserstein GAN	125
7.3 Auto-Regressive Generative Models	128
7.3.1 PixelRNN	129

7.3.2 PixelCNN	130
7.3.3 Gated PixelCNN.....	130
7.3.4 PixelCnn++	130
7. References.....	132
8. Attention Mechanism.....	133
8.1 Sequence to Sequence Learning and Attention	133
8.1.1 Attention in Seq2Seq Models	133
8.1.2 Bahdanau Attention and Luong Attention	133
8.2 Transformer.....	134
8.2.1 Positional Encoding.....	134
8.2.2 Self-Attention Layer.....	135
8.2.3 Multi Head Attention.....	137
8.2.4 Transformer End to End.....	137
8.2.5 Transformer Applications.....	138
9. Computer Vision	141
9.1 Object Detection.....	141
9.1.2 You Only Look Once (YOLO).....	141
9.1.4 Spatial Pyramid Pooling (SPP-net)	144
9.2 Segmentation.....	146
9.2.1 Semantic Segmentation vs. Instance Segmentation	146
9.2.2 SegNet neural network.....	146
9.2.3 Atrous Convolutions (Dilated Convolutions)	148
9.3 Face Recognition and Pose Estimation	149
9.3.1 Face Recognition.....	149
9.3.2 Pose Estimation	151
9.4 Few-Shot Learning.....	153
9.4.1 The Problem.....	153
9.4.2 Metric Learning	153
9.4.3 Meta-Learning (Learning-to-Learn).....	155
9.4.4 Data Augmentation.....	156
9. References.....	157
10. Natural Language Processing.....	158
10.1 Language Models and Word Representation.....	158
10.1.1 Basic Language Models.....	159
10.1.2 Word representation (Vectors) and Word Embeddings	161
10.1.3 Contextual Embeddings	165

10. References.....	170
11. Reinforcement Learning (RL).....	171
11.1 Introduction to RL.....	171
11.1.1 Markov Decision Process (MDP) and RL	171
11.1.2 Bellman Equation	173
11.1.3 Learning Algorithms.....	178

1. Introduction

1.1 What is Machine Learning?

1.1.1 The Basic Concept Artificial Intelligence (AI)

בינה מלאכותית הינה תחום בתוכנאות מחשב אֶלמנטן טכנולוגיית-המתקנה מנגנון חסיבה אנושי. בתחום רחב זה יש רמות שונות של בינה מלאכותית – יש מערכות שמסוגלות למדוד דפוסי התנהגות ולהתאים את עצמן לשינויים, ואילו יש מערכות שאינן מחקות-מנגנון חסיבה אנושי אך הן לא מתחכמתות מעבר למה שתכננו אותן בתחלת-השואב רובוטיזיודע לחשב את גודל החדר או אפסולו הניקוי האופטימלי-פועל לפי פרוצדורה ידועה מראש, ואין לכך תחוכם מעבר לתכונות הראשוני שלו. לעומת זאת תוכנה היודעת לסנן רישים באופן מסתגל, או להמליץ על שיירוף בגין מזיקה בהתאם לسانון של המשמש, משתמשות במבנה מלאכותית ברמה גבוהה יותר, כיון שהן לומדות עפ' הזמן דברים חדשים.

המונה בינה מלאכותית מתיחס בדרך כלל למערכות שמקהה התנהגות אנושית, אֲךְ היא יאשגרתית, לא לומדת משוה חדש, ועושה את אותו הדבר כל הזמן=מערכת זו יכולה להיות משוכלתת ולהשับ דברים מסווגים ואף להסתיק מסקנות על דוגמאות חדשות שהיא מעתים לא ראתה, אך תמיד בעבור אותן החלט (Input), היה אותן החלט (Output).

ニיחח לדוגמא מערךת סטרימינג של סרטיים, למשל Netflix. חלק משיפור המערכות והגדלת זמני הצפייה הניתן לבנות מגנון המלצות הבניי על היסטוריית השימוש של לקוחות של' במערכת=אייזה סרטים הם רואים, איזה 'زانרייף ומתי=קסיש' מעט צופים ומעט סרטיים, ניתן לעשות זאת באופן דינמי=למלאת בלילות של הנזוטים לנתנו קאותם ידניות ולבנות מערכת חזקים שמהווה מנוע המלצות מבודדים=AI. ניחח לדוגמא=אדם שצופה בה"פארק היורה" וב"אינדי אנדר" ג'ונס"=סביר שהמערכת **המליצה** לו לצפות גם ב-"**פולטרג'יסט**"=אדם שצופה לעומת זאת בא"האהבה בין הכרמים" ו"הבית על האגם", ככל הנראה כדי להמליץ לו על "הגשרים של מחוז מדיסון".

מערכת זו יכולה לעמוד טוב, אך בנסיבות מסוימות כבר לא ניתן לנוכח אותה כפרוצדורה מסודרת וכואסוף של חוקים ידוע מראש. מאגר הסרטים גדול, נוספים סוגים נוספים של סרטים (כמו למשל סדרות-תוכניות ריאליות ועוד) ובו נסוכות רצים להתייחס לפרמטרים נוספים-האם הצופה ראה את כל הסרט או הפסיק באמצע, מה גיל הצופה ועוד-מערכות הבניה באופן קלאסי, איננה מסוגגת להתמודד עם כמיון-המידע הקיימות, כמוות הכללי-פונדרש לחושב עליהם מראש היא עצומה ומורכבת לחישוב.

נתבונן על דוגמא נוספת – מערכת לניהוט רכב. ניתן להגדיר כל פשטוט בו אם משתמש יוציא מטל אובי ורואה להגעה לפתח תקווה-אך האפקטיביטה תיקח אותו דרך מסלול ספציפי-שנבחר מראש.=מסלול זה לא מתחשב בפרמטרים קרייטיים כמו מה השעה,האם יש פקידי-אפקטיביזציה=כמויות הפרמטרים שיש להתייחס אליהם איננה ניתנת לטיפול על ידי מערכת כללים ידועה מראש, וגם הפוטנציאליות המתאפשרת היא מוגבלת מאוד.=למשל לא ניתן לחזות מה תהיה שעת הפגיעה וכדומה.

Machine Learning (ML)

למידת מכונה הוא תחת חומר של בניית מלאכותית, הבא להתרמודד עם שני האתגרים ששתוראו קודם---היכולה לתכונת מערכת על בסיס מסופר של נתונים ופרמטרים, וחיזוי דברים חדשים כתלות בפרמטרים רבים שיכולים להשנותם עם הזמן. מנגנון- ML -מנתחים כמיות אדריות של דאטא ומנסות להציג לאיזו תוצאה. אם מדובר באפליקציית ניוט, המערכת תנתח את כל אוטם הפקטורים ותנסה לחשב את משך הנסעה המשוער. נניח והיא חצתה 20 דקות נסעה. אם בסופו של דבר הנסעה ארוכה-30 דקות, האלגוריתם ינסה להבין פקטורי השתנה במהלך הדרך ומדוע הוא נכשל בחיזוי (למשל---הכבד נושא נטבימט-אבלט-בקט-עמיסי---זהו אומצטמצף לאחד-זקמי-צער-יעוכב-זקע-יעוכב-קבוע-ברובב-שעווה-הימרה-ול-פקק-אקראי). בcheinת מספיק מקרים כאלה, האלגוריתם "מבחן" שהוא טועה, והוא פשוט יתקן את עצמו ויכניס למערך החישובים גם פקטורי של מספר נתיבים ויריד או' את המשקל של הטופרטור ביחס. וככה באפין חזרתו-האלגוריתם שוב ושוב מקבל קלט, מוציא פלט-בודק את התוצאה הסופית. לאחר מכן הוא בודק היכן הוא טעה. משנה אף עצמה. מתקו את המשקל שהוא גוטו לפקטורי שווים ומשתכלל מוסיפה לו' נסעה.

במערכות אל-הקלטן-נשאר לכארה קבוע, אבל הפלט' משתנה – עברו זמני יציאה שונים, האלגוריתם יעיר זמני נסעה שונים, כתלות בMagnitude הפרמטרים הרלוונטיים.

Mמערכות אַכְלָרַשְׁתּוֹת הַפְּרָסּוֹוְהַגְּדוֹלוֹת. כל אחת מנסה בדרכה שלה לחזות למשל, איזוח-משתמש שהקליק על המודעה צפי שיבצע רכישה. הפלטפורמות=מנסנות לזרות=כוונה=(Intent) על ידי למידה מניסיון. בהתחלה הן פשוט ניחשו על פי כמה פקטוריים שהזוזנו להם על ידי בני אדם. נניח, גוגל החליטה שמי שצופה

בஸרטוני יוטיוב של [datatboxing](#) הוא מוצג גבואה של רכישה. בהמשך הדריך, בהנחה והמשתמש מבצע רכישה כלשהי, האלגוריתם מקבל "נקודה טוביה". אם הוא לא קנה, האלגוריתם מקבל "נקודה רעה". ככל שהוא מקבל יותר נקודות טובות ורעות, האלגוריתם יודע לשפר את עצמו, לתת משקל גדול יותר לfrmteristic טוביים ולהזניח frmteristic פחות משמעותיים. אבל רגע, מי אמר למערכת להסתכל בכל בסרטוני [datatboxing](#)?

האמת שהיא שאף אחד. מישחו, בנאדים, אמר למערכת לזהות את כל הסרטוניים המשמשים לצופה בהם ביטויים=להזחות מתוך הסרטון, האודיו, תיאור הסרטון ומילוי המפתח וככל=אייזה סוג סרטון זה. יתקשחארי מיילארדי ציפוי בסרטונים, האלגוריתם מתייחס למצואו קשר בין סוג מסוים של סרטונים לבין פעולות כמו רכישה באתר. באופן זהה, גוגל מזינה את האלגוריתם בכל הפעולות המשמשים מבצע. המיללים שהוא קורא, המיקומות שהוא מסתובב בהם, התמונות שהוא מעלה לענן, ההודעות שהוא שולח=כל מידע שיש אליו גישה. הכל נשפר לטור מאגר הנתונים העצום בו מוסה גוגל לבנות פרופילים ולמצוא קשר בין הסיסמי שלו לרכושו או כל פעולה אחרת שבאה לה לזהות.

המכונה המופלאה זו לומדת כל הזמן דברים חדשים ומנסה כל הזמן למצאותה קשרים, לחזות תוצאה, לבדוק אם היא הצלחה, ואם לא לתקן את עצמה שוב ושוב עד שהיא פוגעת במטרה. חשוב לציין שלמכונה אין סנטימנטים, כל המידע קביל ואם היא תמצא קשר מוכח בין מידת הנעליה של הסרטוניים ביבי שארך, אז היא תשתחמש בו גם אם זה לא נשמע הגיוני.

חשוב לשים לבטען המטריה היא לא המצאה של האלגוריתם. הוא לא קם בבודק ומחליט מה האפליקציה שלכם צריכה לעשות. המטריה מוגדרת על ידי היוצר של המערכת. למשתמש=חישוב זמן נסעה, בנייה מסלול אופטימלי ביק^{א-ל-ב-ג-ו-ו}. המטריה של גוגל=שימוש יבצע רכישה=^ווהכל מתנקז לזה בסוף, כי גוגל בראש ובראשו היא מערךת פרטום. אגב, גם ההגדרה של מסלול "אופטימלי" היא מעשה ידי אדם. המכונה לא יודעת מה זה אופטימלי, זו רק מילה. אז צריך לעזור לה ולהגיד לה שאופטימלי זה מינימום זמן, מעט עצירות, כמה שפחות רמזורי^{ו-ו}=לסיום, המטריה מואפינית על ידי האדם ולא על ידי המכונה. המכונה רק חותרת למטריה שהוגדר לה

יש מנגןנו-ML מהתבססים על DATA דאטה מסודר ומתויג כמו Netflix, עם כל המאפיינים של הסרטים אבל גם עם המאפיינים של הצופים (מדינה, גיל, שעת צפייה וכו'). לעומת זאת יש מנגןנו-ML שמקבלים טיפה יותר וחופש ומתבססים על מידע חלקית מכך (יש להם מידע על כל הסרטים, אבל אין להם מידע על הצופה)=מנגנים אלו לא בהכרח מנסים למנוע המלצות אלא מנסים למצוא חוקיות בנתונים, חריגות וככל

כך או כך, המערכת הסבור זהה הקורוי-ML**Device** בבניי אלגוריתמים שונים המiomנים בניתוח טקסט, אלגוריתמים אחרים המתמקדים בעיבוד אודיו, ככל המתוחים היסטוריית גישה או זיהוי מתוך דף ה-Web^ובו אתם צופי&זע. عشرות מאות מנגנים נאלה מסתובבים ורצים ובונם את המפה השלמה. ככה רוב רשות הפרסום הגדלות עובחת. ככל שהמכונה של גוגל/פייסבוק תהיה חכמה יותר, ככה היא תדע להציג את המודעה המתאימה למשתמש הנכוון, בזמן הנכוון ועל **Device** המתאים.

1.1.2 Data, Tasks and Learning

כאמור=המטרה הבסיסית של למידת מכונה היא יכולת להכליל מתוך הניסיון, ולבצע משימות באופן מדי-קי-ככל הניתק על-دادטה חדשני לא נצפה, על בסיס צבירת ניסיון מDATA קי-ים=באופן כללי ניתן לדבר על שלושה סוגים של מידע:

למידה מונחי-(*supervised learning*)=הدادטה הקיים הינו אוסף של דוגמאות, ולכל דוגמא יש תווית (*label*). מטרת האלגוריתם ב厶בוקה זהה=^{א-ל-ס-ו-ו}דוגמאות נצפבותהילך הלמידה=באופן פורמלי', עברו דאטטה $x \in \mathbb{R}^{n \times d}$, יש אוסף=^oabels $y \in \mathbb{R}^{1 \times d}$, ומחפשים את האלגוריתם שמבצע את המיפוי $X \rightarrow Y$:^ובצורה הטובה ביותר,قولמר בהינתן דוגמא חדשה $x \in \mathbb{R}^n$, המטריה היא למצוא עבורה את \hat{y} ההנכוון. המיפוי נמדד ביחס לפונקציות מחיר, כפי שיסביר בהמשך בוגר לתהילך הלמידה.

למידה לא מונחי-(*unsupervised learning*)=הدادטה הקיים הינו אוסף של דוגמאות&במרחבי, בלי שנותן עליה רק מידע כלשהו המבחן בינהן=ב厶בוקה זה, בדרך כלל האלגוריתם מיפוי חפשו מודל המסביר את התפלגות הנקודות=למשל חלוקה לקבוצות שונות ובדומה

למידה באמצעות חיזוק-(*reinforcement learning*)=הدادטה בו נעזרים אינו מצובעת תחילה התוכנית אלא נאסף עם הזמן. ישנו סוכנים הנמצאים בסביבה מסוימת ומעבירים מידע למשתמש, והוא בתורו למד אסטרטגיה בה הסוכנים ינקטו בצדדים הטובים עבורה

האלגוריתמים השונים של הלמידה מתחלקים לשתי קבוצות=**מודלים דיסקרימנטיביים** המוצאים פلت על בסיס מידע נתון, אך לא יכולים ליצור מידע חדש בעצם, ומודלים גנרטיביים, שלא רק לומדים להכליל את הדעתה הנלמד גם עבור דוגמאות חדשות, אלא יכולים גם להבין את מה שהם רואו וליצור מידע חדש על בסיס הדוגמאות שנלמדו.

כאמור, בשביל לבנות מודל יש צורך בדעתה. מודל טוב הוא מודל שמציל להכליל מהדעתה הקיימם גם לדעתה חדשה. המודל למשה מנסה למצוא דפוסים בדעתה הקיימם, מהם הוא יכול להסיק מסקנות גם על דוגמאות חדשות. כדי לוודא שהמודל אכן מצליח להכליל גם על דוגמאות חדשות, בדרך כלל מחלקים את הדעתה הקיימם לשני חלקים=**aimon (training set) ו-kboczachmbach (test set)**=אם המודל מצליח למצאו דפוסים=Epsut האימוקמאפר לחת מdad להצלחת המודול=**=-kboczachdogmato עליה** ncoanim גם לדוגמאות חדשות שיבואו=**=-kboczachdogmato עליה** המודול מתאמן, וקבוצת ולידציה (validation set) המשמשת להימנע מverfitting שיפור בהמשך

מגון התחומיים בהם משתמשים בכלים של למידה הוא עצום, עד כדי כך שכמעט אין תחום בו לא נכנס השימוש באלגוריתמים לומדים. דוגמאות בולטות למשימות בהם משתמשים באלגוריתמים לומדים: סיווג, רגסיה (מציאת קשר בין משתנים), חלוקה לקבוצות, מערכת המלצות, הורדרמן, ראייה ממוחשבת, עיבוד שפה טבעית ועוד.

1.2 Applied Math

האלגוריתמי=**shelmidotmekoneh-nesmech-beikarpe-uleshod-unpiyeh-matmati**=אלגברה לינארית, חישוב דיפרנציאלי והסתברות. פרק זה נציג את העקרונות הנדרשים בלבד, ללא הרחבת, על מנת להבין את הנושאים הנדרשים בספר זה.

1.2.1 Linear Algebra

וקטורים ומרחבים וקטוריים

באופן מתמטי מופשט, וקטורים, המנסנים בדרך כלל ע"פ או על ידך, הינם אובייקטים הנמצאים במרחב וקטורי $(+, \cdot)$ מעל שדה \mathbb{F} . מהו אותו מרחב וקטורי?

ראשית השדה \mathbb{F} , הוא קבוצת מספרים המקיימים תכונות מתמטיות מסוימות. לדין בספר זה, השדה הוא קבוצה המספרים המשי=**=R**, או קבוצת המספרים המרוכבים=**C**. שנית, נשים לב כי המרחב הוקטורי דרוש גם הגדרת פעולה חיבור $(+)$.

כעת, $(+, \cdot)$ היא מרחב וקטורי אם הוא מקיים את התכונות הבאות:

- (I) קיימ איבר אפס (וקטור אפס) כך שכל \vec{x} בקבוצה V מקיים: $\vec{x} = \vec{x} + \vec{0} = \vec{x}$.
- (II) לכל איבר בשדה a ולכל \vec{x} ו- \vec{y} בקבוצה V , גם $\vec{y} + \vec{x} \cdot a$ הינו איבר בקבוצה V .

הערה: קיומותדרישות נוספות למרחב וקטורי, אך הם מעבר לנדרש בספר זה.

דוגמאות:

A. וקטורים גאומטריים=**M**
מערך $\vec{x} = (x_1, x_2, \dots, x_n)$ (x_i סדרה) נקרא וקטור גאומטרי=**vector**, כאשר רכיביו ווקטוריים=**vectors** איברים בשדה \mathbb{F} . האיבר \vec{x} , המוצג על ידי האינדקס? מתאר את מיקום האיבר. מרחב זה מסומן ע"י \mathbb{F}^n .
נראה שמרחב זה הוא אכן מרחב וקטורי

חיבור וקטוריים:

$$\vec{x} = (x_1, x_2, \dots, x_n), \quad \vec{y} = (y_1, y_2, \dots, y_n) \rightarrow \vec{x} + \vec{y} = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n)$$

וקטור אפס:

$$\vec{0} = (0, 0, \dots, 0)$$

כפל בסקלר:

$$\vec{x} = (x_1, x_2, \dots, x_n) \rightarrow a \vec{x} = (a x_1, a x_2, \dots, a x_n)$$

הערה: לשם פשוטות, בהמשך, נenna וקטור גאומטרי כ"וקטור" בלבד.

B. מטריצות:

מערך דו ממד $\mathbb{F}^{n \times m}$, אשר רכיביו הם איברים בשדה \mathbb{F} , נקרא מטריצה מסדר $m \times n$, כאשר A_{ij} הוא מספר השורות ו- A_{ji} הוא מספר העמודות במערך. האיברים במטריצה A_{ij} מיוצגים ע"י שני אינדקסים i, j המתארים את השורה והעמודה בהתאם. מרכיב זה מסומן בדרך כלל ע"פ $\mathbb{F}^{m \times n}$.

ונכון שמרחב זה הוא אכן מרחב וקטורי:

חיבור מטריצות:

$$\hat{A} = \begin{pmatrix} A_{11} & \dots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nm} \end{pmatrix}, \hat{B} = \begin{pmatrix} B_{11} & \dots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{n1} & \dots & B_{nm} \end{pmatrix} \rightarrow \hat{A} + \hat{B} = \begin{pmatrix} A_{11} + B_{11} & \dots & A_{1m} + B_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} + B_{n1} & \dots & A_{nm} + B_{nm} \end{pmatrix}$$

מטריצת אפס:

$$\hat{0} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

כפל בסקלר:

$$\hat{A} = \begin{pmatrix} A_{11} & \dots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nm} \end{pmatrix} \rightarrow a \hat{A} = \begin{pmatrix} a A_{11} & \dots & a A_{1m} \\ \vdots & \ddots & \vdots \\ a A_{n1} & \dots & a A_{nm} \end{pmatrix}$$

ניתן לבדוק כי הווקטורים היגיאומטריים שהוגדרו בדוגמה א', הם בעצם מטריצות $m \times 1$.

ג. פולינומים:

פולינומים מסדר- i הם ביטויים מהסוג $a_n x^n + \dots + a_2 x^2 + a_1 x + a_0$, כאשר- i מייצג את החזקה הגדולה ביותר \hat{i} הם איברים בשדה. מרכיב זה מסומן בדרך כלל ע"פ $\mathbb{P}_n(x)$.

בכל הדוגמאות לעיל קל לראות שהן אכן מהוות מרחב וקטורי. רשימה חלקית לדוגמאות נוספת לווקטורים (ולמרחבים וקטוריים) כוללת למשל מרחבי פונקציות או אפילו אוטות אלקטرومגנטיים. כאן בחרנו רק את הדוגמאות הרלוונטיות לשפר זה.

פעולות חשבון על מטריצות וקטורים:

כמו שנצכר לעיל, הווקטורים היגיאומטריים שהוגדרו בדוגמה א', הם בעצם מטריצות $m \times 1$. הפעולות החשבון מוגדרות באופן זהה:

• חיבור וחיסור בין שתי מטריצות:

$\mathbb{F}^{m \times n} \in \hat{A}, \mathbb{F}^{n \times k} \in \hat{B}$ כאשר A_{ij}, B_{ij} הם האיברים בשורה i בעמודה j של המטריצות \hat{A}, \hat{B} בהתאם. אז, האיבר בשורה i בעמודה j של מטריצת הסכום (או ההפרש) הינו

$$(A \pm B)_{ij} = A_{ij} \pm B_{ij}$$

(הגדרת חיבור המטריצות בעצם כבר ניתנה בדוגמה א' לעיל).

שים לב: ניתן לחסר ולחסור מטריצות רק בעלות אותו הממד.

• כפל בין שתי מטריצות:

$\mathbb{F}^{k \times m} \in \hat{A}, \mathbb{F}^{m \times n} \in \hat{B}$ הן שתי מטריצות, כאשר מספּה העמודות במטריצה \hat{A} שווה למספר השורות של מטריצה \hat{B} (אך שתי המטריצות אינן בהכרח בעלות אותן ממד). במקרה זה, מכפלת המטריצות מוגדרת על ידי:

$$\hat{A} \cdot \hat{B} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} + \dots + A_{1k}B_{k1} & \dots & A_{11}B_{1n} + \dots + A_{1k}B_{kn} \\ \vdots & \ddots & \vdots \\ A_{m1}B_{11} + \dots + A_{mk}B_{k1} & \dots & A_{m1}B_{1n} + \dots + A_{mk}B_{kn} \end{pmatrix}$$

למעשה כל איבר בתוצאה הינו סכום של מכפלת שורה i ממטריצה A בעמודה j ממטריצה B :

$$(\hat{A} \cdot \hat{B})_{ij} = \sum_r A_{ir} B_{rj}$$

שים לב: על מנת שכפל המטריצות יהיה מוגדר מספר העמודות ב- \hat{A} שווה למספר השורות ב- \hat{B} .
עבור מטריצות ריבועיות (מסדר $n \times n$), מוגדר גם הכפל $\hat{B}\hat{A}$ וגם $\hat{A}\hat{B} \neq \hat{B}\hat{A}$, אולם יתכן $\hat{A}\hat{B} = \hat{B}\hat{A}$.

- **שחלוף (transpose):**

החלפת שורות בעמודות, או 'סיבוב' המטריצה. נניח מטריצה $\mathbb{F}^{n \times m} \in \hat{A}$, אז השחלוף שלה, המסומן \hat{A}^T הוא:

$$(\hat{A}^T)_{ij} = A_{ji}$$

ובאופן מפורש:

$$\hat{A} = \begin{pmatrix} A_{11} & \dots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nm} \end{pmatrix} \rightarrow \hat{A}^T = \begin{pmatrix} A_{11} & \dots & A_{n1} \\ \vdots & \ddots & \vdots \\ A_{1m} & \dots & A_{nm} \end{pmatrix}$$

שים לב שהמטריצה החדשה \hat{A}^T היאQM $m \times n$. בנוסף ניתן להוכיח כי מתקיים:
שחלוף של וקטור שורה, נותן וקטור עמודה ולהפך.

- **מטריצה יחידה:**

מטריצה יחידה, הינה מטריצה ריבועית (מסדר $n \times n$), המסומנת על ידי \mathbb{I}_n ומוגדרת כך שכל איבריה אפס מלבד איברי האלכסון הראשי המקבלים את הערך 1:

$$(\mathbb{I}_n)_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

ובאופן מפורש:

$$\mathbb{I}_n = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix}$$

מטריצה זו מקיימת $\hat{A} = \hat{A} \cdot \mathbb{I}_n = \mathbb{I}_n \cdot \hat{A}$ לכל מטריצה \hat{A} מסדר $n \times n$
הערה: לעיתים סדר מטריצת היחידה אינו משנה או טריויאלי, ולכן המטריצה מסומנת רק על ידה ולא ציון המממש.

- **מטריצה הופכית:**

למטריצות ריבועיות (מטריצות עם מספר זהה של שורות ועמודות; מסדר $n \times n$) יתאפשר מטריצה הופכיה \hat{A}^{-1} שמקיימת את הקש:

$$\hat{A} \cdot \hat{A}^{-1} = \hat{A}^{-1} \cdot \hat{A} = \mathbb{I}_n$$

$$\hat{A}^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \text{ לכן, במקרה זה } \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{I}_2.$$

- **מטריצה צמודה/hermitian:**

עבור מטריצה A , המטריצה $A^\dagger = A^*$ נקראת הצמוד הרמייטי של A , ומתקיים:

$$(A^*)_{ij} = \overline{A_{ji}}$$

הצמוד הרמייטי הוא שחלוף של A , כאשר לכל איבר במטריצה המשוחלפת לוקחים את הצמוד המרוכב.
אם A מטריצה ממשית, המטריצה הצמודה שלה היא למעשה המטריצה המשוחלפת של A .

• מטריצה אוניטרית:

מטריצה אוניטרית היא מטריצה ריבועית מעל המספרים המרוכבים המקיימת את התנאי:

$$A^* A = AA^* = \mathbb{I}$$

מערכת משוואות לינאריות:

מערכת משוואות לינאריות מוצגת באופן כללי באופן הבא:

$$\begin{array}{ccccccccc} A_{11}x_1 + A_{12}x_2 + & \dots & + A_{1n}x_n & = & b_1 \\ \vdots & \ddots & \vdots & & \vdots \\ A_{m1}x_1 + A_{m2}x_2 + & \dots & + A_{mn}x_n & = & b_m \end{array}$$

נשים לב כי מערכת משוואות לינארית ניתנת לייצוג באופן קומפקטי על ידי הפרדה בין רשימת המשתנים, המקדמים של משתנה, והאיבר החופשי, באופן הבא:

$$\hat{A} \vec{x} = \vec{b} = \begin{pmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

מטריצה \hat{A} הינה מטריצת המקדים מסדר $m \times n$, כאשר זה הוא מספר המשתנים, ואלהו אż' מספר המשוואות במערכת=

$$\text{קטורי } \vec{x}, \text{ הינו וקטור عمودה (לעתים גם מסומן עלי } \vec{x}^T \text{), המיצג את וקטור המשתנים.}$$

$$\text{קטורי } \vec{b}, \text{ הינו וקטור عمודה, שאיםיו הם האיבר החופשי.}$$

פתרונות של מערכת המשוואות הלינארית, $\vec{b} = \hat{A} \vec{x}$, אם הם קיימים ויחדים, נתונים עלי $\vec{b} = \hat{A}^{-1} \vec{b}$.

מכפלה פנימית, נורמה, אורטוגונליות

מרחב מכפלה פנימית, מוגדר על יד מרחב וקטורי \mathbb{F} (המוגדר על גב-שדה \mathbb{F}) ועל צדי פעולה "מכפלה פנימית"=
מכפלה פנימית-הנקראאל-עיטים רק מכפלה, הינה בעצם פונקציה המתקבלת שני וקטורים מרחב וקטורי \mathbb{F} ומחזירה סקלר (=מספר) בשדה \mathbb{F} . מכפלה זו, מסומנת בדרך כלל עלי $\langle \cdot, \cdot \rangle$ (או עלי $\mathbb{F} \rightarrow V \times V$: $\langle \cdot, \cdot \rangle$), חיבת ליקים מספה תכונת:

לכל $v \in V$ ו- \vec{w}, \vec{v} (כל שלושה וקטורים למרחב הווקטורי V), ולכל $\lambda \in \mathbb{F}$ (סקלר בשדה \mathbb{F}):

- $\langle \vec{v} + \vec{w}, \vec{u} \rangle = \langle \vec{v}, \vec{u} \rangle + \langle \vec{w}, \vec{u} \rangle$ •
- $\langle \lambda \vec{v}, \vec{u} \rangle = \lambda \langle \vec{v}, \vec{u} \rangle$ •
- $\overline{\langle \vec{v}, \vec{u} \rangle} = \langle \vec{u}, \vec{v} \rangle$ •
- $\langle \vec{v}, \vec{v} \rangle \geq 0$ •

ההגדרה עצמה של המכפלה משתנה כתלות במרחב הווקטורי הנutan. לדוגמה:

א. מכפלה סקלרית על מרחב הווקטורים הגיאומטריים:

נתונים $\vec{v}, \vec{w} \in \mathbb{C}^n$, ו- וקטורים גיאומטריים מסדר n מעל שדה המספרים המרוכבים. מכפלה פנימית בין שני וקטורי אלה, נקראת גם מכפלה סקלרית, המוגדרת על יד:

$$\langle \vec{v}, \vec{w} \rangle = \vec{v}^T \cdot \vec{w} = \sum_{i=1}^n v_i w_i$$

כאשר \vec{u} הינו הצמוד המרוכב של u .

ב. מרחב הילברט – מרחב מכפלה פנימית על מרחב הפונקציות:

נניח ש- \mathbb{C}^n הפונקציות מרוכבות $\rightarrow \mathbb{C}$: \vec{f} אינטגרביליות בתחום כלשהו (כמו שהוזכר לעיל, גם מרחב הפונקציות הוא מרחב וקטורי), אז המכפלה פנימית מוגדרת על ידי:

$$\langle f(x), g(x) \rangle = \int_I f^*(x)g(x)dx$$

כאשר f^* הינו הצמוד המרוכב של f .

ניתן להגדיר גם מרחבי מכפלה פנימית נוספים, נניח עבור מרחב המטריצות – נורמה:

נורמה, מוגדרת על ידי מכפלה פנימית של וקטור בעצמו, ומסומנת $\| \cdot \|$, זאת אומרת:

$$\| \vec{u} \| = \sqrt{\langle \vec{u}, \vec{u} \rangle} \geq 0$$

שוויון מתקיים אך ורק עבור וקטור האפס; $0 = \vec{u} \Leftrightarrow \| \vec{u} \| = 0$.

תמונה נוספת, נקראת א-בישוקה המשולש, מוגדרת על ידי:

$$\| \vec{u} + \vec{v} \| \leq \| \vec{u} \| + \| \vec{v} \|$$

או שוויון נוסף הקשור לנורמות נקרא או שוויון קושי שוורץ (Cauchy-Schwarz inequality):

$$\langle x, y \rangle \leq \|x\| \cdot \|y\|$$

כאשר $\langle x, y \rangle$ הינה המכפלה הפנימית בין שני הווקטורים, המוגדרת מעל הטבעיים $x = \sum_i x_i e_i$, $y = \sum_i y_i e_i$, והbij'וי $\|x\| \cdot \|y\|$ הוא מכפלת הנורומות.

דוגמה:

א. במרחב \mathbb{R}^3 הינו המכפלה הפנימית בין שני הווקטורים, הגדרת הנורמה היא בעצם הגדרת אורך (או גודל וקטור). נניח עבור הווקטורים $x, y \in \mathbb{R}^3$ מוגדרת $\|x\| = \sqrt{x^2 + y^2 + z^2}$.

ב. במרחב הילברט נורמה של פונקציה $\mathbb{C} \rightarrow \mathbb{C}$: f הינה $\|f\| = \int_I |f(x)|^2 dx$.

אורתוגונליות

הגדרת מכפלה פנימית מאפשרת לנו להגדיר אורתוגונליות (או אנכיות) של שני וקטורים במרחב מכפלה פנימית מסוים. שני וקטורי $\vec{u}, \vec{v} \in \mathbb{V}$ נקראים אורתוגונליים זה לזה אם ורק אם המכפלה הפנימית שלהם הינה אפס:

$$\langle \vec{u}, \vec{v} \rangle = 0 \Leftrightarrow \vec{u} \perp \vec{v}$$

כאשר מתייחסים למרחב הווקטורים הגיאומטריים, קל להבין את מושג האורתוגונליות.

אורתוגונליות היא הכללה של תכונת הניצבות המוכרת מגאומטריה. בגאומטריה, שני ישרים במשור האוקלידי ניצבים זה זה אם הזרויות הנוצרת בנקודת החיתוך שלהם היא זווית ישרה (90°). מושג האורתוגונליות מכליל תכונה זו גם למרחב וקטורי-גיאומטרי. על מנת להכליל את מושג הניצבות ישראשית להגדיר זווית בין שני וקטורים:

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\| \cdot \|y\|}$$

לפ"א שוויון קושי שוורץ מתקיים $\|x\| \cdot \|y\| \leq \langle x, y \rangle$ ובקהיבתיו באגד' ימ' תמיד קטן או שווה בערכו המוחלט $\sqrt{\|x\|^2 + \|y\|^2}$.

1. כיוון שכך, תמיד ניתן לחשב זווית בין שני וקטורי-בазרת מכפלה פנימית.

לוקטוריים אורתוגונליים חשובות רבה כאשר חוקרים מרחב וקטורי יש מספר תכונות נוחות כאשר הוא אורתוגונל (כל אבריו אורתוגונליים זה לזה ובויל אורפ=1). יתר על כן, מתרבר שבהינתן בסיס כלשהו למרחב וקטורי ניתן לקבל ממנו בסיס חדש שכל אבריו אורתוגונליים זה לזה, כך שתמיד ניתן למצוא בסיס נורמי לכך. דבר זה נעשה על ידי תהליך גראם-שmidt (gram-schmidt).

שני וקטוריים אורתוגונליים יסומנים על ידי \mathbf{v} . עבור וקטוריים אורתוגונליים מתקיימות התכונות הבאות:

- $\mathbf{v} \perp \mathbf{u}$, אז $\mathbf{u}^\top \mathbf{v} = 0$.
- $\mathbf{v} \perp \mathbf{u}$, אז לכל סקלר c גם $\mathbf{u}^\top c\mathbf{v} = 0$.
- $\mathbf{v} \perp \mathbf{u}$ ו- $\mathbf{v} \perp \mathbf{w}$, אז $\mathbf{u}^\top (\mathbf{v} + \mathbf{w}) = 0$.
- אם וקטור אורתוגונלי לקבוצה של וקטוריים אז הוא גם אורתוגונלי לכל צירוף ליניארי שלהם (נובע משפט התכונות הקודמות).

וקטוריים עצמיים וערכים עצמיים

תהי $\mathbb{F}^{n \times n}$ מטריצה ריבועית, וקטור $\mathbf{v} \in \mathbb{F}^n$ ב- \mathbb{F} הוא וקטור העצמי של A אם $A\mathbf{v} = \lambda\mathbf{v}$.

$$A \cdot \mathbf{v} = \lambda \cdot \mathbf{v}$$

ניתן להראות שעבור מטריצה A הווקטוריים העצמיים המתאימים לסקלר λ הם כל פתרונות המשוואה ההומוגנית $(A - \lambda I_n) \mathbf{v} = 0$.

אם נסמן $\mathbf{v}_1, \dots, \mathbf{v}_n$ אזי מתקיים:

$$A = V \operatorname{diag}(\Lambda) V^{-1}$$

כאשר Λ הוא ערכי האלכסון של המטריצה A .

פירוק לערכים סינגולריים

ניתן לפרק מטריצה $\mathbb{R}^{m \times n} \in M$ למכפלה של שלוש מטריצות באופן הבא:

$$M = U \Sigma V^*$$

כאשר $\mathbb{R}^{m \times m} \in U$ היא מטריצה אוניטרית מרוכבת (או ממשית), $\mathbb{R}^{m \times n} \in \Sigma$ היא מטריצה אלכסונית-שליליים, $\mathbb{C}^{n \times n} \in V$ היא מטריצה אוניטרית-מרוכבת (או ממשית). פירוק זה נקרא פירוק לערכים סינגולריים (Singular value decomposition - SVD).

ערך האלכסון של M מסודרים מגדול לקטן, והערכים הסינגולריים של M בנוסף, זה העמודות של U נקראות הווקטוריים הסינגולריים השמאליים של M , ובהתאם להעמודות של V הוקטוריהם הסינגולריים הימניים של M . שלוש המטריצות מקיימות את התכונות הבאות:

- הווקטוריים הסינגולריים השמאליים של M הם וקטוריים עצמיים של $M^* M$.
- הווקטוריים הסינגולריים הימניים של M הם וקטוריים עצמיים של $M M^*$.
- הערכים הסינגולריים (איבריה האלכסון של Σ) שאינטש אפסיהם שורשי בירובעים של הערכים עצמיים השונים מאפס של $M^* M$ ושל $M M^*$.

לפירוק SVD יש שימושים בתחוםים רבים, ואף ניתן להציג בעזרתו נורמות חדשות.

1.2.2 Calculus

פונקציות

פונקציה הינה התאמה (או העתקה), המתאימה לכל איבר x (בתחום מסוים), ערך ייחודי, ומסמנת באופן הבא: $f(x) = y$. קבוצת האיברים, נקראת תחום, וקבוצת הערך-ים נקראת ערך. קבוצות התחום והטווח יכולות להיות רציפות (למשל מספרים ממשיים חיוביים) או בדידות (למשל קבוצה $\{0, 1\}$). בדרך כלל ה計算 מופיע כר $\rightarrow Y \rightarrow X$, כאשר X ו- Y הינם התחום והטווח בהתאם להתאמה.

דוגמאות $\mathbb{R}^+ \rightarrow \mathbb{R}^2$: הינה פונקציה, הולוקחת וקטורים גיאומטריים דו-ממדיים, ומחזירה מספר ממשי א-שלילי. הפונקציה עצמה \cdot היא הנורמה של הווקטור, כפי שהוגדרה בפרק הקודם.

נגזרת

עבור פונקציות ממשיות, נגזרת מוגדרת על ידי מידת השתנות של הפונקציה $(x) \mapsto$ יחסינו קטן (אינפיניטסימלי) Δx . באופן גיאומטרי, הנגזרת הינה השיפוע של הפונקציה בנקודה x . נגזרת מסומנת בדרך כלל על ידי $f'(x) = \frac{df}{dx}(x)$.

נגזרות של פונקציות אלמנטריות ניתן לחשב באמצעות כללים ידועים. לדוגמה:

- לכל $n \neq 0$ מתקיים: $\frac{d(x^n)}{dx} = nx^{n-1}$
- חיבור או חיסור פונקציות: $\frac{d(f(x)+g(x))}{dx} = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$
- מכפלת שתי פונקציות: $\frac{d(f(x) \cdot g(x))}{dx} = \frac{f(x)dg(x)}{dx} + \frac{g(x)df(x)}{dx}$
- כלל שרשרת: $\frac{df(g(x))}{dx} = \frac{df}{dg} \frac{dg}{dx}$

כיוון שנגזרת של פונקציה ממשית מכמתת את קצב שינוי הפונקציה, אז בתחום שבו הפונקציה יורדת הנגזרת שפה תהיה שלילית, ובתחום שבו היא עולה הנגזרת תהיה חיובית. ככל שקצב ההשתנות גדול יותר כך ערכה המוחלט של הנגזרת גדול.

הערה: לא לכל פונקציה מוגדרת נגזרת. למספר זהணיה שהפונקציה אנליטית ולכן גדרה

הערת-נוספת=כיוון שנגזרת של פונקציה היא גם פונקציה, ניתן גם להגדיר נגזרת שנייה או נגזרת מסדרים גבוהים יותר. בדרך כלל הסימון הינו $f''(x) = \frac{d^2f}{dx^2}(x)$ לנגזרת מסדר שני וכוכב

נקודות אקסטרום

נקודות אקסטרום של פונקציה, הן נקודות שבהם הפונקציה מקבלת ערך מקסימום או מינימום באופן מקומי. במקרה אחד, הנגזרת של הפונקציה "משנה ציוויל" (מפונקציה עולה לפונקציה יורדת או להפך) ולכן מקבלת את הערך אפס=יש לשים לבשחתאפסות הנגזרת בנקודות המינימום והמקסימום היא תנאי הכרח=אך לא מספיק. יתכן שהנגזרת מתאפסת בנקודה מסוימת, אך נקודה זו אינה מינימום או מקסימום מקומי, אלא נקודת פיתול.

לדוגמה: $x^3 = f(x)$. נגזרת הפונקציה הינה $3x^2 = f'(x)$ והוא מתאפסת בנקודה $0 =$

גרדיאנט, יעקוביאן והסיאם

עבור פונקציה מרובת משתנים=נגזרת חלקית להיות הנגזרת של הפונקציה לפי אחד המשתנים שלה, והיא מסומנת=ב- $\frac{\partial f(x_1, \dots, x_n)}{\partial x_i}$. כאשר גוזרים לפי משתנה מסוים, שאר המשתנים הם קבועים ביחס לנגזרת.=בהינתן הפונקציה $f(x_1, \dots, x_n)$, וקטור הנגזרות לפי כל המשתנים נקרא גרדיאנט:

$$\nabla f(x_1, \dots, x_n) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \leftrightarrow [\nabla f]_i = \frac{\partial f}{\partial x_i}$$

עבור פונקציות התלוויות $\in \mathbb{R}^m$ המשתנים, הייעקוביאן הוא מטריצת הנגזרות החלקיים:

$$\mathcal{J}_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{n \times m} \leftrightarrow [\mathcal{J}_f]_{ij} = \frac{\partial f_i}{\partial x_j}$$

עבור פונקציה $f(x_1, \dots, x_n)$, מטריצת הנגזרות מסדר שני נקראת הסיאם

$$\mathcal{H}_f = \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}_{n \times n} \leftrightarrow [\mathcal{H}_f]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

שני כללים חשובים בחישוב נגזרות של מטריצות:

$$\nabla_x(a^T x) = a$$

$$\nabla_x(x^T A x) = (A + A^T)x$$

1.2.3 Probability

תורת ההסתברות היא תחום המספק כל-=נition למאורעות=המכליל=פ=ממד=של אקראיות ואינטגרליות. הסתברות של מאורע הוא ערך=מספר למידת הסבירות שהוא יתרחש, כאשר ערך זה נעה בין 0 ל-1 – מאורע בלתי אפשרי הוא בעל הסתברות 0, ומאורע ודאי הוא בעל הסתברות 1.

הגדרות בסיסיות

Ω = מרחב המדגם – מכלול האפשרויות השונות של ניסוי. לדוגמה עבור הטלת קובייה: $\{\Omega = \{1,2,3,4,5,6\}\}$.

קבוצה – חלק ממכלול המדגם. לדוגמה עבור הטלת קובייה: $A = \{2, 4, 6\}$ = even number

מאורע – תוצאה אפשרית של ניסוי

הסתברות= $=\text{-סיכוי}$ של מאורע להתרחש. עבור תת קבוצה A של מרחב המדגם Ω , ההסתברות לקיום מאורע מקבוצת A שווה לחלק היחסי של מספר איברי הקבוצה מתוך קבוצת המדגם:

$$p(A) = \frac{\#A}{\#\Omega}, 0 \leq p(A) \leq 1$$

$A \cap B$ = איחוד של שתי קבוצות הוא אוסף האברים של שתי הקבוצות=Aיחוד של הקבוצה A ו- B והוא אוסף האברים המופיעים לפחות באחת משתי הקבוצות A או B . לדוגמה עבור הטלת קובייה:

$$A = \text{even number} = \{2, 4, 6\}, B = \text{lower than } 4 = \{1, 2, 3\}$$

$$\rightarrow A \cup B = \{1, 2, 3, 4, 6\}, \quad p(A \cup B) = \frac{5}{6}$$

$A \cap B$ = חיתוך=חיתוך של שתי קבוצות הוא אוסף האברים המופיעים בשתי הקבוצות. חיתוך של הקבוצות A ו- B הוא אוסף האברים המופיעים גם ב- A וגם ב- B . עבור דוגמא הקודמת

$$A \cap B = \{2\}, p(A \cap B) = \frac{1}{6}$$

מאורעות זרים – מאורעות שהחיתוך שלהם ריק, כלומר אין להם אברים משותפים:

$$A \cap B = \emptyset, p(A \cap B) = 0$$

מאורע משלים – מאורע המכיל את כל האברים שאינם נמצאים בקבוצה מסוימת

$$A \cup A^c = \Omega \rightarrow p(A \cup A^c) = 1, p(A) = 1 - p(A^c)$$

מאורעות בלתי תלויים= $=P(A \cdot P(B) = P(A \cap B)$. באופן אינטואיטיבי ניתן לחשב על כך שבמקרה זה-=האחד אינו משנה על הסיכוי של השני.

אם המאורעות זרים (והם בעלי סיכוי שונה מ-0), הם בהכרח תלויים

$$P(A \cap B) = 0 \neq P(A) \cdot P(B) > 0$$

$p(A|B)$ = הסתברות מותנית – בהינתן מידע מסוים, מה ההסתברות של מאורע כלשהו?

$$p(A|B) = \frac{p(A \cup B)}{p(B)} \leftrightarrow p(A|B) \cdot p(B) = p(A \cup B) = p(B|A) \cdot p(A)$$

בעזרת ההגדרה של הסתברות מותנית ניתן לתת הגדרה נוספת למאורעות בלתי תלויים:

$$A, B \text{ תלויים} \leftrightarrow p(A|B) = p(A)$$

נשים לב שהמשמעות של שתי ההגדרות זהה – המידע על B לא משנה את חישוב ההסתברות של A .

נוסחת ההסתברות השלמה וחוק ביחס

נוסחת ההסתברות השלמה היא נוסחה פשוטה המאפשרת לחשב מאורעות מסוימים נחיתון לפרקי מרחב הסתברות לאיברי פערם וacz' לחשב את ההסתברות של כל איבר בפניהם עצמו. אם ניקח את כל ההסתברויות המתקבלות, וככפי' כל אחת מהן במשקל של אותו איבר, נקבל את נוסחת ההסתברות השלמה:

$$P(B) = \sum_i P(B|A_i) \cdot P(A_i)$$

מתוך נוסחה זו ניתן בקלה לחקוק ביחס, המאפשרת לחשב הסתברות מותנית באמצעות ההתנית ההפוכה

$$p(A|B) = \frac{p(A \cap B)}{p(B)} = \frac{p(B|A)p(A)}{p(B)}$$

משפט הכללה והדחאה

כדי לסייע עצמים בקבוצה, אפשר לכלול ולהוציא את אותו עצם שוב ושוב, כל עוד בסוף ההליך נספר כל עצם לפחות אחת. עקרון פשוט זה מתרגם לנוסחה הבאה

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n-1} |A_1 \cap \dots \cap A_j|$$

עבור 2 קבוצות הנוסחה נהיה יותר פשוטה

$$|A \cup B| = |A| + |B| - |A \cap B|$$

במקרה זה, כאשר A, B זרות, אז $|A \cup B| = |A| + |B|$

עבור שלוש קבוצות מתקובלת הנוסחה:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$$

משתנים אקראיים

Ω : משתנה מקרי – פונקציה המתאימה לכל מאורע השיר למרחב ההסתברות ערך מסווני, המהווה את הסיכוי של המאורע להתרחש.

פונקציית ההסתברות של משתנה מקרי X נותנת את הסיכוי של כ'א אפשרי

$$f_X: \Omega \rightarrow [0,1] = p(X=x)$$

פונקציה זו מקיימת שלוש אקסiomות:

- הסתברות של כל מאורע למרחב המדגם גדולה או שווה ל-0.
- סכום ההסתברויות של כל המאורעות למרחב שווה ל-1: $\sum p(X=x) = 1$
- סכום ההסתברויות של שני מאורעות זרים שווה להסתברות של איחוד המאורעות

עבור משתנה מקרי רציף יש אינסוף מאורעות אפשריים, לכן ההסתברות של כל מאורע יחיד היא 0. ל痼בור משתנה מקרי רציף מקרים את פונקציית ההסתברות לפונקציה הנקראת פונקציית ההתפלגות (או פונקציית הצפיפות המצטברת), המחשבה את ההסתברות שמאורע יהיה קטן מערך מסוים.

$$F_X(a) = p(X \leq a) = \int_{-\infty}^a f_X(x)dx$$

ניתן לחשב בעזרת פונקציה זו את ההסתברות שמאורע \mathcal{A} בטווח מסוים:

$$p(a \leq X \leq b) = F_X(b) - F_X(a) = \int_a^b f_X(x)dx$$

פונקציית ההתפלגות מקיימת את התכונות הבאות

- $\lim_{a \rightarrow -\infty} F_X(a) = 0$
- $\lim_{a \rightarrow \infty} F_X(a) = 1$
- $\int_{-\infty}^{\infty} f_X(x)dx = 1$
- הפונקציה מונוטונית עולה במובן החלש: $a \leq b \Rightarrow F_X(a) \leq F_X(b)$
- $p(X \geq a) = 1 - F_X(a)$

תכונות ופרמטרים עבור משתנה מקרי

תוחלת – ממוצע משוקל של כל הערכים האפשריים, כל אחד מוכפל בהסתברות שלו

$$\mathbb{E}[X] = \sum_i x_i P(X = x_i) = \int_{-\infty}^{\infty} xf(x)dx$$

תכונות

- $\mathbb{E}[c] = c$
- $\mathbb{E}[\mathbb{E}[X]] = \mathbb{E}[X]$
- $\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$, $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

שונות – ממד פיזור הערכים ביחס לממוצע המשוקל (-התוחלת)

$$Var[x] = E[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \int_{-\infty}^{\infty} x^2 f(x)dx - (\mathbb{E}[X])^2$$

סטיית תקן מוגדרת להיות שורש השונות

תכונות

- אי שליליות: $Var[x] \geq 0$
- $Var[aX + b] = a^2 Var[X]$

שונות משותפת – ממד ליחס אפשרי בין שני משתנים מקרים

$$cov(X, Y) = \mathbb{E}[X \cdot Y] - \mathbb{E}[X] \cdot \mathbb{E}[Y]$$

כאשר: $\mathbb{E}[X \cdot Y] = \sum_j \sum_i x_i y_j P(X = x_i \cap Y = y_j) = 0$.

מקדם המתאים – נרמול של השונות המשותפת: $\rho(X, Y) = \frac{cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$. המקדם מקיים $-1 \leq \rho \leq 1$

שני משתנים מקרים מוגדרים בלתי מתואמים בלתי תלויים אפוא $cov(X, Y) = 0$. אם המשתנים בלתי תלויים אז הם בהכרח בלתי מתואמים.

בازURRENT השונות המשותפת ניתן לכתוב: $Var[X + Y] = Var[X] + Var[Y] + 2 \cdot cov(X, Y)$

פונקציה יוצרת מומנטים (התמרת לפילס של פונקציית הצפיפות):

$$M_X(t) = \mathbb{E}[e^{tX}] = \begin{cases} \sum_{i=0}^n e^{t \cdot x_i} p_X(x_i) \\ \int_S e^{t \cdot x} f_X(x) dx \end{cases}$$

בעזרת פונקציה זו ניתן ליצור מומנטים, שימושיים ללמידה על המשתנים:

$$\frac{\partial^n M_X(t)}{\partial t^n} \Big|_{t=0} = \mathbb{E}[X^n]$$

המומנט הראשון הוא התוחלת והמומנט השני הוא השונות

התפלגות מיוחדות (בדיוק)

ישן כל מיני התפלגות מיוחדות, שופיעות בטבע בכל מיני מקרים ויש להן נוסחאות ידועות.

התפלגות ברנולי: $X \sim Ber(p)$

ניסוי בעל שתי תוצאות אפשריות "הצלחה" או "כישלון". המשתנה המקרי מקבל שני ערכים בלבד – 1, בהתחשב להצלחה וכישלון

$$P(X = k) = \begin{cases} 1, & k = 1 \\ 0, & k = 0 \end{cases}, \mathbb{E}[X] = p, V[X] = pq = p(1-p)$$

התפלגות בינומית: $(p, n) \sim B$

בהתפלגות בינומית חוזרים על אותו ניסוי ברנולי n פעמים באופן בלתי תלוי זה בזה. מגדירים k להיוות מספר ההצלחות שהתקבלו בסה"כ. נסמן p סיכוי להצלחה בניסוי בודד וב- q סיכוי לכישלון בניסוי בודד.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \mathbb{E}[X] = np, Var[X] = npq$$

צריך לזכור (1) חוזרים על אותו ניסוי באופן בלתי תלוי (2) חוזרים על הניסוי n פעמים (3) מוגדר כמספר ההצלחות המתקבלות בסה"כ

התפלגות גיאומטרית: $(p) \sim G$

חווזרים על ניסוי ברנולי. כאשר מבטאת את מספר הניסויים שבוצעו עד ההצלחה הראשונה קמסמן את הסתברות ההצלחה בניסוי בודד

$$P(X = k) = pq^{k-1}, \mathbb{E}[X] = \frac{1}{p}, Var[X] = \frac{q}{p^2}$$

להתפלגות זו יש שתי תכונות נוספות:

1) "תכנת חוסר זיכרון": $P[X = (n+k)|X > k] = P(n|X > k)$.

2) ההסתברות שיעברו k ניסויים ללא הצלחה: $P(X > k) = q^k$.

כמו כן, אcumulative distribution function (CDF) הנדרש עד להצלחה ראשונה – יש לחשב את התוחלת של המשתנה המקרי X .

התפלגות אחידה: $X \sim U[a, b]$

בהתפלגות זו לכל תוצאה יש את אותה הסתברות. הערכים המתקבלים בהתפלגות החל מ-1 עד b הם בקפיצות של יחידה אחת (לדוגמה הגרלה של מספר שלם בין 1-100).

$$P(X = k) = \frac{1}{b - a + 1}, k = a, a + 1, \dots, b, \mathbb{E}[X] = \frac{a + b}{2}, Var[X] = \frac{(b - a + 1)^2 - 1}{12}$$

התפלגות פואסוני: $(\lambda) \sim poi(\lambda)$

התפלגות זאת מתאפיינת במספר אירועים ליחידת זמן כאשר גַּם הוא פרמטר המיצג את קצב האירועים ליחידת זמן הנבחרת

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}, k = 1 \dots \infty, \mathbb{E}[X] = Var[X] = \lambda$$

יש לשים לב שכאן ההתפלגות נמצדת ליחידת זמן

התפלגות היפר גאומטרית $\sim H(N, D, n)$

נתונה אוכלוסייה שמכילה N פריטים סה"כ, מתוך D "מיוחדים" בעלי תכונה מסוימת. בוחרים מאותה אוכלוסייה n פריטים ללא החזרה. מגדירים אז X להיות מספר הפריטים ה-"מיוחדים" שנדרשו

$$P(X = k) = \frac{\binom{D}{k} \binom{N-D}{n-k}}{\binom{N}{n}}, \mathbb{E}[X] = \frac{nD}{N}, Var[X] = \frac{nD}{N} \left(1 - \frac{D}{N}\right) \frac{N-n}{N-1}$$

התפלגותBINOMIALE SHLILIT $\sim NB(r, p)$

חווזרים על אותו ניסוי ברנולי זה אחר זה באופן בלתי תלוי עד אשר מצליחים בפעם ה- r . כלומר, מבצעים את הניסוי עד שמבצעים r פעמים. מגדירים את X להיות מספר החזרות עד שהתקבלת r הצלחות

$$P(X = k) = \binom{k-1}{r-1} p^r (1-p)^{k-r}, k = r, r+1, \dots, \infty \quad \mathbb{E}[X] = \frac{r}{p}, Var[X] = \frac{r(1-p)}{p^2}$$

התפלגות מיוחדת (רציפה)

התפלגות מעריכית: $(\lambda) \sim exp(\lambda)$

התפלגות רציפה המאפיינת את הזמן עד להתרחשות מאורע מסוים שהוא ממוצע מספר האירועים המתרכשים ביחס לזמן (אותו פרמטר מההתפלגות הפואסונית). $(\lambda) < 0$

גם בהתפלגות זו יש את תכונות חוסר הזיכרון: $P(X > (a+b)|X > a) = P(X > b)$

התפלגות אחידת $\sim U(a, b)$

זו ההתפלגות שפונקציית הצפיפות שלה קבועה בקטע a - b .

פונקציית הצפיפות: $F(t) = \frac{t-a}{b-a}$. פונקציית ההתפלגות המctrברת: $f(x) = \frac{1}{b-a}$.

$$\mathbb{E}[X] = \frac{a+b}{2}, Var[X] = \frac{(b-a)^2}{12}$$

התפלגות נורמלית: $(\mu, \sigma^2) \sim \mathcal{N}(X)$

התפלגות נורמלית היא ההתפלגות חשובה מאוד כיון שהיא מופיעה בהמוני מקרים. פונקציית הצפיפות של ההתפלגות הנורמלית נראה כמו פעמן, כאשר עלוקמה קוראים גם עיקומת גאות. ההתפלגות הנורמליות נבדלות אחורמתהשנית באמצעות הממוצע וסטיית התקף (הפרמטרים שמאפיינים את ההתפלגות). התפלגות נורמלית סטנדרטית היא התפלגות נורמלית בעלת תוחלת 0 ושונות 1.

$\sim \mathcal{N}(0,1)$

עבור תוחלת ושונות μ, σ^2 , פונקציית הצפיפות של משתנה נורמלי הינה

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

ניתן להשתמש במומנטים כדי למצוא קשרים בין ההתפלגות. למשל עבור שני משתנים המתפלגים נורמלית:

$$X \sim \mathcal{N}(\mu_x, \sigma_x^2), Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$$

המומנטים מקיימים

$$M_X(t) \cdot M_Y(t) = e^{\mu_x t + \frac{1}{2} \sigma_x^2 t^2} \cdot e^{\mu_y t + \frac{1}{2} \sigma_y^2 t^2} = e^{(\mu_x + \mu_y)t + \frac{1}{2} (\sigma_x^2 + \sigma_y^2)t^2} = M_{X+Y}(t)$$

ולכן ניתן לחשב את ההתפלגות של $X + Y$:

$$X + Y \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$$

אי שיוניון

מרקז

בהינתן $0 \leq X \leq \mathbb{E}[X]$, עבור פרמטר $a > 0$ מתקיים:

$$p(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$$

צ'בישוב

בבינתן התוחלת $\mathbb{E}[X]$ והשונות $Var[X]$, עבור פרמטר $a > 0$ מתקיים:

$$p(|X - \mathbb{E}[X]| \geq a) \leq \frac{Var[X]}{a^2}$$

צ'רנווק

בבינתן התוחלת $\mathbb{E}[X]$, עבור שני פרמטרים $a, t > 0$ מתקיים:

$$p(X \geq a) \leq \frac{\mathbb{E}[e^{tx}]}{e^{ta}}$$

ינט

עבור משתנה מקרי X בעל תוחלת, עבור פונקציה g קמורה $\mathbb{R} \rightarrow \mathbb{R}$ מתקיים:

$$g(E[x]) \leq \mathbb{E}[g(x)]$$

התפלגות דו ממדית

$$F_{x,y}(a, b) = P(x \leq a, y \leq b)$$

תכנים

$$\lim_{a,b \rightarrow \infty} F_{x,y}(a, b) = 1$$

$$\lim_{a \rightarrow -\infty} F_{x,y}(a, b) = \lim_{b \rightarrow -\infty} F_{x,y}(a, b) = 0$$

$$\begin{aligned} P(c < x < a, d < y < b) &= P(x < a, y < b) - P(x < a, y < d) - P(x < c, y < b) + P(x < c, y < d) \\ &= F_{x,y}(a, b) - F_{x,y}(a, d) - F_{x,y}(c, b) + F_{x,y}(c, d) \end{aligned}$$

אם y בלתי תלויים איז מתקיים:

$$\forall a, b \ F_{x,y}(a, b) = F_x(a) \cdot F_y(b)$$

זוג משתנים נקרא דו-ממדי רציף אם קיימת פונקציית צפיפות דו-ממדית $f_{x,y}(s, t)$, כך שמתקיים:

$$P(x, y \in A) = \int f_{x,y}(s, t) ds dt$$

באופן שקול מתקיים:

$$f_{x,y}(s, t) = \frac{\partial^2}{\partial s \partial t} F_{x,y}(s, t) = \frac{\partial^2}{\partial t \partial s} F_{x,y}(s, t)$$

$$F_x(s) = P(x \leq s) = P(x \leq s, y \leq \infty) = \int_{-\infty}^s \int_{-\infty}^{\infty} f_{x,y}(x, y) dx dy$$

נוסחת ההסתברות השלמה לצפיפות (באופן שקול גם $\hat{f}_y(t)$):

$$f_x(s) = \frac{d}{ds} F(x_s) = \int_{-\infty}^{\infty} f_{x,y}(s, y) dy$$

כעת ניתן גם לכתוב תנאי שקול למשתנים בלתי תלויים – y, x בלתי תלויים אם ורק אם

$$\forall x, y f_{x,y}(X, Y) = f_x(X) \cdot f_y(Y)$$

ստטיטיסטיקה היסקיא

אם ידועים את סוג ההתפלגות אבל לא ידועים את מרכיביה, ניתן לאמוד את המרכיבים בעזרת מדגם. המדגם מאפשר לנו להשתמש באמצעות מספר מאורעות שניי ההתפלגות= X – נניח רוצם למדוד גובה של קבוצה מסוימת= X כל תלמידים בבית ספר מסוים. ידוע שגובה מתפלג נורמלית, אבל לא ידועים כאן את התוחלת והשונות. לשם כך ניתן להשתמש באומד – פונקציה שמנסה לנתח את המאורעות ומתורן כרך להסיק את התוחלת והשונות

בנитוח נצא מנקודת הנחה $\text{שידועהעריכים במדגם נלקחים כולם מתוך ההתפלגות}=X$, השיכת המשפחה של ההתפלגות שתלויות בפרמטר אחד או יותר שאינם ידועים. (למשל בדוגמא= $N \sim \mathcal{N}(\mu, \sigma^2)$, אלא ידועים). בפועל נתוננו= n ידיגימות בלתי תלויות מתוך ההתפלגות= X_1, X_2, \dots, X_n , ורוצים לאמוד את הפרמטרים הלא ידועים (כפונקציה של הערכים שדגמוני).

אומד בלתי מוטה: אומד מוגדר להיות בלתי מוטה אם התוחלת של האומד=שווה לפרמטר אותו אנו מנסים לאמוד= θ , אז האומד הוא חסר הטיה. במילים אחרות= θ – אומד יהיה חסר הטיה אם התוחלת של המשתנה המקרי המוחושב לפי θ שווה $\hat{\theta}$ עבור כל θ .

דוגמאות לאמדים בלתי מוטים:

אומד בלתי מוטה לתוחלת= s ממוצע חשבוני:

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\mathbb{E}(\hat{\theta}) = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n x_i\right] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[x_i] = \mathbb{E}[x_i] = \theta$$

אומד בלתי מוטה לשונות= s^2

$$\mathbb{E}[s^2] = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

הוכחה= \Rightarrow

$$\begin{aligned} \mathbb{E}[s^2] &= \mathbb{E}\left[\frac{1}{n-1} \cdot \sum_i (x_i - \bar{x})^2\right] = \frac{1}{n-1} \sum_i \mathbb{E}(x_i - \bar{x})^2 \\ &= \frac{1}{n-1} \sum_i \mathbb{E}[(x_i - \mu) - (\bar{x} - \mu)]^2 \\ &= \frac{1}{n-1} \sum_i \mathbb{E}[(x_i - \mu)^2 - 2(x_i - \mu)(\bar{x} - \mu) + (\bar{x} - \mu)^2] \end{aligned}$$

$$\begin{aligned} & \frac{1}{n-1} \sum_i \mathbb{E}[(x_i - \mu)^2] - \mathbb{E}[2(x_i - \mu)(\bar{x} - \mu)] + \mathbb{E}[(\bar{x} - \mu)^2] \\ & \frac{1}{n-1} \sum_i \sigma^2 - 2 \left(\frac{1}{n} \sum_j \mathbb{E}[(x_i - \mu)(x_j - \mu)] \right) + \frac{1}{n^2} \sum_j \sum_k \mathbb{E}[(x_j - \mu)(x_k - \mu)] \\ & \frac{1}{n-1} \sum_i \left[\sigma^2 - \frac{2\sigma^2}{n} + \frac{\sigma^2}{n} \right] \\ & \frac{1}{n-1} \sum_i \left[\frac{(n-1)\sigma^2}{n} \right] = \frac{n-1}{n(n-1)} \sum_i \sigma^2 = \sigma^2 \blacksquare \end{aligned}$$

אומד נראות מרבית = Maximum likelihood estimator (MLE)

בහינתן סדרת דגימות מתוך התפלגות עם פרמטר לא ידוע-ונגדיר את פונקציית הנראות שלחן כמכפלת ההסתברויות של כל הדגימות, או "הנראות של המדגם".

$$L(x_1, x_2 \dots x_n | p(\theta)) = \prod_i P_\theta(x_i)$$

זהוי פונקציה הן של הדגימות והן של הפרמטר

אם ההתפלגות רציפה מגדרים במקום זאת פונקציית הנראות להיות מכפלה הצפיפות

$$L(x_1, x_2 \dots x_n | p(\theta)) = \prod_i f_\theta(x_i)$$

נראות מקסימלי עבור θ אם $\hat{\theta} = \arg \max_{\theta} L(x_1, x_2, \dots, x_n | p(\theta))$. אומדן הנראות המקסימלי הוא פשטן הערך של הפרמטר שמקסם את פונקציית הנראות. כולם זה הוא אומדן הנראות המקסימלי.

מכoon ש- \log -הינה מונוטונית-לא递减 למקסם או L שקיים למקסם את \hat{L} (\log likelihood), וזה לרוב יותר קל, מכיוון שהמכפלה הופכת לסדרה עולה.

$$\log L(x_1, x_2 \dots x_n | p(\theta)) = \sum_{i=1}^n \log f_\theta(x_i)$$

נראה מספר דוגמאות לחישוב ה-MLE:

א. מציאת הפרמטר λ בהתפלגות פואסונית:

$$X \sim poi(\lambda)$$

שלב א'=
נגידר את אומדן הנראות $= \prod_i P_\lambda(x_i) = L$. בתפלגות פואסונית מקיימת

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

$$\prod_i P_\lambda(x_i) = \prod_i \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}$$

מסובך למצוא לזה מקסימום, אך נוציא לו

$$\ln\left(\prod_i \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right) = \sum_i \ln\left(\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right)$$

$$= \sum_i \ln(e^{-\lambda} \lambda^{x_i}) - \ln(x_i!) = \sum_i \ln(e^{-\lambda}) + \ln(\lambda^{x_i}) - \ln(x_i!)$$

$$= \sum_i x_i \ln(\lambda) - \lambda - \ln(x_i!)$$

cut נגזרת

$$\frac{\partial L}{\partial \lambda} = \sum_i \frac{x_i}{\lambda} - 1 = \sum_i \frac{x_i}{\lambda} - \sum_i 1 = \sum_i \frac{x_i}{\lambda} - n$$

וכשנשווה ל-0 נקבל:

$$\sum_i \frac{x_i}{\lambda} = n$$

ובודד את הפרמטר אותו מנוטים לאמוּן

$$\lambda = \frac{\sum x_i}{n}$$

וקיבלנו אומד עבור הפרמטר λ , וכך נקבע סט התוצאות, פשוט נציב אותן, ונמצא מפורשות את הערך של האומד. זה בעצם תהליכי מציאת MLE . cut נבדק האם האומד הוא מוטה או לא, כאשר משתמש בעובדה שבעבור התפלגות פואסונית $\lambda = \mathbb{E}(x)$

$$\mathbb{E}(\lambda) = \mathbb{E}\left(\frac{\sum_i x_i}{n}\right) = \frac{1}{n} \sum_i \mathbb{E}[x_i] = \frac{n\lambda}{n} = \lambda$$

קיבלנו שתווחלת האומד שווה לפרמטר, וכך הוא בלתי מוטה.

ב. התפלגות נורמלית

$$X \sim (\mu, \sigma^2)$$

פה יש שני פרמטרים לאמוד – התוחלת והשונות. ראשית נגדיר את הנראות:

$$f(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

לכן הנראות תהיה (נשים לב שהמכפלה תעבור לסכום במעירך של האקספוננט):

$$\prod_i f(x) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2} = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\frac{1}{2\sigma^2}\sum_i (x_i-\mu)^2}$$

נוציא לוג:

$$\begin{aligned} \ln(L) &= \ln\left(\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n\right) + \ln\left(e^{-\frac{1}{2\sigma^2}\sum_i (x_i-\mu)^2}\right) \\ &= n \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \ln\left(e^{\frac{1}{2\sigma^2}\sum_i (x_i-\mu)^2}\right) \end{aligned}$$

נשים לב שבביטוי הראשון מה שיש בתוך ה- $\ln(2\pi)^{-\frac{1}{2}} + (\sigma^2)^{-\frac{1}{2}}$, וזה בערך $\ln(2\pi)^{-\frac{1}{2}} + (\sigma^2)^{-\frac{1}{2}}$, ואז המעריך יכול לרדת מבחן \ln :

$$= -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2$$

cut בשביל לאמוד את התוחלת יש לגזור לפי μ , וכך לאמוד את השונות יש לגזור לפי σ^2 :

$$\frac{\partial L}{\partial \mu} = -\frac{(-2)}{2\sigma^2} \sum_i (x_i - \mu) = \frac{1}{\sigma^2} \sum_i (x_i - \mu)$$

וכשנשווה ל-0 נקבל

$$\hat{\mu} = \frac{\sum_i x_i}{n}$$

ניתן להוכיח **כעבור התוחלת האומד הוא בעצם הממוצע של המדגם.** אפשר לבצע תהליכי דומה על השונות, ומתקבל
הביטוי:

$$\hat{\sigma}^2 = \frac{\sum_i (x_i - \hat{\mu})^2}{n}$$

1. References

Intro:

<https://www.analytics.org.il/2019/12/ai-vs-deep-learning-vs-machine-learning/>

2. Machine Learning

2.1 Supervised Learning Algorithms

2.1.1 Support Vector Machines (SVM)

(SVM) –**Support Vector Machine** – מודל למידה מונחית המשמש לניתוח נתונים לצורך סיווג חיצוני ורגסיה. המודל מקבל אוסף של דוגמאות מתוויות במרחב-a-mmd, ומנסה למצוא מישור המפריד בצורה טובה כמה שיותר בדוגמאות האימון השתייכות לקטגוריות השונות.

המשמעות הנוצרת באמצעות מודל-SVM היא קילינארית כאשר חלוקת הדוגמאות במרחב הוקטור-בנישית באופן צזחי ויציב מרוחק גדול ככל האפשר ביחס לערך המינימום המרחק בין הנקודות הממוקמות היכי קרוב אליו. מרוחק זה מכונה –**margin** – (margin) – כאשר בצד אחד של השוליות נמצאו דוגמאות אסוציאטיביות –**label** – בצד השני נמצא דוגמאות –**label** – בצד השני את המפריד ניתן לייצג באמצעות הנוקוטו –**המתקיים** – $0 = b - \vec{x} \cdot \vec{w}$ – כאשר –**הו** – וקטור נורמלי של המישור.

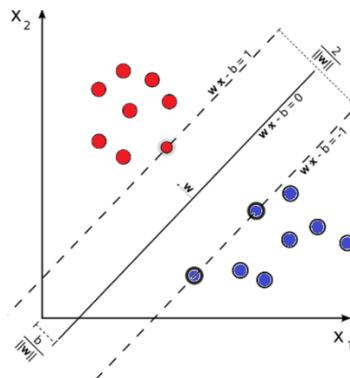
נסח את האלגוריתם באופן פורמלי – נתן אוסף של נקודות $(x_i, y_i) \in \{-1, 1\}$ – $\forall i \in \{1, \dots, n\}$ – מינימיזג את התויג המתאים לדוגמא*i*, $\vec{w} \in \mathbb{R}^d$ – הוא וקטור המאפיין –**המתאים** – מודל-SVM מיציר מישור המפריד את המרחב לשני מרחבים אחד מהם אמרור להכליל עיקרי דוגמאות מסווג תיווג אחד. בנוסף –**המודול** – מיציר שני מישורים מקבילים לו, אחד מכל צד, במרחב זהה וגדול ככל האפשר:

$$w^* = \operatorname{argmin}_w \left(\frac{1}{2} \|w\|^2 \right), \text{ s.t. } \forall i (x_i \cdot w + b \geq 1)$$

כלומר, רוצים למצוא את וקטור המשקל w –**שהמיציר** – $\|w\|^2$ – $y_i(x_i \cdot w + b) \geq 1$ – וAIN –**בתוך** – השוליות –**לא מתקיים** – $y_i(x_i \cdot w + b) < 1$ – $y_i(x_i \cdot w + b) > 1$ – ישנו מספר גישות למציאת המפריד, ונפרט על כמה מהן:

Hard-Margin (hard SVM)

במצב הפשוט ביותר, המשווה עברו כל אחד מצדדיו של המפריד הינה פונקציה ליניארית של המאפיינים וכל הדוגמאות אשר סווינו נכונה. מצב זה מכונה " הפרדה קשיחה " בו האלגוריתם מוצא את המישור עם השול הרחב ביותר האפשרי, ולא מאפשר לדוגמאות להיות בין הווקטוריים התומכים. זהוי למעשה הפרדה מושלמת, והווקטוריים התומכים הם למעשה הנקודות בקצוות השוליות, כפי שניתן לראות באירוע:



איך –**סיווג** – באמצעות אלגוריתם-SVM עם מפריד בעל השוליות הרחבים ביותר. היקו האמצעי מינימיזג את המפריד, הקווים המקווקווים –**מייצרים** – מישורי השוליות –**דוגמאות האימון** – המטלדות עם מישורי השוליות –**נקראות וקטורי תומכי** – (support vectors), ומכאן נגזר שם האלגוריתם.

את המישוריים בקצוות השוליות ניתן לייצג באמצעות –**המරחק** – $b = 1 - \vec{x} \cdot \vec{w}$ – or –**המישוריים** – הוא $\frac{2}{\|\vec{w}\|}$ – ולכן על מנת למקסם את המרחק זהה, יש מהביא למינימום את $\|\vec{w}\|$. על מנת שדוגמאות האימון לא יכללו בשוליות המפרידים, יש להוסיף אילוץ לכל דוגמא*i*, באופן הבא:

$$y_i(\vec{x}_i \cdot \vec{w} - b) \geq 1$$

ailoz זה מחייב שכל דוגמא תהיה בצד הנכון של המפריד. לכן, במקרה זה יש לקיים את הדרישה הבאה:

$$\min_{w,b} \|w^2\|$$

$$s.t \quad y_i (\vec{w} \cdot \vec{x}_i - b) \geq 1 \quad \forall i = 1 \dots n$$

Soft-Margin (soft SVM)

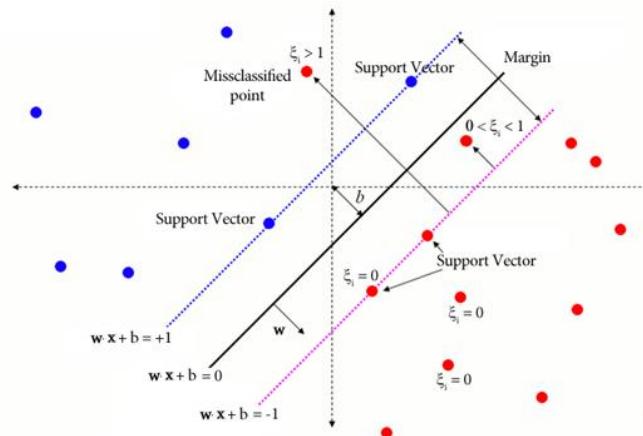
הפרדה מושלמת באמצעות מישור לינארי לעתים קרובות אינה אפשרית, ולכן נחיה את המודל כך שיאפשר לנקודות מסוימות לא להיות בצד המותאים לה. הרחבה זו, היוצרת " הפרדה רכה ", מאפשרת לטפל בעיות שבהן אין הפרדה לינארית בין הקבוצות, כמו למשל שיש נקודות חרגנות. משמעו החרחבה היא שכל וקטור ממוקם בצד אחד מהאלזיטים – אך עם זאת נרצה让他 גע למצב בו האלזיטים מופרים "כמו שפוחות" – הפרדה רכה יוצרת מצב בו trade-off בין רוחב השולץ לבין השגיאות ומיציאת המשקלים האופטימליים של המסוג ובגרסתazzish לרשום באופן מושג – שונחתת בעית האופטימיזציה, כאשר מתווסף משתנה המתיחס לנקודות שאין נמצאות בסיווג המתאים לו לפוי המפריד.

$$\min ||w^2|| + C \sum_{i=1}^n \xi_i$$

$$s.t \quad y_i (\vec{w} \cdot \vec{x}_i - b) \geq 1 - \xi_i \quad \forall i = 1 \dots n \quad \xi_i \geq 0$$

לשם קבלת אינטואיציה, נשים לב לתפקיד המשתנים:

אם $\xi_i = 0$, מתקיים התנאי שנדרש בהפרדה קשיחה, כלומר הנקודה x_i נמצא בצד הנכון של המפריד וגם מתקיימת הדרישה לשמירה על השולץ $y_i(\vec{w} \cdot \vec{x}_i - b) = 1$. אם $\xi_i > 0$ – אז הנקודה x_i נמצא בצד הנכון של המפריד המסוג – אבל המסוג קרוב אליה, כך שהנקודה נמצאת בתוך השולץ $y_i(\vec{w} \cdot \vec{x}_i - b) < 1$. במקרה ערך $\xi_i = 0$ – נזקן מעדיף הכללה (שלויים רוחבים), גם במקרה האימון הספציפיות אין מסוגות נכון.



איו-2. סיווג באמצעות אלגוריתם SVM. עם הפרדה רכה=המשתנה ξ שווה לאפס אם הנקודה ממוקמת בצד הנכון של המפריד=וגדול מאפס כאשר הנקודות נמצאות בצד הלא נכון של המפריד=

Non-linear Separation

מסוגים לינאריים מוגבלים ביכולת ההכללה שלהם בגלל הפשטות שלהם. לכן, כאשר לא ניתן להפריד איסוף דוגמאות באמצעות מפריד לינארי, משתמשים ב" הפרדה אל-لينארית ". גישה זו מאפשרת להשתמש ב-SVM Kernel Trick לא לינאריך על ידי טרנספורמציה לא-لينארית, כמו למשל "תעלול הגערין" (Kernel Trick). בגישה זו מבצעים מיפוי לדataspace מרחב אחר, בו ניתן למצוא עבורה הפרדה לינארית, ומילא אליה אפשר להשתמש באlgorigrithm=SVM=Kernel mapping=אפשרות ליצור=משמעותי=חדשני=על=ידי=העלאת=ערכי=המאפיינים=הקיים=חזקה מסויימת, הכפלת&בפונקציות טריגונומטריות וככ

באופן פורמלי, נחפש פונקציית מיפוי להעתקת מרחב $F \rightarrow \mathcal{X}$: ψ כך שבסמך $y = \sum_{i=1}^N \psi(x_i)$ במאזעות מסווג ליניארי= ψ כר, משתמשים בטrik קרナル שמקבל כקלט וקטורים למרחב המקורי ומחזיר את המכפלה הפנימית (dot product) של הווקטורים למרחב החדש (נקרא גם מרחב התכונות=feature space):

$$K(\vec{x}_i, \vec{x}_j) = \psi(\vec{x}_i)^T \psi(\vec{x}_j)$$

דוגמאות של פונקציות קרナル נפוצות:
קרナル ליניארי:

$$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$$

זהה הפונקציה היכי פשוטה המוגדרת על יד מכפלה פנימית של הווקטוריים= $\vec{x}_i \cdot \vec{x}_j$ זה מרחב התכונות ומרחב הקלאס זהים ונחזר לפתרון בעזרת SVM ליניארי:
קרナル פולינומי:

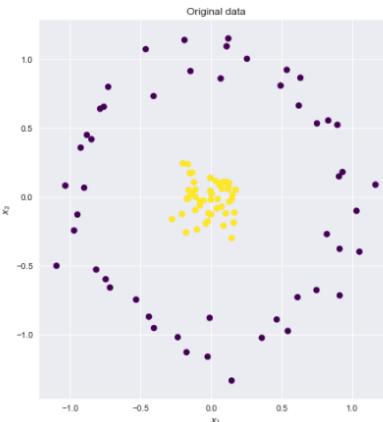
$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + c)^d$$

העתקה מהמרחב המקורי למרחב שמהווה פולינום ממעלה $d \geq 0$ \geq זה הוא פרמטר חופשי המשפייע עליהו בין סדר גובה לעומת סדר נמוך בפולינום. כאשר $c = 0$, ה الكرナル נקרא הומוגני.
קרナル גאוסיאני:

$$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma(\vec{x}_i - \vec{x}_j)^2), \gamma > 0$$

הפרמטר γ =ממלא=תפקידי=חסוב, יש לבחור=uות忿בהתאמלבעיה העומדת בפנים. אפההער שלו קטן מאוד, האקספוננסיאלית כמעט לא פונקצייה של מרחב אחר מממד גובה יתחילה לאבד מכוחה להלא ליניארי. מצט שני, אפננערו אותו יתר על המידה, הפונקצייה תהייה סידרכ=גבול ההחלטה יהיה רגיש מאוד לרעש בתונען האיןן.

המהות של טrik קרナル היא שניתן לבצע את העתקה גם מבלי לדעת מהי הפונקצייה, אלא הידע שלפניהם לצורך קבלת אינטואיציה והמחשה נביא דוגמא. נתון מערך הנתונים הבא:



ניתן לראות שלא ניתן להפריד בין הנקודות הצהובות לשגולות על ידי מישור הפרדה ליניארי=לכן נחפש מרחב אחצ' מאותו ממד אך בעל ממד גבוה יותר=נותקה להפריד בין נקודות אלה באופן ליניארי=לצורך כך לבצע את הפעולות הבאות:

- א. נמפה את התכונות המקוריות למרחב הגבוה יותר (מיפוי תכונות=)
- ב. נבצע SVM ליניארי למרחב החדש.
- ג. נמצא את קבועות המשקלות התואמות את מישור גבול ההחלטה.
- ד. נמפה את מישור המפריד בחזרה למרחב הדו-ממד' המקורי כדי לקבל גבול החלטה לא ליניארי.

ישנם הרבה מרחבים מממדים גבוהים יותר בהם נקודות אלה ניתנות להפרדה ליניארית. נציג דוגמא אחת:

$$x_1, x_2 \rightarrow z_1, z_2, z_3$$

$$z_1 = \sqrt{2}x_1x_2 \quad z_2 = x_1^2 \quad z_3 = x_2^2$$

לפנעה נעזרנו בטריק קרナル=כאמור, בהינתן ש"מ φ את הוקטורים ממרחב \mathbb{R}^n למרחב תכונות כלשהו \mathbb{R}^m =אז המכפלה הפנימית של x_1 ו- x_2 א' במרחב זהה היא $(\varphi(x_1))^T \varphi(x_2)$. קרナル היא פונקציית φ שהxicת המכפלה הפנימית זו, כולם $K(x_1, x_2) = (\varphi(x_1))^T \varphi(x_2)$ =אם נוכל למצוא פונקציית קרナル המקבילה למפה התכונות שלעיל, נוכל להשתמש בפונקציה ייחד עפ-VM φ לינארי וכך לבצע את החישובים בעילום.

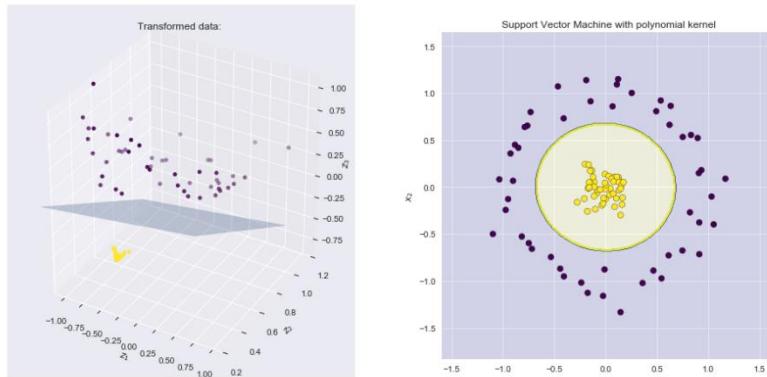
מתברר שמרחב התכונות שלעיל תואם את השימוש בקרナル פולינומי ידוע= $K(x, x') = (x^T x')^d$ =נבחר=2. נסמן $x = (x_1, x_2)^T$ ונקבל:

$$K\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}\right) = (x_1 x'_1 + x_2 x'_2)^2 =$$

$$2x_1 x'_1 x_2 x'_2 + (x_1 x'_1)^2 + (x_2 x'_2)^2 = (\sqrt{2}x_1 x_2 x'_1 x'_2) \begin{pmatrix} \sqrt{2}x'_1 x'_2 \\ x'_1^2 \\ x'_2^2 \end{pmatrix}$$

$$K\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}\right) = \varphi(x)^T \varphi(x')$$

$$\varphi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} \sqrt{2}x_1 x_2 \\ x_1^2 \\ x_2^2 \end{pmatrix}$$



איו-3. שימוש ב-VM φ =לצורך הפרדה לאחר ביצוע Kernel trick=המעגל באור הימני מופנה למשור הפרדה לינארי במרחב מממד גבוה יותר, כפי שניתךראות באור השמאלי=ניתן לראות שאחרי המיפוי שנעשה בעזרת kernel trick, המיקודות אכן מופרדות בצורה לינארית.

2.1.2 Naive Bayes

סיווג ביסיאנְהוּס מודל המשמש בחוק ביחס על מנת לסוג אובייקט \mathbb{R}^n א' בעקבות מאפיינְהוּס כלאות מ- K =קטגוריות אפשריות=יחד עם השימוש בחוק ביחס, המודל מניח "נאייבות"=בהתנחת סיווג של אובייקט מסוים, אין תלות בז' המאפיינְים השונים של-

נוי-השיש מודל-המקבל וקטו:=מאפיינְהוּס כלאות מ- K =cab ראש, משטעל, חום גבורה], ומסווג האם אדם בעל תכונות אלה חוליה בשפעת או לא=באופן כללי ניתן לומר=שיש תלות ביחס על בין חום גבורה=כלומר העובדה שיש לאדם חום מעלה את ההסתברות שהוא גם משטעל=למרות זאת, ניתן להניח באופק"נאייב" שאם כבר יודעים שאדם חוליה בשפעת=אז כבר אין יותר תלות בין היותו משטעל להיותו בעל חום=באופן פורמלי=אם גם שמייר להניח שמתתקי'פ (משתעף) $k > (\text{חום}| \text{משטעל})$, אך ניתן להניח נאייבות ולקבל=(שפעת|משטעל) $k = (\text{שפעת}, \text{חום}| \text{משטעל})$.

באופן כללי, סיווג ביסיאני נאייב מניח שבהינתן הסיווג של אובייקט מסוים, המאפיינְים של-ביבליות תלויים. הנחחה כמובן לא-תמייד-מודיקת, ומילא גפ-ערכיה ההסתברויות הנובעים=FROM ממנה ומשמשים לשיווגאים מדייקים, אף ההנחה

מקלה מאוד על חישוב ההסתברויות של הסיווג הביאו-במקרים רבים תחת ההנחה זו התקבלו תוצאות סיוואת הסיבה להצלחת המודל נועצה בכך שבבבליות סיווג העיקר הוא למצוא את הסיווג הסביר ביותר לאובייקט (שפעה-אלא-שפעת לנבדק **דוגמא**, וולאו דואק לקל ההסתברות מדויקת לכל סיווג. במקרים רבים למורות שההסתברות הנובעת מההנחה הנאייבית אינה מדויקת עבור שני סיווגים אפשריים, היא בכל זאת שומרת על סדר ההסתברות שליהם).

נתבונן בוקטור $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, היכל להיות שיר לאחת מ- K =**קטגוריות** $= (y_1, \dots, y_k)$. התחפלגות הפריאורוֹת $p(y_k | \mathbf{x})$ ידועות התחפלגות המותנות של המאפיינים בהינתן הסיוואת $= p(x_i | y_k)$ בעזרה הנתונים האלה רוצים לסwoג את כל אחת מהקטגוריות כלומר למצוא אף y_k עשבו**ר** הביטוֹב $(y_k | \mathbf{x})$ מהו**י** מי. באופן פורמלי ניתן לנוכח זאת כה

$$y = \arg \max_k p(y_k | \mathbf{x}), k = 1 \dots K$$

בשביל למצוא אף y_k האופטימלי ניתן להיעזר בחוק ביביָה:

$$p(y_k | \mathbf{x}) = \frac{p(y_k, \mathbf{x})}{p(\mathbf{x})}$$

המכנה לא תלוי ב- k , ולכן מספיק למצוא אף y_k שעבור המונה מקסימלי. לפי כלל השרשרת מתקיין**ו**:

$$p(y_k, \mathbf{x}) = p(y_k, x_1, x_2, \dots, x_n) = p(x_1 | y_k, x_2, \dots, x_n) \cdot p(y_k, x_2, \dots, x_n)$$

$$= p(x_1 | y_k, x_2, \dots, x_n) \cdot p(x_2 | y_k, x_3, \dots, x_n) \cdot p(y_k, x_3, \dots, x_n)$$

$$= p(x_1 | y_k, x_2, \dots, x_n) \cdot p(x_2 | y_k, x_3, \dots, x_n) \cdots p(x_{n-1} | y_k, x_n) \cdot p(x_n | y_k) p(y_k)$$

כעת נשתמש בהנחות הנאייבות, לפי בהינתן הסיוואת y_k , אין תלות בין המאפיינים. לפי הנחה זו נוכל לפשט את הביטוי

$$= p(x_1 | y_k) \cdot p(x_2 | y_k) \cdots p(x_{n-1} | y_k) \cdot p(x_n | y_k) p(y_k)$$

$$= p(y_k) \prod_{i=1}^n p(x_i | y_k)$$

בביטוי זה כל האיברים ידועים, ולכן כל שנותר זה רק להציב את הנתונים ולקבל אף y_k עבורי ביטוי זה הכי גודל.

$$y = \arg \max_k p(y_k) \prod_{i=1}^n p(x_i | y_k)$$

בדוגמא שהובאה לעיל, המאפיינים $= \mathbf{x}$ עריכ-קבידים, ולכן ניתן לחשב את ההסתברות המותנית של כל מאפיין $(y_k | \mathbf{x})$ קעל ידי ספירה-כמה הפעמים שמנוףיע כ-מאפיין-באוכלו**ו** הנדגמת חלק בגדול המדגם=**עבו**רericums רציפים (כמו למשל מחיר מניה, גובה של אדם וכדו'), אין אפשרות לחשב כך את ההסתברות המותנית. במקרים כאלה יש **ל**הניח התפלגות מסוימת עבור המדגם, ולהסביר את הפרמטרים של התפלגות שיטות שונות (למשל בעזרת נראות מרבית = MLE). עבור מדגם המתפלג נורמלית, ההסתברות המותנית היא גואויאן:

$$p(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}}$$

כאשר σ_y^2 , μ_y הם הפרמטרים של התפלגות, ואמרור הם משוערים בעזרת MLE או שיטת שערוך אחרית=**אף** התפלגות היא לא נורמלית=**וניתן** להשתמש באלגוריתם Kernel density estimation=**עבו**ר שערוך התפלגות גישה אחרת להתמודדות עם מאפיין=**היכולים** לקבל עריכים רציפים היא לבצע=Diskretization=לערפן את המאפיינים יכולים לקבל.

