## ארכיטקטורת המודל:

אופן הצגת המידע:

ראשית נוסיף לכל השמות את התו '$' המבטא את סוף המילה (END_TOKEN). ע"י כך נאפשר למודל לחזות את סיום המילה. האותיות האפשריות במודל שלנו הינן A-Z, a-z (כל האותיות האנגליות), וכן END_TOKEN. לצורך קידוד השמות, נבצע קודם לכן קידוד של האותיות. נקודד כל אות כוקטור one hot encoding באורך 53 (כמספר האותיות הכולל). ע"י כך, נוכל לקודד שם כמטריצה אשר שורותיה מורכבות מוקטורי one hot encoding (מספר השורות הינו כמספר האותיות בשם).

שכבות

על מנת לתפוס את הדיאלקטים והניואנסים הייחודיים לכל שפה באופן מיטבי, נגדיר לכל שפה שכבות fully connected עצמאיות משלה אשר פועלות באופן בלתי תלויות משכבות שאר השפות. על מנת לגשת לשכבות המתאימות של כל שפה נגדיר לכל שכבה מילון הממפה בין שפה לבין השכבה המתאימה. לכל שפה נגדיר שתי שכבות fully connected הפועלות בנפרד זו מזו (מבנה השכבות עצמו זהה בין השפות).

שכבת הקלט – בשלב t שכבה זו מקבלת כקלט את הוקטורים הבאים:

- letter_input - מבטא את האות שנקבל כקלט בזמן t (אורכו 53 כמספר האותיות הכולל).
- hidden_input – מבטא אינפרומציה מן הקלטים בשלבים 1, 2, ... , t-1 (אורכו 500 כגודל השכבה הנסתרת).

נאחד אותם לכדי וקטור יחיד באורך 553 (גודל השכבה הנסתרת + מספר האותיות הכולל). נסמנו combined. איחוד זה מבטא התייחסות לזמן הנוכחי וכן התייחסות לזמני העבר.

שכבת i2h – שכבה זו מכילה 500 נוירונים (גודל השכבה הנסתרת) ווקטור הקלט שלה הינו וקטור combined. לאחר חישוב ערך הצירוף הלינארי בין משקולות כל נוירון בשכבה זו לבין ערכי וקטור combined, נקבל וקטור באורך 500. נסמן וקטור זה ב-hidden. וקטור הפלט hidden בשלב t יהיה וקטור hidden_input בשלב t+1. כן נוכל להעביר את האינפורמציה הנוכחית לזמן עתידי.

שכבת i2o – שכבה זו מכילה 53 נוירונים (כמספר האותיות הכולל) ווקטור הקלט שלה הינו וקטור combined. לכן לאחר ערך הצירוף הלינארי בין משקולות כל נוירון בשכבה זו לבין ערכי וקטור הקלט, נקבל וקטור באורך 53. נסמן וקטור זה בoutput. לצורך יצירת פרדיקציה על האות בשלב t+1 נפעיל את פונקציית logsoftmax אשר משרה ערכים הסתברותיים על ערכי output (לצורך חיזוי מפורש של האות נבחר באות בעלת ההסתברות הגבוהה ביותר).

באופן מופשט, בשלב t המודל מקבל כקלט letter_input, hidden_input, language ומחזיר כפלט:

- output – מבטא את ההסתברות של כל אות להיות בשלב t+1.
- hidden – מבטא את וקטור hidden_input בשלב t+1.

(הקלט language משמש לצורך בחירת סט המשקולות המתאים לשפה ע"י שימוש במילונים).

אימון המודל

לצורך אימון המודל וביצוע backpropagation, נבחר בפונקציה Cross Entropy Loss (multiclass) גרסת הסכום (נפרט על כך בהמשך), כלומר זוהי הפונקציה אותה אנו מנסים למזער בשלב האימון. לצורך ביצוע מטרה זו, נבצע שיטת גרדיאנט עם קצב למידה של 0.0025.

מספר סבבי האימון שנקבע הינו 100. בכל סבב אימון נעבור על כל השפות. עבור כל שפה, נגריל 50 שמות באופן אקראי (ללא חזרות) מסט האימון שלה. עבור כל שם נבצע את הפעולות הבאות:

בהינתן האות ה-t של השם נחזה את האות ה-t+1 (ע"י הכנסת האות t, hidden_input ושפת השם למודל). לאחר מעבר כל אותיות השם, נחשב את ה-loss בין הפרדקציות ההסתברויות של המודל לבין האותיות האמיתיות (בייצוג הוקטורי שלהן). מכיוון שהגדרנו את פונקציית ה-loss בגרסת הסכום, הערך שנקבל הינו ערך ה-loss של המודל על השם כולו. לאחר מכן נבצע backpropagation לצורך עדכון המשקולות בהתאם.