במקרה **המולטיבומי**, בההתפלגות היא רפمدיה=**ומציגת** תוצאה של סדרה בלתי תלואה, יש לחשב את הנראות באופן המתאים לההתפלגות מולטינומית. כדי להבין את החישוב נביא קוד-בוגרמא – נניח ורוצים לבנות מודל סיוואת בייסיאנית-המזהה הودעות ספאם. נתנו x_1, x_2 הודעות, מתוכן x_1 אמיתיות ו- x_2 ספאם. כעת נניח-כל ההודעות מורכבות-

מאוסף של ארבע מילים, בתפלגות הבא:

Real (R) – {Dear, Friend, Lunch, Money} = {8, 5, 3, 1}.

Spam (S) – {Dear, Friend, Lunch, Money} = {2, 1, 0, 4}.

נחשב את הנראות – ההסתברות של כל מילה בהינתן הסיווג

$$p(\text{Dear}|R) = \frac{8}{17}, p(\text{Friend}|R) = \frac{5}{17}, p(\text{Lunch}|R) = \frac{3}{17}, p(\text{Money}|R) = \frac{1}{17}$$

$$p(\text{Dear}|S) = \frac{2}{7}, p(\text{Friend}|S) = \frac{1}{7}, p(\text{Lunch}|S) = 0, p(\text{Money}|S) = \frac{4}{7}$$

כעת נבחן מה ההסתברות שהצירוף "Dear friend" הוא אמיטיתית=הצירוף הוא למעשה התפלגות מולטינומית, כיוון שהוא מכיל שתי מיללים שאין בין ההסתברויות שלhn קשור ישר

$$p(\text{Dear friend is R}) = p(R) \cdot p(\text{Dear}|R) \cdot p(\text{Friend}|R) = 0.67 \cdot 0.47 \cdot 0.29 = 0.09$$

$$p(\text{Dear friend is S}) = p(S) \cdot p(\text{Dear}|S) \cdot p(\text{Friend}|S) = 0.33 \cdot 0.29 \cdot 0.14 = 0.01$$

מספרים אלה ניתן להסיק שהצירוף "Dear friend" אינו ספאם.

באופן כללי, עבור וקטור מאפייני- $x \in \mathbb{R}^n = (x_1, \dots, x_n)$, הנראות מחושבת באופן הבא:

$$p(x|y_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p(y_{ki})^{x_i}$$

על הציר הלוגריטמי, בעזרתו נסוכה זו ניתן לבנות מסווג לינארי:

$$p(y_k|x) = \frac{p(y_k, x)}{p(x)} \propto p(y_k) \cdot \prod_i p(y_{ki})^{x_i}$$

$$\rightarrow \log p(y_k|x) \propto \log p(y_k) + \sum_i x_i \cdot \log p(y_{ki}) \equiv b + w^T x$$

החישוב בשימוש במסווג בייסיאני נאיבי בעיות מולטינומיות נועז בכך שיש הרובה צירופים שלא מופיעים יחד בסוט האימון, ולכן הנראות שלהם תמיד תהיה 0, מה שפוגם באמינותות התוצאות.

מקרה דומה להתפלגות מולטינומית הוא מקרה בו המאפיינים משתנים ברונול, מקבלים ערכי-קבינאריים. במקרה זו הנראות הינה:

$$p(x|y_k) = \prod_{i=1}^n p_i^{x_i} (1 - p(y_{ki}))^{1-x_i}$$

עבור דата לא מאוזן, ניתן להשתמש באלגוריתם שנקריך-complement naive Bayes (CNB) לфи אלגוריתם זה במקומו ללקחת את $(x_i|y_k)$ במקומות $\arg \max_k p(y_k) \prod_{i=1}^n p(y_{ki})$

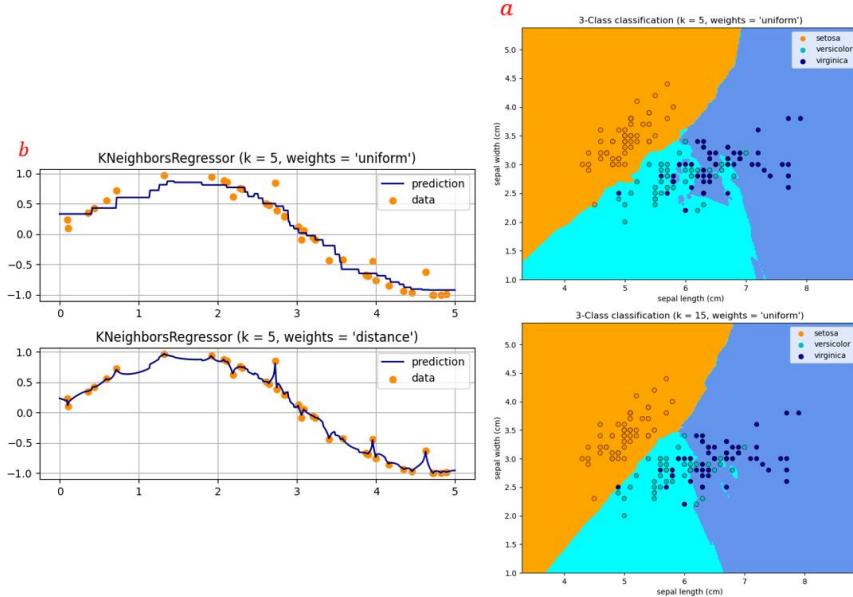
$$\arg \min_k p(y_k) \prod_{i=1}^n \frac{1}{p(x_i|y_k)}$$

שימוש באלגוריתם זה הוכח כיעיל במקרים בהם הדטה אינם מאוזן והביצועים שלו מסווגים בייסיאנים אחרים (גאוסיאני או מולטינומי) היה לא מספיק טוב.

2.1.3 K-Nearest Neighbors (K-NN)

אלגוריתם השכן הקרובינו אלגוריתם של למידה מונחית, בו נתנות מספר דוגמאות ובונספ' ידוע ה-label של כל אחת מהן. אלגוריתם זה מתאים הן לבעיות סיווג (שיוך נקודה חדשה למחלקה מסוימת) והן לבעיות רגרסיה (נתינת ערך מסוין לנקודה חדשה). האלגוריתם הינו מודל חסר פרמטרים, והוא מבצע סיווג לנ נתונים בעזרת הכרעת הרוב. עבור כל נקודה במדגם, המודל בוחן את ה-label של nearest neighbor והקורהות אליו ביותר, ומסווג את הנקודה לפי ה-label שקיבל את מרבית הקווות. מספר הנקודות הקרובות, K, הוא היפר-פרמטר שנקבע מראש.

אלגוריתם השן הקרוב הוא אחד המודל הנפוצים והפושים ביותר בלמידה מוכנה, כאמור בסוף לסתוג הוא מתאים גם לביעות גרגסיה. המודל יפעל בצורה דומה בשני המקרים, כאשר ברגרסיה יתבצע שקלול של מוצע בין השכנים הקרים, ולא הכרעת הרוב, כלומר, התוצאה לא תהיה סיווג ל-label-but-most-of-label'השכנים הקרים, אלא חישוב ממוצע של כל ה-label-but-most-of-label'השכנים. התוצאה המתקבלת היא ערך רציף, המיצג את הערכיה בסביבת התצפית. ניתן להתחשב במרקח של שן מהצפיה-בצורה שווה-uniform), וכן לתת משקל שונות לכלהן בהתאם למראק של מהנקודה אוטה רצים לחשב, כך שכל שן מושם קרוב יותר לנוקודה אוטה רצים לחשב כך הוא יותר ישפי עלייה, ביחס של הופכי המראק בין השן לנוקודה (distance).



א. 2.4-א: מושג-בצורת אלגוריתם KNN-**K**-מושג-בהתאם ל-K-השכנים הקרים ביותר, כך שאם תבוא נוקודה חדשה היא מושג-בהתאם לצבע של האזור שלו, הנקבע באמצעות השכנים הקרים ביותר-ניתן לראות שיש הבדל בין ערך K-שוניים, וככל ש-K-יותר גבוה ככל האזוריים יותר חלקים יש פוחות מובלעות=K-ברגרסיה בעדרת אלגוריתם KNN-**K**-קביעת ערך ה-K-בהתאם ל-K-השכנים הקרים ביותר. ניתן לתת משקלים שווים לכל השכנים, או לתת משקל ביחס למראק של כל שן מהנקודה אוטה רצים לחשב.

לעתים נאמר על המודל שהוא "עצלן". הסיבה לכך היא שבשלב האימון לא מבצע תהליכי ממשועוט, מלבד השמה של המשתנים וה-label-but-most-of-label'ים של המחלקה, כמו גם כל נוקודה מסוימת למחלקה מסוימת. עוקב לכך, כל מבחן האימון (או רבו) נדרש לצורך התחזית, מה עשוי להפוך את המודל לאיטי כאשר יש הרבה נתונים. למוראות זאת, המודל נחשב לאחד המודלים הקלטיים הבולטים, בזכות היתרונותיו שלו. הוא פשוט וקל לפירוש, עובד היטב עם מספר רב של מחלקות, ומתאים-לבעיות רגרסיה וסיווג. בנוסף לכך הוא נחצב אמין במיוחד, כיון שהוא לא מניח הנחות לגבי התפלגות הנתונים (כמו רגרסיה ליניארית למשל).

מנגד, יש לו מספר חסרונות. עקב העבודה שהוא דורש את כל נתונים האימון לצורך התחזית, הוא עשוי להיות איטי כאשר מדובר על דатаה עשיר. מסיבה זו הוא גם יעיל מבחינת זיכרון. מכיוון שהמודל דורש את כל נתונים האימון לצורך המבחן, כשר הכללה שלו עשוי להיות להיפגש (Generalization). ניקח לדוגמא-מורה של Cieta בית ספר, המנסה לסוג את התלמידים למספר קבוצות. אם יעשה זאת לפי צבע שער, עיניים, לדוגמאות, סביר להניח שלא יתקשה בקשר אפלואומת זאת הוא-ינסה לסוג לפי צבע שער, עיניים, חולצה, מכנסיים, נעליים, וכך-סביר שיתתקל בקושי. במצב זה, כל תלמיד רחוק מרעהו באופן שוא-הכיוון שאין שני תלמידים שזהם לחוטין בכל הפרמטרים-מה שמקשה על חישוב המראק. בעיה זו מכונה קלתת הממדיות=Curse of dimensionality (Curse of dimensionality reduction)

קושי נסף-הקיי-יבמודול הוא הצורך בבחירה-K-הנקון, מטלה שעשויה להיות לא קלה לעיתים-בכל מימוש של אלגוריתם השן הקרוב-K-היא היפר-פרמטר שצריך להיקבע מראש. היפר פרמטר זה קובע את מספר הנקודות אשר האלגוריתם יתחשב בהן בעת בחרית סיווג התצפית. בחירת היפר-פרמטר קטן מדי, לדוגמה K=1, יכולה לגרום למצבי-המודול מותאם יתר על המידה לנוקדי האימון, מה שוביל לדיווק גבוה נתונים האימון, וזיקוק נזוף בנתוני המבחן-מן העבר השניא-כאשר-גבוה מידי, למשקל 100 = K, נוצר מצב ההיפר-מודול שמתהשך יותר מכך בDATA-וקולא מצליח למציאת הכללה נכונה לסתוג. מומלץ לבחוח=A-זוגי בغالל אונן הפעולה של האלגוריתם-הכרעה

הרוב. כאשר בוחרים \hat{z} זוגי, עלולים להיתקל במצב של שווין אשר עשוי להוביל לתוכאה מוטעית, ולכך כדי להימנע מתיקו כדאי לבחור \hat{z} אי-זוגי.

כמו אלגוריתמים רבים מבוססי מרחק, אלגוריתם השכן הקרוב רגש לערכים קיצוניים (Outliers) או שימוש באלגוריתם ללא טיפול בערכים קיצוניים עשוי להוביל לתוצאות מוטעות בלבד זאת, חשוב לנормל את הנתונים לפני שימוש במודל. הסיבה לכך היא שהאלגוריתם מבוסס מרחק בלבד זה, יתכונו מרחקים בין תכיפות אשר עשויים להשפיע על החלטה המודל, למراتות שמרחקים אלו הם חסרי משמעות לצורך הסיווג. דוגמא לכך היא משתנה שעשויה שימוש ביחסות מידת שונות (מיילומטרים). ההחלטה האם להשתמש בקילומטרים או במילים עלולה להטות את תוצאות המודל, למراتות שבפועל לא השתנה דבר.

השיטה הנפוצה ביותר למדידת מרחק בין משתנים רציפים היא מרחק אוקלי $d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$. בינהם יוחשב לפניהן סיגנום $\text{sgn}(x_1 - y_1)$. במידה ומדובר במשתנים בדידים, כגון טקסט, ניתן להשתמש במטריקות אחרות כגון המיניג, המודד את מספר השינויים הדרושים כדי להפוך מחרוזת אחת לחרוזת שנייה, ובכך למדוד את הדמיון ביניהן.

לפני שימוש באלגוריתם השכן הקרוב לוודא שהמחלקות מאוזנות. במידה ומספר דוגמאות האימון באחומי המחלקות גבוהה מאשר בשאר המחלקות, האלגוריתם יטה לסוג למחלקה זאת. הסיבה לכך היא שבשל מספרן הגדל, מחלוקת זו צפואה להיות נפוצה הרבה יותר בקרבת השכנים של כל תכיפות. הדבר עשוי להביא לתוצאות מוטטות, ולכן מראש שכן יש איזון בין המחלקות השונות.

2.1.4 Quadratic\Linear Discriminant Analysis (QDA\LDA)

Quadratic Discriminant Analysis – מודל ניסוי-תiode של דата לקבוצות שונות, המニア-שבדה-הינתן סיגנום ש-אובייקט מסוים – מתקבלת התפלגות נורמלית, כולם בהינתן K , $y_k, k \in \{1, \dots, K\}$, מתקיים

$$x | y_k \sim N(\mu_k, \Sigma_k)$$

ובאופן מפורש, עבור $x \in \mathbb{R}^{n \times d}$ הפילוג המותנה הוא:

$$p(x|y=k; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right)$$

בעזרת הנחה זו, ניתן למצוא מסווג אופטימלי עבור $(x|y_k) = y$. לפי חוק ביבי:

$$p(y_k|x) = \frac{p(x|y=k)p(y)}{p(x)}$$

המכנה קבוע ביחס y_k , ואת $(y|k)$ ניתן לשער בקבילות על פי השכיחות של כלabel $\frac{\mathbb{I}_{y=k}}{n}$ במדגם. נסמן את הביטוי הנוסף במנוגה $= (x|y=k)$ – שמשמעותו מתפלג נורמלית, ניקח לשער בבעזרת הנראות מרבית (MLE) ונקבל:

$$\theta_{MLE} = \arg \max_{\theta} p(x|y) = \arg \max_{\theta} \log p(x|y; \theta)$$

$$= \arg \max_{\theta} \log \sum_{i=1}^n p(x_i|y_i; \theta)$$

ניתן לפרק את הסכום לפי ה-label של כל דגימה:

$$= \arg \max_{\theta} \log \sum_{i \in y_i=1} p(x_i|y_i=1; \theta) + \log \sum_{i \in y_i=2} p(x_i|y_i=2; \theta) + \dots + \log \sum_{i \in y_i=K} p(x_i|y_i=K; \theta)$$

כעת בשביל לחשב פרמטרים עבור \hat{y}_k מספיק להסתכל על הדגימות עבור $k = y$, כולם:

$$\theta_{k_{MLE}} = \arg \max_{\theta_k} \log \sum_{i \in y_i=k} p(x_i|y_i=k; \theta_k)$$

על ידי גזירה והשווואה $\neq 0$ ניתן לחשב את הפרמטרים האופטימליים:

$$\mu_k = \frac{\sum_{i \in y_i=k} x_i}{\sum_i \mathbb{1}_{y_i=k}}$$

$$\Sigma_k = \frac{\sum_{i \in y_i=k} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \mathbb{1}_{y_i=k}}$$

ניתן לשים לב שהתוחלף נקבעה למשא ממוצע הדגימות עבור k ערך הפורמטרים המשוערכינית לבנות את המסויוג:

$$\begin{aligned} y &= \arg \max_k p(y_k | x; \mu_k, \Sigma_k) = \arg \max_k \log p(x|y=k)p(y) \\ &= \arg \max_k -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log p(y) \end{aligned}$$

עבור המקירה בו מטריצת covariance היא אלכסונית, כלומר אין תלות בין משתנים שונים, מקבל המסויוג הביסיאני הגאוסיאני (תוצאה זו הגיונית כיוון שהמסויוג הביסיאני מניח שבاهינתן סיווג של אובייקט מסוים אין יותר תלות בין המשתנים)

עבור המקירה הבינארי, ב- $\{0, 1\}$, מקבל סיווג בצורה של משווה ריבועית

$$y = 1 \Leftrightarrow -\frac{1}{2} \log |\Sigma_1| - \frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + \log p(y=1) > -\frac{1}{2} \log |\Sigma_0| - \frac{1}{2} (x - \mu_0)^T \Sigma_0^{-1} (x - \mu_0) + \log p(y=0)$$

ונמקה

$$a = \frac{1}{2} (\Sigma_1^{-1} - \Sigma_0^{-1})$$

$$b = \Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0$$

$$c = \frac{1}{2} (\mu_0^T \Sigma_0^{-1} \mu_0 - \mu_1^T \Sigma_1^{-1} \mu_1) + \log \frac{p(y=1)}{p(y=0)} - \log \frac{\sqrt{|\Sigma_1|}}{\sqrt{|\Sigma_0|}}$$

ונקבל:

$$y = 1 \Leftrightarrow x^T a x + b^T x + c > 0$$

וזהו משטח הפרדה ריבועית

ב-**Linear Discriminant Analysis**, LDA, ב-**מן ה- y** מטריצת covariance זהה לכל ה-labels, Σ_k . במקירה זיהמתקבלי

$$\log p(x|y=k)p(y) = -\frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) + \log p(y)$$

הביטוי $(x - \mu_k)^T \Sigma^{-1} (x - \mu_k)$ נקרא מרחק מהלוביס, והוא מבטא את מידת הקשר בינה לבין כדי התחשבות בשונות של כל משתנה-למעשה ניתן להסתכל על מסויוג LDA כמסויוג המשיר אובייקט ל-label בעבור המרחק על פי מטריקת מהלוביס הוא הערך קטן. על ידי גזירה והשווואה ל-0 מקבל השערור:

$$\mu_k = \frac{\sum_{i \in y_i=k} x_i}{\sum_i \mathbb{1}_{y_i=k}}$$

$$\Sigma_k = \frac{1}{n} \sum_i (x_i - \mu_k)(x_i - \mu_k)^T$$

ומסווג המתקבל הימצא

$$y = \arg \max_k p(y_k | x; \mu_k, \Sigma_k) p(y) = \arg \max_k -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log p(y=k)$$

$$= \arg \max_k -x^T \Sigma^{-1} \mu_k + \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log p(y = k)$$

ניתן לסמך:

$$a = \Sigma^{-1} \mu_k$$

$$b = \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log p(y = k)$$

ומתתקבל מטוג לינארי (ומכאן השם של האלגוריתם)

$$y = \arg \max_k a x^T + b$$

מטוג זה מחלק כל שני אזורים בעזרת מישור לינארי $a^T x + b = 0$ הפרדה עברו המקרא הבינאי מתקיים:

$$a = \Sigma^{-1} (\mu_1 - \mu_0)$$

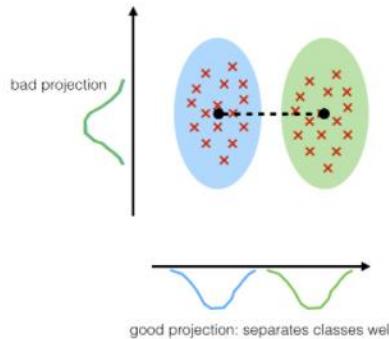
$$b = \frac{1}{2} (\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1) + \log \frac{p(y = 1)}{p(y = 0)}$$

והסיג היפוך

$$y = 1 \Leftrightarrow a^T x + b > 0$$

אלגוריתם QDA פשטוט יותר מאלגוריתם LDA-QDA שיש פחות פרמטרים לשערך, אך יש לו שני חסרונות עיקריים – הוא לא גמיש אלא לינארי, ובנוסף הוא מניע שטח co-variance בין ה-labels, מה שיכל לגרום לשיאו בסיג. כדי להתמודד עם הבעיה השניה ניתן להשתמש באלגוריתמים המנסים למצאו את מטריצת co-variance שתביא לפ██וג הטוב ביותר האפשרי (למשגיח Oracle Shrinkage Approximating Ledoit-Wolf estimator ו-).

באופן גרפי ניתן להסתכל על אלגוריתם LDA-QDA – במציאות כיוון ההפרדה יש את השונות הגדולה ביותר בין התפלגיות נורמליות ובונוס ישבבתה ההפרדה המקסימלית בין הקבוצות השונות לאחר מציאת הקוו האופטימי, ניתן לחשב את התפלגיות של הקבוצות השונות כהתפלגיות נורמליות על הישר המאונך לקו ההפרדה:



איור 2.5 אלגוריתם LDA באופן גרפי: מציאת הכיוון של התפלגיות והטלת המידע על הציר האנכי לכיוון ההפרדה.

2.1.5 Decision Trees

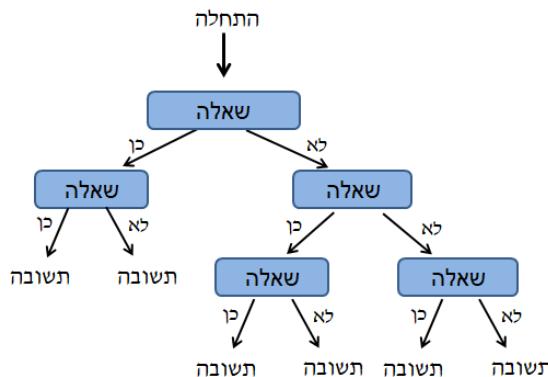
1. הקדמה

עדי החלטה הינו אלגוריתם לומד היכול לשמש הן לביעות סיווג והן לביעות רגסיה. באופן כללי, עדי ההחלטה לוקחים את המשתנה שברצוננו להסביר (משתנה המטריה/הхиורי), ומחלקים את המרחב לשלבלי ביצועים (segments). לכל קבוצה של סט האימון מחשבים "ממוצע" (Mean) או "שכיח" (Mode), וכאשר מתקבלת תצפית חדשה ממשיכים אתה לקבוצה בעלת הממוצע או השכיח הדומה ביותר – הסיווג לכל הesty של הקבוצות השונות יכול להציגו בצורה עצ, אלגוריתם זה נקרא "עדי החלטה".

הגישהות השונות בעדי החלטה פשוטות וឥטיות להבנה, אולם רק לא מצלחות להתחרות במדדי הדיקוק של מודלים אחרים של supervised learning. לכן, בפרקים הבאים נציג שיטותensemble בהקמת ביצועים בנייה של כמה עדי-ההחלטה במקביל, אשר משלבים בסופו של דבר למודול יחיד. ניתן להראות שימוש במספר גדול של עזיף

יכול לשפר דרמטית את מדדי הדיק של המודל. עם זאת, ככל שימושים נוספים ביותר עצים כך יכולת הפרשנות של המודל נהיית מורכבת יותר ופחות אינטואיטיבית לצופה שאין מעורב במבנה המודל (למשל גורם עסקי בארגון שאנו מונינים להסביר לו את תוצאות המודל).

ראשית נקבע מבנה בסיסי של עץ ונגיד עבורו את המושגים הרלוונטיים:



איור 2.6 דיאגרמה של עץ החלטה

על מנת להבין את מבנה העץ וליצור שפה משותפת נציג את השמות המקובלים בעבודה עם עצים:

- Root (שורש) – נקודת הכניסה לעץ (חלקו העליון ביותר של העץ).
- (צומת) – נקודת ההחלטה/פיקול של העץ – השאלות
- (עלים) – הקצות של העצים – התשובות. נקראים גם terminal nodes Leaves
- (ענף) – חלק מתוך העץ המלא (תת-עץ)
- (עומק) – מספר ה-nodes במסלול הארוך ביותר בעץ.
- Depth

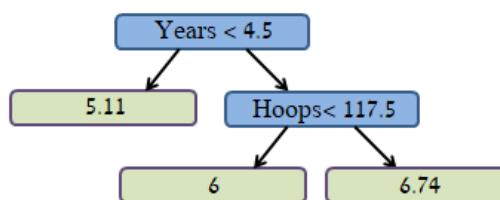
יסודות של עצי החלטה

עצים רגסיה:

כאמור, עצים החלטה יכולים לשמש הן עבור סיווג והן עבור רגסיה, ונתחייב עפ"ד גמאנט-פושטה לבניית עץ עבור בעייה רגסיה – חיזוי שכיר (באלפי שקלים) שחקקי כדורסל באמצעות עצי החלטה. נרצה לחזות את השכר של שחקקים בהתאם על הנתונים הבאים:

- שנים - מספר העונות שאוטו שחקק שייחס בילגט העץ
- סלים - מספר הסלים שהוא קלע בשנה הקודמת.

תחילה נסיר ציפיות ללא ערכי נתונים המוסבר לנו, הלא הוא "שכר" אבןוסף נבע על אותו משתנה-Log transform ב כדי שהוא ככל הנין בקירוב להתפלגות נורמלית (עקומת גאות/פעמון).



איור 2.7 דוגמא של עץ החלטה עבור בעיית רגסיה של עץ החלטה.

כאמור, האיזומטריא – עץ המנסה לחזות או לחשוף (באלפי שקלים) של שחקק בהינתן מספר עונות הותק שלו וכמות הסלים שהקלע בעונה הקודמת – ניתן לראות שהלן העץ (Root) מוחלך ל-2 ענפים. הענף השמאלי מתיחס לשחקנים בעלי ותק של יותר מ-4.5 שנים ($years < 4.5$), והענף הימני מתיחס לשחקנים פחות ותיקים – עץ יש-2 צמתים (=שאלותノード) ו-3 עלים (=תשבותノード). המספר בכל עלה שבתחלת העץ הוא הממוצע של כל התצפיות שסוגו לעלה זהה. לאחר שבניית העץ הסתיימה, כל תצפית חדשה

בסיום של דבר העז מחלק את השחקנים לשלוש קבוצות:
לבנות את העז וכייד לקבוע את כליל הפיצול בהתאם לדאטה הקיימת
שתסוג לעלה מסוים תקבל את הערך הממוצע של התוצאות של בסיסם נבנה העז-במהשך הפרק יסביר כיצד

- שחוקנים ששיכקו **4 עונות או פחות**:

$$R_1 = \{X | Years < 4.5, Hoops\} = 5.11$$

- שחנים ששיחקו נזונות או יותר וקלעו פחות מ- 18 קליע בעונה שחלפה:

$$R_2 = \{X | Years > 4.5, Hoops < 117.5\} = 6$$

- שחזורים שישיחקו עונות או יותר ושהלכו יותר מ-18 חודשים בעונה שלאפה:

$$R_3 = \{X | Years > 4.5, Hoops > 117.5\} = 6.74$$

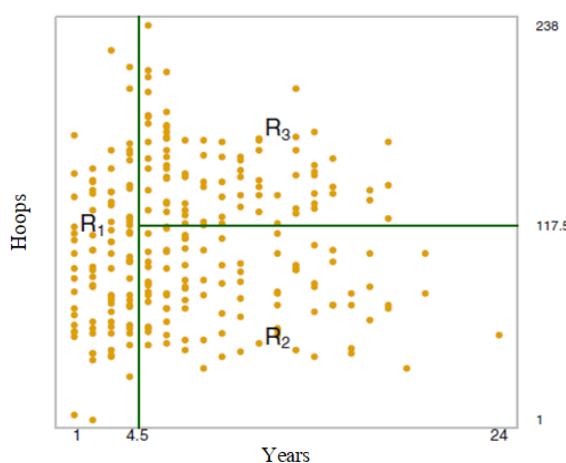
בהתאם לאנלוגיה של העץ, הקבוצות Terminal nodes, R_1, R_2, R_3 מכונות בשם "עלים" של העץ.

אפשר לפרש את עץ הרגstry שבודגמאותה נספחת `Years` הוא הפערטן המרכזי ביותר בקביעת מה יהיה גובה השכר, ושהקנים עם פחות שנות ניסיון ירוויחו פחות משחקנים מנוסים יותר=עבורי-שחקני חיסכית חדש (עד-ヶשניף בלבד) הפערטן של כמהות הסליםאותה קלע **בעונה** הקודמת מתברר כפחות משפייע על השכר. לעומת זאת, כל עוד בלגיה) הפערטן של כמהות הסליםאותה קלע בעונה הקודמת מתברר כפחות משפייע על השכר. לעומת זאת, כל עוד לשחקן איז-ヶשנונו-טנישין, כמהות הסליפ-פהיא יחסית שולית ביחס לחוסר הניסיון עבר קביעת שכור-עלמות זאת, בקרבת שחקנים ערך-5.4 שנות ניסיון או יותר, כמהות הסלים שהם קלעו בשנה הקודמת כן משפיעה על השכר-ושחקנים שקלעו הרבה סלים בשנה הקודמת נראתה ירוויחו יותר כסף מאשר שחקנים עם אותן ניסיון אך קלעו פחות סלים.

ע"ז הרגסיה שהובא בדוגמה מפשט-קצת "יתר על המידה"-את הקשר "האמתי" ב-Years=Hoops=Salary=3. והחיזוי של העץ לא מספיק טוב בגין מודלים אחרים לרגרסיה, כמו למשל-רגסיה לינארית-שתוסבר בפרק 3. עם זאת, מודל זה פשוט יותר להבנה ופרשנות וכל יחסית ל'פוג' באמצעות גורפים.

2. חיזוי באמצעות ריבוד (Stratification) של מרחב המשתנים:

עפ"ה הוסבר באופן אינטואיטיבי מהו עץ החלטה וכייד הוא פועל-האתגר המרכזי הוא לבנות את העץ בצורה טובות ככלשחיזוי שלו אכן יהיה תואם למציאות-בכדי להבין כיצד בונים עץ בצורה יعلا, נציג את הדוגמא-הקדמת באופן נסוף:



איור 2.8 דוגמא של עז החלטה עברו בעית רגסיה של עז החלטה.

באיור זה ניתן לראות שהנתונים נפרטו על פני מרחב דו ממדי, וחולקו באמצעות קווי החלטה בהתאם לnodes.icut גדרית את תהליכי החקולקה והחיזוי בשני שלבים

1. מחלקם את מרחב הערכים האפשרים של המשתנה אותו רצים לחזות בפ' איזורים נפרדים R_1, R_2, \dots, R_j . כתלות פפרמטרים השונים X_n, X_{n-1}, \dots, X_1
 2. לאחר החלוקה בפ' שונפלטבאות i מקבל את ערך המוצע של הנקודות מסוימות האימון שמרכזו בפ' את הקבוצות i .

באופקטיורטי, לכל אזור יכולה להיות כל צורה שהיא. אולם לטובת הפשטות, אנו בוחרים לחלק את המשטנה \hat{y} "אזורים-רוביים". המטרה היא למצוא את האזורי-שمبادאים לминימום את סכום ריבועי ההפרשיות של הנתונים בסט האימון לבן הערכים של כל אזור. מגד זה נקרא residual sum of squares (RSS), שמשמעותו $\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$

כאשר \hat{y}_{R_j} מציין ממוצע תצפיות האימון באזורי- j , y_i – ערך המרחק-ביקול תצפית בין הערך הממוצע באזורי-זה. כאמור, המטרה של מגד זה היא למצוא את מקבץ התצפיות בכל אזור, כך שריבוע המרחק בז' הערך הממוצע לבין כל תצפית יהיה מינימלי – כיוון שבדיקת כל החלוקות השונות האפשריות אינה-ריאלית מבחינה חישובית, לרבות משתמשים באלגוריתם חמד-גראדי (greedy recursive binary splitting) אשר עובד "מלמעלה למטה". גישׂה זו ביחס לעז החטלה נקראת "פיזול בירנרי רקורסיבי" (recursive binary splitting), והוא נקראת "מלמעלה למטה" כיוקשה אמתה מתחילה מראש העץ (root) – היכן שככל התצפיות עדין-שייכות לקבוצה אחת גדולה. לאחר סימון נקודת ההתחלה, האלגוריתם מפצל את מרחב-הערכים של המשטנה-אותו רוצים לחזות, כאשר כל פיזול מסומן באמצעות שני ענפים חדשניים בהמשך העץ.

האלגוריתם כאמור הוא חמד-גראדי מתבטה בכך שבעל בתהילן בנית העקבות-הרכים לבצע את הפיזול הטוב ביותר עבור השלב הנוכחי-מוביל להתחשב-בכמה שלבים קדימה. בגין זה יתכן ובערך הנוכחות-פיזול-יוטר עיל לטוויה ארוך, אך הוא לא יבחר אם הוא לא הפיזול האופטימלי בשלב זה-הניען להמחיש את דרך הפעולה של אלגוריתם חמדן לעמידה בפקק תנועה, ומתו-הנישן לעקוף את הפיק-נקודות-הפניה הראשונה שנראית פחות פוקוקה, מבליך להתחשב בפקק שעשו לבוא בהמשך-כמובן שפניה זו לא בהכרח תוביל בדרך יותר מהירה, ויתאפשר-בראייה יותר ארוכת טווח דזוקא היה מוטב להימנע מפניה זו כיוון שהיא מובילת לפיקק גדול יותר בהמשך.

כדי לבצע את אותו "פיזול בירנרי רקורסיבי", יש לבצע את התהליך האיטרטיבי הבא:

עבור כל פרמטר אפשרי X וקו חלוקה t , נקבל שתי קבוצות

$$R_1(j, s) = \{X | X_j < s\}$$

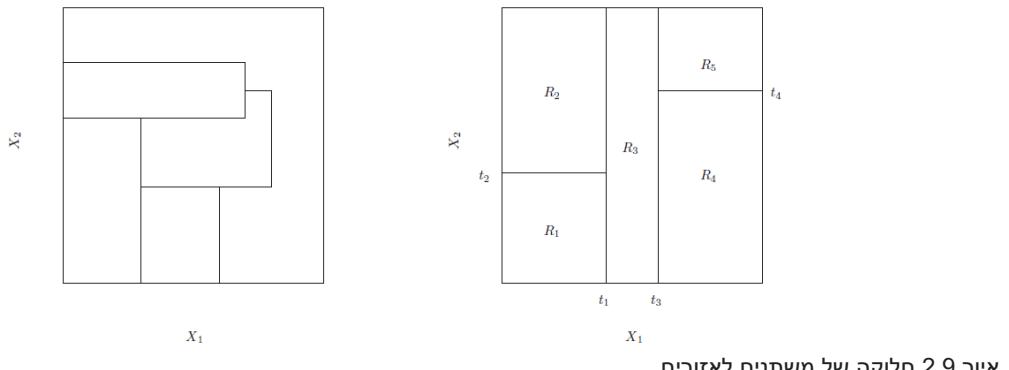
$$R_2(j, s) = \{X | X_j \geq s\}$$

עבור שתי קבוצות אלה נחפש את הערכים של j ו- s שմבאים למינימום את המשוואה הבאה

$$\sum_{i: x_i \in R_1(j, s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j, s)} (y_i - \hat{y}_{R_2})^2$$

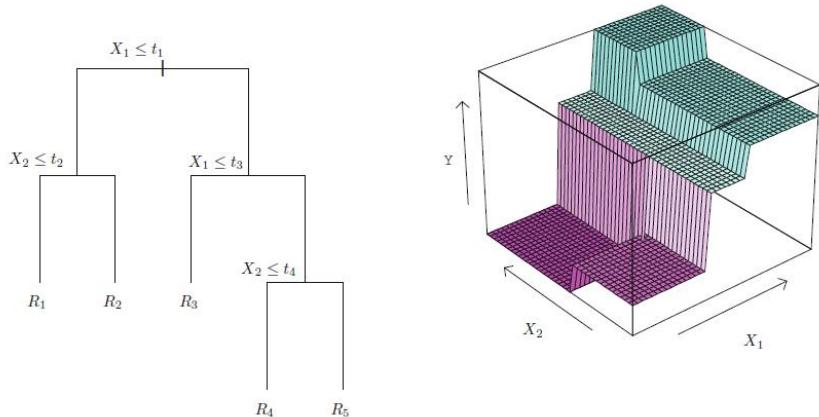
מציא-ערךס, שմבאים למינימום את המשוואה יכול להתבצע די בקלות כשמות הפרמטרים לא גדולות מיד – אך זה תהילך שעלול להיות די סבוך כשייש הרבה פרמטרים.

למעשה תהיליך זכמי-צחען אחד שיוצא מה-root, ולכך-עלים. כעת-מבצעים את התהליך שוב, מתח-מטרה למצוא את הפרמטר-הבסיסי לMINIMUM RSS – בכל קבוצה-ובכך לקובל-קבוצות. ובאופן דומה, נרצה לפצל את אחות מאוות-קבוצות שכבר יש לנו כדי למצוות עוד את RSS – התהליך הזה נמשך איטרטיבית עד שmagics "לקרייטרי" עזירה – מוסים, כמו למשל מספר פיצולים, מספר מקסימלי של תצפיות בכל אזור וכו' – אחריו שהתבצעה החלוקה לקבוצות R_1, \dots, R_J , ניתן להפוך את הקבוצות לעץ, ולהשתמש בו כדי לחזות נקודות חדשות.



איור 2.9 חלוקה של משתנים לאזוריים.

באיו-המוצרף ניתן לראותו שתי חלוקות – החלוקה הימנית הינה חלוקה דו-ממדית של שני משתנים באמצעות פיצול ביןארי רקורסיבי=החלוקה השמאלית היא גפחה דו-ממדית של שני משתנים, אך האילא יכלת להיווצר באמצעות פיצול ביןארי רקורסיבי, כיוון שהיא מורכבת מידי וaina תואמת את הגישה החמדנית שרצה "חלוקת פשוטה ומהירה"



איור 10. תיאור איזורי חלוקה בעץ ביןארי. מצד שמאל מוצג העץ התואם לחולקה מהאייר הקודם, מצד ימין ישנה פרטקטיביה רחבה כיצד חיזויים מתבצעים באמצעות העץ

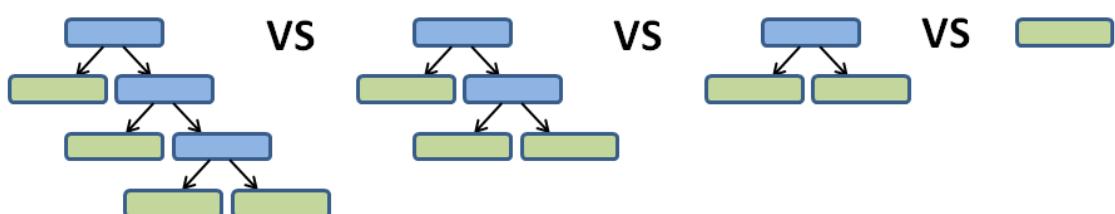
3. גיזום (pruning):

התהlik שתוארכשף את התהlik הקלאסי של יצירת עץ גרסיה, והוא מסוגל לנפקחיזים טובי^פ(מכחינת מודד RSS=ושונות נמכה). עם זאת העץ עשוי לבצע התאמת-יתר (Over-fitting) על הנתונים, מה שיוביל לביצועים לא טובים עבורה דאטת חדשנית או עלולה לנבוע מכשהעך שנבנה בתהlik-האימקעשי להיות "מורכב מידי" ומונאפה יתר על המידה לנדרי האימון, מה שפוגע ביכולת ההכללה שלו. לעומת זאת עץ מרכיב, עץ דיליל יותר בעל פחות פיצול יחס (לומר פחות קבוצות R_1, \dots, R_n)^ג יתזוויף בערך הטעיה (Bias) (Agdolia et al., 2018) יותר בהשוואה לאלטרנטיביה הראשונה, אך נראה יוביל לשונות קטנה יותר בין סט האימון לבין סט המבחן, ובנוסף מודל זה יהיה קל יותר לפרשנו.

גישה פשוטית בכך שהמודד עם בעיה זו היא לקביעךך כלשהו^ד שבל פיצול הירידה RSS^ה=תעלוקעךך גזע כלומר, פיצול שימושי לירידה שלoit יחסית RSS^ה=לא יבצע-באופן זהה העציפשיטיקבלו יהיו קטנים יותר ויש פחות חישש over-fitting. החיסרונו בגישה זונגנוקבךך שהיא קצרת-ראיה. תכוף מבזב על תרומה קטנה יחס-פאך הוא יכול להוביל לפיצול אחר בעל תרומה גדולה RSS^ה=זזה שביאלירידה גדולה RSS^ה=במהשך. שיטה זו תدل על אותו פיצול בעל ערך קטן, מאחר והוא לא עבר את הרף שהוגדר.

גישה אחרת, שמתברר והיא טובה יותר-הוילת בכיוון הפוך. בשלב הראשון יבנה עץ גודל מאוזן (T_0), ולאחר מכן נציגו ממנו כל מיינט פיצולים בצד להימנע over-fitting=את מקום של הענפי השתרנו יתפוז עליה בודד שמקבל ערך ממוצע המתחשב במספר גדול יותר של ציפוי-כםובן שצד זה יגרום לירידה בביטויים על פני סט האימון, אך השאייפה היא שזה ישתלם בבדיקה שהגיאת צטט המבחן

השאלה הגדולה היא כמובק מהי הדרכ הטובה ביותר לגוזם את העץ=אינטואיטיבית, המטרה שלנו היא לבחוח subtree בעקבות העץ המקורי שימושו לשגיאת הקטנה ביחס-אומדן זה יכול להתבצע על ידי בדיקת השגיאת העץ החדש ביחס לסט המבחן=כיוון שאפשר לבדוק את כלתתי העצים האפשריים-הנוגה להשתמש בגישה הנקראית subtree pruning=Cost complexity pruning (ידועה גם בשם weakest link pruning)=בגישה זעב-בקום להתחשב בכל subtree אפשרי, אנו משתמשים על רצף מסוים של subtrees למשתתת חלקים שונים המרכיבים את העץ=העץ המקורי שבנייה בהתחלה

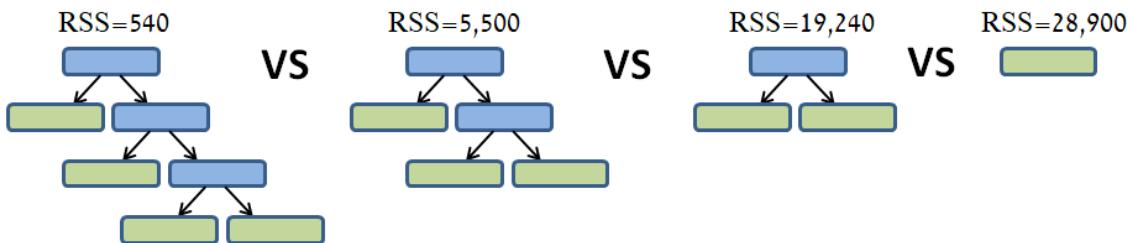


איור 11. חיפוש תת עץ אופטימלי ביחס ל- T_0 . מתחילהים מ- T_0 (העץ השמאלי) וכל פעם מורדים פיצול יחיד עד למגעים ל-root.

כעת מחשבים עbow=כל אחד מרכז העצימשה התקבל את RSS=שלו=לפנושסביה באיזה עז לבחור, נסכם את השלבים שביצעו עד כה

- א. בניית עץ רגסיה גודל כל הדאטה (אלא רק מסט האימון) בגישה recursive binary splitting (הגיisha החמדנית שתוארה לעיל)=העץ ימשיך להבנות עד קритריון עצירה מסוים (כמו למשל=הגעת למינום של ציפוי). עץ זה יסומן על T_0 = RSS
- ב. כל עלה בעץ מקבל את הערך הממוצע של ציפוי הpermuto-shabototo עלה, ועבור כל עלה פיצול מחושב ה-
- ג. סכימת כל ערכי RSS שקיבלו בכל העליים. הסכום שהתקבל הוא RSS של כל העץ T_0
- ד. כעפמבעציעת שלבי RSS-A-כל-subtree-הבא ברכף. זהו עקמנסה שכמעט זהה לעץ המקורי, למעט שני עלים שנגזרו מהעץ הקודם-כך חזרים על התהיל-בעבור כל subtree-h, עד ל- T_0 -האהחרון שמורכב רק מה-root של העץ המקורי

התוצר של השלב האחרון RSS-A-כל-subtree-הבא ברכף העצים=נitin להבחן כבכל פעם שענפים מסוימים הוסר-ה-Shallree שהתקבל נהיקגדל יותר לעומת subtree הקודם-הווצה זו הגיונית-השורי כפיזול במקורה נועד להקטין את השגיאה-באמצעות הקטנת ה RSS, ואילו גיזופ-הוואה את היפך-over-fitting גם במחיר של שגיאה גדולה יותר



איור 2.12 ביצוע גיזום וחישוב RSS לכל תת עץ (subtree) שהתקבל

ה. כעת יש לבחור אחד מה-subtrees, וכמו זה גישת pruning cost complexity גישה זו משקלת עבור כל תת עץ את מד RSS שלו יחד עם מספר העלים בעץ. באופן פורמלי:

$$\text{Tree score} = \text{RSS} + \alpha T$$

כאשר זהו מספר העלים בעץ (Terminal nodes) שהוא פרטט-רגולרי-ץ-הקובע את היחס בין מספר העלים לבין מד RSS פרטט-רגולרי-ץ-הקובע את היחס בין מספר העלים לבין מד RSS ה- α -המצערת trade-off-המצערת cross-validation גישת-pruning גישת-pruning Tree Complexity גישת-pruning over-fitting גישת-pruning Penality, וכאמור הוא נועד "לפצות" על ההפרש במספר העלים שבין subtrees השונים שנבחנים.

משלב ד' כבר קיבל RSS לכל subtree RSS, וכעת נשאר רק לחשב לכל אחד מהם את ה-tree score

נשים לב כי:

- o כאשר $\alpha = 0$, העץ המקורי (T_0) יהיה בהכרח בעל tree score הנמור ביוטר, כיוון שכאשר $\alpha = 0$ כל הרכיב αT בנוסחה שווה ל-0. במקרה זה ה-tree score נקבע $\text{Tree score} = \text{RSS}$

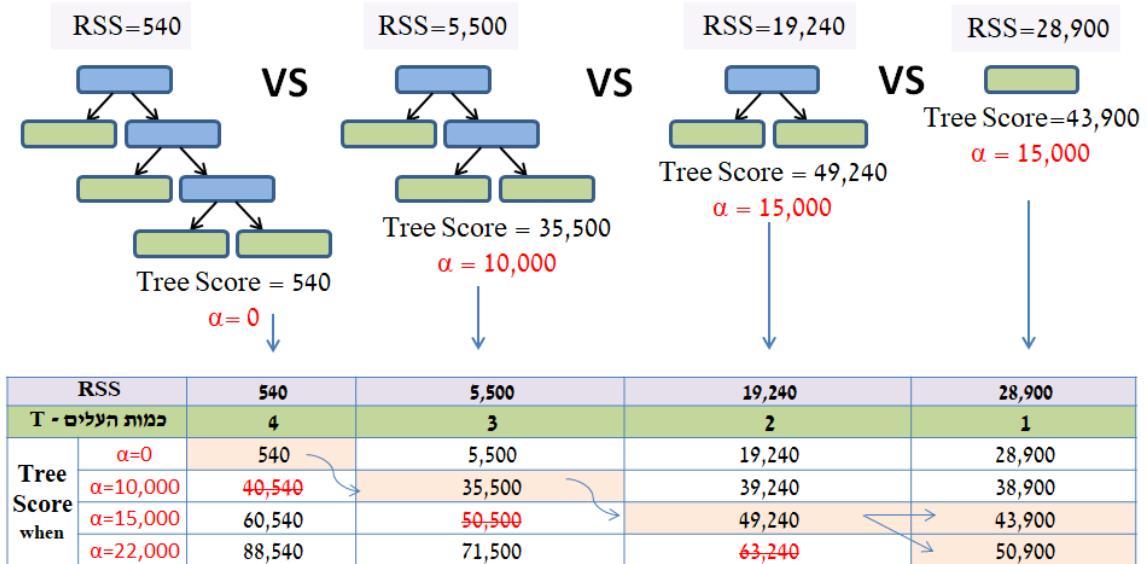
$$\text{Tree score} = \text{RSS} + 0 \cdot T = \text{RSS}$$

滿ילא שאר שלבי החישוב מיותרים, כי אנחנו כבר יודעים שהעץ T_0 הוא בעלה RSS=nmor ביחס לכל subtrees, וכן בעל tree score הנמור ביוטר. לכן, נרצה לקבוע $\alpha > 0$. נתבונן מה קורה עבור ערך α שונה

ו. עובי=0,000 ש- T_0 היה=40,540. ולכן שווה לנו לגוזף=עלים ולקבל עbor אוט-א-ץ-ין נמור יותר מאשר 40,540 (העץ השני ממשאל עם ציון ש- T_0 =35,500) ושוב, אם נשאר ב-10,000 = אוט-א-ץ-ין נגמר נגוזם 2=עלים נוספים (אנו כעת בעץ השלישי ממשאל), נקבל ציון ש- T_0 =39,240=שזה עדיין גבוהה יותר מאשר 35,500 ולכן זה לא טוב לא

עבור $\alpha = 15,000$ ניתן לראות שהציוון המתקבל יהיה נמוך יותר מה- subtree =הקדם. כעת נשים לב-
שבמעבר ל- subtree ($h=\text{root}$) לא משנה א- α -יגדר-א- α שאר אותו דבר, בכל מקרה הציוון של ה-
 root יהיה נמוך יותר מה- subtree הקודם.

למעשה חזרה על אופטימיזציה כדי שא- α -ות מלה-גוזם-ב- subtree =תהליך ערכיו-שונו-פ-ש-ל-א-תנו רצף של
עצים, מעז מלא ועד לעלה בודד. יוצא מכך שכל ערך ש- α קיימ- subtree ממתא- $T_0 \subset T$ כך זה-
score המתקבל יהיה קטן ככל האפשר.

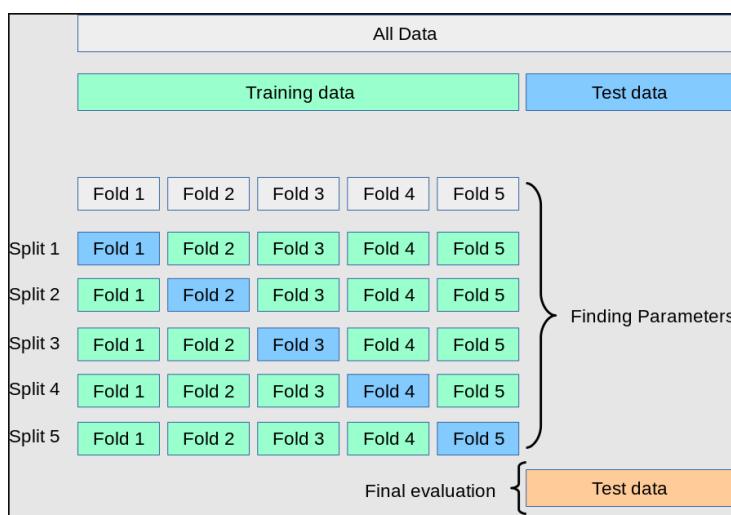


איור 2.13 חישוב ערך α מותאם לכל עץ משנה כך שיתקבל Tree score קיטן ככל האפשר.

ח. כתוב נבחר ב- subtree core-הנמור בiot- subtree =כלומר העץ השני משמאלי עם ציון של 35,500. במקביל יש לזכור מהם ערכי α האשל ה- subtrees השונים שהתקבל (ב- subtree ה-קדם), כיוון שהם יהיו שימושיים לחישוב ערך α אופטימלי:

- ערכי- α -חשובים, כיווק-של-כל- α -עשוי להתקבל עץ אחר בעל ציון מינימלי. עד לשלב זהה ערכ-גנינה ביחס לכל הדאט-ה (בלי חלוקה ל-Train=Test). א- α המטריה-יא-לבוחן מהו ערך α -ה-שייתן את התוצאה האופטימלית בעזרת העץ הגוזם-עbor דאט-ה חדש.

ט. כדי לבחור את ערך α האופטימלי ניתן להשתמש ב- Cross-Validation . לשם כך, יש לקחת את הדאט-ה המלא (ממנו נבנה העץ הראשון – T_0), ולחלקו-אות-באוף הבא:

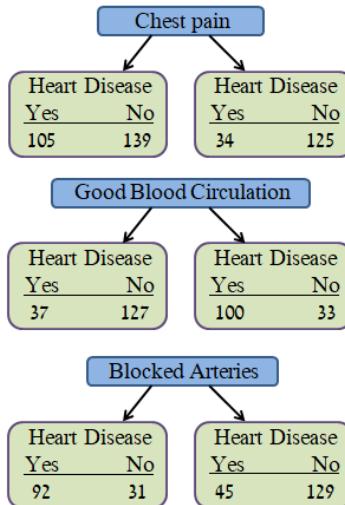


איור 2.14 $K_{fold} = 5$ ערך cross-validation 2.14

- ב-cross-validation המודל מתאים תחילת לפ-1 (או-1.4.2 לעיל) על התוצאות שבחלק'יך הירוקים, ובוחן את החיזויים על סט התוצאות הכלול.
 - התהילך זהה נעשה $K=$ פעמים, כאשר בכל איטרציה האימון נעשה את התוצאות שבחלק'יך הירוקים, ובוחינת החיזויים (וחישוב RSS) נעשה על התוצאות שבחלק'יך.
 - . בכל אחד מה-cross-validation המודל משתמש רק ב-training data כדי לבנות עץ מלא (נסמך אותו הפעם ב- T_1 ורץ) (sequence) חד-שטי- K -subtrees (sequence) לMINIMUM את RSS (אוטו סדר פעולות כמו בסעיפים א'-ד'), בעזרתו מרכז' α שהתקבלו בסעיף β .
 - א' כעת יש ליחס אפק- RSS =כלכל- K -subtree במאכזות שימוש ב-test data= T_2 =בלבד- K -subtree של RSS הנמור בioter. נניח שבאירציה הראשונה העץ שקיבל את RSS=nmor בioter ב- קיבל את RSS=nmor בioter. נניח שבאירציה הראשונה העץ שקיבל את RSS=nmor בioter ב- Testing data הוא דוקא העץ שב- $\alpha = 10,000$
 - ב' את התהילך של סעיפים י-יא יש לבצע $K=$ פעמי- α פעם אחת עבור- K -split= α כאשר בכל איטרציה האימון מtabצע על התוצאות שבחלק'יך הירוקים, ובוחינת החיזויים (וחישוב RSS) מtabצע על התוצאות שבחלק'יך הכלול.
 - ג' בסופו של התהילך, כל אירציה- K -subtrees RSS=משום, וכן ערכ- α -שנקבעו כבר מראש בסעיף ז' לאח- K -האירציות לבדוק מיהו העץ עם RSS=nmor מבין כל האיטרציות, ומהו ערך α העץ הזה, וזה יהיה הסופי של α .
 - ה' לבסוף יש לחזור לעץ המקור- T_0 זהה- K -subtrees שנבנו מ- K -data set מהמלא (שמכי- K -אפק- RSS =אגף את test) ולבוחור את העץ שמתאים לערך α ה�וזם הסופי- α שיבחר.
- לסיכום:**
- אח- K -ההיפוך שובי- α מצא- α השהעץ T_0 הוא בעל RSS=nmor בioter, אף-יתכן והוא יסבול מ-overfitTING.
 - כדי לפנות על ק- K -ונוס רכיב שנועד להתחשב גם ב- K -העוצם שהינה- K -בעל RSS=nmbah יוטר, ו- "מעניש" את העץ המקורי (T_0) עקב ריבוי העלים שבו.
 - באופן הרוח-התקבל ציוף-האפשרות לחשות בין העצים ולבחור את העץ הטוב ביותר בioter. העץ הזה מהווה מען איזון בין הרצן RSS=nminil לבין שונות נוכחה בין Train ל-Test .
 - ב כדי למצוא את RSS=nminil, שמצד אחד- K -האופטימלי, שמצד שני RSS=noptimal=ומצד שני נמנע כמה שניתן מ-cross-validation Overfitting ניתן להיעזר ב- K -mosion.

4. עץ סיאם

עץ סיאם ד' דומה לעץ רגרסיה, רק שהמטרה היא שונה – במקום לקבל תשובה כמותית כמו ברגרסיה, עץ סיאם יתקיים תוויה-label (label) לתוצאות המבוקשת. כאמור לעיל לעץ רגרסיה מספק ה- K -cross-validation מושג-הברחתהตาม לערך המוצע של תוצאות האימון ששייכות לאו- K -node terminal בעץ סיאם לעומת זאת כל תוצאות- K -כל קובוצה (class) בעל תווית משותפת. למשל, נניח ומונחים לסוג מסויל מסויים האם יש לו מחלת לב או לא. אנו יכולים לבנות עץ החלטה עץ בסיס מאפיינים של חולים שאובחנו בעבר ואנו יודעים להגיד מי מהם באמת חוליה לב ומ' לא, ועל בסיס העץ הזה להחליט עבור כל מטופל חדש האם הוא דומה במאפיינים שלו למטופלים שאובחנו בעבר חוליה לב או לא. כ- K שהתשובה שהעץ נותן היא לא "ערך מוגזע" כמו שראינו בעצ' רגרסיה, אלא פשוט החלטתו – "כן חוליה לב" או "לא חוליה לב" – מלבד החיזוי של התווית, עץ סיאם מספק גם יחסים ב- K -קובוצות השונות בקשר לתוצאות האימון שנoplים באותו אזור. נתבונן בדוגמא שתמחיש את העניין:



איור 2.15 השפעת פרמטרים שונים על הסיכוי לחולות במחלה ל^ב

באיר לעיל'נition לראות שלושה פרמטרים (sharpness) שבסקרה זה הם סימפטומים של מטופל=בעזרתם מנוטר-לטוג האם למטופל יש מחלת לבאו לא. כפי שניתן לראות אף אחד מהמשתנים אין יכול לענות על שאלת זו בפני עצמו, כיוק שבאף אחד מהעלים אין איחדות בתוצאות המשתנים כאלה, אשר איןם יכולים בפני עצמם לספק סיגוג מושלם, נקראים משתנים-לא הומוגניים (impure=לא-טהורים)=Azure. מכיוון שברוב המקרים כל המשתנים אינם הומוגניים, יש למצאו דרכו כיצד-לבחר באחד מהמשתנים להיות המשתנה-שבראש העץ (Root node)=כלומר, יש ליצר מdad הבוחן ומשווה את רמת ה- i impurity של כל משתנה. ישנו מספר מדדים, ונתקמקד בשני מהם – Entropy=Gini index

A. מדד=Gini index

נסמן את ההסתברות לשירות תווית מסוימת לקבוצה j ב-d, ונגידו:

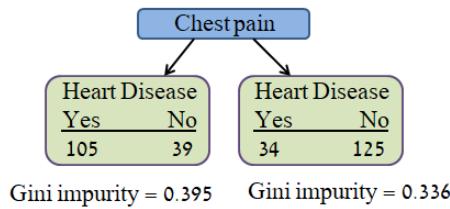
$$Gini = 1 - \sum_{j=1}^J p_j^2$$

באופן אינטואיטיבי מדד=Gini מיצג את הסיכוי לקבלת סיגוג שגוי עבור בחירה רנדומלית של נקודה מהדатаה בהתאם לפוטופורציות של קלאס=בדאות=digim זאת על אחד הפרמטרים שבודוג מאה קודמת=Chest pain. עבור הענף הימני מתקיים:

$$Gini = 1 - \left(\frac{34}{34 + 125} \right)^2 - \left(\frac{125}{34 + 125} \right)^2 = 0.336$$

עבור הענף השמאלי מתקיים:

$$Gini = 1 - \left(\frac{39}{39 + 105} \right)^2 - \left(\frac{105}{39 + 105} \right)^2 = 0.395$$



איור 2.16 חישוב מדד Gini עבור הפרמטר Chest pain

אחרי שיחסנו את מדד=Gini לשני העליים, נחשב את מדד=Gini הכלול של כל המשתנה=Chest Pain. בשביל חישוב זה יש לאזן בין מספר התוצאות בכל עלה, באופן הבא:

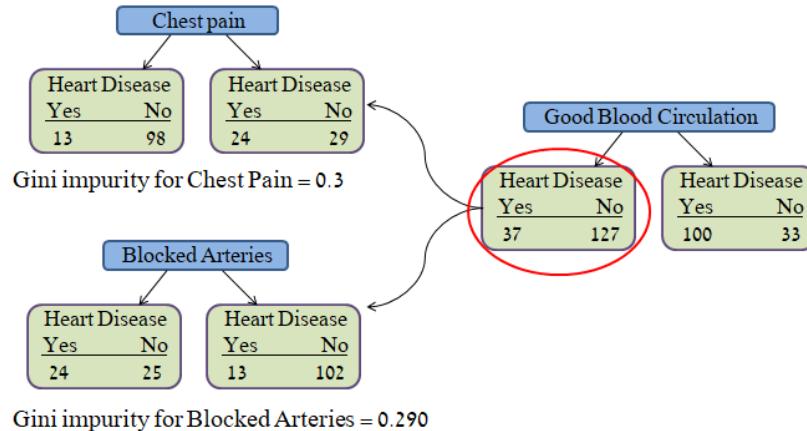
$$\text{Gini impurity for chest pain} = \left(\frac{144}{144 + 159} \right) \times 0.395 + \left(\frac{159}{144 + 159} \right) \times 0.336 = 0.364$$

באופן דומה ניתן לחשב את מzd Gini גם עבור יתר המשתנים ונקבע:

$$\text{Gini impurity for good blood circulation} = 0.36$$

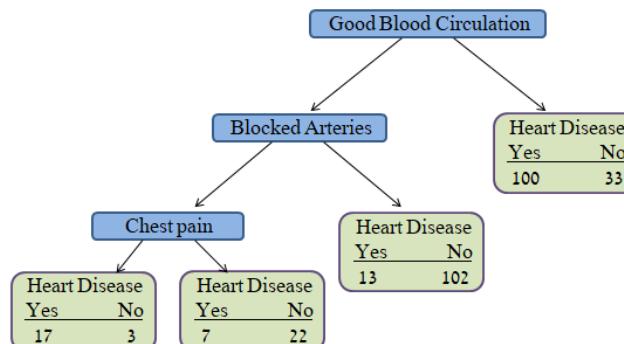
$$\text{Gini impurity for blocked arteries} = 0.381$$

למשתנה **Good blood Circulation** יש את הציון הכי נמוך, מה שהוא מסוג הכי טוב את המטופלים עם ובל' מחלת לב; ולכן נשתמש בו כמשתנה המסוג הראשון בראש העץ($=\text{root}$)=לקבוע את הפיצול הבא, יש להתבונן כיצד שאר הפרמטרים מסווגים את התצפויות של the root . נניח למשל ומתיק'ה:



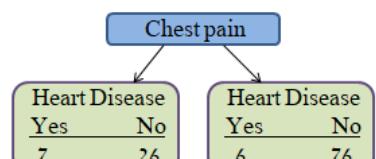
איור 2.17 חישוב מzd Gini עבור שאר הפרמטרים לאחר קביעת ה- root .

כפי שניתן לראות למשתנה **Blocked arteries** יש ציון נמוך יותר ולכן הוא זיהשנבחר להיות הפיצול הבא=
לבסוף נבחן כיצד הפרמטר האחרון מסביר את התצפויות של הפיצול שלפניו, ונקבל את העז הבא:



איור 2.18 חישוב מzd Gini עבור פיצול לפי הפרמטר **Chest pain** ביחס לשאר העץ

נשים לב שניתן לפצל גם את העלה $13/102$ =לפי הפרמטר **Chest pain**, באופן הבא (המספרים כמפורט תלויה בהתצפויות האמיתיות):



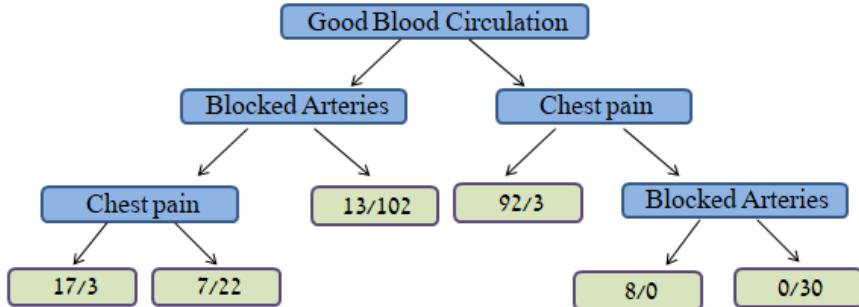
$$\text{Gini impurity for Chest Pain} = 0.29$$

איור 2.19 חישוב מzd Gini עבור פיצול לפי הפרמטר **Chest pain** ביחס לעלה $13/102$

מדד Gini שהתקבל הינו 0.29, בעוד שבלי הפיצול הממד של העלה היה 0.2, ולכן במקרה זה עדיף להשאיר אותו כפי שהוא לפני ניסיון הפיצול. לעומתו דומה לבני גם את הענף ימני של העץ

1. נחשב את מדד Gini.
2. אם לענף הקימ שצין Gini נמוך יותר, אז אין טעם לפצל עוד, והוא הופך להיות node terminal.
3. אם פיצול הענף הקימ מביא לשיפור, בוחרים את המשנה המפצל בעל ציון Gini הנמוך ביותר.

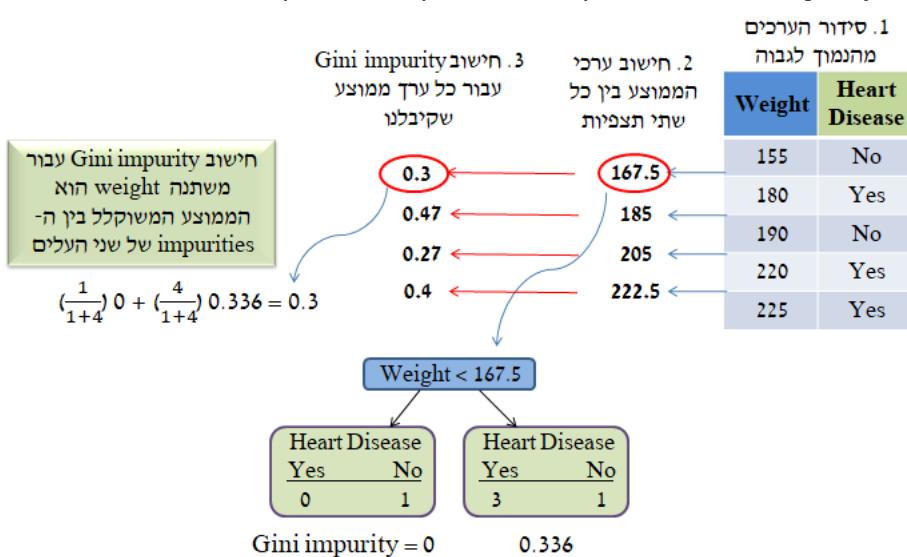
ע"ז טיפולו לאחר סיום התהילה נראה כך



איור 2.20 חישוב מדד Gini עבור פיצול לפי הפרמטר Chest pain ביחס לעלה 102/3

מדד Gini שהתקבל הינו 0.29, בעוד שבלי הפיצול הממד של העלה היה
התהיל שתוואר מטאים למצבים בהם הפרמטרים מתפללים באופןBINARI, כלומר הפיצול של כל פרמטר נקבע על ידי שאלה שעלה יש תשובה של כן או לא=במקרים בהפישנוףמשתנים רציפים, הפיצול דורש כמושלבים מקדים

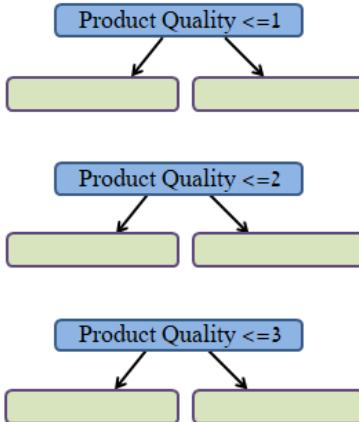
1. סידור ערכי הפרמטרים מהערך הנמוך ביותר לערך הגבוה ביותר
2. חישוב הערך הממוצע בין כל 2 תציפות
3. לחישוב Gini impurity עבור כל ערך ממוצע שהתקבל בשלב הקודם.



איור 2.21 פיצול פרמטרים רציפים לפי מדד Gini.

אנחנו מקבלים את ה-Gini הנמוך ביותר מתי שאנו חנו קבועים weight<205 של threshold. אז זהו ה-Gini שנשתמש בהם כשאנו מושווים את המשנה weight<weight של המשתנה השני עד עכשו דיברנו על איך מחלקים משתנה רציף ואיך מחלקים משתנה בינארי (שאלות כן/לא)=כעת בדבר איך מחלקים משתנים קטגוריאליים. למשל: משתנה מדורג שמקבל ערכים מ-1-4. למשל: טיב מוצר מ-1-4. או משתנה ש מכיל מספר קטגוריות. למשל: צבע מודעף (מכל ערכיהם: אדום, ירוק, כחול וכו'). משתנה מדורג מאוד דומה למשתנה רציף, חוץ מזה שאנו צריכים ליחס בו את האיזון עבור כל חלוקה אפשרית

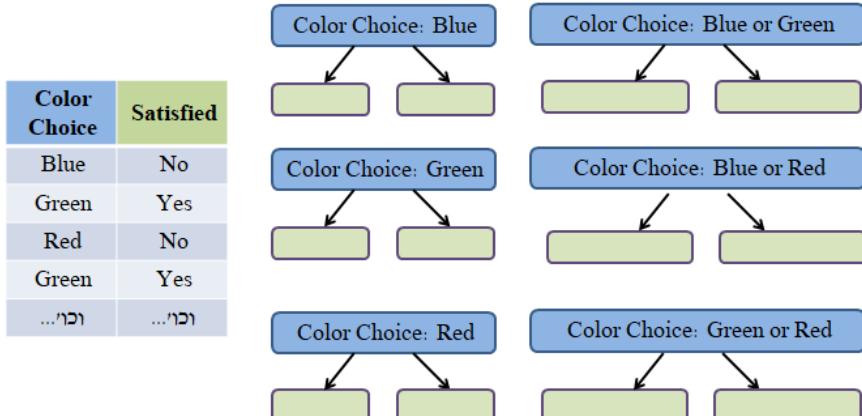
Product Quality	Satisfied
1	No
1	Yes
3	No
1	Yes
... וכו' וכו' ...



שימוש לב: אנחנו לא צריכים לחשב את ה- Gini על- Basis of impurity score. השה יכלול בעצם את כלם

איור 2.22 פיזול פרמטרים קטגוריאליים לפי מדד Gini.

כשיש משתנה קטגוריאלי עם כמה קטגוריות, אפשר לחשב את ציון ה- $\text{Gini} = \text{Gini}$ עבור כל קטgorיה, כמו גם עבור כל קומבינציה אפשרית.



איור 2.23 פיזול פרמטרים קטגוריאליים לפי מדד Gini.

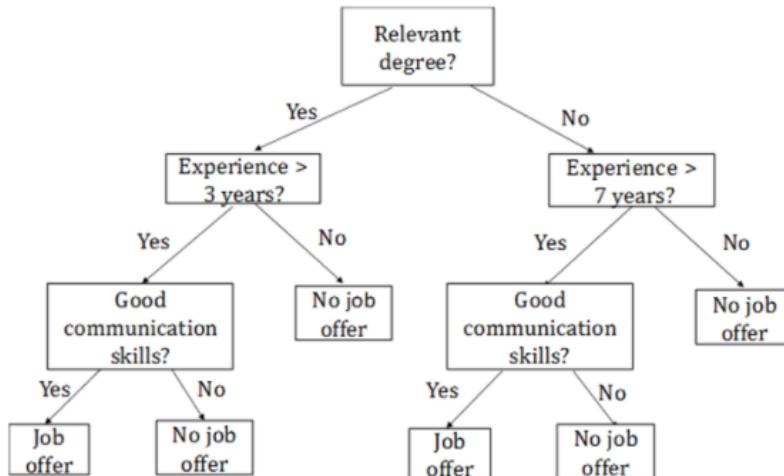
ב. מדד Entropy & Information Gain

מדד נוסף לפיזול צמתי העץ מtabסס על האנתרופיה של העלים, בעזרתה ניתן לבחוקאת ה- gain information gain המקיים מכל פיזול=אנתרופופיה באיה למדוד אופת השגיאה של התפלגות המשתנה הנבחן מול משתנה המטריה-נניח וישן n תוצאות אפשריות, כל אחת מהן בעלת הסתברות p_i , אז האנתרופיה מוגדרת באופן הבא

$$H(x) = - \sum_{i=1}^n p_i \log p_i$$

בדומה למדד Gini =הפייזול האופטימלי נבחר על ידה המשתנה בעקבות מדד האנתרופיה הנמוך ביותר. אם כל התוצאות בעלה מסוים מסוימות לאותה class, אז מדד האנתרופיה יהיה 0. מайдך, כאשר בעלה מסוים יש התפלגות שווה ב-2 class-ים של המשתנה המוסבר, מדד האנתרופיה יהיה 1 (זהה הערך המקיים שמדד אנתרופיה יכול לקבל).

לעילפירטנו בשלב אחריו שלב את חישוב מדד=Gini=Use-Case של חולוי לב.aset כעת ניקח דוגמא אחרת פשוטה יותר הנוגעת לקבלת מועד לתפקיד מסוים, כאשר מטרת העץ היא להחזיר "Yes" אם רוצים לקבל את המועד, אחרת "ochzr=No".



איור 2.23 עץ סיווג עבור קבלת מועדן לתקפיך מסוים

הדוגמא מוארת את אחד המאפיינים העיקריים של עצי החלטה מתקבלת על ידי הסתכילות היררכית על הפרמטרים השונים, כאשר בכל פעוף מתמקדים בפרמטר אחד ולא על כלם בבבב אחת. תחילה, נלקח בחשבון המאפיין החשוב ביותר (תואר רלוונטי במרקחה שלפנינו), לאחר מכן נלקחים שנות הפסיכון, ורק הלאה. נניח שהסיכוי בקרוב כלל העובדים להתקבל לעבודה הוא = 20%, והסתוכי לא להתקבל הוא = 80%. במרקחה זו האנטרופיה תוחשב באופן הבא (הלוגריתם יכול להיות מחושב בכל בסיס, פה נלקח הבסיס הטבעי)

$$H = -0.2 \ln 0.2 - 0.8 \ln 0.8 = 0.5004$$

כעת נניח בנוסחה -30% מהמועמדים בעלי תואר רלוונטי מקבלים הצעת עבודה והואו מtower אלה שאינם בעלי תואר רלוונטי, רק 10% מקבלים הצעת עבודה. האנטרופיה עבור מועדן בעל תואר הינה:

$$H = -0.3 \ln 0.3 - 0.7 \ln 0.7 = 0.61$$

ועבר מועדן ללא תואר רלוונטי לתמונה:

$$H = -0.1 \ln 0.1 - 0.9 \ln 0.9 = 0.32$$

כニיה-המועמדים מתפלגים באופן שווה בין בעלי תואר לכלה שאינם בעלי תואר, כולם $\frac{1}{2} = 50\%$ - מהמועמדים יישתוาร רלוונטי ול- $\frac{1}{2} = 50\%$ - אין, הרו' שתוחלת האנטרופיה במרקחה זה הינה:

$$\mathbb{E}_{H(x)} = 0.5 \times 0.61 + 0.5 \times 0.32 = 0.46$$

לאחר כל החישובים, נוכל לבחון את רמת **the-happiness** שמתיקבלת מהידיעה האם למועדן מסויים יש תואר רלוונטי או לא. כולם, כמה הידיעה שלמועדן מסויים יש תואר רלוונטי מפחיתה מוחסן ה odds של לקבל את המשרה. אם אי ה odds נמדדת באמצעות מדד Entropy, הרו' שהרווח מהמידע הוא

$$\text{Information gain} = 0.5004 - 0.4680 = 0.0324$$

הן עבו-מועמדים בעלי תואר והן עבו-כללה ללא תואר-המשתנה-shmakksum את הרווח מהמידע הצפוי (ירידת האנטרופיה הצפוי) הוא מס' שנות הניסיון. כאשר למועדן יש תואר רלוונטי, הס' עbow' שנות ניסיון-הממקפה את הרווח מהמידע הצפוי הוא $= 7$ שנים. לעומת זאת, אם התואם למועדן-החסך של שנות ניסיון-הממקפה את הרווח מהמידע הצפוי הוא < 7 שנים. לפיכך, שני הענפים הבאים הם: " $\text{ניסיוק} > 7$ " ו-" $\text{ניסיוק} \leq 7$ ". באותו האופן בונים את יתר העץ

Misclassification rate

לאחר בניית העץ, יש לבדוק את רמת הדיווק שלו על דата חדש. בעץ רגסיה זה נעשה בעזרת מדדי RSS ובעיות סיווג מקובל למדוד **Accuracy** Misclassification rate. מדד זה בא לכמת את היחס בין כמות התוצאות שהמודול סיווג באופן שגוי לבין כמות ההצלחות של התוצאות. במרקחה זה פונקציית המבחן תהיה

$$\mathcal{L}(\tilde{y}, y) = I\{\tilde{y} \neq y\}$$

פונקציית המבחן-zero-one loss, כאשר תחת פונקציה זו חישויים נכונים יקבלו ציוק וושגיאות יקבלו ציוק, ולא תלות בגודל השגיאה. פונקציית המרבייה misclassification rate נראית כך:

$$R(h) = \mathbb{E}[I\{h(x) \neq y\}]$$

סיכום

עż החלטה (Decision Tree) הינו אלגוריתם לשילוב אַלְחִיזָיָן ערכו של משתנה, כאשר המאפיינים מסודרים לפי סדר החשיבות. לצורך סיווג, קיימים שני מדרדים אלטרנטיביים לאז וודאות: מדרד אנטרופיה (Entropy) ומדרג ג'ינְטְּוֹנְטְּ (Gini). כאשר ערכו של משתנה מסוים נחזה, אי הוודאות נמדד באמצעות RSS=חסיבתו של מאפיין הינו הרוח מההידוע הצפוי שלְכָךְ (Expected Information Gain) =הרווח מההידוע הצפוי נמדד על ידי הירידה באז הוודאות הצפוי אשר יתרחש כאשר יתקבל מידע אודות המאפיין.

במקרה של חלוקה משתנה-קטgorיאלי, המידע המתkeletal הינו על פי רוב אודות קטגוריה (Label) של המשתנה למשול: צבע מועדף (מכיל ערכיון: אדום, ירוק, כחול וכו'). במקרה של משתנה-רציף לקבוע ערך סף (Threshold) אחד (או יותר) המגדיר שני טווחים (או יותר) עבור ערכיה המשתנה. ערכי סף אלו נקבעים באמצעות מוקסם אפקט gain information הצפוי.

אלגוריתם עż ההחלטה קובע תחיליה את צומת השורש (Root Node) האופטימלי של העץ באמצעות קרייטריון "מרקזום הרוח מההידוע" שהוגדר לעיל. לאחר מכן הוא ממשיך לעשות אותו הדבר עבור הצמתים הבאים. הקצוות של הענפים הסופיים של העץ מכונים צמתים עלים (Leaf Nodes).

במקרים בהפעζ ההחלטה משמש לשילוג, צמות העלים כוללי-פְּבָטוּכְּפָאָת ההסתברויות של כל אחת מהקטגוריות להיות הקטgorיה הנכונה. כאשר עż ההחלטה משמש לחיזוי ערך נומר-עלומת זאת, אז צמות העלים מספקים את ערך התוחלת של היעד. הגיאומטריה של העץ נקבעת באמצעות סט האימוק (Training Set) – אחר הסטטיסטי קיוק שweiskt ברמת הדיק של העץ צrica כמו תמיד בלמידת מכונה לבוא מתוך סט הבדיקה (Test Set) ולאחר מכן מटוק סט האימון.

אחרית דבר:

מדרג RSS ומדרג misclassification rate בפרק זה הינם רק חלק מהמדרדים המקובלים. לכל מדרד יתרונות וחסרונות, והמדד הרלוונטי יבחר בהתאם לסוג הנתונים והבעיה העסקית. נדון מעקבות ובחולשות של עץ ההחלטה.

יתרונות:

- בהשוואה לאלגוריתמים אחרים, עץ ההחלטה דרישים פחות השקעה בתהילך הכנת הנתונים (pre-processing).
- עץ ההחלטה לא דורש גרמול של הדאטא.
- ערכים חסרים בדתאה לא משפיעים על תהליך בניית העץ.
- עץ ההחלטה-תואם לאופק שבסביבה מרבית הבנייה א-דוחש-ב-עקב-יער-מסויימת והאפשרות להסביר מושגיה מומחיה.
- אין שום דרישת שיקשרות בין המשתנה המוסבר והמשתנים המסבירים יהיה לינארית.
- העץ בוחח אוטומטית במשתנים הטובים ביותר על מנת לבצע את החיזוי.
- עץ ההחלטה רגיש פחות לנסיבות חריגות מאשר רגסיה.

חסרונות:

- שינוי קטן בנתונים יכול לגרום לשינוי גדול במבנה עץ ההחלטה ולגרום לחוסר יציבות.
- לעיתים החישוב בעץ ההחלטה יכול להיות מורכב מאוד ביחס לאלגוריתמים אחרים.
- עץ ההחלטה לעתים תכופות דרישים יותר זמן הרצה לאימון המודל, ומשכך מדובר באלגוריתם "יקר" במשאביהם.
- האלגוריתם של עץ ההחלטה אינו מספיק ליישום רגסיה ולניבוי ערכים רציפים.
- עץ ההחלטה נוטה לעתים תכופות לנוטה Overfitting.

2.2 Unsupervised Learning Algorithms

2.2.1 K-means

אלגוריתם K-means הוא אלגוריתם של למידה א-מנוחית, בו מtabצעת תחזית על נתונים כאשר ה-label לאינו נתון. אלגוריתם זה מטאים לבוות של חלוקה לאשכולות(Clustering), ובנוסף יכול לשמש בשלב הצגת וניקוי הנתוני (EDA). עבור כל נקודה במדגם, המודל מזעיר את סכום ריבוע המרחקים (WCSS) מכל מרכז אשכול (סנטרואיד centroid), ולאחר תהליך של התכנסות – נקבעים האשכולות והסנטרואידים הסופיים. מספר האשכולות הנדרש הוא היפר-פרמטר שנקבע מראש. כמו כן האלגוריתם השיכים למידה הבלתי-מנוחית, ב-K-means לא מtabצע אימון, ולמעשה התחזית מtabצעת על כל הדאטה הנתון.

סנטרואיד הוא מונח מתחום הgiometria, והוא מתאר את הממוצע האריתמטי של כל הנקודות שמתפרשות על פנים צורה כלשהי – באופן אינטואיטיבי ניתן לחשב ערך סנטרואיד-כנקודות איזון של צורה גיאומטרית כלשהיא, כך שאם ננסח להניח צורה, משולש לדוגמה, באופן מסוין, אז הסנטרואיד הוא הנקודה שבה המשולש יתאזן ולא יפול לאחד הצדדים. בפועל, סביר שהוצאות איתן מתמודדים במצב צזה, במצב צפה, במצב צפה. במקרה, הסנטרואיד יהיה הנקודה בה סכום המרחקים של כל נקודה באשכול מהסנטרואיד יהיה מינימלי. כמובן, המודל ימוך את מרכזו של כל אשכול כפ-means: אלגוריתם מבוסס סנטרואידים המזעיר את סכום ריבוע המרחקים – כל הנקודות באשכול. מدد זה נקרא WCSS, והוא ממד משמעותי ביחס לקרוב אלגוריתמים שמציעים חלוקה לאשכולות, K-means. הסיבה לחזקה במשווה היא שאנו רוצים להגבר את ההשפעה של המרחק, מעין "עונש" לתוצאות רחוקות מהמרכז.

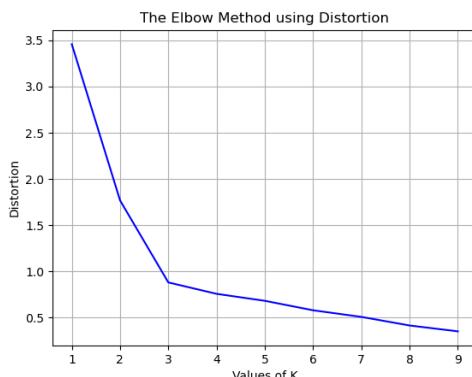
מדד WCSS הוא אחד הדריכים המקבילים ביוטר להעיר את תוצאות החלוקת לאשכולות – K-means – היתרונות שמדד זה הוא האפשרות לראות באופן ממוצע את מידת ההצלה של המודל, ככל שהוא משלב מספר ממש שמאז את החלוקת המודל – מנגנון WCSS – הוא מסוף ללא תחום מסוים והוא מושג באמצעות חישוב תוצאה רעה מודול – ערך מסוים יכול להיחשב תוצאה טוביה במקורה מסוים, ובמקרהacha את הבחירה להיחס תוצאה רעה מאוד. ניתן להשוות WCSS בין מודלים אחד ורך כאשר יש להם את אותו מספר אשכולות ואיתו מספר תוצאות – באופן פורמלי, ערך זה מחושב באופן הבא:

$$WCSS = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

כאשר K הוא מספר האשכולות, ו- n הוא מספר הנקודות במדגם.

ישנו – בין השאר – ממדד ה-WCSS – ממדד trade-off – השואף למזער את מספר האשכולות – גבול יותר, כך ה- WCSS יקטן. הדבר מתיישב עם הריגון – פיזור סנטרואידים רבים (כלומר, חילקה ליותר אשכולות) על פני הנתונים יוביל לכך שבhartoon סכום המרחקים של התוצאות מהסentrואידים יקטן או לא ישנה. כיוון שתוצאות מושיכת לסתטרואיד הקרוב אליה ביוטר, אם התווסף – סentrואיד שקרוב לנקודה מסוימת – ה-WCSS – קטן. ואם הסentrואיד רחוק מכל שאר הנקודות במדגם יותר מהסentrואידים הקיימים – חילקת התוצאות לאשכולות לא תשנה. וערך ה-WCSS לא ישנה.

לכך נרצה לבחור K גדול שימזע את ה-WCSS; מצד שני, הסיבה שהשתמשנו ב-K-means – מלכתחילה היא בכך לפשט את הנתונים למספר סביר של אשכולות, כזה שיאפשר לנו לעורך אנליזה נוכה. שיטת המפרק (Elbow method) היא טכניקה שימושת לפתרון סוגיה זו. הרעיון הוא לבחור את ה- K -הקטן ביותר שסמן השיפוע במדד ה- WCSS הוא מטען במידה סבירה. שיטה זו היא היוריסטיית ואינו דרך חד שמעית לקבוע שה- K הנבחר הוא האופטימלי. בדרך זו ניתן לשכנע מדוע ה- K -הבחירה הוא הנכון, אך ההחלטה הסופית נתונה לשיקול דעתו ש- K המשתמש:



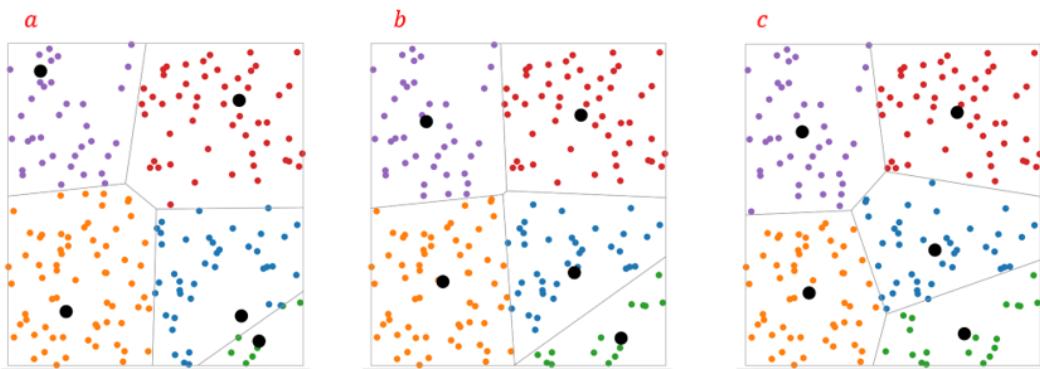
כאמור – האלגוריתם מחלק את הנתונים לאשכולות בדרך שמצוירת את סדרי-ובע-המරחקים של כל תצפית ממוקץ האשכול. באופן פורמלי האלגוריתם מתבצע **ב-4 שלבים**:

- א. אתחול: המודל מציב את הסנטרואידים באופן רנדומלי.

ב. שיור: כל תצפית משוויכת לסנטרואיד הקרוב אליו ביותר

ג. עדכון: הסנטרואיד מוזז שכן שה-WCSS של המודל ימוצע.

ד. חזרה על שלבים ב, ג עד אשר הסנטרואידים לא צדדים לאחר העדכון, כלומר יש התכנסות



א) $a=K$ -means ב) אטחון ק-סנטרואידים (K-Means Clustering) – שיכל נקודה לסentrואיד הקרוב ביותר אליו. c) חזרה על b עד להתקנות= לפि ממד ה- $c=WCSS$

K-means קידוע בקר שהוא אלגוריתם פשוט ומהיר. לרוב, הבחירה הראשונה בפתרון בעיות של חלוקה לאשכולות תהיה ב-K-means-עם זאת, לאלגוריתם ישנו גם חסרונות. ראשית, בחירת-

בעה נוספת להתעורר בהבירות המיקומית הראשו ש-**הסנטרואידי**-means מציין השבחירה היא רנדומלית, ניתן להיקלע לתוכנו-במיניהם מקום-שהוא אינו המינימום הגלובלי. כדי להתמודד עם בעיה זו ניתן לשתמש באлогוריתם++K+B של ראש האלגוריתם בוחר למקם סנטרואידי אחד באופן רנדומלי. לכל תצפית, האלגוריתם מחשב את המרחק בין התצפית לסנטרואיד הקרוב אליה ביותר. לאחר מכן, תצפית רנדומלית נבחרת להיות הסנטרואיד החדש. התצפית נבחרת בהתאם להטפלות משוקلات של המרחקים, כך שיכל שתצפית יותר רחוקה-**means** גובר הסיכוי שהיא תבחר. שני השלבים האחרונים נמשכים עד ש-**הסentrואידי**-means יתגשם. כאשר כל הסentrואידי-means מוקמו, מבצעים-**means** מילוג על שלב האתחול (שלב בו ממקמים אופתאים סentrואידיים)-++K+B מוביל לתוכנו מהירה יותר. ומוריד את הסיכוי להטפלות לאופתאים מוקמו.

2.2.2 Mixture Models

אלגוריתם K-means מחלק נקודות ל-K-קבוצות על פמරחק של כל נקודה ממוקד מסוים=בדומה ל-K-means-algorithm, it divides points into K-clusters based on distance from a central point. K-means-algorithm is a mixture model clustering algorithm that finds K clusters by iteratively calculating the mean of each cluster and assigning points to the closest cluster center. The algorithm starts with initial cluster centers and iterates until convergence. It is sensitive to outliers and can be slow for large datasets.

ניתן לדעת על איזה דוגמאות לנסות ולמצוא התפלגות מסוימת? עקב בעיה זו, לעיתים משתמשים קודם באלגוריתם K-means על מנת לבצע חלוקה ראשונית לקבוצות, ולאחר מכן למצוא לכל קבוצה של נקודות התפלגות מסוימת

ראשית נניח שיש a אשכולות, אז נוכל לרשום את ההסתברות לכל אשכול

$$p(y = i) = \alpha_i, i = 1, \dots, k$$

וכמוון לפי חוק ההסתברות השלמה מתקיים $\sum_i \alpha_i = 1$

בנוסף נניח שכל אשכול מתפלג נורמלית עם פרמטרים (μ_i, σ_i) , אז נקודה השויכת לאשכול i מקיימת:

$$x|y = i \sim \mathcal{N}(\mu_i, \sigma_i), i = 1, \dots, k$$

אם מגיעה נקודה חדשה ורוצים לשער אותה לאחד האשכולות, אז צריך למשהו למצוא את האשכול שבעורו הביטוי $p(x|y = i)$ הוא הכי גדול. לפי חוק ביאר מתקיים:

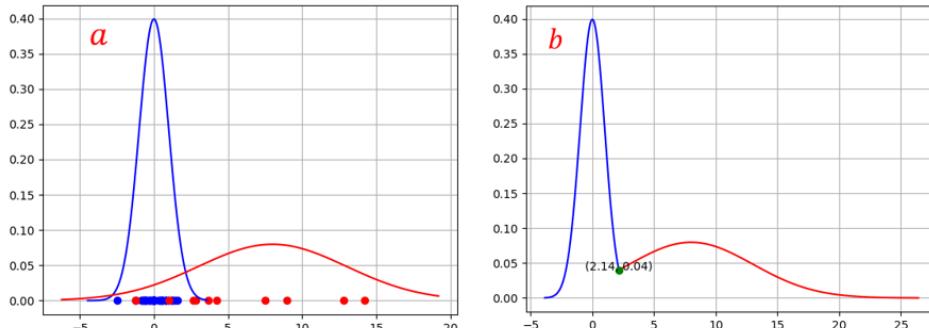
$$p(y = i|x) = \frac{p(y = i) \cdot p(x|y = i)}{p(x)}$$

המכנה למשהו נתון, כיוון שההתפלגות של כל אשכול ידועה ונותר לחשב את המכנה:

$$f(x) = f(x; \theta) = \sum_i p(y = i) f(x|y = i) = \sum_i \alpha_i \mathcal{N}(x; \mu_i, \sigma_i)$$

ובסך הכל

$$p(y = i|x) = \frac{\alpha_i \cdot \mathcal{N}(x; \mu_i, \sigma_i)}{\sum_j \alpha_j \mathcal{N}(x; \mu_j, \sigma_j)}$$

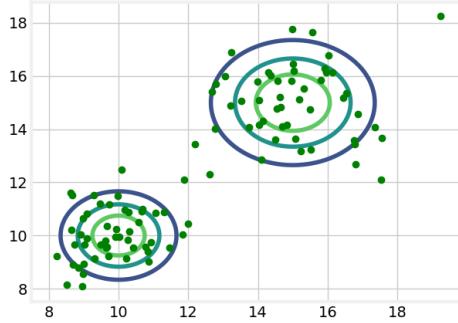


אנו- \mathcal{N} -העובי של שני גausians בבלב ראשון מחלקם את הנקודות לשני אשכולות מתאימים לכל אשכול התפלגות מסוימת. במקרה זה אשכול אחד (מוסמן בכחול) הותאם להתפלגות $\mathcal{N}(0, 1)$, ואשכול אחד (מוסמן באדום) הותאם להתפלגות $\mathcal{N}(8.5, 1)$. (b) נקודה חדשה x תסוג לאשכול הכחול אם $p(x|y = 1) > p(x|y = 2)$. באופן דומה, הנקודה x תסוג לאשכול האדום אם $p(x|y = 2) > p(x|y = 1)$.

כאמור, כדי לשער נקודה חדשה לאחד מה气colonות, יש לבדוק את ערך ההתפלגות בנקודה החדש. ההתפלגות שעוברת ההסתברות (x) קהיא הגדולה ביותר, היא זאת שאליה תהיה משוכנת הנקודה. ההתפלגות יכולות להיות ביחס-ממד, אך הן יכולות להיות גם בממד-יותר גובה. למשל אם מסתכלים על מישור, ניתן להתאים לכל אשכול ההתפלגות נורמלית דו-ממדית. במקרה ה- n ממד', ההתפלגות נורמלית $(\Sigma, \mu) \sim \mathcal{N}(\Sigma, \mu)$ היא בעלת הצפיפות:

$$f_X(x_1, \dots, x_n) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)}$$

Σ הוא הדטרמיננטה של מטריצת ה-covariance



אינטראקציית covariance של שני גaussians בדו-ממד=אשכול אחד מתאים לגaussians עם וקטור תוחולות $\mu_1 = [10, 10]$ וקוטר תוחולות $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, והאשכול השני מתאים לגaussians עם וקטור תוחולות $\mu_2 = [15, 15]$ ומטריצת covariance $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$.

כיוון שהאלגוריתם mixture model מספק התפלגיות, ניתן להשתמש בו כמודל גנרטיבי, כלומר מודל שיעד לייצר דוגמאות חדשות. לאחר התאמת התפלגות לכל אשכול, ניתן לדגום מההתפלגיות השונות ובכך לקבל דוגמאות חדשות.

2.2.3 Expectation–maximization (EM)

אלגוריתם מיקסום התוחלת הינו שיטות איטרטיבי למציאת הפרמטרים האופטימליים של התפלגיות שונות, במקרים בהם אין נוסחה סגורה למציאת הפרמטרים. נתבונן על מקרה של מיקра Mixture of Gaussians, ונניח שיש אשכול מסוים המתפלג נורמלית עם תוחלת ו纷離度 (μ, σ^2) כדי לחשב את התפלגות אשכול זה ניתן להשתמש בלוג הנראות המרבית:

$$L(\theta|x_1, \dots, x_n) = \log \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} = \sum_{i=1}^n \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(x_i - \mu)^2}{2\sigma^2}$$

כדי למצוא את הפרמטרים האופטימליים ניתן לגזר ולהשוות ל-0:

$$\frac{\partial L(\theta)}{\partial \mu} = \sum_{i=1}^n \frac{x_i - \mu}{\sigma^2} \rightarrow \mu_{MLE} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\frac{\partial L(\theta)}{\partial \sigma^2} = \frac{1}{2\sigma^2} \left(-n + \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 \right) \rightarrow \sigma_{MLE}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

cutet נניח ויש אשכולות וכל אחד מתפלג נורמלית. cutet סט הפרמטרים אותם צריך להעריך הינה

$$\theta = \{\mu_1, \dots, \mu_k, \sigma_1^2, \dots, \sigma_k^2, \alpha_1, \dots, \alpha_k\}$$

עבור מקרה זה, הלוג של פונקציית הנראות המרבית יהיה

$$L(\theta|x_1, \dots, x_n) = \log \prod_{i=1}^n \sum_{j=1}^k \alpha_j \mathcal{N}(x_i, \mu_j, \sigma_j^2) = \sum_{i=1}^n \log \left(\sum_{j=1}^k \alpha_j \mathcal{N}(x_i, \mu_j, \sigma_j^2) \right)$$

אם נגזר ונשווה ל-0 נקבל בדומה למקרה הפשו

$$\sum_{i=1}^n \frac{1}{\sum_{j=1}^k \alpha_j \mathcal{N}(x_i, \mu_j, \sigma_j^2)} \alpha_j \mathcal{N}(x_i, \mu_j, \sigma_j^2) \frac{(x_i - \mu_j)}{\sigma_j^2} = 0$$

נוסחה זו אינה ניתנת לפתרון אנליטי, ולכן יש הכרח למצוא דרך אחורובכדי לחשב את הפרמטרים האופטימליים של התפלגיות הרצויות. נתבונן בחלוקת מהביתי שקיבילנו:

$$\frac{1}{\sum_{j=1}^k \alpha_j \mathcal{N}(x_i, \mu_j, \sigma_j^2)} \alpha_j \mathcal{N}(x_i, \mu_j, \sigma_j^2) = \frac{p(y_i = j) \cdot p(x_i | y = j)}{p(x_i)} = p(y_i = j | x_i) \equiv w_{ij}$$

קייבלו למשה את הפואטורי $\hat{Q}(\theta)$ האשכול אליו רוצים לשיר אות x_i , אך הוא לא נתן אלא הוא חביב-כדי לחשב את המבוקש-ננחיש שערך התחלתי θ_0 ובעזרתו נחשב אות y_i , ואז בהינתן y_i לבצע עדכן לפורמטרים= $=\text{נבחן מהו סט הפורמטרים שסביר בצורה הטובה ביותר את האשכולות שהתקבלו בחישוב } \hat{Q}_i$ = $=\text{באופן פורמל-כשי השלב-$ מנוסחים כפ- $=\text{בהתיק אוסף נקודות א-וערך עבור הפורמטרים המתאים לכל נקודה, כולם-כל נקודה}$

$=\text{x תואם לאשכול מסוימת } y_i$. עבור כל הנקודות y_i נחשב תוחלת ובעזרתה נגידר את הפונקציית $Q(\theta, \theta_0)$, כאשר θ הוא פורמטר חדש $\hat{\theta}$ הוא סט הפורמטרים הנוכחיים

$$Q(\theta, \theta_0) = \sum_{i=1}^n \sum_{j=1}^k p(y_i = j | x_i; \theta_0) \log p(y_i = j, x_i; \theta) = \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log p(y_i = j, x_i; \theta)$$

$$\sum_{i=1}^n \mathbb{E}_{p(y_i | x_i; \theta_0)} \log p(y_i = j, x_i; \theta)$$

$=\text{M-step} - \text{מחשבים את הפורמטר } \theta \text{ שיביא למקסימום אוט } Q(\theta, \theta_0) \text{ ואז מעדכנים את } \theta_0 \text{ ל-} \theta \text{ החדש:}$

$$\theta = \arg \max_{\theta} Q(\theta, \theta_0)$$

$$\theta_0 \leftarrow \theta$$

$=\text{ חוזרים על התהליך-באופן איטרטיבי-עד להתקנסות}$

$=\text{עבור Mixture of Gaussians נוכל לחשב באופן מפורש את הביטויים:}$

$$Q(\theta, \theta_0) = \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log p(y_i = j, x_i; \theta)$$

$$= \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log p(y_i = j; \theta) + \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log p(x_i | y_i = j; \theta)$$

$$= \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log \alpha_j + \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log \mathcal{N}(\mu_j, \sigma_j^2)$$

$$= \sum_{i=1}^n \sum_{j=1}^k w_{ij} \log \alpha_j - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^k w_{ij} \left(\log \sigma_j^2 + \frac{(x_i - \mu_j)^2}{\sigma_j^2} \right)$$

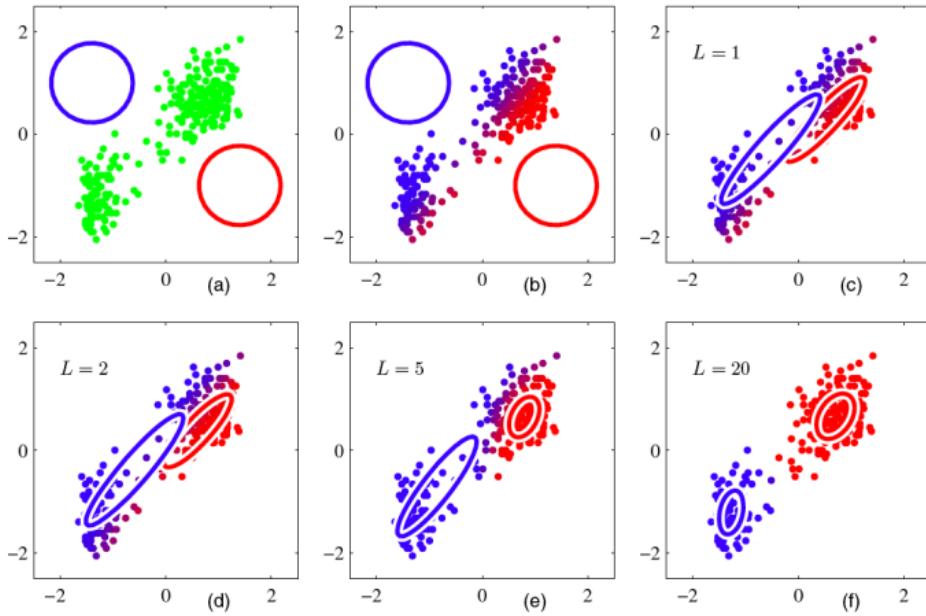
$=\text{וכעת ניתן לגזר ולמצוא אופטימום:}$

$$\hat{\alpha}_j = \frac{1}{n} \sum_{i=1}^n w_{ij}$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^n w_{ij} x_i}{\sum_{i=1}^n w_{ij}}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^n w_{ij} (x_i - \mu_j)^2}{\sum_{i=1}^n w_{ij}}$$

$=\text{עבור התפלגיות שונות שאין בהכרח נורמליות יש לחזור לביטוי של } Q(\theta, \theta_0) \text{ ולבצע עבורי את האלגוריתם.}$



איוֹרְגָּרִיזִיּוֹת =2.10= מינוח שאל אלגוריתם EM=מתחלים אקראי של התפלגות, ובכל איטרציה יש שיפורן שההתפלגות מייצגות בצורה יותר טובה את הדטה המקורי.

ונכון שהאלגוריתם משתמש בכל איטרציה, ככלור שעבור θ_0 מתקיים $\log p(x; \theta) \geq \log p(x; \theta_0)$

$$\begin{aligned} \log p(x; \theta) &= \sum_y p(y|x; \theta_0) \log p(x; \theta) = \sum_y p(y|x; \theta_0) \frac{\log p(x, y; \theta)}{\log p(y|x; \theta)} \\ &= \sum_y p(y|x; \theta_0) (\log p(x, y; \theta) - \log p(y|x; \theta)) \\ &= \sum_y p(y|x; \theta_0) \log p(x, y; \theta) - p(y|x; \theta_0) \log p(y|x; \theta) \end{aligned}$$

נשים לב שהאיבר הראשון הוא בדיקת $Q(\theta, \theta_0)$. האיבר השני לפי הגדרה הוא האנטרופיה של התפלגות $p(y|x; \theta_0)$:

$$H(\theta, \theta_0) = -\sum_y p(y|x; \theta_0) \log p(y|x; \theta_0)$$

cut עבור שני ערכים שונים של θ מתקיים:

$$\begin{aligned} \log p(x; \theta) - \log p(x; \theta_0) &= Q(\theta, \theta_0) + H(\theta, \theta_0) - Q(\theta_0, \theta_0) - H(\theta_0, \theta_0) \\ &= Q(\theta, \theta_0) - Q(\theta_0, \theta_0) + H(\theta, \theta_0) - H(\theta_0, \theta_0) \end{aligned}$$

לפְּאֵ-שִׁיוּן גִּיבּוֹ מתקיים $H(\theta_0, \theta_0) \geq H(\theta, \theta_0)$, לכן

$$\log p(x; \theta) - \log p(x; \theta_0) \geq Q(\theta, \theta_0) - Q(\theta_0, \theta_0)$$

ולכן עבור כל עדכון של θ שטבי לאופטימום את $Q(\theta, \theta_0) - Q(\theta_0, \theta_0)$ יהיה חיובי וממילא יהיה שיפור ב- $\log p(x; \theta)$.

2.2.4 Hierarchical Clustering

אלגוריתם נוסף של למידה לא מונחית עבור חלוקת אוכלות=Ashkolanot-enkeria מחלק לשתי שיטות שונות

agglomerative clustering=Bashlev הראשונ-מגדירים כל נקודת-אשכול=Ashkolanot-enkeria אחד, ואז בכל פעם מאחדים שני אשכולות ובקה-מורדים את מספר האשכולות-ב-1, עד שmaguimel=Ashkolanot-enkeria האיחוד בכל שלב-נעשה על ידי מציאת שני

האשכולות הקרובים ביותר זה לאו איחודם לאשכול אחד-ראשית יש לבחור מטריקה לחישוב מרחק בין שתי נקודות (למשל מרחק אוקלידי, מרחק מנהטן ועוד), ולאחר מכן-Calculating distances between points in two adjacent clusters (for example, Euclidean distance, Manhattan distance, etc.). After that, choose a metric to calculate the distance between the centers of the clusters. The result will be a single number representing the distance between the two clusters. This number can be used to determine whether the clusters are similar enough to merge them into a single cluster or if they are too different to be merged.

complete-linkage clustering: $\max\{d(a, b) : a \in A, b \in B\}$.

single-linkage clustering: $\min\{d(a, b) : a \in A, b \in B\}$.

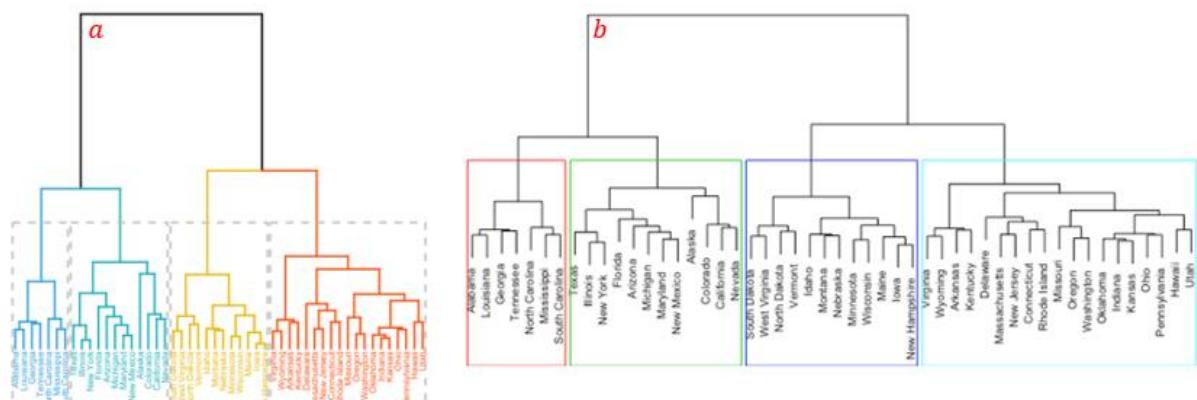
Unweighted average linkage clustering (UPGMA): $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$.

Centroid linkage clustering (UPGMC): $\|c_s - c_t\|$ where c_s, c_t are centroids of clusters s, t, respectively.

עם התקדמות התהילה יש פחות אשכולות מאשר בתחילת השם נקודה אחת בלבד אלא הם הולכים וגדלים=שיטה *bottom-up* כיוון שבחתירה כל נקודה הינה אשכול עצמאו-ובכל צעד של האלגוריתם מספק האשכולות רק בחלק אחד מthem, האלגוריתם בונה את האשכולות ממצב שבו אין למשה חיבור לאלגוריתם האשכולות באחד-במילים אחרות, האלגוריתם בונה את האשכולות ממצב שבו אין למשה חיבור לאלגוריתם האשכולות נאכזב אשכולות ההולכים וגדלים=

divisive clustering – בשיטה זו מוצעים פעולה הפוכה – מסתכלים על כל הנកודות כאשכול אחד, ואז בכל שלב מבצעים חלוקה של אחד האשכולות לפחות לחלק שנקבע מראש – עד שmagיע – כל אשכולות – כיוון שיש "דריכיף חלק את המdatum, יש הכרה לנוקוט בשיטות היוריסטיות כדי לקבוע את כל החלוקה המתאימים בכל שלב – שיטה מקובלת לביצוע החלקה נקראת (DIANA) – Dlvisive ANAlysis Clustering – ולפייה בכל שלב בוחרים את האשכול בעל השונות הכי גדולה ומחלקים אותו לשני – שיטה זמcona – top-down פואנושבה תחלה – יש אשכול ייחיד אבל אעד של האלגוריתם מתוויסף עד אשכול

את התצוגה של האלגוריתם¹ ניתן להראות בצורה נוחה באמצעות dendrogram—דיאגרמה הבנויה כעץ המיציג קשרים בין קבוצות.



11.2.2. הדרוגרָם (dendrogram) – חישוב הארכיטקטורה היררכית (Hierarchical Clustering)

2.2.5 Local Outlier Factor (LOF)

אלגוריתם=Local Outlier Factor=הינו אלגוריתם של למידה לא מונחית למציאת נקודות=חריגות (Outliers)=
האלגוריתם מחשב לכל נקודה ערך הנקרא=Local Outlier Factor (LOF), פועל פ' ערך זה ניתן לקבוע עד כמה
הנקודה היא חלק מקבוצה או לחילופין חריגה ויצאת דופן

בשלב ראשון בוחרים ערך \neq מסוים. עבור כל נקודת x_i נסמן את השכנים הקרובים ביותר של x_i כ- $N_k = \{x_j | d(x_i, x_j) \leq k\}$. נגיד אם $k=3$, אז $N_3 = \{x_1, x_2, x_3, x_4, x_5\}$. הוא סט המכיל את שלושת השכנים הקרובים ביותר של x_i . מחרק מהשכנים הקרובים ביותר של x_i נבחר מחרק בין שני שכנים נתנו לבחירה – זה יכול להיות למשל מחרק אוקלידי, מחרק מנהטן או מחרק ביחס למרכזו של מחרק אוקלידי. ניתן להסתכל על k -distance – הרדיוס של מעגל המינימל-האכלי את כל הנקודות השוכנות ל- x_i .

לאחר חישוב k -distance של כל נקודה, מחשבים לכל נקודה Local Reachability Density (LRD) באופן הבא:

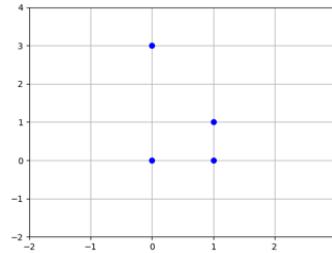
$$LRD_k(x_i) = \frac{1}{\sum_{x_j \in N_k(x_i)} \frac{RD(x_i, x_j)}{k}}$$

כasher(x_i, x_j) = max($k - \text{distance}(x_i, x_j)$) = $\text{hagadol} = RD(x_i, x_j)$
המრחקים ב'יק' אלבי'ך' השכנים הקרובים אליו. ככל שנproxה יותר קרובהה-ך' השכנים שלהvr כר-h-LRD-ל-
ויתר, ו-LRD-ל-קטן משמעתו שהproxה יחסית רוחקה מאשוכ'ה-הקרוב אליו.
בשלב האחורי בוחנים עברו כל נקודה i את היחס בין ה-LRD-ל- $N_k(x_i)$ זהה הוא ה-LOF,
והוא מחושב באופן הבא:

$$LOF_k(x_i) = \frac{\sum_{x_j \in N_k(x_i)} LRD(x_j)}{k} \times \frac{1}{LRD(x_i)}$$

הביטוי הראשון במכפלה הוזכר ממאמרה-LRD-ל- $N_k(x_i)$, ולאחר חישוב הממוצע מחלוקתם אותו ב-
 RD -ל- $N_k(x_i)$ עצמה=אם הערכים קרובים, אז ה-LOF-ל- $N_k(x_i)$ =1, ואם הנקודה x_i אבامت לא שייכת
לאשכול של נקודות, אז ה-LOF-ל- $N_k(x_i)$ =הממוצע של השכנים שלה-ו-ומילא-ה-
שלה יהיה גבוה. אם עברו נקודה x מתקובל $1 \approx LOF$, אז סביר שהיא חלק מאשכול מסוים.
כדי להמחיש את התהילה נסתכל על הדוגמה הבאה: $\{A = (0,0), B = (1,0), C = (1,1), D = (0,3)\}$, ונקבע $k = 2$. נחשב את ה- k -distance של כל נקודה במנוחים של מרחק מנהטו:

$$\begin{aligned} k(A) &= \text{distance}(A, C) = 2 \\ k(B) &= \text{distance}(B, A) = 1 \\ k(C) &= \text{distance}(C, A) = 2 \\ k(D) &= \text{distance}(D, C) = 3 \end{aligned}$$



נחשב את ה-LRD-ל-

$$LRD_2(A) = \frac{1}{\frac{RD(A, B) + RD(A, C)}{k}} = \frac{2}{1+2} = 0.667$$

$$LRD_2(B) = \frac{1}{\frac{RD(B, A) + RD(B, C)}{k}} = \frac{2}{2+2} = 0.5$$

$$LRD_2(C) = \frac{1}{\frac{RD(C, B) + RD(C, A)}{k}} = \frac{2}{1+2} = 0.667$$

$$LRD_2(D) = \frac{1}{\frac{RD(D, A) + RD(D, C)}{k}} = \frac{2}{3+3} = 0.334$$

ולבסוף נחשב את ה-LOF-ל-

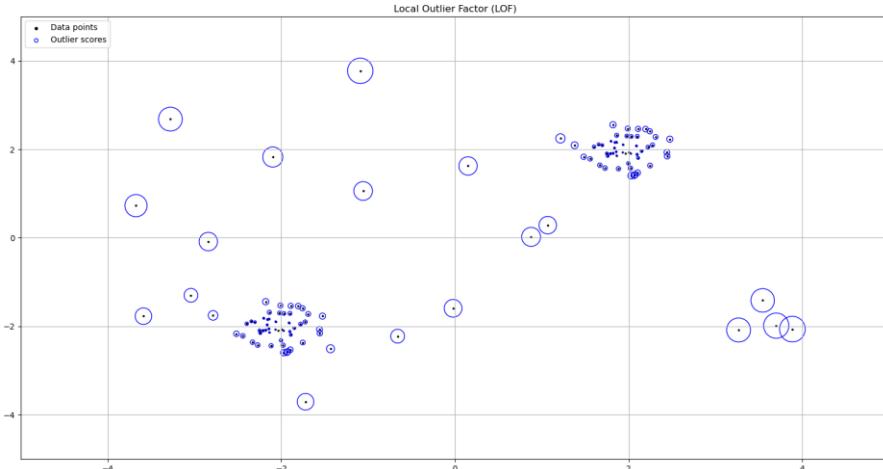
$$LOF_2(A) = \frac{LRD_2(B) + LRD_2(C)}{k} \times \frac{1}{LRD_2(A)} = 0.87$$

$$LOF_2(B) = \frac{LRD_2(A) + LRD_2(C)}{k} \times \frac{1}{LRD_2(B)} = 1.334$$

$$LOF_2(C) = \frac{LRD_2(B) + LRD_2(A)}{k} \times \frac{1}{LRD_2(C)} = 0.87$$

$$LOF_2(D) = \frac{LRD_2(A) + LRD_2(C)}{k} \times \frac{1}{LRD_2(D)} = 2$$

כיוון ש- $1 \ll (DLOF_2)$ ביחס לשאר הנקודות, נסיק כי נקודה D היא outlier.



איך. 12.2 Local Outlier Factor (LOF) – מציג את נקודות חריגות על ידי השוואת ערך ה-LOF של כל נקודה למרכז ה-LOF של השכנים שלה. ככל שהיא גודל יותר (היעילו הכלול), כך הנקודה יותר רחוקה מאשר כל נקודה

שני אתגרים מרכזים בשימוש באlgorigthms זה – ראשית יש לבחוח את מטא-אחסנה של נתונים ייחודיים קטנים וטוב עובך נקודות רועשות; ואך יכול להיות בעיתי במרקטים בהם יש הרבה מאוד נקודות הצמודות אחת לשפיה, וכן נודה שמעס רוחק ממאויסת זהה כחריגת מרותה שהיא באמת כן שיכת אלוי – גודל-לעומת זאת – יתרגבר על בעיה זו, אך הוא לא יזהה נקודות חריגות שנמצאות בקשר בלבד לאתגר זה, יש צורך למת פרשנות לתוצאות המתקובלות, ולהחליל-על סוף מוסים ש-LOF, שהחל ממנה נקבעה מסווגת חריגות LOF-קטן מ-1-הוא בוודאי לא-utliner אבעור עריכת LOF-גדולים מ- k -אין כלל חד משמעי עבור איזה ערך הנזקודה הייא-utliner עבורו איזה ערך היא לא-כדי להתמודד עם אתגרים אלגורitmischen הרחבות לשיטה המקורית, כמו למשקל-שימוש בסטטיסטיות שונות המורידות את התלות בבחירה הערך – (LoOP – Local Outlier Probability), או שיטות סטטיסטיות העוזרות לתהפרשנות לעריכים המתקובלים – Interpreting and Unifying Outlier Scores).

2.3 Dimensionally Reduction

הורדת ממד (Dimensionality Reduction) הינה-הטרנספורמציה של דatasה מממד=גבוה למספר נרחב שהורדת הממד=לא תנסה באופן מהותי את מאפייני הדatasה המקורי. הורדת הממד של datasה נתון נדרשת משפט סיבות עיקריות; הראשונה טכנית וקשורה לשיבוכיות גבואה במערכת מרובת=ממדיים, ואילו הסיבה השנייה יותר עקרונית ומהותית—הורדת הממד של datasה קשורה לניסיוק להבין מהם המשתנים העיקריים המשניים, הפחות חשובים להבנת datasה (אלו שפחות מאפיינים דוגמא נתונה ביחס לדוגמאות אחרות). לעתים התחשבות במסתנים המשניים משפיעה לרעה על ביצוע המודל, למשל על ידי הוספת רעש ולא מידע. תופעה זו נקראת כללת הממדויות (curse of dimensionality). יתרון אסף של הורדת ממד טמון בויזואלייזציה של המידע, כפי שנition להציגו על יד-2 או 3 ממדים עיקריים. בערךת גרפ-דו-ממד או תלת-ממד בהסתמך

דוגמא למערכת מרובת-המודדים יכולה להיות מדידת רמות חלבונים (פרוטאיןים) של גנים (genes) המבוטאים בתא, כאשר כל ממד, או מאפיין (פי'צ'ר), מתאים לגן אחר. באופן כללי, יתכן ונמודדים בכל ניסוי מאות תאים, כאשר לכל תא נמדדות רמות ביוטו של מאות או אלפי גנים. כמות עצומה זו של מידע במנגד-גבוה (אלפי תאים ואלפי גנים בכל תא) מאגירת את המחקקה=הן מבחינת זיהוי המאפיינים, או רמות הגנים המבוטאים, הרלוונטיות להמשפיעים בioter= והן מבחינות ניסיון למדל את הדadata בצורה כמה שיותר פשוטה. במחקר משנת 2007 נלקוח 505 דוגימות של תא= סרטן ש, כאשר לכל דוגימה (אקסוגמא) נמדד רמות התבטאות של 6487 גנים שונים=כמוצע שלנתה את המדיע בצורה הגלומית זו מושימה בלתי אפשרית, ויש לכך לבצע עליון מניפולציה כלשהיא כדי שהייה אפשר לעבד אותה

ישן שיטות מרובות להורדת ממד לדאטה נתון, כאשר ניתן לסייע לשני חלקיים עיקריים (feature selection), והטלת מאפיינים (features projection). השיטה הראשונה היא ניסיון לבחור את המאפיינים (המשתנים) המתאים באופן מספק את המידע הנתון. השנייה, שבה עוסק פרק זה, נוקטת בגישה של הטלה-טרנספורמציה, של המאפיינים הקיימים לסת של מאפיינים חדשים=חשוב להציג שבשיטות בחירת המאפיינים אף בעצם משמשים מאפיינים פחות רלוונטיים. בוגדור לכך, בשיטה שנדון בערך, שיטת הטלת המאפיינים, כל מאפיין חדש

הוא צירוף לינארי של כל האחרים, ולא רק של חלקם. כך, המאפיינים החדשניים מقلילים, או לוקחים בחשבון, כל אחד מהמאפיינים הנמדדים המקוריים, ללא השמטה.

ניתן לבצע הטלה מאפיינים באמצעות טרנספורמציה ליניארית או לא-LINIARITY. בפרק זה עוסוק בטרנספורמציה ליניארית אחת, הנקראת ניתוח גורמים ראשיים (principal component analysis) ובשתי טרנספורמציות לא-LINIARITY (t-SNE, UMAP).

2.3.1 Principal Components Analysis (PCA)

כפשוויזר לעיל-ליניאר-גורמים ראשיים מבוסס על טרנספורמציה ליניארית של המאפיינים המקוריים. הגורם הראשון (first principal component, PCA₁) הינו הצירוף לינארי של המאפיינים הנתונים בעל השונות הגדולה ביותר. הגורם הראשי השני (second principle component, PCA₂) הוא גם צירוף לינארי של המאפיינים הנתונים, השונות שלו היא השניה הגדולה ביותר, ובנוסף דורשים ש-PCA₂ אורתוגונלי ל-PCA₁. הגורם השלישי (third principle component, PCA₃) הוא צירוף לינארי בעל השונות השלישית הגדולה ביותר, ומאונך לשני הגורמים הראשונים PCA₁ ו-PCA₂. וכך גם גורםPCA_i הוא צירוף לינארי כרך שהגורם הראשי מסדר i , והוא בעל השונות ה- i -ית הגדולה ביותר בתורת תחת אילוץ של גורמים מאונכים $\sum_{j=1}^i \text{PCA}_j = 0$.

לאחר שאפיינו את הגורמים הראשיים בהם אנו מעוניינים, עולה השאלה כיצד ניתן לבצע טרנספורמציה ליניארית שבuzzetta ניתן למצוא את הגורמים הראשיים הללו. נניח שבידינו דאטה $\vec{X} \in \mathbb{R}^{M \times N}$, כלומר נתוננו M דוגמאות שונות, שכל אחת מהן היא בעל-מאפיינים [למשל, עבור הדוגמא של תא סרטן השד, נתון מידע $M = 105$ תאים שונים, כאשר עבור כל תא נמדדו רמות ביוטי $N = 27,648$ גנים שונים]. נסמן את מטריצת המאפיינים על ידי

$$\vec{X} = \begin{bmatrix} \vec{X}_1 \\ \vdots \\ \vec{X}_M \end{bmatrix} \in \mathbb{R}^{M \times N}, \quad \text{כאשר } \vec{X}_m \text{ וקטור שורה של } m \in \{1, \dots, M\} \text{, הינו נתוני המדידות של מאפיין ה-} m \text{ עליון עבור וקטור שורה שורה של מאפיין ה-} m \text{ עליון.}$$

המאפיינים השונים בדוגמה מס' m , בהתאם, וקטור عمודת $\vec{x}_m \in \mathbb{R}^n$ (שימו לב לשינוי סימונו, איןדקס על-אי עבור וקטור עמודה), הינו נתוני המדידות של מאפיין מסוים על כל הדוגמאות. נניח שסכום המדידות עבור כל מאפיין הוא אפס, זאת אומרת שכל מאפיין מתקיים:

$$\text{mean}(\vec{X}^m) = \sum_{n=1}^M X_{m,n} = 0$$

מכיוון שכל עמודה של המטריצה מסמלת ערכים של מאפיין מסוים במדידות שונות, סכום כל עמודה במטריצה ה- \vec{W} הוא אפס. עתה, נרצה לבצע הטלה (טרנספורמציה) ליניארית, זאת אומרת נכפיל את מטריצת \vec{W} במטריצת משקלים \vec{W} :

$$\hat{T} = \vec{X} \cdot \vec{W}$$

אם נסמן את השורה k -ית במטריצה \hat{T} על ידי \vec{T}_k , נקבל:

$$\vec{T}_k = \vec{X}_k \cdot \vec{W}$$

כאשר המטריצה $\vec{W} \in \mathbb{R}^{N \times K}$, כך ש- $\vec{W} \in \mathbb{R}^{M \times K}$. הטלה זו מביאה לכך שלאחר הטרנספורמציה נשארים רק K מאפיינים. כיוון שאנו מעוניינים בהורדת הממד, קרי הורדת מספר המאפיינים, נדרש $N \leq K$. את תהליך מציאת מטריצת המשקלים ניתן לנתח באופן פורמלי על ידי שלושה תנאים:

(1) כל עמודה של מטריצת המשקלים הינה מנורמלת: $\|\vec{W}\|^2 = \sum_{m=1}^M (W_{m,k})^2 = 1$

(2) השונות בעב-המאפיין k -י, המוגדרת על ידי $s_k^2 = (\vec{T}_k)^T \vec{T}_k = \sum_{m=1}^M (T_{mk})^2$, מקיימת $N \leq K$.

(3) העמודות של \vec{W} אורתוגונליות זו לזו, זאת אומרת $\vec{W}^k \perp \vec{W}^l$ לכל שתי עמודות k, l .

נראה זאת באופן מפורש: נתחילה במציאת העמודה הראשונה \vec{W}^1 . נדרש:

$$\vec{W}^1 = \underset{\|\vec{W}\|=1}{\text{argmax}}(s_1^2)$$

זאת אומרת:

$$\begin{aligned}\widehat{W}_1 &= \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}(s_1^2) = \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\vec{T}^1\right)^T \cdot \vec{T}^1\right) = \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\widehat{X}\widehat{W}^1\right)^T \cdot \widehat{X}\widehat{W}^1\right) \\ &= \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\widehat{W}^1\right)^T\left(\widehat{X}\right)^T \cdot \widehat{X}\widehat{W}^1\right)\end{aligned}$$

ולכן העמודה הראשונה של מטריצת המשקלים \widehat{W} נתונה על ידי

$$\widehat{W}^1 = \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\widehat{W}^1\right)^T \cdot \widehat{S} \cdot \widehat{W}^1\right)$$

כאשר מטריצה $\widehat{S} = (\widehat{X})^T \cdot \widehat{X} \in \mathbb{R}^{(N \times N)}$ הינה מטריצת השונות המשותפת (covariance), המוגדרת על ידי $\widehat{X} = S_{v_1, v_2} = \sum_{m=1}^M X_{v_1, m} X_{m, v_2}$, מגדירה את השונות המשותפת בין שני מאפיינים, כאשר ניתן לשים לב כי מטריצה זו סימטרית ומשנית (ולכן הרミיטית).

לפי משפט המינימום-מקסימום (קורנט-פישר-ויל): עבורה מטריצה הרמייטית ($S_{ij} = S_{ji}^*$), בעלת ערכים עצמיים $\lambda_1 \geq \dots \geq \lambda_K$

$$\lambda_1 = \underset{\|\widehat{W}\|=1}{\operatorname{max}}\left(\left(\widehat{W}^1\right)^T \cdot \widehat{S} \cdot \widehat{W}^1\right)$$

כאשר \widehat{W}^1 הינו הווקטור העצמי המתאים לערך העצמי המקורי שנקרא λ_1 .

כעת, כדי למצוא את הווקטור העצמי הבא, \widehat{W}^2 , והערך העצמי המתאים לו λ_2 , נגדיר מטריצה חדשה \tilde{X} :

$$\tilde{X} = \widehat{X} - \widehat{X}\widehat{W}^1(\widehat{W}^1)^T$$

$$\begin{aligned}\widehat{W}^2 &= \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}(s_2^2) = \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\vec{T}^2\right)^T \cdot \vec{T}^2\right) \\ &= \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\widehat{W}^2\right)^T\left(\tilde{X} + \widehat{X}\widehat{W}^1(\widehat{W}^1)^T\right)^T \cdot \left(\tilde{X} + \widehat{X}\widehat{W}^1(\widehat{W}^1)^T\right) \widehat{W}^2\right) \\ &= \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\widehat{W}^2\right)^T\left(\tilde{X}\right)^T \cdot \left(\tilde{X}\right) \widehat{W}^2\right)\end{aligned}$$

כאשר \widehat{W}^2 הינה הווקטור העצמי המתאים לערך העצמי המקורי שנקרא \tilde{X} , ובפועל הוא הערך העצמי השני בגודלו עבור מטריצה $\widehat{X}^T \widehat{X} = \widehat{S}$. (בчисלוב השתמשנו בעובדה כי $\widehat{W}^1 \perp \widehat{W}^2$).

באופן כללי, כדי למצוא את \widehat{W}^k והערך העצמי המתאים לו λ_k , נגדיר מטריצה חדשה \tilde{X} באופן הבא

$$\begin{aligned}\tilde{X} &= \widehat{X} - \sum_{i=1}^{k-1} \widehat{X}\widehat{W}^i(\widehat{W}^i)^T \\ \widehat{W}^k &= \underset{\|\widehat{W}\|=1}{\operatorname{argmax}}\left(\left(\widehat{W}^k\right)^T\left(\tilde{X}\right)^T \cdot \left(\tilde{X}\right) \widehat{W}^k\right)\end{aligned}$$

כך ש λ_k הינו הערך העצמי המקורי k -י של מטריצת השונות המשותפה $\widehat{X}^T \widehat{X} = \widehat{S}$.

ניתן גם, באופן פשוט יותר, להשתמש בשיטת פירוק לרכיבים נגוניים, כאשר נמצא את הפירוק המתאים למטריצת השונות המשותפה

$$\widehat{S} = \widehat{W} \cdot \widehat{\Lambda} \cdot \widehat{W}^T$$

כאשר $\widehat{\Lambda}$ הינה מטריצה אלכסונית, $\Lambda_{ii} = \lambda_i$ הינם הערכים העצמיים של המטודרים לפי גודלם מהגדול לקטן, ומטריצה \widehat{W} מורכבת מוקטוריו עמודה שהינם הווקטורים העצמיים המתאימים לערכים $\lambda_M \geq \dots \geq \lambda_2 \geq \lambda_1$.

העצמיים. הוקטורים העצמיים בהגדרתם הינם אורתוגונליים זה לזה, וכך $\hat{W}^k = \text{לכל } k$, הם בעצם אורתוגונרמליים:

לטיכום, על מנת למצוא את הגורמים הראשיים עבור המידע הנוכחי \hat{X} :

$$\hat{X}^m = \hat{X}^m - \text{mean}_n(\hat{X}^m) \quad \text{א. "מרכז" את הנתונים כך שהממוצע עבר כל מאפיין הוא אפס:}$$

$$\hat{S} = (\hat{X})^T \cdot \hat{S} \quad \text{ב. מצא את מטריצת השונות המשותפת } \hat{X}^T \cdot \hat{S}.$$

$$\hat{S} = \hat{X}^T \cdot \hat{X} \quad \text{ג. מצא את } \hat{W}^T \cdot \hat{X} \cdot \hat{S}.$$

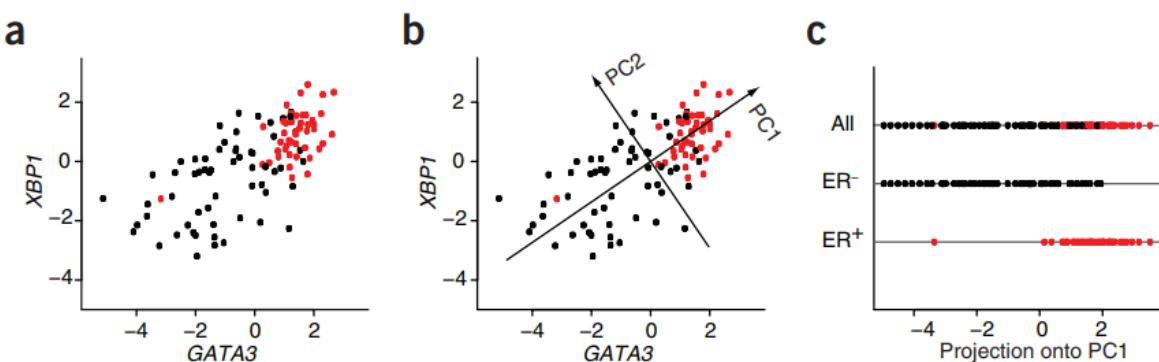
$$\hat{W}^T \cdot \hat{X} = \hat{S} \quad \text{ד. חשב } \hat{W} \cdot \hat{X} = \hat{S}.$$

הגורמים הראשיים נתונים על ידי וקטורי העמודה $\vec{W}^k \equiv PCA_k$, וההטלה של מדידה m למערכת המאפיינים החדשה נתונה על ידי $\vec{X}^m = \hat{X}^m + \text{mean}_n(\hat{X}^m)$.

נזכיר שלשיטת הניתוח של גורמים ראשיים יש מספר מגבלות. ראשית, היא נותנת "משקל יתר" על מאפיינים שהשונוות בהם גדולה, ללא קשר לחשיבותם, או ליחידות שבahn המאפיין נמדד (זאת אומרת לדוגמה שגובה שנמדד בסנטימטרים ינתן "משקל" גובה יותר מאשר גובה הנמדד במטרים). שנית, שיטת זו מיניחה כי הממד החשוב הוא השונות המשותפת שהוא בעצם קורלציה בין שני משתנים, ואולם יתכן במערכות מסוימות שדווקא הקורלצייה הלא-ליניארית היא החשובה יותר. כמו כן, ליעיתם "מרכז" המידע גורם לתוצאות לא-בד ממשמעותן.

כדי להתגבר על המוגבלות בשיטת-PCA השחגנו לעיל פוטחו שיטות נוספות או משלימות. לדוגמה, ניתן למזער את השפעת יחידות המידע על המאפיינים על ידי הפיכתם לחסרי יחידות. בנוסף, יש שיטות הלוקחות בחשבון קורלציות לא-ליניאריות, לדוגמה שיטהPCA kernel, או שיטות להתחממות עם בעית המידע על ידי דרישת משתנים חיוביים (NMF).

לצורך המחשה ניתן שתי דוגמאות. ראשית נחזור לדוגמא שהזכרנו בתחילת פרק זה – מחקר שפורסם בשנת 2007 ובו נלקחה 405 דוגמאות של תא סרטן אחד, כאשר לכל דגימה נמדד רמות התבטאות של 27,648 גנים שונים. לשפה הדגמה, נשתמש בניתוח שפורסם כנסה לאחר מכן (ב-2008) על ידי אחד מעורכי המחקר המקורי. שם, החוקר מציא רמות של שני חלבונים; האחד בשפה GATA3, והשני בשפה XBP1, כאשר הוא מסווים את דגימות תא סרטן לפי סוג קולטני האסטרוגן שלהם (+ או -). ב庆幸, על ידי "סיבוב" מערכת הציר- X - Y ב围着ור טרנספורמציה ליניאריתPCA קפיצה שהושобр לעיל – נמצא כי ניתן לסווג, ללא אי-בוד מידע רב, את מצב קולטני האסטרוגן בתאי סרטן השד על ידי הגורם הראשי PCA₁, כפי שניתן לראות באירוע. יש לשים לב שהגורם הראשי PCA₁ מכיל מידע משפחתי החלבוניים.



איור 2.13 (a) רמות ביוטי של שני חלבונים GATA3 (ציר-X) ו-XBP1 (ציר-Y). קולטני אסטרוגן חיוביים או שליליים מסומנים באדום ו���ור בהתאם (b) מציאת הגורמים הראשיים, וסיבוב מערכת הצירים בהתאם לטיוריה, ניתן להבחין כבהשנות של המידע על גבי הציר החדש PCA₁ (הינה מקסימלית) הציג תוצאות המידע כפונקציה של PCA₁ בבלבד בגרף זה ניתן לראות בבירור כיצד הורדת הממד מס'יעת למציאת הבנה פשוטה (בمعد אחד) בין קולטני האסטרוגן.

נבייא בנוסף דוגמא חשובה מפורטת. נניח וננתן המערכת הדדמומי הבאה

$$X = \begin{pmatrix} -0.5 & -0.4 \\ -0.4 & -0.1 \\ 0.1 & 0 \\ 0.3 & 0.3 \\ 0.5 & 0.2 \end{pmatrix}$$

מערך הנתונים מכיל 5=דוגמאות, כל=דוגמא=نمذדו שמי מאפיינים. זאת אומרת $M = 5, N = 2$, כך=שורות המטריצה מציגות את המדידות השונות, והעמודות מייצגות את מאפייניהם.

מערך זה כבר ממורכז, כלומר מתקיים עבור המאפיין הראשי:

$$mean_1(X^m) = \sum_{m=1}^5 X_{m1} = -0.5 - 0.4 + 0.1 + 0.3 + 0.5 = 0$$

עבור המאפיין השני:

$$mean_2(X^m) = \sum_{m=1}^5 X_{m2} = -0.4 - 0.1 + 0 + 0.3 + 0.2 = 0$$

נחשב את מטריצת השונות המשותפת:

$$\begin{aligned} S = (\hat{X})^T \hat{X} &= \begin{pmatrix} -0.5 & -0.4 \\ -0.4 & -0.1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.3 & 0.5 \\ 0.3 & 0.2 & 0.2 \end{pmatrix} \begin{pmatrix} -0.5 & -0.4 \\ -0.4 & -0.1 \\ 0.1 & 0 \\ 0.3 & 0.3 \\ 0.5 & 0.2 \end{pmatrix} \\ &= \begin{pmatrix} 0.5^2 + 0.4^2 + 0.1^2 + 0.3^2 + 0.5^2 & 0.5 \cdot 0.4 + 0.4 \cdot 0.1 + 0.1 \cdot 0 + 0.3^2 + 0.5 \cdot 0.2 \\ 0.5 \cdot 0.4 + 0.4 \cdot 0.1 + 0.1 \cdot 0 + 0.3^2 + 0.5 \cdot 0.2 & 0.4^2 + 0.1^2 + 0^2 + 0.3^2 + 0.2^2 \end{pmatrix} \\ &= \begin{pmatrix} 0.76 & 0.43 \\ 0.43 & 0.3 \end{pmatrix} \end{aligned}$$

על מנת ללקסן מטריצה זו, נפתחו:

$$0 = |\hat{S} - \lambda \hat{I}| = \begin{vmatrix} 0.76 - \lambda & 0.43 \\ 0.43 & 0.3 - \lambda \end{vmatrix} = (0.76 - \lambda)(0.3 - \lambda) - 0.43^2 \approx (\lambda - 1.02)(\lambda - 0.04)$$

כאשר לפולינום אופיני זה שתי פתרונות $\lambda_1 \approx 1.02, \lambda_2 \approx 0.04$ (שים לב שבחירת $\lambda_1 > \lambda_2$, התוצאות המובאות בחלק זה מקובלות ולכן הסימן \approx).

נמצאת הווקטור העצמי המתאים לערך העצמי הגדל מבין השניים $=\lambda_1$. וקטור זה, המסומן על ידי \hat{W}^1 , מקיים

$$\hat{S}\hat{W}^1 = \lambda_1 \hat{W}^1$$

כך ש-

$$0 = (\hat{S} - \lambda_1 \hat{I})\hat{W}^1 \approx \begin{pmatrix} -0.83 & 0.107 \\ 0.107 & -0.94 \end{pmatrix} \begin{pmatrix} W_{11} \\ W_{21} \end{pmatrix} \Rightarrow \hat{W}^1 \approx \begin{pmatrix} 0.86 \\ 0.51 \end{pmatrix}$$

הווקטור העצמי השני, \hat{W}^2 , המתאים לערך העצמי $\approx 0.04 =\lambda_2$, מחושב באותו אופן, ומתקיים:

כך שמטריצת המשקלים נתונה על ידי

$$\hat{W} = (\hat{W}^1 \quad \hat{W}^2) = \begin{pmatrix} 0.86 & 0.51 \\ 0.51 & -0.86 \end{pmatrix}$$

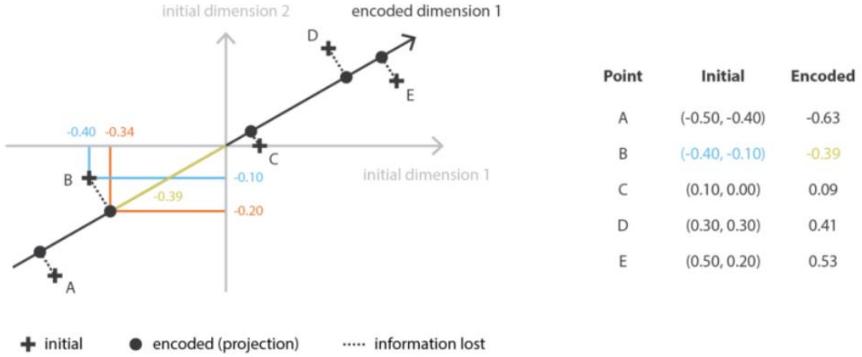
הטלה המדידות למערכת המאפיינים החדשה נתונה על ידי

$$\hat{T} = \hat{X} \cdot \hat{W}$$

לכן, המדידות של הגורם הראשי הראשון, נתונת על ידי

$$\hat{T}^1 = \hat{X} \cdot \hat{W}^1 \approx \begin{pmatrix} -0.5 & -0.4 \\ -0.4 & -0.1 \\ 0.1 & 0 \\ 0.3 & 0.3 \\ 0.5 & 0.2 \end{pmatrix} \begin{pmatrix} 0.86 \\ 0.51 \end{pmatrix} = \begin{pmatrix} -0.5 \cdot 0.86 - 0.4 \cdot 0.51 \\ -0.4 \cdot 0.86 - 0.1 \cdot 0.51 \\ 0.1 \cdot 0.86 \\ 0.3 \cdot 0.86 + 0.51 \cdot 0.3 \\ 0.5 \cdot 0.86 + 0.2 \cdot 0.51 \end{pmatrix} \approx \begin{pmatrix} -0.63 \\ -0.39 \\ 0.09 \\ 0.41 \\ 0.53 \end{pmatrix}$$

נראה זאת באופן גרפי:



איור 2.14 הורדת מידע של נתונים מ-2-ממדים לממד אחד.

נספח: משפט המינימום- מקסIMUM (קורנט-פישר=ויל)=

ובו \hat{S} מטריצה הרミיטית ($S_{ij} = S_{ji}^*$) מסדר M עם ערכי עצמיים $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$, מתקיים:

$$\begin{aligned} \lambda_m &= \min_U \left\{ \max_{\substack{x \in U, \\ \|x\|=1}} \{x^\dagger S^\dagger x \mid x \in U, x \neq 0\} \middle| \dim(U) = M - m + 1 \right\} \\ &= \min_U \left\{ \max_{\substack{x \in U, \\ \|x\|=1}} \left\{ \frac{x^\dagger S^\dagger x}{x^\dagger x} \right\} \mid x \in U, x \neq 0 \right\} \mid \dim(U) = M - m + 1 \end{aligned}$$

הערך העצמי המקסימלי מקיים:

$$\lambda_1 = \max_{\|\hat{W}\|=1} ((\hat{W}^1)^T \cdot \hat{S} \cdot \hat{W}^1)$$

כאשר \hat{W}^1 , הינו הערך העצמי המתאים ל- λ_1 – ערך העצמי המקסימלי של \hat{S} .

2.3.2 t-distributed Stochastic Neighbors Embedding (t-SNE)

אלגוריתם הורדת הממתק-PCA פועל באופן LINEAR, מה שמקל על תהליכי החישוב שלו, אך מגביל את יכולות ההכללה שלו. אלגוריתם אחר, לא LINEAR, נקרא t-SNE=t-distributional-Nearest-Neighbors, והוא מנסה לקחת את הדadata בממד גובה ועל ידי מיצד אחד למשתמש במאגר נתונים $\vec{X} \in \mathbb{R}^{M \times N}$, כאשר M הוא מספר הדוגמאות, ו- N הוא מספר המאפיינים (או המשתנים). חשוב לשים לב Ci כל מדידה מיוצגת על ידי וקטור שורט \vec{X}_m . הרעיון הכללי של השיטה הוא לmaps את סט המדידות באופן צזה שמדידות דומות יותר, קרי מדידות "קרובות" יותר במרחב ה- N -ממדי, ייצגו על ידי נקודות קרובות יותר במרחב חדש K -ממדי, כאשר לרובה $3 \leq K \leq N$ נסמן את המרחב המקורי \mathcal{X} ואת המרחב החדש \mathcal{Y} , כאשר בשני המרחבים המדידות מוצגות על ידי נקודות בגרף scatter plot (scatter plot). המטריקה המשמשת למדידת דמיון (similarity) בין שתי נקודות במרחב המקורי \mathcal{X} הינה הסתברותית. עבור שתי מדידות m_1, m_2 במרחב המקורי \mathcal{X} , ההסתברות הנורמלית המשוותת P_{m_1, m_2} הינה

$$P_{m_1, m_2} = \frac{\mathcal{Z}_1^{-1}}{2N} \exp \left(-\frac{\|\vec{X}_{m_1} - \vec{X}_{m_2}\|^2}{2\sigma_1^2} \right) + \frac{\mathcal{Z}_2^{-1}}{2N} \exp \left(-\frac{\|\vec{X}_{m_1} - \vec{X}_{m_2}\|^2}{2\sigma_2^2} \right)$$

כאשר i – נקרא perplexity (perplexity) והוא פרמטר שנקבע מראש, כך היה קבוע הנוורמליזציה, המוגדר על ידי $\mathcal{Z}_i = \sum_{k \neq i} \exp \left(-\frac{\|\vec{X}_i - \vec{X}_k\|^2}{2\sigma_i^2} \right)$. עבור נקודות קרובות יותר, עבור הביטוי $\|\vec{X}_{m_1} - \vec{X}_{m_2}\| = \text{קטן}$, ההסתברות

שהנקודות שוכנה של $\vec{X}_{m_1} - \vec{X}_{m_2}$ גדולה. לעומת זאת כאשר הנקודות רחוקות זו מזו, ככל מה שהנקודות \vec{X}_{m_1} קטנה מאוד עד אפסי-ה-הסתברות.

כעת, כפי שהוזכר לעיל, נרצה למפות את סט המדידות \vec{Y}_M כרך שהມמד=של-ה-הינו נמור (2=אך-ממדים). בנוסף, נדרש שנקודות דומות ("שכנות") במרחב \mathcal{X} , ישארו שכנות לאחר המיפוי למרחב \mathcal{Y} =מתבגר שפונקציית ההסתברות המותנית, המתאימה לתיאור דמיון בין נקודות שכנות במרחב החדש \mathcal{Y} , הינה התפלגות, הניקראת גם התפלגות סטודנט עם דרגת חופש אחת (נדון ברענין לבחור בפונקציות הסתברות אלו בהמשך). כרך-

כמות את הדמיון בין m_1 ו- m_2 , על ידי ההסתברות המשותפה Q_{m_1, m_2} המוגדרת באופן הבא:

$$Q_{m_1, m_2} = 3^{-1} \frac{1}{1 + \|\vec{Y}_{m_1} - \vec{Y}_{m_2}\|^2}$$

כאשר $3 = \sum_{k \neq j} \left(1 + \|\vec{Y}_k - \vec{Y}_j\|^2\right)^{-1}$ הינו קבוע נורמלי-ציה.

המייפוי בין מרחב המקור \mathcal{X} לבין המרחב החדש \mathcal{Y} הוא מיטבי אם הוא "משמר" את השכניםות של נקודות (מדידות) קרובות. לשם כך נגידר את פונקציית המחיר על ידי Kullback-Leibler divergence, הבוחן מרחק בין שתיים התפלגיות:

$$C = \mathcal{D}_{KL}(P|Q) \equiv \sum_{m_1} \sum_{m_2} P_{m_1, m_2} \log \left(\frac{P_{m_1, m_2}}{Q_{m_1, m_2}} \right)$$

נרצה למצוא את הווקטור \vec{Y}_{m_i} עבורו פונקציית המחיר מינימלית, ולשם כך נשתמש בגרדי-אנט לפי:

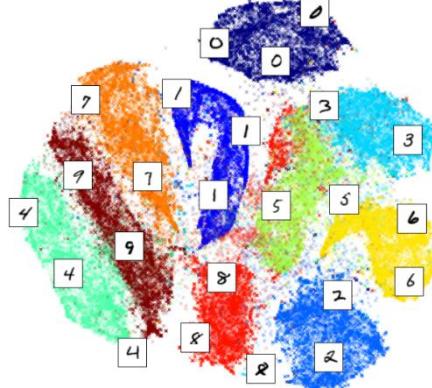
$$\begin{aligned} \frac{\delta C}{\delta \vec{Y}_{m_i}} &= \frac{\delta}{\delta \vec{Y}_{m_i}} \left[\sum_{m_1} P_{m_1, m_i} \log \left(\frac{P_{m_1, m_i}}{Q_{m_1, m_i}} \right) + \sum_{m_2} P_{m_1, m_2} \log \left(\frac{P_{m_i, m_2}}{Q_{m_i, m_2}} \right) \right] \\ &= 4 \sum_{m_1} (P_{m_1, m_i} - Q_{m_1, m_i}) \left(1 + \|\vec{Y}_{m_1} - \vec{Y}_{m_i}\|^2 \right)^{-1} (\vec{Y}_{m_1} - \vec{Y}_{m_i}) \end{aligned}$$

чисוב המינימום באופן אנלטי לא תמיד אפשרי או לא תמיד יעיל, ולכן מקובל להשתמש בשיטה gradient descent, שהינה שיטה איטרטיבית למציאת המינימום של פונקציה (פирוט על שיטה זו ווריאציות שונות שלה מופיע בחלק 4.3.5). עבור הורדת הממד, חישוב המינימום בעזרת שיטה זו יעשה באופן הבא:

- א. אתחול: * נתוך $\vec{y}^{M \times N} \in X$.
- * פרמטר לפונקציית הדמיון: בחירת השונות σ^2 .
- * בחירת פרמטרים לאופטימיזציה: קצב הלמידה η , מומנטופ $\alpha(t)$.
- ב. חשב את P_{m_1, m_2} .
- ג. אתחל את המיפוי $(s, \hat{I}_M) \sim N(0, s\hat{I}_M)$ בחר את הערכים ההתחלתיים לפני התפלגוה גאוסיאנית עם ממוצע 0 וסטיית תקן s (s נבחר להיות קטן, נניח $s = 10^{-4} = \hat{I}_M$ מטריצת יחידה).
- ד. עברו איטרציה t : * חשב את Q_{m_1, m_2} .
- * חשב את הגראדי-אנט של פונקציית המחיר $\frac{\delta C}{\delta y}$.
- * עדכן: $y^{(t+1)} = y^{(t-1)} + \eta \frac{\delta C}{\delta y} + \alpha(t)[y^{(t-1)} - y^{(t-2)}]$.

נעיר כי בשפות תכנות רבות, האלגוריתם עצמו כבר מוגדר על ידי פונקציות מובנות, ויש רק להגדיר את הפרמטרים הדרישים:

במאמר המקורי שהציג את השיטה הובאה דוגמא של שימוש באלגוריתם עבור הטלה של הספרות 0-9, המיצג את על ידי תמונות במדד גבוק $\mathbb{R}^{28 \times 28}$, למרחב דו-ממדי. דוגמא זו נלקחה מ-6,000 תמונות של ספרות ומיפוי אותן למרחב דו-ממדי. במרחב זה ניתן לראות בבירור כיצד כל תמונה מופתעת לאחור אחר, כיוון שבפועל נוצרו עשרה אשכולות שונות, המובחנים בצורה ברורה אחד מהשני. ביצוג הדסמןדי אין משמעות לצירם, כיוקש באלגוריתם זה יש חשיבות רק למרחק היחסיבי בין הנקודות



איזומטריה (יזואלייזציה) דו-ממדית של מערך נתונים עבור כתבי-יד של ספרות (MNIST) על ידי שיטות SNE- t . כל דוגמא $\vec{X}_m = 2.15 -$ הציג (יזואלייזציה) דו-ממדית על ידי $784 \times 28 = 28^2$ עריכים (פיקסלים בגווני אפור) ומסוגת להיות ספרה בז'ט-9. באיזור מזגוגות 6,000 נקודות (מדידות) כלו, כאשר צבעים שונים מייצגים ספרות שונות מלבד ההבחנה בין הספרות, ניתן לראות שספרות דומות קרובות זו לזו גם במרחב החדש (למשל הספרה 1 קרובת הספרה 7, שבתוර קרובה לספרה 9).

כאמור פונקציית הדמיון במרחב המקורית הינה פילו-הנורמל-המשותף של שתיהן קודות, ואילך במרחב החדש פונקציית הדמיון הינה התפלגות t . שתי העורות חשובות על בחירות אלו:

א. סימטריה

פונקציית הדמיון הגאומטרית הבינן ששתי נקודות במרחב \mathcal{X} הינה פונקציה סימטרית-כלומר $P_{m_1, m_2} = P_{m_2, m_1}$. אולם ניתן להגיד גם פונקציית דמיון א-סימטרית, המבוססת על התפלגות מותנת (במקום התפלגות משותפת). הפונקציה המותנת נתונה על ידי:

$$P_{m_1|m_2} = Z_2^{-1} \exp\left(-\frac{\|\vec{X}_{m_1} - \vec{X}_{m_2}\|^2}{2\sigma_2^2}\right)$$

כך ש

$$P_{m_1, m_2} = \frac{P_{m_1|m_2} + P_{m_2|m_1}}{2N}$$

ב. בחירת פונקציית הדמיון במקום פונקציית t

באלגוריתם שתואר, פונקציית הדמיון בין שתי נקודות במרחב ה- \mathcal{X} נתונה על ידי התפלגות t . ניתן להגיד גם פונקציה אחרת, למשל את פונקציית דמיון גאומטרית עבור שתי מדידות במרחב \mathcal{Y} . שיטה זו נקראת SNE, והגדיר את של פונקציית המחרhir במקרה זה נתונה על ידי:

$$\frac{\delta C}{\delta \vec{Y}_{m_i}} = 4 \sum_{m_1} (P_{m_1, m_i} - Q_{m_1, m_i}) (\vec{Y}_{m_1} - \vec{Y}_{m_i})$$

אולם, פונקציית דמיון גאומטרית במרחב \mathcal{Y} יכולה לגרום לכך שנקודות לא מאד קרובות במרחב \mathcal{X} , ימוץلن קודות קרובות במרחב \mathcal{Y} , כיוון שהגאומטריאן בעצם גורם לאטרקטור (משיכה) יחסית חזק בין שתי נקודות, גם במקרים בהם הנקודות אינן מאד קרובות. לעומת זאת, כאשר פונקציית הדמיון הינה התפלגות סטודנט- t , שהינה התפלגות עם זנב כבד יותר, שתי נקודות שאיןן מאד קרובות יMOVFO בצורה ראייה למרחב \mathcal{Y} שכן "נמשכות" או מתקרבות זו לזו. שיטה אחרת, הנקראת SNE-UNI, מציעה להשתמש בתפלגות אחת, אך גם לה חסרן דומה ל-SNE, כאשר שתי נקודות לא מאד דומות זו לזו, אין "דוחות" אחת את

באותן אינטואיטיבי, ניתן לחשב על גרדיאנט פונקציית המחיר כשדה כוח, ועל פונקציית המחיר כטור פוטנציאלי, קר שהכוח הפועל הוא בעצם כוח קפיץ.

לשיטת t-SNE יש שלוש מגבלות עיקריות

- א. הורדומגדה השיטה ממשמשת ליזואיצ'ר של מידע מממד גבורה-בגדים מגד אולם-באוק עקרוני, יתכן ונרצה להוריד את הממד לא שם הצגתן, אלא לצרכים אחרים, כאשר הממד החדש הינו גודל מ-3= t -טיקוב-בגדים גבורה-פונקצי'ר התפלגוז-טונקס'ע דרגת חופש אחת, אשר לה משקל גבוה יחסית במרקחים גובהים, לא תשמיר את המבנה של המידע המקורי. לכן, כאשר נרצה להוריד לממד גובה מ-3= t -פונקצי'ת התפלגות \neq עם יותר מדרגת חופש אחת מתאימות יותר.

ב. קילת המדדיות=SNE- t -מבוססת על מאפיינים מקומיים בין נקודות. השיטה, המבוססת על מטריקת מרחק אוקlidית, וכר מניחה לינאריות מקומית על גבי היריעה המתמטית בה מתק'יות הנקודות. אולם, במרחב נתונים בו הממד הפנימי גבוה, שיטות=SNE- t -עלולה להיכשל כיוון שהנחה הלינאריות לא מתק'ית. למרות שישנן מספר שיטות למצער תופעה זו, עדין, בהגדירה, כאשר הממד הפנימי גבוה, לא ניתן להוריד ממד t שמבנה המידע ישמר באופן מלא.

ג. פונקצי'ת מחיר לאקמורה: הרבה שיטות למידה מבוססות על פונקצי'ת הפסד קמורה, כר שתיאורטיות מציאות אופטימיזציה (יחידה) לפונקציה זו אפשרית תמיד. אולם, בשיטות=SNE- t , פונקצי'ת המחיר אינה קמורה, והפתרון המתkeletal על ידי האופטימיזציה משתנה בהתאם לפרמטרים הנבחרי'.

2.4 Ensemble Learning

2.4.1 Introduction to Ensemble Learning

נניח כי יש בידינו אוסף נתונים מוסכמים, ורוצים לבנות מודל המנתה את הנתונים האלגוריתם מסויים – כמעט תמיד, המודל לא יהיה מדויק במידהacha, והוא יהיה בעל שנות או בעל הטיה. נזכיר להשתמש במקרה תמיד, המודל יושנו בהתאם למבוקשים. על אותו אלגוריתם רצויו – בוכך לקבל מודל משוקל-בעל שנות/הטיה (Ensemble) של מודלים.

בכדי להבין אפקיטור טוב את החישובו של שילוש-ensembles, יש להרחיב עליה-Trade-off בין שונת המודל להטיה שבו. מודל אופטימלי יתאפשר בשונות נמוכה ובhetia גבוהה. לעומת זאת, השוני בין התוצאות לא יהיה מהותי ובמוצע התחזית תהיה קרובה מאוד לערך האמתי. מודל כזה יהיה מודל אמיק-ונוכלבלסס עליון את צעדינו. למרבה הצער, מודל שכזה לרוב האינטראקציות אחר של מודל יהיה המודל הגראן, ההפוך למודל האופטימלי. זהו מודל עם שונות גבואה והטיה גדולה. מודל שכזה יציג טווח רחב של תוצאות על נתונים, ובממוצע יהיה רחוק מאוד מהערך האמתי. מודל זה כל אינטראקציית.

בפועל, המודלים במציאות ינעו לאורך שני קצוטות: מודלים עם שונות גבוהה והטיה נמוכה, ומודלים עם שונות נמוכה והטיה גבוהה. הזרחי של המיקום שלנו לאורך ציר זה קריטי, כיוון שהוא מאפשר לנו לבחור את דרך ההתמודדות הטובה ביותר **by voting**-מספר משפחות של model ensembles-ושני העקריים שבהם נקראים Bagging and Boosting כאשר-ניתקל במודלים עם שונות גבוהה, לעומת מודל הסובב-Overfitting, לרוב נרצה להשתמש באנסטבל מסווג-**Bagging**-מנת להוריד את השונות במודל הסופי. אלגוריתם מסומן-**Boosting**-יטפל במקרה השני, בו הטיה גבוהה ומשונות ומינימלית.

2.4.2 Bootstrap aggregating (Bagging)

Bagging היא משפחת אלגוריתמים אשר פועל-כ-ensemble – כלומר מספר אלגוריתמים שפעלים יחד, על מנת להגיע לתוצאה מושפרת=**overfitting** אלגוריתמים מסווגים נועדו להגדיל את יציבות המודל והעלאת הדיק שלו, זאת תוך הורדת השונות והימנעות מ-**overfitting** מרכיב ממוצע רב של אלגוריתמים, המכונים "לומדים חלשימים" (Weak learners), כאשר כל אחד מהם מבצע למידה ותחזית על חלק מן הנתונים; מטרתה להגיעה לתוצאה אינטואיטיבית=**bagging** הינה נשיטה נפוצה ו פשוטה יחסית לשיפור ביצועים, אם כי היא עשויה

אלגוריתמים מסוג bagging מוגבלים במתהיל בפחות מאשר שדריכים: Bootstrapping and Aggregating.

בשלב ה-Bootstrapping יוצרים מהנתונים המקוריים קבוצות חדשות, כאשר כל קבוצה נוצרת על ידי דוגמה (עפ"י נסיבות מחרוזת מה樣ותה המקוריית. אופו צה שגודל כל קבוצה חדשה הוא בגודל של הדוגמאות המקוריות) של איררכיות מהירשרות המוקוראליות.

השני, Aggregating-hbkatzot chadashot-enkenpo-ekklat l'lo'madim chalshim", אלגוריתמים פשוטים יותר, אשר עובדים במקביל על תחזית, כולם יוצרים מודל נפרד לכל קבוצה של נתונים מסוימת בשלה הסופי=יתבצע איחוד של כל המודלים על מנת ליצור מודל משוקל בעל שונות קטנה יותר מאשר מודל המסתמך על הדאטה המקורי כפי שהוא

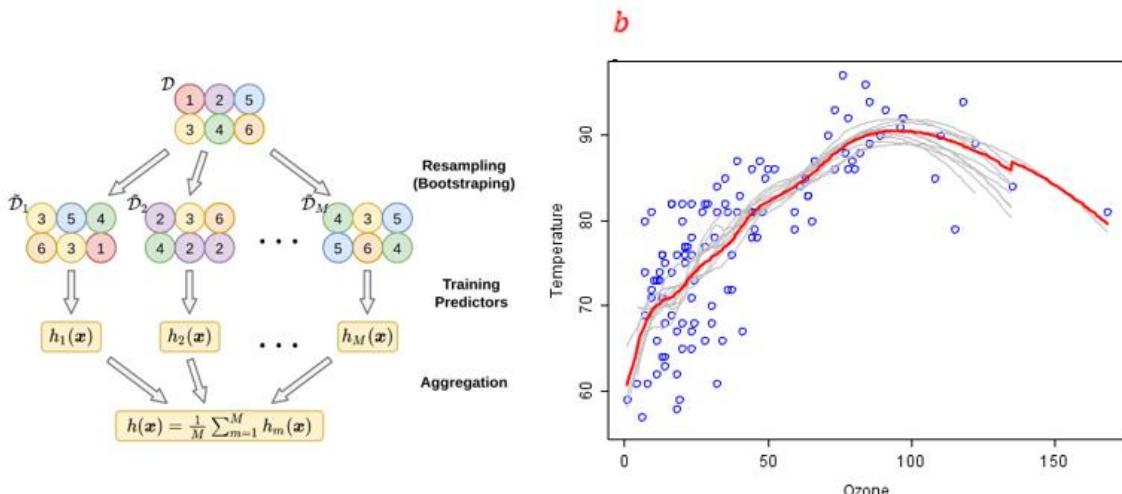
עפונ חיבור המודלים הפתבע ב-*bagging* מבօסס על אותו רעיון של NN-K, כאשר מודלים אלו יכולים לשמש הן למטרות סיווג והן למטרות גרסיה=כאמור לעיל (פרק 2.1.3), באלגוריתם השכן הקרוב כל "שכן" העד על התוויות שלו, ולאחר הכרעת הרוב נקבעה התוויות של התצפית החדשאה. במקורה שבו נספר את תדיות כל התוויות השכנות, והтоויות הנבחרת תהיה של התצפית הנפוצה ביותר-נעשה זאת כASH=NN-K. יעבדו כמסוג. במקרים בהם NN-K יעבד כגרסתה, יבצע ממוצע של כל התוויות השכנות, וזאת גם תהיה התחזית. כאשר *bagging* יעבד כמסוג, כל *weak learner* יבצע תחזית, והתוויות השכיחה ביותר תהיה התוצאה של האנסמבל-הכרעת הרוב. כASH=bagging יעבד כגרסתה, כל מודל יבצע תחזית, אבל התוצאה של האנסמבל תהיה הממוצע של כל המודלים.

באופן פורמלי, עבור DATA $\in \mathbb{R}^{n \times d}$, ניצור M -קבוצות חדשות=באותן גודל של הדאטה המקורי=
עבור כל קבוצה m נבנה מודל $c_m(x)$ =עבור בעיוקסיו ובה החליטה תתקבל על פיה הצבעת
הברך.

$$C(x) = \text{majority}(\{c_1(x), \dots, c_M(x)\})$$

ועבור בעיות רגסיה ההחלטה תבוצע באמצעות מיצעך המודלים:

$$C(x) = \frac{1}{M} \sum_{m=1}^M c_m(x)$$



16. ג' אלגוריתם Bagging – בשלב ראשון יוצרים הרבה מחלקות שונות מהדטה המקורי (Bootstrapping) ולאחר מכן יוצרים מודל המתאים לכל מחלקה (Aggregating), ולבסוף יוצרים מודל יחיד המבוסס על כל המודלים הקודמים. (ז) דוגמא לבניית מודל לרגרסיה בעזרת אלגוריתם Bagging. ניתן לראות שהמודול המשוקל הוא בעל שונות קטנה יותר מכל שאר המודלים.

בין אם משתמשים בהכרעת הרוב ובין אם משתמשים במיינר של המודלים, המודל המשוקל שנוצר הופך להיות חלק יותר ובעל פחות שיפורים חדים, מה שמקטין את overfitting, ומילא מפקת את השונות. ניתן להבין זאת על ידי דוגמא פשוטה – נניח שיש התפלגות נורמלית (μ^2, σ^2) , אז השונות שתקדיגות בלתי תלויות הינה $\frac{\sigma^2}{n}$.Cut

$$Var\left(\frac{1}{m}\sum_{i=1}^m \text{single cycle}\right) = \frac{1}{m}(1-\rho)\sigma^2 + \rho\sigma^2$$

אם נבצע הרבה מאוד ניסויים, ככלומר ניקוח גדול מאוד, נקבע

$$\lim_{m \rightarrow \infty} \frac{1}{m} (1 - \rho) \sigma^2 + \rho \sigma^2 = \rho \sigma^2$$

ובסך הכל השונות הסופית הינה bagging ² סק, וביתוי זה לרוב קטן מאשר השונות של מודל-המבוסס על הדאטה המקורי ללא שימוש ב-ensembles. ניתן לשים לב שכלל שהקורסיבית בין הקבוצות-קפטנה, כך השונות של המודל המשוקלל גם כן קטנה יותר.

מודל נפוץ מאוד מסוגי bagging -הוא Random Forest . אלגוריתם זה משלב ב- bagging החלטה בין הרעיון הבסיסי של bagging , כאשר הוא מפעיל את הנתונים ואת המשתנים לעצמי החלטה ריבים-ולכל אחד מהם מקבל חלק מסוים מן השלם. העצים הם בעלי שונות גבוהה, ככל אחד מהם הוא overfitting - себנוי עצמו, אך עם זאת הקורלייזציה ביןיהם נמוכה, מה שמקל על הורדת השונות והימנענות מ- overfitting -לבסוף, השקלול של כל המודלים ביחס-מצליה לייצר מודל בעל שונות נמוכה, ומוביל לתוצאות טובות

ל- bagging יתרונות רבים. הוא מוריד את השונות, והוא גם חסין לעריקים Outliers . יכולת העבודה של bagging במקביל עשויה לאפשר לו להגיע לתוצאות באופן מהיר יותר

עם זאת-ל- bagging יש גם חסרונות. הוא אינו מוריד את ההטיה, ולכן עשוי לא להתאים במרקם רבים. במודלים של בינה מלאכותית יש חשיבות רבה ליכולת הפרשנות של המודל-לראובן, ידרש הסבר פחות טכני של תוצאות המודל למבקלי ההחלטה או לצרכינם. הם עשויים לא לקבל כלל החלטות של מודל שראה כ"קופסה-שchorah". יש קושי רב לתת פרשנות-להחלטות-של-מודל-המצביע bagging , ומדובר מקשה על השימוש בו. מעבר לכך bagging עשוי להיות קיר מבחן חישובי. עקב כך, הוא שימושי מאוד במרקם-הפה bagging שיפור זעיר עשוי להוביל להצלחה, אך-

תינון עדיפות למודלים פשוטים יותר של ensembles

2.4.3 Boosting

כאמור-המושג boosting למשחתת אלגוריתמים המשמש-באוסף של-מודלים "חלשים" על מנת ליצור מודל אחד "חזק", כאשר מודלים אלו מתמקדים בניסיוק-הפקית את ההטיה שיש למודל. מבחינה אינטואטיבית, מודל חלש הוא זהה שתוצאותיו מעט טובות יותר מ nichos אקראית בעוד שאותו חזק מתקרב לביצועים אופטימליים. בינו-גוד לטכניקות-ensemble-boosting-shallow שפועלות במרקם, העקרון המנחה כאן הוא לשרש את המודלים באופן גזע-של-מודל שמתווסף-יטפל בשגיאות שקדמי פספסו. היופי נועז בכך ש- boosting -ומוכיח כי למידה חלה בהכרח מצבעה על קיומ של שיטת למידה חזקה, לרוב, מודל-combining בועיות סיג בינה-ensembles

באופן פורמלי-המושג $\text{"learned from scratch"}$ - "לומד חלש" ו- "לומד חזק" עברו בעית סיג בין-modular א-algorithms נקרא לומד חזק אם $\text{L}(\hat{x}, \hat{y}) < \delta$, ϵ האלגוריתם מסוגל (\hat{x} עבור אוסף נתונים גדול מספיק) לבנות מסוג (x, c) שמקיים $\hat{y} = \sum_{i=1}^n c_i(x_i)$ ביחס-תברות גדולה מ- δ - 1. לומד חלש הינו א-algorithms של- $\text{L}(\hat{x}, \hat{y}) > \delta$ שעבור אוסף נתוני מספיק גדול, האלגוריתם מסוגל לבנות מסוג שמקיים $\hat{y} = \frac{1}{2}(x, c)$ בהסתברות גדולה מ- δ - 1. כאמור, המטריה של-boosting-הינה לחת אוסף של מודלים חלשים ובעזרתם ליצור מודל חזק, כאשר הצליחו להוציא-השנית להפוך כל לומד חלש לומד חזק על ידי בניית קומבינציה לינארית של מסוגים אשר נוצרו בעזרת הלומד החלש.

נמחיש את הרגעון של-boosting-בឧ-dogma: נניח שיש בידינו אוסף נתוני X , המחולק באופן אקרא לשולש קבוצות שווות (כל אחת מכילה שליש מהנתונים) $= x_1, x_2, x_3$, x_1, x_2, x_3 מודל לצורך סיג בינה-המסומן ב- \hat{y}_h . נמצאות $\frac{1}{3}$ מודלים באורה טובה רק לקבוצה x_1 , $\frac{1}{3}$ מודלים באורה לא טובה את פרטיה הקבוצות x_2, x_3 ש- \hat{y}_h משליך מסך הנתונים-שגייאת הסיג גודלה- (x_1) מודל חלש, ונרצה לשפר אותו-בכך לעשות זאת ניקח רק-חלקה מהנתונים- X $\in X$, ונדאג איך-יכל הרבה מאייר- x_3 . כתע נבנה מודל ניסוף (x_2) על בסיס- X -מתקור כונה שמודל זה יתמקד גם בקבוצת x_3 א-ויסוג את איבריה באורה טובה. כתע נניח שמודל זה אכן מסוג בaczora נאותה את אייר- x_3 , אך הפעמה-מודל שוגה באורה גסה בסיג-אייר- x_2 . עקיב השגיאה בסיג- x_2 מהמודל השני גם הוא מודל חלש, אך כתע יש בידינו שני מודלים חלשים שהוחולשה בכל אחד-נובעת מקבוצת איברים אחד של אוסף הנתונים-המקור X . אם נמצא דרך הולמת לחבר את שני המסוגים, יוכל ליצור מודל בעל פוטנציאל להצליח לסוג את X כמכך שציר.

באופק-כללי, אם רוצים לאמן מודל (x, C) בעזרת אלגוריתם L על אוסף הנתוני \mathcal{D} , יש לבצע את השלבים הבאים:

$$1. \text{ אתחול הנתונים: } \mathcal{D}_1 = \mathcal{D}$$

$$2. \text{ עבור } T = 1, \dots, \#$$

$$\text{אימון מודל חלש ערך: } c_t(x) = L(\mathcal{D}_t) = \mathcal{D}_t$$

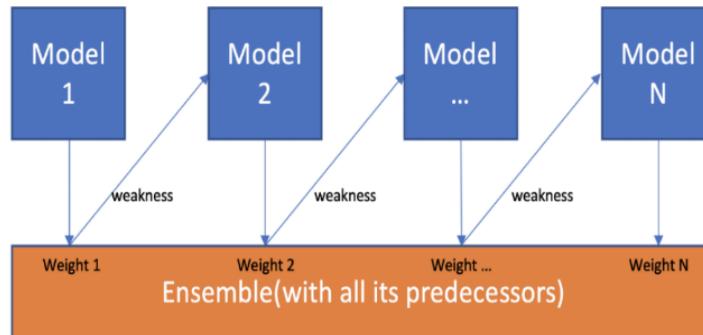
$$\text{чисוב שגיאת המסוג: } \bar{c}_t = P_{x \sim \mathcal{D}_t}(c_t(x) \neq f(x))$$

. $\mathcal{D}_{t+1} = \text{Adjust Distribution}(\mathcal{D}_t, \epsilon_t)$ התאמת הנתונים עבור האיטרציה הבאה=

. $C(x) = \text{combine outputs}\{c_1(x), c_T(x), \dots\}$ איחוד המודלים החלשים: { $c_1(x), c_T(x), \dots$ }

יש כל מיני שיטות כיצד לבצע את השלבים השונים באלגוריתם boosting, ונפרט את המרכזיות שבזה:

Model 1,2,..., N are individual models (e.g. decision tree)



איו-פ-17. --סכמה כללית של boosting. המודלים (במקרה זה מוחבר בעץ החלטה רדו), אך זה תקין לכל מודל חלש) מוחברים אחד לשני באופן שכל אחד לומד מהתפלגות המשוקלת בהתאם לשגיאות של המודלים הקודמים.

Adaptive-Boosting (AdaBoost)

Adaboost היא אחת הטכניקות הראשונות של boosting, ועל אף שקיימות טכניקותboosting-עכשו, היא עדיין הפופולריות ביותר בתחום (אם כי יש לה מסגר לא מבוטל של וריאנטים). העצמה הכלומה בטכניקה זו נובעת מכך שגם בהינתן מספר מוגבל של אלגוריתם מצליח להיגע פחות מ"קללה המדדיות" ולשמור על יכולות ניבעות, בינו-גוד לאלגוריתמים אחרים של סיווג, כמו למשת-**SVM** או אפיילו רשתות נירונית.

זכור, תחת ההנחה שהקיים אלגוריתם לומד חלש($c(x)$)=המטרה היא למצוא דרך להפוך אותו למודל חזק($C(x)$)=באופן אינטואיטיבי היה ניתן לחושב שאפשר פשונelly-על תות קבוצות של הדאות המקור-עכשו אפשרות לחיפוי ביחס-קבוצות(=להשתמש ב-**vote-majority**=ובכך לשרש את ההיפותזות של כל המודלים לפלאט אחד. גישת זו מובילה-בנוסף לספקת מקרה במספרית המודלים שוגים=גישה טוביה יותר תהיכ-לבנות מודל על בסיס חלק מהדאות-לבוחן את מידת הצלחה של המודל על יתר הדאות-ולפ-ההצלחה של-במשימה זו לסתות מושך **AdaBoost** המודלים הקודמים שגו בסיווג שלם=ובכך-בכל שלב ינתן יותר גודלה מאשר המודל הקודם-חלק זה הינה-האלגוריתם (Adaptive) באלגוריתם, על שמו נקרא האלגוריתם.

cut, נסביר כיצד ניתן להרכיב מסווג חזק באמצעות אוסף של מסווגים חלשים עבור אוסף נתונים $\mathbb{R}^N \in X$.

1. ראשית יש לאותל משקלות באופן אחד עבור כל אחת מ- N הדוגמאו-בסט הנתונים $w_i^{t=0} = \frac{1}{N}$
2. לאחר מכן יש לבצע איטרציות באופן הבא:

בנין מסווג אופטימלי($c_t(x)$) c_t ביחס לאוסף הנתונים המשוקל:

$$\text{חישוב שגיאת הסיווג של } (c_t(x)) : c_t(x) \neq y_i \} = \epsilon_t = \sum_i w_i^t \{ c_t(x)$$

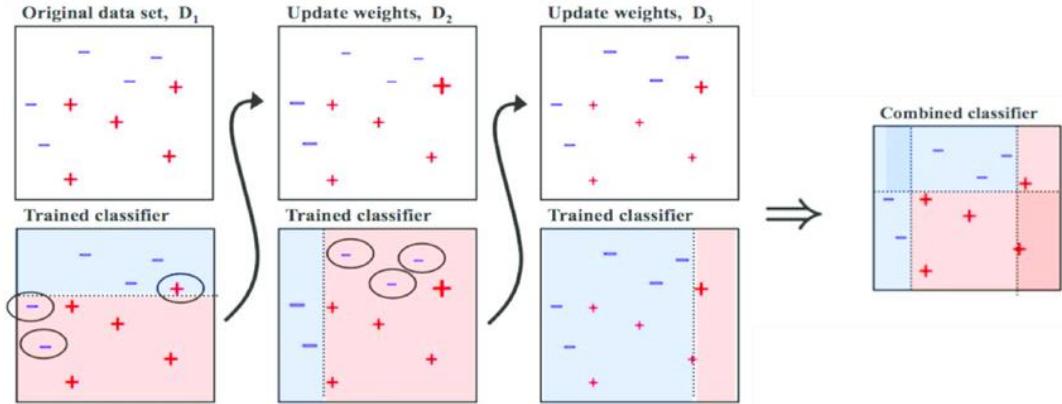
$$\text{חישוב משקל עבור מסווג זה: } \alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$$

$$\text{עדכון המשקלים: } w_i^{t+1} = w_i^t \exp(-\alpha_t y_i c_t(x_i))$$

$$\text{נرمול המשקלים בהתאם לsoftmax הכלול: } N_{t+1} = \sum_i w_i^t \rightarrow w_i^{t+1} = \frac{w_i^t}{N_{t+1}}$$

3. חישוב המשׂוג המשׂוקל, שהוא קומבינציה לינארית של המשׂוגים החלשים:

$$C(x) = \text{sign} \left(\sum_t \alpha_t c_t(x) \right)$$



איור 2.18 – דוגמא לשימוש ב-AdaBoost עבור מודל סיווג בינארי

2. References

SVM:

https://commons.wikimedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

<https://svm.michalhaltuf.cz/support-vector-machines/>

<https://medium.com/analytics-vidhya/how-to-classify-non-linear-data-to-linear-data-bb2df1a6b781>

https://xavierbourretsicotte.github.io/Kernel_feature_map.html

Naïve Bayes:

https://en.wikipedia.org/wiki/Naive_Bayes_classifier

https://scikit-learn.org/stable/modules/naive_bayes.html

K-NN:

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

EM:

https://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/13_mog.pdf

https://stephens999.github.io/fiveMinuteStats/intro_to_em.html

Hierarchical Clustering:

<https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/>

LOF:

<https://towardsdatascience.com/local-outlier-factor-lof-algorithm-for-outlier-identification-8efb887d9843>

PCA:

Saal, L.H. et al. (2007). *Proc. Natl. Acad. Sci. USA* 104, 7564–7569.

3. Linear Neural Networks

פרק זה עוסק בבעיות רגראטיות – כיצד ניתן בעזרת אוסף נתונים לבנות מודל המסוגלת לספק מידע על נקודות חדשות שיגיעו ויחסר עליהן מידע מהמודל? פירושו בפרק זה מת'יחסים לדעתה השניהן למצוא עבורי הפרדה לינארית – כולם נניתן למצוא קווים לינאריים מהחלוקת את הדאטא לקבוצות שונות – החלק הראשון של הפרק יעסוק ברגראטיה לינארית (Linear regression) והחלק השני יעסוק ברגראטיה לוגיסטייה (Logistic regression). כלומר יוצג מבנה שקול לביעות הרגראטיה בעזרת רשת נירוניים פשוטה, ומבנה זה יהיה הבסיס לפרך הבא העוסק ברשתות נירוניים עמוקות, הבאות להתמודד עפ"ד נתונים שאין לבצע עבורי הפרדה לינארית.

3.1 Linear Regression

3.1.1 The Basic Concept

המודל פשוט ביזור-הינך-linear regression מודל זה מנסה למצוא קשר לינארי בין מספר משתנים או מספר מאפיינים – בהנחה שמתוך יחס לינאריביותם משתנים בלתי תלויים \mathbb{R}^d אל המשתנה תלך \mathbb{R} עניתן לכתוב את הקשר ביניהם בצורה הבאה:

$$\hat{y} = w^T x + b = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b$$

כאשר \mathbb{R}^d הם המשקלים – w , b נקרא bias.

דוגמה ניתן לטעון כמahir הבטים באזורי מסויים נמצא לינאריביליטי-פרמטרים – גודל הדירה – איזה קומה היא נמצאת – כמה שנים הבניין ק"י. תחת הנחה זו, יש לבחוק את המודל עבור דוגמאות ידועות ובכך למצוא את המשקלים וbias. לאחר מכן ניתן יהי-קל-להשאפת המודל ולוחש את מחיר הדירה עבור בתים שמחירים לא ידוע, אך הפרמטרים שלהם כן נתונים.