מספר הערות:

1. בעת הכנסת האות הראשונה נאתחל את hidden_input להיות וקטור אפסים (אין אינפורמציה מן העבר).
2. עבור האות הראשונה בשם שנחזה, נאתחל אותה כאות הראשונה של השם עליו אנו עוברים (כאשר נייצר שמות האות הראשונה תהיה נתונה לנו ולכן לא נצטרך לחזות אותה).
3. בעת מעבר על אותיות השם לא נצטרך לעבור על האות האחרונה שכן לא חוזים ממנה אף אות.
4. עבור שם נתון, קלט השפה אשר נכניס למודל זהה בין כל אותיות השם.

**בחירת hyperparameters של המודל:**

בחירת קצב הלמידה: קצבי הלמידה שנבדקו הינם 0.0001, 0.00025, 0.0003, 0.001.
תחילה נבדק הערך 0.001. במספר סבבי האימון הראשונים ערך פונקציית ה-loss אכן קטן אך לאחר מכן הוא החל לגדול בקצב מהיר. בעקבות כך הסקתי כי קצת למידה זה הינו גבוה מידי וכי עלי להקטין אותו בסדר גודל. לאחר מכן נוסה קצב האימון 0.0001. קצב זה אכן הצליח להוביל לירידת ערך פונקציית ה-loss באופן עקבי אך הירידה הייתה איטית מאוד. לכן הגדלתי את קצב הלמידה על מנת להוביל לירידה מהירה יותר בערך הפונקציה. הערכים 0.00025, 0.0003 הובילו לתוצאות דומות אם כי הערך 0.00025 הוביל לתוצאה טובה יותר ולכן לבסוף בחרתי בו.

גודל כל סבב אימון ובחירת מספר סבבי האימון: מספר השמות בסט האימון שונה בין שפה לשפה. מכיוון שהגדרתי שכבות בלתי תלויות לכל אחת מן השפות רציתי לאמן אותן באופן שווה בכל סבב אימון. בנוסף, רציתי לבחור מספר שמות שיהיה קטן ממספר השמות הכולל של כל שפה. לבסוף, החלטתי שבכל סבב אימון, נתאמן על 50 שמות מכל שפה. מספר סבבי האימון שבחרתי הינו 100. ערך זה מבטא איזון בין הצלחת מזעור פונקציית ה-loss (כפי שניתן לראות בגרף המצורף בהמשך) לבין משך זמן האימון.

בחירת גודל השכבה הנסתרת: הערך ההתחלתי אשר בחרתי הינו 500. ערך זה מבטא איזון בין בחירת ערך קטן מידי (לא יצליח ללמוד את הקשרים המורכבים בשפה) ולבין ערך גדול מידי (יבטא over-fitting וכן יאריך את זמן אימון המודל). ערך זה הניב תוצאות טובות ולכן החלטתי להשאירו.

## תוצאות המודל המאומן

לאחר סיום שלב האימון נרצה לחזות שמות. לצורך כך נשתמש במודל המאומן האופן הבא:

הקלט המתקבל ע"י המשתמש הינו אות ושפה. נכניס קלט זה למודל המאומן על מנת לקבל פרדיקציה על האות הבאה של השם. נוסיף את הפרדיקציה לשם הזמני שמיוצר, ולאחר מכן נתייחס אליה כאל אות הקלט הבאה בשם אותו ננסה לחזות. נמשיך בתהליך זה עד אשר המודל חזה END_TOKEN המבטא את סוף המילה. לבסוף נחזיר את השם הנוצר למשתמש (לאחר הסרת END_TOKEN מהשם).

*עבור כל אחד משלבי פרדיקצית האותיות, השפה המוכנסת למודל הינה קלט השפה שקיבלנו מן המשתמש.

להלן חמש דוגמאות לשמות שונים שהמודל ייצר:

1. הקלט שהוכנס הינו English, 'A', והשם שהתקבל הינו 'Alle'.
2. הקלט שהוכנס הינו Vietnamese, 'L', והשם שהתקבל הינו 'Lac'.
3. הקלט שהוכנס הינו Arabic, 'M', והשם שהתקבל הינו 'Malouf'.
4. הקלט שהוכנס הינו Chinese, 'Z', והשם שהתקבל הינו 'Zhan'.
5. הקלט שהוכנס הינו Korean, 'H', והשם שהתקבל הינו 'Hon'.

בנוסף מצורף גרף של ערך פונקציית Cross Entropy Loss (multiclass) כפונקציה של מספר סבבי האימון (ערך פונקציית ה-loss תחת סבב אימון הינו ערך ה-loss הממוצע של שם בסבב האימון).