בכדי לבנות מודל המאפשר לשער בזורה טובה את y בהינתן סט מאפיינים, יש לדעת את המשקלים bias. כיון שהם לא ידועים, יש לחשב אתם – בעזרת אוסף של דוגמאות ידועות. ראשית יש להגדיר פונקציה מחיר (Loss) – הקובעת עד כמה הביצועים של מודל מסוים טובים – פונקציית המחיר היא פונקציה של הפרמטרים הנלמדים – (w, b) , והבאתה למינימזציה את הערך האופטימליים של המשקלים וbias – פונקציית מינימוקובלת הינה השגיאה הריבועית הממוצעת (MSE) – המחשבת את ריבוע ההפרש בין החיזוי לבין הפלט האמתי:

$$L^{(i)}(w, b) = \frac{1}{2}(y_i - \hat{y}_i)^2$$

כאשר נתונות y – דוגמאות ידועות, יש לסקום את כל ההפרשים הללו:

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2}(y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n \frac{1}{2}(y_i - w^T x_i - b)^2$$

כעת בשביל למצוא את הפרמטרים האופטימליים, יש למצוא את b , w שմבאים את פונקציית המינימום:

$$\hat{w}, \hat{b} \equiv \hat{\theta} = \arg \min L(w, b)$$

עבור המקרה הכללי $b = b$ – כולם יש מאפיין יחיד ומונוטם למצוא קשר בין פלט מסוים – הקשר הלינארי הוא $y = ax + b$. עבור המקרה זה, פונקציית המחיר תהיא

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2}(y_i - w^T x_i - b)^2$$

ובכדי למצוא אופטימום יש לגזר ולהשווות $\frac{\partial L}{\partial \theta} = 0$:

$$\frac{\partial L}{\partial w} = \frac{1}{n} \sum_{i=1}^n (y_i - w^T x_i - b) \cdot (-x_i) = 0$$

$$\frac{\partial L}{\partial b} = \frac{1}{n} \sum_{i=1}^n (y_i - w^T x_i - b) \cdot (-1) = 0$$

מתקבלות סט משוואות לינאריות:

$$\begin{aligned} w \sum x_i^2 + b \sum x_i &= \sum y_i x_i \\ w \sum x_i + bn &= \sum y_i \end{aligned}$$

ובכתיב מטריציוני:

$$\begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{pmatrix} \begin{pmatrix} w \\ b \end{pmatrix} = \begin{pmatrix} \sum y_i x_i \\ \sum y_i \end{pmatrix}$$

על ידי הצבה של הדוגמאות הנתונות ניתן לקבל את הפרמטרים של הקשר הלינארי:

לשם הנוחות ניתן לסמן את \mathbf{w} כפונקציית המבחן נספח

$$\hat{y} = \mathbf{w}^T \mathbf{x} + b = (\mathbf{w}^T \mathbf{b}) \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}, \quad \tilde{\mathbf{w}}, \tilde{\mathbf{x}} \in \mathbb{R}^{d+1}$$

עבור המקרה הוקטוריף דוגמאות כלומר ישוף מאפייני בלתי תלויים ומנסים למצאו את הקשר ביניהם לביצ' פלט מסוים. במקרה זה $\mathbf{z}^T = (x_1, \dots, x_n)^T, Y = (y_1, \dots, y_n)^T$

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

המינימום של הביטוי הזה שקול למינימום של $\|Y - X\mathbf{w}\|^2$

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}^T \mathbf{x}_i) \cdot (-\mathbf{x}_i) = 0$$

$$\rightarrow X^T (X\mathbf{w} - Y) = 0$$

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T Y$$

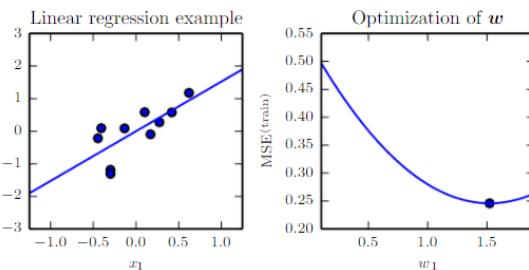
ובה ניתן אוסף דוגמאות

$$X = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}, \quad \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

אזי הפתרון של הרגרסיה הלינארית הינו:

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T Y = \begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum y_i x_i \\ \sum y_i \end{pmatrix}$$

דוגמה למציאת קו הרגרסיה והמשקל האופטימלי עבור בעיה סקלרית:



איור 3.1 רגרסיה לינארית אופטימלית עבור אוסף דוגמאות נתון (שמאל) ואופטימיזציה עבור המשקל w ביחס לפונקציית המבחן (ימין)

3.1.2 Gradient Descent

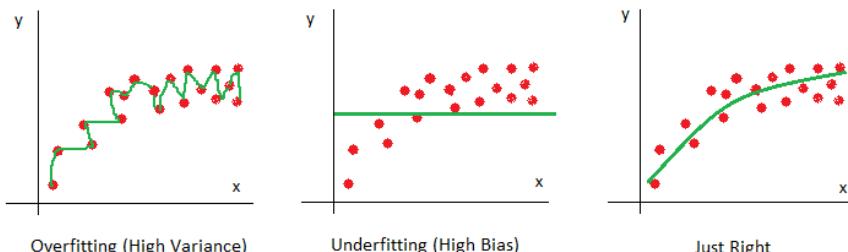
הרבה פעמי-המציאת המינימום של פונקציית המחיר היא משימה קשה. דרך מקובלת להתמודד עם חישוב ה-**האופטימל=היא שיטה**=gradient descent (GD). בשיטה זו מפעילים מינוחש מסוים עבור הפרמטרים, וכל פעם מבצעים צעד לכיוון הגרדי-אנט-השלילי. הגרדי-אנט הוא הנגזרת של הפונקציה, והוא מגדיר את הכוון שערך הפונקציה עולה בו בaczורה מקסימלית=אם לקובחים את הכוון השלילי של הגרדי-אנט-בעצם הולכים לכיוון בו יש את הירידה הכי גדולה, ולבסוף הגיעו למינימום=יש לבצע את הצעד בכיוון הגרדי-אנט השלילי=בכדי להימנע מהיקלעות לנקודת אוכף, מושגים איבר הנקרא=Learning rate (lr) (פISONן באוטה)=מביצעים את הגירה ושינוי הפרמטרים באופן איטרטיבי עד נקודת עצירה מסוימת=באופן פורמלי=עבור ניחוש התחלתי θ_0 , בכל צעד=מבצע הקידום באופן הבא (העדרון מתבצע באופן סימולטני עבור כל θ_j)

$$\hat{\theta}_{j+1} = \hat{\theta}_j - \epsilon \cdot \frac{\partial}{\partial \theta_j} L(\hat{\theta}_j)$$

קיודם זה יבוצע שוב ושוב עד התכנסות לערך מסוים=כיוון שהבעיה קמורה מובטח שתהיה התכנסות למינימום, אך היא יכולה להיות איטית עקב צעד ערך ערך גודלים או קטנים מדי=פרמטרה=ה- ϵ =learning rate=է, קובע את קצב ההתכנסות, אך רצוי לבחור פרמטר לא קטן מדי כדי לא להאט את ההתכנסות ולא גדול מדי כדי למנוע ההתכנסות.

3.1.3 Regularization and Cross Validation

אחד-האתגרים המרכזיים של בעית הריגסיה (ושאר בעיות הלמידה) הוא לפתח מודל שייהה מוצלח לא רק עבור אוסף-הדוגמאות הידע- \hat{S} האימון, אלא-שייהה מספיק טוב גם עבור דוגמאות חדשות ולא מוכחות=-(קבוצת מבחן). כל מודל יכול לסביר מהטיה לשוני כיווני=Underfitting=Overfitting=היא מצב-בנינתי-הערכות יתר לכל נקודת-בסט האימון-מה שגורר=מודול גבו-ה-ב-בעל-שונות גודלה-ב-ב-מצב זה המודל מתאים ורק- \hat{S} האימון, אך הוא לא מצליח להסביר גם נקודות חדשות-היא המצב הפוך=Underfitting=מודול שלא מצליח למצואו קו מגמתה המכיל מספיק מידע על הדוגמאות הנתונות, ויש לו רעש חזק.



איו-ה- \hat{S} = \hat{S} -נתינת מסקנית-תכליל נקודת-ה-גבורות גבורה- \hat{S} (שמאל)=Underfitting=מודול-ב-על-רערש חזק-מייצג בaczורה מספיק טובא את המידע (אמצע). מצב מאוזק=מודול-ב-על שגיאה מינימלית, המתאר בaczורה טובא את המידע, ובנו-סף נמנע משגיאת יתר עבור דוגמאות חדשות- \hat{S} (ימין).

בכדי להימנע מהטויות אלו, יש לבצע regularization=הוספת אילוץ המונע מהמודול-ל-הוות מותה באופן הפגע בתוצאות. לאחר הוספת האילוץ, פונקציית המחיר תהיה בaczורה:

$$\text{Regularized Loss} = \text{Loss Function} + \text{Constraint}$$

יש מספר דרכים לבצע את ה-regularization:

Ridge Regression / L2 Regularization

דרך אחת לבצע את ה-regularization היא להוסיף איבר נוסף המתיחס לריבוע הפרמטרים:

$$L(\theta) = \text{MSE}_{\text{train}} + \lambda w^T w = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (y_i - w^T x_i)^2 + \lambda \|w\|^2$$

cutת האופטימום של הביטוי הינו:

$$\hat{w} = (X^T X + \lambda I)^{-1} X^T Y$$

הוספת האילוץ גורמת ל-**השיטות** ל训דר את ריבוע הפרמטרים, ובכפלנות להקטין עד כמה שניתן את הערך של פרמטר אלה מנגע מצב בו נתונים משקל יתר לחלק מהפרמטרים, על מנת לא ל訓דר את השונות של המודול וכבר עשוי להיות למונע **overfitting**.

Lasso / L1 Regularization

דרך נוספת לבלצע את ה-Regularization היא להוציא אילוץ המתיחס לערך המוחלט של הפרמטרים:

$$L(\theta) = \text{MSE}_{\text{train}} + \lambda|w| = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (y_i - w^T x_i)^2 + \lambda|w|$$

הוספת האילוקומינריה את סכום הפרמטרים של היערכמה שיתפרק טריך למספר כמה שניות את פונקציית המחדיר בפועל אליו זכם ביא ל"רידוז-משקלים" (sparse), כולם כופה חלק מהמקדמים להיות אפס וכך למעשה יש מעץ feature selection – בחירה הפרמטרים המשמעותיים יותר

ניתן לשים לב כיעבו=2.6 ההפועה של המשקלים על פונקציית המחיר היא ריבועית= לכן במקרה זה הרגולריזציה תשאך להקטין את הפרמטרים הגדולים, ובאופן כללי תנסה לדאוג לכך שכל הפרמטרים יהיו קטנים, ובאותו סדר גודל=1.5 לעומת זאת שואך להקטין את כל האיברים כמה שייותר ללא קשר לאורכם, ולהקטנת פרמטר מסוים מ-0-ב-ל-9 יש אותה השפעה כמו הקטנה של פרמטר מ-1000-ל-999. לכן במקרה זה הרגולריזציה תגרום לפרמטרים הפחות חשובים להתאפס. והמודול נהייה פשוט יותר

Elastic Net

ניתן לשלב ב-**Ridge Regression**=בליברְגְּסָס, ובכך לנסוט לכוון את המודל עברו היתרונות של כל שיטה=גאך להימנע מנתינונ=משקל יתר=פרמטרים וגמ=ণיסין לאפס פרמטרים, בכך=לקבל מודל פשוט ככל הניתן=פונקציית המחיר במרקבה זו תביה מהצורה:

$$L(\theta) = \text{MSE}_{\text{train}} + \lambda \|w\|^2 + \lambda_2 |w|$$

עבור כל מודדים, יש למצאו את הפרמטר λ להאופטימלי עבור ה- m -loss (במקרה של Elastic Net regularization).
הפרמטר λ הוא למשה וקטורי $\lambda = [\lambda_1, \lambda_2]$. שיטה מקובלת למציאת λ היא **Cross validation**:
חלוקת והדוגמאות של n האימוקל-פתק בקבוצות-איימון כל תתי הקבוצות בלבד אחת, ואז בדיקת הפרמטרים שהתקבלו בשלב האימון על הקבוצה שנותרה. בכל איטרציה מוצאים חלק מסויף הדוגמאות והופכים אותן לקבוצת מבוחן-וכך מוצאים את הפרמטר λ להאופטימלי המונע מהמשקלים להגעה-**fitting** (בדרך כלל לוחכים את הממוצע של כל λ מכל האיטרציות). נפוץ השתמש $k=5$, ולמשה עבור בחירה טיפוסית λ^* זיהוי-**AIC** איטרציות, שבכל אחוז מההאימון יבוצע ערך- 80% אפסט האימון, ולאחר מכן תבצע הבדיקה של הפרמטרים שנלמדו על-ה- 20% הנומרי.

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

איור 3.3 Cross validation עם חלוקה ל-5 קבוצות ($k = 5$). בכל פעם קובוצה אחת משמשת ל-validation (הקובוצה הכהולה).

בחירה של $k = n$ נקראת **leave-one out cross validation** שלמענה בכל איטרציה ישוגמא אחת בלבד נכללת בסט האימון ועליה ממבצעת הבדיקה של הפרמטרים שנלמדו

3.1.4 Linear Regression as Classifier

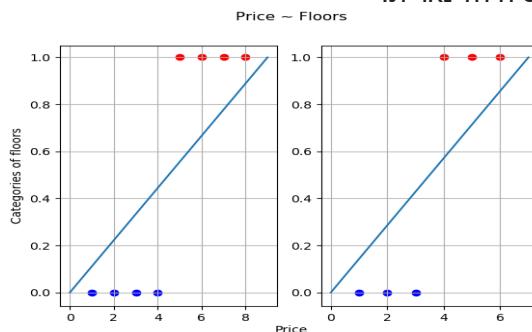
משימת סיווג מוגדרת באופן הבא: בהינתן סט פרמטרים-מוסרים $\{x_n, x_1, \dots, x_m\}$ = אחסין לetzpit מסויימת ישלוועו
אותו לאחת מトוו-מוקטניות אפשריות $\{1, \dots, m\} \in \text{ע-לדוגמה}:$ נתונה תמונה בעלי-הופיקסל-פה מייצגת חיה,
יש לקבועהיזו חיה היא, כאשר הבחירה נעשית מטופחה ווקטור ע-לה-הווקטור ע-מורכב ממשפרים שלמים, לכל

אחד מהם מייצג בחרה מסוימת. בדוגמה של החיות, ניתן לחת $\{dog, cat, chicken\} = \{1, 2, 3\} \in \mathcal{Y}$, כאשר המספרים מייצגים את סט החיות.

ניתן להשתמש במודל של רגסיה לינארית למשימות של סיוג. עבור המקרה ש $\mathcal{Y} = \{m, n, o\}$ יש שתי קטגוריות אפשרויות, ולמקרה יש צורך כל נקודה לאחת משתי הקטגוריות. בעזרת רגסיה לינארית ניתן לבצע מיפוי מ \mathbb{R} ל \mathcal{Y} , כלומר כל נקודה במרחב ממוקה לאחד משני ערכי אפסריים: קבועים ערך $T = 0.5$, ועבור נקודה חדשה x_n אובדוק אם היחס בין הביטוי $w^T x_n + b < 0.5$ שלביקור הסוף אבאה נקודה החדשה מקיימת $w^T x_n + b > 0.5$ או נקודה חדשה תהייה בקטgorיה 1. אחרת, הנקודה החדשה תהייה בקטgorיה 0. באופן פורמלי:

$$y = sign(w^T x_{new} + b - 0.5) = \begin{cases} 1 & w^T x_{new} + b > 0.5 \\ 0 & w^T x_{new} + b < 0.5 \end{cases}$$

בחירה בערך הסוף $T = 0.5$ נובעת מכך שיש שתי קטגוריות $\{0, 1\}$, וערך הסוף נקבע להיות נקודת המיצע ביניהן. דוגמא לנתוניuchi-objektivo-כל אחד מהפיזיומטרים והאחסונים יש בו קומה אחת או שתיים=Ccut רצים לבחון את היחס ביחס למספר הקומות ולקבוע עבור מחיר בית נתון מה מספר הקומות שלוש=Bמקרה זה יש=Ccut שתי קטגוריות=Afloor, $y \in \{0, 1\} = \{1 floor, 2 floors\}$, ויש להיעזר בידיעות על היחסים בצד לבנות מודל מסווג=Cdistr עלשות זאת היא לבצע רגסיה לינארית, ואבצבוחנים מחיר של בית, $w^T price + b \cdot 3.5 + b = 4.5$. או קטן ממנו, כאשר (w, b) הם הפרמטרים של הרגסיה הלינארית.



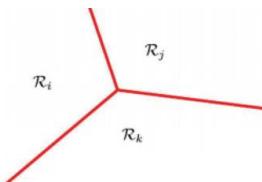
איו-4. רגסיה לינארית כמסוג בינה: מיפוי הנקודות בתאפלט למוקם ביחס לנקודה הדרישה של הרגסיה הלינארית=Bדוגמא הימנית ערך הסוף=Cמתקבל עבור $w^T x + b = 0.5$, $w^T x + b = 3.5 + b = 4.5$. עבור כל בית חדש, בהינתן מחיר ניתן לסוג אותו לאחת משתי הקטגוריות, בהתאם ליחסו לערך הסוף 0.5.

עבור מנקודות ידועות= $(x_1, y_1), \dots, (x_n, y_n)$, $y_i \in \{0, 1\}$:

$$L(\theta) = \sum_{i=1}^n 1_{\{y_i \neq sign(w^T x_i + b + 0.5)\}}$$

הפונקציה(θ)=Fמכילה סט של פרמטרים=(w, b) = θ =Ccיוון שהנגזרת של הפונקציה לפפ' כל אחד מהפרמטרים של θ לא תלוי רק באותו פרמטר, קשה למצאו את θ המבאים למינימום $L(\theta)$.

ניתן להרחב אבאה מסוג גם עבור מקרים בהם יש יותר משתי קטגוריות=multi-class, האימופתראה כמו במרקחה הבינה, ואילך=Cמכל כערומתקטגוריות=m, $\dots, 1, 0, y_i$ =במרקחים אלו יש ליצור מספר קאנטליינאריים, המפרדים בין אזורים שונים. כדי לחשב את הקווים מבצעים התהילה שנקרא all versus one, בו בכל פעם לוקח יחיקתgorיה אחת ובזקקים מהו קי' ההפרדה בין לבין שאר הקטגוריות=Bפרמטרים הנלמדים של קי' ההפרדה ייחס הסט המורכב מכל הפרמטרים של הרגסיה= $w_1, b_1, \dots, w_m, b_m$



איו-5 רגסיה לינארית מרובה – הפרדה בין מספר אזורים שונים על ידי מספר קוים לינאריים=Cבמקרה זהה, נקודה חדשה תסוג לקטgorיה לפי הביטוי הבא:

$$y(x) = \arg \max_i (w_1^T x + b_1, \dots, w_m^T x + b_m)$$

וכל אזכור יוגדר לפיה:

$$R_i = \{x | y(x) = i\}$$

בדומה ל McKee ה bivari, פונקציית המחיר תהיה

$$L(\theta) = \sum_{i=1}^n 1_{\{y_i \neq \hat{y}_i\}} \text{ s.t. } \hat{y}_i = \arg \max_i (w_i^T x + b_i)$$

המסוג האופטימלי יהיה וקטור הפרמטרים המביא את פונקציית המחיר למינימום

$$\hat{\theta} = \arg \min_{\theta} L(\theta)$$

גם ב McKee זה יכין שהגזרת של פונקציית המחיר של כל פרמטר אינו-תלויה רק באותו פרמטר, בפועל קשה למצאו אף θ האופטימלי המביא את $L(\theta)$ למינימום

3.2 Softmax Regression

3.2.1 Logistic Regression

המסוג הנווצר מהרגסיה הלינארית הינה "מסוג קשה"—כל דוגמא חדשה שמתבלט מסוגות לקטgorיה מסוימת, ואין שום מידיעד כמה הדוגמאות זו דומה לקטגוריות האחרות—מסוג זה אינו מספיק טוב עבור מגוון בעיות, שכן מעוניינים לדעת לא רק את הקטgorיה, אלא גם מידיע נספ' על היחס בין כל הקטgorיות. לדוגמה: בהינתן מידע של גידול מסויפורצים לדעת אם הוא ממאייר או שפיר. ב McKee זה ההכרעה היא לא תמיד חישומית, ויש עניין לדעת מה הסיכוי של הגידול להיות ממאייר או שפיר, שהרי יתקשתה טיפול אליה שונה בין McKee בו יש ~40% שהגידול הזה הוא מסווג לביקורתו בו ~40% שהגידול הוא מסווג זהה כדי להימנע מהסתורתיות הקטgorוי, יש ליצור מודל הסתברותי, בו כל קטgorיה מקבלת הסתברות מסוימת. אחד המודלים הבסיסיים הנקרא רגסיה לוגיסטיבית (Logistic regression). עבור המסוג זהה ראשית יש להגדיר את פונקציית הסיגמוואיד:

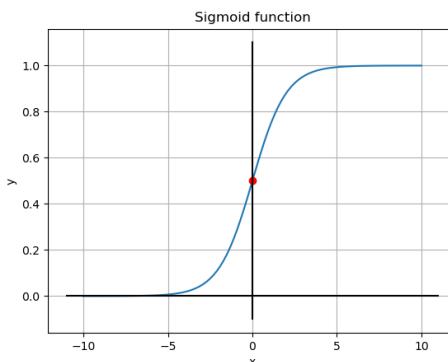
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

פונקציה זו רציפה על כל הישר, ובעצמותה ניתנת להגדיר מסווג עבור McKee ה bivari

$$p(y = 1|x; \theta) = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}}$$

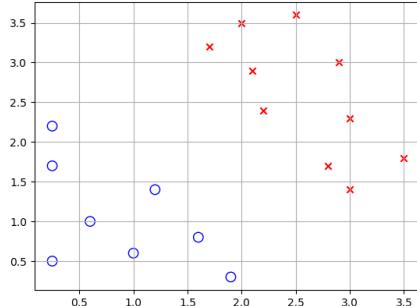
$$p(y = 0|x; \theta) = 1 - \sigma(w^T x + b) = 1 - \frac{1}{1 + e^{-(w^T x + b)}} = \frac{e^{-(w^T x + b)}}{1 + e^{-(w^T x + b)}}$$

המסוג לוקח את קבוחה של ה bivari ומעביר אותו לפונקציה המחזירה ערך בטווח [0, 1], כאשר הערך המוחזק הוא הסתברות להיות בקטgorיה מסוימת. בצדיה להבין יותר טוב את משמעות המסוג, יש להסתכל על גרף הסיגמוואיד:



איור 3.6 גרף הפונקציה: $y = \frac{1}{1+e^{-x}}$. הנקודה (0,0.5) מודגשת באדום

כאשר הפונקציה $b = w^T x + b = 0.5\sigma$. המשמעות של התוצאה זו היא שאם עבור סט פרמטרים w מתקיים $w^T x_n + b > 0$, אז ההסתברות של הנקודה הזו להיות משוכנת לקטגוריה 1 אדולקט מחצי, בעוד ש- $w^T x_n + b < 0$, אז ההסתברות של הנקודה x_n להיות משוכנת לקטגוריה 1 אדולקט מ-0.5. בואו סימטריך= $w^T x_n + b = 0$, אז השאלה היא שזאת תלו依 בקבוק הפרדה משוכנת לקטגוריה 1. קטענו מחלוקת מושג'ת על שני פרמטרים= w_1, w_2 , ועבור כל נקודה $(x_1, x_2) \in \{blue, red\}^2$ גם מה הקטgorיה שלה (זאת $y \in \{0,1\}$):



איור 3.7 דוגמא למספר מדידות התלויות בשני פרמטרים= w_1, w_2 , ומושג'ת לאחת משתי קטgorיות= $\{blue, red\}$

כיוון שננותן הנקודות, ניתן לייצר בעזרת \hat{x} הדוגמאנני-השיש-שלושה-פרמטרים= $w^T = [1, 1, -3]$. הפרמטרים האלויים מרכיבים את הקו הלינארי $3x_2 - 3x_1 + b = 0$, כלומר, עבור כל נקודה אפ' מתקיים $3x_2 - 3x_1 + b > 0$ אם והיא מושוגת כ-'blue', אחרת היא תהיה מושוגת כ-'red'. ק' זה הוא למעשה הפרדה שניית בעזרתו לוסף נקודות חדשות-נקודות זו מתקיים את המושוגה- $w^T x + b = 0$, אולם אם תהיה נקודה חדשה שגם מתקיים $w^T x_n + b = 0$, המשמעות היא שנקודה זו נמצאת בבדיקה על קו ההפרדה-נקודה צו תקבל-ההסתברות ש-50% להיות משוכנת לכל אחת מהקטgorיות. ככל שהנקודה החדשה תתרחק מקו ההפרדה, כך הביטוי= $w^T x + b$ יתרחק מה-0, ולכן גוף= $w^T x + b$ ס' יתרחק חצי ויתקרב לאחד מערבי הקצה $= 1$, והמשמעות היא כמובן שיש יותר סיכוי שנקודה זו שייכת לקטgorיה אחת ולא אחרת.

כמובן שניית לחתת גם את המושוג ההסתברותי הזרול-הסתברותי כמסוג קשה: עבור דוגמא חדש-הלא-локחים את ההסתברויות=שללה \hat{y} לכל=Aחת-המקטgorיות, ומוסוגים את הדוגמא-הקטgorיה בעלת ההסתברות הגבוהה ביותר ביטוי= $\hat{y} = \arg \max_i p(y=1|x, \theta)$ במקורה הבינה-פוקטור ההסתברותי-והינט[$p(y=1|x, \theta)$] ש- \hat{y} זהה הגדולה ביותר

3.2.2 Cross Entropy and Gradient descent

בכדי למצוא את הפרמטרים= $(w, b) = \theta$ האופטימליים בהינתן-הדוגמאות= $\{(x_i, y_i)\}_{i=1}^n$ להחליף את קרייטריון-השגיאה הריבועית הממוצעת בקריטריון אחר למצוור פונקציית המיחס= $=$ Cross entropy קרייטריון זה אומר שיש ישלה-הביבטיא לMINIMUM את מינום הלוג של סך הדוגמאות (הביבטיא נובע משערוך הנראות המרבית – Maximum likelihood)

$$-\log P(Y|X; \theta) = -\frac{1}{n} \sum_{i=1}^n \log p(y_i|x_i; \theta) = L(\theta)$$

למעשה, יש למצוא את סט הפרמטרים $\hat{\theta}$ המביא את הביטוי לMINIMUM: $\hat{\theta} = \arg \min_{\theta} L(\theta)$.

בכדי לחשב את הביטוי יש לפתח קודם את הביטוי עבור נגזרת הסיגמודואיד:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \rightarrow \frac{\partial \sigma(z)}{\partial z} = \frac{-1}{(1 + e^{-z})^2} \cdot e^{-z} \cdot (-1) = \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}} = \sigma(z)(1 - \sigma(z))$$

כך, $\sigma'(z) = \sigma(z)(1 - \sigma(z))$, כלומר, $\sigma'(z) = 1 - \sigma^2(z)$, ולכן $\sigma'(z) = 1 - \sigma(z)$.

$$\frac{\partial L(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \left[-\frac{1}{n} \sum_{i=1}^n \log p(y_i|x_i; \theta) \right] = -\frac{1}{n} \sum_{i=1}^n \frac{\partial \log p(y_i|x_i; \theta)}{\partial \theta_j} = -\frac{1}{n} \sum_{i=1}^n \frac{\partial \log \sigma(z_i)}{\partial \theta_j} = -\frac{1}{n} \sum_{i=1}^n \sigma(z_i)(1 - \sigma(z_i)) \frac{\partial z_i}{\partial \theta_j} = -\frac{1}{n} \sum_{i=1}^n \sigma(z_i)(1 - \sigma(z_i)) x_{ij}$$

בהתאם, הנגזרת של לוג סיגמודואיד ה- σ :

$$\frac{\partial \log \sigma(z)}{\partial z} = \frac{1}{\sigma(z)} \cdot \frac{\partial \sigma(z)}{\partial z} = (1 - \sigma(z))$$

$$\frac{\partial \log(1 - \sigma(z))}{\partial z} = \frac{1}{1 - \sigma(z)} \cdot \frac{\partial(1 - \sigma(z))}{\partial z} = -\sigma(z)$$

כעת יש לשים לב שהנגזרת של $\log p(y=1|z) = 1 - \sigma(z)$, והנגזרת של $\log p(y=0|z) = \sigma(z) - y_i$. לכן ניתן לרשום בקיצור $\frac{\partial}{\partial \theta} \log p(y_i|z) = y_i - \sigma(z)$. במקרה שרגression לוגיסטי, מוחפשים את הנגזרת של $\frac{\partial}{\partial \theta} \log p(y_i|x_i; \theta)$, ולפי היפition המקדים ניתן לרשום את זה כך:

$$\frac{\partial}{\partial \theta} \log p(y_i|x_i; \theta) = (y_i - \sigma(w^T x + b)) \cdot \frac{\partial}{\partial w} (w^T x + b) = (y_i - p(y_i = 1|x_i; \theta)) \cdot x_i$$

כעת לאחר היפition ניתן לחזק חזרה לביטוב $L(\theta)$ ולהציג:

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \theta} \log p(y_i|x_i; \theta) = -\frac{1}{n} \sum_{i=1}^n (y_i - \sigma(w^T x + b)) x_i = -\frac{1}{n} \sum_{i=1}^n (y_i - p(y_i = 1|\theta; x)) x_i$$

3.2.3 Optimization

בדומה לרוגression לינארית, גם כאן חישוב הערך האופטימלי של $\hat{\theta}$ יהיה איטרטיבי בשיטה gradient descent:

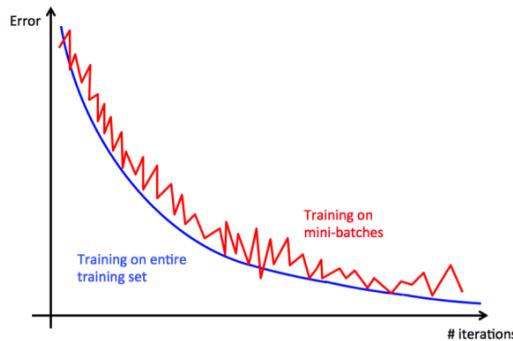
$$\hat{\theta}_{j+1} = \hat{\theta}_j - \epsilon \cdot \frac{\partial}{\partial \theta_j} L(\theta)$$

כאשף הוא הפרמטר של ה-learning rate. כיוון שפונקציית המבחן $L(\theta)$ קעורה, מובהק שתיהה התוצאות נסובך.

במקרים רבים הדאטה טווט הוא גדול, ולחשב את הגראדיאנט עבור כל הדאטה נדרש הרבה חישוב-בכל צעד של קידום. ניתן לחשב את הגראדיאנט עבור חלק מהדadata, וביצוע את הקידום לפה-כיוון של הגראדיאנט המתפרק למשתנים נקדים (SGD). נקודה אחת ולחשב עליה את הגראדיאנט ביחס לה-חישוב בושיטה GD יכול לגרום לשונות גודלה-ככל שהחישוב מתתקדם, ולכך עדיף לחתוף מספר נקודות. חישוב הגראדיאנט בשיטה זו נקרא mini-batch learning (לעומת batch learning) המבוצע על כל הדאטה הנקרא (batch learning). באופן פורמלי, הגראדיאנט בשיטת mini-batch הינו

$$\frac{\partial L}{\partial \theta} = \frac{\partial}{\partial \theta} \left[-\frac{1}{|V|} \sum_{i \in v} \log p(y_i|x_i; \theta) \right] \approx \frac{\partial}{\partial \theta} \left[-\frac{1}{n} \sum_{i=1}^n \log p(y_i|x_i; \theta) \right]$$

אמנם ככל-צעד הוא קירוב לגראדיאנט, אך החישוב מאד מהיר ביחס לגראדיאנט המדויק=וזה יתרון ממשמעותי שיש לשיטה זו על פני batch learning.



איוף.8. השגיאה של gradient descent על כל הדאטה הנקרא (batch learning) מיצגת את השגיאה בשיטה gradient descent בה הגראדיאנט-בכל צעד מחושב על כל הדאטה-והגרף האדום מיציג את השגיאה בשיטה mini-batch learning, בה כל צעד הגראדיאנט-מחושב רק על חלק מהדadata הנבחר באופן אקראי.

בԴօմակ-լինէրը, գֆբ-հօգիստիկ սունին հրցոլրիցի, շնուր լեմուր մամուլլատ մշկէ յիշ-լիլ նկուհ (Overfitting) պֆլա լիչքատ հճատիկ բշուրա մսպիկ տուի (Underfitting)=նիտն լիհօվի լեմշէ այլօչ բիչս լրիբու փրմտէ:

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i; \theta) + \lambda \|\theta\|^2$$

ուզ հնգարտ հինա:

$$\frac{\partial L}{\partial \theta} = -\frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \theta} \log p(y_i | x_i; \theta) + 2\lambda \|\theta\|$$

փրմտէ լու աօպտիմլ մխօշւ ուլ յդի բիչուհու համար:

3.2.4 SoftMax Regression – Multi Class Logistic Regression

բԴօմակ-լինէրը, գմ բ-հօգիստիկ սունին լորի ատ մասուցամ սուուշուս (մկրա բ-ի յուր մշտի կտցորութ) պմ իհալլա լմկրա մորիա կտցորութ միփո շլ լլ կտցորութ լոստիրութ բահօվ [1, 0, 1] ռկ ւու փոնկչի իհա մշտմիմ իհա քոմ սամօակ-SoftMax փոնկչի մուպուլտ ուլ սծրա, ուիչ մօցդրտ կի:

$$\text{SoftMax}(z_1, \dots, z_n) = \left(\frac{e^{z_1}}{\sum_{j=1}^n e^{z_j}}, \dots, \frac{e^{z_n}}{\sum_{j=1}^n e^{z_j}} \right)$$

մոնոկմաշբ-ակսոնունց-բչզկ է, ումկնա մորմլ օրհատօչա, էր շսր լլ աիբրմ լահր փոնկչի հօւ-1. բմկրա բո յի մուպու կտցորութ միշ մուպու կու իի փրժա, ուլլ ած մամ յի ստ փրմտրմ թ=բինտն նկուհ չդժրակ նիտն բցրա-SoftMax լոտ ստիրութ լլ կտցորութ:

$$p(y = i | x; \theta) = \text{SoftMax}(w_1^T x + b, \dots, w_n^T x + b_n)$$

ում մասունին լոկ սուու կշհ-լոկ չիմ ատ աիբր բուլ ստիրութ գցուա բիուրց բմկրա զա փոնկչի մխի-բ քross entropy լոտ իի հ-ը:

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i; \theta)$$

նշան ատ հնգարտ շլ իի բիտօ բուր սուու լութ:

$$\begin{aligned} \frac{\partial}{\partial \theta_i} \log p(y_i = s | x_i; \theta) &= \frac{\partial}{\partial \theta_i} \log \frac{\exp(w_s^T x + b)}{\sum_{j=1}^n \exp(w_j^T x + b)} = \frac{\partial}{\partial \theta_i} \left(w_s^T x + b - \log \sum_{j=1}^n \exp(w_j^T x + b) \right) \\ &= 1_{\{i=s\}} x - \frac{\exp(w_i^T x + b) x}{\sum_{j=1}^n \exp(w_j^T x + b)} = \left(1_{\{i=s\}} - p(y = i | x) \right) x \end{aligned}$$

կաշր սումու 1_{\{i=s\}} իի է պ-է ս = i ու 0 ածրա. ւու նիտն լոհի ատ իի բիտօ ածրան ինգարտ շլ (θ):

$$\frac{\partial L}{\partial \theta_i} = -\frac{1}{n} \sum_{t=1}^n (1_{\{y_t=k\}} - p(y_t = i | x_t; \theta)) x$$

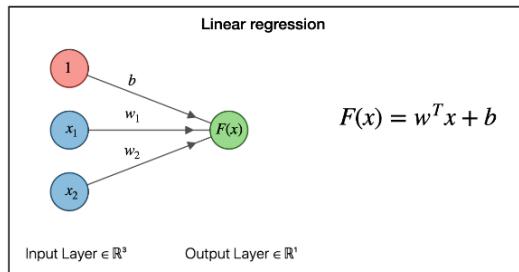
ւու նիտն լոհի արէ թ աօպտիմլ բշտի դստէ:

$$\theta_{i+1} = \theta_i - \epsilon \frac{\partial L}{\partial \theta}$$

3.2.5 SoftMax Regression as Neural Network

לשיטת logistic regression מספר יתרונות=היא יחסית קלה לאימון, מספקת דיוק טוב לדאטסטים פשוטים=יציבה ל-overfitting, מציעה סיווג הסתברות=ומתאימה גם לקרה בו יש יותר משתי קטגוריות. עם זאת, יש חסרון ממשמעותי=קוווי הפרדה של המודל הינם לינאריים, וזו הפרדה שאינה מספקת טובہ עבור בעיות מורכבות=יש מגוון בעיות בהן על מנת לבנות מודל המסוגל להפריד בין קטגוריות שונות, יש צורך במנגנון הפרדה לא לינארי.

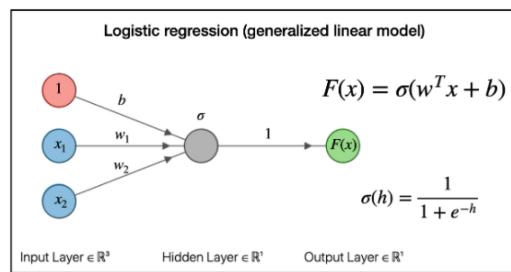
דרך מקובלת=לביצת מודלים לא לינאריים שמשה ברשתות נירונים עמוקות, ובכדי להבין את הקונספט של זה היטב, ראשית יש ל'יצג את המודל לינאריים כשכבה של נירונים, כאשר המודל הזה שקול לחלוטין לכל מה שהוצע עד כה. ביעילות=Linear regression פולקחת סט של מאפיינים ומכלילה כל אחד מהם במשקל=ולאחר מכן סוכמת את כל האלמנטים=bias (בצירוף=bias) בלבד משתנה יחיד הקבוע מה הקטgorיה של ספציה=ניתן לייצג את המודל על ידי התיאור הגרפי הבא:



איור 3.9 יציג רגסיה לינארית כרשת נירונים עם שכבה אחת.

בתיאור זה יש 2 מאפיינים המהווים את ה-\$x\$-תק�ו, וכל אחד מהם מחובר למצאה בתוספת הכפלת המשקל. בנוסף יש bias, ובצירוף המאפייניהם מוכפלים במסקלים וה-\$bias\$-מתקבל המוצא \$= F(x) = w^T x + b = w_1 x_1 + w_2 x_2 + b\$. כל עיגול באIOR נקרא ניירון מלאכותי – אלמנט היכל קבל קלט, לבצע פעולה חישובית ולהוציא קלט.

רגסיה לוגיסטיבית נתנת לתיאור באופן דומה, כאשר הנירונים של סט ה-\$x\$-תקופת=לא מחוברים ישירות למצאה אלאüberים דרך סיגמודoid במקורה הבינארי או דרךSoftMax במקורה בו יש יותר משתי קטגוריות:



איור 3.10 יציג רגסיה לוגיסטיבית כרשת נירונים עם שכבה אחת.

מלבד המעבר בפונקציית הסיגמודoid, יש הבדל נוסף בין הייצוג של הרגסיה הלינארית לייצוג של הרגסיה הלוגיסטיבית: בעוד הרגסיה הלינארית מספקת=במושג=מספר יחיד במצוא (מסווג-קשה), הרגסיה הלוגיסטיבית מספקת=במושג=וקטור באורך של מספר הקטגוריות=באופן צהה-שלכל קטgorיה יש הסתברות מסוימת=שה-\$x\$-תקופת=שייר לאותה קטgorיה.

בפרק הבא=יצצטבנעה-הבעל מספר שכבות של נירונים, כאשר בין שכבה לשכבה יש פונקציה לא לינארית. באופן זה המודל שיתקבל יהיה מיפוי של סט מאפיינים=באופן לא לינארי לוקטור הסתברויות למצוא=הगמישות=של המודל תאפשר להתמודד עם משימות בעלות DATA מושג.

3. References

<https://www.deeplearningbook.org/>

Fitting:

<https://www.calloftechies.com/2019/08/solving-overfitting-underfitting-in-machine-learning.html>

Cross validation:

https://scikit-learn.org/stable/_images/grid_search_cross_validation.png

linear regression:

מצגות מהקורס של פרופ' יעקב גולדברג

<https://joshuagoings.com/2020/05/05/neural-network/>

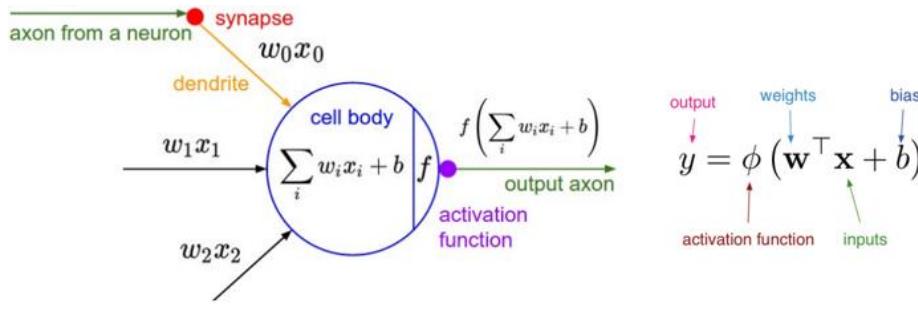
4. Deep Neural Networks

פרק זה עוסק ברשתות נירונים עמוקות=רשת נירונים הינה חיבור של יחידות עיבוד בסיסיות (נירונים מלאכותיים) על ידי משקלים ופונקציות לא לינאריות=רשת נירונית=NETWORK שאמכה היא מכילה יותר שכבה חבוי אחת=אלחיה הצגת הבסיס הרעיוני והפורמלי, יסביר כיצד ניתן לחשב את המשקלים של הרשת בצורה ישרה בעזרת מבנה המוכנה Computational Graph הלמידה ושיטות לבחון עד כמה המודל המתאים אכן מוביל בצורה טובה את הדעתה עליו הוא מאומן.

4.1 Multilayer Perceptron (MLP)

4.1.1 From a Single Neuron to Deep Neural Network

ראשית יש לתאר את המבנה של יחידת העיבוד הבסיסי=нейרון מלאכותי. יחידת עיבוד=אזטנקראט קר עקבה דמוך שלה לנירופיזיולוג=יחידת העיבוד הבסיסית במוח האדם האנושי. הנירוקן יכול לקלט מספר קלטים ולחבר אותם, אז להעביר את התוצאה בפונקציית הפעלה=activation function פשאייה בהכרח לינארית=באופן סכמטי ניתן לתיאור הנוירון הבא כ:

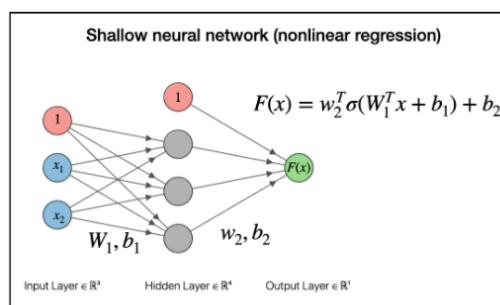


איור 4.1 ייצוג של נירון מלאכותי, המקלט קלט, סוכם אותו ומעביר את התוצאה בפונקציית הפעלה.

הקלט של הנירון הוא סכום משקלים מוכפלים במשקלים $=x^T w$ ועוד= b bias $\in \mathbb{R}^d$. לאחר מכן הסכם עובר דרך פונקציית הפעלה= f והתקבל המוצא $=f(\sum_{i=1}^d w_i x_i + b)$. במקרא הפרטיו בו פונקציית הפעלה היאSoftMax/יגמואיד/ \max והו מוצא לא מחובר לשכבה נוספת, אך למעשה מתקבל את הרגסית הלוגיסטי.

במקרא בדוח נירונים המוחברים ל-שכבת-אינטראקצייתם מוכפלים במשקלים ומתחברים לשכבה נוספת שلنירונים, אצת שכבה המוחברת ל-שכבת-נקראת שכבה חבוי (hidden layer)=שכבת-אינטראקצייתם מוכפלים במשקלים בין השכבה החבוי ושהכבה חבוי בו יש לפחות שכבה חבוי אחת, הקשר בין היכניסה למוצא אין לינארית, וזה היתרון שיש למודל זה. נתבונן במקרא של שכבה חבוי ונחשב את הקשר בין היכניסה למוצא: נסמן את המשקלים בין היכניסה לבין השכבה החבוי כ- b_1, w_1 ואות המשקלים בין השכבה החבוי לבין המוצא כ- b_2, w_2 ונקבל שלאorch השכבה החבוי מתקבל הביטוי= $y = f_2(w_2^T x + b_2) + b_1$. ביטוי זה עובר בפונקציית הפעלה נוספת ומתקבל המוצא

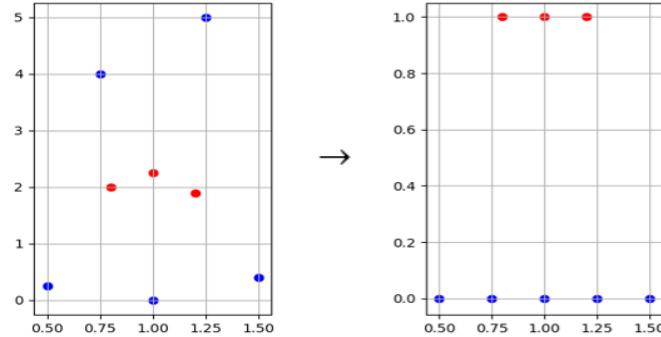
$$\hat{y} = f_2(w_2 \cdot f_1(w_1^T x + b_1) + b_2)$$



איור 4.2 רשת נירונים בעלת שכבה חבוי אחור

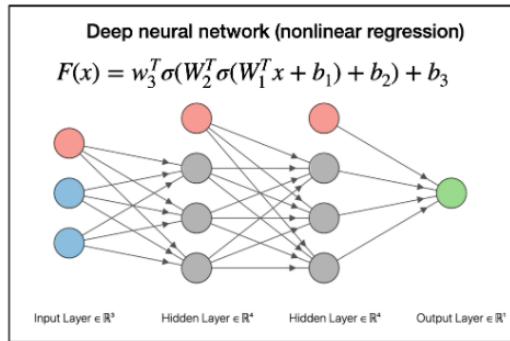
חשיבות להציג שיטרת הרשת היאלביצוע-פעלות לא לינאריות על-השכבות-כך שהוא יסודר באופן חדש הניתך להפרדה לינארית. למעשה לא מבלתי את הפרדה הלינארית=הנעשית בעזרת הרגסית, אלא מבצעים לפניה שלב מקדים של העתקה לא לינארית=תהליך זה נקראלמידה "צוגים" (representation learning); כאשר בכל שכבה

מנס' של למידה יציג פשטוט יותר לדעתך על מנת שהוא יוכל להיות מופרד באופן ליניאר=המייקוד של הרשות הוא אינט' במשימת סיווג אלא במשימת יציג, כך שבסתו של דבר ניתן יהיה לסייע את הדעתה בעזרת סיווג ליניאר=פושע (רגרסיה ליניארית או לוגיסטיבית).



איור 4.3 – העתקה לא ליניארית של דוגמאות על ידי המשוואה $\hat{y} = \begin{cases} 1, & \text{if } 3 \leq (x^2 + y^2) \leq 8 \\ 0, & \text{else} \end{cases}$. העתקה זו מאפשרת להבחין בפער הדוגמאות בעזרת קוו פרדה לינאר.

כאשר מחברים יותר משכבה חבוייה אחת, מקבלים רשת עמודה=ההיבור בין השכבות נעשה באופן זהה=הכפלת של משקלים, סכימה זהה עבורה בפונקציית הפעלה.



איור 4.4 רשת נירונים בעלת שתי שכבות חבויות

רשת נירונים בעלת לפחות שכבה חבוייה אחת הינה=הapproximation Universal, כלומר, ניתן לייצג בקרוב כל התפלגות מותנית בעזרת הארכיטקטורה הזאת. ככל שהרשת יותר עמוקה, כך היכולת שלה להשיג=דיק טוב יותר גדלה

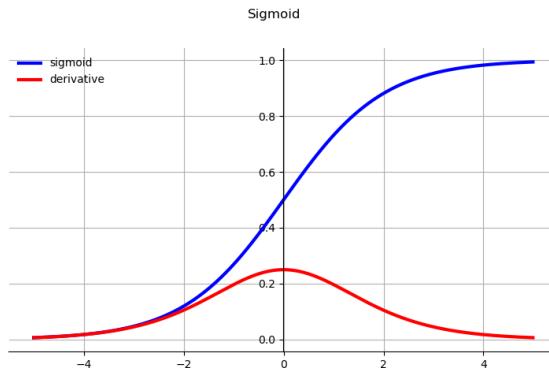
4.1.2 Activation Function

האלמנט המרכזי בכל ניירון הוא פונקציית הפעלה, ההופכת אוטומטית עיבוד לא ליניארי. יש מספר פונקציות הפעלה מקובלות – Sigmoid, tanh, ReLU –

Sigmoid

פונקציית הסיגמאיד הוצגה בפרק של רגרסיה לוגיסטיבית, ועת נרחיב עליה. הפונקציה והנגזרת שלה הן מהצורה:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad \frac{\partial}{\partial z} \sigma(z) = \sigma(z)(1 - \sigma(z))$$



איור 4.5 פונקציית סיגמואיד והנגזרת שלה.

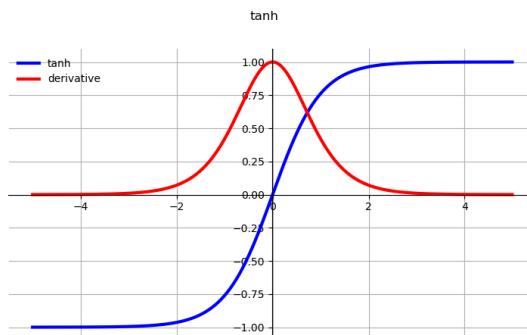
יש לפונקציה זו שלושה חסרונות:

- א. עבור ערכים גדולים, הנגזרת שואפת ל-0. זה כמובן יוצר בעיה בחישוב הפרמטר האופטימלי בשיטת Gradient descent, שהרי בכל צעד=הतווסף תליה בגרדיינט, ואם הוא מתা�פס=לא ניתן לחשב את הפרמטר האופטימלי.
- ב. הסיגמואיד לא ממורכז סביב ה-0, וזה יוצר בעור דאטוטשאים מטורם.
- ג. הן הפונקציה והן הנגזרת דורשות חישוב של אקספוננט, ובאופן יחס' זו פעולה יקרה לחישוב.

tanh

פונקציית טנגןס היפרבולי הינה מהצורה:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \quad \frac{\partial}{\partial z} \tanh(z) = 1 - (\tanh(z))^2$$



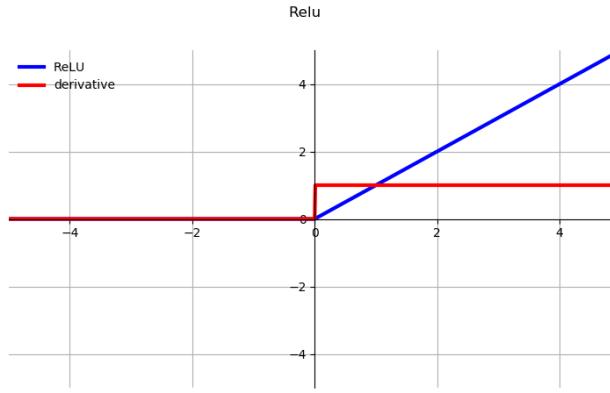
איור 4.6 פונקציית טנגןס היפרבולי והנגזרת שלה.

גם בפונקציה זו יש את הביעות של חישוב אקספוננט והטאפסות הגרדיינט עבור ערכים גדולים, אך היתרון שלו הוא שהוא ממורכז סביבה $\textcircled{2}$.

ReLU (Rectified Linear Unit)

פונקציית ReLU היא פונקציה מטאפסת ערכים שליליים ואידישה כלפי ערכים חיוביים=הfonקציה מחזירה את המקסימום מבין המספרים שהיא מקבלת ובירן 0. באופן פורמלי צורת המשוואה הינה:

$$ReLU(z) = \max(0, z), \quad \frac{\partial}{\partial z} ReLU(z) = 1_{\{z>0\}} = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases}$$



איור 4.7 פונקציית ReLU והנגזרת שלה.

פונקציית ReLU היא פונקציה לא-лиニアרית לחישוב מהפונקציות הקודמות, כיוון שיש בה רק בדיקת של סימן המספר, ואין בה כפל או חילוק. בפונקציה זו הגרדיינט לא מתאפשר בערכים אבודים. יתרון נוסף שיש לפונקציה זו הוא שהיא מוגדרת בכל היבנים. מתכונת יותר מהפונקציות הקודמות $(ax)_+$ (לפונקציה יש שני חסודות עיקריים: היא לא ממורצת סיבובית) ועבורה אתחול משקלים לא טוב מרבית הנוירונים מתאפסים וזה יחסית בזמני-צדדי להתגבר על הבעה האחורנית. להשתמש בורותיות של הפונקציה, כמו למשל PReLU ו-PReLU:

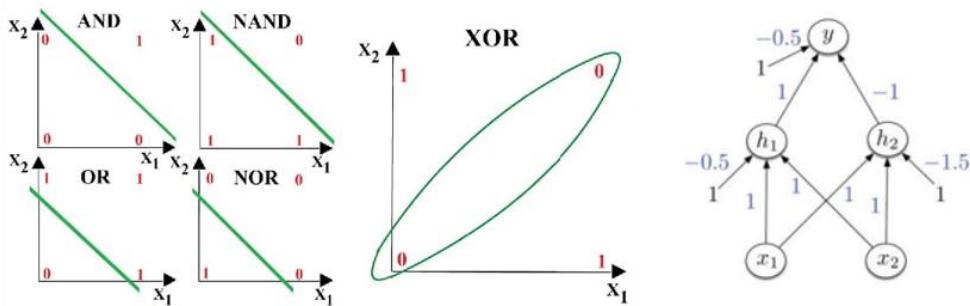
$$PReLU(z) = \max(ax, z), ELU(z) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}$$

בפונקציית PReLU, המקירה הפרטיה $\alpha = 0.1$ נקראת Leaky ReLU. בפונקציית ELU, הפרטיה הוא פרמטר נלמד.

ישנן עוד פונקציות, אך אלה הן העיקריות, כאשר לרוב מקובל להשתמש בReLU ובורותיות של:

4.1.3 Xor

אחד הדוגמאות הידועות ביותר שאין ניתן להפרדה לינארית היא בעיית ה-XOR. שתי כניסה x_1, x_2 ואו המוצאים y אם הכניסות שוות ו-0 אם הן שונות. פונקציה זו מיפה שתי כניסה ליציאה, כאשר יש שתי קטגוריות במוחא, ואין אפשרות להסביר קוו לינאר שיבחין בין הדוגמאות השונות. לעומת זאת, ניתן לבצע שלב מקדים של הפרדה לא-לינארית, ולאחריה ניתן יהיה לבנות מסויים על בסיס קו הפרדה לינארית.



איור 4.8 אופרטור XOR אינו ניתן להפרדה לינארית, בשונה מאשר האופרטורים הלוגיים. באמצעות רשת נוירונים בעלי שכבה חבויה אחת ניתן ליצור מודל פשוט לאופרטור XOR.

בדוגמא המובאת באיזה הרכניות עבורות דרישת שכבה חבויה אחת בעלת שני נוירונים h_1, h_2 , מקבלים בנוסף גם bias. פונקציית הפעלה של נוירונים אלו היא פונקציית הסיגנום, ונitin לכתוב את המוצא של שכבה זו כה

$$h_1 = \text{sign}(x_1 + x_2 - 0.5), h_2 = \text{sign}(x_1 + x_2 - 1.5)$$

לאחר השכבה החבויה הנוירונים מחוברים למוצא, שגם לו יש bias, והסכום של הכניסות והbias עוביים מושגים:

$$y = \text{sign}(h_1 - h_2 - 0.5) = \begin{cases} 1 & \text{if } h_1 - h_2 - 0.5 > 0 \\ 0 & \text{if } h_1 - h_2 - 0.5 < 0 \end{cases}$$

נבחן את המשמעות של הנוירונים: הנירון h_1 יהיה 0 אם שתי הכניסות שוות 0, אחרת הוא יהיה שווה 1. הנירון h_2 יהיה שווה 1 אם שתי הכניסות שוות 1, ובכל מצב אחר הוא יהיה שווה 0. באופן זה לאחר השכבה החבויה הראשונה,

אם גם $\hat{h}_1 \neq \text{שונו-ה-0}$ אז יש לפחות כניסה אחת ששוות-1, וצריך לבדוק בעזרת \hat{h}_2 את המצב של הכניסה השנייה. אם גם הכניסה השנייה שווה-1, אז כניסה שול-עכ (יחד עם ה-bias) יתקבל מספר שלילי, ובמוצא יתקבל-0. אפ-ה הכניסה השנייה היא-0, אז $sign(0.5) = 0$, כלומר ב המצב בו שתי הכניסות ה-0, יתקי-מ-0 $= h_1 = h_2$, ואחר-ה bias ישפי-ע, וכיון שהוא שלילי שוב יתקבל 0 ב מצב-ה.

נרשום בפירוט את הערכים בכל שלב, עברו על הכניסות האפשריות

x_1	x_2	h_1	h_2	$h_1 - h_2 - 0.5$	y
0	0	0	0	-0.5	0
0	1	1	0	0.5	1
1	0	1	0	0.5	1
1	1	1	1	-1.5	0

4.2 Computational Graphs and propagation

4.2.1 Computational Graphs

כפי שהסביר לעיל, רשת נוירונים عمוקה היא רשת בעלת לפחות שכבה עמוקה אחת, והמטרה של כל שכבה היא ללמידה יציג פשטוט יותר של המידע שנכנס אליה, כך שבסופו של דבר ניתן יהיה להבחין בין קטגוריות שונות בעזרת הפרדה לינארית. מה שקובע את השינוי של הדאט-הבעבר שלו בראשת הם המשקלים והנוירונים המבצעים פעולות לא לינאריות. בעוד הפעולות אותן מבצעים הנוירונים קבועות (סכימה ולאחר מכן פונקציית הפעלה), המשקלים קבועים בהתחלה באופן אקראי, ובעדות הדוגמאות הידועות-הניתן לאמן את הרשות ולשנות את המשקלים כך שיביצעו את למידת הייצוג החדש בצורה אופטימלית.

תהליך האימון מתבצע בשני שלבים – ראשית-המכניס-פ-ודגמאות-ידיועה לתחילה הרשות ו"מפעעים" אותה עד למוצא (Forward propagation), כמובן, מחשבים את השינוי שהיא עוברת כאשר היא מוכפלת במסקלים וועורמת בנוירונים החבויים. לאחר שmagיעים למוצא-משווים את מה שהתקבל למה שאמור להיות במצבו לפי מה שידוע על דוגמא-ז, וא-מבצעים פעוף לאחר (Backward propagation)= שטטרתו לתקן את המשקלים בהתאם למזה שהתתקבל במצבו-השלב השני הוא למשה-חישוב עיל-של-פ-על פנ- כל שכבות הרשות=מחשבים את הנגזרת ב- $\frac{\partial L}{\partial w_i}$ המשקל- i -של-בין פונקציית המבחן- (θ) , ואז מבצעים עדכון בשיטה-פ-ודגמאות-ידיועה – $w_i = w_{i+1} - \epsilon \frac{\partial L}{\partial w_i}$.
כיון שהרשות יכולה להכיל מילוני משקלים, יש למצאו דרך עיליה לחישוב הגרדיאנט עברו כל משקל-

נ-ה-עשות את התהליך הדשלבי זהה בעזרת Computational Graphs, שזהו למעשה גרפ' הבנו-מצמת-ה-המייצאים את התהליך-שהדאט-ה-עובר בטור הרשות-ה-גרף יכול ליצג כל רשות, ונitin באמצעות נגזרות מורכבות-ה-באופן פשוט יחסית-ל-ה-שלב הראשון בו מעבירים-פ-ודגמאות-בכל חלק-ה-גרף, ניתן למשל-ל-ח-ש-ב-א-פ-ה השגיאה הריבועית המוצעה- $(y - \hat{y})^2$, להגדיר אותה כפונקציית המבחן, ולמצוא את הנגזרת של כל משקל לפי פונקציה זו – $\frac{\partial L}{\partial w_i}$, כאשר הנגזרות החלקיים מחושבות בעזרת כל השרשת.

4.2.2 Forward and Backward propagation

באופן פורמלי, עבור N משקלים התהליך מנוטה כ-:

Forward pass:

For i in 1 ... N:

Compute w_i as function of $w_0 \dots w_{i-1}$

Backward pass:

$$\overline{w_N} = 1$$

For i in $N - 1 \dots 1$:

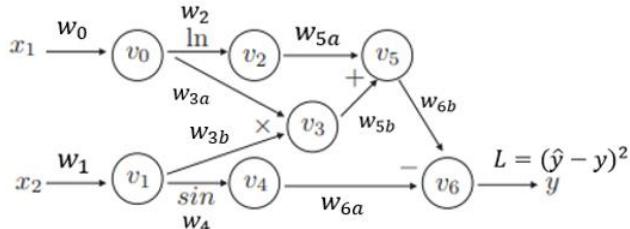
$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial w_N} \cdot \frac{\partial w}{\partial w_{N-1}} \dots \frac{\partial w_{i+1}}{\partial w_i}$$

$$\overline{w_i} = w_i - \epsilon \frac{\partial L}{\partial w_i}$$

בשלב הראשון מחשבים כל צומת על סמך הצמתים הקדמים לו, ובשלב השני בו חוזרים אחורה, מחשבים את הנגזרות של כל משקל בעזרת כל השרשרת החל מהmozo עד לאותו משקל, ומעדכנים את המשקל=Nestcall למשל בדוגמה הבאה

$$y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$$

לפונקציה זו שתי כניסה, העוברות כל אחת בנפרד דרך פונקציה לא לינארית, ובנוסף מוכפלות אחת בשנייה. באופן גרפי ניתן לאייר את הפונקציה כה



איור 4.9 הפונקציה $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ מתוארת באופן גרפי.

בגרף זה יש 7 צמתים:

$$v_0 = x_1, v_1 = x_2$$

$$v_2 = \ln(v_0), v_3 = v_0 \cdot v_1, v_4 = \sin(v_1)$$

$$v_5 = v_2 + v_3$$

$$\hat{y} = v_6 = v_5 - v_4$$

לאחר שbowut החישוב עbowut, ניתן לחשב את הנגזרות החלקיות, בעזרת כל השרשרת

$$\frac{\partial L}{\partial w_{6a}} = -1, \frac{\partial L}{\partial w_{6b}} = -1$$

$$\frac{\partial L}{\partial w_{5a}} = \frac{\partial L}{\partial w_{6b}} \frac{\partial w_{6b}}{\partial w_{5a}} = -1 \cdot 1 = 1, \quad \frac{\partial L}{\partial w_{5b}} = \frac{\partial L}{\partial w_{6b}} \frac{\partial w_{6b}}{\partial w_{5b}} = -1 \cdot 1 = -1$$

$$\frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial w_{6a}} \frac{\partial w_{6a}}{\partial w_4} = -1 \cdot (-\cos w_4) = \cos w_4$$

$$\frac{\partial L}{\partial w_{3a}} = \frac{\partial L}{\partial w_{6b}} \frac{\partial w_{6b}}{\partial w_{5b}} \frac{\partial w_{5b}}{\partial w_{3a}} = -1 \cdot 1 \cdot w_{3b}, \quad \frac{\partial L}{\partial w_{3b}} = \frac{\partial L}{\partial w_{6b}} \frac{\partial w_{6b}}{\partial w_{5b}} \frac{\partial w_{5b}}{\partial w_{3b}} = -1 \cdot 1 \cdot w_{3a}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial w_{6b}} \frac{\partial w_{6b}}{\partial w_{5a}} \frac{\partial w_{5a}}{\partial w_2} = -1 \cdot 1 \cdot \frac{1}{\ln w_2}$$

המשקלים בכניסה, w_0, w_1, w_2 , רק מעבירים ללא שינוי את הכניסות לצמתים v_0, v_1, v_2 , אך הם שוויים

לאחר שכל הנגזרות החלקיות חושבו, ניתן לעדכן את המשקלים לפי העיקרון של GD: $w_{i+1} = w_i + \epsilon \frac{\partial L}{\partial w_i}$

התפקיד הגדול של חילוקת הרשת לגרף עפ-צמתים נובע בכך שכאשר כותבים את הנגזרת של \hat{y} בקשר לכל השרשרת, אז כל איבר בשרשראת בפני עצמו הוא יחסית פשוט לחישוב. למשל – נגזרת של חיבור היא 1, נגזרת של כפל היא המקדם של המשטנה לפיו גוזרים, וכן באותו אופן עבור כל אופרטור שימושיים בזיכרון מסויים=Lשיטה זו קוראים backpropagation ויה מאוד נפוצה ברשותות עמוקות עוקב ייעולותה בחישוב המשקלים=Bבשונה מבועו ורגรสיה, חישוב האופטימום ברשותות עמוקות היא לא בעיה קמורה, ולכן לא תמיד יש לה בהכרח מינימום גלובל

עם זאת, עדכון הממשקלים בשיטת backpropagation הוכיח את עצמו=lמרות שהמשקלים לא בהכרח הגיעו לאופטימום שלמה

4.3 Optimization

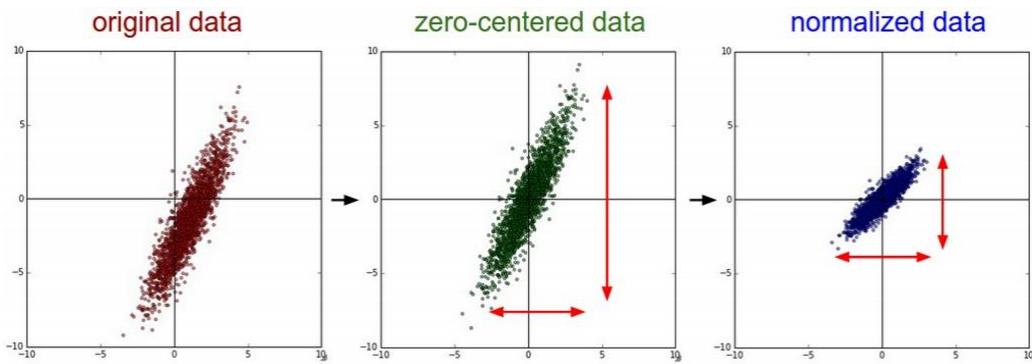
מציאת אופטימום למשקלים על פב-כל העומק של הרשת היא בעיה לא קמורה, ולכן אין לה בהכרח מינימום גלובלי=локלי=עדכון המשקלים בשיטת backpropagation יוביל לביצוע אופטימייזציה מנוספת על הרשת על מנת לשפה-אפקטיביים שלמה

4.3.1 Data Normalization

חלק מפונקציות הפעולה אין ממורכבות סיבוב =0, ועבור ערכים גבוהים הן קבועות בקריבוב ולכן גרדיאנט בערך-CONSTANT, דבר שאינו מאפשר לעדכן את המשקלים בשיטה=GCD כדי להימנע מהגעה לתחום ה"רוויה" בו הגראיננס מתאפס, ניתן לנורמל את הדadata כך שהיא בעל תוחלת 0 ושונות 1, ובכך הוא יהיה ממורכב סיבוב =0:

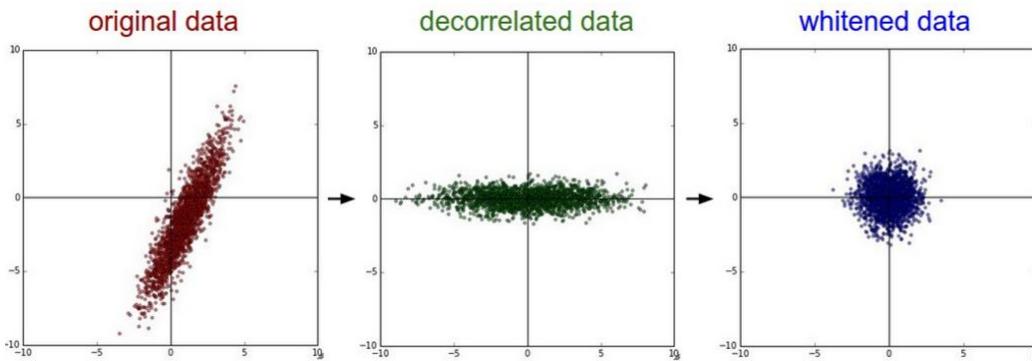
$$X_i = \frac{X_i - \mu_i}{\sigma_i}$$

ובאופן חזוותי



איור 4.10 נרמול נתונים בשני שלבים – איפוס התוחלת (ירוק)=ונרמול השונות -1 (כחול)

שלב זה הוא למעשה שלב pre-processing הנועד להcin את הדatos לפני כניסה לרשת, כדי לשפר את אימוץ הרשות=ישנים נוספים לנורמל את הדatos=ליכסן אופטימטרית covariance של הדatos=אל הפה-אורותה למטרית היחידה



איור 4.11 דרכים נוספים לנורמל את הדatos – ליכסן את מטרית covariance (ירוק) או להפוך אותה למטרית היחידה (כחול)

4.3.2 Weight Initialization

ענין נוספת לשיכול להשפייע על האימון וניתן להתייחס אליו עוד בשלב ה-pre-processing והוא אתחול המשקלים. אף כל המשקלים מאותחלים ב-0=איך הם מוצאים כל הגראיננס יחסית 0, ולא יבצע עדכון למשקלים=לכך שבלוחות את המשקלים ההתחלתיים בצורה מושכלת, כלומר, להציג אותם מהתפלגות מסוימת שתאפשר אימון טוב של הרשת.

אפשרות אחת לאותחול היא להציג עבור כל משקל ערך קט-טהתפלגות נורמלית עם שונות קטנה= $(\alpha, 0, N)$, כאשר $\alpha = 0.01$ or 0.1

בערכיים קטנים גורם לאיפוס הגרדיינט מהר מדי'. כדי להתמודד עם בעיה זו, ניתן לבחר $1 = \alpha$, אך זה יכול לגורם להתקדרות הגרדיינט=שייטה עיליה יותר נקרא Initialization Xavier, הנקראת בחשבון אופרגודל של השכבות – האתחול יבוצע באמצעות נורמלית, אך השונות לא תהיה מספקת כמעט, אלא תהיה תליה במספר השכבות – $\frac{1}{\sqrt{n}} = \alpha$ =שייטה זו טובה גם לרשות עם הרבה שכבות, אך היא בעייתית במקורה בו פונקציית הפעלה הינהReLU, כיוון שהאתחול מניח שפונקציית הפעלה ממורכצת סיבי=0 (כמו למשקה). כדי לאפשר גמישות גוף מבחן פונקציית הפעלה, ניתן לבחר $\frac{2}{\sqrt{n}} = \alpha$, ואז האתחול יתאים גם ל-ReLU.

Xavier-Initialization הפרמטרים היא להגביר מהתקדרות אחת, כאשר באופן דומה ל-
גם כאן הגבולות הייחודיים בגודל השכבות $= \left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right] U$.

4.3.3 Batch Normalization

כאשר מבצעים normalization, למשה דוגמים לכך שבכניסה לרשת הדאטץיה מונומל-סיב-0=באופן זהה נמנע מהגעה למשב בו יש ערכים גבוהים בעומק הרשת, הגרומים להתקדרות או להתקדרות של הגרדיינט בפועל, הנרמול הזה לא תמיד מספיק טוב עבור כל השכבות, ואחרי כמה שכבות של הכפלה במשקלים ומעבר בפונקציות הפעלה הרבה פעמים מתקיים ערכיך=גבויים=באופן Data normalization ל-פונקציות המבצעו לאימון, ניתן תוך כדי האימון לבצע normalization שודואג לנרמול הערכים שנכנסים לנירוניים בשכבות החבויות. התהילך נעשה בשלושה שלבים:

- א. עבר כל ניירון בעל פונקציית הפעלה לא לינארית=מחשבים אופחתות והשונות של כל העריכיך=הויצאים ממנה
 - ב. מנרמלים את כל היציאות=מחסרים מכל=יציאת התוחלו=ומחלקים את התוצאה בשנות(בתוספה אפסיון, כדי להימנע מחילוקה ב-0).
 - ג. הנרמול יכול לגרום לאיובן מידע, לכן מבצעים לתוכה המנורמלת=scale and shift=הזהה ושינוי קנה המידה. התיקון מתבצע בעזרת פרמטרים נלמדים
- עבור שכבות גדולות חישוב התוחלת והשונות י��יכוון של ניירון יש הרבה יציאות, לכן לוקחיף-ריך חלק מהיציאות=עבור שכבת מינית $\mathcal{B} = \{x_1 \dots m\}$. Mini Batch: בואופן פורמלי ניתן לנסח את ה- $\text{BN}_{\gamma, \beta}(x_i)$

$$\begin{aligned}\mu_{\mathcal{B}} &= \frac{1}{m} \sum_{i=1}^m x_i, \quad \sigma_{\mathcal{B}}^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \\ \hat{x}_i &= \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \\ y_i &= \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i)\end{aligned}$$

כאשר γ, β הם פרמטרים נלמדים (עבור כל ניירון יש פרמטרים שונים) בשלהמבחן=השונות והתוחלת שבעזרת ממצאים את הנרמול אין נלקחיף-מהיציאות של הנירונים, אלא לוקחיף ממוצע של כמה מה- Mini Batch האחרון.

יש כמה יתרונות לשימוש $\text{Batch Normalization}$: האימון נעשה מהר יותר, יש פחות ריגשות לאתחול של המשקלים, אפשר שימוש ב-rate learning גודל יותר (מוני מהגרדיינט להתקדר או להתקדר), אפשר שימוש במוגן פונקציות הפעלה (אם אלה שאין ממורכצות סיבי=0)=ומוסף באופן חילקי גם רגולרייזציה=שונות נמוכה במאזן).

4.3.4 Mini Batch

במקרים רבים הדטה-ט גדול, ולחשב את הגרדיינט עבור כל הדאטץצורך הרבה חישוב. בכל צעד של קידום ניתן לחשב את הגרדיינט עבור חלק מהדטה, ובוצע את הקידום לפי הכוון של הגרדיינט המתקביל למשקל=ניתן לבצע באופן אקראי נקודה אחת ולחשב עליה את הגרדיינט=בוחירה-ה-נקראות SGD (Stochastic Gradient Descent).

כיוון שבכל צעד יש בחירה אקראיית של נקודה. בחירה אקראיית של נקודה בודדת יכולה לגרום לשונות גדולות ככל שהחישוב מתќדם, ולכן בדרך כלל מבצעים learning mini-batch – חישוב הגרדיאנט על חלק מהדатаה. באופן זה גם יש הפחטה של כמות היחסותיים, וגם איפשרו גבואה. אם מבצעים את החישוב בשיטה זו יש לדאוג שהדאטא מעורבב כדי שהמשקלים אכן יתעדכו בצורה נכונה, ובנוסף שה-mini-batch הוא מוגבל כך שהוא רק שפהmini-batch (אם הדאטא נספְךEpoch) והגודל שלו קטן מ- S , אז כל Epoch הוא N/S איטרציות.

אמנם כל צעד הוא קירוב לגרדיאנט, אך החישוב מאד מהיר ביחס לגרדיאנט המדויק, וזה יתרון ממשמעותי שיש לשיטה זו על פניה learning batch שמתќדים המשקלים קרובים מאד לאלו שהיו מתќבים באמצעות batch learning.

4.3.5 Gradient Descent Optimization Algorithms

בשיטת GD, עדכון המשקלים בכל צעד הוא $w_{i+1} = w_i - \epsilon \frac{\partial L}{\partial w}$, כאשר ϵ הוא פרמטר שנקרא Learning Rate (lr). והוא קבוע עד כמה יש לשנות המשקל בכוון הגרדיאנט – בניית בעיות רגסיטר – אופטימיזציה – רשות נוירונים ϵ – לרוב בעיה שאינה קמורה, שכן לא מובטחת התכנסות למינימום הכלובאל. משום כך, אם בכל צעד הולכים יותר מדף לכיוון הגרדיאנט – השילוי, ניתן להתכנס לנקודות אוכף או למינימום – לא קלי שוואו איטם בהכרח המינימום הכלובאל. מצד שני אם מתקדים מעט מדי לכיוון הגרדיאנט, המשקל בקצבן מתעדכן. פרמטר ϵ – גזענו – להתגבר על בעיות אלו, לפחות שוואו לא יהיה גדול מדי (אחרת תהיה התבדרות של המשקלים או התכנסות למינימום – לא קלי) ולאחר מכן קטפדי (אחרת לא תהיה התקדמות או שהיא תהיה מאד איטית). כיוון שאין ערך אבסולוטי שמתאים לכל בעיות – יש מגוון שיטות המנסות למצוא את העדכון האופטימלי בכל צעד – יש שיטות שימושísticas בפרמטר משתנה – lr – ויש שיטות שימושיפות פרטיצ'יות אחרות לביטוי של העדכון – adaptive lr.

Momentum

ישנם מצבים בהם יש כל מני פיתולים בדרך לנקודות מינימום. במקרה זה, בכל צעד הגרדיאנט יפנה לכיוון אחר, וההתכנסות לנקודות מינימום תהיה איטית – הדבר דומה לנחל שזרום לים, אך הוא לא זורם ישר אלא יש לו הרבעה פיתולים. כדי להאיץ את ההתכנסות במקורה זה – ניתן לנסות לבחון את הכוון הכללי של הגרדיאנט על סמך כמה צעדים, ולהוסיף התקדמות גם לכיוון זהה. שיטה זו נקראת מומנטום, כיון שהיא מחפש את המומנטום הכללי שפיה הגרדיאנט. החישוב של המומנטום מתרחש בסיסמה בנוסחה רקורסיבית:

$$m_{i+1} = \mu m_i - \epsilon \frac{\partial L}{\partial w}$$

ואז העדכון הינו:

$$w_{i+1} = w_i + m_{i+1}$$

הפרמטר μ הינו פרמטר דעיכה עם ערך טיפוסי בטווח [0.9, 0.99]. ניתן להבין את משמעותו על ידי פיתוח של ערך אייבר בסיסמת המומנטום:

$$m_{i+1} = \mu m_i - \epsilon \frac{\partial L(w_i)}{\partial w} = \mu^2 m_{i-1} - \mu \epsilon \frac{\partial L(w_{i-1})}{\partial w} - \epsilon \frac{\partial L(w_i)}{\partial w}$$

ניתן לראות שככל שהולכים אחורה בצעדים, כך החזקה של גידלה. אפסי μ , אז עם הזמוקה – נקלר ויקטן, וכן תהיה פחות השפעה לצעדים שכבר היו לפני הרבעה עדכוניים – תחת הנחה שהגרדיאנט – נקי – לא פותח נוסחה סגורה לרקורסיה:

$$m_{i+1} = \mu m_i - \epsilon \frac{\partial L(w)}{\partial w} = \mu^2 m_{i-1} - \mu \epsilon \frac{\partial L(w)}{\partial w} - \epsilon \frac{\partial L(w)}{\partial w} = \dots = -\epsilon \frac{\partial L}{\partial w} (1 + \mu + \mu^2)$$

הביטוי שמתќבל הוא סדרה הנדסית מתכנסת, ובסעך הכל מתќבל הביטוי:

$$w_{i+1} = w_i - \frac{\epsilon \frac{\partial L}{\partial w}}{1 - \mu}$$

היעילות של המומנטום תלויות בבעיה – לפעמים היא מאייצה את ההתכנסות ולפעמים כמעט ואין לה השפעה, אך היא לא יכולה להזין.

וリアציה של שיטת המומנטום נקראת=Nesterov Momentum בשיטה זו לא מחשבים את הגרדיינט על הצעה הקודם, אלא על המומנטום הקודם:

$$m_{i+1} = \mu m_i - \epsilon \frac{\partial L}{\partial w} (w_i + \mu m_i)$$

$$w_{i+1} = w_i + m_{i+1} = (w_i + \mu m_i) - \epsilon \frac{\partial L}{\partial w} (w_i + \mu m_i)$$

שיטה זו בעודדת טוב יותר עבור בעיות קמורות-כלהomer היא מצליחה להתכנס יותר טוב מאשר המומנטום הרגיל, אך היא איטית יותר.

learning decay

באימון רשותות עמוקות בדרך כלל כדי להקטין את ה- $\frac{\partial L}{\partial w}$ הזמן-הסיבה לכך היא שיכל שמתקדמים לכיוון המינימום, יש צורך בצעדים יותר קטנים כדי להצליח להתכנס אליו ולא לחוץ מסביבו מצד לצד. עם זאת, קשה לקבוע כיצד-בדוק להקטין את ה- $\frac{\partial L}{\partial w}$ הינה מהירה שלו תימנע הגעה לאזור של המינימום, והקטנה איטית שלו לא תעוזז להתכנס למינימום כאשר מגעים לאזור שלו. ישנו שלושה סוגים נפוצים של שינוי הפרמטר:

א. שינוי הפרמטר בכל מתקופה Epoch. מספרים טיפוסיים הם הקטנה בחצי כל Epoch אוכלוקה ב-0.4%.

ב. דעיכה אקספוננציאלית של ה- $\frac{\partial L}{\partial w}$, כלומר $\epsilon_0 e^{-kt}$, כאשר k הם היפר-פרמטרים, ו- t יכול להיות צעד או Epoch.

ג. דעיכה לפז' $\frac{1}{1+kt} = \epsilon_0 e^{-kt}$, כאשר k הם היפר-פרמטרים, ו- t הינו צעד של עדכון.

Adagrad and RMSprop

בעוד השיטה הבודהה מעדכנתה-ז'גבצור-הקבועה מראש, ניתן לשנות אותו גם באופן מסתגל לפי ההתקדמות בכיוון הגרדיינט-בכל צעד-ניתך-בלחן עד כמה גדול היה השינוי בצעדים הקודמים, ובהתאם לכך-אפשר ללחוץ-

$$w_{i+1} = w_i - \epsilon_i \frac{\partial L}{\partial w}, \epsilon_i = \frac{\epsilon}{\sqrt{\alpha_i + \epsilon_0}}, \alpha_i = \sum_{j=1}^i \left(\frac{\partial L}{\partial w_j} \right)^2$$

כאשר ϵ_0 הוא מספר קטן הנועד למנוע חלקה ב-0. כדי-

הו-אובייחס ישר ללחוץ ההתקדמות בכיוון הגרדיינט. בכך מרווחים דעיכה של-ה- $\frac{\partial L}{\partial w}$, בקצב המשתנה לפיה ההתקדמות.

באופן ייחודי, הדעיכה של-ה- $\frac{\partial L}{\partial w}$ מחרירה, כיוון שהסתוכסח $\sum_{j=1}^i \left(\frac{\partial L}{\partial w_j} \right)^2$

יש שיטות-ה-קונוטנים יותר משקל לצעדים האחרוניים-ופחות לעמידים שכבר עברו מזמן-השיטה הפופולרית-נקראת RMSprop; ובשיטה זו במקומם לסקום את ריבוע הגרדיינט של כל הצעדים הבודהה-באותו שווה-ממצאים moving average, וככל שעברו יותר צעדים מוצאים עד צעד הנוכחי, כך תהיה לו פחות השפעה על דעיכת ה- $\frac{\partial L}{\partial w}$.

$$w_{i+1} = w_i - \epsilon_i \frac{\partial L}{\partial w}, \epsilon_i = \frac{\epsilon}{\sqrt{\alpha_i + \epsilon_0}}, \alpha_i = \beta \alpha_{i-1} + (1 - \beta) \left(\frac{\partial L}{\partial w} \right)^2$$

Adam

ניתן לשלב בין הרעיון של מומנטום לבין adaptive learning rate

$$\alpha_i = \beta_1 \alpha_{i-1} + (1 - \beta_1) \left(\frac{\partial L}{\partial w} \right)^2, m_i = \beta_2 m_{i-1} + (1 - \beta_2) \frac{\partial L}{\partial w}$$

$$\hat{\alpha}_i = \frac{\alpha_i}{1 - \beta_1^i}, \hat{m}_i = \frac{m_i}{1 - \beta_2^i}$$

$$w_{i+1} = w_i - \frac{\epsilon}{\sqrt{\hat{a}_i + \epsilon_0}} \hat{m}_i$$

מספרים טיפוסיים: $\epsilon = 5^{-4}$ or $\beta_1 = 0.99, \beta_2 = 0.9, \beta_3 = 0.9$. האלגוריתם למשה גם מוסיף התקדמות בכיוון המומנטום (הכיוון הכללי של הגрадיאנט), ו證明באי לעיצה אדפטיבית שלא-זקח האלגוריתם הילדי פופולרי-ברשתות עמוקות, אך הוא לא מושלם ויש לו שתי בעיות עיקריות: האימון הראשוני לא יציב=Cיוק שבתחלת האימון יש מעט נקודות לחישוב הממוצע עבור m . בנוסף, המודל המתתקבל גוטה ל-*overfitting* ביחס ל-SGD עם מומנטום.

יש הרבה וריציות חדשות על בסיס-**Adam**=שנוועדו להתגבר על בעיות אלו:=ניתן למשוך להתחילה לאמן בקצב נמוך, וכאשר המודל מתגבר על בעית ההתייצבות הראשונית, להגבר את הקצב(Learning rate warm-up)=במקביל, מתקלה התחילה עפ**SGD**=אשר קרייטריונים מסוימים מושגים=מתתקי פים.=כך ניתן לנצל את התכונות מהירה של Adam בתחלת האימון, ואת יכולות ההכללה של**GD**

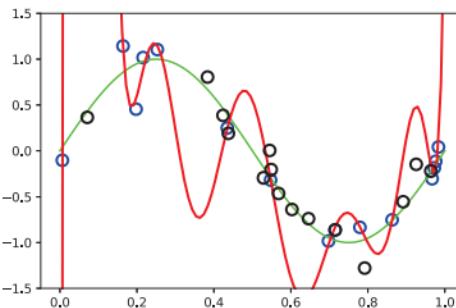
4.4 Generalization

כל מודל שנבנה נסמכ על-דאטהקיים, מתיוך מגמה שהמודל יתאים גם לדאטא-החדש-לכן יש חשיבות גודלה שהמודל ידוע להכליל כמה שיותר טוב, על מנת שההערכות-בצורה טוביה לא רק לדאטא-הקיים אלא גם לדאטא-החדש-במייל-פה אחרות, ש לוודאי שהמודל לא-מתאים את הפרמטרים שלו רק לדוגמאות שהוא רואה, אלא שינסה להבין מトוך הדוגמאות מה החקויות הכלליות, שמתאימה גם לדוגמאות אחרות.

4.4.1 Regularization

כפי שהוסבר בפרק 3.1.3, מודל יכול לסייע מהטיה לשני כיוונים – Overfitting ו– Underfitting. מצב בו ניתנת הערכתית יתר לכל נקודת האימון, מה שגורר מודל מסדר גבוה בעל שונות גדולות. במצב זה המודל מתאים רק לחלק האימון, אך הוא לא מצליח להכיל גם נקודות חדשות – Underfitting. הוא מצב ההופך מודל שלא מצליח למצואו מוגמה המכיל מספיק מידע על הדוגמאות הנתונות, ויש לו רושץ חזק.

ברשת נוירונים, ככל שמספר הפרמטרים גדול, כך השגיאה של ה- test -dataset תהיה הולכת וירדנית עד נקודת מסוימת, ומשם היא גדלה בחזרה. בהתאם השגיאה יורדת כיון שמל Hitchim לבנות מודל יותר מדויק ונמנעים מ-underfitting, אך במקרה מסוימת יש יותר מדי פרמטרים והם נהנים מותאמים יותר מ- overfitted האימוץ ומתקיים overfitting =למעשה צרי למצוא את היחס הנכון בין מספר הפרמטרים (סדר המודל) לבייגוד הדאטה. כיון שאנו לא יכולים Overfitting =בעזרת ה-training בלבד, שהרי ה-Loss Function כולל שיטות יותר פרמטריות- non-technical לחלק ארכידאטורי- shallow חלקיקי- deep . בשלב ראשון בונים מודל בהינתן ה-training set ולآخر מכון בוחנים את המודל על ה-validation set=אם המודל לא מתאים ל-validation set משיש overfitting , כלומר המודל מתאים רק לדוגמאות שהוא הושנתן להם הערצת יתרה= $\text{validation accuracy}$ של ה-validation set.



איור 4.12 בדיקת overfitting בעזרת הנקודות החקולות. הנקודות החקולות שיכtot ל-training והשחורות שיוכtot ל-validation. המודל מתקדם רק לנקודות החקולות, אך לא מצליח שאותו נוטה \rightarrow overfitting. המודל הירק לעומת זאת מתקדם גם לדוגמאות חדשות.

האפשרות הכי פשוטה להימנע מ-overfitting היא פשוט להוציא את גודל הרשות לבניום-Epoch=Early stopping בכל Epoch=Eloss=Fשל ה-validation, וכך הוא מתחילה לעלות בהתאם ל-AUC=אינטגרטָן של שיטות מאוד-ליליאוֹן, אֲבִישְׁקָשִׁיטָן אֶחָת ש-משופקָה ביצועים יותר טובים, ונבחן להפסיק את האימוק-שיטות אל-פְּשָׁטוֹת מואוד-ליליאוֹן, אֲבִישְׁקָשִׁיטָן אֶחָת ש-משופקָה ביצועים יותר טובים, ונבחן

4.4.2 Weight Decay

בדומה לרגולריזציה של linear regression, גם ברשת נוירונים ניתן להוסיף איבר=**יבוע-פונקציית המחר', מה שמכונה regularization^{L2}**

$$Cost(w; x, y) = L(w; x, y) + \frac{\lambda}{2} \|w\|^2$$

ההוספה של הביטוי האחרון דואגת לכך שהמשקל לא יהיה גדול מדי, שהרי רצים למזער את פונקציית המחר, וכך אף לכך שהbeitovi הריבועי יהיה כמו שיתור קטן. בתוספת האיבר=עדכון של המשקלים יהיה:

$$w_{i+1} = w_i - \epsilon \left(\frac{\partial L}{\partial w} + \lambda w \right)$$

הbeitovi זהה דומה מאוד ל-GD רגיל, כאשר נוסף איבר=Aλε. אם $\lambda < 0$, אז ללא קשר לגראדיאנט המשקל יורד בכל צעדים, וזה נקרא "Weight decay"

ניתן לבצע רגולריזציה עם איבר לא ריבועי, מה שמכונה regularization=L1

$$Cost(w; x, y) = L(w; x, y) + \lambda \sum_i |w_i|$$

ואז העדכון יהיה:

$$w_{i+1} = w_i - \epsilon \left(\frac{\partial L}{\partial w} + \lambda \cdot sign(w) \right)$$

בעוד regularization=L2 מושך משקל יחיד וניסיה להקטין אותו, regularization=L1 מושך משקלים רבים וlidol ממספר הפרמטרים של הרשת.

4.4.3 Model Ensembles and Drop Out

עבור פדאטטיקים ניתן לבנות מספר מודלים, ואז כשבאים לבחוקד את הבדיקה מושך את המודלים ולקח Ichet את המוצע. סט המודלים נקרא ensemble. ניתן לבנות מודלים שונים במספר דרכים:

a. לאמן רשות עם אתחולים שונים למשקלים

b. לאמת מספר רשות על חלקים שונים של הדאטא

c. לאמן רשות במספר ארכיטקטורות

יצירוהensemble-בדריכים אלה יכולה לעזור בהכללה, אך יקר ליצור את ה-ensemble ולפעמים קשה לשלב ביהם מודלים שונים

יש דרך נוספת ליצורensemble=**Dropout**, כלומר למחוק באופן אקראי נוירון אחד או יותר. אם יש רשות מסוימת ומוחקים את אחד הנוירונים=למקרה מתקבלים רשות אחרת, ובפועל אפשר לקובלensemble בעזרת רשות אחת של פעם מוחקים ממנו נוירון אחד או יותר. היתרון של יצירוהensemble-בדרך זו הוא שההרשאות חולקות את אותן פרמטרים ולבסוף מקיבלים רשות אחת מלאה עם כל הנוירונים והמשקלים. בפועל עבור כל דוגימה מגරלים רשות (מוחקים כל נוירון בהסתברות $= 0.5$) וכך לומדים במקביל הרבה רשותות שונות עם אותן פרמטרים. באופן זה כל נוירון מוכחה להיות יותר משמעותית בלי אפשרות להסתתר על נוירונים אחרים שיעשו את הלמידה, כיון שלא תמיד הם קיימים. אמנם כל ריצה יחידה יכולה להיות בעלת שונות גבוהה אך המוצע של המשקלים מביא לשינוי נסוכה

בשלב המבחן, לא מפעילים את **Dropout**=אלא לokaneים את כל הנוירונים, כאשר מוחלקים את כל המשקלים בחצי הסיבה לכך היא שניתן להניח שבשלב האימון חצי מהפעמים המשקל היחלוף כיוון שהנוירון הקשור אליו נמחק, ובחצי מהפעמים היה משקל שנלמד. ניתן גם לקחת הסתברות אחרת למחיקת נוירונים, למשקל $= 0.25$, ואז כמשמעותם את כל הרשותות השונות יש לחלק בהסתברות המתאימה. החיסרונו של שיטה זו הוא שלווח לה זמן התוכנו.

4.4.4 Data Augmentation

שיטה אחרת לטעמְנָה-הַפְּתִּיה היא להגדיל אֶפְּסֵט האימון, וכך המודל שנוצר יתאים ליותר דוגמאות. ניתן לעשות זאת על ידי יצירתיות וריאציות של הדוגמאות המקוריות. שיטה זו נקראת **Data Augmentation**, והרעיון הוא לבצע עייפות קטן לכלי-דגם אך שהיא עדין תשמור על המשמעות המקורית שלה, אך תהיה מספקת שונה מהמקורה בכך ליהו-דגם נוספת משמעותית \neq האימון. בדומין של תמונה האוגמנטציה הנפוצות הן:

- סיבוב תמונה בزاوية מסוימת (rotate), הנבחרת מהתפלגות איחידה מהתחום $[0, 2\pi]$.
- הוספת רעש לכל פיקסל, כאשר הרעש משתנה מפיקסל לפיקסל, והוא קטן מ-1.
- שינוי הגודל (rescaling) של התמונה בפקטור מסוים – בדרך כלל הפקטור שייר לתוחוכת $\left[\frac{1}{1.6}, 1.6\right]$.
- שיקוף התמונה (flip).
- מתיחה ומריצה של התמונה (shearing and stretching).

4. References

MLP:

מצגות מהקורס של פרופ' יעקב גולדברג

<https://joshuagoings.com/2020/05/05/neural-network/>

Xor:

<https://www.semanticscholar.org/paper/Simulations-of-threshold-logic-unit-problems-using-Chowdhury-Ayman/ecd5cb65f0ef50e855098fa6e244c2b6ce02fd48>

5. Convolutional Neural Networks (CNNs)

הרשאות שתוארו עד כה הין (FC), Fully-Connected, כל נוירון מחובר לכל הנוירונים בשכבה שלפניו וכל הנוירונים בשכבה לאחריו. גישה זו יקרה מבחן חישובית-ופערוביוטאי צורך בכל הקשרים בין הנוירונים. לדוגמה, תמונה בגווני אפור (grayscale) בעלת $1 \times 256 \times 256 = 65,536$ נוירונים, כאשר כל קשר הינו משקל מהתעדכן במהלך הלמידה. קטגוריות במאזק-מכילה יותר ≈ 1000 נוירונים, אולם יש קשרים בין נוירונים, וכך כל קשר הינו שכבתי מעשן אם יש מספר שכבות הרבה המספר נהיה עצום ממש – וכאן מושג רשות הקשרים והפרמטרים גדלה, ואפוא צזה שבתי מעשן למחזק את הרשות. מלבד-בעי-יה הגדל, בפועל לא תמיד יש קשר-בכל הקשרים, כיון שלא תמיד יש קשר בין כל איברי הכניסה. למשל, עבור תמונה מהמודולר לרשota, במשימות רבות קשר בין פיקסלים ייחודי – בקשר בין פיקסל אחד לשכבה הראשונה ולקשר בין כל שתי שכבות סמוכות שימושות, שכן איקחסיבות-ולחבר את הכניסה לכל הנוירונים בשכבה הראשונה או שכבות קונבולוציה, שאינן מקשורות בזאת-בזאת. כדי להימנע מבעיות אלו לרוב הינה-קדאי להשתמש בשיטות ברשות-האנו או שכבות קונבולוציה, שאינן מקשורות בזאת-בזאת. כל שני נוירונים, אלא רק בין איברים קרובים, כפי שיפורט. רשות מודרנית-ורבו-המבועות על שכבות קונבולוציה, כאשר על גבי המבנה הבסיסי-יבנו ארכיטקטורות מתקדמות.

5.1 Convolutional Layers

5.1.1 From Fully-Connected Layers to Convolutions

האלמנט הבסיסי, ביותר ברשותות קונבולוציה הינו שכבת קונבולוציה, המבצעת קונבולוציה-לינארית על פונקציית $y[n] = \sum_{m=1}^{K-1} x[n-m]w[m]$ בצד יציג אחר ופיסט יותר של. לרוב, שכבת קונבולוציה מבצעת פעולה קרוס-קורלציה בין וקטור המשקלים לבין $x[n]$ – וקטור הכניסה או וקטור היוצא משכבה חניה. וקטור המשקלים נקרא – גרעין הקונבולוציה (kernel), ובעצרתו מבצעת פעולה הקרויס-קורלציה הבאה

$$y[n] = \sum_{m=1}^{K-1} x[n-m]w[m]$$

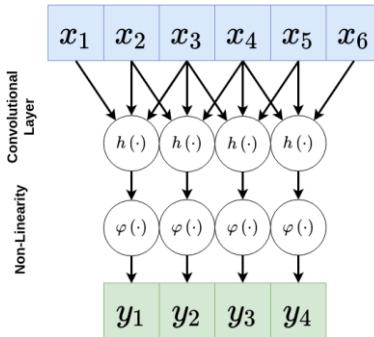
כאשר $x[n] \in \mathbb{R}$ הוא וקטור הכניסה ואיל- $w[m] \in \mathbb{R}$ הוא וקטור המשקלים אשר נלמדים במהלך האימון. וקטו המשקלים-וואזהה לכל הكنيסיות בשכבה וכן מסpter הפרמטרים הנלמד-על-לעומת שכבות CNN הינו קטן בהרבה – שכבות CNN מכילות K משקלים בלבד (לרוגם מתקיים $K \ll N_{inputs} \times N_{outputs}$).

מלבד-הктונת-כמויות המשקלים-השימוש בגרעין קונבולוציה-מסיע לזריה-דפוסים-ולמציאת מאפיינים-יכולה את נבשוף-מאפיי פעולה הקונפלוציה, הבודקת חփוק-בין חלק-מוקטורה-הכניסת-בלב-גרעין הקונבולוציה-הקונבולוציה, יכולה למציאת מאפיינים בסיגנל, ושנעם גרעיני קונבולוציה שיכולים לבצע אוסף פעולות שימושיות, כמו למשל החלקה, גזרת ועוד. אם מטילים על תמונה הרבה גרעינים שונים, ניתן למצוא בה כל מיני מאפיינים – למשל אם הגרעין הוא בזורה של עין או אף, אז הוא מסוגל למציאת האזוריים בתמונה המקוריים הדומים לעין או אף.



איו-1. קונבולוציה-חד מדיה-הביבשתי פונקציית $=_1$ איה-ינט-לב-גבורה – עפרעוש קתק (כחול), $-_2$ איה-ינו גרעין קונבולוציה-המלב-בנ-שרץ על פני כל הישר-כתום). פעולה הקונבולוציה (חוור-בודקת את החיפוי בין הסיגnal בין הגראין, ניתן להראות-שאכן סביר – $x \pm 0.5 \pm 0.1$ יש איזור עם הרבה חיפוי. (b) קונבולוציה זו מחייבת למציאת קוווי מתחאר של בטור תמונה

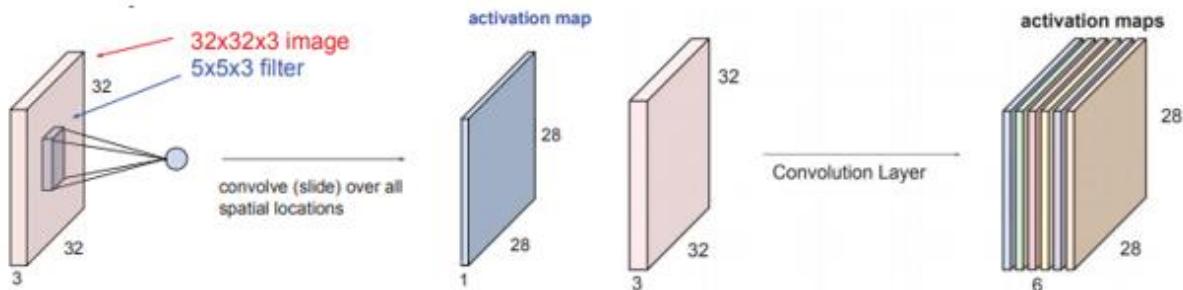
הmoזא של שכבת הקונבולוציה עבר בפונקציית הפעלה לא לינארית (בדרך כלל – tanh או ReLU) והוא מכונה-מפה הפעלה או מפת מאפיינים (feature map). הקונבולוציה יחד עם האקטיבציה נראה כ-



איור 5.2 דיאטרכ אובר דרך שכבה קונבולוציה ולאחריה פונקציית הפעלה, ובowitz מתקבלת מפת אקטיביזציה.

לרוב בכל שכבה קונבולוציה יהיו כמה מסננים – אשככל אחד מהם אמור למודם אפסי – לאחר בתמונה – כל שרשרא הולמת ועמוקה, כך המאפייניות בתמונה אמורים להיווכח בוחנן פעואן חד יותר מאשר השני, וכן המשמעות בשכבות העומקות אמרורים להבדיל בין דברים מרכיבים יתרכז למשל, פעמים רבות ניתן לבדוק אם המשמעות בשכבות הראשונות יזהו את שפות האלמנטים שבתמונה או בצורות אבסטרקטיות – ואילו מסנני בשכבות העומקות יותר יזהו אלמנטים מורכבים יותר כמו איברים או חפצים שלמים בעלי צורה ומוגדרת.

הקלט של שכבה הקונבולוציה יכול להיות רב ערכוי (למשל, תמונה צבעונית המייצגת לרוב בעזרתRGB). במקרה זה הקונבולוציה יכולה לבצע פעולה על כל הערכים יחד ולספק פלט חד ערכוי והוא יכול לבצע פעולה על כל ערך בנפרד ובכך לספק פלט רב ערכיז – גרעין הקונבולוציה יכול להיווכח פסדי, כלומר וקטופשופועל על קלט מסוים, אך הוא יכול להיות גם מממד גבוה יותר – לרוב, מסננים הפעילים על תמונות הינם דו ממדיים – פעולה הקונבולוציה מבצעת בכל שלב כפל בין המסלן לבין איזור דו ממד אחר בתמונה.



איור 5.3 מסנן $\mathbb{R}^{5 \times 5 \times 3}$ פועל על קלט $\mathbb{R}^{32 \times 32 \times 3}$ ואומתකבלת מפה אקטיביזציה $\mathbb{R}^{28 \times 28}$ (ימין). המסלן יכול לעבור דרך מספר מסננים וליצור מפה אקטיביזציה עם מספר שכבות – עברו שישם מסננים הממד של המפה הינה $\mathbb{R}^{28 \times 28 \times 6}$ (ימין).

5.1.2 Padding, Stride and Dilation

כמו ברשף FC, גם ברשת קונבולוציה יש היפר-פרמטרים הנקבעים מראש וקובעים את אופן פעולה הראשית. ישנו שני פרמטרים של שכבה הקונבולוציה – גודל המסלן ומספר ערכי הקלט וכן שלושה פרמטרים עיקריים נוספיםifik הקובעים את אופן פעולה הקונבולוציה:

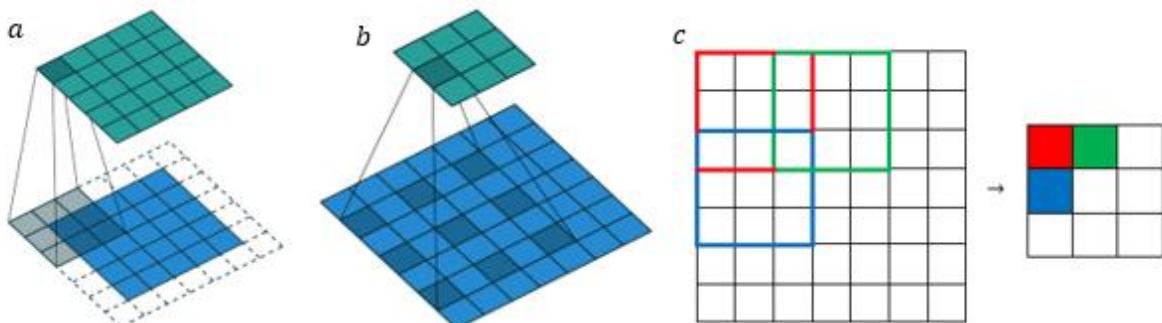
ריפורד (Padding): פעולה הקונבולוציה המוגדרת בעזרת המסקנה פועלה מרחבית, כלומר, המסקנה פועל על מספר איברים בכל פעולה. בנוסף, נשים לב כי פעולה הקונבולוציה לא מוגדרת על איברי הקצוות لكن לא נוכל להפעיל אותה במסנן בנקודות אלו. באירור 5.2 ניתן לראות כיצד פעולה על תמונה בממד של 32×32 מתקבלת 28×28 , דבר הנבע מכך שהקונבולוציה לא מוגדרת על הפיקסלם בקצוות התמונה וכן לא מופעלת עליהם. אף רוצים לבצע את הקונבולוציה גם על הקצוות, ניתן לרפד את שולי הקלט (באפסים או שכפול שערכי הקצה). עובוט מסנן בגודל $K \times K$, גודל הריפורד הנדרש הינו: $\text{Padding} = \frac{K-1}{2}$.

תרחבות (Dilation): על מנת לצמצם עוד במספר החישובים, אפשר לפעול על אזורים יותר גדולים – פתרון הנחיה שעריכים קרובים גיאוגרפית הם בעלי ערך זהה. לשם כך נתקלה בהרחבת פעולה הקונבולוציה תוך השמטה של עריכים קרובים. התרחבות טיפוסית הינה בעלת פרמטר $d = d$.

גודל צעד (Stride): ניתן להניח שלרוב הקשר המרחבי נשמר באזורי קרובים, לכן על מנת להקטין בחישובים ניתן לדלג על הפלט ולהפעיל את פעולה הקונבולוציה באופן יותר דליל – כלומר, אין צורך להטיל את המסקנה על כל האזוריים

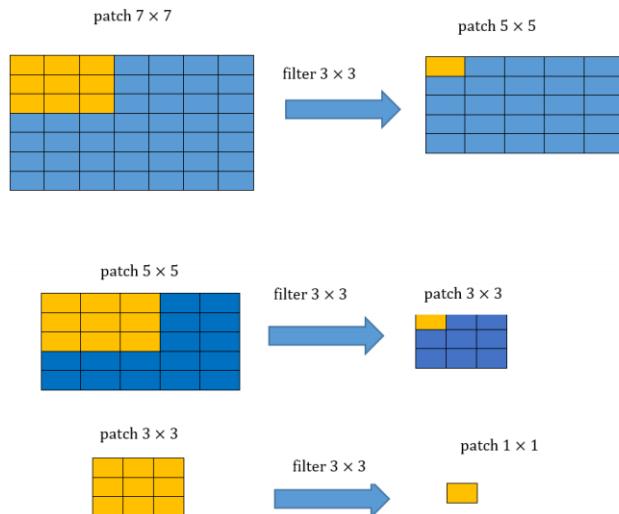
האפשרים בראשת, אלא ניתן לבצע דילוגים, כך שלאחר כל-חישוב קונבנצייה-יבצע דילוג-בגודל הצעד=לפניה הקונבנצייה הבאה. גודל עד טיפוס הינו 2^d .

גודל שכבת הפלט לאחר ביצוע הקונולוציה תלוי בגודלים של הכניסה והמשנן, בריפוד באפסים ובגודל הצעד. גודל שכבת הפלט נקבע על ידי הנוסחה $\frac{W-K+2P}{S} + 1 = 0$, כאשר W הוא גודל הכניסה, K הוא גודל פורמל - ניטן לחישוב גודל שכבת הפלט, P הוא גודל המשנן (padding), S הוא גודל הצעד. גודל המשנן ק-זהה הריפוד באפסים ו-ק-זה גודל הצעד=מספר שכבות הפלט-הינו כמספר המשנן-(casar שכבת פלט יכולה להיות הרבה פעמים) יש לשים לב שערכי ההיפר-פרמטרים (padding, dilation and stride) וכן גודל הגרעון נדרש להיות מספר טבעי אשר מקיים את נוסחת גודל שכבת הפלט(0) הנ'ל, כך ש-ק-זהינו מספר טבעי.



השנתת איברים סמוכים מtar הנחה שכנראה הם דומים. (c) הzzת המסן בעד של $s = 2$.

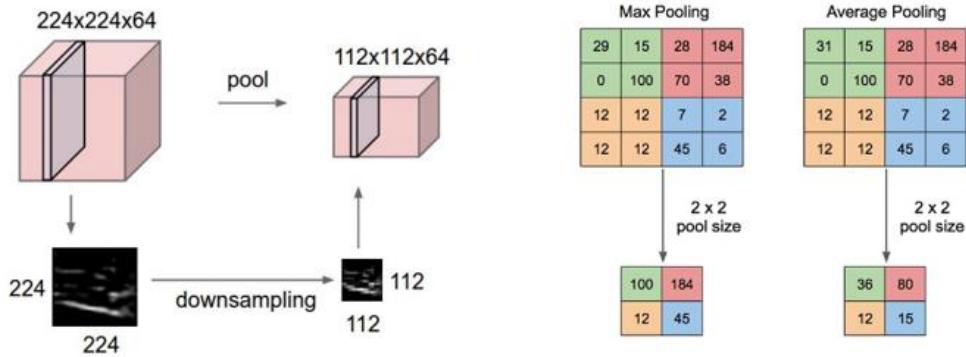
תמרק (Receptive field) של איבר ברשותו מוגדר להיוועוך כל התוחזק בכוון אליו אונטז איבר לאחר השכבות



א/or שלערך מסויים בmoואץ של שלוש שכבות קונבולוציה רצופות עם מסנן בגודל 3×3 . Receptive field 5.5

5.1.3 Pooling

פעמים רבים דатаה מרחבי מאופין בכר שאיברים קרובים דומים אחד לשני, למשל –פיקסלים סמוכים לרוב הרצף בעלי אותו ערך. ניתן לנצל עובדה זו בכך להוריד את מספר החישובים הדרוש בעזרת דילוגיף (Strides) או הרחבת (dilation) כדי שתואר לעיל. שיטה אחרת לניצול עובדה זו היא לבצע פולינג (pooling) –אחורי כל ביצוע קונבולוציה, דגימת ערך ייחד מאזור בעל ערכיים מרובים, המציג את האזור. את צורת חישוב הערך של תוצאת ה-*pooling* ניתן לבחור בכמה דרכי, כאשר המקובלות הן בחירת האיבר הגדול ביותר באזורי שלפּ (max pooling) או את הממוצע של האיברים (average pooling).



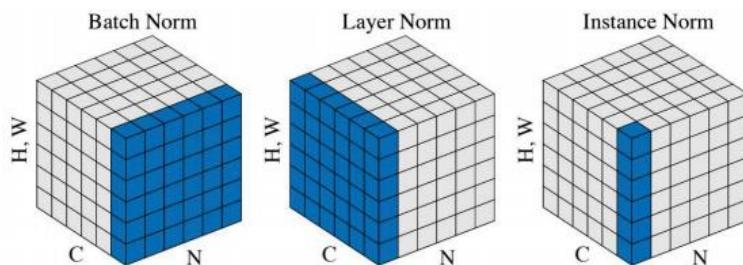
איור 5.6 הקטנת הממד של הדטה בעזרת pooling (שמאל), והמחשה מספרית של ביצוע max/average pooling בגודל 2×2 .

5.1.4 Training

ככל שתהlixir האימון של רשת קומבולוציה זהה לאימון של רשות FC – כאשר ההבדל היחיד הוא בארכיטקטורה של הרשות יש לשימושם למשנים–מפעלים על הרבה אזורים שונים, כאשר המשקלים של המשנים–בכל צעד שווים – וכן אותם משקלים פועלם על אזורים שונים–לשם הפשטוקונינח ויש מסנן ייחודי–כלומר–מטריצה אחת נלמדת על משקלים. מטריצה זו מוכפלת בכל אחד מהאזורים השונים של הדטה, וכדי לבצע עדכון למשקלים שלא ישקלל את הגדריאנטים של כל האזורים השונים – בפועל – הגדיריאנטים של הגדריאנטים על פני כל הדטה, עבור הכלל–בו יש **לא** אזורים שונים עליהם מופעל המסנן הגדריאנט יהיה

$$\frac{\partial L}{\partial w_k} = \sum_{i=1}^N \frac{\partial L}{\partial w_k(i)}$$

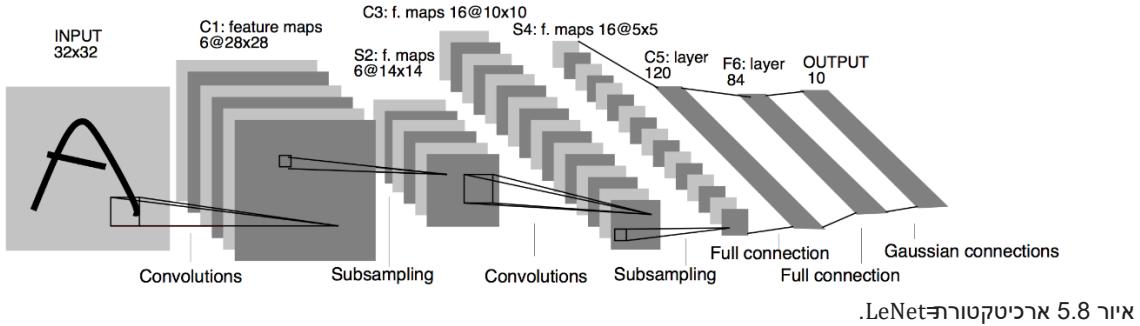
בדומה ל-FC–gam ב-CNN ניתן לבצע normalization של הנרמול על סכום–וקטורי מסוימים (לשם הנוחות נתych פולוקטורי של הדאטא–כטמונות) – אפשרות פשוטה ונפוצה היא לנормל כל–מסנן–בפני עצמה על כמה תמונות (Batch Norm), כלומר לחתות את כל הפיקסל–בסט של תמונות ולנормל בתוחלת ובשונות שלהם – אפשרות נוספת היא לחתך חלק מהມידע של סט תמונות, אך לנормל אותו ביחס לאותם מידע על פנים–מסנן–אחרים (Layer Norm) – יש וריאציות של הנרמולים האלה, כמו למשן–Instance Norm, הלוקט מסנן אחד ותמונה אחת ומnormל את הפיקסלים של אותה תמונה.



איור 5.7 נרמול שכבות של רשת קומבולוציה

5.1.5 Convolutional Neural Networks (LeNet)

בעזרת שרשור של שכבות וחיבור כל האלמנטים השווים לקומבולוציה ניתן לבנות רשת למתקבב מגוון ממשימות שונות – לרוב במקרה שכבות הקומבולוציה יש שכבה אחת או מספר שכבות FC – מטרת ה-FC היא לאפשר חיבור של המידע המוכל במאפיינים שננספו במהלך שכבות הקומבולוציה. ניתן להסתמך על הרשות הcolelatechni שלביב – בשלב הראשון מבצעים קומבולוציה עפומסננ–שונים, לכל אחד מהםנווד לזהות מאפייך, ובשלב השני מחברים זהה את כל המידע שנансף על ידי חיבור כל הננוירונים באמצעות FC – לראשונה השתמש בארכיטקטורה זיכרון – ב-1998, בראשת הנקראות LeNet (על שם Yann LeCun), ומוצגת באירוע 5.8 – רשת זו השיגה דיק ש-98.9% ב-**בזיהוב** – ספורות, כאשר המבנה שלה הואה שתי שכבות של קומבולוציה – ושלוש שכבות FC, כאשר לאחר כל אחת משכבות הקומבולוציה מבצעים pooling.

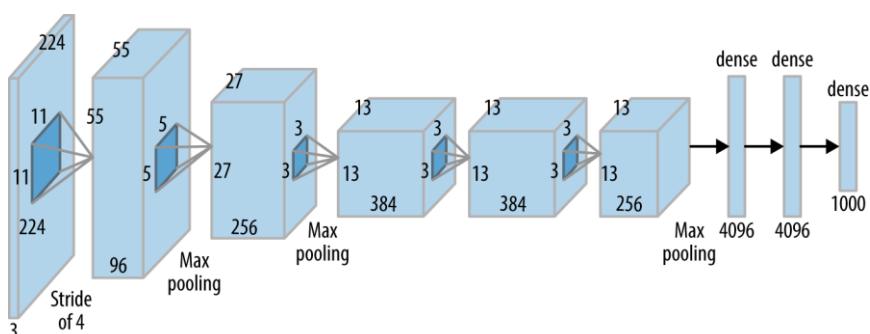


5.2 CNN Architectures

5.2.1 AlexNet

רשת AlexNet ש清华 את ההשראה למבנה שלה מארQUITטורה LeNet, כאשר היכלות שלה להתמודד עם שימושי יותר מרכיבות מסווג GPU=NVIDIA-LeNet מכך שנהיפדתatosים גדולים מאוד שניתן לאמן עליהם את הרשות, ובנוסף כבר היה קיימת שבעזרתו ניתן לבצע חישובים מורכבים.=הארQUITטורה של הרשת מורכבת מחמש שכבות קונבולוציה ושלוש שכבות FC, כאשר לאחר שתי השכבות הראשונות של הקונבולוציה=מתבצע pooling=pooling normalization=ה-softmax הוא מממד $3 \times 224 \times 224$, ומופעלים עליו $96 = \text{מספר ניטרול}= \text{בגודל} = 11 \times 11$, עם גודל צעד $= 4$ =ולא ריפוד באפסים. לכן המוצא של הקונבולוציה הינו מממד $96 \times 55 \times 55 = \text{מספר מקן מתבצע}= \text{max}$ שמחית את שני הממדים הראשוניים, ומתחלקת שכבה במממד $96 \times 27 \times 27$. בשכבת הקונבולוציה השפה יש $=256 = \text{מספר ניטרול}= \text{בגודל} = 5 \times 5$ = עם גודל צעד $= 1 = \text{וריפוד באפסים}= 2 = r$, לכן נמצא הממד הוא שפחים של $27 \times 27 \times 256$, ואחר pooling=max-pooling שכבת במממד $13 \times 13 \times 13 = \text{מספר שכבות של}= 27 \times 27 \times 256$ קונבולוציה עפ"ם שפחים של $1 = \text{וריפוד}= 1 = r$, גודל צעד $= 1 = \text{מספר ניטרול}= \text{בגודל}= 3 \times 3$, והוא שכבת קונבולוציה אחרונה עם קונבולוציה עפ"ם שפחים של $1 = \text{וריפוד}= 1 = r$, גודל צעד $= 1 = \text{מספר ניטרול}= \text{בגודל}= 3 \times 3$, והוא שכבת FC=המוצאים במממד $3 \times 3 = s$. במושג הkonvoluziot יש עוד max-pooling, ואז שלוש שכבות FC=המוצאים במממד $3 \times 3 = s$. המציג 1000 קטגוריות שונות שיש בDATA=55

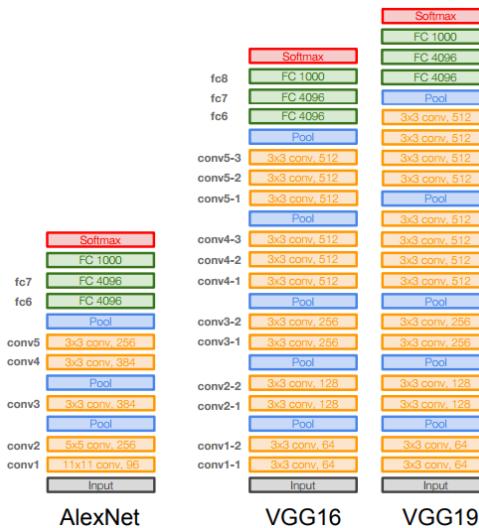
פונקציית האקטיבציה של הרשת הינה ReLU (בשונה מ- LeNet = \tanh) ובהיפר פרמטרים הם: ערך 0.6 מיליאן.



איור 5.9 ארכיטקטורת AlexNet

5.2.2 VGG

שנה לאחר FNet הוצגה בתרומות רשות עמוקה בעלת 19 שכבות המנצלות תרבות-שכבות הקונבולוציה. מפתח הרשת הראוי כי ניתן להחליף שכבת פילטרים של 7×7 בשולש שכבות של 3×3 ולקבל את אותו תמך (receptive field), כאשר מרווחים חסכו משמעותית במספר הפרמטרים הנלמדים. לפילטר בגודל d על עורקי קלט ופלט יש $c^2 d^2$ פרמטרים לעומת $7 \times 7 \times c^2 = 49c^2$ פרמטרים נלמדים ואילו לשולש שכבות של 3×3 יש $c^2 \cdot 3^2 = 27c^2$ פרמטרים נלמדים חיסכו ש-45% הרשת המקורית שפיתחו נקרואה VGG16 והיא מכילה 138 מיליון פרמטרים, ויש לה וריאציה המוסיפה עוד שתי שכבות קונבולוציה ומוכנה GG19.

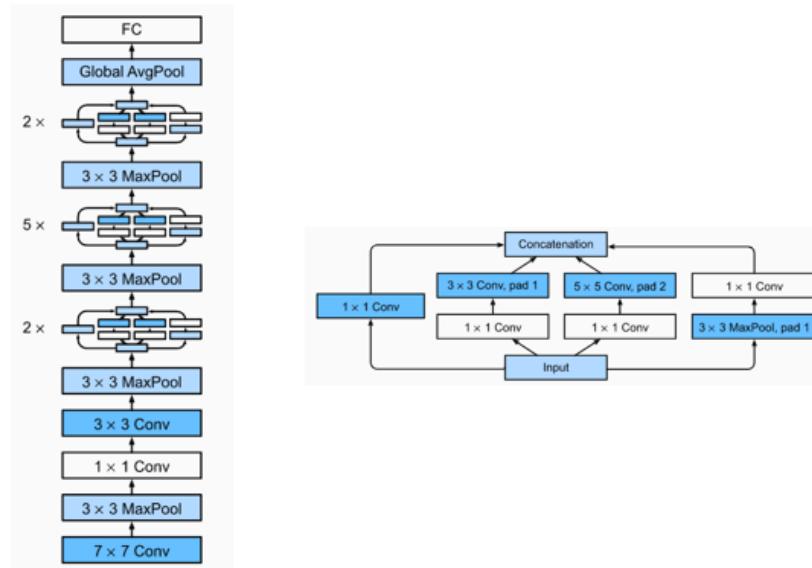


איור 5.10 ארכיטקטורת VGG (ימין) ביחס לארQUITקטורת AlexNet (שמאל).

5.2.3 GoogleNet

המודלים הקודמים היו יקרים חישובית עקב מספר הפרמטרים הגדל. כדי להציג להציג לאוטם ביצועים עם אותו עומק אבל עם הרבה פחות פרמטרים, בוצעת מפתחים מגול הציגו שנקרא inception module. בлок המבצע הרבה פעולות פשוטות במקביל, במקומות לבצע פעולה אחת מורכבת. כל-בלוק מקבל x ומחזק x עליו ארבעה חישובים במקביל, כאשר המגדים של מוצאי כל הענפים שוים כרך שנייתן לשרשן אותם יחד. ארבעת הענפים הם: קונבולוציה 1×1 , קונבולוציה 1×1 ולחירתה קונבולוציה 3×3 עם padding 1 , קונבולוציה 1×1 ולאחר מכן קונבולוציה 5×5 עם padding 2 , ו- 3×3 max pooling עם padding 1 ולאחר מכן קונבולוציה 1×1 . לבסוף, הפלטים של ארבעת הענפים משוררים יחד ומהווים את פלט הבלוק.

המבנה הזה שקל למספר רשותות במקביל, כאשר היתרון של המבנה זה הוא כפול: כמה פרמטרים נמוכה ביחס לרשותות קודמות בחישובים יחסית מהירים כיוון שהם נעשים במקביל. ניתן לחבר שכבות קונבולוציה רגילים עפ' בלוקים אלה, ולקיים רשותה עמוקה הרבה יותר כדי למצוא אפקטיב הינו בין הרכיבים והממינים בכל שכבה המבאים לביצועים אופטימליים.



איור 5.11 Inception Block 5.11 (ימין), וארכיטקטורת GoogleNet מלאה (שמאל).

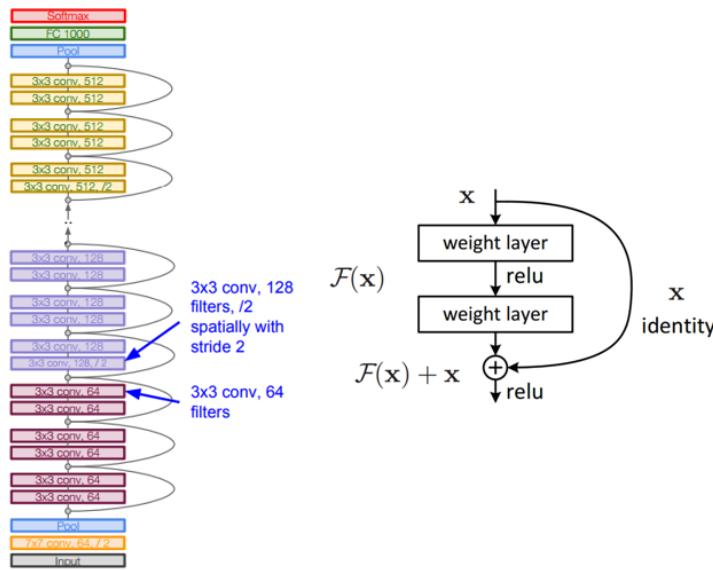
5.2.4 Residual Networks (ResNet)

לאחר שראו שככל שהרשת عمוקה יותר כך היא משיגה תוצאות טובות יותר, ניסרבנות רשותות עם מאות שכבות אך הן השיגו תוצאות פחות טובות מהרשתות הקודמות שהיו בעלות סדר גודל של 20 שכבות. הבעיה המרכזית של הרשותות העמוקות נבעה מכך שללאור מספר שכבות מסוים התקבל ייצוג מספק טוב וולך השכבות היו צרכית לפחות לשנות את הקלט אלא להנבר את הייצוג כמו שהוא. בשיביל לבצע זאת המשקלים בשכבות אלו צריכים להיות 1. הסתבר לשכבות קשה ללמידה את פונקציית הזרחות והקלומה פגעו בתוצאות=אגף-נוסף ברשתות עמוקות בכך מהקיים לבצע אופטימיזציה כמו שצריך למשקלים בשכבות עמוקות.

ניתן לנוכח את הבעיה המרכזית באופן שונא=בביניינך רשותת עפנאי-שכבות, יש טעם להוסיף שכבה נוספת רק אם היא תוסיף מידע שלא קיים עד עכשוויך כדי להבטיח ששכבה תושיף מידע, או לכל הפחות לא תפגע במידע הקיימן. בנו רשת חדשה בעזרת Residual Blocks =יצירת בלוקים של שכבות קוונבולוציה, כאשר בנווסף למעבר של המידע בתוך הבלוק, מחברים גם בין הרכיבים למוצא שלו. כעת אם בלוק מבצע פונקציה מסוימת(x) \mathcal{F} , אז המוצא יינו $x + \mathcal{F}(x)$. באופן זה כל בלוק ממוקד בלמוד-משהו שונה ממה שנלמד עד עכשו, ואם אין מה להוסיף=הfonkציית(x) \mathcal{F} =פשות נשארת=בנוסף, המבנה של הבלוקים מונע מהגדיאנט בשכבות העמוקות להתבדר או להתאפס, והאימון מצליח להתכנס.

באופן זה הפתוח רשותת בעלות 52 שכבות=אשכחציה ביצועים מעולים ביחס לכל שאר הרשותות באותו תקופת השכבות היו מורכבות משלשות של בלוקים, כאשר בכל בלוק יש שתי שכבות קוונבולוציה=בין כל שלשה יש הכפלת Batch normalization=pooling=pooling pooling normalization המהיפר-פרמטרים המומנטום= $r=0.9$,SGD+momentum= 0.9 , Xavier initialization=בשיטה=batch size=256 ומחולק ב-10 בכל פעם שה-error validation מתיישר, weight decay= $1e-5$

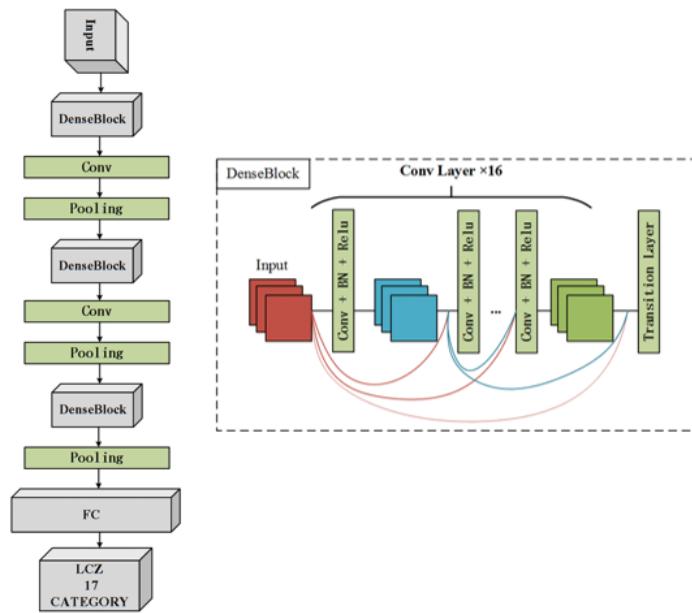
רשתות מתקדומות ותבשייבו את גישת ה-Inception יחיד על ResNet. על מנת לשלב בין היתרונות של שתי השיטות



איור 12 Residual Block 5.12 יחיד (ימין), וארQUITקטורת ResNet מלאה (שמאל).

5.2.5 Densely Connected Networks (DenseNet)

ניתן להרחיב את הרעיון של Residual Block כך שלא רק מ לחברים את הכניסה של כל בלוק למוצא שלו, אלא גם שומרים את הכניסה בפני עצמה, ובודקים את היחס שלה לשכבות יותר עמוקות=Dense block=הו בלוק בעל כמה שכבות הבניי כשבניהם של כל שכבה מחוברת לכל הENSIONS של השכבות אחרות=Nitin כמונן לשרשר כמה בלוקים=Calleיחד ולבצע ביניהם כל מיני פעולות כמחpooling או אפילו שכבת קובולציה עצמאית. כיוון שהשכבות כמה כניסה של בלוקים שונים, יש בעיה של התאמת ממדים, משומן שכל בלוק מגביל את מספר העורצים, חיבור של כמה בלוקים יכולם ליצור מודל מורכב מדי. כדי להתגבר על בעיה זו הוספו שכבות=transition block כבוסף המבצעות קובולציה 1×1 עם רוחב צעד 2 = s, ובכך מספר העורצים נותר סביר ומהודל לא נעשה מורכב מדי.

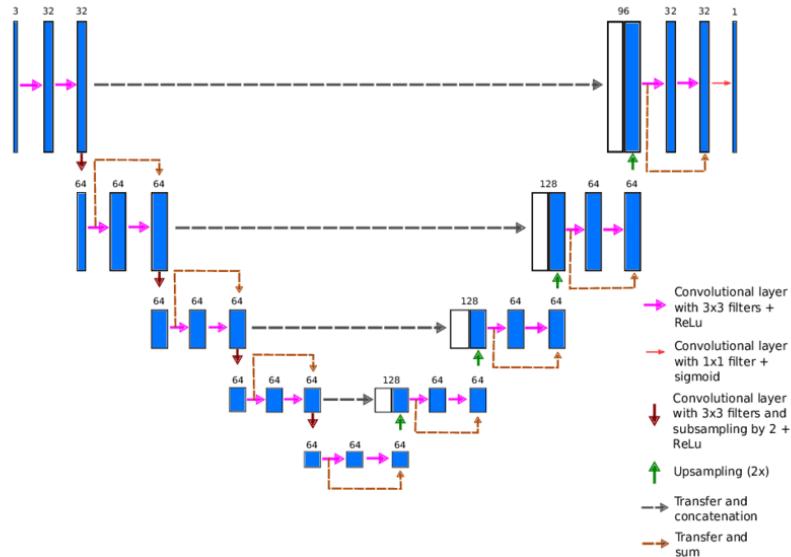


איור 13 Dense Block 5.13 יחיד (ימין), וארQUITקטורת DenseNet מלאה (שמאל).

5.2.6 U-Net

ברשותות קובולצייה מיועדר לסייע, בסוף התהילה פתקבל וקטור של הסטברויות, כאשר כל איבר הוא הסטברות של labels=משתמשים=במשתמשים סגמנטציה זה בעיתוי, כיוון שצריך בסוף התהילה לא רק ללמידה את המאפיינים=שבנתמונה ועל פיהם לקבוע מה יש בתמונה, אלא צריך גם לשחרר אפקטומי הפיקסלים והתיוגים שלהם ביחס לתמונה המקורית עם הסגמנטציה המתאימה=כדי להתמודד עם בעיה זהה恰אנו את ארQUITקטורת U-Net, המכילה שלושה חלקים=עיקריים=כיצוץ, צוואר בקבוק והרחבה (contraction, bottleneck, and expansion section).

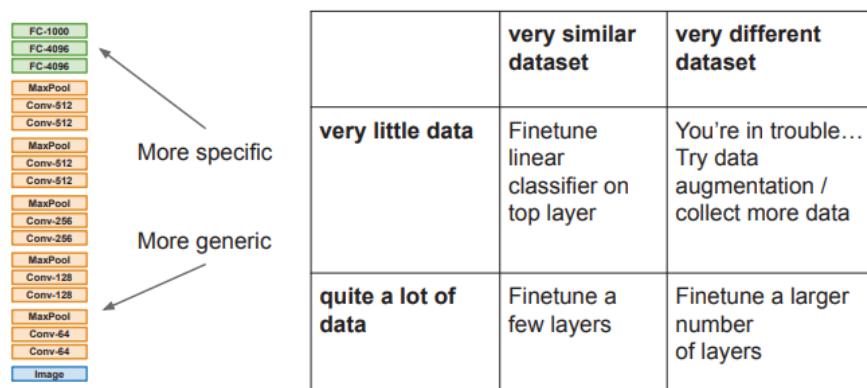
באיור, בחלק הראשון יש טופולוגיה רגילה של רשת קונבולוציה, המבוצעת בעזרת שכבות קונבולוציה וביצוע pooling השוני בין השלב זהה לבין רשת קונבולוציה קלאסית הוא החיבור שיש בין כל שלב בתהילר לבין חלקים בהמשך התהילר=
לאחר המעבר בצדואר הבקבוק יש למשה שחזור של התמונה עם הסגמנטציה=השזוזנעה בעזרת sampling-sampling-קביעת הוקטוטה התקבל במצוא צוואר הבקבוק יחד עם המידע שנשאר מהחלק הראשון של התהילר פונקציית ההמחיר המשמשים בرشת זו נקראות pixel-wise cross entropy loss והבודקת כל פיקסל ביחס ל-label האמתי אליו הוא שייך



איור 5.14 ארכיטקטורת Net-U.

5.2.7 Transfer Learning

כאשר נתקלים במשימה חדשה, אפשר לתקן עבורה ארכיטקטורה מסוימת ולאמן רשת עמוקה. בפועל זה יקר ומסובך להתאים רשת מיוחדת לכל בעיה ולאמן אותה מהתחלתה, ולכן ניתן להשתמש ברשנות הקיימות שאומנו כבר ולהתאים אותן לביעות אחרות. גישה זו נקראת **Transfer Learning**, וההגיאון מאחרורית-טוען שעבור כמעט כל סוג דאטה השכבות הראשונות למודות אותו דבר (ziehi שפות, קווים וצורות כלליות, מאפיינים כלליים וכו') וכן ניתן להשתמש בהן פעמים רבות ללא שינוי כלל. משום כך, בפועל בדרך כלל לוקחים רשת קיימת ומחליפים בה את השכבות האחרונות אֲחֵמָסִיפִים לה עוד שכבות בסופה, ואֲמַמְנִים את השכבות החדשונען הדאטה החדש כך שהתקהינה מוכנות לדאטה הספציפי של המשימה החדשה. ככל שיש יותר דאטה חדש ניתן להוסיף יותר שכבות ולקבל דיווק יותר טוב, וככל שהמשימה החדשה דומה יותר למשימה המקורית של הרשת כך יש צורך בפחות שכבות חדשות. כמו כן, משום שבשיטה זו נדרשם לאמן מספר שכבות קטן יותר, קטן מהבע ממה חוסר בדאטה



איור 5.15 Transfer Learning 5.15.

5. References

Convolutional:

<https://github.com/technion046195/technion046195>

מצגות מהקורס של פרופ' יעקב גולדברג

AlexNet:

<https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecc96>

VGG

<https://arxiv.org/abs/1409.1556>

GoogleNet

http://d2l.ai/chapter_convolutional-modern/vgg.html

ResNet

<https://arxiv.org/abs/1512.03385>

DenseNet

<https://arxiv.org/abs/1608.06993>

<https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>

U-Net

<https://arxiv.org/abs/1505.04597>

6. Recurrent Neural Networks

היתרוף של שכבות קונבולוצי-על-פנ- C^2 הוא ניצול הקשר המרחבי שיש בין איברים שונים בדאטא, כמו למשל פיקסלים בתמונה. יש סוגים מסוימים יוצרים סדרה שיש לסדר האיברים חסיבות, כמו למשל טקסט, גלי קול, רצף DNA ועוד. כמובן שדאטא מהסוג זהה דורש מודל הנוטן חסיבות לסדר של האיברים, מה שלא-קיי-פרשות קונבולוציה. בנוסף, הרבה פעמים הממד של הקלט לא ידוע או משתנה, כמו למשל מספר המיל-פ' במשפט, וגם לכך יש לתת את הדעת. כדי להתמודד עם אטגרים אלו יש לבנות ארכיטקטורה שמקבלת סדרה של וקטורים וממציאות וקטור יחיד, כאשר הווקטור היחיד מקודד קשרים על הדאטא המקורי-שנכנס אליו. את וקטור המוצא ניתן להעביר בשכבה C^2 או בכל מסוג אחר, תלוי באופן המשימה

6.1 Sequence Models

6.1.1 Vanilla Recurrent Neural Networks

רשתות רקורסיביות הן הכללה של רשתות נוירונים עמוקות, כאשר הן מכילות משקלות המאפשרות להקלת המשמעות לסדר של איברי הכנסה. ניתן להסתכל על משקלות אלה כרכיב זיכרון פיני, כאשר כל איבר שוכן משקל בלבד ביחס לפונקציה קבועה-בתוספת רכיב משתנה שתלו ערך העבר-כאשר נכנס וקטור אחד הוא מוכפל במשקל w_{xh} ונכנס לרכיב זיכרון h_t , כאשר h_t הוא פונקציה של x_t, h_{t-1} :

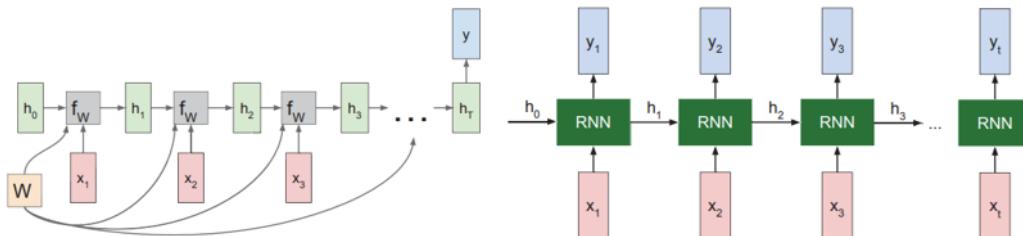
$$h_t = f(h_{t-1}, x_t)$$

מלבד המשקלים הפעילים על וקטור הכנסה, יש גם משקלים שפועלים על המשקלות הפנימיות (רכיב הזיכרון) – משקלים הפעילים על המוצא-של רכיב זה – w_{hh} – המשקלים w_{hx}, w_{hy} זהים לכל השלבים, והם מתעדכנים ביחס לפונקציה f – קיימת קבוצה לכל האיברים, למשקל w_{hh} sigmoid=tanh, למשקל w_{hx} ReLU, למשקל w_{hy} tanh. התהיליך נראה כך:

$$h_t = f_W(w_{hh}h_{t-1} + w_{xh}x_t), f_w = \tanh/ReLU/sigmoid$$

$$y_t = w_{hy}h_t$$

באופן סכמתי התהיליך נראה כך:

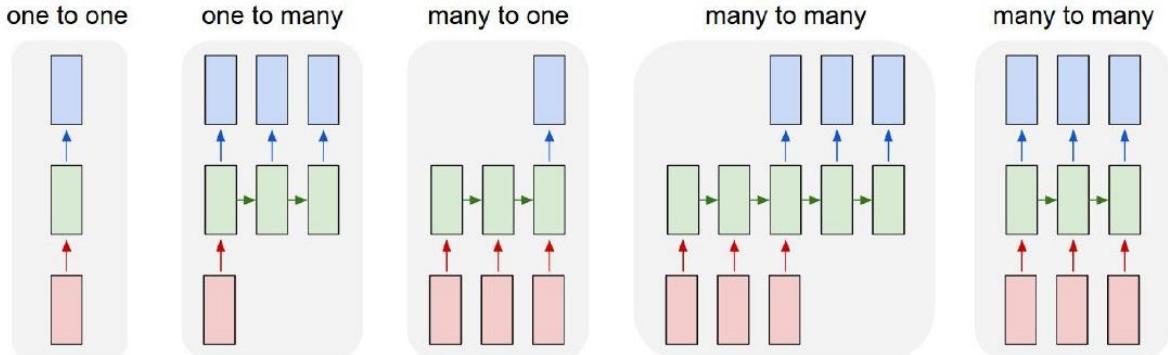


איור 6.1 ארכיטקטות RNN בסיסיות: (משמאל) Many to One (מיין)-One to many (מיינן) – על כל חץ יש משקל מתאים עליו מתבצע הלמידה

כמובן שניתן גם לשרשן שכבות חビות ולקבל רשת عمוקה, כאשר פלט של שכבה מסוימת הופך להיות הקלט של השכבה הבאה. ישנו מודלים שונים של RNN, המתאים לביעיות שונות:

One to many – יש קלט יחיד ורוצים להוציא סדרת-פלטים, למשתמכנים תמונה לרשת ורוצים משפט שייתאר אותה – One to one – רצים לקבוע שהוא ייחיד עבור קלט-בסיסדי, למשל מקבלים משפט ורצו-פלטסווג את הסנטימנט שלו – האם הוא חיובי או שלילי.

Many to many – עבור כל סדרת קלט יש סדרת פלט, למשל תרגום-משפה אחות לשפה אחרת – מקבלים משפט וממצאים משפה



איור 6.2 מודלים שונים של NN²

6.1.2 Learning Parameters

$x = (x_1, \dots, x_n), (y_1, \dots, y_n)$, הגדיר את פונקציית המחר':

$$L(\theta) = \frac{1}{n} \sum_i L(\hat{y}_i, y_i, \theta)$$

כאשר הפונקציה $L(\hat{y}_i, y_i, \theta)$ מושגנת המשמש בעבור משימות סיווג ובעור בעיות ריגראטיון שמשתמש בקריטריון MSE. האימון יבוצע בעזרת GD. אך לא ניתן להשתמש backpropagation הרגיאטטיון שכך משקל מופיע מספר פעמיות – למשקל w_{hh} פועל על כל הכניסות $-x_{hh}$ – פועל על כל רכיבי הזיכרון כדי לבצע backpropagation through time (BPTT). המשקל w_{hh} מושתכל בעל הרשת הנפרשת כרשת אחת גדולת מחשבים את הגרדיינט עבור כל משקל, ואז סוכמים או מוצאים את כל הגרדיינטים. אפקט אוטובכניוס הוא בגודל a , כלומר יש עדגימות בזמן, אז יש רק זיכרון אחד – ומשקל w_{hh} . لكن הגרדיינט המשוקל יהיה

$$\frac{\partial L}{\partial w_{hh}} = \sum_{n=1} \frac{\partial L}{\partial w_{hh}(t)} \quad \text{or} \quad \frac{\partial L}{\partial w_{hh}} = \frac{1}{n-1} \sum_{n=1} \frac{\partial L}{\partial w_{hh}(t)}$$

כיוון שהמשקלים זהים לאורך כל הרשת, $w_{hh} = (t)$ והשני בזמן יהיה רק לאחר ביצוע ה-BPTT יהיה רלוונטי רק לקטור הבא:

הצורה הפשוטה של ה-BPTT יוצרת בעיה עם הגרדיינט. נניח שרכיב הזיכרון מיוצג באמצעות הפונקציה הבאה

$$h_t = f(z_t) = f(w_{hh}h_{t-1} + w_{hx}x_t + b_h)$$

לפי כלל השרשרת:

$$\frac{\partial h_n}{\partial x_1} = \frac{\partial h_n}{\partial h_{n-1}} \times \frac{\partial h_{n-1}}{\partial h_{n-2}} \times \dots \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial x_1}$$

כיוון ש- w_{hh} קבוע ביחס בזמן לעבר וקטורי כניסה יחיד, מתקבל:

$$\frac{\partial h_t}{\partial h_{t-1}} = f'(z_t) \cdot w_{hh}$$

אם נציב זאת בכלל השרשרת, נקבל שעבור חישוב הנגזרת $\frac{\partial h_n}{\partial x_1} = \frac{\partial h_n}{\partial h_{n-1}}$ – מכפליים 1 – ופעמים ב- w_{hh} . אך אם מתיקים $1 > |w_{hh}|$ – אז הגרדיינט יתבדר, ואם $=1$ – הגרדיינט יתאפס. של התבדורות או התאפסות הגרדיינט, יכול להופיע גם בשרותות אחרות, אבל בגלל המבנה של RNN – והlinearיות של ה-BPTT – בשרותות רקוריוביות זה קורה כמעט תמיד clipping.

בעור הבעיה של התבדורות הגרדיינט ניתן לבצע clipping – אם הגרדיינט גדול מקבוע מסוים, מנורמלים אותו:

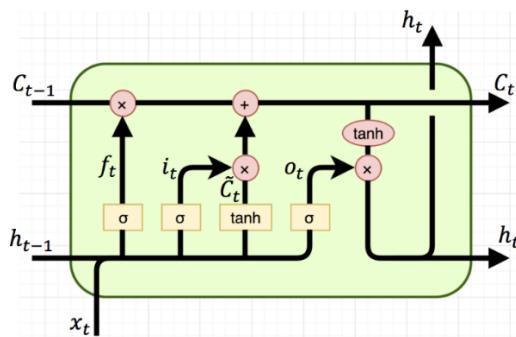
$$\text{if } \|g\| > c, \text{ then } g = \frac{cg}{\|g\|}$$

הבעיה של התאפסות הגראדיאנט=Aטמן לא גורמת לחישובים של מספרים עצומים, אך היא בעצת=mbטלת את השפעה של איברים שנמצאים רחוק אחד מהשני. אם למשל יש משפט ארוך, אז במקורה בו הגראדיאנט=Dזע-במהלך ה-BPTT=B-כמעט ואין השפעה של המילה הראשונה על המילה الأخيرة. במילים אחרות=Bהתאפסות הגראדיאנט=Gוררת עירקה=Long-term, יכולր קשה ללמידה=Dאט-הבעל תלות בטוווח ארוך, כמו משפט ארוך או תופעות שימושיות לאט. בגלל הבעיה זו לא משתמשים ב-RNN=RNN=הקלאסו=שנקרא גפ=Vanilla RNN), אלא מבצעים על-יפיורים, כפי שIOSBR בפרק הבא

6.2 RNN Architectures

6.2.1 Long Short-Term Memory (LSTM)

כדי להתגבר על בעיה זו, הגדיר אנטנה מוגנת מהרשת לשימוש בזיכרון ארוך טווח כדי לרשום מחדש את המידע על העבר, אלא שהיא גם בעל שליטה על איקריום מהרשת לשימוש בזיכרון קצר. RNN-הפשוט לרכיב הזיכרון יש שתי כניסה x_{t-1} , h_t ובעזרת מכחישים את המוצא על ידי שימוש בפונקציית f_w למשה רכיב הזיכרון הוא קבוצה של מידע מתבצעה רק במקלט-ב-ΜΥΝ-פ' שמייניות עיקריות (x_t, h_{t-1}) . מלבד הכניסות הרגילות, עוד קיימות הנקראות memory cell state ומסומנת ב- c_{t-1} , ובונוס לכ- c_t מחושב בצורה מורכבת יותר-בأfon הזיהalarmant t . זאתאג לחיצון ארוך טווח של דברים, c_t אחראי על הזיכרון של הטעות הקצר. נתבונן בארכיטקטורה של תא הזיכרון



איור 6.3 תא זיכרון בארכיטקטורת MST

הצמוד x_{t-1} , h_t , σ]=ונכון לאותם מוכפל במשקל=A=ולאחר מכעובר בנפרד דרך ארבעה שערifs(יש לשים לב שלא מבצעים פעולה ב**ק-אבל**= $t-1$ =אלא הם נשאים בנפרד ואת כל הפעולות עושים על כל איבר בנפרד)=**השער הראשי** ב $[x_t(\sigma), \sigma(h_{t-1})]$ = σ =זהו שער שחחה והוא אחראי על מחלוקת חילקמהיז'ירון=A=אם למשלי'ש משפט ומופיע בז' נושא חדש, אז שער זה אמור למחוק את הנושא שהוא שומר בז'יכרונות=השער השנוי=זהו שער זיכרונו אחראי על כמרא'יש לזכור את המידע החדש שלטווח אורך=A=לדוגמא אבן יש במשפט מסוים נושא חדש, אז השער=יחיליט שיש לזכור את המידע הזה=A=אם לעומת זאת המידע החדש הוא תיאור שלא רלוונטי להמשך=A=اذן טעם לזכור אורך השער הריבועי=t=זהו שער מוצא והוא אחראי על כמלה מהמידע ולוונט בלדאטרה הנוכח=t,x, ככלומר מה יהיה הפלט של התא בהינתן מידע העבר. שלושת השערים האלו נקראים מסכות(Masks), והם מקבלים ערךים ב**ק-אבל**-**ק-כיווק** שהם עוברים דרך סיגמאיד=ישער נוטף=t(לפעמים מסומן באותג) שאחראי על השאלה כמלה מהיז'ירון להעיבר לתא הבא. שער זה משלב את המידע המתkeletal חד עפקן, שאומר עד כמה יש לזכור להמשר את המידע החדש.

בapon זהה מקבלים \hat{h}_t שאחראי על היזכרן לוטוח הקצץ כמו ב-RNN-Vanilla, והן את c_t שאחראי על זיכרוך של כל העבר-ארכיטקטורת הרכיב אפשרה להתychס לאלמנטים נוספים הקשורים ל \hat{h}_t . ניתן לשוכח חלקים לא רלוונטיים של התא (f_t) להתייחס בapon סלקטיבי (c_t). וולוחזיא רק חלק מה מידע המשוקל התא (c_t). בapon פורמלי ניתן לנתח את פעולות התא כה

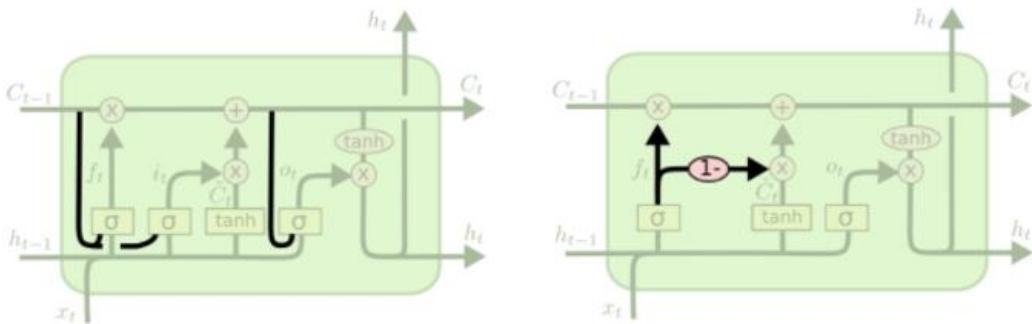
$$\begin{pmatrix} i \\ f \\ o \\ \tilde{c} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} = \begin{pmatrix} \sigma(w_i \cdot [x_t, h_{t-1}] + b_i) \\ \sigma(w_f \cdot [x_t, h_{t-1}] + b_f) \\ \sigma(w_o \cdot [x_t, h_{t-1}] + b_o) \\ \tanh(w_{\tilde{c}} \cdot [x_t, h_{t-1}] + b_{\tilde{c}}) \end{pmatrix}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, h_t = o_t \odot \tanh(c_t)$$

כשהי אופרטור Θ מסמל כפל איבר איבר (כיוון של שערם נכנס הזוג $[x_t, h_{t-1}]$, אם במושג מוצאים מכפלת מסוימת, יש לבצע אותה על כל אחד מהאיברים \underline{x}).

יש וריאציות שונות של רכיב $MSTF$, ניתן לمثال לחבר ארכיטקטורה $t-1$ - t לא רק למצוא t אלא גם לשאר השערים. חיבור כזה נקרא **peephole**, כיון שהוא מאפשר לשערים להתבונן ב- $t-1$ - t בראופן ישירות שארcitקטורות שמחברות את $t-1$ - t כל השערים, ויש ארcitקטורות שמחברות אותו רק לחלק מהשערים=חיבור כל השערים $t-1$ - t משנוכחותם. את משגאות השערים. במקופת $(w \cdot [x_t, h_{t-1}] + b)$, המשווהה החדשה $t-1$ (w).

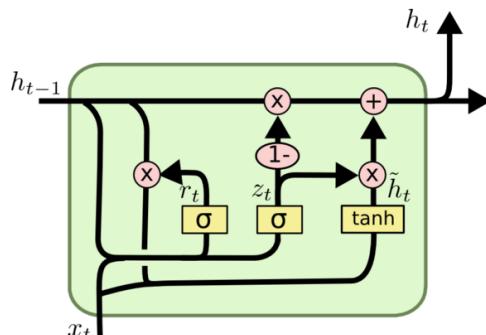
וריאציה אחרת של LSTM מאחדת ב'קשע' השכחה c_t לפני שער הזיכרון f_t , וההחלטה עד כמה יש למחוק מידע מהזיכרון מתකבלת יחד עם ההחלטה כמה מידע חדש יש לכתוב בשינוי זה ישפיע על ה-*memory cell*, memory cell, כאשר במקום



איור 6.4 וריאציות של LSTM עם תאים זרים (ימין) – coupled forget and input gates (שמאל).

6.2.2 Gated Recurrent Units (GRU)

ישנה ארכיטקטורה נוספת של תאצ'িרון הנקראת Gated Recurrent Units (GRU), והיא כוללת מספר שינויים ביחס ל-LSTM:



איור 6.5 תא זיכרון בארכיטקטורת GPU

השינוי המשמעותי מ-LSTM הוא העבודה שאותה memory cell state, וכל השערים מתבססים רק על הקלט והמוחזק של התא הקודם כדי לאפשר זיכרון הן לטוח אורך והן לטוח קצר, יש שני שערים – Update gate ו-Reset gate, והם מחושבים על פי הנוסחאות הבאות:

Update: $z_t = \sigma(w_z \cdot [x_t, h_{t-1}])$

Reset: $r_t = \sigma(w_r \cdot [x_t, h_{t-1}])$

בעזרת שער ה-*reset* מחשביםCandidate hidden state

$$\tilde{h}_t = \tanh(w \cdot [x_t, r_t \odot h_{t-1}])$$

ראשית ישלים לפכְּ $\in_t [0, 1]$ שהוא תוצאה של סיגומואיד. כעת נתבונע ב- y_t ביחס לרכיב זכרוקפיטוט ש- t מתקבל מ- $f_w(w_{hh}h_{t-1} + w_{xh}x_t)$. אסף- t קרוב- t -1 מתקבל הביטוי $=$ Vanilla RNN

$$\tilde{h}_t = \tanh(w \cdot [x_t, r_t \odot h_{t-1}]) \approx \tanh(w[x_t, h_{t-1}]) = \tanh(w_{hx}x_t + w_{hh}h_{t-1})$$

המשמעות היא ש- $r_t \rightarrow 1$ מתקבל רכיב הזיכרון הקלאסי, השומר על זיכרון לטוח קצר. אם לעומת זאת $r_t \rightarrow 0$ אז מתקבל רכיב הזיכרון \tilde{h}_t $\approx \tanh(w \cdot [x_t, 0] \odot h_{t-1}) = \tanh(w_{xh}x_t + w_{hh}h_{t-1})$, ולמעשה הזיכרון של הטווח הקצר מתאפס (reset).

לאחר החישוב של \tilde{h}_t מחשבים את המוצא של המצב החבוי בעזרת z_t , שגם הוא מקבל ערכים בין 0 ל-1.

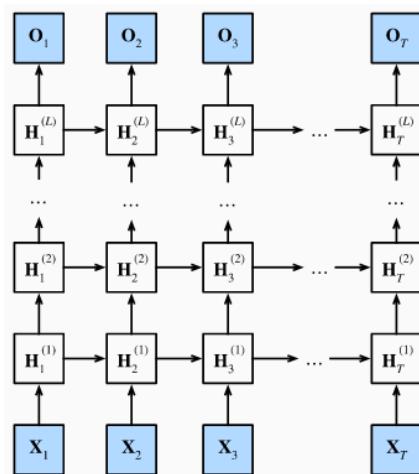
$$h_t = (1 - z_t)\Theta h_{t-1} + z_t\Theta \tilde{h}_t$$

אם $z_t \rightarrow 0$, אז $h_t \approx h_{t-1}$, כלומר מתחשבים ב- \tilde{h}_t ולמעשה מעבירים את המצב הקודם כמו שהוא. אולם אם $z_t \rightarrow 1$, אזCandidate hidden state $h_t \approx \tilde{h}_t$,Candidate hidden state מה המצב הקודם כמו שהוא ולוקחים אותו כמזההCandidate hidden state עבור כל ערך אחר של z_t ,Candidate hidden state מקבלים שקול של המצב החבוי הקודם וה>New hidden state.

ארכיטקטורה זו מאפשרת גם לזכור דברים לאורק זמן, וגם מצילה להסתמוך עם בעיות הגראדיאנט. אפשר שער העדכון הקרוב ל- t כל הזמן, אז בעצם מעבירים את המצב החבוי כמו שהוא, ולמעשה הזיכרון נשמר לאורק זמן. בנוסף, אין בעיה של התבדדות הגראדיאנט, כיון שאין שאמם השינוי בין תא לתא לא גדול, אז הגראדיאנט קרוב ל- $\frac{1}{T}$ כל הזמן ולא מתבודד.

6.2.3 Deep RNN

עד כה דובר על רכיבי זיכרון ייחודיים, שנitinן לשרשר אותם יחד ולקבל שכבה שיכולה לנתחזניתן להרחב את המודול הפשוט לרשת בעל מספר שכבות עומוקות



איור 6.6 ארכיטקטורת Deep RNN

נתאר את הרשת באופן פורמלי=בכל נקודת זמן יש וקטור כניסה $x_t \in \mathbb{R}^{n \times d}$ (וקטור בערך איברים), כאשר כתיב והוא ממה=d=איברי הסדרה נכון לשכבות בערך= L שכבות דגש איברים בכניסה, כאשר עבור כל נקודת זמן יש= L שכבות (א-מצבים חווים). כל שכבה מכילה את מצבים חווים= c אשכח שכבה בפניהם נקבעות $H_t^{(L)} \in \mathbb{R}^{n \times h}$ בכל נקודת זמן יש גם וקטור מוצאים אורקיז= $o_t \in \mathbb{R}^{n \times q}$. סמן $x = H_t^{(0)}$, ונניח שב нескבב אחת לשניה משתמשים באקטיבציה לא לינארית. בעזרת סימונים אלה נקבל את הנוסחה הבאה

$$H_t^{(\ell)} = \phi_\ell \left(H_t^{(\ell-1)} w_{xh}^{(\ell)} + H_{t-1}^{(\ell)} w_{hh}^{(\ell)} + b_h^{(\ell)} \right)$$

כאשר $w_{xh}^{(\ell)} \in \mathbb{R}^{h \times h}$, $w_{hh}^{(\ell)} \in \mathbb{R}^{h \times h}$, $b_h^{(\ell)} \in \mathbb{R}^{1 \times h}$ הם הפרמטרים של השכבה החבוייה וה- ℓ . הפלט שתליי באופן ישר רק בשכבה ה- L , והוא מחושב על ידי:

$$o_t = H_t^{(L)} w_{hq}^{(L)} + b_q^{(L)}$$

כאשר $w_{hq}^{(L)} \in \mathbb{R}^{h \times q}$, $b_q^{(L)} \in \mathbb{R}^{1 \times q}$ הם הפרמטרים של שכבת הפלט

ניתן כמובן להשתמש במצבים החבויים-ברכיבי זיכרון GRU או LSTM, אך לקבל Deep Gated RNN.

6.2.4 Bidirectional RNN

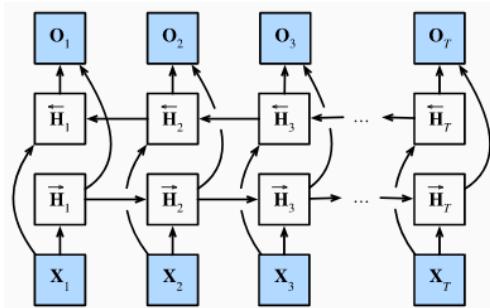
כל הרכיבים והרטות שנידנו עד כה עוסקים בסדרות סיבתיות, כלומר, סדרות בהקל איבר מסווף מקודמי או אף לא מallow הבאים אחריו. למשל, ערך מניה ביום מסוים קשור לערכיים הקודמים, אך הערך שלא ביום המחרת (שכל עוד לא ידוע) לא משפיע בשום צורה על ערכיהיים הנוכחים=דוגמאניסופר=ההתפתחות של גל בזמן תליה בערכי הקודמים של הגל אף אינה משפעת מוצבי הגל בעתיד=זהה אמן המצבה הייתר מצוי=אך ישנו מצבים בהם יישׂודר דלאו דווקא סיבתי, כפניתן לבדוק את הקשר בין איבריה משני הכוונים=נוקח לדוגמא את משימת ההשלמה הבאה:

I am _.

I am _ hungry.

I am _ hungry, and I can eat a big meal.

כעת נניח שבכל אחד מהמשפטים צריך לבחור את אחת מהמיליות שבס={}happy, not, very={}=כמובע שטוף הביטוי, במקורה וק"י, תורף=מידע משמעות על איזו מילה לבחור. מודל שאינו מסוגל לנצל את הידע לאחר המילה החסורה יכול לפספס מידע חשוב, ולרוב יכול לנחש מילה שאינה מסדרת עם המשך המשפט מהינה תחבירית וች מבוחנת המשמעות. כדי לבנות מודל שמתיחס לכל חלק המשפט, יש לתכנן ארכיטקטורה שמאפשרת לנתח סדרה שני הכוונים שלה=ארQUITקטורה זו נקראת Bidirectional RNN, והיא לומשה מבצעתנית של סדרה משפה הכוונים שלה במקביל. באופן זה כל מצב חבוי נקבע בו זמן על ידי הנתונים של שני מצבים חבויים אחרים – זה שלפניו וזה לאחריו



איור 6.6 ארכיטקטורת RNN Bidirectional.

עבור כל כניסה $x_t \in \mathbb{R}^{n \times d}$ נחשיב מקביל שני מצבים חבויים= $\vec{H}_t \in \mathbb{R}^{n \times h}, \bar{H}_t \in \mathbb{R}^{n \times h}$, כאשר זהה מספרם בזיכרון בכל מצב חבוי. כל מצב מחושב באופן הבא:

$$\vec{H}_t = \phi(x_t w_{xh}^{(f)} + \vec{H}_{t-1} w_{hh}^{(f)} + b_h^{(f)})$$

$$\bar{H}_t = \phi(x_t w_{xh}^{(b)} + \bar{H}_{t+1} w_{hh}^{(b)} + b_h^{(b)})$$

כאשר $w_{xh}^{(b)} \in \mathbb{R}^{d \times h}, w_{hh}^{(b)} \in \mathbb{R}^{h \times h}, b_h^{(b)} \in \mathbb{R}^{1 \times h}$ הם הפרמטרים של המודל=לאחר החישוב של \vec{H}_t = \bar{H}_t משרשים אותם יחד ומתקבלים את $H_t \in \mathbb{R}^{n \times 2h}$, ובעזרת מחשבים את המוצא:

$$o_t = H_t w_{hq} + b_q$$

כאשר $w_{hq} \in \mathbb{R}^{2h \times q}, b_q \in \mathbb{R}^{1 \times q}$ הם הפרמטרים של שכבת הפלט

6.2.5 Sequence to Sequence Learning

ב

<https://buomssoo-kim.github.io/blog/tags/#attention-mechanism>

6. References

Vanilla:

<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

LSTM, GRU:

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Deep RNN, Bidirectional RNN:

http://d2l.ai/chapter_recurrent-modern/index.html

7. Deep Generative Models

המודל פשחצגו בפרקם הקודמים הינם מודלים פסיקרי פונטיביים מקוריים הנטמאו מיניפלכט בעלות על בסיס-פז'אטי נתון, אך לא יכולים ליצוף פיסוקים מיידניים דוגמתו חישוש וביצמפה-בניגוד אליה הפקה. מודלים גנרטיביים מ- $\mathcal{R}^{n \times d}$ או אף ליצור פיסוקים מיידניים הדגשוו עלי בסיס הדוגמאות שנלמדו-בałופ פורמלי-בבה' נתקאים-ב-דוגמאות \mathcal{R}^d או אף תגיות \mathcal{R}^d , מודל דיסקרמינטיבי-פמאון לשער את ההסתברות $(x|y) = \text{מודל גנרטיבי-בלועמת זאותלומד א-ב-הסתברות}(y, x) \Pr(x)$ (א-ב-מקורה שהtagיות אין נתנות) כאשר y , x צמד נתון של-דוגמאות-abel.

ישנם שני סוגים עיקריים של מודלים גנרטיביים: סוג אחד של מודלים מאומן למציאותו מפורש אופקוני-קייט הפילוא-של הדאטקה הנთוק-ובעזרת הפליגול-יצחודגמאות חדשות (על ידי דגימה מההתפלגות שנלמדה). סוג שני של מודלים-איינו עוסקים בשערור הפליגול-הdataset המקורי-אללא��סוגל ליצחודגמאות חדשות-בדרכיהם אחריות-בפרק זה נדפס במודלים הפופולריים בתחום = VAE, -GANs, והשני של המודלים הגנרטיביים-

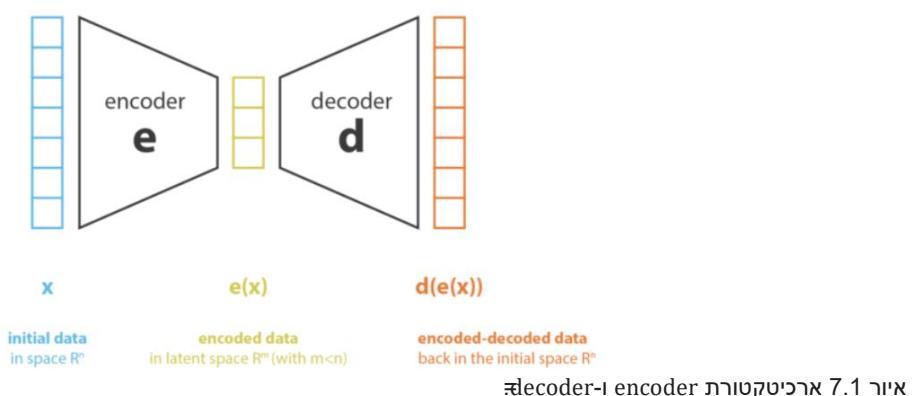
7.1 Variational AutoEncoder (VAE)

7.1.1 Dimensionality Reduction

במקרים רבים, הדטרקטור רצים לנתח הוא בעל ממד=גובה, כולם=Lכל דגימה יש מספר רב של מאפיינים (features). לרבות=Lלא כל המאפיינים ממשמעותיים באוטה מידת. לדוגמה=Lמחיר מניה של חברה מסוימת מושפע ממספר רב של גורמים, אך ככל הנראה גובה הכנסות של החברה משפיע על מחיר המניה הרבה יותר מאשר הגיל המצווע של העובדים=Dוגמא=Nוסף=Lבמשמעותו גיל של אדם על פותמונת ההפנים שלו, לא כל הפיקוליף בתמונה הפנים יהיו בעלי חשיבות לצורך החיזוי. כן שקשה לנתח דатаה מממד=גובה ולבנות מודלים עכוב דатаה צהוב=Bמקרים רבים=מנסים להויר את הממד=Lשל הדאטה תור איבוד=Mידע=Mינימל=Lעד כמה שניתן. בטליר הורדת הממד=מנסים לקליל Y'צג חדש של הדאטה בערך מממד=יותר נמור, כאשר הי'צג זהה מרכיב מהמאפיינים היכי ממשמעותיים של הדאטה. יש מגוון שיטות להורדת הממד כאשר הרעיון המשותף לכך הוא ליצג את הדאטה קומפקט מור יותר. בו אמורים לדידי בטוי רק המאפיינים המשמעותיים של הדאטה.

תהליך האימון הוא דו-שלבי= encoder - decoder =שמטרה להפוך מהדатаה את הייצוג הlatentי של x ($x \in \mathbb{R}^m$, כאשר $m < n$). לאחר מכן התוצאה מוכננת ל- decoder לשחזר את הדטה המקורי, ולבסוף מתקובלוקטור= decoder ($x \in \mathbb{R}^n$) ($d = e(x)$). אם מתקיימת השוויון $d = e(e(x))$ אז d מושך לאבד שום מידע בתהילך, אך אם לעומת זאת $d \neq e(e(x))$ לא אבד שום מידע מוצאים אבד עקב הורדת המmediaela היה ניתק לשחזר אותו במלואו בפונקציית באופן אינטואיטיבי= decoder - encoder - decoder את הדטה המקורי צורה של במא

```
הנמור בדיק טוב מספיק, נראה שהיא ייצוג הlatentי הצליח להפוך את המאפיינים המשמעותיים של הדטה המקורי.
```



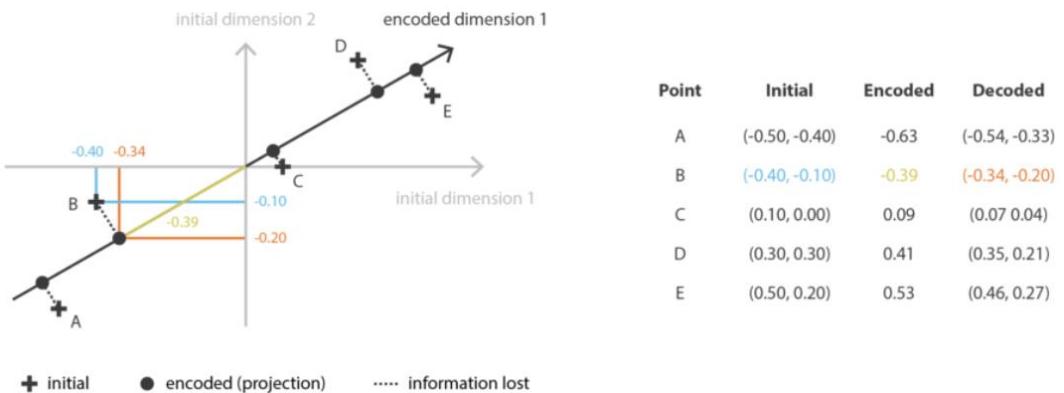
113

כאמורְהמטרה העיקרית של השיטות להורד מידע הינה לקבל ייצוג לטנסי אינטלי עד כמה שניתן. הדרך לעשות זאת היא ~~אלמִקָּאת זוגה~~-encoder-decoder, שמאפשר מיפוי מוקסימלי מידע בערך ידוע למינימום=אנו שגיאת שחזור בעיטה פעונה. אם נסמן בהתאם-~~encoder-decoder~~ הזוגות ~~encoder-decoder~~ האפשרי מינימלית לנוסח את בעית הורדת הממד באופן הבא:

$$(e^*, d^*) = \arg \min_{(e, d) \in E \times D} \epsilon(x, d(e(x)))$$

$\epsilon(x, d(e(x)))$ הוא שגיאת השחזור שבין הדטה המקורי לבין הדטה המשוחזר

אחד השימושות העיקריים להורד מידע שאפשר להסכך עליה בצורה זו היא PCA (Principal Components Analysis). בשיטה זו מטילים (בצורה לינארית) דטה מממד $n < m$ שמאפיינים של הייצוג הלטנטי של הדגומות המקוריות היו אורותוגונליים. תהליך זה נקרא ~~feature decorrelation~~-המתפרק לשלהיא לסייע את המרחק האוקלידי בין הדטה המקורי לדטה המשוחזר, בצורה לינארית גם כן, מהו ייצוג החדש במרחב \mathbb{R}^m

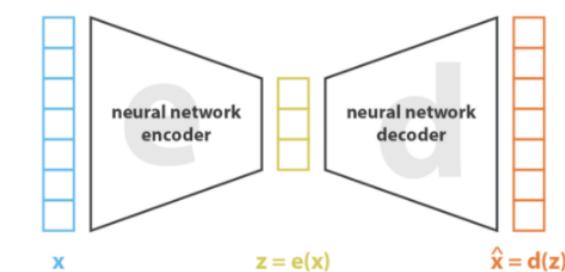


איור 7.2 דוגמא להורדת ממד בשיטת PCA

במנוחים ~~encoder-decoder~~, ניתן להראות כי אלגוריתם PCA ממחפש את הזוג ~~encoder-decoder~~ שני תנאים: ~~encoder~~-ה-~~decoder~~ מבצע טרנספורמציה לינארית על הדטה כך שהמאפיינים החדשים (בממד נמוך) של הדטה יהיו אורותוגונליים. ב. ~~encoder~~-~~decoder~~ הלינארי המתאים יביא לשגיאה מינימלית במונחים של מרחק אוקלידי בז' הדטה המקורי לבין זה המשוחזר מהו ייצוג החדש. ניתן להוכיח שה-~~encoder-decoder~~ האופטימי מכיל את הווקטור ~~העצמיים~~ של מטריצת covariance של ~~design~~-~~decoder~~, וה-~~encoder~~ הוא השחלוף של ~~the~~-~~decoder~~

7.1.2 Autoencoders (AE)

ניתן לקחת את המבנה של ~~the~~-encoder-decoder המtauור בפרק הקודם ולהשתמש ברשת נירונים עבור בנייה: Autoencoder. מבנה זה נקרא ~~the~~-~~encoder-decoder~~ הינו ייצוג חדש ועובר השחזור.



$$\text{loss} = \|x - \hat{x}\|^2 = \|x - d(z)\|^2 = \|x - d(e(x))\|^2$$

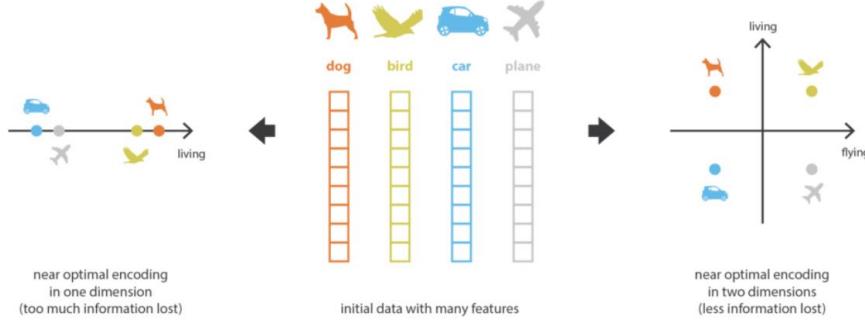
איור 7.3 – שימוש ברשתות נירונים עבור הורדת הממד והשחזור

באופן זהה, האריכתukturת=~~לדטה~~=~~יצרת~~=~~בדוק~~=~~מיצד~~=~~informtion bottleneck~~=~~שברך~~ המאפיינים החשובים של הדטה שבעמצעות פנימית לשליחת אוטבקד וקטוב: ישמש כל יצוא במרקבה הטנטיבק ~~activation functions~~ הפעונטיבוב כל רשות ישדרק שכבת חבוייה אחוריה של אמת שמשה בפונקציית הפעולה(~~functions~~) לא לינאריות; נוottle encoder-~~decoder~~ ה-~~autoencoder~~ יאפשר טרנספורמציה לינארית שבדטה שבעמצעות נוottle

ב>Show Dimensionality Reduction, בדוקו PCA. גורש ה-PCA מושפע מהטבלה שמשתמשה ב-PCA. המאפיינים מהמקורי צווגים כטבלה נמוכה מופקע על ידי כלאי היבנה הכרח זהה ל-PCA.PCA מושפע מ-PCA. המאפיינים החדשניים (לאחר הורדת מדד) עשויים לצטט לא אורתוגונליים (היררכיה שונה מ-0).

icut nich sherashtot shel decoder we encoder han rashtot umokot v meshutashot befonkzot hafulal la inarot. b'mikra zeh, kcl she arctik torah shel rashtot morchbet yoter, kfrashah encoder-decoder yot-hamidatot. kol habamzut hadercoder shel b'zur shazor la-akel-ayibod midu. baofetiaioti, am la-encoder-wol-decoder mespix dergot hofesh (l'mash mspik shabot brashot nivronim) -nitin l'hafchi-hemad shel cl datha l'hod-mad=la-akel ayibod midu. um zat, hafchot hemad=drastitit schatzulol-kalgorom ledatha meshowzer la-abd at hamevorchot. lkn yachshivot gdola b'bachirat mspur hemad=shel merhab hlatnui, kr shatzach ach anken ytbazu niyvi sh-kmafinim=pchot meshumotim v matz shni hmidu udin yhia beul meshumot leshimotkom hoshenot=cidi l'hachish at matava leil-nikhd l'dogma=umract shmekhlatah tmonot shel cab, cifor, mconit v matos v mnasa l'mazoa at permatrim haikrim hmbchimim bigam:

המבחנים בינם:



איור 7.4 דוגמא לשימוש ב-Autoencoder

לפריטים אלו יש הרבה מאפיינים, וקשה לבנות מודל שمبוחן בינויהם על סמך כל המאפיינים. רשות ניירונים מורכبة מספיק אפשרות לבנות "ցוג של כל הדוגמאות על קו ישר", כך שיכל שפרט מסוים נמצא יותר ימיןה, אך הוא יותר "ח'". באופן זהה אמם מתקובל "ցוג חד-ממדית", אבל הוא גורם לאיבוד המבונוקהסמנטי של הדוגמאות ולא באמת ניתק להבון את ה הפרדה ביןיהן. לעומת זאת ניתן להוריד את הממד של תമונות אלכלדמוץ ולהתיחס רק לפרמטרים "ח'" ו"ע'", וכך לקבל הבחנה יותר ברורה בין הדוגמאות-כਮובן שהפרדה זו היא הרבה יותר פשוטה מאשר הסתכלוות על כל הפרמטרים-(הפיקסלים)=של הדוגמאות=דוגמא=זו מראה את החשיבות שיש בבחירה הממדים של ה-encoder.

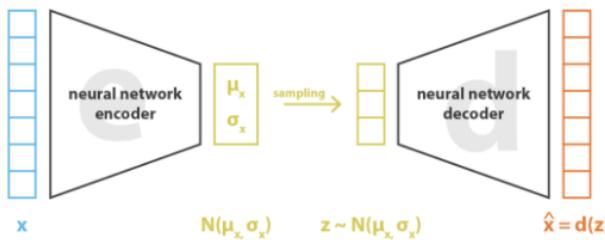
7.1.3 Variational AutoEncoders (VAE)

ניתן לקחת את ה-AE ולהפוך אותו למודל גנרטיבי, כלומר מודל שמסוגל לייצר בעצמו דוגמאות חדשות שאכן מתפלגות כמו הפלוג של הדטה המקורי. אם מדובר בדומין ישן תמנונות למשל, אז נרצה שהמודול יהיה מסוגל לייצר תמנונות שנראות אוטנטניות ביחס לדאטאנליאו $\text{AE}^{\text{מואפקט}}$ את הדטה במודוןמור, שוליך בחשבון את המאפייניות העיקריות, ולאחר מכן משחזר את התוצאה למודוןמור. אולם, מנגנון זה אינו מאפשר להשפייע על האופן בדף הדאטאנליאו במרחב הלטנטי. אם יוגרל וקטור כלשהו מהמרחב הלטנטי קרוב לוודאי שהוא לא יהווה "צ'אשdon" לדטה המקורי – אם היינו מכנים אותו decoder , סביר להניח שתתוצאה לא תהיה דומה בכלל לדטה המקורי – במקרה של AE $\text{AE}^{\text{אפקט}}$ לאויס – שתתמנונות של כליבים יתוגמם באקרראפוקטו – מהמרחב הלטנטי פועלן, הסיכופלקבל תמנות כלב כלשהו לאחר השחזר של ה- decoder הינו אפסי.

כדי להתמודד עם בעיה זו, ניתן להשתמש ב-Variational AutoEncoder (VAE). בשונה מ-VAE, ShallowAE משלב תופעות נורמליות עם תוחלת 0 ו- $\text{covariance} = \mathbb{I}$. בהינתן התפלגות $p_{\text{encoder}}(z|x)$, מתייחסים למשתנה z כRANDOM VARIABLE. מטריצת covariance של התפלגות $p_{\text{encoder}}(z|x)$ מוגדרת כ- Σ_z , ומטריצה פוטוריינית כ- Φ_z . מטריצה פוטוריינית מוגדרת כ- $\Phi_z = \Sigma_z^{-1/2}$. לאחר מכן דוגמיה נספחים ל- Φ_z ו- μ_z ומשתנה z מוגדר כ- $z = \mu_z + \Sigma_z^{1/2} \cdot \epsilon$, כאשר ϵ נספחים ל- Φ_z ו- μ_z מוגדר כ- $\mu_z = \Phi_z \cdot \mu_{\text{encoder}}$. מטריצת covariance של התפלגות $p_{\text{decoder}}(x|z)$ מוגדרת כ- Σ_x , ומטריצה פוטוריינית כ- Φ_x . מטריצה פוטוריינית מוגדרת כ- $\Phi_x = \Sigma_x^{-1/2}$. מטריצת covariance של התפלגות $p_{\text{decoder}}(x|z)$ מוגדרת כ- $\Sigma_x = \Phi_x \cdot \Sigma_z \cdot \Phi_x^T$.

בapon זהה, הלמידה דואגת לא רק להורדת המזד-של הדעתה, אלא גם להתפלגות המושרית על המרחב הלטני-כasher התפלגות המותנית במצוֹעַז אֶת-ובה, קרי קרוביה להתפלגות המקורית של-א, ניתן בעזרתה גם ליצוף דוגמאות חדשות, ובעצם מתќבל מודל גנרטיבי.

כאמור, ה-encoder פונקציונאל נספה ליציג את הדadata המקורי באמצעות התפלגות במנגד נמור יותר, למשל התפלגות נורמלית עם תוחלת ומטריצת covariance $\sigma_x = \mu_x = N(x|z-p)$. חשוב לשים לב להבדל בתפקיד של ה-decoder לעומת AE הוא ועוד לתהיליך האימון בלבד ובפועל מה שחשוב זה הייצוג הלטנטיבי, ב-VAE decoder חשוב לא פחות מאשר הייצוג הלטנטיבי, כיוון שהוא שימוש ליצירת DATA חדשה לאחר תהיליך האימון, או במילים אחרות, הוא הופך את המערכת למודול גנרטיבי.



איור 7.5 ארכיטקטורה של EA

ונאר באופן פורמלי את בעית האופטימיזציה $\hat{\theta}$ מנוסה לפטור=נסמן את הווקטורים של המרחב הlatent ב- Z ; אוסף הפרמטרים של ה- decoder , ואות הפרמטרים של ה- encoder . כדי למצוא את הפרמטרים האופטימליים של שתי הרשותות, נרצה להביא למקסימום אופ(θ) = $\mathcal{L}(X, \hat{X})$, כלומר למקסם את הנראות המרבית שבסט האימוקחות θ. כיוון שפונקציית \log מונוטונית. יוכל לקחת את לוג ההסתברות:

$$L(\theta) = \log p(x; \theta)$$

אם נביא למקסימום את הביטוי הזה, נקבל את $\hat{\theta} = \text{האופטימלי}$. כיוון שלא ניתן לחשב במפורש את $\hat{\theta}(x)$, יש להשתמש בקירוב. נניח והפלט של ה-`decoder` הוא בעל התפלגות $(\lambda; x|z)$ (מה ההסתברות לקבל אפקבה נתן x בנכינה), וננסה ליצא אתהתפלגות הזו באמצעות רשות נוירונים עם סט פרמטרים λ .Cutת ניתן לחלק ולהכפיל את $\hat{\theta}(x)$ על מנת לקבל אפקבה.

$$\log p(x; \theta) = \log \sum_z p(x, z; \theta) = \log \sum_z q(z; \lambda) \frac{p(x, z; \theta)}{q(z; \lambda)} \geq \sum_z q(z; \lambda) \log \frac{p(x, z_i; \theta)}{q(z; \lambda)}$$

כאשר אי השוויון האחרון נובע=**mai-sho'iin yonot**, והביטוי שמיינן לאי השוויון נקרא Lower BOund ($ELBO(\theta)$). ניתן להזכיר שהפרש בין ה-ELBO לבין הערך שלפני הקירוב הוא המרחק בין שתי ההתפלגויות: $\mathcal{D}_{KL}(p(z|x), q(z))$, שנקרא Kullback-Leibler divergence ומוסומן ב-

$$\log p(x; \theta) = ELBO(\theta, \lambda) + \mathcal{D}_{KL}(q(z; \lambda) \| p(z|x; \theta))$$

אםשתיהתפלגיותזהות, אז מרחוק \hat{L}_{KL} בינהן הוא $=$ ומתקיים $=$ $ELBO(\theta, \lambda) = p \log(x; \theta)$. זכור, אנחנו מփשים למקסם את פונקציית המחדיף (θ, λ) , וcut בעזרת הקירוב ניתן לרשום:

$$L(\theta) = \log p(x; \theta) \geq ELBO(\theta, \lambda)$$

$$\rightarrow \theta_{ML} = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} \max_{\lambda} ELBO(\theta, \lambda)$$

cut ניתן בעזרת שיטת **Gradient Descent** (GD) על מנת למצוא את האופטימום של הביטוי, וממנו להפיק את הפרמטרים האופטימליים של encoder וה-decoder. נפתח יותר את ה- $\text{ELBO}(\theta, \lambda)$ עבור VAE, ביחס לשתי התפלגיות:

לפי הגדרה $p(x|z; \theta)$ – ההסתברות ש- x יופיע סט פרמטרים θ יוציא z בהינתן x
 $q(z|x; \lambda)$ – ההסתברות ש- z יופיע סט פרמטרים λ יוציא x בהינתן z בנסיבות

$$ELBO(\theta, \lambda) = \sum_z q(z|x; \lambda) \log p(x, z; \theta) - \sum_z q(z|x; \lambda) \log q(z|x; \lambda)$$

את הביטוי $p(x, z) = p(x|z) \cdot p(z)$ ניתן לפתח לפי חוק ביאו $\log p(x, z; \theta)$

$$\begin{aligned}
&= \sum_z q(z|x; \lambda) (\log p(x|z; \theta) + \log p(z; \theta)) - \sum_z q(z|x; \lambda) \log q(z|x; \lambda) \\
&= \sum_z q(z|x; \lambda) \log p(x|z; \theta) - \sum_z q(z|x; \lambda) (\log q(z|x; \lambda) - \log p(z; \theta)) \\
&= \sum_z q(z|x; \lambda) \log p(x|z; \theta) - \sum_z q(z|x; \lambda) \frac{\log q(z|x; \lambda)}{\log p(z; \theta)}
\end{aligned}$$

הביטוי השני לפि הגדרה שווה ל- $(\theta; z) \parallel p(z)$, שכן מתקובל

$$= \sum_z q(z|x; \lambda) \log p(x|z; \theta) - \mathcal{D}_{KL}(q(z|x; \lambda) \| p(z))$$

הביטוי הראשון הוא בדיקת התוחלת ש \hat{z} מתפלג נורמלית, ניתן לרשום:

$$= \mathbb{E}_{q(z|x; \lambda)} \log N(x; \mu_\theta(z), \sigma_\theta(z)) - \mathcal{D}_{KL}(N(\mu_\lambda(x), \sigma_\lambda(x)) \| N(0, I))$$

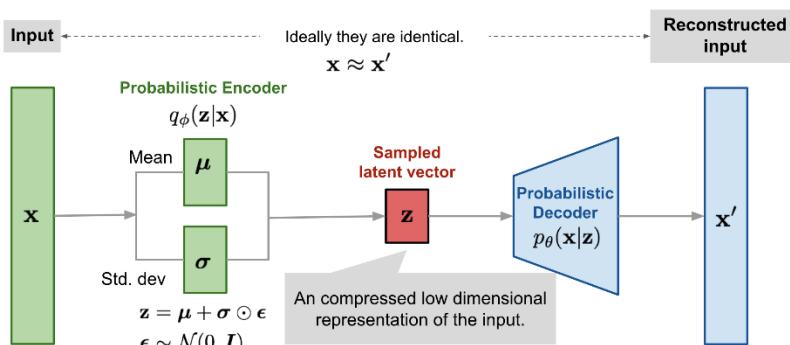
כדי לחשב את התוחלת ניתן פשוט לציג דוגמאות מההתפלגות $(x \sim N(\mu_\theta(x), \sigma_\theta(x)^2))$ ולקבל:

$$\mathbb{E}_{q(z|x; \lambda)} \log N(x; \mu_\theta(z), \sigma_\theta(z)) \approx \log N(x; \mu_\theta(z), \sigma_\theta(z))$$

ועבר הביטוי השני יש נוסחה סגורה

$$\mathcal{D}_{KL}(N(\mu, \sigma^2) \| N(0, I)) = \frac{1}{2}(\mu^2 + \sigma^2 - \log \sigma^2)$$

icut משיש בידינו נסוחה לחישוב פונקציית המחר, נוכל לבצע את תהליך הלמידה. יש לשים לב שפונקציית המחיה המקורית הייתה רק ב-θ, אך באופן שפיתחנו אותה היא למעשה דואגת גם למצורר ההפרש בין הכנסיס-ה-*encoder* לבין המוצב-*של*, וגם למצורר המרחק בין התרגולות הפרויורי-*של* לבין התרגולות-*ז'* שבמוצב-*ה-VAE*.



$$x_t \rightarrow \mu_\lambda(x_t), \Sigma_\lambda(x_t) \rightarrow z_t \sim \mathcal{N}(\mu_\lambda(x_t), \Sigma_\lambda(x_t)) \rightarrow \mu_\theta(z_t), \Sigma_\theta(z_t)$$

$$\text{ELBO} = \sum_t \log \mathcal{N}(x_t; \mu_\theta(z_t), \Sigma_\theta(z_t)) - \mathcal{D}_{KL}(\mathcal{N}(\mu_\lambda(x_t), \Sigma_\lambda(x_t)) || \mathcal{N}(0, \mathbb{I}))$$

איור 7.6 תהליכי הלמידה של VAE.

אשר נתן t -dgm אוניברסלי, ניתן להעביר כל-dgm אוניברסלי t -encoder ו- t -decoder לעבורה אוניברסלית. לאחר מכן וקטור לטנטוץ מההתפלגות עם פרמטרי-פאל, מעבירים אותו ב- t -decoder ומתקבלים אוניברסליים. לאחר התהיליך ניתן להציג את הפרמטרים המתוקבים ב- t -ELBO ולחשב את ערך פונקציית המchiaר. ניתן לשימוש בשה- t -ELBO מרכיב משני=Aיבר יסוד= t -האיבר הראשוני מושער את הדמיוקבון הדגם אוניברסלי לבני התפלגות שמתකבלת במוואצ, והאיבר השני מבצע רגוליזציה להתפלגות הפרוריות במרחב הלטנטי. הרגוליזציה גורמת לכך שההתפלגות במרחב הלטנטי מתחיה קרוביה עד כמה שנייתן להתפלגות הפרורית. אם ההתפלגות במרחב הלטנטי קרוביה להתפלגות הפרורית, אז ניתן בעזרת t -decoder ליצור דוגמאות חדשות, ובמונע זהה ה-VAE הוא מודול גנרטיבי.

הדגימה ש**zmanif** מראה הפלגות מרוחקות הלטנטי יוצרת קושי בחישוב הגרדיאנט של ELBO, אך פדרך כל ממצאים Reparameterization trick $\nabla_{\theta} \log p_{\theta}(x) = \nabla_{\theta} \log p_{\theta}(z) + \nabla_z \log p_{\theta}(x|z)$ מוגדר באמצעות נורמלית טנדרטית, ועוד כדי לקבל אופערר הדגימה ש**zmanif** משתמשים בפרמטרים של ה-encoder (x, μ, σ) בוגישה הזו כל התהילה נהיה דטרמיניסטי=
מגראיליף z מראש ואז רק נשאר לחשב באופן סכמטי את התפשטות הערך בראשת

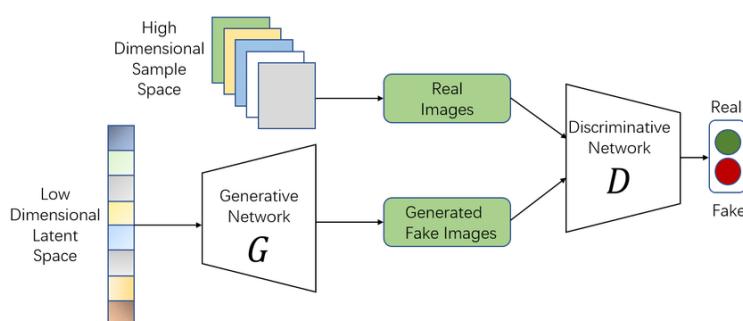
7.2 Generative Adversarial Networks (GANs)

גישה-אחרת של מודל גנרטיבי-ניראתי=Generative Adversarial Networks (GANs), ובשוונה מ-VAE בגישה זו לא-מנסים לשערף התפלגות של DATA בצורה מפורשת (על ידי מציאת הפרמטרים הממקסימים את הנראות המריבית של סט האימון)=אליה יוצרת-דעת-באופן אחר. הרעיון הוא לא-מן שתי רשותות במקביל=רשota אחוף שלומדת ליצירת דוגמאות, ורשota שנייה שלומדת להבחין ביצירות-אמיתית-בפסוט האימוקלבן תמון-coresו נתקשה-שנוצרה על ידי הרשותה-הראשונה=הרשותה הראשונה מאומנת ליצור דוגמאות שיגרמו לרשותה השנייה לחשב שהן אמיתיות, בזמן שהמטרה של הרשותה השנייה היא-אל לא לחשוף לרשותה הראשונה לבלבן אותה=באופן היזהו-ראשונה מהו זה למעשה מודל גנרטיבי, לאחר מכן שלב האימון היא מסוגלת ליצור דוגמאות-אמיתיות-של-לא ניתן להבין בינו לבקדרות-אמיתיות

7.2.1 Generator and Discriminator

בפרק זה נסביר את המבנה של ה-GAN=הקלואס=שהומצא בשנת 2014 על ידי Goodfellow et al. נזכיר כי קיימים מאות רבות של וריאנטים שונים של GAN שהוצעו מאז, ועוד יתחום זה פועל מאוד מבחינה מחריפה.

GAN מבוסס על שכבת אלמנטי ϕ מרץ- ϕ =-רשות שיזכרת-דאטה(generator) ורשות שמכריע-האם הדאטה הזרה-ינטנסיבית או אמיתית(discriminator)=כאשף-האימון געשה על שתי הרשותות ייחודה-discriminator מקבל כקלטuka ה-אפורח-הוגמאו-האמיתיות והן את הפלט של ה-generator=Cד-למוד-להבחן בקדאות-אמתית לבודאות-ינטנסיבית. ה-generator מיציר דוגמאות ומתקבל פידבק מה-discriminator=Cד-iscrimינטור ליצירת-הוגמאו-הנראות אמיתיות=Nסומך את ה-generator ב-d-G ואות ה-discriminator ב-D, ונתקבל את הסכמה הבאה:



איור 7.7 ארכיטקטורת GAN

discriminator הוא מושג שחלקו ההפוך הוא הסתברות שהקלט הינdeg;מאמית;ונסמן $\text{discriminator}(x)$.
 אורה הסתברותה זו כדי לאמן את discriminator. נרצה להציג שני דברים: א. למקום $\text{discriminator}(x)$ עבור אמצעי האימוץ קלומר לטעות כמה שפחות בזיהוי DATA מזענאות (x) . ב. עבור DATA סינתטי קלומר לזהות נכוון כמה שיטות לדוגמאות סינתטיות שייצרו על ידי generator. באופן דומה נרצה לאמן את discriminator כשהדגימות שהוא מייצרת תהינה כמה שיטור דומות לדוגמאות אמיתיות, קלומר ה-generator מעוניין לגרום לכך discriminator להוציא ערכים כמה שיטור גבויים עבור הדatta הסינטטי שהוא מייצרת בשביל לאמן יחד את שפה חלקי המודול. נבנה פונקציית מחיר בעלית שני איברים. באפשר הבא:

$$V(D, G) = \min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{x \sim Data} \log D(x) + \mathbb{E}_{z \sim Noise} \log (1 - D(G(z)))$$

נסביר את הביטוי המתkeletal discriminator שמשמעותו $\text{discriminator}(x)$ יהיה כמו שיפור קרוב ל-1 $\text{discriminator}(G(z))$. יהיה כמה שיפור קרוב ל-0. ה-generator יספק לעומת discriminator זהה להביא למינימום את פונקציית המחיר, כך ש- $\text{discriminator}(G(z))$ יהיה כמה שיפור קרוב ל-1, כלומר ה-generator יחשוב $\text{discriminator}(G(z))$ הוא דאטץ אמיתי.

כעת האימון נעשה באופן איטרטיבי, כאשר פעם אחת מקבעים או G ומאמנים אותו, ופעם אחרת מקבעים אותו. אם מקבעים אותו, אז למעשה מאנים מסווג בינהר, כאשר מ Chapman את האופטימיזציה בוקטו הפרמטרי ϕ_d :

$$\max_{\phi_d} \mathbb{E}_{x \sim Data} \log D_{\phi_d}(x) + \mathbb{E}_{z \sim Noise} \log \left(1 - D_{\phi_d}(G_{\theta_g}(z)) \right)$$

אפלומת זטרומקסים אותו, אזי ניתן להעתלם מהאיבר הראשון כיון שהוא פונקציה של ϕ_d בלבד וקבוע ביחס ל- θ_g . לכן שאר רק לבדוק את הביטוי השני, שמחפש את ה-generator שמייצג את דאטץ שנראה אמיתית בצורה הטובה ביותר:

$$\min_{\theta_g} \mathbb{E}_{z \sim Noise} \log \left(1 - D_{\phi_d}(G_{\theta_g}(z)) \right)$$

כאמור המטריה יאלמן אותו בעזרת GRADIENT (במצבו הנוכחי) כדי שיהיה מסוגל ליצור דוגמאות הנראות אוטנטיות. האימון של ה-generator מושג באמצעות GRADIENT DESCENT (מżywot פונקציית המחיר ביחס ל- θ_g) והוא אימון של ה-discriminator מושג באמצעות GRADIENT ASCENT (מaksymum פונקציית המחיר ביחס ל- D). האימון מתבצע במשר Epochs, כאשר כאמור מאנים ליטרואן אותו $D = \text{batch_size} \cdot \text{batch_size}$, מספר מסויים של x_m, \dots, x_1 מגדימות של רעש (z_m, \dots, z_1) , ומכוונים את הקלט ל- G הגדיאנט של פונקציית המחיר לפि הפרמטרים של generator במהלך האימון מחושב באופן הבא:

$$\nabla_{\theta} V(G_{\theta}, D_{\phi}) = \frac{1}{m} \nabla_{\theta} \sum_{i=1}^m \log \left(1 - D_{\phi}(G_{\theta}(z_i)) \right)$$

וכאשר מאנים את ה-discriminator, הגרדיינט נראה כך

$$\nabla_{\phi} V(G_{\theta}, D_{\phi}) = \frac{1}{m} \nabla_{\phi} \sum_{i=1}^m \log D_{\phi}(x_i) + \log \left(1 - D_{\phi}(G_{\theta}(z_i)) \right)$$

נוהג לבצע מודיפיקציה קטנה על פונקציית המטריה של generator. כיון שבהתחל הדגימות מיצרות על ידי generator דומות לחילופין לא מושג מושג discriminator. האימון ה- D בקצב מוגבל-0. עניין $\text{discriminator}(G(z))$ מקבל ערכים מאד קרובים ל-0, ומילא גם הביטוי $\log(1 - D(G(z)))$. זה גורם לכך שהgradient של generator ייה מאד קטן, ולכן כמעט ולא מתרחש שיפור ב-generator. לכן במקומות לחפש מינימום שלו $= \text{generator}$ מינימום מינימום לביטוי $\mathbb{E}_{z \sim Noise} \log(1 - D(G(z)))$. הביטויים לא שווים לגמרי א-שניהם מובילים לאותר פרטור של בעיית האופטימיזציה אותה הם מייצגים, והביטוי החדש עובד יותר טוב נומריית ומצליח לשפר את generator בצורה עילית יותר.

הערכים האופטימליים של D ו- G :

כזכור, פונקציית המחיר הינה

$$V(D, G) = \min_G \max_D \mathbb{E}_{x \sim Data} \log D(x) + \mathbb{E}_{z \sim Noise} \log \left(1 - D(G(z)) \right)$$

כעה נרצה לחשב מה הערך האופטימי של generator discriminator שיעבורו לחשב את הערך של פונקציית המחיר. לשם הנוחו נסמן את התפלגות הדאטה האמיתית ב- p_g ואת התפלגות הדאטה סינטטיות המיצרת על ידי generator G . עבור G קבוע, ניתן לרשום את פונקציית המחיר כ-

$$V(D, G) = \int_x p_r(x) \log D(x) + p_g(x) \log(1 - D(x)) dx$$

כדי להביא את הביטוי זהה למקסימום, נרצה למקסם את $\text{האינטגר}=עובי-כל-ערך-האפשרי$ מיל'ן הפונקציה לה מעוניינים למצוא אופטימום הינה:

$$f(D(x)) = p_r(x) \log D(x) + p_g(x) \log(1 - D(x))$$

נזכור את הביטוי האחרון ונשווה ל-0 כדי למצוא את הערך האופטימלי $D(x)$ עובי x נתון:

$$\frac{\partial f(D(x))}{\partial D(x)} = \frac{p_r(x)}{D(x)} - \frac{p_g(x)}{1 - D(x)} = 0$$

$$\rightarrow p_r(x)(1 - D(x)) - p_g(x)D(x) = 0$$

$$D(x)_{opt} = \frac{p_r(x)}{p_r(x) + p_g(x)}$$

הביטוי שהתקבל הינו הערך האופטימלי של discriminator -generator בעובי discriminator (ביחס לקלט=אנטו)=נשים לב שעבור המקרה בפה-AN=G מצליח לייצר דוגמאות שנראות אמיתיות לחלווט, כלומר $p_g(x) = p_r(x)$, אך מתקיים $D(x) = \frac{1}{2}$. הסתברות=זו ממשוערת=שה discriminator לא ידע להחליט לגבי הקלט המתקבל, והוא קובע שההסתברות שהקלט אמיתי זהה לזו שהקלט סינטטי

כעת נבחן מהו ערך פונקציית המחיר כאשר D אופטימלי:

$$\begin{aligned} V(G, D) &= \mathbb{E}_{x \sim Data} \log D(x) + \mathbb{E}_{z \sim Noise} \log(1 - D(G(z))) \\ &= \mathbb{E}_{x \sim Data} \log \left(\frac{p_r(x)}{p_r(x) + p_g(x)} \right) + \mathbb{E}_{z \sim Noise} \log \left(1 - \left(\frac{p_r(x)}{p_r(x) + p_g(x)} \right) \right) \\ &= \mathbb{E}_{x \sim Data} \log \left(\frac{p_r(x)}{p_r(x) + p_g(x)} \right) + \mathbb{E}_{z \sim Noise} \log \left(\frac{p_g(x)}{p_r(x) + p_g(x)} \right) \\ &= \mathbb{E}_{x \sim Data} \log \left(\frac{p_r(x)}{\frac{(p_r(x) + p_g(x))}{2}} \right) + \mathbb{E}_{z \sim Noise} \log \left(\frac{p_g(x)}{\frac{(p_r(x) + p_g(x))}{2}} \right) - \log 4 \end{aligned}$$

הביטוי $\text{המתקבלה-הינקה-המרחיק-בין}$ $\text{התפלגיות-}p_g, p_r$, והוא נקרא \mathcal{D}_{JS} =Jensen-Shannon divergence ומוסומן ב- \mathcal{D}_{JS} מרחוק זיהינ-גראסה סימטרית ש- \mathcal{D}_{KL} =Kullback–Leibler divergence, ובעובי שתהתפלגיות Q , הוא מוגדר באופן הבא:

$$\mathcal{D}_{JS} = \frac{1}{2} \mathcal{D}_{KL}(P||M) + \frac{1}{2} \mathcal{D}_{KL}(Q||M), M = \frac{1}{2}(P + Q)$$

קיילמו שעבור D אופטימלי, פונקציית המחיר שווה למרחק \mathcal{D}_{JS} בין p_g עד כדי קבוע, ובאופן מפורש:

$$V(G, D_{opt}) = \mathcal{D}_{JS}(p_r, p_g) - \log 4$$

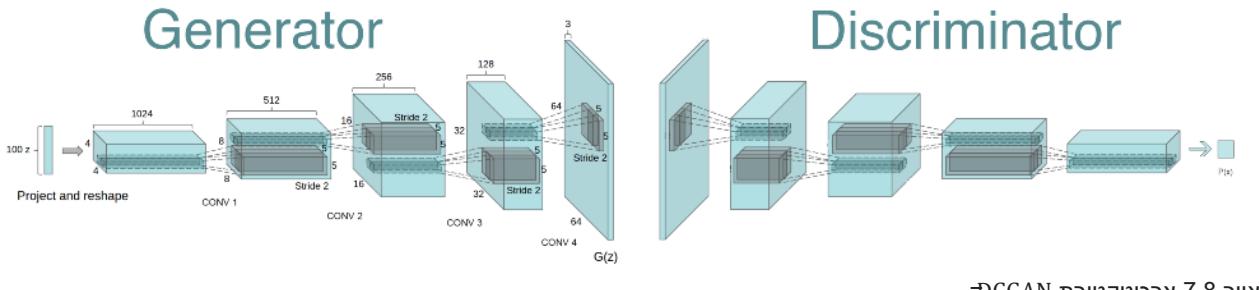
כאשר $\text{G}=אופטימלי$ ומתקיים $p_g(x) = p_r(x)$ אז המרחק בין התפלגיות שווה 0, כלומר $\mathcal{D}_{JS}(p_r, p_g) = 0$ וילך מתקבָּל

$$V(G_{opt}, D_{opt}) = -\log 4$$

יש משמעות גודלה לביוטי שהתקבל – ככל שנצליח למצער יותר את $\mathcal{D}_{JS}(p_r, p_g)$, כך נצליח לקבל AN=G יותר טוב

7.2.2 Deep Convolutional GAN (DCGAN)

כפי שהוסבר בפרק 5, רשותות קובולוצי-עלויות יותר בדומיין של תמונות מסוימות-**GAN**. אך ניתן לזכור רשותות קובולוצי-ולබנות בעזרת **generator-discriminator**-generator-discriminator. generator מקבל וקטור אקראי ומעביר אותו דרך רשת קובולוציה על מנת ליצור תמונה, וה-discriminator מקבל תמונה ומעביר אותו דרך רשת קובולוציה שعواשה סיווג ביןאי אם התמונה אמיתי או סינטטי CGAN. CGAN הומצא ב-2015 ומאז פותחו רשותות שמיצירות תמונות יותר איותיות הן מבחינת הרוחולציה והן מבחינת הדמיון שלהן לתמונות אמיתיות, אך החשיבות של המאמר נעהча בשימוש ברשותות קובולוציה עבור GAN-ים מייעוד לדומיין של תמונות.



איור 7.8 ארכיטקטורת DCGAN

7.2.3 Conditional GAN (cGAN)

לעתים מודל גנרטיבי נדרש ליצור דוגמא בעלת מאפיין ספציפי ולא רק דוגמא שנראית אונטנית. למשל, עבור אוסף תמונות המציגות את הספרות מ-0 עד 9, ונרצה שה-GAN ייצור תמונה של ספרה מסוימת= $\text{במקירם}=אלו=$ בנוסף לווקטו-הכניתה², ה-GAN=מקבילה לתנאי נוסף על הפלט אותו הוא צריך ליצור לייצר, כמו למשתמש ספרה ספציפית אותה רציף לקביל. GAN כזה נקרא **conditional GAN** (או בקיצורGAN), ופונקציית המחיר שלו דומה מאוד לפונקציית המבחן של GAN רגיל למעט העובדה שהבאים הופכים להיות מותניים

$$\mathcal{L}_c(D, G) = \min_G \max_D \mathbb{E}_{x \sim Data} \log D(x|y) + \mathbb{E}_{z \sim Noise} \log(1 - D(G(z|y)))$$

7.2.4 Pix2Pix

כפי שראינו, ה-**GAN**=הקלואן שתוואר לעיל מסוגל ליצרת דוגמאות חדשות=מוקטור אקריאפ-**Z**=המוגרל מהתפלגות מסוימת (בדרך כלל התפלגות גאומטרית סטנדרטית, א-ז-הלא מוכחה)=ישן גישות נוספת ליצור דטה חדש, כמו למשל יוצר תמונה חדשה על בסיס=**KDD** מתאר כללי=**FSHLA**=ט האימוקבמקרה זה הבנייה=המציאות של תמנונות ווסקיצות שלמה

שיטות Pix2Pix מושתמשת בארכיטקטורה של GAN, אשר במקום לדגם אפקוטו-המרה הפלגות כלשהי, מקבלת סקיצה של תמונה בתווך קלט, והזמנת generator לומד להפוך את הסקיצה לתמונה אמיתית הארכיטקטורה של discriminator. generator ישנו שואף לכך למודל קודם לכך לפרט להתחממה לבניית הקטלט, אך ה discriminator מושתג על ידי generator, הוא מקיבלאן מזג תמונה זו את הסקיצה ואת התמונהorigin (פעמיים). תמונהorigin היא מושתגת על ידי generator על ידי discriminator. discriminator מקבלת את התמונהorigin וריאיציה זו של discriminator מושתגה גם את פונקציית המחיה generator נזקירה ללמידה שני דברי זה גם ליצור תמונה אותובוגרף discriminator שמיימן שהאמתות, וגם למצער את המרחב בין התמונה שנוצרת לבין התמונה אמיתית השיכוף לסקיצה.

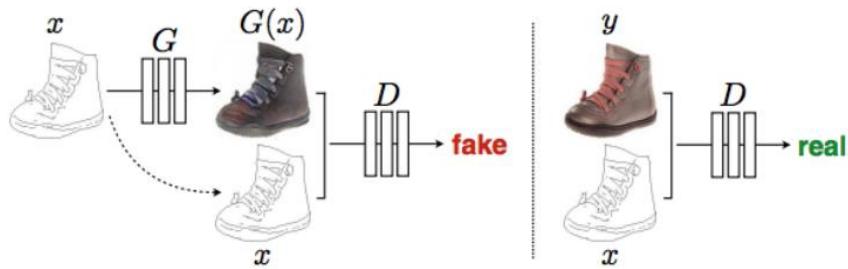
כעומסמן תמונה אמיתית השיכת לסקירה בעזונרגשות את פונקציית המחייבכשי חלקיים נפרדית=
רגיל של GAN מරחיק אוקליידיזבין תמונה המקור לבין הפלט

$$V(D, G) = \min_G \max_D \mathbb{E}_{x,y} \left(\log D(x, y) + \log (1 - D(x, G(x))) \right)$$

$$\mathcal{L}_{L1}(G) = \min_{\theta_g} \mathbb{E}_{x,y} \|G(x) - y\|_1$$

$$\mathcal{L}(G, D) = V(D, G) + \lambda \mathcal{L}_{L1}(G)$$

ניתן להסתמך על pix2pixKBator GAN המאפשר תמורה לתמונה (image-to-image translation). נציין שבמקרה זה הקלט הפלט של pix2pix שייכים לתחומים (domains) שונים (סקייטה ותמונה רגילה).

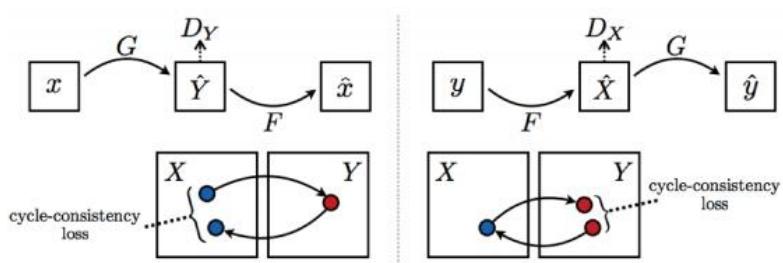


איור 7.9 ארכיטקטורת Image-to-Image Translation - Pix2Pix

7.2.5 CycleGAN

ב- Pix2Pic הדרישה המקורית הייתה \approx -סקירה ואיתה תמונה אמיתית. זוגות של תמונות זה לא דבר כל כך צפוי, ולכשישפו אפתהלהיל'ר האימומך שיהיה ניתן לבצע אותו על שני סטים של נתונים מתחום \mathcal{G} = shonis =הארcticutto\mathcal{G} עבור המשימה זו מרכיב משפטgenerators-בהתחלת מכניםFDGMAmhdomן הראשו\mathcal{G}=shonos להפוך את כל דוגמא מחדומי\mathcal{G} השנוy, והפלט נכנס לgenerator-shonos לשחרר את המקו\mathcal{G}. המוצא של\mathcal{G} נכנס לא רק לgenerator אלא גם לdiscriminator שנותעד לזהו מה אם התמונה שתהתקבלה היאamina תית או לא$\text{(ubov)$ הדומי\mathcal{G} שy)=ניתן לבצע את התהיל'ר הזהה באופן דו-איuboy-מכנים\mathcal{G}ים אפקט\mathcal{G} על מנת לקבל אפקאות המוצא מכנים\mathcal{G}ים ל-discriminator שבדידי לבע ציוגy בינו>> ו-\mathcal{G} על מנת לנסות לשחרר את המקו\mathcal{G}-generator השפה\mathcal{G} נועוד לשפר את התהיל'ר הלמידה-shonos לאחר שshonos הופיע-y, ניתן לקבל חזרה אפקאים נעריר או אפקט\mathcal{G} שמתוך צפיה לקב\mathcal{G} $\approx (x)$ ($\mathcal{G} = \text{cycle-consistency}$) והוא מօיס\mathcal{G}

$$V(D_x, D_y, G, F) = \mathcal{L}_{GAN}(G, D_y, x, y) + \mathcal{L}_{GAN}(F, D_x, x, y) \\ + \lambda (\mathbb{E}_x \|F(G(x)) - x\|_1 + \mathbb{E}_y \|G(F(y)) - y\|_1)$$

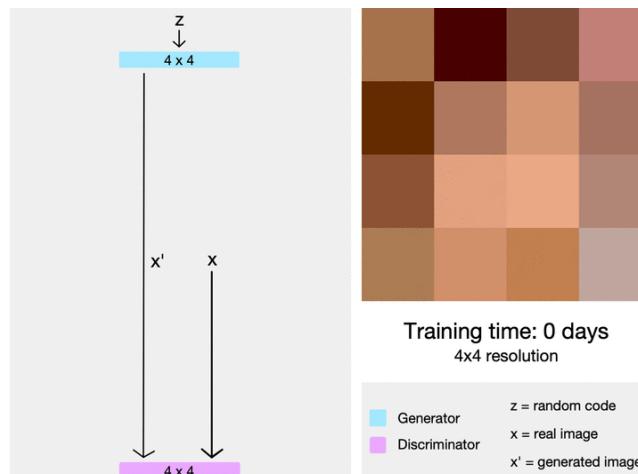


.איור 10.7 ארכיטקטורת CycleGAN

7.2.6 Progressively Growing GAN (ProGAN)

כאמור לעיל, עבור דומין של תמונות, הגיוני להשתמש ברשומות קובולוציה עבור יצירת תמונות חדשות, וזה הרעיון הבסיסי שמאחוריו=DGAN= למורות היכולת המרשימה של=DGAN= בפיירה של תמונה באיכות גבואה, יכולת זו אוטומטית מוגבלת לתמונות בגודל מסוים= \times =כל שהרזולוציה של תמונה גבוהה יותר, כך יותר קל להבחין אם תמונה זו אמיתית או=Nכזירה על ידי רשת גנרטיבית.=בעוד ש-DCGAN=DGAN= מצליח ליצור תמונות שנראות גבוהות יותר, כמו למשל רזולוציות 32×32 , 64×64 , 128×128 , והוא מנסה ליצור תמונות ברזולוציות גבוהות יותר, כמו למשל רזולוציות 256×256 =ProGAN=ба לתמת מענה לכך, והוא הינה ה-GAN הראשון שפרץ את מחסונ הרזולוציה והצליח ליצור תמונות איכותיות מזו=(בمانר המקובל ש=DGAN= עד רזולוציה של 1024×1024 ב- שיהיה ניתן להבחין שתמונות אלה סינטטיות=אמנם עוד לפני ProGAN=DGAN= שהצליחו ליצור תמונה בעלת רזולוציה גבוהה מזו מוגברת אחרת ברזולוציה גבוהה (זוקן \rightarrow זוקן \rightarrow מושימה אחרת, מכיוון שבשבילה צריך רק ללמידה לשנות תוכנות של תמונות קטלט, ולא לייצר תמונה חדשה לגמרי מאפס.

הרעין העיקרי מadvisor-GAN ProGAN, שהוצע ב-2017 על ידי חוקי ממחברת Nvidia; הינולייצר תמונות ברוחולוציה הולכת וגדלה בצורה הדרגתית. ככלMORE; במקומ לנסוט לאמן את כלgenerator השכבות של generator אחות, כפי שנעשה בכלNs-GAN; לפניו קיינית לאמן אותו לייצר תמונות ברוחולוציה משותנה—בהתחלת הוא מתאפשרליצ'ר תמונות ברוחולוציות מאוד נמוכה (4×4), לאחר מכן המשיכו ליצור תמונות ברוחולוציה 8×8 , אחר 16×16 וכך עד יוצרה של תמונה ברוחולוציה של 1024×1024 .



איור 7.11 ארכיטקטורת ProGAN.

כדי לאמן GAN ליצר תמונות בגודל 4×4 , התמונות מוסט האימון הוקטנו לגודל זה (down-sampling). אחרי שהGAN לומד לייצר תמונות בגודל 4×4 , מושגים פָּרָאַמְּנָטִים שבעד שכבה המאפשרת להכפיל את גודל התמונות המיצירות-קְרָבֶּץ ליצור תמונות בגודל 8×8 =לצ"י שהאימון של הרשות-עם השכבה הנוסף-מתחייב עם המשקלים שאומנו קודם לכך לא "מקפאים" אותם, כלומרם מעודכנים גם כן תוך כדי אימון הרשות-בשביל לייצר תמונה ברזולוציה כפולה. הגזלה-הדרגתית של הרזולוציה-המאLASTת את הרשותות להתמקד תחיליה בפרטים ("הגים") של התמונה (דפוסים בתמונה מוטשטשת מאוד)=לאחר מכן הרשות "לומדת" לבעצמָה-קְרָבֶּץ(להכפיל את הרזולוציה) של התמונות המוטשטש-האלה. תהליך זה משפר את יכולת התמונה הסופי-האלאן שבאופן ז-הסברים שהרשות תלמד דפוסים שגויים קטנה ממשמעותו.

7.2.7 StyleGAN

StyleGAN, שיצא בשלהי שנת 2018, מציע גרסה משודרגת של generator, עם דגש על רשות-ה-~~generator~~-מחברת המאפשרת לוPCI היתרון הפוטנציאלי של שכבות ProGAN מהמייצרת-תנו-בעמ'יקותן לשולחן בתוכנות (מאפיינים) ויזואליות שונות של התמונה, אם משתמשים בהן כראוי. ככל שהשכבה והרזולוציה נמוכה יותר, כך התכונות שהיא משפיעה עליה גסות יותר.

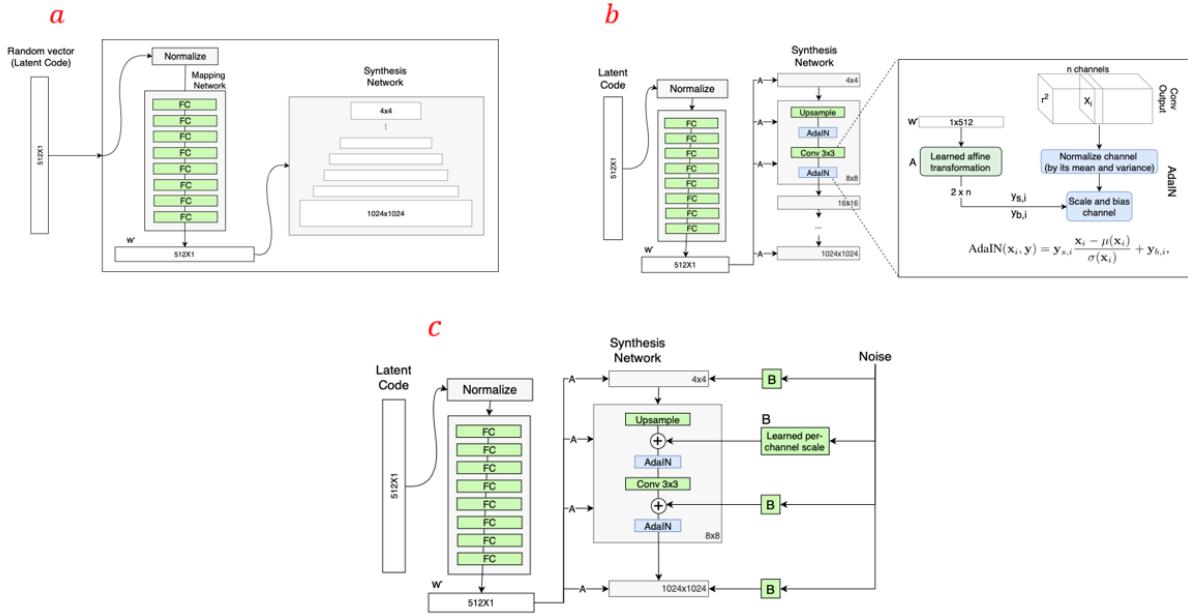
למעשה-הה-GAN-הראשון שנותן יכולת לשולחן מאפיינ-יזואליים (אומנם לא בצורה מלאה) של התמונה הנוצרת. מחברי StyleGAN חילקו את התכונות היזואליות של התמונה ל-~~פָּרָאַמְּנָטִים~~

- **גָּו:** משפיע על תנוצה, סגן שיער כללי, צורת פנים וכך
- **אמצעית:** משפיע על תווי פנים עדים יותר, סגן שיער, עיניים פקוחות/עכומות וצדדים
- **רזולוציה דקה:** משפיעה על צבע (עיניים/שיער/עור) ועל שאר תכונות המיקרו של התמונה.

כדי להעניק ל-~~GAN~~-StyleGAN-אלהו, נדרש מספק-שינויי-פְּבִיחָס=אל-ארקיטקטורה של GAN-~~פְּבִיחָס~~(נתאר רק את שלושת השינויים החשובים ביותר כאן):

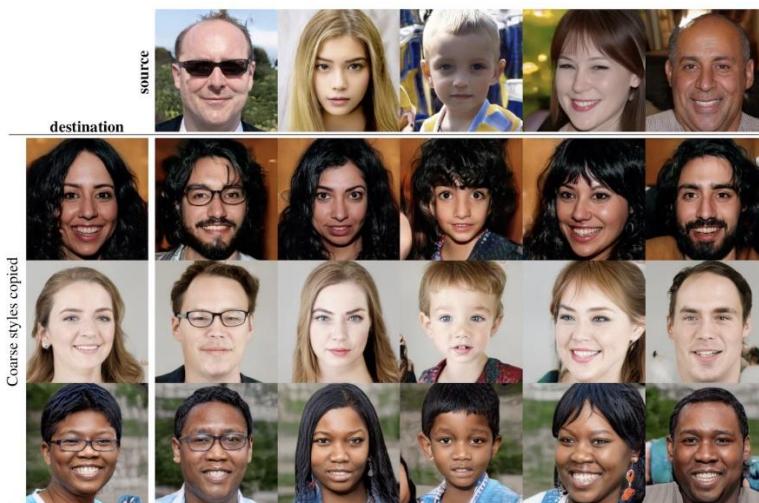
- **הוספה-רש-מיפוי=**מטרת רשות המיפוי היא קידוד וקטור הקולט לווקטורBINI-פְּבִיחָס(הנקרא וקטור סגנון)=אשר האיברים השונים של שולטים בתוכנות ויזואליות שנוצרו של התמונה הנוצרת. זהו תהליך לא טריוויאלי=מכיוון שהיכולת של הרשות לשולחן בתוכנות ויזואליות באמצעות וקטור הקולט-הינה-מוגבלת. הסיבה לכך טמונה בעובדה שוקטור הקולט נאלץ "לעקוב אחר ציפיות ההסתברות של סט האימון" שגורם לתופעה הנקרא-~~ה-פְּבִיחָס~~(FE=feature entanglement)-~~פְּבִיחָס=מאפיינים~~-(פְּבִיחָס=מאפיינים). בין תכונות צבע השיער והמגדיר-יכולים להופיע אם למשל-~~פְּבִיחָס~~ האימון מגמה כללית של גברים עם שיער קצר ונשים בעלות שיער ארוך. במקרה זה-הה-רשות תלמד שగברים יכולים להיות בעלי שיער קצר בלבד ולהיפך אצל נשים. כתוצאה לכך, אם "נשחק" עם רכיבי וקטור הקולט כדי ליצר תמונה של גבר בעל שיער ארוך, בסופו של דבר מגדרו ישתנה גם כן ונקבל תמונה של אישה
- רשות המיפוי-ההווסף לארקיטקטורה-הופכת את וקטור הקולט לווקטורBINI-פְּבִיחָס שאיינו צריך לעקוב אחר התפלגות של סט האימון, וכך-יש פחו-הערבות המאפיינ-במילים-אחרות, רשות זו פ-אפשרת א-ה-יכולה לשולחן ~~פְּמָאַפְּיִינִי-יזואליים~~ של התמונה הנוצרת באמצעות שני רכיביו של וקטורי-ה-פ-ושווא. רשות המיפוי מורכבה משמונה שכבות FC וגודל הפלט שלו זהה לגודל הקולט

- **החלפה BN בא-BN**=**Adain**=**Batch Normalization** (BN)=**AdaIN**=**שינור א-טמפרטרים** (Batch Normalization=BN). בשונה מ-BN, הטרטורים של המוצע ושל השונו-טבג'ישת AdaIN=למדים מוקטו והסגור נא-**ח**(הם בעצמת רנספורמציה לינארית) לשאלות עם משקלים למדים. להבדיל מ-AdaIN, במנגנון BN=טמפרטרי פרמטרים אלו למדים כמו המשקלים האחרים ולא תלויים במצב של שכבה כלשהי.
- **ויתור על אקריאות של וקטור קלט**=**StyleGAN**=**generator** וקטור הקלט אינו וקטור המוגדר מהתפלגות גausית אלא וקטור=Dטרמיניסטי עם רכיבים למדים. וקטור הרעש מתווסף כ-**ישירות** לפולטים של ערכיו קובולוציה ברשתות generator=העוממת להעזה מ-**generator** כל-ערוך בפנירד=שימוש בוקטור קלט Dטרמיניסטי במקום בוקטור אקראי מכך הנראה על הפרק מהמיופי ייעל ידי רשת המיפוי (ויתר קל לעשות זאת על וקטור קבוע מאשר להתאים את משקל רשת המיפוי לווקטור כניסה אקראי ימ').



איור 12. השינויים העיקריים בארכיטקטורת StyleGAN. (a) שימוש ב-Adain. (b) שימוש ב-StyleGAN+. (c) שימוש ב-StyleGAN++.

יש עוד כמה שינויים יותר מינוריים ב-StyleGAN ייחוסית ל-ProGAN, כמו שינוי של היפר פרמטרים של הרשתות, פונקציית מחיר וכו'. התוצאות הן לא פחות ממרשים מה-StyleGAN+-**_styleGAN**=**StyleGAN**=**StyleGAN+** יוצר תמונות שנראות ממש אמיתיות ולבונוס מקנה יכולת לשילוט בחלק מהתכונות החזויות של התמונה.



איור 13. תמונות שיזרו באמצעות StyleGAN.

7.2.8 Wasserstein GAN

אחסונגה-GAN=GAN-generator מוחשב כבוקטור ה- GAN , והוא נוגע בבעיה שיש בפונקציית המחבר בה משתמש הרב-הויריאנטים של-GAN-ים. כאמור, תהליך הלמידה של הרשת המיצרת דאטא-ה- GAN -generator=Discriminator- GAN נעשה באמצעות משוב המתקבל מה-discriminator. בעוד שה-sha-discriminator מאמון להבחן בין דאטאות לdatsets נינט-ה- GAN discriminator המתקבל מהתיקל דאטאות דאטאות אמיתיות אלא בעזרת דאטאות מה-sha-discriminator. משום כך, בתחום הלמידה, כאשר ה-sha-discriminator מושם על דוגמאות הדוגמאות רק על המשוב מוחזק. discriminator משומן כר, בתחילת הלמידה, כאשר ה-sha-discriminator מושם על דוגמאות הסינתטיות שהוא מיציר אכן דומות כל לדאטאות האמיתיות, וה-sha-discriminator מבחין בקבוצות ביניהם. במיללים אחרות, בתחילת תהליכי הלמידה ה-sha-discriminator מושם על ה-sha-generator מושם על ה-sha-generator. פער זה יוצר בעיה בתהליכי ההש趴רואת של-sha-generator, כיון שההיפור מתבסס על ה-sha-generator. generator ה-sha-generator מושם על פונקציית המחייב loss, תלוי בערכיהם אוטם מוציא ה-sha-discriminator. כדי להבין מדוע תהליכי העברות המידיע באופן זהה בעיתות loss של-sha-generator, generator על דאטאות על דאטאות ה-sha-discriminator. generator על דאטאות ה-sha-discriminator על דאטאות זהו שלהרחב מעט על תהליכי היזירה של דאטאות על דאטאות ה-sha-discriminator.

ההנחה היסודית ברוב המודלים הגנרטיביים, ובפרט ב-GANs, היא שהדאטה הרב ממד (למשל תמונה) "חיה" במשטח מממד נמוך בתוכו. אפשר להסביר על משטח בתור הכללה של ת-מרחב וקטורי מממד נמוך-הנפרה על ידי ת-קוצץ של וקטורי בסיס של מרחב וקטורי מממד גבוה יותר. גם המשטח נוצר מתת-קוצץ של וקטורי הבסיס של "מרחבי האם", אך ההבדל בין ת-מרחב וקטורי מתבआ בכרך של משטח עשוי להיות צורה מאוד מורכבתיחסית לתופ-מרחב וקטורי. משתמש מכך שניתן ליצור דאטה רבעמדי על ידי טרנספורמציה של וקטור מרחב בעל ממד נמוך (וקטור לטנסי). למשל, ניתן עזרת רשת נירונית ליציר תמונה בגודל- $k \times 3 \times 64 \times 64$ פיקסליהם בלבד. זאת אומrette-שגם התפלגות התמונות של הרשות הגנרטיבית-וגם התפלגות של הדאטאות האמיתית נמצאים ב"משטח בעל ממד נמוך" בתור מרחב בעל ממד גבוה של הדאטאות המוקופ. באופן פורמלי יותר, משטח זה נקרא-יריעה (manifold), וההשערה שתואירה מעלה מהוות הנחת יסוד בתחום הנקריא למידת-יריעות (manifold learning). מכיוון שמדובר במסטחים בעלי ממד נמוך ביחס למרחב בעל ממד גבוה, קיימת סבירות גבוהה שלא יהיה שום חיתוך-בין המשטח בו "חיה" הדאטה האמיתית לבין זה של הדאטה הסינטטי (לכל הפחות בתחלוף תהליך האימון של GAN), כלומר-מכך, המרחק-בין משטחים אלה עשוי להיות-ידי גדול. מכך נובע-שהdiscriminator עשוי למודד להבחין בין הדאטאות האמיתית ל-סינטטי בקבוקות, כיון שבמרחב מממד גבוה יש מרחק גודל-בキー-אמיות-בלב-היריעות-הdatatex. בנוספְּחָנָרָה יתן לדוגמאות סינטטיות צויניטקס (score) ממש קרובים לאפס כי אכן נמצא "משטח הפרדה" בין שתי היריעות-זה של הדוגמאות האמיתיות וזה של הסינטטיות, כיון שהם נתונים להיות רוחקים מאוד אחד מהשני.

רקע זה מסיע להבין מדוע הפער שיש ב**generator-discriminator** מבוחנת אופי הלמידה מהוועה בעיה כאמור generator-discriminator מעדכן את המשקלים שלו על סמך היצירום שהוא מקב'לה discriminator-discriminator (דרך פונקציית המחדיח-שבלב-GAN). אבל אפהה generator-discriminator מוציא יצירום מאוד נומוכים (עקב מרחק גדול ביחס להיריעות שתואר לעלה) לדוגמאות המיצירות על ידי generator, generator-generator פשוט לא יכול לשפר את איכות התמונה ששהוא מייצר. במילים פשוטות generator הרבה יותר מדי טוב יחסית ל-G".אתגר זה בא לידי ביטוי גם במצבה של פונקציית המחיר, שלא מאפשרת "הברחה עיליה של ידי" מערך discriminator-generator.

יש מספר לא קטן של שיטות הבאות לשפר את תהליכי האימון של GAN, אך אף אחת מהן אינה מטפלת בבעיה זו באמצעות שינוי של פונקציית המחיר. השיטות הבולטות הן:

- feature matching (פְּרִים) התקשרות
 - minibatch discrimination הבדנה בין-בatches
 - virtual batch normalization טבוניזציה בatches�ים
 - mixup טבוניזציה מילוקית

כפי שהוסבר, הבעה ש**שלהמראבקבצהיריעוומשתקפת** במבנה של פונקציית המחדל-וכיוון שכך-ניתן לנוטה ולפטור את הבעה מהשורש על ידי שימוש בפונקציית-מחיה-ייתור מתאימה= \hat{f} לשם כך ראשית-נסמן את התפלגות הדטה האמיתית p_{true} , ואת התפלגות הדטה ה סינטטית המיצרת על ידי ה- generator \hat{g} =לעיל הראיינו שפונקציית המחיר Jensen-Shannon divergence יתוארכעל $D_{JS}(p_{\text{true}}, p_{\hat{g}})$.

ניתן להוכיח כמראח \mathcal{D}_g בין התפלגיות p_g , r_g וכל רגיש-לשוניים ב- \mathcal{D}_g כאשר המשטחים שביהם "חימם" p_g מושגים אחד מהשניים-כולם, מרחוק \mathcal{D}_g כמעט ולא ישתנה-אחרי עדכן המשקלים של-ה-generator, generator לא רקוחים ישלוף את המרחק המעודכן בין שתי התפלגיות p_g - r_g $= \Delta \mathcal{D}_g$ שפונה הבעיה המהותית ביותר עם פונקציית המחיצ'ה המקורית של-ה-generator, שעדכן המשקלים לא משפיע כמעט עלי \mathcal{D}_g , כיוון שמדובר בתפלגיות p_g - r_g קרחוקות.

ב-Wasserstein GAN generator המשמש בפונקציית מחיר אחרת, בהעדרו המשקלים של γ generator גם במרקח בין ההתפלגויות μ ו- ν פונקציית המחיר החדש מוגבהת על מරחיק הנקרטי (EM) Earth Mover (EM) מරחיק וסרטני מסדרי ≥ 1 קבין שתי מידות הסתברות μ , על מרחיב M מוגדר באופן הבא:

$$W_p(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \int_{M \times M} d(x, y)^p d\gamma(x, y) \right)^{\frac{1}{p}}$$

כאשר (μ, ν) הן כל מידות הסתברות על מרחיב המכפלה (product space) של M עם עצמו (זהו למשמעות מרחב המכיל את כל הזוגות האפשרים של האלמנטים M - M) עם פונקציות שליליות (marginal) השווות μ , ν בהתאם ל- μ , ובאופן מפורש $\nu = \text{softmax}(M \cdot \mu)$.

$$EM = W_1(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{M \times M} d(x, y) d\gamma(x, y)$$

הגדירה זו נראית מאוד מושכנת ונוסה לתת עבורה אינטואיציה, ולהבין מדוע עבבו $= k$, מרחיק וסרטני נקרא M . לשם הפשטות-נניח שהמרקח M הוא חסמי, כלומר k ישר, ועליזעشر משקלות של 0.1_{kg} אחת המפוזרות באופן הבא: משקלות 0.6_{kg} ($x=0$), 0.4_{kg} ($x=1$), 0.3_{kg} ($x=2$), 0.2_{kg} ($x=3$), 0.1_{kg} ($x=4$), 0.05_{kg} ($x=5$), 0.02_{kg} ($x=6$), 0.01_{kg} ($x=7$), 0.005_{kg} ($x=8$).

כמובן שיש הרבה דרכים לבצע אופרצות המשקלות, ונרצה למצוא את הדרך היעילה ביותר. לשם כך נגידיר באמצעות מכפלה של משקל מרחיק אותו מזים את המשקל (בפיזיקה מושג זה נקרא עבודה-טחון המופעל על גופו לאורכו מסלול). בדוגמה המובאת, המאמץ המינימלי מתקיים על ידי הצעת המשקלות באופן הבא:

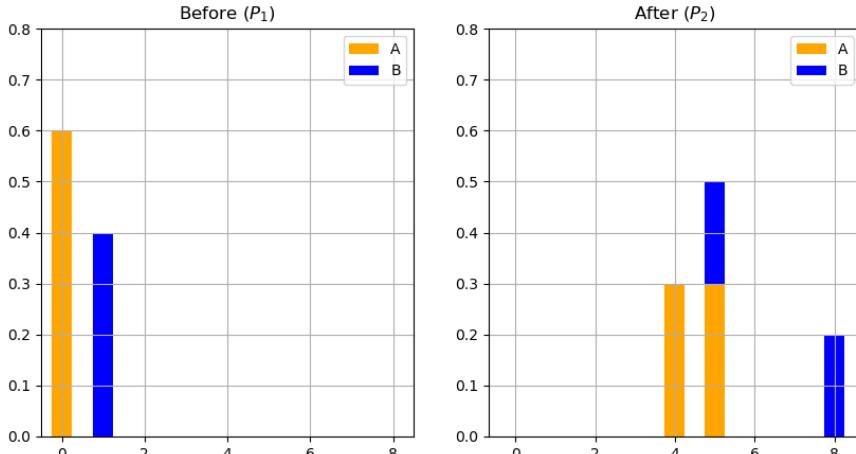
$$(4 - 0) \cdot 0.3 = 1.2_{kg}$$

$$(5 - 0) \cdot 0.3 = 1.5_{kg}$$

$$(5 - 1) \cdot 0.2 = 0.8_{kg}$$

$$(8 - 1) \cdot 0.2 = 1.4_{kg}$$

$$\text{סך המאמץ המינימלי שווה ל-} 4.9 = 1.2 + 1.5 + 0.8 + 1.4$$



איור 7.14 העברת משקלות באופן אופטימלי. P_1 מייצג את המצב ההתחלתי, P_2 הינו המצב לאחר הצעת המשקלות

כעת, במקום להסתכל על משקלים, נתיחס להtaplegiyot p_1, p_2 , המוגדרות באופן הבא:

$$p_1(x) = \begin{cases} 0.6, & x = 0 \\ 0.4, & x = 1 \\ 0, & \text{else} \end{cases} \quad p_2(x) = \begin{cases} 0.3, & x = 4 \\ 0.5, & x = 5 \\ 0.2, & x = 8 \\ 0, & \text{else} \end{cases}$$

השאלה כיצד ניתן להعبر ממנה הסתברותית Pr_1 ל- Pr_2 שמתאפשרת רק במקרה של הזוזת המשקולות=מרחץ-Earth Mover בין שתי התפלגיות μ_1, μ_2 , כלומר "מאמץ'=המינימלי' הנדרש בשביל להعبر או המסה ההסתברותית'=משקל- Pr_2 , או במילים אחרות=מרחץ-Earth Mover מגדיר מה כמות ה'"עבודה'=מאמץ'=המינימלי'ת הנדרשת בשביל להפוך- Pr_1 ל- Pr_2 =אם נחזור לדוגמא של המשקולות, נוכל להבין מדוע Pr_2 עבור 1 = קנקרא מרחץ-Earth Mover=מרחץ ביחסית התפלגיות שקיים ככל מה מאמץ נדרש להعبر כמות אדמתה משקל מסוים כדי לעبور מחלוקה מסוימת של אדמה לחולקה אחרת=באופן יותר פורמל'=מידת ההסתברות על מרחב המכפלה בנוסחה של מרחץ-Earth Mover את האופן שבו אנחנו מעבירים את המסה ההסתברותית (משקל מסוים של אדמה), כאשר הביטוי $\gamma(x,y)$ מתאר כמה מסה הסתברותית מועבקת מנקודת x לנקודה y.

לאחר שהסביר מהו מרחוק וירטואלי \mathcal{P} ומהו מרחוק EM, ניתן להבין כיצד אפשר להשתמש במושגים אלו עבור פונקציית מחיר של $\text{GAN}_{\mathcal{P}}$. בין מידות ההסתברות מתחשב בפתרונות של הקבוצות עליה קמידות אלו מוגדרות בצורה מפורשת, על ידי התחשבות במרקם קבוצות אלו: תוכונה זו היא למעשה בדיקת מה שצרכ' בפועל בצוותה מפורשת, על ידי התחשבות במרקם קבוצות דומות לזו. מושג זה מגדיר את המרחק $d_{\mathcal{P}}$, שמייצג את המרחק בין התפלגות האמצעית של דאות \mathcal{P} לבין התפלגות הדעתה הסינטטית s .

בשביל למדוד את המרחק בין התפלגות האמצעית ובין התפלגות השבחה \mathcal{P} , חווית השבחה מושג באמצעות קולומרים מודפסים אפורה ייעודית. מושג זה מגדיר את המרחק ביחס לטענה \mathcal{H} , אשר נוכל לדעתו בעזרת מושג $d_{\mathcal{P}}$ עד כמה השבחה המרתק ביחס לטענה \mathcal{H} שזה לא קורה ממשמשים בפונקציית המחיר המקורית הנמצדת באמצעות D_{js} . במקרה בודד, בעזרת פונקציית המחיר החדשה המבוססת על מרחוק EM, ניתן לדעתי עד כמה המשקלים מקריב או מרחיק את p_r מ- g .

בapon תיאורתי זה מצוין, אך עד'יך'זה לא מופיע, כיון שצריך למצוא דרך לחשב את \mathcal{D}_W , או לכל הפחות את המקרה הפרטני שלו עbow=1 = k , ככלומר את מרחק EM. במקור מרחק זה מוגדר כבעית אופטימיזציה של מידות הסתברות על מרחב המכפלה, וצריך למצוא דרך להשתמש בו כפונקציית מחיר. בשביל לבצע זאת, ניתן להשתמש בצורה דואלית של \mathcal{D}_W עbow=1 = k – שוויון קאנטורוביץ' (Rubinstein-Kantorovich), לפאניתן לחשב את k , \mathcal{D}_W באופן הבאות

$$W(p_r, p_g) = \frac{1}{K} \sup_{||f||_L} E_{x \sim p}[f(x)] - E_{x \sim p_s}[f(x)]$$

כ-אש-(x) **הינה-פונקצייה-K-ליפשי-דיסקרימינטור** (kolmorfunktsiya-k-lipshiz-discriminator) עם קצב השתנות החסום על י-ד-K) =cutet nni-k ש-(w) **הינה-פונקצייה-K-ליפשי רציף הדמתור**=discriminator בערך ה-permatorip=ה-he-machshab baavon makorot at merakik bin hahteflogiot baavon habat

$$L(p(r), p(g)) = W(p(r), p(g)) = \max_{w \in W} \mathbb{E}_{x \sim p_r} [f_w(x)] - \mathbb{E}_{z \sim p_r(z)} [f_w(g_\theta(z))]$$

פונקציית מחיר זו מודדת אופנה מרוחקת \hat{W} ביחס לתפוגיות p_r , p_g וככל שפונקציה זו מקבל ערכים יותר נמוכים כך generator יצליח ליצור דוגמאות שמתפלגות באופן יותר דומה לדאטת המקור: בשונה מ-GAN discriminator מוציא הסתברות עד כמה הדוגמא-אותה הוא מתקבל אמיתי, פהה-discriminator לא מאמין discriminator בזיהויו של דוגמא אמיתי לointpatch, אלא מאמין למודול פונקציית K -Lipshitz רציפה=המודדת את \hat{D}_W בהבחן בין p_r , p_g generator לעומת זאת מאמין למצער את L כאשר רק האיבר השני שתלו' ב- p_g התפוגיות p_g , p_r . ה-generator מתקבב יותר ככל שפונקציית המחר הולכת וקטנה, כך m מתקבב יותר ככל \hat{K} .

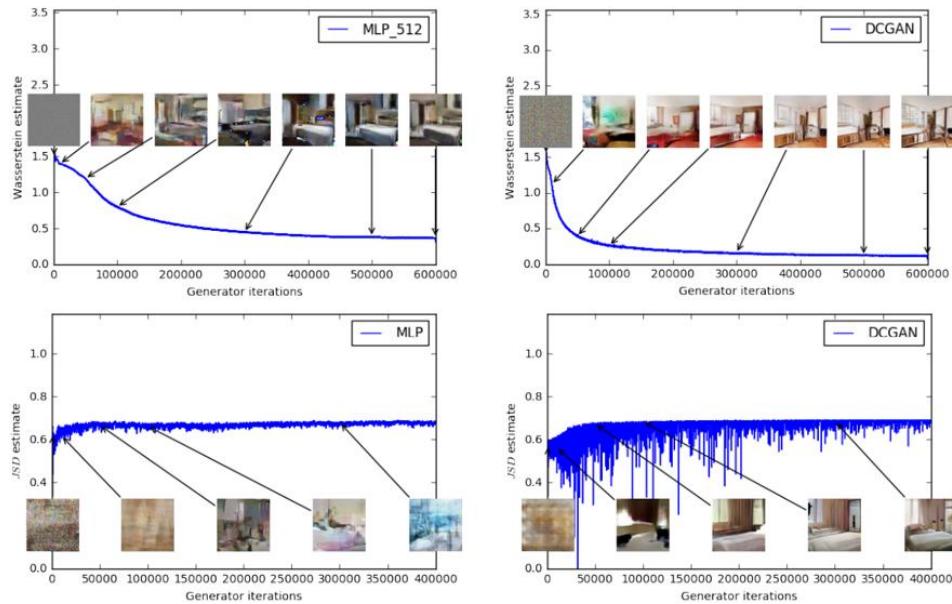
כאמור, תנאי הכרחי לשימוש במרקח זה בפונקציית המחיר הינה שפונקציית התהיה- \mathcal{A} -ליפשץ-רכיצה=מסתבר שקיום תנאי זה אינו ממשימה קלה כל-כך להבטיח את קיומו של המאמר המוקורי היציע-לבצע=קטירה=של משקלן. כיון discriminator של פולטוח סופי מוסף=NNTIC[0.01, 0.01]– ניתן להראות כי קטינה זו מבטיחה אכן שתהיה- \mathcal{A} ליפשץ רציפה. אולם, כמו שכתבתי המאמר מודים בעצמו, ביצוע קטינה בכדי לדאוג לכך ליפשץ יכול לגרום בעיות אחרות=למעשה=כאשר חלוקהקטימך של המשקלים-כמ"ד=הגרדי-אנטום של Wasserstein געלוים לתאפס, מה שיאט את תהליכי הלמידה. מצד שני, כאשר חלון זה רחב מכך, התוכנות עלולה להיות מאוד איטית. נציג שיש עד מספר דרכי לכפות ערך w_f להיות ליפשץ-רציפה למשל gradient penalty.

האימון של Wasserstein GAN דומה לאימון של הGAN המקורי, למעט שני הבדלים עיקריים:

- א. קיזוצטווח המשקלים על מנת לשמר על רציפות-ליפשייך.
 ב. פונקציית שמחיר המסתמכת על D_W במקומם על D_S .

תהליכי הימידם מתקיימים ב순דרה ההפוכה. לאחר עדכון משקלים של discriminator (gradient ascent), מתקיים עדכון משקלים של generator (gradient descent). לאחר מכן מושתמשים בgenerator כדי ליצור נתונים אטראקטיביים (realistic-looking) ובדוח על discriminator אם הם מזויפים (fake).

GAN-Wasserstein מצליח לגרום לכך שהקורלציה בין איקות התמונה הנוצרת על ידי-generator-לבן ערך של פונקציית לוסטיה הרבבה יותר בולטות מאשר-GAN=רגיל בעל אותה ארכיטקטורה. ניתן להמחיש זאת היבר באמצעות גרפים הבוחנים את היחס בין D_{GAN} לבין D_{WGAN} :



איו-15. ק' שערוך מוחק א' ב' ק' כפונקציה של מספר האיטרציות (בגרפים הקיימים), לעומת שערוך מוחק ב' ק' כפונקציה של מספר האיטרציות (בגרפים הקיימים)

ניתך להראות בבירור כי ככל שאיכות התמונות ש-*generator*-*z* מיצר עולמית-*Ch* הולך וקטן, ואילו מרחק *S* לא מראה שום סימן של ירידה. הצלחה זו נובעת מהשינוי בפונקציית המחיר, שגרם לאימון להיות יותר יעיל, והביא לכך שהדוגמאות הסינטטיות תהינה דוחות הרבה יותר לדاطה המקורי.

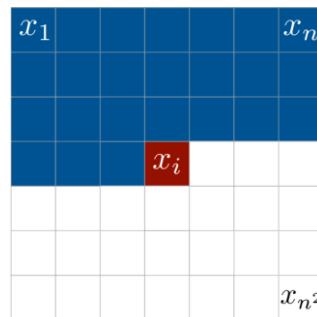
נקודה נוספת שיכולה להסביר את ההחלטה היחסית-השלמה שהושם ב-WEB-נובע-השימוש ורשטייך-חלשהיחסית למטריקות/², וננסה להבהיר נקודה זו

באופקאי-ישראל, כאשר אנו מאמנים מודל ה-*ייניו* רוצים להיות בטוחים שאם ננעה בזרה נאותה ובכל צעד נעדק המודל בבדיקה על פי הוראות הגדרי-אנט, נסימן-או-האימון בנקודה מסוימת אופטימלית=אול-כבר-בפועל זה לא תמיד כך, כיון שישנן בעיות שעבורן מטריקות מסוימות יגיעו לנקודה זו ואחרות לא. ניקח לדוגמה-שנינו אナンשי-শ-עומדים על סף-תוהוף ורוצים להציגו לעמך=האחד מודד את הגובה ומתקדם על פיזי-ולכך-הוא יגיע למטרבקלות יחסית. האחר מתעניין במיקומו על ציר צפון דרום, וכך הוא עשוי להיתקל בקש-יב-במהלך הירידה, וגם אם הוא אכן יגיע למטרה, זה בהכרח יהיה בתהילך איטי יותר=באופן דומה=כasher לווקחים זוג מטריקות=באופן פורמל-פניטן להגדי-שאים התכנסות-ושל סדרה ה-תפלגיו-ותחת מטריקה אוחז-גורה-ההתכנסות-של הסדרה-תחת מטריקה-אחרת, איזה-המטריקה הראשונה חזקה יותר מהמטריקה השפהיה. העובדה ש- \mathcal{D}_W -חלש יותר מ- \mathcal{D}_{JS} -בעצמאותה שיתכן ישבעיות שעבורן תתקבל תוצאה אופטימלית עבורי- \mathcal{D}_W אך לא עבורי- \mathcal{D}_{JS}

7.3 Auto-Regressive Generative Models

משפה נוספת שמודלים גנרטיביים נקראות Auto-Regressive Generative Models (VAE), ובדומה ל-VAE גם מודלים אלו מוצאים התפלגות מפורשת של מרחב מסוים:=ובעדרת התפלגות זו מייצרים=dataset=חדש=:עם זאת, בעוד VAE מוצא קירוב להתפלגות של המרחב הlatent=:שיטות AR=:מנסות לחשב במדויק התפלגות מסוימת, וממנה לדגום וליצרך=dataset חדש.

בapon צה שהוא תלוי בכל הפיקולים שלפניו. תМОנו נ-אַכְבוֹד לָאָח × וְהִיא לְמַעֲשֵׂה רַצְף שֶׁל² אַכְפִּיקְסּוֹלִים. כַּאֲשֶׁר רֹצִים לְיוֹצֵר תְּמוֹנוֹת, נִתְן לְיוֹצֵר כָּל פָּעָם כָּל פִּיקְסּוֹלִים.



איור 15.7 תמונה כרצף של פיקסלים.

כל פיקסל הוא בעל התפלגות מותניתה

$$p(x_i|x_1 \dots x_{i-1})$$

כאשר כל פיקסל מורכב משלושה צבעים (RGB), لكن ההסתברות המדוייקת היא:

$$p(x_{i,R}|x_{<i})p(x_{i,G}|x_{<i}, x_{i,R})p(x_{i,B}|x_{<i}, x_{i,R}, x_{i,G})$$

כל התמונה השלמה היא מכפלת ההסתברויות המותניות:

$$p(x) = \prod_{i=1}^{n^2} p(x_i) = \prod_{i=1}^{n^2} p(x_i|x_1 \dots x_{i-1})$$

הביטוי (x) קहוא ההסתברות שלדעתם ל x לתמונה אמיתית, لكن נרצה למקסם את הביטוי הזוהה לקבץ מודל שמייצג תמונות שנראות אוטנטיות עד כמה שניתן.

7.3.1 PixelRNN

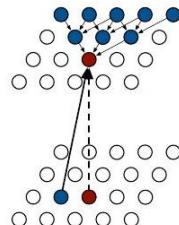
אפשרות אחרת לחשב אורך(x) קהיא להשתמש ברכיבי זיכרון-CMSTM עבור כל פיקסל=באופן טבעי הינו רציף linked כל פיקסל לשכנים שלו

$$\text{Hidden State } (i,j) = f(\text{Hidden State } (i-1,j), \text{Hidden State } (i,j-1))$$

הבעיה בחישוב זה היא הזמן שהוקח לבצע אורך=כיוון שכן פיקסל דורש לדעת את הפיקסל שלפניו – לא ניתן לבצע אימון מקבילי לרכיבי LSTM. כדי להתגבר על בעיה זו הוצעו כמה שיטות שונות לאפשר חישוב מקביל:

Row LSTM

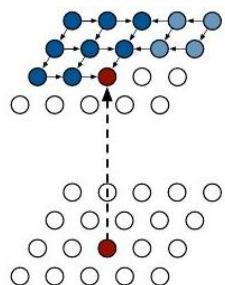
במוקם להשתמש במצב החבוי של הפיקסל הקודם, ניתן להשתמש רק בשורה שמעל הפיקסל אותו רוצים לחשב. שורה זו עצמה מחושבהלא פנוי כעל ידי השורה שמעליה ובקצה למשה לכל פיקסל יש receptive field של מושלש. בשיטה זכנית לחשב באופן מקבילי כל שורה בנפרד, אפשר לכך מחיקת איבוד הקשר בין פיקסלים באותה שורה (loss context).



איור 16 Row LSTM 7.16 – כל פיקסל מחושב על ידי $\geq k$ פיקסלים בשורה שמעליהם.

Diagonal BiLSTM

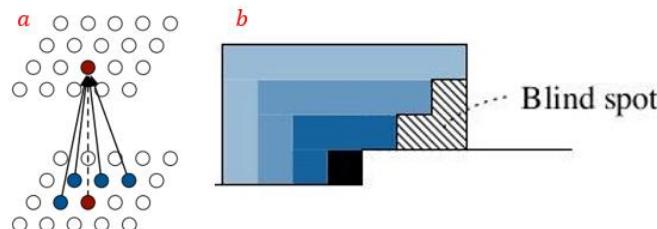
כדי לאפשר גם חישוב מקבילי וגחסנירה על קשר עם כל הפיקסלים, ניתן להשתמש ברכיבי זיכרון דו כיווניים=בכלי שלבי מחשבים את רכיבי הזיכרון משני הצדדים, וכך כל פיקסל מחושב גם באמצעות הפיקסל שלידו וגם על ידי זה שמעליו=באופן הזרה-field receptive field גודל ותחזוקה אפשר, אך החישוב יותר איטי מהשיטה הקודמת, כיון שהשורות לא מחושבות בפעם אחת אלא כל פעם שני פיקסלים



איור 7.17 Diagonal BLSTM – כל פיקסל מחושב על יד $\geq k$ פיקסלים בשורה שמעלitz' כדי לשפר את השיטות המשמשות ברכיבי זיכרון ניתן להוסיף עוד שכבות, כמו למשתמש residual blocks שיעזרים Masked convolutions כדי להפריד את התלות של הערכים השונים של כל פיקסל.

7.3.2 PixelCNN

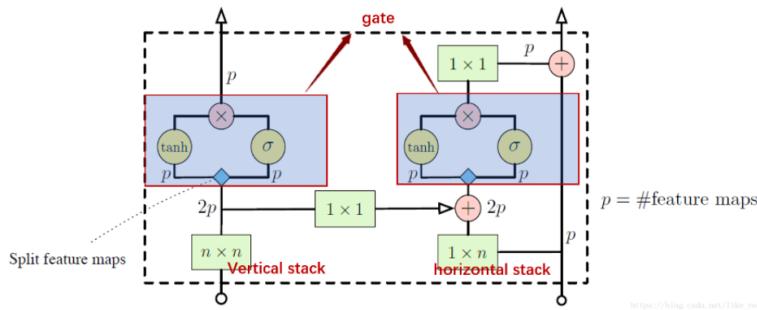
הчисIRON העיקרי של PixelRNN=Nובע מהאימון האיטי שלובם מקום רכיבי זיכרון ניתן להשתמש ברשת קונבולוציה, בכר להאיץ את תהליך הלמידה והגדיל את receptive field. גם בשיטה זו מתחילה מפהיקסל הפינתי, רק כעת הלמידה היא לא בעדרת רכיבי זיכרון אלא באמצעות שכבות קונבולוציה=היתרונות של שיטה זו על PixelRNN=Mtabteakatzoozmanot של תהליכי האימון, אך התוצאות פחות טובות=חסיטו מושג בשיטה זו נובע מהמבנה של המסלנים=ה receptive field=pיקסל מtabso עלי שלושה פיקסלים שמעל, והם בתורם כל אחד תלוי בשלושה פיקסלים בשורה שמעל. מבנה זה מנתק את התלות בין פיקסלים קרובים יחסית אך אינם ב receptive field=blind spot



איור 7.18 receptive field של PixelCNN. (a) החיסIRON של PixelCNN – ניתוק בין פיקסלים יחסית קרובים.

7.3.3 Gated PixelCNN

בכדי להתגבר על בעיות אלה=ביצועים לא מספיק טובים והתעלמות מפיקסלים יחסית קרובים שאינם ב receptive field – נעשה שימוש ברכיב זיכרון הדומה ל-LSTM, המשלב את רשותות הקונבולוציה בתוך RNN.



איור 7.19 שכבה של Gated PixelCNN.

כל רכיב זיכרון בני משני חלקים =horizontal stack and vertical stack, כאשר כל אחד מהם הוא למעשה שכבות קונבולוציה. ה- vertical stack בנייתו מזיכרון של כל השורות שהו עד כה בתמונה, וה- horizontal stack הומצא יחד על הקלט הנוכחי. ה- horizontal stack עובר דרך שער של אקטיביזציה לאינאריות ובנוסף מתחבר ל- vertical stack, כאשר גם החיבור ביניהם עובר דרך שער של אקטיביזציות לאינאריות. פנוי כל כניסה של stack לתוך שער stack, המסננים מתפצלים – חצי עבריים דרך tanh וחצי דרך סיגמאיד. בסך הכל המוצא של כל שער הינו

$$y = \tanh(w_f * x) \odot \sigma(w_g * x)$$

7.3.4 PixelCNN++

שיפור אחר של PixelCNN הוצע על ידי OpenAI, והוא מבוסס על מספר מודיפיקציות

- שכבת MaxSoftPool שקובעת את צבע הפיקסל כorzכת הרבה זיכרון, כיוון שיש הרבה צבעים אפשריים. בנוסף, היא גורמת לארדיינט להתאפשר מהר. כדי להתגבר על כך ניתן לבצע דיסקרטיזציה לצבעים, ולאחר טווח צבעים קטן יותר. באופן זה קל יותר לקבוע את ערכו של כל פיקסל, ובנוסף תהליכי האימון יותר יעילים.
- במקרה לבצע בכל פיקסל את ההתניתה על כל צבע בנפרד (כפי שהראינו בפתחה), ניתן לבצע את ההתניתה על כל הצבעים יחד.
- אחד האתגרים של PixelCNN הוא יכולת המוגבלת למצוא תלויות בין פיקסלים רחוקים. כדי להתגבר על כך ניתן לבצע down sampling, ובכך להפחית את מספר הפיקסלים בכל מסנן, מה שמאפשר לשמור את הקשרים בין פיקסלים בשורות רחוקות.

- בדומה לNet-U, ניתן לבצע חיבורים בעזרת Residual blocks ולשמור על יציבות במהלך הלמידה
- שימוש ב-Dropout לצורף רגולרייזציה והימנעות מ-fitting

7. References

VAE:

<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

<https://jaan.io/what-is-variational-autoencoder-vae-tutorial/>

<https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html>

GANs:

<https://arxiv.org/abs/1406.2661>

<https://arxiv.org/pdf/1511.06434.pdf>

<https://phillipi.github.io/pix2pix/>

<https://junyanz.github.io/CycleGAN/>

<https://arxiv.org/abs/1710.10196>

<https://arxiv.org/abs/1812.04948>

<https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431>

<https://arxiv.org/abs/1701.07875>

AR models:

<https://arxiv.org/abs/1601.06759>

<https://arxiv.org/abs/1606.05328>

<https://arxiv.org/pdf/1701.05517.pdf>

<https://towardsdatascience.com/auto-regressive-generative-models-pixelrnn-pixelcnn-32d192911173>

https://wiki.math.uwaterloo.ca/statwiki/index.php?title=STAT946F17/Conditional_Image_Generation_with_PixelCNN_Decoders#Gated_PixelCNN

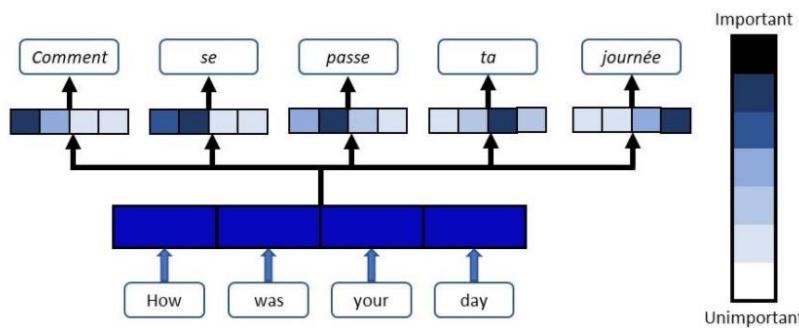
8. Attention Mechanism

8.1 Sequence to Sequence Learning and Attention

8.1.1 Attention in Seq2Seq Models

ניתוקסדרות בהן יש קשר בין האיברים-יכל להיעשות בעזרת רשתות עם רכיבי זיכרון, כפי שהואר בפרק 6=ברשות אל-הסדרה הנכונות לרשות עוקב והדריך encoder-decoder ידוע מושך המציג אופחסדרה המקורית-תוך התוצאות **סדר** של איברי הסדרה ובקש בינהם-לאחר מכון וקטור זה עובר ב-**decoder**=seq2seq מפענה את המידע שיש בוקטוף להציג אותו בצורה אחרת. למשל בתרגום משפה לשפה שפה(seq2seq) מודול שפה אחת לווקטור מסוים ולאחר מכן מפענה את הווקטור לשפה השנייה

הדרך המקובלת-ליצור אופחאוקטוף-ולפענה אוניהיתה שימוש-bearbeitketות שנותן של-RNN, כמו למשרשות עמו קומסוט-STM או GRU. המכללה רכיבי זיכרון-מודלים אלו נתקלו בבעיה בסדרות ארוכות, כיון שהווקטור מוגבל ביכולת שלו להכיל קשרים ביקספ-רבד של איברים כדי להתמודד עם בעיה זינית-לונקוש-בגישה שונת-במקומם ליצור וקטור במצאת encoder, ניתן להשתמש במצבים החובים של ה-encoder בשילוב המצביעים של ה-decoder וכך-למצוא תליות-בן אירט-סדרת הקלט-אליבר-סדרת הפלט(general attention) וקשרים בין איברי סדרת הפלט-עצמם (self-attention)=נוקיח לדוגמא-תרגם של המשפט "How was your day"="מאנגלית לשפה אחרת-במקורה זה-המנגן ה-attention מייצת-ווקטור חדש-עבור כל מיל-הבסורה הפלט-כאמון רכיב צווקט-ומכמפעע כמה-המילה הנוכחית במצוא-קשרה לכל אחות-המילים במשפט המקורי-באופן זהה כל איבר-סדרה הפלט ממוקל-כל אחד מאיבר סדרת הפלט. מנגנון זה נקרא attention.



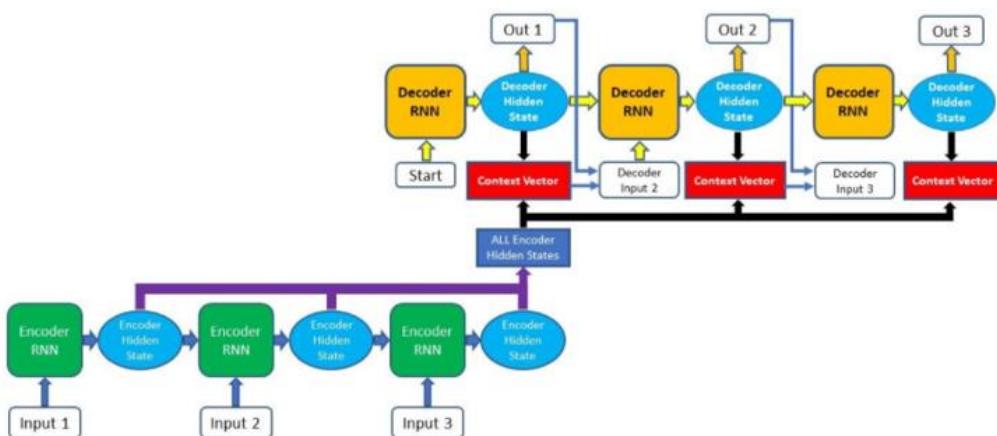
איור 8.1 מנגנון attention – נתינוקמישקל לכל אחת מילות הפלט ביחס לכל אחת מילות הפלט.

במאמר משנת 2017 שנקרא "Attention is All You Need" הוצע להשתמש ב-self-attention בלבד ללא רשות מסות או GRU, ומאמר זה פרץ דרך לשימושים רבים במנגן זה תוך קבלת ביצועים מעולים.

בחלק זה יוצג היחס המשלב self-attention וDecoder RNN לבני transformer המשמש ב-self-attention ו-self-position encoding ו-self-attention.

8.1.2 Bahdanau Attention and Luong Attention

הגישה הראשונה שהוצעה נקראת Bahdanau Attention על שם הממצא שלה – Dmitry Bahdanau



איור 8.2 ארכיטקטורת Bahdanau Attention

הרעין של גישה זו היא לבנות ארכיטקטורה בה משתמשים בכל הממצבים החבויים- \vec{h} של רכיבי הדיזרקטור-encoder ומעבירים אותם ל- \vec{H}_d -decoder כתוכאה אסכמה- \vec{H}_e . מוחשב את המוצא לא רק על סך מצביה קודמים, אלא משקל לאוטופיחד עם הממצבים החבויים- \vec{h} של encoder- \vec{h} עבור כל אחד מאיברי סדרת הפלט- \vec{H}_d מוחשיים- \vec{H}_e score context vector מנגנון attention, וכך קשור בין הפלט לפט, ובנוסף מוחשב עובוכל איבר של סדרת קלט- \vec{H}_d מסקל ייעשה למתוך כל אחד מאיברי הקלטה- \vec{H}_e .

ביצוע פעולה זו יוצרת לכל אחד מאיברי הפלט- \vec{H}_d ייחודי משלו הונבנה גם הממצב הקודם וגם מאיברים- \vec{h} encoder, בשונה מהארQUITקטורות הקודומות של seq2seqencoder- \vec{h} , שהן לא הינה ניתנת להעיבר מידע באופן ישיר מהמצבים החבויים של decoder- \vec{H}_e מוחשיים- \vec{H}_d את ה- \vec{H}_d מוצאים של האיבר הקודם ב- \vec{H}_d ,decoder- \vec{H}_e עם הממצב החבוי הקודם יוצרים את הממצב החבוי הבא, שבעזרתו מוצאים את הפלט של האיבר הנוכחי- \vec{H}_d בעופון פורמלי, אם נסמן ב- w_d , H_d את הממצבים החבויים של encoder- \vec{H}_e מוחשיים- \vec{H}_d , w_e alignment score יתקבב על יחס:

$$\text{alignment score} = w_{\text{alignment}} \times \tanh(w_d H_d + w_e H_e)$$

כאשר w_d , w_e הם המשקלים הנלמדים של ה- \vec{H}_d וה- \vec{H}_e encoder- \vec{H}_e , decoder- \vec{H}_d בינם לביןם את התוצאה: מعتبرים דרך SoftMax(H_e , w_e) מתקבלים את ה- \vec{H}_d מוחשיים- \vec{H}_e context vector

$$\text{context vector} = H_e \times \text{SoftMax}(\text{alignment score})$$

הווקטור המתיקב מכך ששל כל אחד מאיברי הפלט יהיה לאיבר הפלט הנוכחי- \vec{H}_d את התוצאה כאמור מוחשיים- \vec{H}_d לפט של האיבר הקודם, ובעזרת הממצב החבוי הקודם מוחשיים- \vec{H}_d את הממצב החבוי הנוכחי, שמננו מחלצים את הפלט של האיבר הנוכחי- \vec{H}_d .

ישנו שיפור של Bahdanau attention ביחס ל- Loung attention. שבחבדלי פער עיקרי יש בין שני המנגנונים: חישוב alignment score- \vec{H}_d מוחשיים- \vec{H}_e מatabase- \vec{H}_d בפועל שונות- \vec{H}_e מdatabase- \vec{H}_d בכל שלב לא משתמשים בממצב החבוי הקודם של decoder- \vec{H}_e , שהוא אלא יוצרים ממצב חבוי חדש ובעזרתו מוחשיים- \vec{H}_d את ה- \vec{H}_d מוחשיים- \vec{H}_e .

8.2 Transformer

לאחר שמנגנון attention- \vec{H}_d הוכח תואזה, הומצאה ארכיטקטורת המבוססת על attention בלבד ולא שופך רכיבי זיכרון. ארכיטקטורה זו הנקראת transformer מושפעת שני אלמנטים חדשים על מנת למצוא קשרים בין איברים בסדרה מסוימת – self-attention – positional encoding

8.2.1 Positional Encoding

ארQUITקטורות מבוססות RNN מושתמשות ברכיבי זיכר- \vec{h} כבסיס ללחשת בחשבון אפסה- \vec{h} של האיברים- \vec{h} בסדרה אחרה-בלי ציוג הסדר בין איברי הסדרה נקראת positional encoding נkirat- \vec{h} ביחס לזמן- t של כל אחד מאיברי הפלט פיסות מידע לגבי המיקום שלו בסדרה, והואוספה זו כאמור באח כתחליף לרכיבי הזיכרון ברשות RNN. בעופון פורמלי- \vec{h} עובוכל סדרת קלט- $\vec{H} \in \mathbb{R}^d$, מוחשיים- \vec{H}_d וקטור $\vec{p}_t \in \mathbb{R}^{d \times 1}$ באופון הבא:

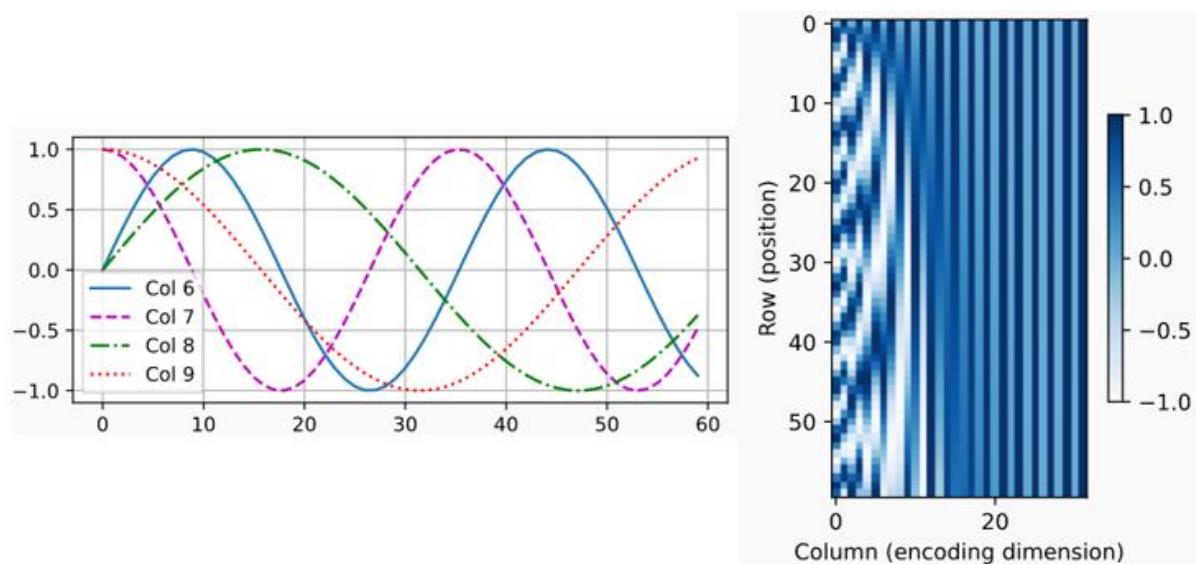
$$p_t(i) = \begin{cases} \sin(\omega_k t), & i \text{ is even} \\ \cos(\omega_k t), & i \text{ is odd} \end{cases}, \omega_k = \frac{1}{10000^{\frac{2k}{d}}} \rightarrow p_t = \begin{bmatrix} \sin(\omega_1 t) \\ \cos(\omega_1 t) \\ \sin(\omega_2 t) \\ \cos(\omega_2 t) \\ \vdots \\ \sin\left(\omega_{\frac{d}{2}} t\right) \\ \cos\left(\omega_{\frac{d}{2}} t\right) \end{bmatrix}_{d \times 1}$$

בכדי להבין כיצד וקטור זה מכיל מידע של סדר בין דברים- \vec{H}_d נציג את הרעיון שהוא מייצג בצורה יותר פשוטה. אף נרצה לקחו רצף של מספרים וליצג אותם בצורה בינה-יתנית- \vec{H}_d נוכל לראות שככל שלביבי יש משקל גדול יותר- \vec{H}_d הוא משתנה בתדריות נמוכה יותר, ולמעשה תדריות שינוי הביט היא אינדיקציה למיקום של-

0:	0	0	0	0	0	8:	1	0	0	0
1:	0	0	0	1	0	9:	1	0	0	1
2:	0	0	1	0	0	10:	1	0	1	0
3:	0	0	1	1	0	11:	1	0	1	1
4:	0	1	0	0	0	12:	1	1	0	0
5:	0	1	0	1	0	13:	1	1	0	1
6:	0	1	1	0	0	14:	1	1	1	0
7:	0	1	1	1	0	15:	1	1	1	1

איור 8.3 יציג בינהר של מספרים זה-MSB משתנה בתדריות הcy נומוכה, ואילו ה-LSB משתנה בתדריות הcy גבואה.

כיוון שמתעסקים במספרים שאינם בהכרח שלמים, הייצוג הבינארי של מספרים שלמים הוא יחסית בזבזני, ולקודם
לפחות גרסה רציפה של אותו רעיון – פונקציות טריגונומטריות עם תדרות הולכת וגדלה זהה בעצתהווקטור – הוא צ^ט
מכל הרבה פונקציות טריגונומטריות בעלות תדרות הולכת וקטנה – ולפי התדריות שמתוווספת לכל איבר בסדרה
המקורית ניתן לקבל אינדייקציה על מיקומו.



=**Positional encoding** =**איזור** =**המחשה** לקבץ השינוי של כל פונקציה בהתפקידו מיקום של האיבר אותו היא מייצגת =**מעין גרסה רציפה** לקבץ שינוי הביטים ביצואו של מספרים שלמים (ימין)

ישנו יתרון נוסף שיש לשימוש בפונקציות הטריגונומטריות=עבור כל צמד פונקציות בעלות אותו תדר= ניתן לבצע טרנספורמציה לינארית ולקבל תדר אוחט^{Relative Positional Information}:

$$M \cdot \begin{bmatrix} \sin(\omega_k t) \\ \cos(\omega_k t) \end{bmatrix} = \begin{bmatrix} \sin(\omega_k t + \phi) \\ \cos(\omega_k t + \phi) \end{bmatrix}, M = \begin{bmatrix} \cos(\omega_t \phi) & \sin(\omega_t \phi) \\ -\sin(\omega_t \phi) & \cos(\omega_t \phi) \end{bmatrix}$$

באותן הזה מתקבל בואופן מיד' ייחוס בין כל ה-positions, מה שיכל לעזור בניתוח הקשרים שבין איברים שונים.

8.2.2 Self-Attention Layer

בנוסף ל-positional encoding, עליה הרעיון לבעצמם attention, רק בין איברי הקלט לאיברי הפלט, אלא גם בז' איברי הקלט בעצמם=הרעין הוא ליציר יציג חדש של סדרת הקלט באותו אורך כמו הסדרה המקורית, כאשר כל איבר בסדרה החדש יציג איבר בסדרה המקורי בתוספת מידע על הקשר שלו לשאר האיברים=הרעין הכלל=אומך שיש לקח מה כל איבר בסדרה, ולחשב את הדמיון של כל איברים בסדרה=איברים דומים=קורוביטים=בסיסה יקבל ערכדים מי-קבוהים=ואילך איברים שונים=רחוקים=בסיסה יתנו ערכים נומינטיבים=ב-PLA זה יכול להיות מילימטר שופיעו בסמכיות, ובתמונה זה יכול להיות פיקסלים דומים=דמיון בין איברים נמדד על פי הקש=שייש ביניהם, והוא מחושב באמצעות מכפלה פנימית בין וקטור יציג של האיברים=כל מכפלה פנימית בין שני איברים נותנת מקדם שהוא מספר ממשי, וכקונטן לסקם את מכפלה כל המקדמים באיברים המקוריים=קלקלבל יציג חדש לאיבר המקורי

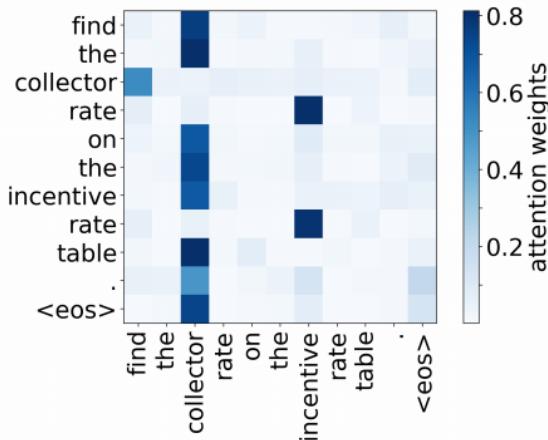
המכלול גם קשר בין האיבר הנוכחי לבין איברים דומים בסדרה-במיילים לאחרות-ניטן להסתכל על וקטור המכלול את הקשרים של איבר מסוים בסדרה כיצוגו החדש המשקף את קשריו עם שאר איברי הסדרה.

באופן פורמלי, בשביל $\text{Attention} = \text{SoftMax}(\text{Query} \cdot \text{Key}^T \cdot \text{Value})$, כאשר כל אחד מהטריצות של מטריצות-הכוניסיה המטריצות נקראות Query , Key , Value עבור סדרת משלבים באיבר הקלט. בעזרה מטריצות אלו מחשבים את-score-attention את-

$$\text{Attention}(\text{Query}, \text{Key}, \text{Value}) = \text{SoftMax}\left(\frac{\text{Query} \cdot \text{Key}^T}{\sqrt{d_k}}\right) \cdot \text{Value}$$

כדי להסביר כיצד הנוסחה מפעילה במציאות קשר בין איברים, נבחן כל איבר-של-הכוניסיה עבור סדרת קלטים x מקבלים שלוש מטריצות, כאשר כל איבר בסדרה המקורית i יוצר שורה בכל אחת מהמטריצות כאשר לוקחים את השורה $x_i = Q_{i \cdot} \cdot \text{Query}_i$ ומתקבלים אותה בכל אחת מהשורות במטריצה $K = \text{Key}_i$ -מקבלים וקטור חדש, שככל איבר-ב-בוקטור אומר עד כמה יש קשר בין האיברים, ובסדרה המקורית-ביבצח-ההכפלה זו עברו כל סדרת הקלטים-מטריצת-חישוב-הכפלה כל שורה מייצגת את הקשר בין איבר מסוים לשאר איברי הסדרה-ההכפלה זו היי-בעצם $K \cdot \text{Value}_i = \text{Value}_i$. וכך כל מכפלת $\text{Query}_i^T \cdot \text{Key}_j$ מייצגת את הקשר בין האיבר-ל-איבר- j את התוצאות המחלקים בשורש של ממד-הכפלה לשמור על יציבות הגרדיינט, ולאחר מכן מNORMALIZATE על יד SoftMax. באופן זה מקבלים מטריצה של מספרים בטוויאן $[0, 1]$, הנקראת w_{ij} , וונכל לקובץ אותו ישירות על ידי הנוסחה

$$w_{ij} = \text{SoftMax}\left(\frac{\text{Query}_i \cdot \text{Key}_j^T}{\sqrt{d_k}}\right) = \frac{\exp\left(\frac{\text{Query}_i^T \cdot \text{Key}_j}{\sqrt{d_k}}\right)}{\sum_{s=1}^n \exp\left(\frac{\text{Query}_i^T \cdot \text{Key}_s}{\sqrt{d_k}}\right)}$$

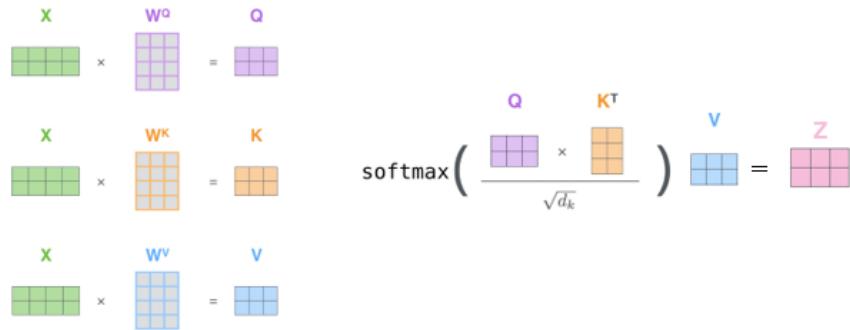


איו-5. מטריצת משקלים של המשפט "Find the collector rate on the incentive rate table" – מכך נראה קשר בין שני מילים חזק יותר – אך המשקל ביניהם גבוה יותר. כמו כן שיש גם משמעות לסדרת המשקל בין "collector" – "Find" – "Find" – "rate" – "on" – "the" – "incentive" – "rate" – "table" – "the" – "rate" – "table" – "rate" – "table".

כעת בעזרה משקלים אלו בונים ייצוג חדש לסדרה המקורית, על ידי הכפלתם בוקטור v

$$z_i = \sum_{j=1}^n w_{ij} v_j = \frac{\sum_{j=1}^n \exp(\text{Query}_i^T \cdot \text{Key}_j^T)}{\sum_{s=1}^n \exp(\text{Query}_i^T \cdot \text{Key}_s^T)} v_j$$

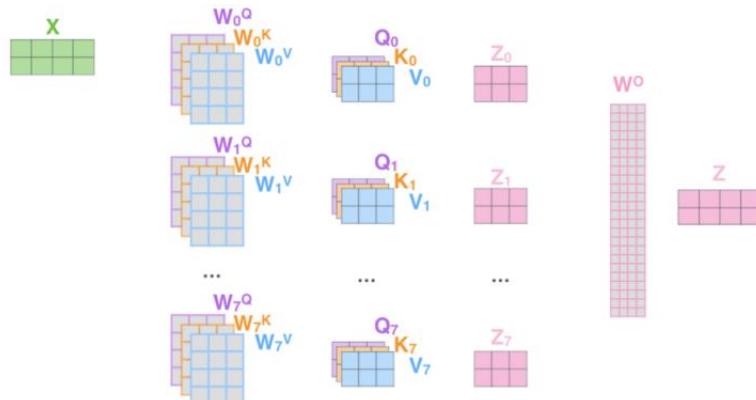
סדרה המתקבלת היא למעשה מעשה ייצוג חדש של סדרה המקורית, כאשר כל איבר- i ייצג איבר בסדרה המקורית יחד עם מידע על הקשרים בין ליבו' שאור איברי הסדרה את הסדרה המתקבלת ניתן להעביר ב- decoder -המכיל שכבות נוספות, ובכך לבצע כל מיני שימוש, כפי שיואר בהמשך.



איור 8.6 ביצוע Self-attention – ייצור מטריצות (ימין) ויחסים (שמאל) Query, Key, Value (מיין)

8.2.3 Multi Head Attention

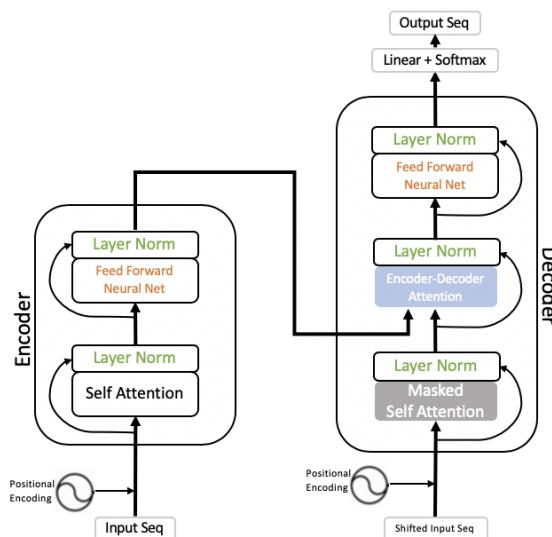
ניתך להשתמש במנגנון self-attention מספר פעמים במקביל – כל פעם מקבלים מטריצה של attention score (Query, Key, Value) (ימין) ובעזרת המחשבים את הייצוגים החדשניים של איברי הסדרה (attention score). כל מנגנון זה נקרא head – attention heads. בפועל הzahlכל איבר כניסה x_i יש כמה ייצוגים שונים z_{ir} , אותן ניתן להכפיל במטריצת משקלים w^o ולקבל את הייצוג המשוקל של אותו איבר באמצעות many attention heads.



איור 8.7 Self-attention with 8 heads

8.2.4 Transformer End to End

בازר מנגנון multi head attention – ארכיטקטורת בעובי Transformer – בניית positional encoding – סדרות המבוססות רק על attention ללא ריבוי זיכרון.



איור 8.8 Transformer

כפי שניתן לראות-abaior ה-*transformer* מורכב משני חלקים-decoder-encoder-ה-encoder מקבל סדרה מסוימת (לרוב אחריו שבעריה embedding מסוים) ומבצע עליה positional encoding. לאחר מכן הסדרה עובדת דרך residual block=attention self-*layer*($x + \text{attention}(x)$). ערך תוצאה זו מבצע עיבוד residual block=fully connected normalization (כפי שהסביר בפרק 5.1.4). לאחר מכן יש residual block=NFS, המכיל שכבות normalization ומשם י יצא הפלט לכליון ה-encoder.

לאחר השכבה הראשונה יש שכבה **multi head attention** שנוספתה הנקראת Encoder-Decoder Attention, כיוון שהיא אינטגרת בין encoder וה-decoder. השכבה מקבלת מה-encoder **Query**, **Key** ו-**Value**. **Query** מגיע מה-decoder ו-**Key** ו-**Value** הם מושגים דמיוניים בין איברים של אותה סדרה אלfabetic של decoder. בoutput של השכבה נוציאו איברים masked (ביצוג שלהם לאחר השכבה ה-encoder). סדרת הפלט (ביצוגם שלם לאחר ה-encoder) בין איברי סדרת הקלט (ביצוגם שלם לאחר השכבה ה-encoder). בשלב זה דומה מאוד ל-attention המוקורי, רק שהיציגים שהתקבלו לא נעזרים ברכיבי זיכרון. כאמור, המफלה **Q** מיצירת מטריצת משקלים שכל איבר בה אומנה בהתאם להיחס בין איבר בסדרה המקורית לבין איבר בסדרה הפלט. אף המטריצה זו מכפילים ב-**A** (מכפילים ב-**A** מתקבל מוצאים שהוא יציג חדש של איבר הפלט הבא. וקטור זה עובד בשכבה FC ובסoftmax, וכך מתקבל איבר הפלט.

נוקuds; המראה כיצד ניתן להשתמש ב-transformer-**DETR** שנקרא ממאמר שנקרא **Query** בשלב הראשון לוחכים כל פיקסל בתמונה ומשווים אותו לשאר הפיקסלים (זהו בעצם המכפל $K \cdot Q$). באופן הזה ניתן למצוא אזורים דומים ושונים בתמונה, כאשר דמיון וושני זה לאו דווקא פיקסל אחד ערכיים קרובים, אלא זה יכול להיות מושג שני אזורים שונים בפנים של אדם. לאחר מכן מיצרים ייצוג חדש לתמונה, בעזרה המשקלים והכפלתם ב-V=שלב זה למשה מאפשר לבצע זיהוי של אובייקטים, בלי לדעת מה הם אוטם אובייקטים. בשביל לביצוע סיווג כל אובייקט שזוהה-מבעירים את הייצוג החדש של התמונה ב-decoder, כאשר ה-**Query**=শমকনিস জড়েন্ট মিনে-লיבלים אפשרים, ומופשיים מבין כל ה-**decoder**=শমচালী ল'ই'ি'ত'র'ম'ন'া'হ'া' শহ'চ'ি' দ'ম'া' ক' **Query**

אם למשל יש תמונה גדולה ויש אזכור מסוים בו יש חתול, אז ה-decoder מוצא איפה החתול בתמונה, וה-decoder משווה את האזור הזה לכל מני חיוט אפנוריות= Query =שלא יהיה חתול, המכפלקה $K \cdot Q$ =תהייה קרובה ל-0, וה-decoder מזקירה שה- Query הנוכחי לא תואם לאובייקט שזויה. אך כאשר ה- Query =יהי חתול, ארכיוון ש- $K \cdot Q$ =תהייה אחד לשני=המכפלקה $K \cdot Q$ =תהייא לכתשה"צוג החדש= $\sum_{j=1}^n w_{ij} v_j$ =זקקיה דומה להחטול=צוג זה העובר בשכבה FC. ולאחר מכן ה-SoftMax יוסיף את התמונה זו כחתול.

8.2.5 Transformer Applications

ה-**transformer** הציג ביצועים מוגזם מושלמים, והוא היזה השראה להמונ "שומים הנשענים על **attention** בלבד. מלבד הרמה הגבוהה של הביצועים, תהליכי האימון של **transformer** מרשחות קובולוץית או רשותות רקורסיביות. כמו במקרים אחרים, גם **transformer** נition לבצע **transfer learning**, כלומר לחת **transformer** שאומן על משימה מסוימת, ולהתאים אותו למשימה חדשה דומה למשימת המקורית. בפועל לא כל הישומים משתמשים בכל ה-**transformer**, אלא בהתאם למשימה לוקחים חלקים מסוימים שלו ובונים מודל עבור משימה מסוימת. נביא מספר דוגמאות:

=Machine Translation מתרגמים משפטים בין שפות שונות הוא יישום טריאויאלי של ה-~~transformer~~ המשימה היא ללקחת משפט ולהוציא משפט בשפה אחרת, וזה נעשה באמצעות ייצוג המשפט המקורי באופן חדש בעזרת self-attention ולآخر מכון המרתנו בעזרת Encoder-Decoder Attention לשפה אחרת

ב-**Bidirectional Encoder Representations from Transformers** (BERT) שפה הוא פונקציה המתקבלת כתוצאה מחלוקת טקסט ומחזירה אפקת ההפוגות למילה הבאה על כל המילים במילון. השימוש בחci מוכר ואינטואיטיבי של מודל שפה הוא השלמה אוטומטית, שמצויה את המילה או המילים הכii סבירות בהינתן מה שהמשתמש הקליד עד כה. כאשר מבצע self-attention על משפטים, למעשה מתקבלים ייצוגים חדשים של הפה יחד עם ההקשרים בין המילים השונים. لكن ה-**encoder**-**transformer** יכול ליצור מודל שפה, אם מאמנים אותו בצורה מתאימה. המפתחים של BERT בפער encoder מקבלים כל מיני משפטים בשני כשי היכווניים – גם מההתחלת הסוף וגם מהסוף להתחלה, וכן הייצוגים שנלמדו קיילו קונקטסטם שלם יותר. בנוסף, הטעיאנו את המודול על משפטים בהם כל פעם באupon רנדומלי עשוים masking למלים מסוימות, ומטרת המודול הוא לחזות את המילים החסרות.

Generative Pre-Training (GPT) – מודל לחיזוי המילה הבאה במשפט. ניתן לנקח משפט שקטוע באמצעותו, ולבחון מהי המילה הבאה באמצעות ה-decoder בלבד. מכניםים משפט קצר לע-decoder ווררים על המונחים בו ובודקים את ההתאמנה שלהן למשפט הנתון, והמילה שהכי מתאימה נבחרת להיות המילה הבאה. המשפט הקטוע הוא למשרקה-Query, והוא-Query שנקנו הוא כל פעם מילה אחרת במילון, וכך בעזרת ה-attention בובוחנים איזה Key תאים בצורה הטובה ביותר ל-Query הנתון.

References

<https://arxiv.org/abs/1409.0473>

<https://arxiv.org/abs/1706.03762>

<https://towardsdatascience.com/day-1-2-attention-seq2seq-models-65df3f49e263>

<https://towardsdatascience.com/transformer-attention-is-all-you-need-1e455701fdd9>

<https://arxiv.org/abs/1810.04805>

9. Computer Vision

להכין (Region Based CNNs (R-CNN Family)

להכניס מטריקות

https://github.com/taldatech/ee046746-computer-vision/blob/spring20/ee046746_tut_05_deep_semantic_segmentation.ipynb

9.1 Object Detection

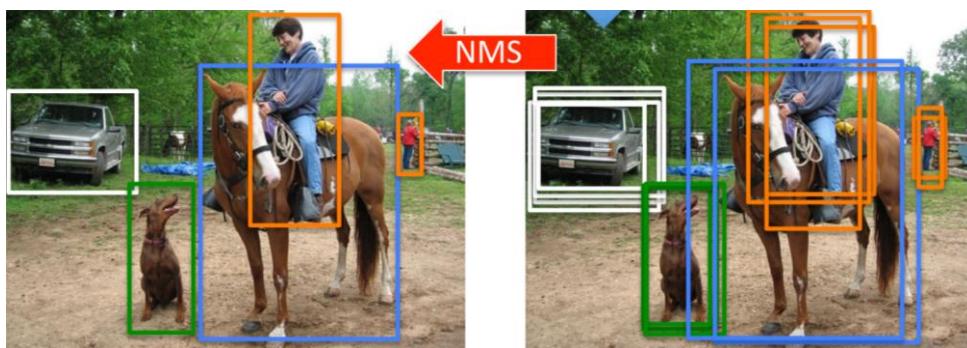
9.1.2 You Only Look Once (YOLO)

עד שנות-2016-2017 נקבעו מנגנון המאפשרים על מנת למשוך מושגים (comme à la famille R-CNN) הידועים דבשלהי-פְּרָוֹפְּזֶזְזָנִזְזָן-השלב הראשון יצר אלפי הצעות (proposals) למסגרות מלבדיו וההשווות אמצעיות אובייקטיבים, ולאחר מכן בזו אחר זו לשלב השני אשר דיבק את המסגרות וביצע סיווג לאובייקטיבים המוכלים בהן.

ארQUITECTURA ONLY YOU-הנגזרת מראשי התיבות של ONLY LOOK ONCE YOU, הוצגה על ידי ג'וזף רדמן ב-2016-ו והייתה הגלאי הראשון שモרכב משלב יחיד, ובו הרשות לנבאת את המסגרות וגם מסווגת את האובייקטיבים שבתוכן במתאף. בנוסף, ארQUITECTURA ONLY YOU מנטה פונקציית במייעוט פרמטרים ומוגות פעולות אРИטמטיות. היא אמנם משלמה על המבנה הרזה והאלגנטית שלא בדיקן נמור יותר, אך המהירות הגבוהה (שנובעת בעיקר מהיעדר האילוץ לעבד מסגרת יחידה בשלב השני בכל מעבר של הלולאה) הפכה את גישת השלב היחיד לאטרקטיבית מאוד, במיוחדם מעבדים קטנים כדוגמת מכשירים mobile. בעקבות עובודה זו פותחו גלאים רבים על בסיס שלב יחיד, כולל גרסאות מתקדמות יותר של ONLY YOU (הgrossה המתקדמת ביותר כיום היא 5).

NMS (Non-Maximum Suppression)

כמעט כל אלגוריתם של זיהוי אובייקטים מייצר מספר רב של מסגרות חשודות, כאשר רוב מינוחות ויש צורך לדלְל את מספרן. הסיבה לכך היא שמספר גודל של מסגרות נובע מ敞开 פעולה הגלאים=בזקנות התוכנות הלוקאליות של פועלות הקונבולוציה, מפת הפיצ'רים בМОץ הגלאי ניתנת לתיאור כטטריצת משਬצות כאשר כל משבצת שකולה לריבוע של הרבה פיקסלים בתמונה המקורית. רוב הגלאים פועלים בשיטת עוגנים (anchors), כאשר כל משבצת בМОץ הגלאי מנבאת מספר קבוע של מסגרות שעשוות להכיל אובייקט (למשל, -ב-2x2זט-המספר הוא-5, ובגרסאות יש המתקדמות יותר המספר הוא-3). השיטה זו יוצרת אלף מסגרות שרק מעוטות מהן הן ממשעות. בנוויכ-יש ריבוי של מסגרות דומות בסביבת כל אובייקט (למשל – 3x3זט-מנבאת יותר מ-7000 מסגרות לכל תמונה). אחות הדצכים הפולרייזולסן את אלף המסגרות ולהשאר רק את המשמעותי=נקראות=NMS=בשיטה זו מתבצעה השוואה בין זוגות של קופסאות מאותה המחלקה (למשל – חתול), ובמקרה שיש ביןיהם חפיפה גבוהה = מוחקים או המסגרות בעלת הוודאות הנמוכה ביותר ונשארים רק עם המסגרות בעלת רמת הוודאות הגבוהה. שיטה זו בגדנינו בחישוב סיבוכיות פרופורציונלית לריבוע מספר המסגרות) ואינה חלק מהמודל המתאים, אך-עם זאת הינה אינטואיטיבית יחסית למימוש, ומושם כר-מציאותי=מושג גלאים, כולל ב-2x2זט.



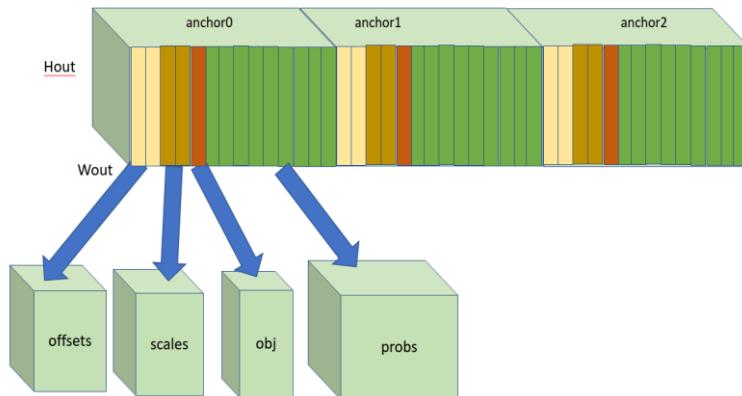
איור 9.1 אלגוריתם SNS

YOLO Head

כמו רוב הгалאים=0.5הינו מבוסס=עוגנים (anchor-based), שהיקתיבות מבניות קבועות ושונות זו מזו בוצרתן. כל עוגן מוקצה מקטע של פיצרים במפת המוצאנר הרשא כל הניבויים במקטע זהה מקודדים סטיות (offsets)

ביחס לממד הугון. כפי שניתן לראות באירור 9.2, הפיצרים של כל תא מרחבי בModelProperty המוצא מוחולקים למקטעים ערך פי העוגנים (שלושה עוגנים במרקחה זהה).

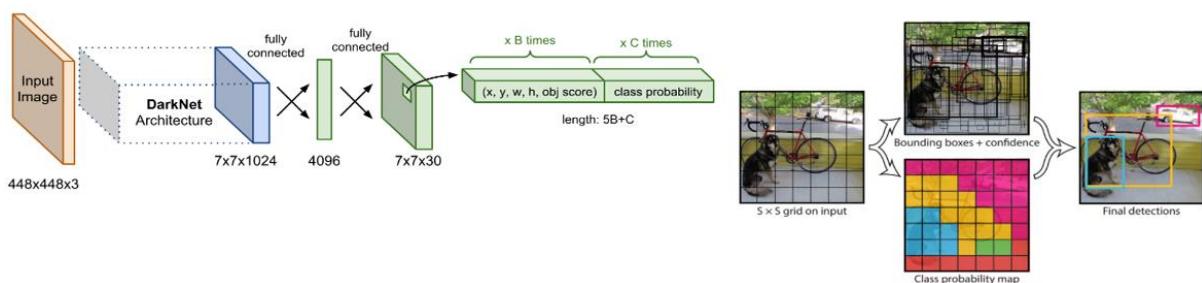
ניבוי הוא מסגרת שמרכזזה נמצאת בנקודות כלשהובשתה התא (השקלן לריבוע של מספר פיקסלים בתמונה המקורית). ההסתה המדוקיקת של מרכז המסגרת ביחס לתא ניתן על ידי שני הפיצרים הראשונים ברצף. לוג הממדים של הקופסה (ביחס לממד הугון) ניתן באמצעות שניה הפיצרים הבאים ברצף. הפיצר החמישי לומד את מידת ה-objectness, כפי שהסבירה לעיל. שאר הפיצרים ברצף של הугון הנו הם הסתבריות המותנה לכל מחלקה (אם אוסף הנתונים מכיל 80 מחלקות, יהיו 80 פיצרים כאלה). על מנת לקבל רמת ודאות סופית, יש להכפיל את מדד ה-objectness במדד הסתבריות המותנה לכל מחלקה.



איור 9.2 ראש YOLOv2

YOLOv1

מודול YOLOmbossים על גרסאות הנקראות Backbone. המשמשות לעיבוד פיצרים מתוך התמונה, ורראש detection המקבל את הפיצרים האלה וממתמן לייצר מהם ניבויים למסגרות סביב אובייקטים. המודל מחלק את התמונה לרשת בעלת 7×7 משבצות, כאשר כל משבצת מנבאת 5 מסגרות של אובייקטים בשיטת העוגנים, כאמור לעיל. כל ניבוי כולל אובייקט אחד: הסטטוס הeoroundness, אשר מרכז המסגרות במשבצת, הגובה והרוחב של המסגרת, ורמת ה-objectness. כפי שהסבירה לעיל-בנוסף, כל מסגרת מבצעת גם סיוג, כולם מנבאות אובייקטים שונים של השתייכות האובייקט לכל אחת מהמחלקות האפשריות=ההידוש באלגוריתם נעוץ בעובדה שחייבי המסגרות ויסווג לאובייקטים שנעשה במקביל, ולא באופן דו-שלבי. הרעיון הוואלה תייחולסיג האובייקט כעוד פיצר שהרשת מנסה לחזות בנוסף למיקום וגודל של המסגרת.



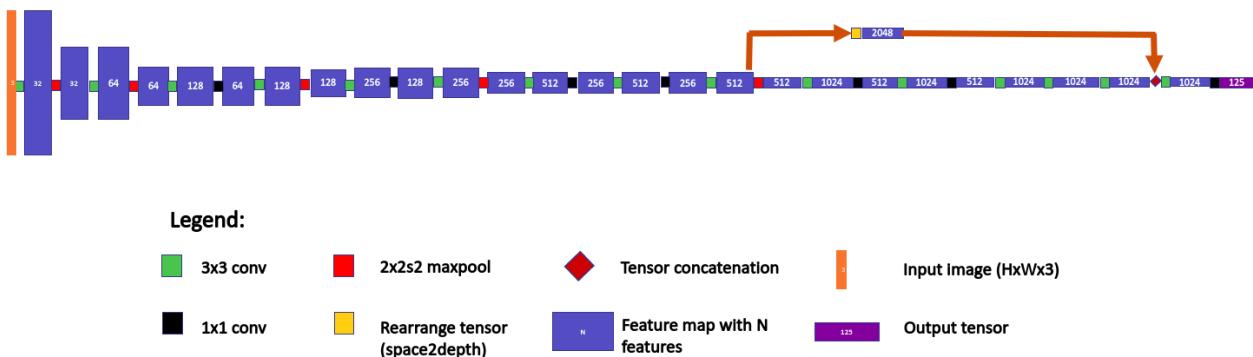
איור 9.3 ארכיטקטורת YOLOv1

הגרסת הראשוּה של ארכיטקטורת YOLO היא שכבת $=fully connected$ שהושירה בגרסאות הבאות. בנוסף, פונקציית LOSS וסדר התוכנות של המסגרות בmozaic הרשות השתנו, אבל הרעיון נותר זהה.

YOLOv2

גרסה זו משתמשת ברשת Backbone הנקראת Darknet19. ובמהלך קובץ ה- MAXPOOL המקבילות על מנת למצוא את הפיצרים. המודל כולל כולקציית כונולוציות (מלבד שכובות ה- MAXPOOL) המאפשרות על מנת למצוא את הפיצרים. ועוד מסלול עוקף בסופו הרשות המחזק את יכולת העיבוד. הכותב של המאמר המקורי, דוד מון, נגה לפתח גם גרסאות "Tiny" לכל מודל. הוריאנט

נמצא יוטר אך הוא מהיר מואוד (727 תמונות לשנייה לעומת 67 של מודול YOLOv2, על מעבד Titan X). מנגנון ליצין של YOLOv2 הוא המודול הראשון שאומן על תמונות במדדים משתנים, תהליך המשפר את דיק המודול.



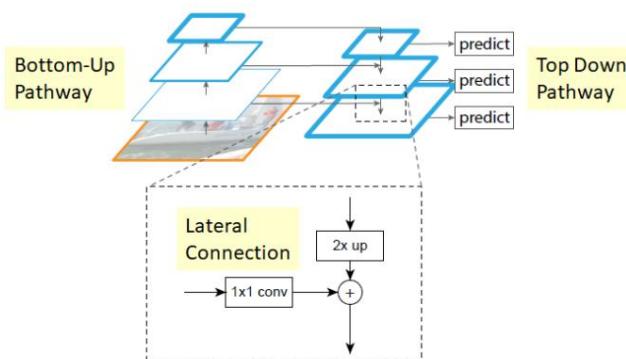
איור 9.4 ארכיטקטורת YOLOv2

YOLOv3

כל דור נוסף של YOLO^{v3} הציג חידושים ארכיטקטוניים שהגדילו את מורכבות החישוב וגודל המודל ושיפרו את ביצועיו. גרסה מספקת מבוססת על רשת Backbone שנקראת Darknet53. Backbone^{v3} גדול בהרבה מ-Backbone^{v2}, במיוחד, אף עליה משמה 53 כונבולוציות. כמו כן הרשת מכילה צוואר של ארכיטקטורת Feature Pyramid Network (FPN).

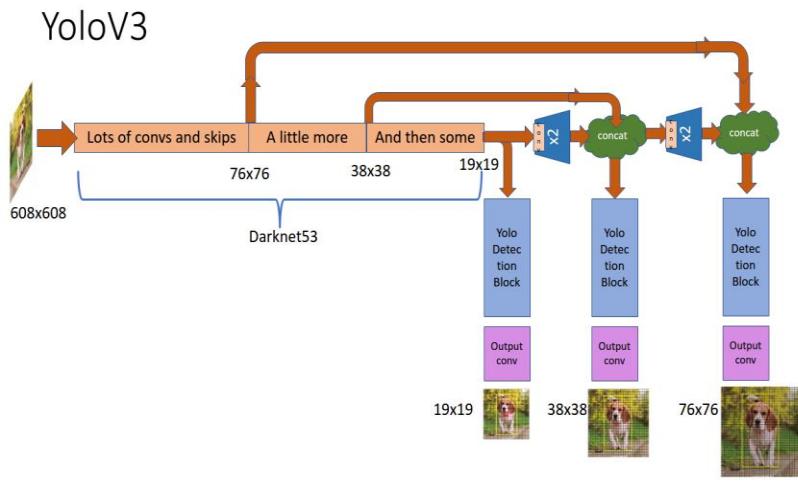
גilio, כאשר כל אחד מהם הוא בעל רוחולזיה שונה $38 \times 19, 38 \times 19, 38 \times 19, 38 \times 19, 38 \times 19$. ארכיטקטורת FPN היא תוספת בקצה Backbone, אשר מגדילה חזקה באופן הדרמטי את מפת הפיצרים=תוספה זו נועד להליכת מסלול Down-Path מבקבי למסלול Feed Forward-Branch. Backbone^{v3} שבוני למעשה בצורת Up-Path, כלומר מתקדים משלבים בפיזור עיבוד ברמה גבוהה יותר מהרלווזיה המרחבית של מפת הפיצרים=הולכת וקטנה. עזרת השימוש FPN^{v3} המודול לומד לנצל את המיטב בשניהם של מושגים במידע שטמונו במפת הפיצרים הגדולה, שהיא מאמן פחות מעובדת אך בעלת פיצרים ברוחולזיה מרחבית גבוהה, ובנוסף הוא מנצל גם אורך מידע ממפת הפיצרים הקטנה, שהיא יאמנת בעלת פיצרים ברוחולזיה מרחבית נוספת, אך עם זאת היא מעובדת יותר.

לאחר כל הגדלת-השכל מפת הפיצרים מתבצע חיבור בין התוצאה לביקום קדימה יותר במדדים זה-זה (מתוך Backbone^{v3}) – זאת בדומה לחיבורם העקיפים ברשת ResNet המסייעים להתקנות האימון=השכבות השונות של רשת FPN. מאפשרות לגליי למצוא מיקום מדויק יותר של האובייקט ברוחולזיות השונות, מה שמעניק לרשת יכולת להבחן אובייקטים קטנים בתמונה גדולה.



איור 9.5 ארכיטקטורת FPN (FPN) – המשלבת מסלול top down לאחר ה-*bottom up*.

לרأس המודול של YOLOv3 יש מספר ענפים detection, אשר כל אחד מהם פועל על מפת פיצרים ברוחולזיה שונה ובאופן טבעי מתמחה בגילוי אובייקטים בגודל שונה (הענף בעל הרוחולזיה הגבוהה מתמחה בגילוי אובייקטים קטנים).



איך 9.6 ארכיטקטורת 3vLOz

YOLOv4

רשות OLv4⁷ היא בעלת ראש צהה לזה של שתי הגרסאות הקודמות, אך ה-Backbone-⁸ שוניה ומורכב יותר. הוא נקרא CSPDarknet53=CSPNET=⁹Cross-Stage Partial Network. רשות זו מפצלת מפוזת פיצרים לטובת קובולוציה בחלקים ואיחוד מחדש. פיזול זה אפשרי, כמפורט במאמר המקורי, החלול טוב יותר של הגרדיינטים בשלב האימון. בנוסף, נעשה ברשות זו שימוש בפונקציית אקטיבציה הנקראת Mish¹⁰ (ולאReLU) כמגברגסאות הקודמות.

YOLOv5

רשת Backbone⁷ מוסיפה עוד שכבולים על רשות הפיזרים. Backbone⁷ של פני זו של הדור הקרוב, ומיצגה אופרטור חדש המארגן פיקסלים סמוכים בתמונה במרחב הפיזרים. אופרטור זה דואג לckerהכנית לושת היא לא בעומק הפיזרים מקובל (RGB), אלא 12 פיזרים, תוך הקטנת הממד המרחב. באופן זה הרשת מתאמת לעיבוד תמונות בהזולוציאיה גבוהה, ואף לזרות אובייקטים גדולים בקהלות רבה יותר, שכן שדה התמך (receptive field) של הקונבולוציות מכיל מידע משליט פסומה גדול יותר.

9.1.4 Spatial Pyramid Pooling (SPP-net)

Spatial Pyramid Pooling (SPP) – הינה שכבת pooling קבוצת נירונים, שמטרתה להסיר את האילוקשנרטוות (convolutional dropout) שדרוש שכבת הכניסה לרשף בגודל קבוע (כמו למשל רשף VGG) המכבלת רק תמונות בגודל 224×224 .

כולם רביעי – צילום מצלמות ניידות, מקצועיות, מצלמות אבטחה ואף מצלמות בטלוויזיה סלולריים ורחפנים. מצלמות שונות עשויות להוציא כפלט תמונות בגודלים שונים, מגוון סיבובות (למשל איקופת התמונה או מטרת המצלש). אם גרצה לבצע סיוג באוטה רשות ניירונים לכל אותן תמונות, נאלץ לבצע שלב נוסף בתחילת הדרכו – התאמת התמונה בכונסה לרשאה

נשים לב כי על מנת להתאים תמונה כלשהי לגודל מסוים, לרוב יוצרים חיתוך (crop), או שינוי גודל (resize/wrap). פעולות אלה עלולות לפגוע בזיהוי עקב שינוי היחס בתמונה (מתיחה/כיווץ), החסירה של פרט מסוים מהתמונה או שילוב של השניים. מוטיבציה זו היא שהובילה לשימוש בטכניקות [QQE](#)

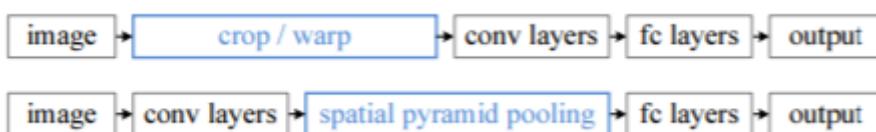
ניתן לראות דוגמא לשלינו גודל באמצעות מתיחה ולהחיתור באיזור הباء



איור 9.7: מימין - שינוי פרופורציה. משמאל – חיתוך

למעשה, הדרישה לתרומות קולט בגודל קבוע בכניסה לרשותת אלה אינה הכרחית, שכן שכבות ה- FC =בעומק הרשות הן השכבות שדורשות בכניסה להן קולט בגודל קבוע.

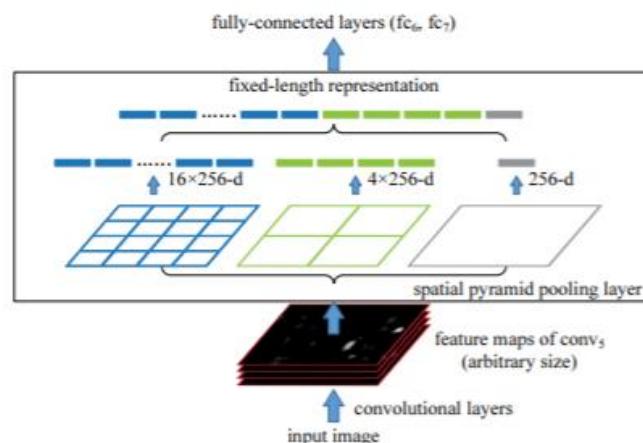
כעת, לאחר שהובאה המוטיבציה ליצור רשת זו, ניתן להבין את אופן פועלתה=על בסיס שכבת ה- $\text{PP}=\text{SPP}$ -Net. החידוש במבנה הרשות, הוא שבמקום שכבות שבכניסה לרשות (לפניהם שכבות ה- FC =תבצע max-pooling על ה= feature maps) תבצע max-pooling על ה= feature maps לאחר מציאת המאפיינים בשכבות ה- FC , כמוואר באיר הבא:



איור 9.8 – מבנה של רשת CNN קלאסית (למעלה) לעומת מבנה של רשת SPP-Net (למטה)

הרעין מאחריו שכבת ה- SPP =הוא חלוקה של הפלט של שכבות ה- FC , ביצוע max-pooling בכל חלק ורשור של התוצאות לווקטור שגודלו אחיד= B milim'ם אחרות, עברו כל ה= feature maps =המתקבלים לאחלה שכבות ה- FC , מייצרים שלושה וקטורים לכל feature

- וקטור בגודל 1 המתקיים באמצעות ביצוע max-pooling על כל הערכים באותו ה- feature
 - חלוקה של ה- feature =כל-4-תת-חלקים= 2×2 וביצוע max-pooling בכל חלק מתוכם. מתקיים וקטור בגודל 4.
 - חלוקה של ה- feature maps =כל-16 תת-חלקים (4×4) וביצוע max-pooling בכל חלק מתוכם. מתקיים וקטור בגודל 16.
 - שרשור כל הווקטוריים ייחד לוקטור בעל 21 ערכיהם
 - בסיום התהילה, תתקבל שכבה ביצוג אחד בגודל=21, בהכפלה במספר ה- features =שהתקבל משכבות ה- FC . שכבה זו "מחליפה" את שכבת ה- pooling הממוקמת לאחר שכבת ה- FC .
- האחרונה ותוצרה הוא היקלט לשכבת ה- FC . ניתן לראות את מבנה הרשות באיר הבא, בו התקבלו 256 features.



איור 9.9: ארכיטקטורת SPP-Net

9.2 Segmentation

אחד=האתגרים הכי משמעותית=בפועל הראייה המוחשבת הוואז'ויי אובייקט=בתמונה והבנת המתרחש בה אחת הטכניקות הקלאסיות=בתמונה עפ"מ שמייה זו הינה ביצוע סגנטציה, ככלומר, התאמת=אחסן כל פיקסל בתמונה=בתהליי=הסגנטציה=מבצעים=חלוקת/בידול=ב'עדים' שונים ב(bitmapה המצלמת באמצעות סיווג ברמה הפיקסל, ככלומר=על פיקסל בתמונה יסוווג וישיר למחילה מסוימת.

ישנפ=שימושים מגוונים באלגוריתמים של=סגנטציה=הפרדה של' עצמים מסוימים מהרקע שמאחוריהם=מציאת קשרים בין עצמים ועוזר=לודגמאות=תוכנות של שיחות וUIDה, skype, teams, zoom, מכוד', מאפשרות בחירת רקע עפ' שוני=בעבור המשתמש, כאשר מלבד הרקע הנבחר רק הגוף של המשתף מוצבבויד'. הפרדת גוף האדם מהרקע והטמעת רקע אחר מtbodyות אלגוריתמים של סגנטציה=דוגמאנויספה=ניתן להזות בתמונה אדם, כלב, וביניהם רצואה, ומכך ניתן להסיק שtopic בתמונה הוא אדם מחזיק כלב בעזרת רצואה. במקרה זה, הסגנטציה מעדת למצוא קשר בין עצמים ולהבין את המתרחש

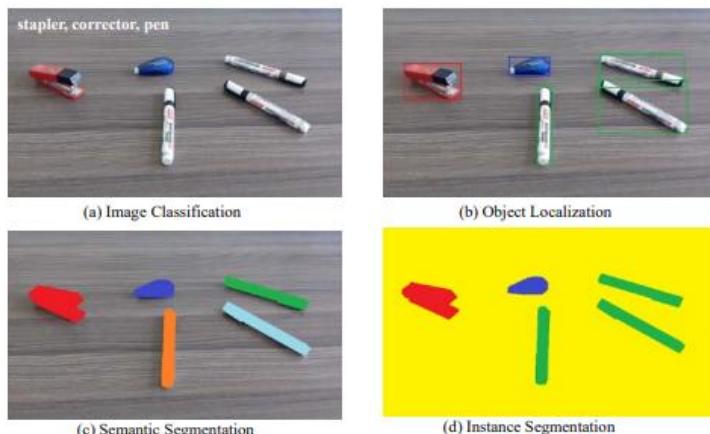
9.2.1 Semantic Segmentation vs. Instance Segmentation

קיימים שני סוגים עיקריים של סגנטציה:

Semantic segmentation (חלוקת סמנטי)=חלוקת של כל פיקסל בתמונה=למחלקה אליה העצם אותו הוא מייצג שיר. למשל, פיקסל יכול להיות משיר לכלי רכב, בן אדם, מבנה וכו'

Instance segmentation (חלוקת מופע) - חלוקה של פיקסל בתמונה למופיע של אותה מחלקה אליה העצם אותו הוא מייצג שייר=במקרה זה=בתמונה מופיע מספר כלי רכב, תבצע חלוקה של כל פיקסל לאיזה כלי רכב אותו פיקסל מייצג – מכונית 1, מכונית 2, אופנוע 1, משאית 1 וכו'.

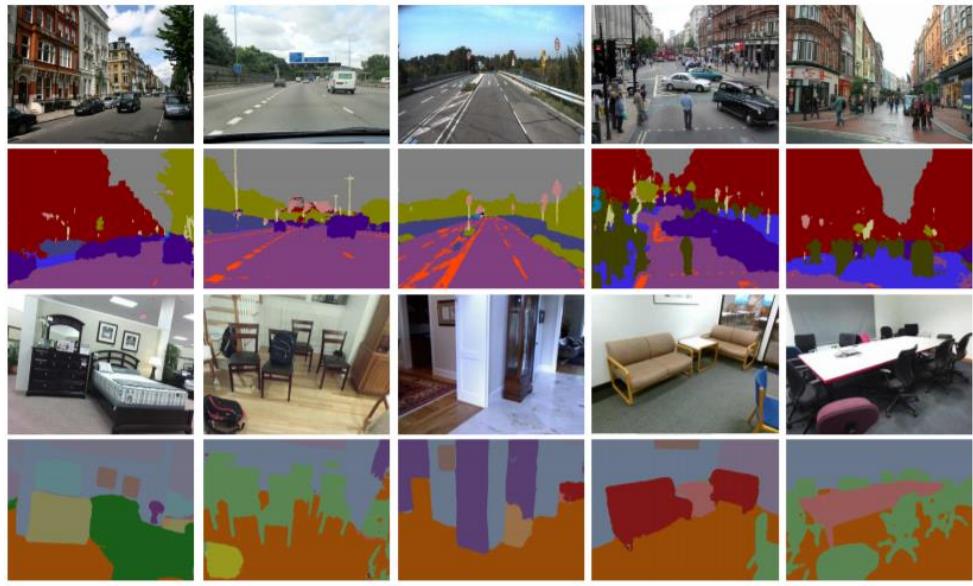
ההבדל העיקרי בין שני סוגים אלו הוא ברמת עומק המידע של פיקסל=המייפוי עשי לסוג את הפיקסל למחלקה כלשהי, או לעצם ספציפי=בתמונה. עומק המייפוי משליך גם על עלווה המייפוי. החלוקה הסמנטית מבצעת ישירות, בעוד שהחלוקת המופעת דורשת בנוסף ביצוע של זיהוי אובייקטים כדי לסוג מופיעים שונים של המחלקה



איך=7. משימות שונות תחת התחום של=סגנטציה Computer Vision. ניתן להבחן בהבדל שב=סגנטציה (התאמת כל פיקסל למחלקה מסוימת) לבין (התאמת כל פיקסל למופיע של מחלקה מסוימת)=Instance segmentation (המחלקה מסוימת) לבין

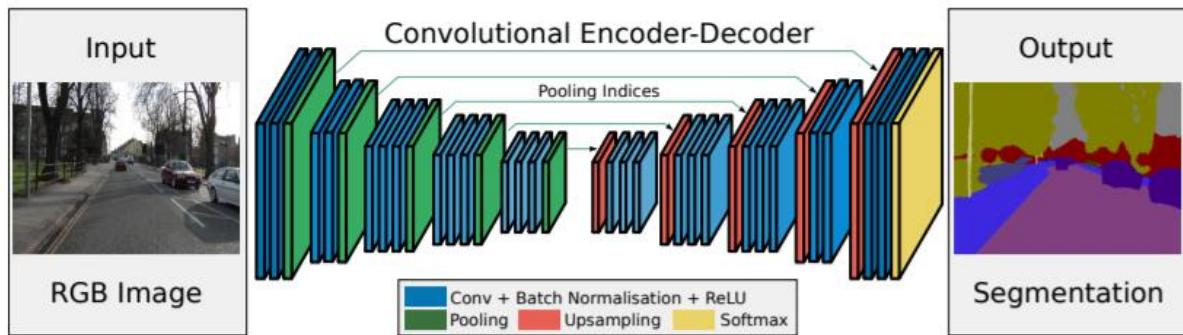
9.2.2 SegNet neural network

רשת SegNet הינה רשת קוונבולוציה عمוקה, שמטרתה לבצע חלוקה סמנטי (Semantic segmentation) לתמונה הקולט. בתחילת הרשת פותחה להבנה של=תמונה חז' (למשל כביש עם מכוניות ובצדדים בתים והולכי רגל) ותמונה פנים (למשל חדר עם מיטה וכיסאות)=הרשת נבנתה מtower מטרקללה=וילקה=בhbeti זיכרון וזמן חישוב, תוך שמירה על דיקוק מעשי.



איור 9.7 סיווג סמנטי בעזרת רשת SegNet עבור תמונות פנים וחוץ.

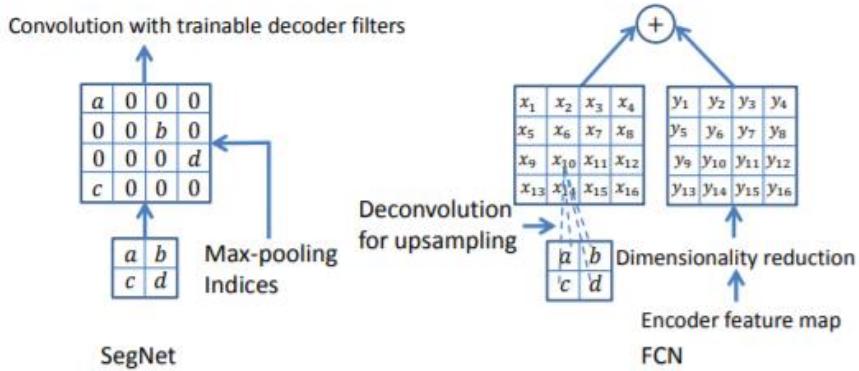
רשת בנויה כארQUITטורת מקודד-מפענה (encoder-decoder). המקודד מורכב מ-3 השכבות הראשונות של רשת VGG16 – שומותרפתלץ א-מפענה המאפיינים (feature maps). לכל שכבה קידוד יש שכבת פיענוח תואמת ומכאן שגם רשת המפענה מכילה 13 שכבות. מטרת המפענה היא לבצע sampling-sampling-קפל-יציר תמונה מאפיינים בגודל המקורי. את מוצא המפענה מעבירים דרך מסוֹגָס SoftMax, המתאים את המחלקה בעלת ההסתברות הגבוהה ביותר לכל פיקסל בנפרד.



איור 9.8 ארכיטקטורת רשת SegNet

המקודד מורכב משכבות קוונולוציה, אחריהן שכבות נרמול (batch normalization) ושכבות אקטיבציה מסוג ReLU (stride=2, subsampling=2, pooling-max-pooling). בעלת גודל חלוף 2×2 ומרוחה בגודל 2×2 .

ההידוש ברשת זו הוא אופן הפעולה של המפענה. בשונה מרשתות אחרות, בהקתקה ה-*Sampling*-*Upsampling*-*Softmax* ביצוע חישובים בעבור הפענוח, כמוואר באIOR הבא, הרעיון ברשת זו הוא שמירת מיקומי ערכי המקסימום הנבחרים מכל רבייה. רק הערכים שנבחרו כמקסימום יושלמו בתהילך ואילו הערכים יתרפסו.



איור 9.9 שכבה פענוח ברשת SegNet לעומת רשת FCN.

ארQUITטורה זו מביאה את הרשות לבייצועים טובי-פבהיבטי זמן חישוב, על חשבון פגיעה במסויים בבדיקה הרשות. למראות זאת, ביצוע הרשות מתאים לשימושים פרקטיים והפגעה בדיקת קטנה מואז.

בדומה למקודם, לאחר כל שכבה sampling-skip ב망, יופיעו שכבות קונבולוציה, שכבות נרמול ושובות אקטיבציה softmax=ReLU=את מוצא המפענה מעבירים דרך שכבה MaxSoftMax=המיצעת סיווג ברמת הפיקסל. מוצא הרשות, שהוא גם מוצא שכבת SoftMax, הינו מטרית הסתברותית, כאשר עבור כל פיקסל יש קטור באורך K, כאשר K הוא מספר המחלקות לסיווג. כנובן שההסיווג מתבצע בהתאם להסתברות הגבוהה ביותר המתאימה לכל פיקסל.

אימון הרשות בוצע על בסיס מידע אחד ש- $\frac{1}{600}$ =תמונה דרך צבעונית בגודל 480×360 , שנלקחו מבסיס CamVid road scene dataset. סט האימון הכיל 367 תמונות וסיט הבדיקה הכיל 233 תמונות הנתרות. המטריה הייתה להזדהות בתמונות אלה 1=מחלקות (דרך, בניין, מכונית, הולכי רגל וכדומה)=כל תמונה עברה נרמול מקומי לערך הניגודיות של תמונה הקלה לפני הכניסה לרשות. האימון בוצע בשיטת SGD, עם קצב למידה קבוע שערכף 0.001=ומומנטום שערכף 0.9=האימון נמשך עד שהשגיאת התכנסה לפני כל Epoch האימון עוזב וחולק ל-24 תמונות. פונקציית המחיר הינה mini-batch של 24.

לעתים, נדרש לבצע איזון-מחלקות (class balancing). מונח זה מתייחס לכך שבו קיימים שונים גודל בין כמות הפיקסליהם המשוכרים לכל מחלוקת, למשל כאשר קיימת הטיה מסויימת=Sכינה שבחובה מכילה בניינים / דרכי. במצב זה, יבוצע משקל מוחודש לפונקציית השגיאה, באמצעות תהליך "איזון התדריות החזיניות" (median frequency balancing). התהליך מושך מחדש את המחלקות בפונקציית המחיר, באופן יחסית לחזין של תדריות הופעות המחלקות בכל סט האימון, תוך חלוקה בתדריות הופעת המחלוקת:

$$\alpha_c = \frac{\text{median freq}}{\text{freq}(c)}$$

משקל זה משנה את היחסים בפונקציית המחיר כך שה торה של כל המחלקות לפונקציית המחיר תהיה שווה. לכן, הוא מעניק למחלקות הגדלות יותר משקל נמוך יותר ולמחלקות הקטנות משקל גבוה יותר.

9.2.3 Atrous Convolutions (Dilated Convolutions)

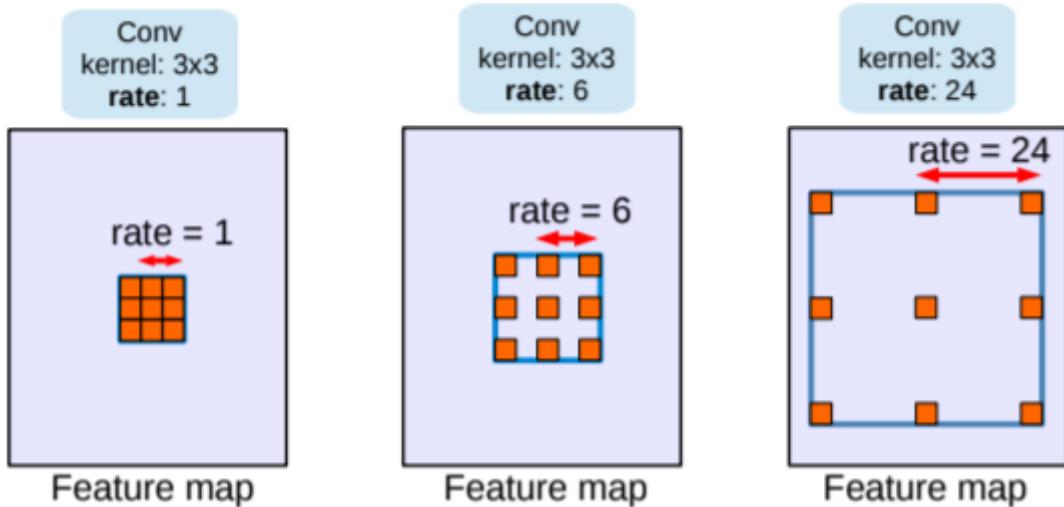
המונח=Atrous convolution=מוכרו בשפה הצרפתית=trous=אמ"ר "מעיים". לקניתן לתרגם את המונח=Atrous convolution=קונבולוציה מחוררת=ובמשמעותו מעט יותר מתאימה=קונבולוציה מרוחקת (או בשפה אחרת=dilated convolution=– קונבולוציה מוחשבת).

בטכניקת קונבולוציה זו, יש שימוש בפרמטר נוסף – dilation rate פתרון זה מסמן את המרווח בין כל איבר בגרעיך הקונבולוציה (הרחבת על פרמטר זה בפרק 9.1.2). נוסחת הקונבולוציה עשויה במקרה היחד עם פרמטר ההתרחבות r ניתנת לתיאור באמצעות הבאה:

$$y[i] = \sum_{k=1}^K x[i + r \cdot k]w[k]$$

עבורה = 1 = אם התקבלת הקונבולוציה הרגילה, ואשף-ישנה dilation rate = 1 > גְּמַתְקָלְתָה^{dilation} קונבולוציה מרוחקת בפקטו-ז-יתרונה של קונבולוציה-געז בעבורו אתו קרנל ובעור אותה כמות חישובים, מרחיבים את ה-field of view (FoV) של הקונבולוציה

ניתן לראות את הרחבת field of view באירור הבא:



איו-ה-9: קונבולוציה מרוחקת דפmdית, עם קרNEL בגודל 3×3 ו-ופרטור התרחבות $= r$. בהתאם לכל פרמטר מתקבלה-field-of-view בגודל שונה – $3 \times 16, 16 \times 49, 49 \times 49$ בהתאם.

9.3 Face Recognition and Pose Estimation

9.3.1 Face Recognition

אחד מהיישומים החשובים בראיה ממוחשבת הינו זיהוי פנים, כאשר ניתן לחלק משימה זו לשולש שלבים:

1. Detection – מציאת הפרצופים בתמונה.
2. Embedding – מיפוי כל פרצוף למרחב חדש, בו המאפיינים שאינם קשורים לתיאור הפנים (למשל – זווית, מיקום, תארורה וכדו') אינם משפיעים על הייצוג.
3. Searching – חיפוש במאגר של תמונות למציאת תמונה פנים הקרובה לתמונה הפנים שהולצתה מהתמונה המקורית.

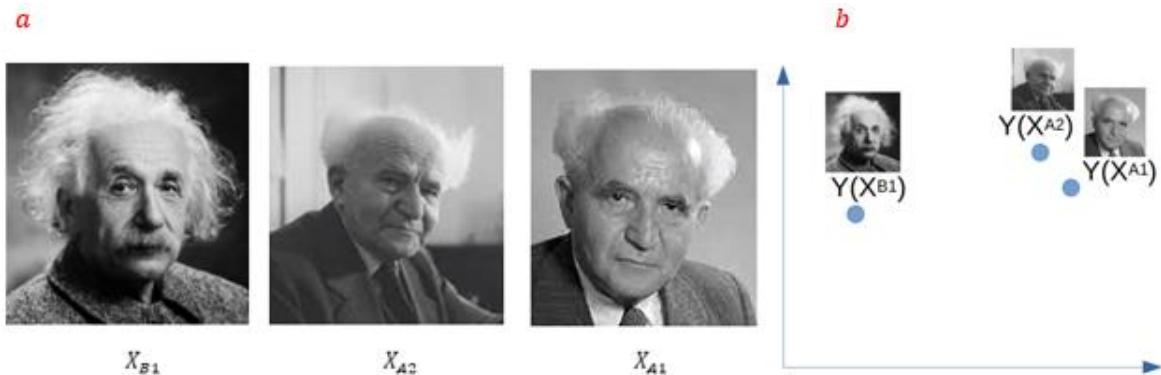
גישה פשוטנית, כמו למשל בניית מסוג המכיל מספר יציאות כמספר הפנים אותן רציחל-זהות, הינה בעייתית מושתת סיבות עיקריות: ראשית יש צורך באלפי דוגמאות לכל אדם (שלא ניתן בהכרח להציג). כמו כן-נוצרת למד אט' המערכת החדש בכל פעם שורצים להוסף מישחו חדש. כדי להתגבר על בעיות אלו מוצעים "למידת מטריקה" (metric learning) בה מזקקיפ-מאפייניב-של פפס ויוצרים קטור יחסית קצר, למשל באורך=128, המכיל את האלמנטים המרכזיים בתמונה הפנים. כתע נפרט את שלושת השלבים:

1. מציאת פנים:

כדי למצוא פרצופים בתמונה ניתן להשתמש ברשותות המבצעות detection, כפי שתואר בפרק 9.1. שיטה מקובלת למשימה זו הינה-סאוז, המבוססת על חלוקת התמונה למשבצות, כאשר עברו כל משבצת בוחנים האם יש בה אובייקט מסוים, מהו אותו אובייקט, ומה ה-*bounding box* שלו.
2. תיאור פנים.

כאמור, המשימה בתיאור פנים נעשית בעזרת *embedding*, metric learning, כאשר הרעיון הוא לא רק פנים לוקטור שאינו מושפע ממאפייניב-שלא שייכים באופק מהותי' לפנים הספציפיות האלה, כגון זווית צילום, רמת תוארה וכדו'. בכך לעשות זאת יש לבנות רשת המקבלת פנים של בנאים ומחזירה וקטור, כאשר הדרישת היא שעבור שתי תמונות של אותו אדם יתקבלו וקטורים מאד דומים, ובעור פרצופים של אנשים שונים יתקבלו וקטורים שונים. למעשה, פונקציית-h-soss-תתקבל בכל פעמי-*minibatch*, ותעניש בהתאם לקרבה בין וקטורים של אנשים שונים וריחוק בין וקטורים של אותו אדם.

cutת נניח שיש לנו קלט X =המכיל אוסף פרצופים. כל איש יסמן באות אחרת= A, B, C ו-תמונהות שונות של אותו אדם יסומנו על ידיאות ומספר, כך שלמשתל X_{A1} -זהי התמונה הראשונה של אדם=A בבסט הקלט X , ומובן ש- X_{A1} = X -הן שתי תמונהות של אותו אדם. באופן גרפי, בדו-ממד=ניתן לתאר זאת כך (בפועל הוקטורים המציגים פניהם יהיו במדד גובה יותר):



איור 9.10 (a) דוגמאות מסט הפרצופים X . (b) איך נרצה שהדата יומפה לממד חדש Y .

כאמור, נרצה לבנות פונקציית loss שמעודדת קירבה ב- X_{A1} - X_{A2} , וריחוק ב- X_{A1} - X_{B1} . פונקציית loss מרכיבת משני איברים, המודדים מרחק אוקלידי בין וקטורים שונים:

$$L = \sum_X \|Y(X^{Ai}) - Y(X^{Aj})\| - \|Y(X^{Ai}) - Y(X^{Bj})\|$$

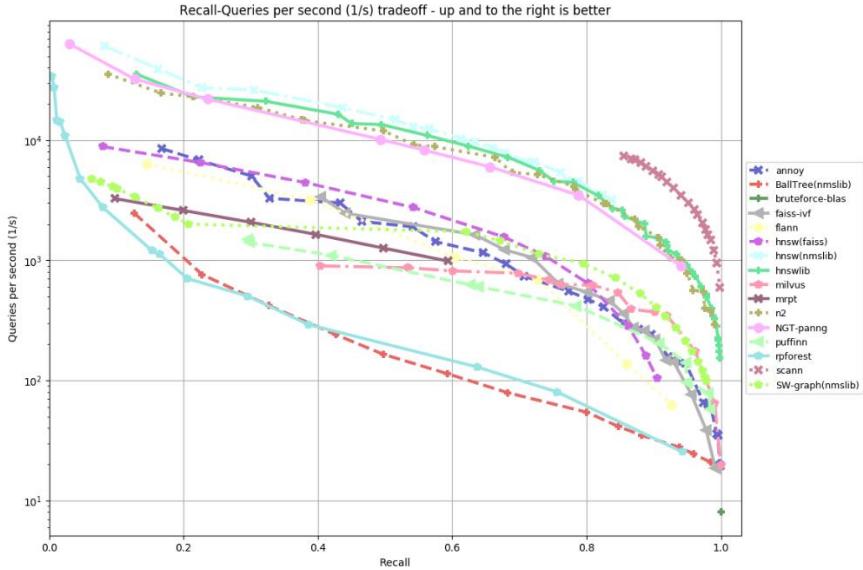
כאשר האיבר הראשון ינסה להביא למינימום וקטורים של אותו אדם, והאיבר השני ינסה להביא למינימום וקטורים של פרצופים שאינם שייכים לאותו אדם. כיוון שנרצה להימנע מתקבל ערכיהם שליליים, נוסיף פונקציית מקסימום בנוסף, ניתן 'להרחיק' תוצאות של פרצופים שונים על ידי הוספה קבוצה, כך שהפרש בין המרחק של פרצופים של אנשים שונים לבין המרחק של פרצופים של אותו איש יהיה לפחות k :

$$L = \sum_X \max(\|Y(X^{Ai}) - Y(X^{Aj})\| - \|Y(X^{Ai}) - Y(X^{Bj})\| + k, 0)$$

loss נקרא triplet loss, כיוון שיש לו שלושה איברי קלט=שתי תמונהות של אותו אדם ואחת של מישחו אחריו. הפלט של הרשת הנלמדת צריך להיות וקטור המאפיין פנים של אדם, ומטרת הרשת היא למפות פרצופים שונים של אותו אדם לווקטורים דומים עד כמה שניתן, ואילו פרצופים של אנשים שונים יקבלו וקטורים רחוקים זה מזו.

3. מציאת האדפט

בשלב הקודם, בו ה被执行 האימון, יצרנו למשה מאגר של פרצופים במרחב חדש. cutת כsigmoid פרצוף חדש, כל שנוטר זה למפות אותו למרחב החדש, ולאחר מכן המשמש בניתוחים קלאסיים של machine learning, כמו למשל חישוב שכן קרוב (כפי שהסביר ב2.1.3). שיטות אלו יכולות להיות איטיות עבור מאגרים המכילים מילוני וקטורים, וישן שיטות חישוב מהירות יותר (ובדרך כלל המהירות באහען חשבון הדיקט)=בעזרת ההשיטה המובייל-הכרעה(SCANN) ניתך להגיע לכמה מאות חיפושים שלמים בשניה (הchipush ב-000 ממינים מתוך מאגר של 10000 דוגמות=



איור 11.9-השוואת ביצועים של שיטות חיפוש שונות. עבור פרצוף נתון, מהפשים עברו וקטור תואם במדד החדש המכיל ייצוג וקטור של הפרצופים הידועים. בכל שיטה יש טרידאוף בין מהירות החיפוש לבין הדיוק

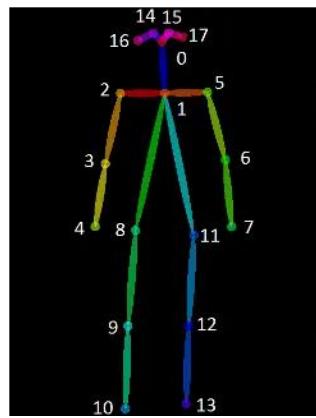
מלבד זיהוי ויזוג פנים, יש גם שיטות של מציאת אלמנטים של פנים הוכולות אף, עיניים וכו'. אחת השיטות המקבילות המשמשת בשערוך הצורה של פנים אנושיות, וניסיון למצוא את איברי הפקם לפי הצורה הסטנדרטית. בשיטה זו ראשית מבצעים יישור של הפנים והתאמאה לסקירה אונשות (על פי מרחק בין האיברים השונים בפנים), ולאחר מכן מטילים 68 נקודות ענייניות מרכזיות על התמונה המיישרת, מתוך ניסיון להתאים בין הצורה הידועה לבין התמונה המבוקשת.



איור 12.9-זיהוי אזוריים בפנים של אדם על ידי התאמת פנים לסקירה אונשית והשוואה למבנה של פנים המכיל 68 נקודות מרכזיות

9.3.2 Pose Estimation

ישום פופולרי נוסף של אלגוריתמים השיכיפ-לראיה ממוחשבותהינו קביעת תנוחה של אדם-האם הוא עומד או יושב, מה התנוחה שלו, באיזה זווית האברים נמצאים וכו'. ניתן להשתמש בניתוח התנוחה עבור מגוון תחומיים= ספורט, פיזiotרפיה, משחקים שונים ועוד. לרוב, תנוחה מיוצגת על ידי המיקומים של חלק גוף עיקריים כגון ראש, כתפיים, מפרקים וכו'. ישם כמה סטנדרטים מקובלים, למשל COCO, COCO=הנוחה-מיוצגת בעזרת מערכת של 17 נקודות (בדו-מיד).



איור 9.13 מיפוי תנוחה ל-17 נקודות מרכזיות.

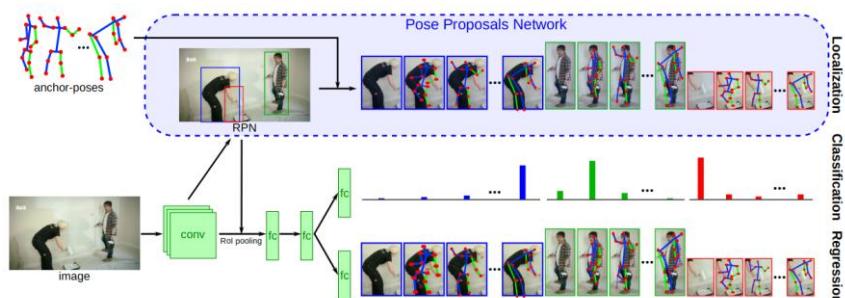
שאר הפורטטים דומים; מוסיפים עוד מידע (למשל מיקום הפה), משתמשים בתלת ממד במקום בדו ממד-משתמשים בסידור אחר וכך. כאשר רוצים לאוסף נתונים על מנח גוף שונים, ניתן לשים חישנים על אותן נקודות מרכזיות וככה לקבל מידע על מנה הגוף לאורך זמן

רשות לצורך קביעת תנוחה יcalelate במקורה כללי, בו יש מספר אנשים בתמונה, או במקרה הפרטי של גוף יחיד. במקרה השנמכובן יותר פשוט, מכיוון שישנו פלט יחיד, אותו ניתן להזוז באמצעות גרגסיה (למשל=COCO=בדו-מד). במקרה זה ניתן לעשות למידה סטנדרטית לחולstein של בעיית גרגסיה בעלי 34 יציאות.

ישנו מספר גישות כיצד להכילה את הרשתק שתוכל לטפל גם במקרה הכללי בו יש יותר גוף אחד בתמונה. באופק נאיבי ניתן לבצע תהליכי מוקדים של מציאת כל האנשים בתמונה, ואז להפעיל על כל אחד מהם בנפרד את הרשת שמבצעת גרגסיה, כפי שתואר לעיל. שיטה נוספת פועלת בכיוון הפוך=ראשית כל הרשת מוצאת את כל האיברים בתמונה, ולאחר מכן משיכת אותן לאנשים שונים. השיטה השנייה נקראת "מלמטה למעלה" (top-bottom), כיון שקדם כל היא מוצאת את הפרטים ולאחר מכן מכלילה אותם. גישה זו עיליה למקורה בו יש הרבה חפיפה בין האנשים בתמונה, כיון שאין לה צורך לבצע תהליכי מוקדים של הפרדת האנשים. השיטה הראשונה, הנקראת "מלמטה למטה" (bottom-top), תהיה פשוטה יותר מאשר המקרה השני בין האנשים בתמונה וכל אחד מהם נמצא באזור שונה בתמונה, כיון שאין צורך לשירות איברים לאנשים.

רשת פופולרית=לקביעת תנוחה=נקראת=Multi-person pose estimation=המודckaה=למערך=משתי תות-רשתות ופועלת בשיטות top-bottom. הרשת שאחריה על שיר חולקי גוף לאדם מסוים, נקראת=part affinity fields (PAF), והרעיון שלו הוא לייצג כל איבר כודה וקטורי. ביצוג זה הווקטוריהם השונים מצביעים לכך אין איבר הגוף ה'בא בתו'ר' (למשל זרוע מצביעת ליד), וככה ניתן לשירות איברים שונים אחד לשני, ואת כל ייחד לגוף מסוים.

רשת פופולרית אחרת, הפעולה בגישת bottom-top, נקראת=LCR-NET, והוא מבוססת על רעיון של 'מיקום-סיוואג' רגסיה' (Localization-Classification-Regression). בשלב הראשוני תות-רשת המיצרת עוגנים עבור אנשים, כלומר=אזריפ=בهم הרשת חושבת שנמצא באדם, ולאחר מכן הרשת משערכת את התנוחות שלהם=בשלב השפה מתבצע=השלב העוגנים, כלומר כל עוגן מקבל ציון המיציג את טיב השערור של העוגן והתנוחה של האדם הנמצא בתוכו=השלב השלים=מלטש את העוגנים ומשקל את השערור הסופי בעזרת מיצוע של הרבה עוגנים. שלושת השלבים משתמשים ברשת קוונבולוציה משותפת, כמתואר באייר.



איור 9.14 Localization-Classification-Regression 9.14

9.4 Few-Shot Learning

יכולת הצלחתם של אלגוריתמי למידה עמוקה נשענת על כמות ואיכות הדата לאימון. עבור מושלמת סיווג תמונות (Image Classification), נדרש שבעור כל קטגורית סיווג תהיה כמות גדולה של תמונות מסווגות (עם הבדלי רקעים, בהיות, זווית וכו'), ובנוסף יש צורך בכמות דומה של דוגמאות בכל קטגוריות הסיווג. חוסר איזון בין כמות התמונות בקטגוריות השונות משפייע על יכולת הלמידה של האלגוריתם את הקטגוריות השונות ועל כן עלול ליצור הטיה בתוצאות הסיווג לטובת הקטגוריות להן יש יותר דוגמאות בפרק זה מאשר בשיטות כיצד ניתן להתמודד עם מצבים בהם הדата אינה מואוזן.

9.4.1 The Problem

התחום של למידה ממיוט דוגמאות (Few-Shot Learning) נוצר על מנת להתמודד עם מצב של חוסר איזון קיצוני בין כמות הדוגמאות של כל קטגוריה לאימון הרשות-באופן פורמלי- K -מאות קטגוריות הנקראות קטגוריות בסיסicas (base classes); עבורף-יש כמות גדולה של דוגמאות-ובנוסף יש נס-קטגוריות חדשניות (novel classes); עבורף-יש כמות קטנה מאוד של דוגמאות-בכדי להגיד את היחס, משתמשים בשני פרמטרים= K -פרמטרי= n -פרמטרי- K -המייצג את מספר הקטגוריות כלומר מספר הדוגמאות הקיימות בסיס האימון מכל קטגוריה חדשה, ופרמטרו= n -ומציין את מספר הקטגוריות החדשניות הקיימות סך הכל= $K+n$ -וגדרת על ידה, ולמשל "k-shot"= k -way learning, ו"one-shot"= 1 -way learning. מצב בו יש חמישה קטגוריות חדשות-ומכל אחת מהן יש רק דוגמא אחת לאימון הרשות. בכלל, בקטגוריות הבסיס תהייה כמות גדולה של דוגמאות. למשל בסיס התמונות האופייני לביעות אל-ImageNet, יש 600 דוגמאות לכל קטגורית בסיס ולרוב 5-10 דוגמאות עבור הקטגוריות החדשניות.

האתגר בלמידה ממיוט דוגמאות נובע מה הצורך להכניס לרשות כמות ידועת נוספת הנרחב הקים, תוך הימנעות מ- $overfitting$ -הצואנה מכמות הפרמטרים הגדולה של הרשות לעומת-הכמות המועטה של הדעתה. לכן גישה נאיבית כמו אימון מחדש של רשות על מעט דוגמאות נוספת ליצור הטיה בתוצאות.

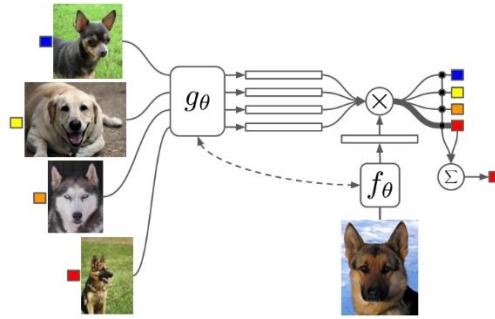
יש לציין כי בכל בעית הלמידה ממיוט דוגמאות-אמנם חסר דעתה עבור האימון, אך השאיפה היא להצליח באופן זה-הגביה-בכל קטגוריות בשלב המבחן, בו לא יהיה חוסר איזון. لكن בעית אלו לוונטיות לשימושים רבים כמו: זיהוי חיויות נדירות באופן זהה לחיות יותר נפוצות, מערכת זיהוי טילים שצריכה להתמודד גם עם איים נדירים יותר (ニヤック לחשב למשל על פצת אוטום), מערכות זיהוי פנים שצריכות לעבוד טוב עבור כל אדם ללא תלות בדתת ששה קיימת באימון הרשות=

פרק זה נתאר את שלוש הגישות העיקריות לפתרון בעית למידה ממיוט דוגמאות. עבור כל גישה נציג את האלגוריתמים המשמעותיים ביותר שנקטו במקרה זו. לאחרונה, מפותחים יותר וייתר אלגוריתמי למידה ממיוט דוגמאות שימושיים יחד ריעונות השאבים מספר גישות יחד אך נשלימים על האלגוריתמים המשמעותיים מהעברית. לבסוף, נציג את התחום של Zero-Shot Learning, כלומר יכולת למידה של קטgorיה חדשה כאשר לא קיימת אף דוגמא שליה לאימון=

9.4.2 Metric Learning

שיטות להתמודדות עם למידה ממיוט דוגמאות הנוקטות בגישת למידת מטריקה, שואפות לייצג את הדוגמאות פוקטוריים של- MF -פינים במרחב רב-ממד- \mathbb{R}^d שניתן יהיה למצאו בקהלות את השיר הקטגוריה של- DG -החדשה, גם אם היא תהיה מקטגוריה חדשה. שיטות אלו מבוססות על עיקנון הגדלת המרחק בין יצוגים וקטורים של דוגמאות מקטגוריות שונות (inter-class dissimilarity), בד בבד שמירה על מרחק קטן בין הייצוגה והקתגוריה של דוגמאות מאותה הקטגוריה (intra-class similarity).

התקומות משמעותית של שיטות אלה הוצגהו במאמר Matching Networks for One Shot Learning ב-2016. שיטה זו משתמשת בזיכרון שהגישה אל- KN -השיכת-באמצעות מנגנון-חישוב (Attentional Network), על מנת לחשב את ההסתברות של דוגמא להיות שייכת לכל קטגוריה, בדומה לשיטות השכן הקרוב (Nearest Neighbors).

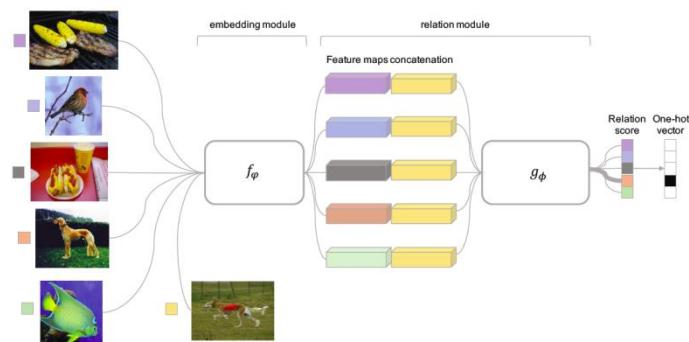


איור 9.15 אילוסטרציה של שיטת Matching Networks

ההידוש המשמעותי בשיטת **Matching Networks** נועד-

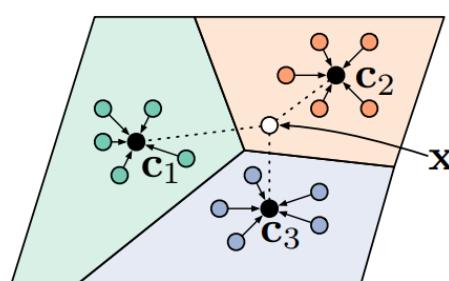
Episodes (Episodes). בשיטה זו האימון מכיל כמות של שימושים, כאשר כל שימוש היא למעשה מבחן של הדטה שבו יש קטגוריות מסוימות ושהן חדשות בסיס. על ידי דוגמאות רבות ויצירת שימושים מלאכותיים כאלה, בהן בכל פעם נלקחות קטגוריות אחרות ליצג את החדשניות, מתבצע אימון המתאים לבעה של מיעוט דוגמאות. שיטת אימון באפיוזודות הפכה לנפוצה ביותר בלמידה ממינית דוגמאות, גם בגין הatrixה שונראת בהמשך

שיטות רבות מtabssot על הרעיון של מאמר זה. למשל שיטת Relation Network מושרשת וקטורי מאפיינים של דוגמת מבחן בין כל דוגמא של קטגוריות האימון. אלו ננסים למודל המשערך ממד דמיון עזרתו ניתן לסואג את דוגמת המבחן



איור 9.16 Learning to Compare: Relation Network for Few-Shot Learning

שיטה נוספת המשמעותית נוספת הנקנתה בגין למידת מטריקה קראט-Prototypical Networks. בגין זו כל קבוצת דוגמאות של קטgorיה מסוימת במרחב וקטורי המאפיינים מקבלת נקודת אבטיפוס אופיינית המוחשבת על ידה המומצע של הדוגמאות בקטgorיה זו. בכר מחשבים מסווג לנארה המפריד בין הקטגוריות. בעת המבחן נסואג דוגמאות חדשות על סמך מרחק אוקלידי מנקודות האבטיפוס



איור 9.17 Prototypical Networks for Few-Shot Learning

בטבלה הבאה ניתן לראות השוואת ביצועים של שיטות למידת המטריקה שהזכו על הקטגוריות החדשניות-יש להציג כל השיטות מגוונות לאחיזה דיק נמכרים משמעותית מהחיזיון הדיק המדוחים במקרים של איזון בין כמה דוגמאות בקטgorיות השונות (לרוב מעל 90% דיק).

Method	5-way 1-Shot	5-way 5-Shot
Matching Networks	46.6%	60.0%
Prototypical Networks	49.42%	68.20%
Learning To Compare	50.44%	65.32%

איור 9.18 השוואת ביצועי דיוק של שיטות למידת מטריקה על קטגוריות חדשות עבור mini-ImageNet.

9.4.3 Meta-Learning (Learning-to-Learn)

גישה שנייה להתמודדות עם מיעוט דוגמאות וחוסר איזון בין הקטגוריות נקראת מטא-למידה (או: למדוד איך למדוד). באופן כללי בלמידה מכונה, כאשר מדובר על מטא-למידה, מתכוונים לרשת שלומדת על סך התוצאות של רשות אחרת. בלמידה ממיעוט דוגמאות הרוין הוא שהרשת תלמיד בעצמה-air להתמודד עם מיעוט הדאטה על ידי ערך הפרמטרים של הלא-אופטימיזציה-של-בעיה של סיווג ממיעוט דוגמאות. לשם כך משתמשים באפיוזות של שימושות לmeta-למידה.

שיטה חשובה בגישה זו היא MAML (Model-Agnostic Meta-Learning). בשיטה זו, שאינה מיועדת ספציפית לסיווג תמונות ממיעוט דוגמאות, בעזרת מספר צעדים מיטים בכיוון הגראדיינט ניתן למדוד את הרשת התאימה מהיר(*fast adaptation*)=למשימה חדשה. כאמור, כל משימה באימון היא אפיוזה שבה קטגוריות מסוימות נבחרות רנדומלית לדמות את הקטגוריות החדשוויות. בכל משלדים פרמטרים של המודל האגנוטטי כך שעדכונו בכיוון הגראדיינט יוביל להתאמה למשימה החדש. הכותבים מצינים שמנקודת מבט של מערכות דינמיות, ניתן להתבונן על תהליך הלמידה שלהם כenza שמנקסם את ריגשות פונקציית המחיר של משימות חדשות ביחס לפרמטרים. כאשר הריגשות גבוהה, שניוי פרמטרים קתנים יכולים להוביל לשיפור ממשמעותו במחair של המשימה. מתמטית, פרמטרי המודל, המיצגים על ידי θ , משתנים עבור כל משימה T_i להיות θ'_i , כאשר

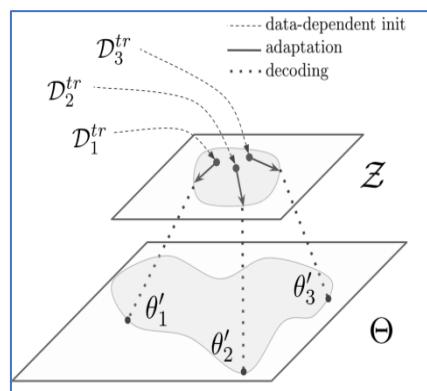
$$\theta'_i = \theta - \alpha \nabla_{\theta} L_{T_i}(f_{\theta})$$

עבור פונקציית מחיר L והפרמטר α . כאשר מבצעים מטא-למידה לעדכון הפרמטרים, מחשבים למעשה SGD:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} (L_{T_i}(f_{\theta}))$$

כאשר המשימות נציגות מトー (T) k - β הוא גודל הצעד של המטא-למידה

שיטה מעניינת נוספת בשם Latent Embedding Classification (LEO) = מימושת את הגישה של מטא-למידה במרחב ייצוג לטוני בעל ממדים נמוכים, ולאחר מכן עבר בחרזה למרחב המאפיינים הרוב ממדים.



איור 9.19 שיטת LEO (Latent Embedding Classification)

השימוש במרחב מאפיינים בעל ממדים נמוכים המשמרים את המאפיינים החשובים לייצוג הקטגוריות, שיפר-באופק ניכר את תוצאות הסיווג, כפי שניתן לראות בטבלה הבאה

Method	5-way 1-Shot	5-way 5-Shot
MAML	48.7%	63.11%
LEO	61.76%	77.59%

איור 9.20 השוואת ביצועי דיק של שיטות מטא-למידה על קטגוריות חדשותיות עבור mini-ImageNet.

9.4.4 Data Augmentation

גישה נוספת להטבות מטא-למידה נזקפת ביצירת דוגמאות כדי להימנע מהטיה. שיטות אוגמנטציה למשה יוצרות דאטא חדש על סמך הדאטא המקורי. השיטות פשוטות יותר מייצרות מהטבות הקיימות תМОנות ראי, שינוי תאוריה וקונטרסט, שינוי סקללה, שינוי צוויות, ואף הוספת רעש רנדומלי. כל אלו הראו שיפורים ביכולות הרשותת למדוד קטגוריות שהו במצב של חוסר איזון. דרך נוספת היא שימוש ברשתו-גנרטיביות (GANs) על מנת לייצר דוגמאות רלוונטיות, למשל דוגמאות של אותו האובייקט מזוויות שונות. שיטה מעניינת של אוגמנטציות היא CutMix – בה פאצ'ים של תМОנות נחטכים ומודבקים בתМОות האימון וגם התיאוגים מעורבבים בהתחם. שיטה זו הגיעה לביצועים מרשימים בסיווג תМОות וגם בזיהוי אובייקטים, ככל הנראה בגלל שהיא מאפשרת למודל להיות גנרי יותר בהתייחסות לחלקים שונים מהטובה המשפיעים על הסיווג לקטgorיה

9. References

Detection:

<https://arxiv.org/pdf/1406.4729.pdf>

Segmentation:

<https://arxiv.org/ftp/arxiv/papers/2007/2007.00047.pdf>

SegNet:

<https://arxiv.org/pdf/1511.00561.pdf>

<https://mi.eng.cam.ac.uk/projects/segnets/#demo>

<https://arxiv.org/pdf/1409.1556.pdf>

<https://arxiv.org/pdf/1502.01852.pdf>

Face recognition:

https://docs.opencv.org/master/d2/d42/tutorial_face_landmark_detection_in_an_image.html

<http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>

10. Natural Language Processing

המשמעותית בתיבת החיפוש ב-Google ו-Grammarly, מערכת לתייען תחבירו לטקסט באנגלית. דקדוק ועוד. כמה דוגמאות לאפליקציות כאלה שפכלנו מכיריים ה-זויי, העוזרת הקולית של אפל, ההשלה אוטומטית (Text summarization), פיתוח כלים המסייעים להתרמודד עם משימות אלה יאפשר (Question answering) ועוד משימות רבות אחרות. פיתוח כלים קוליות, מערכות תרגום, מערכות אוטומטיות לבדיקות (between the lines) לפתח אפליקציות שיעזרו לנו ביום יום כגון עוזרות קוליות, מילון אונליין ועוד. המטרת העיקרית היא לפתוח שיטות ומודלים שיאפשרו "להבין" את התוכנה הטקסטואלי, ואת הנזיאנסים והקשרים של השפה. H-P-NLP =עולם המספר רב של-'תמי' משימות, כגון ניתוח סנטימנט של טקסט (Sentiment analysis), תמצאות אוטומטי של טקסט (Text summarization), ועוד.

10.1 Language Models and Word Representation

ראשית נגיד ר מהו מודל שפה-מודול שפה-המגדי-התפלוגות-המורכבות-הסדרות האפשריות-של-מילך (לומם-משמעותי-פֿסְקָא-וכדומה) מודל זה מקבל סדרה של מילים ותפקידו הוא לחזות מה היא המילה הבאה שתنبي את הסתברות המרבית לרצף ביחד עם המילה הנוספת.

כעת נתאר באורה מתמטית מיהו מודל שפה. נניח ונთן משפט עם n מילים = w_1, \dots, w_n , אז ההסתברות לקבל את המשפט הזה הינה:

$$P(w_1, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{t-1})$$

ניך לمثال את המשפט הבא:

Take a big corpus

הסתברות של משפט זה ניתנת לחישוב אופן הבא:

$$P(Take, a, big, corpus) = P(Take) \cdot P(a|Take) \cdot P(big|Take, a) \cdot P(corpus|Take, a, big)$$

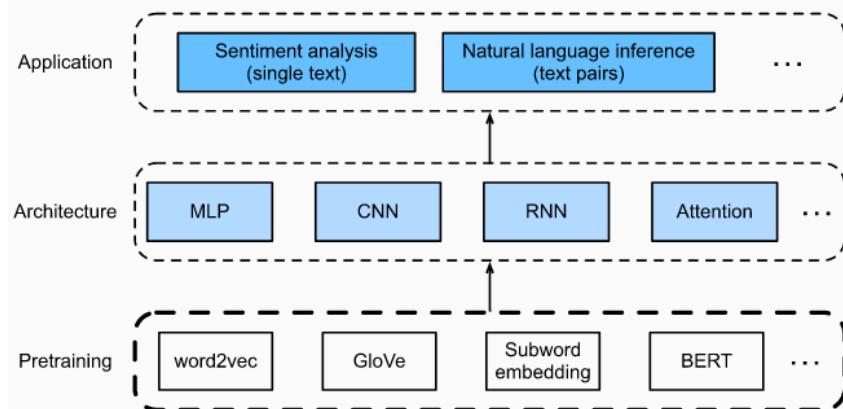
באותן הפסטיבNi בייתר נוכנְלַשְׁעָרָה את ההסתברויות האלו באופן הבא: ניקח **טוארכו-טומספיק** גדול ועשיך בענין מיל'פ'שנות, כמו למשל כל ערci ויקפדייה, פשוט נספוחת כל המיל'ים והצירופים שモפייעים בז'ההסתברות של כל מילה תהיה אחוֹד הפעמים שהיא מופיעה בטקסט, וההסתברות המונטנית תחשב באופן דומה – על ידי מנית מספה המופיעים של צירוף כלשהו וחולקה במספר הפעמים שהמיל'ה עצמה מופיעה. באופן פורמלי נוכל לרשום זאת כך:

$$P(W_i) = \frac{\#W_i}{N}, P(W_i|W_j) = \frac{\#W_i, W_j}{\#W_i}$$

ב'אם נחליט לחת את מודל השפה זהה או שניצר מודלים מתוחכמים יותר כפי שנראה בהמשך המודל הוא אח'ך הדברים היסודיים ביותר בשפה, כיוון שבאזורנו ניתן לבצע מגוון שימושות.

כאמור – ב כדי שנוכל לבנות מודל שפה או לאמן כל מודל אח-נ-צטראָר קודם כל לייצג את הטקסט בצורה כלשהיא – מיצגת באמצעות צירוף אותיות. כך למשל המילה הראשונה במשפט בשיטה שהוצגה לעיל – כל מיל-ה-ת (Token) –

שראינו מרכיבת מצירוף של האותיות a, k, T שיפורט בהמשך נוגע לכך לדבר על ייצוג למילים עצמןvr כר שכך
חידה אוטומטית מילה וצירוף של היחידות האלה ירכיבו ייצוג של משפה
באופן סכמטי, ניתן לתאר את משימת עיבוד השפה מקצתו לסתה באופן הבא:



איו-1.1.1 כתהlixir פיתוח אלגוריתם של מודל שפה: א. מייצגים את הטקסט בזורה כלשהו (ניתן כמובן לקחת ייצוג קיימש שנבנה על בסיס דאטסהספ-אחר). ב=מאמינם מודל שמקבל כ-~~טקסט~~ את הטקסט ואוטו יציגו בדרך כלשהיא~~א-טקסט~~ מוציא-~~טקסט~~ מסויים. למודל כזה יכולות להיות ארכיטקטורות שונות. ג. באמצעות המודל המואמן ניתן לבצע משימות קיצה שונות.

בהמשך פרק זה נתמקד בשכבה התחכונה של התרשימים: נתאר מספר שיטות מרכזיות ליצוג טקסטים, ונראה כיצד ניתן לאמן מודלי שפה שונים היכולים לבצע כל מיני משימות

10.1.1 Basic Language Models

מודל השפה הראשון אותו נציג הינו **n -Grams** – מודל סטטיסטי-המניח שהסתברות למילה הבאה תליה אף ורק ב- n המילים שקדמו לה בסדרה. הנחה זאת נקראת ‘הנחה מרקוב’ (Markov assumption), ובאופן כללי יותר, מודלי מרקוב (או שרשרת מרקוב במרקלה הדיסקרט) מגדירים מודל הסתברות-המניחים שניתקלחוות הסתברות של אירוע עתידי – בהתבסס על האירועים שהתרחשו-עד לזמן שקדמו למאורע – מבלי להתחשב באירועי עבר רוחקים מזמן

מודול-ה- n -Gram – הפשטוט ביותר נקרא **unigram**. במודול זה אנחנו חוזים את המילה הבאה לפי הತדיות של המילה עצמה – **bigram** – מוביל להתחשב במקודם לה. כМОון שחויזי צזהה-קבוע-תיכמיון שהמילה הבאה חיבת לה **תלויה במילים שקדמו לה** – כמו כן יהיו מילים **הנחות** – שמשמעותם באופן תDIR בטעס-שאיין בהכרח משפיעות על ההקשר. לכן, נסתכל על מודל קצר יותר מרכיב הנקרא **bigram**, המתיחס למילה האחרונה-הקדמתה למילה הנחוצה במודול **bigram** אנחנו מניחים את המשווואה הבאה:

$$P(w_n | w_{n-1}, w_{n-2}, \dots, w_1) = P(w_n | w_{n-1})$$

כלומר – רק למילקה-האחרונה יש השפה ערך-החיזוי של המילה הבאה, וכל המילים שלפניה הן חסרות השפה ערך-ה-**התפלגות של המילה הנחוצה** (וממילא גם על המשך המשפט) – באופן כללי, מודול- n -grams – מניח את המשווואה הבאה:

$$P(w_n | w_{n-1}, w_{n-2}, \dots, w_1) = P(w_n | w_{n-1}, w_{n-2}, \dots, w_{n-N}), N \leq n$$

ניקח לדוגמא מספר משפטים וננתן אותם בשיטת **bigram**:

- I know you
- I am happy
- I do not know Jonathan

ונניח-ו-נרצה לבחוקף את ההסתברות שהמילה **Jonathan** היא המילה הבאה אחרי הסדרה **do not know** – **I=ראשי** – נגיד את המילון, המכיל את כל המילים האפשרות בשפה

$$V = \{I, know, you, am, happy, do, not, Jonathan\}$$

כעהונכל להעיר את ההסתברות לכך על ידי ספירת כמה הפעמים שהצמד $(know|Jonathan)$ =מופיע בטקסט
ולנրמל בכמות הפעמים ש-won't מופיע בטקסט עם מילה כלשהי (כולל הפעמים שמוופיע עמו Jonathan)=באופן
פורמלי נגדיר את המשוואה הבאה

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_{w \in V} C(w_{n-1}w_n)}$$

כאשר האות ? =מסמנת את מספר הפעמים שצמוד מסוימ-בטקסט. ניתקלשים לב שהביטוי במכנה למשהו-סופר א�
כמה הפעמים ש- w_{n-1} קיימ בטקסט, ולכן נוכל לפשט את המשוואה האחורונה ורשום במקומה

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

אם כך, ההסתברות לכך שהמילה Jonathan מופיע אחרי המילה know היא:

$$P(Jonathan|know) = \frac{1}{2}$$

באופן דומה ניתן לנסוט לשערך אפקת ההסתברות של המילה הבאה על סמך יותר ממילה אחת אחרת, למשל לחת
-מילים אחרות. מודל זה נקרא trigram, וכך שנאמר לעיל, באופן כללי מודל המסתמן ערך 1 – A-밀יפלצורך
חישוב ההסתברות של המילה הבאה בטקסט נקרא N-gram.

המודל המרקבובי-סובל ממספר בעיות

1. טבלת ההסתברויות המתקבלת מודד דלילה. כיוון שה-digrams (שהוא סט האימון) הינם בגודל מוגבל-ללוולם לא
אפשר לראות את כל הקומבינציות הקיימות-מחייבים עבור צמדים שלא מופיעים בטקסט
האימון.

2. כאשר נרצה לחזות בעזרה מודל זה את ההסתברויות על טקסט חדש-כגנאה שנתקל במילים שלא נתקלו בהן
בטקסט האימון וכן לא נוכל להגיד את ההסתברות עבור ה-N-Grams. המכלול מיללים אלו.

בשביל להתמודד עם בעיות אלה – באופן מלא או חלקי – נוכל להפעיל שיטות החלקה (Smoothing) על ההסתברויות
של צמדים שהופיעו מעט/כלל לא הופיעו בטקסט האימון. שיטות החלקה במהותן לוקחות קצת מסת הסתברות
מהאורעויות השכיחיות (כלומר, ה-digrams-N-המורפיעים hei הרבה) או "טורמות" לאיורים פחות שכיחים. ישנו מגוון
שיטות להחלה הסתברות ונסקור כעת חלק מה-

Laplace Smoothing

הדרך פשוטה ביותר לבצע החלקה היא להוסף מספר קבוע κ לכל האירועים – באופן זהה אלו דואגיפלטת משמעות
גם-לצירופי-נדירים-שמופיעים מסpter מועט של פעמים (א-באים) לא מופיעים בכלל). אם למשל יש צירוף שמוופיע רק
פעם אחת בלבד ווותו יש צירוף שמוופיע 1000 פעמים, אז הוספה החלקה ככזה שהופיע $(\kappa + 1000)$ פעמים. הוספה זו
הראשון ככזה המופיע $(\kappa + 1)$ פעמים וביחס לציירוף השבנתיתיחס ככזה שהופיע $(\kappa + 1)$ פעמים. הוספה זו
כמעט ואינה משפיעה על הציירוף השני, אך היא יכולה להכפיל פי כמה את ההסתברות של הציירוף הראשון-הנרגול. באופן
שכאשר אנחנו מתעסקים עם הסתברויות נרצה לאחר הוספת הקבוע-במונה לתקן גם ארכ-מקדרה-הנרגול. פורמלי,
החלקה לפסל מוגדרת באופן הבא

$$P_{Laplace}(w_n|w_{n-1}) = P_{Add-\kappa}(w_n, w_{n-1}) = \frac{C(w_{n-1}w_n) + \kappa}{\sum_w C(w_{n-1}w_n) + \kappa} = \frac{C(w_{n-1}w_n) + \kappa}{C(w_{n-1}) + \kappa \cdot |V|}$$

כאשר $|V|$ =הוא גודל המילוק (כמה המילים המופיעות ב-dataset)=היתרון בשיטה זאת היא הפשטות שלה, אך-על-
זאת יש לה חסר-קובול הנובע מכך שהיא משתמשת את ההסתברויות של מאורעות תדיירים וכבר בעצם
משבשת את ההסתברויות שנלמדות מהtekst. כפועל יוצא יותר מדי מסת הסתברות גבוהה מהמאורעות השכיחיות
למאורעות עם הסתברות נמוכה. השיטה הבאה מנסה לתקן בדיק את זה.

כਮון שnitpick בוחור כל ערך שרווח-פואוטו להוציא-למכנה-באופן זה-קנינטן לשולץ-ברמה מסוימת-כמה להגדיל
את ההסתברויות לקומבינציות שאינן מופיעות בספט האימון על חישוב הקומבינציות השכיחות (כתלות ב- κ). בחירה
למשל ש- $\kappa = 0.5$ מאפשרת למזערת העיונות יכול להתקבל משינוי הספרה:

Backoff and Interpolation

שיטת החלוקת בסעיף הקודם נותרה קבועה להסתברות של מאורעות המקבילים הסתרות 0, וישן גישואה נוספת לשונות מענה למוגבלות האלה= nnn ואנחנו משתמשים במודל trigram, מודל המניח שהסתברות למילה הבאה תלולה בשתי המילים שבאות פניה= am נבעאת את השערות של כל שלישיה לפי הגדרה (=ספירת המופיע שליהם בטקסט), כל שלישית מילים שאינה מופיעה כרצף בטקסט תקבל הסתרות 0, ובuczם תקיים את המשוואה:

$$P(w_n | w_{n-1}, w_{n-2}) = 0$$

בכדי להימנע מלחת הסתרות 0 בaczא מצב, ניתן להיעזר גם בהסתברויות של הigram וunigram:

$$P(w_n | w_{n-1}), P(w_n)$$

שיטת backoff מציעה לקחת את כל המקרים בהם קיבלה>=ולשערר אותם מחדש באמצעות מודל bigram= bigram מכון ניקח את כל המילים שעדיין נותרו עם הסתרות>= $\text{(כלומר, צמדי מילים שאין מופיעים בטקסט)}$ ולשערר אותם באמצעות unigram= unigram . באופן הזרם מקווים להציג במצב בו מספר המילים בעלי הסתרות>= $\text{היא קטנה (עד אפסית),}$ כיוון ששימוש במודלים יותר פשוטים מאפשר שימוש מגוון רחב יותר של קומבינציות הקיימות בטקסט. שיטה דומה נקראת Interpolation, ובה במקובל לקחת את הרצפים בעלי הסתרות>=ולשערר אותם במודל הנמצא בדרجة אחת מהחת (למשל= $\text{=לשערר בעזרת bigram במקומם=trigram=}$ המודל מתחילה לkombinatsiya כלשהי של ה- bigram, unigram ו-trigram (למשל kombinatsiya לינארית=בשיטה זו גם מקרים שאינט-בעלי הסתרות>= ונזריף ב-m -bigram and bigram ב-unigram and bigram של השפה).

10.1.2 Word representation (Vectors) and Word Embeddings

עד כההמיל' פהשונויהו מצגוגובערתאותיות. כר' לדוגמא= $\text{המילה כל ביצוע על ידי צירוף האותיות GO=בווא=}$ שהמילה חתול תוצג על ידי הצירוף CAT. יציג זה מכל מאפיינס-נטקנספ= (תחבירים=) של השפה, קרפאי-המילה נכתבת עם זאת= $\text{יצוג זה חסר מאד, כיון שהואיתקה בלמידות=יצוא של מאפיינים סמנטיים. דוגמא להבנה סטטיסטית של השפה היא ההבנה שכלב וחתול הן לא מילים נרדפות=ארן כפקשורה=אחת לשפה באופן כלשהי= שתיהן מייצגות חיית מחמד שאנשים מגדלים בביטם. מודל המבוסס על יציג טקסטואל-של השפה לא יכול להציג להבנה סמנטית שלה, ולכן נרצה שהייצוג שלנו יהיה מספיק עשיר ויכילן מאפיינים תחביריים והן מאפיינס-סמנטיים של השפה}$

בפועל= $\text{נוכל ל�ות את היציג הטקסטואל-ליצוגו=טבצורה פשוטה=בזורה=One-Hot vectors=}$ מערך=בגודל המילון שלו, המציג כל מילה ב�לון בעזרת פקטור המתאים במערך. לדוגמא נתון המילון הבא:

Index	Word
0	Dog
1	Cat
2	Lion

נוכל ליציג את המילים השונות בעזרת קטובי-פבאורך 3, באופן הבא:

$$\text{Dog} \rightarrow [1, 0, 0]$$

$$\text{Cat} \rightarrow [0, 1, 0]$$

$$\text{Lion} \rightarrow [0, 0, 1]$$

כך, לכל מילה יהיה יציג וקטורי יחוי. עם זאת, יציג פשטי זיהי-גביעתי מכיוון שהיא קשה ללמידה ממנה מאפיינס-סמנטיים. כדי להבין את הסיבה לכך ראשית יש להגדיר את מושג הדמיון בעולם של וקטורים.

Cosine similarity

מעבר ליציג וקטורי של מילים דרוש מיתנו להגדיר דמיון בין וקטורים למרחב שנוצר. אחת מההגדרות הפופולריות לדמיון בין וקטורים היא Cosine similarity= $\text{=כמו רובההשיטות לחישוב דמיון בין וקטורי=גס פונקציית דמיון}=Z$ מבוססת על מכפלת-פנימית=של וקטורים. נניחו= $\text{ונטורים}=v, v_i$, שניים= $\text{בגיא}=N$. המכפלת הפנימית (dot product) בין וקטורים אלה פוגדרת-באופן הבא:

$$v \cdot w = v^T w = \sum_{i=1}^N v_i w_i$$

באמצעות הגדרה $\text{Cat} = \text{One-Hot vectors}$ את הדמיון בין הייצוג של המילים כלב וחתול במרחב הוקטורוני שנוצר בעקבות ייצוג בעזרת One-Hot vectors. כאמור, $\text{Cat} = [0, 1, 0]$. בעודם רצים שוקטוריים אלה כן יהיו דומים במובן מסוים, אך המילה כלב באוטו מרחיב הינה: $[0, 1, 0] \rightarrow [1, 0, 0]$. לכן, המכפלה הפנימית במרחב זה תהיה:

$$[0, 1, 0] \cdot [1, 0, 0] = 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 = 0$$

תוצאה זו מדגימה את הביעיות ביצוג פשוטי זה מכיוון שהיא בין קשורה לבין מילויים דומים במובן מסוים, ולפחות תוצאה זהה מתרחשת על אף שכל אין קשר בין שתי המילים למשמעות בשיטה זו הדמיון בין ייצוגים של כל זוג מילים יהיה אפס.

Bag-of-words

דרך אחת לייצר קשר בין מילים בעלות קשורה סמנטי $\text{Cat} = \text{One-Hot vectors}$ לא רק למילים בודדות אלא גם להקשרים בתוך המשפט עצמו. באופן זה נוכל להגיד כי ייצוג הוקטורוני של מילה על ידי ספירה של כמות הפעמים שמליה אחרת נמצאת אינה באוטו הקשר. שיטה זאת נקראת Bag-of-Words , כלומר שטחן של מילים מופיעה אבסנסית גם לחתת רקהלוּף של מספר מילים מתוך המשפט (לרוב אורך החלון קטן מאוד המשפט). לדוגמה, נניח נתונים לנו הטקסט והמילון הבא:

טקסום:

- [The dog is a domestic mammal, not wild mammal], is a domesticated descendant of the wolf, characterized by an upturning tail.
- [The cat is a domestic species of small mammal], It is the only domesticated species in the family.

מילוק:

1. The	5. Mammal	9. Animal	13. Tail
2. Is	6. Not	10. Descendant	14. Cat
3. A	7. Natural	11. Dog	15. Species
4. Domestic	8. Wild	12. Wolf	16. Small

כעת נרצה לייצג את המילים Dog ו- Cat באמצעות Bag-of-words (=כמות המילים לפני ואחרי המילה שנרצה לייצג. חלון זה מסומן בטקסט טבוסורגים מרובעווים בצלב אדום). נבנה מטריצה עם מספר עמודות כאורח המילון ומספר שורות כמספר המילים אותן נרצה לייצג (לרוב מספר השורות יהיה כמספר המילים במלון, אך לשם הדוגמא נציג כאן טבלה קטנה יותר). עבור כל מילה, נבדוק כמה פעמים היא נמצאה בטקסט באותו חלון יחד עם מילים אחרות.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Dog	1	1	1	1	2	1	1	1	0	0	0	0	0	0	0	0
Cat	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1

כעת נוכל לראות מה מכפלה הפנימית בין שטחן הוקטוריים המיצגים את המילים Dog ו- Cat עם זאת, ישנו שתבבויות נוספות הנובעות מפשטות פתרון זה:

1. מילוק פחוות המשמעותית $\text{Cat} = [1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$, כלומר למספר שפה לא בהכרח מושיפות מידע משמעותי לייצוג.
2. מילוק המופיע פעמיים בטקסט של עיתיפה רוחנית הייבולות ייצוג הוקטורי מואז-דיליל מושגדי לשובה האחריה. הסיכוי למכפלה הפנימית בעלת ערך קטן מאוד (ע-אפס) עם רוב הוקטוריים האחריים.

TF-IDF

הדרך הטבעית לפתרון של הבעיה הראשונית – רעש שנוצר ממילאים בעלות תדירות גבוהה שאינן תורמות בייצוגו היא $\sum_{i=1}^n \frac{1}{\log(n)} \cdot \log(\frac{1}{P(w_i)})$. פתרון זה אינו יעיל, כיוקשה תדריות של מילים משנה ביחסם/ $\log(n)$ שמהם נלקח הטקסט. לכן פותחה שיטה יצוג הנקראת IDF-TF, ומטרתה להפחית את רעש זה באופן אוטומטי.

בשיטת IDF-TF מגדים פונקציה ניקוד אחרת על מנת רכיבם-במקומם להסתכל על חלון אחד בלבד לכל מילה, ניתן להסתכל על כל המילים בחולנות הנוצרים מסביב למילה המייצגת וולעת לה ניקוד לפי התדרוות של אותה מילה. באופן פורמלי $tf \cdot idf$ מוגדר בצורה הבאה:

$$tf = \log(C(w, c) + 1)$$

משמעות הביטוי היא שאפקטופרים מכוחם ללחצמו פועעה בקונטקסט (=בחלון) של המילה המייצגת (על התוצאה מפעלי \log) בשביל rescaling של ערכים גבוהים מאוד).

עד כה השיטה אינה שונה מהמאות מספירה כמו שראינו בwords-Bag, אך מה שעושה אותה-IDF-TF שונה הוא הביטוי השני. הביטוי idf מוגדר בצורה הבאה

$$df = \text{count}(w \in \text{Context})$$

$$idf = \log\left(\frac{N}{df}\right)$$

המונח df מציין את כמות הפעמים שהמילה שמהופיעה בקונטקסטים אחרים מ- df מופיע עד כה. מילאים במילון, נשים לב שגם מילה מסוימת, למשל the , מופיעה בכל הקונטקסטים של כל המילים במילון, אז הביטוי בתוך הלוג יהיה $\log(N - df) + 1$ וולעת זאת אם מילוק-מסומנת מופיעה אף ורק בקונטקסט אחד, אז הערך שהוא בתוך הלוג יהיה N .

לבסוף, TF-IDF מוגדר באופן הבא:

$$TF - IDF = tf \cdot idf$$

מדד משקלם זה מציין עבור ייצוג של מילה מסוימת מלהות הנמצאות אליה בקונטקסט בואפן תדרי-אך אין נמצאות בקונטקסט של מילים אחרות

PPMI - Positive Pointwise Mutual Information

כעת נרצה לפתור את הבעיה השפיעתית – בביטוי הייצוג הדليل למילאים שאינם תדריות בטקסט – שיטת PPMI. פונקציוניקוד המחשבת את היחס בין הסיכוי של שתי המילים במאז'יחד לעומת סיכוי לרואותן נפרד – $\frac{P(x,y)}{P(x)P(y)}$. כעת בשילוב חישב את הערך של התא המייצג את המילה עקבותיו הייצוג של המילה אנשטיין בהסתברות הנ"ל ונפעילו. אם ההסתברות של ראות המילוי, איבריה שווה לכפל ההסתברויות לראות כל אחד יחד נקבע שערך הביטוי הוא $\log(\frac{P(x,y)}{P(x)P(y)})$. לעומת זאת אם הסיכוי של המילוי יחד גדול מהסתוכי שיראו אותם יחד. אז נקבע ערך הגדל מ-1=ישנו מקרה נוסף, בנסיבותיו לראות את הביטויים ביחד קטן מהסתוכי לראות אותם יחד. במקרה זה הביטוי שנקבע יהיה קטן מ-1 – אולי הלוואי היה שלילי, אולם קיומו שהערכים השילולים נוטים להיות לא אמינים אלא אם הטקסט שלנו גדול ממספריק), נוסף עוד אלמנט קטן לפונקציית החישוב:

$$PPMI = \max\left(\log\frac{P(x,y)}{P(x)P(y)}, 0\right)$$

באופן זה נוכל לנורמל את הערך הנמוך עבור מילים נדירות בטקסט

Word2Vec

השיטות שראינו עד כה לחישוב וקטורי-היצוג של המילים מאפשרות לנו לקודד מאפיינים סמנטיים בייצוג המילאים. עם זאת יש כמה חסרונות לשיטות אלו – ראשיתן יוצרות וקטורים מאוד דילימטיים ובונוסף לכך גודל הווקטורים תלוי בכמות המילים שיש לנו במילון – מה שיצור וקטורים גדולים שמכבים על החישובים המשמשים במשימות השפה השונות – למשל – ראינו קודפסניטן ליצג מילה באמצעות וקטור שכולו אפסים למעט תא אחד – עם הערכות במתיקות ייחודי לכל מילה עבור שפה עם אלפי מיללים – ואף יותר מכך – כל וקטור המיצג מילה הוא באורך עצום, ועם זאת הוא מאוד דיליל כיון

שים-בו רק מספר תאים מועט שערכ-שונה מ-0=~~לכל-~~נרצה לפתח שיטה שתיצור וקטורי-~~יזוג~~-~~זגדות~~-~~ים~~(dense) בערך
ממך קטע יותה

שיטות=~~Word2Vec~~=הינה שיטת=Self-Supervised שפותחה למטרת יצרה של וקטורי-~~יזוג~~-~~זגדות~~-~~ים~~ של מיל-~~ים~~
הפרודיגמה של למידה=~~דומה~~=Self-Supervised learning (ריליה, אך התוצאות אינן נתונם אלא נוצרים באופן אוטומטי מתוך הדעתה ללא מתואג. באופן זה ניתן לאמן מודלים עם כמות גדולה של נתונים לא מותיג בצורה יעה ולא צור-בתיאוג (שלול להיות מודע לך). בהקשר זה, אלגוריתם=~~SGNS~~=Skip-Grams-With-Negative-Sampling שבדון=~~self-supervision~~ באופן קיזוף=~~Word2Vec~~=~~גנרטיב~~ המשמש ב-
הוא להגדיר בעית סיווג שמרתה לחזות מה ההסתברות של מילים שונות להיות בקונטקסט של מילה נתונה=~~בבסוף~~
האימן לנקחים את המילים שנוצרו בעקבות תהליך אימון המשימה הראשית, והם יהיו היצוג של המילה=~~בכך~~
להבין זאת לעומק, נבחן טקסט פשוט יחסית בעזרת אלגוריתם=~~Word2Vec~~=~~גנרטיב~~ המשפט הבא חלק מהטקסט
האימן שלנו:

Folklore, legends, myths, and fairy tales have followed childhood through the ages.

ראשית נקבע א-אוּרְחָן הַלּוֹן (=מספר המילים עליה מסתכמים בסביבות כל מילה)=3. ערך-עובי=על הטקסט
וניצוח-תוגים בין כל מילה במלון ליתר המילים=למשל עbow=המיל-~~ים~~=tales=N-סיף לדאות טלנו דוגמאות חיוביות
של המילים שנמצאות בקונטקסט עbow=בנוסף=~~בכל~~ למנוע התנונות של כל היוזגים לוקטור בודד=~~נכטר~~
"להראות"=~~למודל~~ איך נראות דוגמאות שליליות וכן נשתמש בשיטה הנקרואט=~~negati~~ve sampling שיטה זו דוגמא=
מהמלון בהסתברות פרוציונלי=~~טלדיות~~ המילה (עם תיקון קטן שנตอน קצת יותר סיכוי למילים נדירות) את המילים
ישימושו ~~אות-~~דוגמאות שליליות. הרעיון מאחרו תהליך זה הוא שכאשר יש לנו מיליון גדול המילים שנגריל לא יהיה
קשריות למילה שעבורה אנחנו יוצרים את הדוגמאות שליליות.

Folklore, legends, [myths and fairy [tales] have followed childhood] through the ages.

word	context	Label
tales	myths	+
tales	and	+
tales	fairy	+
tales	have	+
tales	followed	+
tales	childhood	+
tales	great	-
tales	April	-
tales	the	-
tales	young	-
tales	orphan	-
tales	dishes	-

כך נוכל ליצור מתואג עבור משימת-הס-יוג=~~ו~~ונוכל להשתמש בתוצאות אלה בשבייל לאמן את המודל=~~הcono~~
במשימות-~~ו~~אג-בקשר ז-המעט שונה מסיווג במובן הפשוט של המילה: מטרת המודל-היא שבהינתן מיל-~~ה~~ש(במקרה
שלنك-~~tales~~)=~~נרצה~~ שהיצוג של מילים המופיעות באותק-קונטקסט עם כל מילה=~~במקרה~~=אל-שם-מופיעות עפ-

(mbhinnat mafalha panimiyet avo) yehi kroob liyizog shlel tales be'ud shel milim shai-komopiyut et-otekh kontekst yehi be'ulot yitzo' "shonra" (mbhinnat mafalha panimiyet avo similarity). neni shebharno sheh izog shel milah yehi vektor bagadol 100. nati'el at ha-tahalir cr shel milah makbil vektor rendomli. nesun et-aktofah yitzog shel milah-shabz-o-avto-hoktov shel milah kontekst -c-e=hasha'ipha hia-shem mafalha panimiyet vektori izog shel milim b'kontekst shel milim bagadol le'il=cosine similarity (hametrika hagadira dimyon bin vektorim) hia be'atzm mafalha panimiyet shel vektorim (=mafalha tofuf) normol). cutt nol la-hagidur be'ut logistic regression ba-afon ha-ba:

$$p(+|w, c) = \sigma(e_w, e_c) = \frac{1}{1 + e^{-e_w e_c}}$$

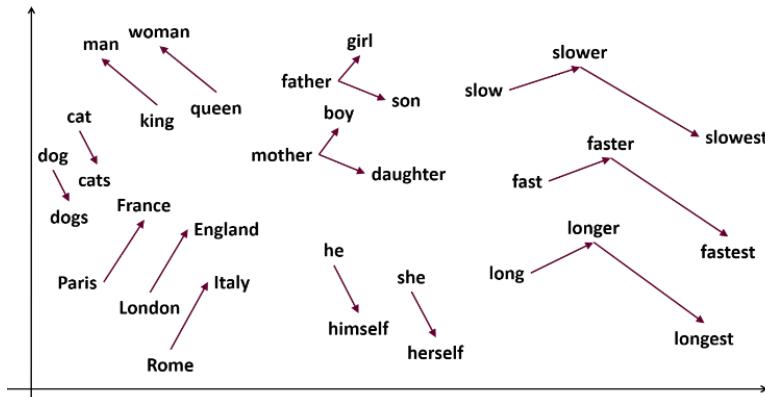
$$p(+|w, c) = 1 - p(+|w, c)$$

vohetam le-cr fonktsiyat metra (Loss) togder ba-afon ha-ba:

$$\begin{aligned} L &= -\log[p(+|w, c_{pos}) \prod_{i=1}^k p(-|w, c_{neg_i})] \\ &= -[\log(p(+|w, c_{pos})) + \log(\prod_{i=1}^k p(-|w, c_{neg_i}))] \\ &= -\left[\log(\sigma(e_w \cdot e_{c_{pos}})) + \sum_{i=1}^k \log(1 - \sigma(e_w \cdot e_{c_{neg_i}}))\right] \end{aligned}$$

noshim le-feshancho minichim ai talot bi-izog shel hadgamaot shel yililot-bekr shenbatz minimiyetza la-fonktsiyat metra zo negorof le-mafalha panimiyet-ben hizog shel milah libin milat kontekst la-hiot goba'ah vbo bzman la-mafalha panimiyet-ben vektorif shainim b'kontekst la-hiot namocha. cr hizog shel milah tales yehi "Doma" (kroob b'monchim shel milim b'kontekst v'shona-hamiyetza at tahalir haminyetza-b'mashr ha-imsh =stochastic gradient descent

achat ha-tzotot ha-yofot vohashivot shel shiyat vec2Word=venitna la-machsha ul id frisat vektori hizog emmad-namor. be'azrat shiyotot matkdomot la-hordot-mad= (cpf shehisbar b'horevha be-parik 2), nitn la-ziv-bsh-mad=ao talot-mad=ao vektori hizog shel milim la-achor ha-imsh



ai-1.2=vektori hizog shel milim la-achor b'iyut embedding=bamutzot vec2word=camdi milim ba-ili meshumot domha-miyotzayim ul id=vekotorim ba-otni ciyon=

nol la-hibin sheh mafalha vektori mukodd ma-pi'anim smantiv=hadgama-nitn la-rot sheh vektor ha-machber bin vektori hizog shel milim King, Man, Queen, Woman mukbil vbo oruk domha vektor ha-machber bin hizog shel milim Queen, Woman, King. Doma nuspat= ha-vekutor-ben shem shel arz le-ir ha-birah shel-makbil vbo oruk domha-vektor shbin arz achrot u-ir ha-birah matayimka. cambon shinitan lamed mcr ul kshrim smantivim, como lamesh sheh-cho King, Man, Queen, Woman zeha lihot shbe' King, Queen, Woman.

10.1.3 Contextual Embeddings

man-gan-niyyut yitzagi milim=embedding=shranimo ud ca lamed yitzogut=ubor cl milah. arz dor zeh ycol la-hiot be'utti mi-cion shafqat-be'utti-dina'it v-tali'ot ha-kshar, alota milim ycolah la-hiot comme pirushim. b'shvil la-hibin at ha-be'utti-tzat naftal ul mishpatim ha-ba'im:

1. We need to **book** the flight as soon as possible

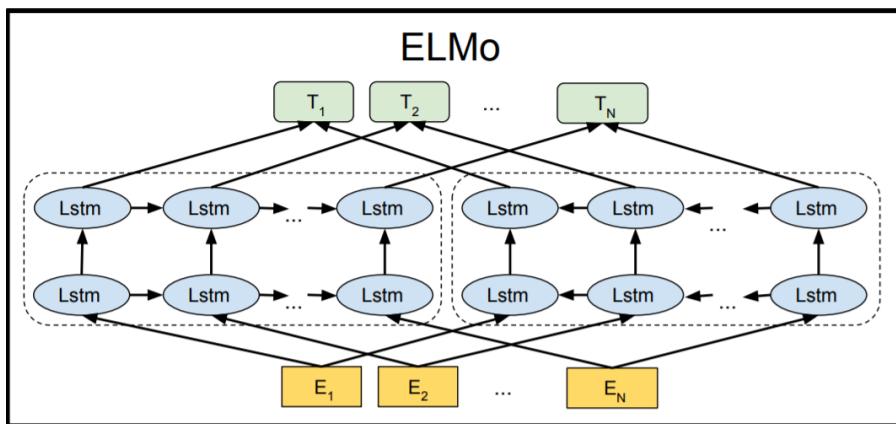
2. I read the **book** already

למייהָkooken¹ כפל משמעות במשפטים האלו. במשפט הראשון המילה משמשת כפועל ובמשפט השני כשם עצם תפקיך סמנטי שונה במשפט. אם כך ברוחו של הקשר שבו המשפט מופיע המשמעות שלה אף מנגנון².

לכן נרצה לפתח מנגנון embedding³ עבור המילים שבסביבתו וקוטור המיצג עברו שני המפעעים.

Embeddings from Language Models (ELMo)

אוחפהשיטו הראשו והציגו טכניקת למדידת ייצוג תלי הקשר למילים הינה ELMo⁴ – ארכיטקטורה⁵ לבניית Models – ארכיטקטורה⁶ לבנייה של מילוי ייצוג תלי הקשר של המילה במשפט בתוך המשפט הרעיון במודול זה הוא לחקות ייצוג של מילה, להוסיף ל המידע נוסף מהקשר של המילה במשפט ולקבל ייצוג חדש התלי גם בהקשר שלה⁷ – בניסוח אחר ניתן לומר לומר Sh-Mo⁸ – פונקציה המתקבלת משפט שבו כל מילה מיוצגת בדרך כלשהיא (למשל Word2Vec), ומוסיפה ליצוג זה גם את ההקשר של המילה בתוך כל המשפט – בפועל זה גושך על ידי מושמת אימוקמודל שפה דו-כיווני⁹ – המודל לומד לחזות גם את המילה הבאה בטקסט וגם את המילה הקודמת, ובכך הוא לומד לתת למילה גם את ההקשר שלה – ארכיטקטורת הרשת נראה כה



אינטראקצייתו של ELMo¹⁰. הקלט הינו משפט המיוצג כלשהו, והפלט הוא אותו משפט אך ככל מילה קיבלה מידע נוספת על ההקשר שלו וכעת מיוצגת באופן חדש. תהליך האימון והוספת ההקשר בין המילים נעשו באמצעות שכבות של רכיבי LSTM¹¹.

כפי שמתואר בפרק 6.2.1 כל בлок של LSTM¹² מקבל קלט שני רכיבי-הזמן יציג את ההיסטוריה של המשפט עד הנקודה בה מופיעה המילה של h_t – הקלט c_t – וקלט a_t של האיבר הנוכחי בסדרה, שבמקרה שלנו זה המילה הנוכחיית (x_t). המוצא של ה-LSTM¹³ יציג חדש המשקיל את רכיבי ההיסטוריה יחד עם הייצוג הנוכחי של המילה¹⁴.

בדומה לשיטות אחרות לייצרת ייצוג וקוטורי למילים אנחנו מאמנים את המודל בעזרת מודול שפה וחוזים את המילה הבאה בהינתן המילים הקודמות. אך בשונה מאלגוריתמים אחרים, ELMo¹⁵ משתמש בארכיטקטורה דו-כיוונית¹⁶ – קשבת הילך האימון משלבת משימת שפה נוספת הנוספת המנסה לחזות את המילה הקודמת בהינתנה¹⁷ של המשפט. הארכיטקטורה של ELMo¹⁸ מבוססת שכבות של LSTM¹⁹ – שמשורכבות זו על גבי זיהוי כתבי המאמר השוכבות התחרתנות מצלחות ללמידה פיצ'רים פשוטים (למשל מאפיינים פוטוטיפיים – סינטקטיים – למיניהם) – בעוד שהשכבות העלינויות למודות פיצ'רים מורכבים (למשל מאפיינים סמנטיים, כמו משמעות המילה בהקשר).

לאחר תהליכי האימון ניתן להקפיא את הפרמטרים של המודול ולהשתמש בו עבור משימות אחרות. הכותבים מציעים²⁰ לשרשר את הייצוג הווקטור של LSTM²¹ בכל שכבה ככה שיכיל אינפורמציה גם מתחילת המשפט עד המילה הנבדקה²² וגם מסוף המשפט עד המילה הנבדקה²³ – מה שקרה בפועל זה שהשכבות החבויות²⁴ של LSTM²⁵ הוכיחו²⁶ עצמן מהווים אובייגט-המיל – במשמעות צואצוא – כלומר-כלומר מיל-המשפט מיזג-על ידי התקשורת בין שכבות ה-LSTM²⁷ – שנעליהם²⁸ הם מוסףם פרמטרים קטנים שמאפשר-כיוון (Fine tune)²⁹ עבור משימה ספציפית. כך³⁰ לדוגמא יוכל להתאים את הייצוג של המילים למשימות מסווג של משפט לעומת משימת תיאוג של ישויות במשפט.

פה חשוב להזכיר נקודת מרכזית – בסופו של דבר התוצר של LSTM³¹ הינו מודל שפה הנקרא ייצוג של טקסט והוא³² אותו ליצוג תלי הקשר³³ – שכבות ה-LSTM³⁴ המשונות מאנו מנת ליצור ייצוג חדש עבור המילים³⁵

המתייחס גם להקשר=later=סימן האימון של מודל השפה=pre-training, נניחו לחת אוטו ולבצע transfer learning, כולם לשימוש פיצוגים שהוא מפיק גם למשימות אחרות על ידי הוספה שכבות בקצה=lآخر פרטום המאמר, Sebastian Ruder (חוקר NLP מפורסם) טען כי:

"It is very likely that in a year's time NLP practitioners will download pretrained language models rather than pre-trained word embeddings"

כלומר, עת מי שירצה לבצע משימת שפה כבר לא צריך רק על ייצוג סטטי של המילים אלא הוא יסתמך על מודל שפה מאומן שידוע לחת את ייצוגה של מיל-פ-ולה-פוך אותו ליצוגים קונטסטואליים=ELMo ועוד מאמרם רבים אחרים אימנו מודלי שפה מאומנים שניים לחת אותם ולהשתמש בהם עבור משימות קצה שונות על ידי הוספה של כמה שכבות ארכיטקטול המודל.

Bidirectional Encoder Representations from Transformers (BERT)

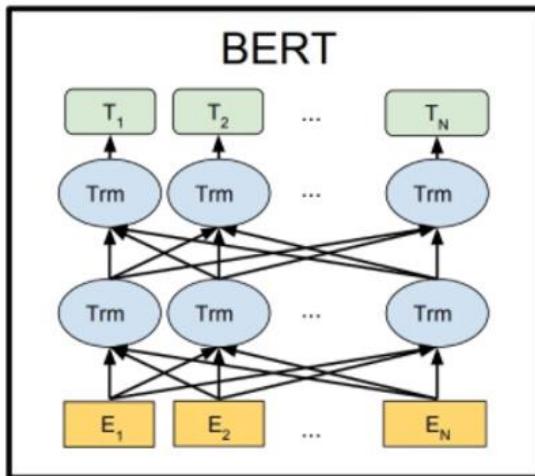
כאמור הרעיון של ELMo הוא ליצות embedding ובקיצור לקבץ מודל שפה המאפשר לחת פיצוג וקטורי של מיל-פ-ול-ה-ע-ש-יר אותו במידע על ההקשר של כל מילה בטקסט=למרות ש-ELMo=משתמש בשני היכונים של המשפט (חוזה את המשך המשפט מתחילה או תחילת המשפט מסופו) הוא אינו לומד משנה היכונים בתהיל-פ-א-ח-ד-א-ל-א-צ-ר לחלק את הלמידה לשני חלקיים שונים. בנוסף-כפי שהסביר בהקדמה לפרק 8, כאשר מתעסקים בעדרות ארכיטקטול מיל-ים, וקטור היצוג של כל איבר נהיה בעיתביביו שהוא מוגבל ביכולת שלו להכיל קשרים בז' מספר רב של איברים. במיל-ים אחרות, כאשר רוצים להוציא וקטור היצוג של מילה מסוימת קשור למיל-ים רחוקות, אנו מאלצים את היצוג "לזכור"-מידע רב, אך היצוג הווקטורי של המיל-פ-ELMo=אינו מצליח לעשות זאת בצוות מספיק טוב. לכן, על אף הצלחתו זבמשימות שונות, היא עדין התקשה במשימות בהן גדרת יכולתelnath טקסטים ארכיטקטול (כמו למשל משימה של summarization). בנוסוף לכך האלגוריתם יחסית איטי, כיון שכל פעם הוא משתמש על מילה אחת בלבד

בדי להתמודד עם בעיות אלו וליצור ייצוג המסוגל להכיל מידע איקוני גם ברכפים ארכיטקטול=ניתן להשתמש ב-self-attention=מיסבר בהרבה בפרק 8=אחד השימוש הראשוני במנגנון זה=attention עבור משימת עיבוד שפה הינה בטרנספורמים, ובפרט בארכיטקטורת רשת הנקרארט=BERT=המבוססת על ה-encoder=של הטרנספורמר המקורי=שימוש זה הינו פריצת דרך בתחום ה-MLP=משתמשים בארכיטקטורת רשת מבוססת attention=למעשה=BERT=מציע שיטה לבני-פ-יצוג קונטסטואלי של מיל-ים הבא להתמודד עפ-החולשות הקיימות ב-ELMo. נתאר בקצרה את העקרונות של מנגן ה-attention, שהוא הלב של BERT:

באופן הכל פשוט בקשר של עיבוד שפה הוא מנגנון שמשערף את הקשרים של כל מיל-ה-tekst של השם self-attention לשאר המילים באותו טקסט=כאמור מבצעים self-attention בערך קטע טקסט, מקבלים ייצוגים חדשניים של המילים הולקים בחשבון גם א-ה-ק-ש-ר-ים בין המילים השנו-ו-ב-או-תו טקסט=ב-ז-כ-ו-ת-ו של מנגנון ה-on-attention, ניתן לבנות ייצוג של מילה שתלו בקשרם של ה-ע-מ-ל-ים הנמצאות רחוק ממנה בקטע טקסט=כ-ל-ו-מ-ר-ה-ה-ק-ש-ר-י-ה המתקבלים בין המילים יכולם להיות מיזגים לצורה טובה גם עבר רצפים ארכיטקטול מיל-ים שאינן נמצאות בסמכות יחסית (שכאמור זה היה אחד החסרונות הגודלים של ELMo). בנוסוף, מנגנון זה מ-י-ת-ר את הצורך לעבור מילה אחת מילה בקטע טקסט לצורך בני-פ-יצוג המילים שבו. במקרים מעבר זה, ה-encoder=BERT מתקבל ה-embedding של המילים של ה-encoder=BERT, מה שעשו להקטין את הזמן הנדרש עבור בני-פ-יצוג של המילים=לכןencoder=BERT=ט-ר-נ-ס-פ-ו-ר-מ-ק יכול לשמש מודל שפה, אם מאמנים אותו בצורה מתאימה.

בשונה ממודול BERT=הShows מודול את המצב בכ-ל-ו-ק-ד-ת ז-מ-ק-ו-ב-ע-צ-ם מקודד את המיקום של כל מילה בק-ה-ק-ל-ט מתקובל כמילה בודדת בכל פעם, מודול הטרנספורמר מקבל את כל הק-ל-ט במת-את. لكن בשайл לחת בחשבון את המיקום של כל מילה במשפט אנחנו משתמשים באלמנט נוסף שנקרא Embedding Positional. אלמנט זה מקודד וקטו-ו-ייחודי לכל מיקום במשפט ובסוף מבצעים חיבור של הווקטור שנוצר מה-ק-ל-ט והווקטור שנוצר מה-מ-יק-ו-ה.

המפתחים של BERT=BERT=א-י-מ-צ-ה-א-ר-כ-ט-ו-ר-ת הטרנספורמר המקורית את ה-encoder=BERT=ו-ה-ג-ד-ר-מ-ש-י-ת א-י-מ-ן ח-ד-ש-ה בצד להפוך אותו למודול שפה=ב-כ-ד-ל-ב-נו-ת מודל מוצלח, תהליך האימוקש=BERT=כלל שתי משימות=1=Masked Language Model (MLM) (NSP)=Next Sentence Prediction (NSP)=המודול מקבל כל-ק-ל-ט-ז-ו-ג-ו-ת-ש-ל-מ-ש-פ-ט-י-פ-מ-ק-ט-ע טקסט=ומטרת המודול היא לחזות/chzotהאם המשפט השפה הוא המשפט המשכו של המשפט הראשי במשמעות המקורית=Arc-It-Ke-Tor ורשות נראית כ:



איוֹ.4.1.4 ארכיטקטורת BERT. הקלט הינו משפט המיצג כלשהו, והפלט הוא אותו משפט אך כל מילה קיבלה מידע נוספים על ההקשר שלו וכעת מיוצגת באופן חדש. תהליכי האימון והוספת ההקשר בין המילים נעשו באמצעות self-attention.

גפְּטָרְבְּדָהוֹמָה בְּBERT, ELMo, מציע בסופו של דבר מודל שפה מאומן הידוע לקחת טקסט השםיצג באופן מסויים ולהוסיף לו מידע עליהיחס בין המילים השונות שבtekst=תהליך יצירת המודל היה אמן יקר, ארככעת ניתן לקחת אותו ויחסית בקלות לצליל או תווואף להוסיף שכבות בקצת עבור שימושים שפה שונים.

GPT: Generative Pre-trained Transformer

עם הכניסה שלמנגנון-huggingfaceattention-טראנספורמרים לעולם NLP-הוצעו יותר ויותר מודלים שפה מבוסיסים attention. לצורך ההמחשה ניתן לציין שבשנים האחרונות שבערו מאז יצאת BERT=GPT=מודל-h-GPT המודלים היוצרים מפורטים הינה הינה self-attention-auto-regression, ככלומר, כאשר המודל חוצה את המילה הוא מוסיף את המילה לקלט עבור האיטרציה הבאה. וכך הוא יכול בעצם ליצור משפטים מהתחלה של מילה בודדת. אם נרצה לדijk, המודלים הללו לא תמיד משתמשים במילים מיוחדות, למשל מילויים ואפיון אוטויאות להם נקרא טוקנים או אסימונים. דבר זה יכול לעזור לנו בהכללה ולהקטיין את הסיכון לטוקן שלא נמצא במילויים Out of Vocabulary.

הארQUITטורה של GPT-בנוי מ-Transformers שמאפשר לבנות ארכיטקטורה عمוקה שמתחשבת בקונטקסט של המשפט עבור כל מילך(=Contextual embeddings)=ארQUITטורת הינה הינה המרכזית של GPT, כאשר בשונה מ-BERT=GPT=משתמשה רק ב-self-attention-h-Decoder (מנגן self-attention-ה-Decoder) שמקודד את הפיצ'רים והפלט של הינו הטוקן הבא.

השכבה הראשונה בארכיטקטורה של GPT היא שכבה הנקראת input encodingencoding והיא הפכת את המילך(או ליתר דיוק הטוקנים) לוקטורים, כלומר היא מבצעת word embedding.

לאחר קידוד הקלט נשימוש במודל-h-Transformer-לפדי פיצ'רים מהם נסיק את הטוקן הבא. התהילר הזה מתבצע באמצעות רכיב הנקרא Masked Self attention. בשונה ממנגנון self-attention שמקודד כל טוקן בעזרת הkowskiט של כל שאר הטקסט=GPT-צריך לקודד כל טוקן רק בעזרת הטוקנים שקדמו לו-פוקשבלב זההມידע היחיד שהקיים זה הטוקנים שנוצרו עד כה(וכmodoן שאינטשעדין לא נוצרו). כאשר מקודדים את הייצוג עבור טוקן מסוים רכיב הנקרא Masked Self attention מופיע כל וקטור של טוקן שבא אחריו-כך שהמודל לא יכול ללמידה יציג התליי מילים שבאות לאחר הטוקן המיצג, אלא עליו להפיק את המירב מהטוקנים הקודמים ל-

כיוון ש-GPT=GPT-פועל בצורה של auto-regressive-ללאhor האימון לייצור טקסט-באמצעות-=ניתף-למוד-להתחלה קירה של טקסט=ונבקש ממנו ליצור את המילים הבאות-כל-שלב נתן לו קלט את הטקסט הראשון ואת הטוקנים שייצר בשלבים הקודמים, והוא ימשיך וייצר עוד ועוד טקסט.

Perplexity

לאחר בניית מודל שפה, נרצה "למדוד" עד כמה הוא מצליח. לצורך זה יש להגדיר מטריקה שתאיימה. המטריקה הנקרא perplexity של מודל שפה הינה מושג הלקו מהתורת האינפורמציה והוא מודד כמה טוב מודל השפה חזקה השפה ב-Corpus שהוא ניסיון למדוד

לפנינו שנסביר את המושג באופן פורמלי: ניתן אינטואיציה למה אנו מצפים לקבל מהמטריקה שנבחר. נניח וננו מבצעים את הפעולה הבאה: ראשית לוקחים משפט שלם וחותכים ממנו את ההתחלתה, אז לוקחים את אותה התחלתה ומכוון יפה כקלט למודל שפה ומקשים מהמודל לחזות את המשך המשפט. כמובן שנרצה לקבל חיזוי שדומה ככל האפשר לשפט המקורי, וכן למדוד הצלחה של מודל על ידי השוואת הפלט שלו לשפט האמתי. באופן יותר כללי ניתן למדוד הצלחה של מודל השפה, ולהשווות את הפלט המתkeletal לטקסט המקורי. ניתן שמודל שפה הינו סטטיסטי, השוואת הפלט לשפט המקורי היא כתובות המילא שפה אינה משקפת בצורה מסוימת טובה את מידת הצלחה שלה. אם למשל במשפט המקורי כתובות המילא "לבנה" או אילו המודל חזה את המילה "רוח", השוואת שתי המילים כשלעצמם מראה לאורה שהמודל שגה לחוץין, אך בפועל אנו יודעים שmilim אלו נרדפות ולכן הפלט של המודל במקורה זהה או דומה כן טוב. לכן, נרצה לבחון מודד המסוגל לבדוק עד כמה סביר לקבל את הפלט של המודל בהינתן חלק מהמשפט המקורי.

מדד perplexity מודדplexity הטענה שמודל השפה אופן בו תיארנו את ההשווואה הפשטית בין טקסט המקורי לבין הפלט של מודל השפה. מודד זה מסתכל רק על הטקסט המקורי, והוא עובר מילה-מילה בテקסט זה ובודק מה הסתברות שמודל השפה ינביא את המילה הבאה בטקסט בהינתן כל המילים שלפנייה. ככל שהסתברות יותר גבוהה, כך המודל יותר מוצלח. באופן פורמלי, perplexity מוגדר באופן הבא:

$$PP(W) = P(w_1, w_2, \dots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \dots, w_N)}}$$

ככל שהמודל מנבא בהסתברות גבוהה יותר את המילים של המשפט המקורי, כך המונה שבתוך השורש יהיה יותר גדול, וממילא כל הביטוי עצמו של perplexity נהייה קטן יותר. לעומת זאת, ככל שערך ה-perplexity קטן יותר, כך המודל מוצלח יותר. נפתח מעט את הביטוי האחרון בעזרת כלל השרשרת

$$= \sqrt[N]{\frac{1}{\prod_{i=1}^N P(w_i | w_{i-1}, w_{i-2}, \dots, w_1)}}$$

למשל עבור מודל מבוסס bigram, המודד יהיה פשוט יותר ויראה כך:

$$= \sqrt[N]{\frac{1}{\prod_{i=1}^N P(w_i | w_{i-1})}}$$

כאמור לעיל, ככל שערך מודד perplexity נמוך יותר, כך מודל השפה איקוטי יותר.

10. References

<http://d2l.ai/>

ELMo, BERT:

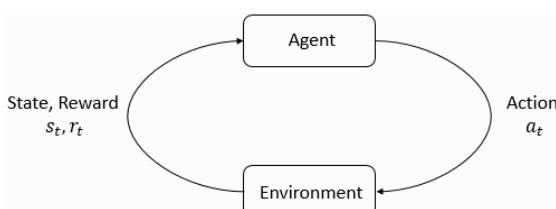
<https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>

11. Reinforcement Learning (RL)

רובי האלגוריתמים שבעולם הלמידה הינם מבוססי נתונים, כלומר, בהינתן מידע מסוים מהמנוע יימצא־בו חוקי־ות מסוימת, ועל בסיסו לבנות מודל שיוכל להתאים למקדים נוספים. אלגוריתמים אלה מחולקים לשני־יְפָאַתְּסִים:

1. אלגוריתמים של למידה מונחית, המבוססים על דאתה={ $y, x = S$ }, כאשר $\mathbb{R}^{n \times d} \in x = \text{הינו אוסף של אובייקט אובייקטים (למשל נקודות במרחב}=x\text{ או שטחים ו��), ו-}\mathbb{R}^n \in \text{עקבות אובייקט}=labels$ לכל אובייקט $x \in \mathbb{R}^d$ יש מתאinfeld $\mathbb{R}^1 \in y$
 2. אלגוריתמים של למידה לא מונחית עבורפהנדאוף $\mathbb{R}^{n \times d} \in$ זה הוא אוסף של אובייקטים לאותlabels ומנסום למצוא כלים מסוימים על דאטא זה (למשל - חלוקה לאשכולות, הורדת ממד ועוד)

למידה מבוססת חיזוקים ה^ז-פְּרִידִיגְמָנוֹסֶפֶת תחַתְּהַתְּחוֹם של למידת מוכנה, כאשר במקורה זה הלמידה לא מסתמכת על דатаה קיימ, אלא על-חקיריה של הסביבה ומצוותה המידניות/האסטרטגיה הטובה ביותר ליותר לעוללה ישנו סוכן שנמצא בסביבה שאינה מוכרת, ועלוי לבצע צעדים כך שהtagmol המctrבראותו הוא יקבל ה^יה מוקסימל-בלמידה מבוססת חיזוקים, בניגוד לפְּרִידִיגְמָנוֹסֶפֶת האחריות של למידת מוכנה, הסביבה לא יודעה מבעוד מועד להסתוכן נמצאת באז וDAOOT ואיבְּרִי-ודע בשום שלב מה הצעד הנכון לעשות, אלא הוא ר^ק-מקבל פְּרִידִיגְמָנוֹסֶפֶת הצעדים שלו, וכך הוא לומד מכך כדי לעשות ומהה כדי להימנע. באופן כליל ניתן לומר שמטרת הלמידה היא ליצור אסטרטגי-חק שבעל מיינן מצבי-פה לא יודיעים הסוכן יבחר בפעולות שבאופן מctrבר ה^יי ה^יי ייעילות עבורי. נתאר את תהליך הלמידה באופן גורף:



איור 11.1 מודל של סוכן וסביבה

בכל צעד הסוכן נמצא במצב t ובוחר פעולה s_t מהמעבירה אותו למצ' s_{t+1} , בהתאם לכך אוסף מקבץ מהסיבובים t הקיימים בה מתבצעת הלמידה היא בעזרת התגמול, כאשר נרצה שהסוכן יבצע פעולה מהמציאות ואו t -תגמול t (חיזוק) כדי מעניק מפועלות העברון הוא מקבל תגמול שלילי, ובמצטרב הוא ימаксם או פולקל התגמול מ-עבור כל הצעדים שהוא בחר לעשות כדי להבין כיצד האלגוריתמים של למידה מבוססת חיזוקים עובדים בראשו-יש להגדיר את המושגים השונים, ובנוסף יש לנסח באופן פורמלי את התיאור המתמטי של חלקי הבעיה השונים

11.1 Introduction to RL

בפרק זה נגדר באופן פורמלטי תחילה מركוב, בעזרתו ניתן לתאר א-בעיות של למידה מבוססת חיזוקים, וכן נראה כיצד ניתן למצוא אופטימום בעיות אלו בהינתן מודל וכל הפרמטרים שלו. לאחר מכן נדוקפקירה במספר שיטות המנסות למצוא אסטרטגיה אופטימלית עבור תהיליך מركוב כאשר לא כל הפרמטרים של המודל נתונים, ובפרק הבא יפוצב נושא על שיטות אלגוריתם הולב של למידה מבוססת חיזוקים, כיוון שהן מנסות למצוא נדב-על אופטימליות אלגוריתם הולב. שיטות אלה הולממשה להלוב של המודל המרתקובי עבור-קורחץ' למצוא אופטימום אסטרטגיה אופטימלית ועל בסיס תגמולים ללא ידיעת הפרמטרים של המודל המרתקובי עבור-קורחץ' למצוא אופטימום

11.1.1 Markov Decision Process (MDP) and RL

המודל המתמטי העיקרי עליו בנוים האלגוריתמים השונים של ST_k -הינתקהיל'ר החלטה מركובי-כלומר תהליכי-שבוב העבריים בין הממצבים מוקים את תכנת מרכוב, לפיה ההתפלגות של מצב מסוים תלויה רק במצב הקודם לו:

$$P(s_{t+1} = j | s_1, \dots, s_t) = P(s_{t+1} = j | s_t)$$

תהליך קבלת החלטות מركז-ומתווך על ידי סט הפרמטרי $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$

- מרחב המצבים של המערכת. המצב ההתחלתי מסומן ב- S_0 . State space (\mathcal{S})

– מרחב הפעולות. A הוא מרחב הפעולות האפשריות במצב S . – Action space (\mathcal{A})

$\text{Transition}(T) = \{ (s, a, s') \mid s, s' \in [0, 1], a \in \{0, 1\}, T(s, a, s') > 0 \}$

בזאת מוכיחים $T(s'|s, a) = \mathcal{P}(s_{t+1} = s' | s_t = s, a_t = a) = a$ הפעולה כעדי ביטוי זכיינית.

למעשה מיצ'אג את המודול – מה ההסתברות שבחירת הפעלה במצatz תביא את הסוכן למצו'ס.

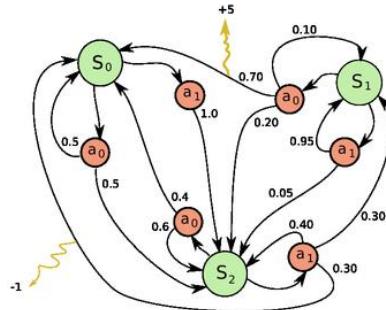
=הביטו= $\mathcal{R}_a(s, s') \rightarrow \mathbb{R}$ הינו פונקציה הנותנת תגמול^הרוווח לכל פעולה a הגורמת למעבר

מאנצ'ז s למצ'ז', כאשר בדרך כלל $[1, 0] \in \mathcal{R}_a$. לעיתים מסומנים את התגמול של הצד בז'מ'ז' ב- $-z'$

המרקביות של התהילה' באה ידי ביטוי בק-ש McCabe המכיל בתוכו את כל המידע הנחוץ בכך לקבל החלטה לגב' a_t , או במילים אחרות – כל ההיסטוריה עצם שמורה בתוך המצב s_t .

ריצ'ק של MDP מואפיינת על ידי הריבועי-הסודורא- $\{s_t, a_t, r_t, s_{t+1}\}$ – פועלך שהמתרחשת בזמן – וגורמת למעבר במצב s_t למצב s_{t+1} , ובנוסף מקבלת תגמול מייד- r_t , כאשר $p(s_{t+1} | s_t, a_t)$.

מסלול (trajectory) הינו סט של שלשות $\{s_0, a_0, r_0, \dots, s_t, a_t, r_t, s_{t+1}\}$, כאשר המצב התחלתי מוגדר מהתפוגות קלשיה- $f(s_t, a_t)$, והמעבר בין הממצבים יכול להיות דטרמיניסטי- $s_{t+1} = f(s_t, a_t)$ או סטטיסטי- $s_{t+1} \sim p(\cdot | s_t, a_t)$.



איו-פ-1.2 קת'היל' קבלת החלטות מרקבי. ישנו שלושה מצבים $\{s_0, s_1, s_2\}$, ובכל אחד מהם יש שף פעולה ואפשרות (עם הסתברויות) מעבר שונות $\{a_0, a_1\}$. עברו חלק מהפעולות יש תגמול שונה מ-0=מסלול יהיה מעבר על אוסף של מצבים דרך אוסף של פועלות שלכל אחד מהם יש תגמול.

אסטרטגיה של סוכן, המסתמנת ב- π , הינה בחירה של אוסף מהלכים-בבעוי והשל למידה מבוססת חיזוקים-נרצ'ק למצאו = **אסטרטגיה אופטימלית** = $\text{Optimal Policy } \pi: S \rightarrow A = \text{המקסמת את התגמול} = \text{המצטב}$ $(\sum_{t=0}^{\infty} R(s_t, \pi) = \text{כיוון}$ שלא תמיד אפשר לחשב באופן ישיר את האסטרטגיה האופטימלית-גנית להגדיר-ערוך החזרה- R =המבטא סכום של תגמולים, ומנסים למקסם את התוחלת של $\mathbb{E}[Return | \mathcal{S}, \mathcal{A}]$ =ערך ההחזרה הכי נפוץ נקרא- $Return$ באפ'ן הבא: עבר פרמטר- $(1) \in \mathcal{A}$, והוא מוגדר באופן הבא: עבר פרמטר- $(1) \in [0, 1]$, discount return

$$Return = \sum_{t=1}^T \gamma^t r_t$$

אם $\gamma = 1$, אזי מתעניינים רק בתגמול המידי, וככל ש- γ גדול כך יותר נתונים יותר מושעות לתגמולים עתידיים-כיוון ש- $[0, 1] \in r_t$, הסכום חסום על ידי $\frac{1}{1-\gamma}$.

התוחלת של ערך ההחזרה-**נקראת Value function**, והיא נתונה לכל מצב ערך מסוים המשקף את תוחלת התגמול שניתן להשיג דרך מצב זה. באפ'ן פורמלי, כאשר מתחאים במצב, ה-value function מוגדר כך:

$$\mathcal{V}^\pi(s) = \mathbb{E}[R(\tau) | s_0 = s]$$

בעזרת ביטוי זה ניתן לחשב **אזה-אסטרטגיה האופטימלית**, כאשר ניתן לנוקוט בגישה ישירה ובגישה עקיפה-הגיישה היישר-המנסה-הממצא בכל מצב מה פעולה הcy כדי. בהתאם לכך, חישוב האסטרטגיה האופטימלית יעשה באמצעות הבא:

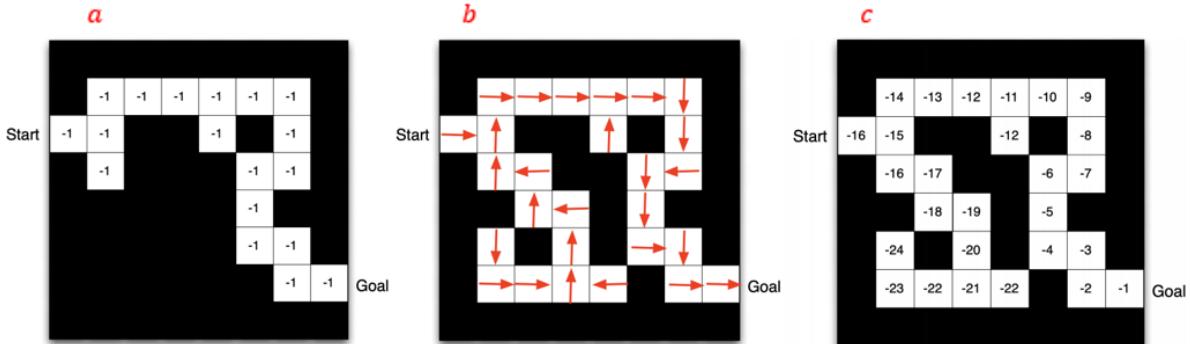
$$\pi(s) = \arg \max_a \sum_{s'} p_a(s, s') (\mathcal{R}_a(s, s') + \gamma \mathcal{V}(s'))$$

לעתיקת חישוב הישר מסובך, כיוון שהוא צריך ללקח בחשבון את כל הפעולות האפשרות, ולכן מסתכנים רק ערך-value function – לאחר שלכל מצב יש ערך מסוים, בכל מצב הסוכן יעבור למצב-בעל הערך הcy גדול מ- γ הממצבים האפשריים אליו הם ניתן לעבור. חישוב הערך של כל מצב געשה באפ'ן הבא

$$\mathcal{V}(s) = \sum_{s'} p_\pi(s, s') (\mathcal{R}_\pi(s, s') + \gamma V(s'))$$

ניתן לשים לב שבעוד הגישה הראשונית מתקדמת במציאות אסטרטגיה/מדיניות אופטימלית=על בסיס הפעולות האפשריות בכל מצב, הגישה השנייה לא מסתכלת על הפעולות אלא על הערך של כל מצב המשקף את התוצאות

התגמול שנייה להשיג כאשר נמצאים במצב זה



איו-3(a) מודל: המצב של הסוקן הוא המשבצת בו הוא נמצא, הפעולות האפשריות הן ארבעת הכיוונים, כל פעולה גוררת תגמול של -1, והסתברויות המעבר נקבעות לפי הצבעים של המשבצות (או אפשר ללחוץ למשבצות שחומות)=ב) מדיניות=החלטה בכל מצב היא צעד לבצע. c) Value של כל משבצת.

לסיכום, ניתן לומר שכל התchrom של \mathcal{V} -מבוסס על שלוש אבני יסוד

- מודל=האופן בו אנו מתארים את מרחב המצבים והפעולות=המודול יכול להיות נתון או שנצרך לשערר אותו, והוא מורכב מהסתברויות מעבר בין מצבים ותגמול עבור כל צעקה
- $\mathcal{P}_{ss'}^a = p_\pi(s, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$
- $\mathcal{R}_{ss'}^a = \mathcal{R}_\pi(s, s') = \mathbb{E}[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s']$
- פונקציה המתארת את התוצאות של התגמולים העתידיים – Value function

$$\mathcal{V}^\pi(s) = \mathbb{E}[R(\tau) | s_0 = s]$$

- מדיניות אסטרטגיה (Policy) = בחירה (דטרמיניסטית או אקראית) של צעד בכל מצב נתוך $\pi(s|a)$

11.1.2 Bellman Equation

לאחר שהגדכנו את ההמטרה של למידה מבוססת חיזוקים, ניתן לדבר על שיטות לחישוב אסטרטגיה אופטימלית. בפרק זה נתייחס למקרה הספציפי בו נתון מודל מركובי עם כל הפרמטרים שלו; כלומר אוסף המצבים, הפעולות והסתברויות המעבר ידועים=אמור=ה-Value function הינה התוצאה של ערך ההחזרה עבור אסטרטגיה נתונה=זאת כאשר מתחילה מ מצב s :

$$\mathcal{V}^\pi(s) = \mathbb{E}[R(\tau) | s_0 = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right]$$

ביטוי זה מסתכל על הערך של כל מצב, בלי להתייחס לפעולות המעבירות את הסוקן ממצב אחד למצב אחר. נתינו-ערך לכל מצב יכולה לסייע במציאת אסטרטגיה אופטימלית, כיוון שהוא מדרגת את המצבים השונים של המודול באופן דומה, ניתן להגיד את ה-Action-Value function של ערך ההחזרה עבור אסטרטגיה נתונה=זאת כאשר במצב s מבצעים פעולה a , ולאחר מכן ממשיכים לפי האסטרטגיה=זאת

$$Q^\pi(s, a) = \mathbb{E}[R(\tau) | s_0 = s, a_0 = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right]$$

ביטוי זה מסתכל על הזוג (s, a_t) , כלומר בכל מצב יש התייחסות למצב הנוכחי ולפעולות האפשריות במצב זה=בדומה ל-Value function, גם ביטוי זה יכול לסייע במציאת אסטרטגיה אופטימלית, כיוון שהוא מדרג עבור כל מצב את הפעולות האפשריות.

נוכל לסמך $\mathcal{V}^\pi(s) = \mathbb{E}[R(\tau) | s_0 = s]$ את הערך של האסטרטגיה האופטימלית=זאת=Optimal Value function ועבור אסטרטגיה זו מתקיים:

$$\mathcal{V}^*(s) = \max_{\pi} \mathbb{E}[R(\tau)|s_0 = s], \mathcal{Q}^*(s, a) = \max_{\pi} \mathbb{E}[R(\tau)|s_0 = s, a_0 = a]$$

הרבה פעמים מתעניינים ביחס שבין \mathcal{V} ו- \mathcal{Q} , ונitin להיעזר במערכות הבאות:

$$\mathcal{V}^\pi(s) = \mathbb{E}[\mathcal{Q}^\pi(s, a)]$$

$$\mathcal{V}^*(s) = \max_{\pi} \mathcal{Q}^*(s, a)$$

באופן קומפקטי ניתן לרשום את $(\mathcal{V}^*)^\pi$ כך

$$\forall s \in S \quad \mathcal{V}^*(s) = \max_{\pi} \mathcal{V}^\pi(s)$$

כלומר, האסטרטגיה π הינה האופטימלית עבור כל מצב s .

כעת נთוקמודל מركובי עם כל הפרמטרים של π אוסף המצביעים והפעולות, הסתברויות המעבר והתגמול עבור כל פעולה, ומעוניינים למצוא דרך פועל-או-פיטימלי עבור מודל זה-ניתן לשנות זאת בשתי דרכים עיקריות-מציאת האסטרטגי-($\pi|a$)^{action}-value של כל מצב-בחירה מצבים בהתאם לערך זה. משימות אלה יכולות להיות מסובכות מאוד עבור משימות מורכבות וגדולות-ולכן לעתים קרובות משתמשים בשיטות איטרטיביות ובקירובים על מנת לדעת כיצד לנוכח בכל מצב-הדרישה-לחישוב($\pi|s$)^{value}-משתמש בBellman equation המבוססת על תכונות-דינמי. נפתח את הביטוי של $\mathcal{V}^\pi(s)$ מתוך ההגדרה של:

$$\mathcal{V}^\pi(s) = \mathbb{E}[R(\tau)|s_0 = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right]$$

נפצל את הסכום שבתוכלה לשני איברים – האיבר הראשון ויתר האיברים:

$$= \mathbb{E}_\pi \left[r_{t+1} + \gamma \cdot \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s \right]$$

cutet נשתמש בהגדרת התוחלת ונקבל:

$$\begin{aligned} &= \sum_{a,s'} \pi(a|s) p_\pi(s, s') \left(\mathcal{R}_\pi(s, s') + \gamma \cdot \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s \right] \right) \\ &= \sum_{a,s'} \pi(a|s) p_\pi(s, s') (\mathcal{R}_\pi(s, s') + \gamma \cdot \mathcal{V}^\pi(s')) \end{aligned}$$

הביטוי המתkeletal הוא מערכת משוואות לינאריות הניתנת לפתורן באופן אנלטי, אם כי סיבוכיות החישוב יקרה. נסמן

$$V = [V_1, \dots, V_n]^T, R = [r_1, \dots, r_n]^T$$

$$T = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}$$

ונקבל משווהה מטריציונית

$$V = R + \gamma T V \rightarrow V = R + \gamma T V$$

$$\rightarrow \mathcal{V}^\pi(s) = (\mathbb{I}_n - \gamma T)^{-1} R$$

בגלל שהערכים העצמיים של T חסומים על ידי 1, בהכרח יהיה ניתן להפוך את $\gamma T - I_n$ מה שmbטיח שייהי פתרון למשווהה, ופתרון זה הוא אף ייחיד עבוק- \mathcal{V}^π . כמשמעותם אוף \mathcal{V} ניתן למצוא גם אוף \mathcal{Q}^π על ידי הקשה

$$\mathcal{Q}^\pi(s, a) = \sum_{s'} p_\pi(s, s') (\mathcal{R}_\pi(s, s') + \gamma \mathcal{V}^\pi(s')) = \sum_{s'} p_\pi(s, s') \left(\mathcal{R}_\pi(s, s') + \gamma \sum_{a'} \pi(a'|s') \mathcal{Q}^\pi(a'|s') \right)$$

Iterative Policy Evaluation

הסיבוכיות של היפוך מטריצה הינה⁽³⁾ $\mathcal{O}(n^3)$, ובעקבות גודל החישוב נהיה מאוד יקר ולא יעיל כדי לחשב את הפתרון באופן יעיל, ניתן כאמור להשתמש בשיטות איטרטיביות=שיטות אלו מבוססות על אופרטור בלמקרה המוגדר באופן הבא:

$$BO(V) = R^\pi + \gamma T^\pi \cdot V$$

ניתן להוכיח שאופרטור זה הינו העתקה מכווצת (contractive mapping), כלומר הוא מקיים את התנאי:

$$\forall x, y: \|f(x) - f(y)\| < \gamma \|x - y\| \text{ for } 0 < \gamma < 1$$

במילים: עבור שני וקטורים במרחב אופרטור f כומספְּקָה חסום ב- $0 < \gamma < 1$, אם נPUT-א-ת האופרטור על כל אחד מהווקטוריים=ונחשב את נורמת ההפרש, נקבל מסFOR-קְטַף יותר מאשר הnormה בין הווקטורים כפול הפקטור γ . אופרטור המקיים תכונה דומה לעתקה מכווצת, כיון שנורמת ההפרש של האופרטור על שני וקטורים קטנה מnorמת ההפרש בין הווקטורים עצמה=הוכחה

$$\|f(u) - f(v)\|_\infty = \|R^\pi + \gamma T^\pi \cdot u - (R^\pi + \gamma T^\pi \cdot v)\|_\infty = \|\gamma T^\pi(u - v)\|_\infty$$

מטריקת אינסוף מוגדרת לפיה: $\|s\|_\infty = \max_{s \in S} |s(s) - v - u|$. לכן נוכל לרשום:

$$\|\gamma T^\pi(u - v)\|_\infty \leq \|\gamma T^\pi\|_\infty \|u - v\|_\infty$$

הביטוי $\|\gamma T^\pi\|_\infty$ למעשה סוכם את כל ערכי מטריצת המעברים, שכן הוא מסתכם ל-1, ונכתב:

$$= \gamma \|u - v\|_\infty$$

ובכך הוכחנו את הדרישה

לפי משפט נקודת השבתה של בנך, להעתקה מכווצת יש נקודת שบทה (fixed point) יחידה המקיים $f(x) = x$ וסדרה $x_t = f(x_{t+1})$ המתכנסת לאו-ת-ה- נקודת שבת=לכן נוכל להשתמש באלגוריתם איטרטיבי=בעובן שיביא את כל נקודת שבת, ולפי המשפט זהי נקודת השבת היחידה ומילא הגענו להתכנסות. בפועל, נשתמש באלגוריתם האיטרטיבי הבא:

$$V_{k+1} = BO(V_k) = R^\pi + \gamma T^\pi \cdot V_k$$

נסתכל על הדוגמא הבאה

$$T^\pi = \begin{pmatrix} 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 & 0 \\ 0 & 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.1 & 0.1 & 0.8 \end{pmatrix}, \mathcal{R}^\pi = \begin{pmatrix} 0.1 \\ 1.3 \\ 3.4 \\ 1.9 \\ 0.4 \end{pmatrix}, \gamma = 0.9$$

באמצעות השיטה האיטרטיבית נקבל:

$$V_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, V_1 = \begin{pmatrix} 0.1 \\ 1.3 \\ 3.4 \\ 1.9 \\ 0.4 \end{pmatrix}, V_2 = \begin{pmatrix} 0.6 \\ 2.6 \\ 6.1 \\ 3.7 \\ 1.2 \end{pmatrix}, \dots, V_{10} = \begin{pmatrix} 7.6 \\ 10.8 \\ 18.2 \\ 16.0 \\ 9.8 \end{pmatrix}, \dots, V_{50} = \begin{pmatrix} 14.5 \\ 17.1 \\ 26.4 \\ 26.8 \\ 18.4 \end{pmatrix}, V^\pi = \begin{pmatrix} 14.7 \\ 17.9 \\ 26.6 \\ 27.1 \\ 18.7 \end{pmatrix}$$

ניתן לשים לב שאחרי 50 איטרציות הפתרון המתתקבל בצורה האיטרטיבית=קרוב מאד לפתרון המתתקבל בצורה האנליטית.

Policy Iteration (PI)

חישוב ה-value function מאפשר לנו לחשב את ערכו של s עבור כל a , אך הוא אינטגרטיבי שנגיעה לאסטרטגיה האופטימלית=נניח והצלחנו לחשב אותה (s, a) π יוממך-Anנו יודעיף לגזורה אסטרטגיה, עדין יתפרק ימת פועלקה=משינויו של מושתלת מהשר הפעולה המוצעת לפי האסטרטגיה הנגזרת π - (s) π בפועל פורמלי ניתן לתאר זאת בצורה פשוטה – נניח שהיחסנו אותו (s, a) π יתכן וקיים פעללה עבורה

for such s, a : $\mathcal{Q}^\pi(s, a) > \mathcal{V}^\pi(s)$

אם קיימת פעולה כזו, אז ישתלבח בבחירה בה ורק לאחר מכן לזרז בהתאם לאסטרטגייה $(s|a)$ הנקראת מחישוב ה-value function. ניתן לחפש את כל הפעולות עבורן כדי לבצע פעולה מסוימת עבורו התגמול יהיה גבוה יותר מאשר האסטרטגייה שבלבאון פורמלי יותר, נרצה להגדיר אסטרטגיית דטרמיניסטית, עבורו בהסתברות 1 נקבע בכל מצב s פעולה ה称之为 a :

$$\pi'(s) = \arg \max_{a'} \mathcal{Q}^\pi(s, a')$$

נשים לב שרגע זה הוא בעצם להשתמש באסטרטגייה גרידית – בכל מצב לנוקוט בפעולה הכי משלימה-קבתו של צעד ייחד – השאלה היא מבוקש – מדוע זה בהכרח נכון? כמובן האם הרעיון של לבחור – בכל צעד את האופטימלי – בהכרח טוביל-לקבלת אסטרטגייה אופטימלית עבור כל הצעדיים כולם יחד? בכך להוכיח זאתணסח זאת ממשפה:

בהתבגרות אסטרטגייה π' , כאשר אדרטמיניסטיית איזה אסטרטגייה $\mathcal{V}^\pi(s) > \mathcal{V}^{\pi'}(s)$ בבחירה לכל-כך-יתק'י מ- π . ראשית נפתח לפני הגדרה:

$$\mathcal{V}^\pi(s) < \mathcal{Q}^\pi(s, \pi'(s)) = \mathbb{E}_\pi[r_{t+1} + \gamma \cdot \mathcal{V}^\pi(s_{t+1}) | s_t = s, a_t = \pi'(s)]$$

כיוון שהסטרטגייה הינה דטרמיניסטית, הפעולה הנבחרת אינה רנדומלית ביחס ל' π' , ולכן נוכל לרשום:

$$= \mathbb{E}_{\pi'}[r_{t+1} + \gamma \cdot \mathcal{V}^\pi(s_{t+1}) | s_t = s]$$

כעת לפני אותו אי שוויון שבנהנזה נוכל לבצע את אותו חישוב גם לצעד הבא: s_{t+2} :

$$< \mathbb{E}_{\pi'}[r_{t+1} + \gamma \cdot \mathcal{Q}^\pi(s_{t+1}, \pi'(s_{t+1})) | s_t = s]$$

זה שוו שווה –

$$= \mathbb{E}_{\pi'}[r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot \mathcal{V}^\pi(s_{t+2}) | s_t = s]$$

וכך הלאה, ולבסוף הוכחנו את הדרישה – נקיית הפעולה הכי עיליה בכל מצב תמיד תהיה יותר טובה מהפתרון של π . כתשיש בידינו שתי טכניקות שאנו יודעים לבצע:

$\mathcal{Q}^\pi(s, a)$ -Evaluation (E) – בהינתן אסטרטגייה מסוימת נוכל לפתור את משוואות בלמן ולקבל את $\mathcal{V}^\pi(s)$

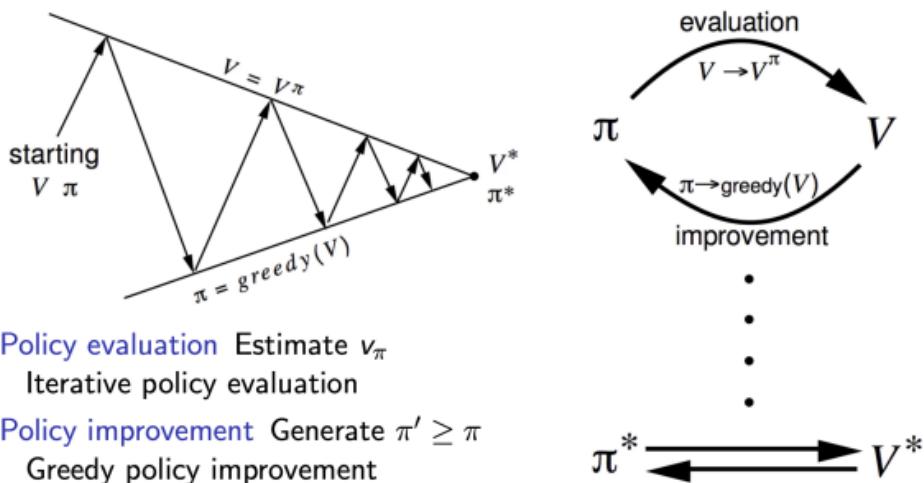
Improvement (I) – בהינתן הערך של value function של \mathcal{V}^π

ניתן להתחיל מאסטרטגייה רנדומלית, ואז לבצע איטרציות המורכבות משתי הטכניקות האלה באופן הבא:

$$\pi_0 \xrightarrow{E} \pi_1 \xrightarrow{I} \pi_2 \xrightarrow{E} \dots$$

תהליך זה נקרא **Policy iteration** – בכל צעד בו יש לנו אסטרטגייה נפתרו עבורה משוואות בלמן ובכך נחשב את ה-value function שלה, ולאחר מכן נשפר את האסטרטגייה באמצעות policy improvement, שכך מבעץ בחירה גרידית שבטווח הקצר טוביה יותר מאשר ה-value function – השיכרנו – ניתן להוכיח שאחרי מספר סופי של איטרציות האסטרטגייה תתכנס לנקודת שבות (fixed point), ואז הפעולה הבאה לפיה האסטרטגייה תהיה לבחירה הגדית:

$$\pi(s) = \arg \max_a \mathcal{Q}^\pi(s, a) = \pi'(s)$$



איך ביצוע איטרציות ש- V על מנת למצוא בכל שלב את ה-value function ולשפר אותו באמצעות בחירה גרידית?

Bellman optimality equations

השלב הבא בשימוש ב- π -Bellman optimality equation הוא להוכיח שהאסטרטגיהالية מתכנסים הינה אופטימלית. נסמן את נקודת השובב- π ונקבל את הקשר הבא:

$$V^{\pi^*}(s) \equiv V^*(s) = \max_a Q^*(s, a) = \max_a \sum_{s'} p_\pi(s, s') \left(R_\pi(s, s') + \gamma \cdot V^*(s') \right)$$

ובאופן דומה:

$$Q^*(s, a) = \sum_{s'} p_\pi(s, s') \left(R_\pi(s, s') + \gamma \cdot \max_{a'} Q^*(s', a') \right)$$

משוואות אלה נקראות Bellman optimality equation. ניתן לשים לב שהן מאוד דומות למשוואות בלמן מהן ייצנו, אך במקומם התוחלת שהייתה לנו בהתחלה, כעת יש \max להראות שהפתרון של משוואות אלה הוא Value- π -Bellman optimality equation. ננסה את הטענה באופן הבא:

אסטרטגיה הינה אופטימלית אם ורק אם היא מקיימת את Bellman optimality equation. כיוון אחד להוכחה הוא טריויאלי – אם האסטרטגיה הינה אופטימלית אז היא בהכרח מקיימת את משוואות האופטימליות, כיוון שהאריך שהן מתקבלות מנקודת החשבת אליה ה- π -Bellman optimality. אם האסטרטגיה לא הייתה אופטימלית אז היה ניתן לשפר עוד את האסטרטגיה ולא היינו מגאים עזין לנקודת החسبת-בשביל להוכיח את הכוון החדש-בשותמש שוביון של העתקה מכווצת. נגדיר את האופרטור הבא:

$$BV(s) = \max_a \sum_{s'} p_\pi(s, s') \left(R_\pi(s, s') + \gamma \cdot V(s') \right)$$

ניתן להראות שאופרטור זה הינו העתקה מכווצת, וממילא לפי המשפט של בנך יש לו נקודת שבת יחידה – כיוון שהראינו שימוש ב- π -Bellman optimality equation. מכאן ניתן לומר את העובדה שהאופרטור שהגדכנו הינו העתקה מכווצת וממילא קיבל שאותה נקודת שבת הינה יחידה, וממילא אופטימלי.

Value Iteration

הראנו שבעזרת שיטת Policy iteration ניתן להגיע לאסטרטגיה אופטימלית, אך התחילה יכול להיות איטרצית. ניתן לנוקוט גם בגישה יותר ישירה ולנסות לחשב באופן ישיר את הפתרון של משוואות האופטימליות של בלמן (ופתרון הינו אופטימלי כיוון שהראינו שהפתרון הוא נקודת שבת יחידה). נתחיל עם פתרון רנדומלי – ולאחר מכן נקבע איטרציות באופן הבא עד שנגיע להתקנסות:

$$\mathcal{V}_{k+1} = \max_a \sum_{s'} p_\pi(s, s') (\mathcal{R}_\pi(s, s') + \gamma \cdot \mathcal{V}_k(s'))$$

נשים לב שבשיטת זו אין לנו מידע לגבי האסטרטגיה אלא רק חישבנו את ה-Value function, אך ממנה ניתן לגוזה את Q ואז לבחור באסטרטגיה גרידית, שהינה במקורה זה גם אופטימלית

$$\pi(s) = \arg \max_a Q^\pi(s, a)$$

ניתן להראות כי בשיטה זו ההתקנסות מהירה יותר ודרושים פחות איטרציות מהשיטה הקודמת, אך כל איטרציה יותר מורכבת

Limitations

לשתי השיטות – Policy iteration ו-Value iteration – יש שני חסרונות מרכזיים:

1. הן דורותות לדעת את המודול והסיבבה באופן שלם ומדויק

2. הן דורותות לעדכן בכל שלב את כל המ מצבים בו זמן. עבור מערכות עם הרבה מצבים, זה לא מעשי.

11.1.3 Learning Algorithms

בפרק הקוד הקודם כיצד ניתן לחשב אופטימלית וערוך החזרה**בהתאם** מודל מרקובי-השתמש בשתי הנחות עיקריות על מנת להתמודד עם הבעיה

1. Tabular MDP – הנחנו שהבעיה סופית ולא גדולה מדי, כך שנוכל ליצג אותה בזכרון ולפתרו אותה.

2. Known environment – הנחנו שהמודול ידוע לנו, כולל הנזונה לחומריצת המעברים שקובעת מה הסיכוי לעבור מצב= s מצב= s' כשלוקטים בעולקה= $\mathcal{P}_{ss'}^a$ (סימנו את זה בתווך= (s, s') , ובנוסף נתנו לנו מה ה- r -reward המתקיים עבור כל action (s) סימנו את זה בתווך= (s, a) $\mathcal{R}_{ss'}^a = \mathcal{R}_{ss'}(s, a)$

בעזרת שתי הנחות פיתחנו את המשוואות בלבד, כאשר הינו לנו שני צמדים של משוואות. משוואות בלבד מוגדרות באמצעות אסטרטגיה נתונה כתובות באופן הבא

$$\mathcal{V}^\pi(s) = \sum_{a, s'} \pi(a|s) \mathcal{P}_{ss'}^a (\mathcal{R}_{ss'}^a + \gamma \cdot \mathcal{V}^\pi(s'))$$

$$Q^\pi(s, a) = \sum_{s'} \mathcal{P}_{ss'}^a \left(\mathcal{R}_{ss'}^a + \gamma \sum_{a'} Q^\pi(s', a') \right)$$

ובנוסף פיתחנו את המשוואות עבור הפתרון האופטימלי:

$$\mathcal{V}^*(s) = \max_a \sum_{s'} \mathcal{P}_{ss'}^a (\mathcal{R}_{ss'}^a + \gamma \cdot \mathcal{V}^*(s'))$$

$$Q^*(s, a) = \sum_{s'} \mathcal{P}_{ss'}^a \left(\mathcal{R}_{ss'}^a + \gamma \max_a Q^*(s', a') \right)$$

הראינו שתי דרכי להגעה לפתרון האופטימלי:

1. Policy improvement – המרכיב מ-**Policy evaluation** ולאחר מכן **Policy improvement**.

2. Value iteration – פתרון משוואות בלבד באמצעות יישיר בעזרת איטרציות על ה-Value function.

כאמור – דרכי פתרון אלו מניחים שהמודול ידוע, ובנוסף שמדובר במצבים איננו גדול מדי וכי יכול להיות מוצג בזכרון האטגר האמי-תיכ-מתחיל בנקודה בטלפון אחת מהנחות אלה אינה תקיפה, ולמעשה פרה-מתחיל התפקיד של אלגוריתמי RL. עיקר ההתקנות של אלגוריתטם אלו היה למצאו באופן יעיל את האסטרטגיה האופטימלית כאשר לא נתונים הפרמטרים של המודול, וזאת לשערך אוטומטי= $Model-based learning$ או למצוא דרך אחרת לחישוב האסטרטגיה האופטימלית ללא שימוש במודל ($Model free learning$). **Model-based learning** יש משחק בין משתמש בלבד המחשב, אלגוריתמי-השיניים ל-**Model based learning** – פינסו ללמידה של המודול של המשחק או להשתמש במודול

קיי-פְּרִזְבֶּעָדֶרֶת המודל הם ינסכלבחן $\text{Citz}=$ יגיב המשמש לכל תור שהמחשב יבחר. לעומת זאת אלגוריתמים מסווג Model free learning לא יתעניינו בכך, אלא ינסו ללמד ישרות את האסטרטגיה הטובה ביותר עבור המחשב

היתר-המשמעות של אלגוריתמים המשמשים על המודל של הבעיה (Model-based) נובע מהיכולת לתקן מסקנה צעדים קדימה, כאשר עבר כל בחירה של פעולה המודל בוחן את התוצאות האפשרות, את הפעולות המתאימות לפחות תגובה, וכן הלהה=דוגמא מפורסמת לכך היא תוכנת המחשב AlphaZero =שאומנה לשחק משחקי לוח כגון שחמט או גו. במקרים אלו המודל הוא המשחק והחוקים שלו, והתוכנה משתמשת בידע זהה בכך כדי לבחון אפקט הפעולות והתגובה למשך מספר צעדים רב ובחירה של הצעד הטוב ביותר

עם זאת=בדרך כל-אך בשלב האמצעי לסוקן מידעה=יצנו מהחץ הנכון באופן אולטימטיבי, ועליו ללמידה רק מהניסיין. עובדה זו מצבה כמה אתגרים, כאשר העיקרי ביניהם הואהסטרטגיה הנלמדת תהיה טובה רק עבור המקרים אותם ראה הסוקן, אך לא תאים למקירב=חדשים שיבואו=אלגוריתמים שמחפשים באופן ישיר את האסטרטגיה האופטימלית אמםם לא משתמשים בידע שיכול להגיעה מבחן צעדים עתידיים, אך הם הרבה יותר פשוטים למימוש ולאימוץ

באופן מעט יותר פורמלי ניתן לנוכח את ההבדל ביחס=ישות:Citz=learning מנוסה למצוא את הפרמטרים המגדירים את המודל $\{\mathcal{R}, \mathcal{T}, \mathcal{A}\}$ =ואז בעזרת חישוב את האסטרטגיה האופטימלית (למשל בעזרת משוואות בלמן). הגישה השנייה לעומת זאת לא מעוניינה=לחישוב במפורש את הפרמטרים של המודל אלא למצוות באופן ישיר את האסטרטגיה האופטימלית $a_t | s_t$ =אשר כל מצב קבוע באיזה פועל להקוט=ההבדל בין הגישות נוגע גם לפונקציית המחיר לה נרצה למצוא אופטימום

בכל אחד משני סוג הלמידה יש אלגוריתמים שונים, כאשר הם נבדלים אחד מהשני בשאלת מההאובייקט=אוזן מעוניינים ללמידה.

Model-free learning

בגישה זו יש שני קטגוריות מרכזיות של אלגוריתמים:

- א. $\text{Q}=\text{Policy Optimization}$ – ניסוח האסטרטגיה כבעית אופטימיזציה של מציאת סט הפרמטרים θ =המ מקס-פה או $(a|\pi)$ =פתרון בעיה זו יכול להישנות באופן ישיר על יד-שיטות Gradient Ascent=מעבר פונקציית המchia-פ $\mathbb{E}[R(\tau)] = \mathbb{E}[\pi(\tau)]$, או בעזרת קירוב פונקציה זו ומציאת מקסIMUM עבורה
- ב. Q-learning =שערוך $=Q(s, a)$ =על יד- $Q_\theta(s, a)$ =מציאת המשערך=האופטימלי יכול להתבצע על ידי חיפוש=שים פק את השערוף הטוב ביחס לשני למסואו=על יד-מציאת הפעולה=שתמוקם את המשערך: $a(s) = \arg \max_a Q_\theta(s, a)$

השיטות המנוסות למצוא אופטימום לאסטרטגיה הן לרובי=policy-on, כולם כל פעולה נקבעת על בסיס האסטרטגיה המעודכנת לפי הפעולה הקודמת. Q-learning – לעומת זאת הוא לרובי אלגוריתם off-policy, כלומר בכל פעולה ניתן להשתמש בכל המידע שנמצא עד כה=היתרון של שיטות האופטימיזציה=נובע מכך שהן מנוסות למצואו ישיר או האסטרטגיה הטובה ביחס= Q-learning =ה- Q -השערך=Af $(a, s)^*$, ולעתים השעריך לאמפסיך ואז התוצאה המתבקשת איננה מספיקת טובה. מצד שני=כasher השעריך מוצלח, הביצועים של Q-learning=ה- Q -השערך=Af (a, s) , ויתר, כיוון שהשימוש ב- Q מידע על העבר מונצץ בזורה עיליה יותר מאשר באלגוריתם ש- Q -learning=המבצעים אופטימיציה של האסטרטגיה=שתי הגישות האלה אינן דומות לחלוון, וישנם אלגוריתמי=p-שמנטים לשלב בין הרוונות ונצל את החזקות והיתרונות שיש לכל גישה

Model-based learning

גם בגישה זו יש שתי קטגוריות מרכזיות של אלגוריתמים:

- א. $\text{Model-based RL with a learned model}$ =אלגוריתם מהמנסים ללמידה אין את המודל עצמה=ה- π או את האסטרטגיה π .
- ב. $\text{Model-based RL with a known model}$ – אלגוריתמים המנסים למצוא את ה- π = Value function האסטרטגיה כאשר המודל עצמו נתן

הבדל בין הקטגוריות=טמן באציג אותו מנסים להתמודד=במקרים בהם המודל ידוע, הממד של אי הוודאות לא קיים , ולכן ניתן להתמקד בביטויים אסימפטומטיים. במקירב=בهم המודל אינו ידוע, הדגש העיקרי הוא על למידת המודל