Activation Functions

2:28 PM Wednesday, September 18, 2024

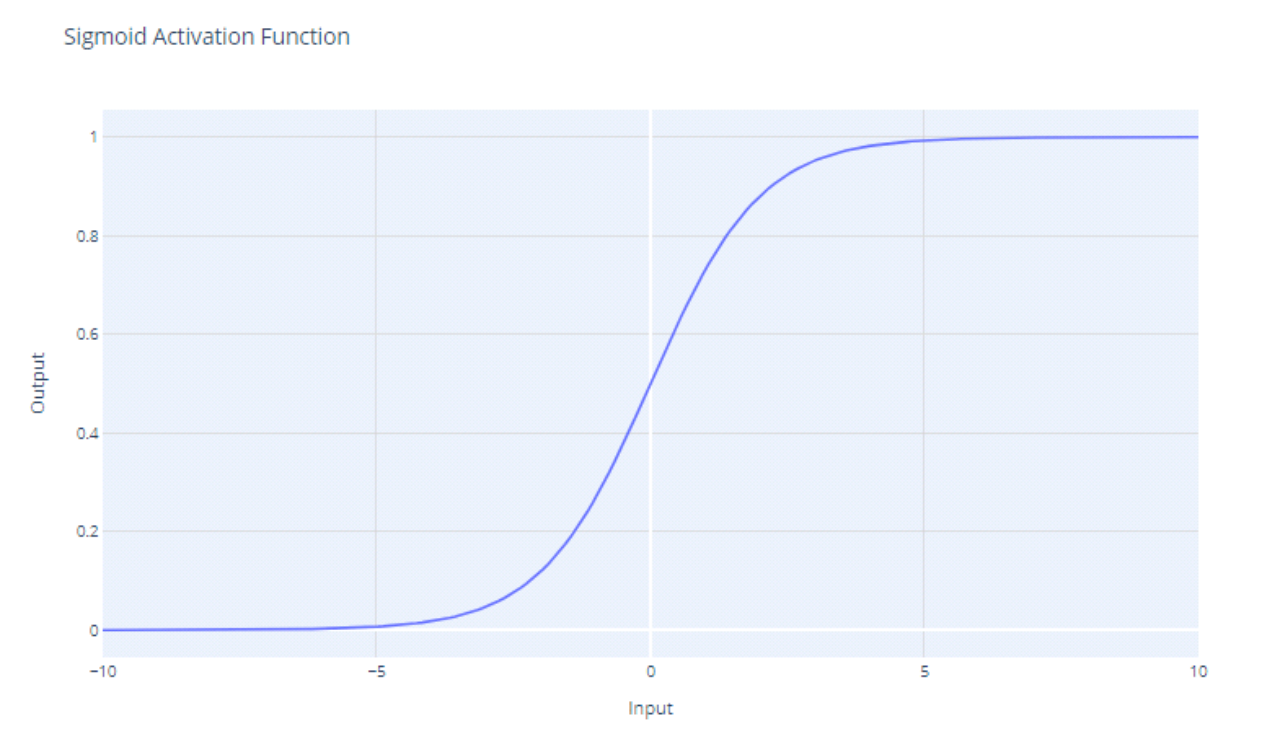
מה הם Activation Functions? -

Activation Functions הם אבני בניין של רשתות נוירונים המאפשרות להן ללמוד דפוסים מורכבים בנתונים. הם הופכים את אות הכניסה של צומת ברשת עצבית לאות פלט שמועבר לאחר מכן לשכבה הבאה.
ללא פונקציות הפעלה, רשתות עצביות יהיו מוגבלות ליצירת מודלים רק של יחסים ליניאריים בין קלטים ופלטים.

לדוגמה, נניח שיש לנו משימה לזהות תמונות של כלבים וחתולים. אם נשתמש במודל ליניארי פשוט, לא נוכל לתפוס את המורכבות של התבניות בתמונות, כמו זנבות, אוזניים או פרווה. במצב זה, המודל שלנו יוכל רק לחלק את התמונות לקבוצות על בסיס קווים ישרים, מה שמוביל לטעויות בזיהוי.
דוגמא נוספת, הקשרים בין מחירי הדירות לגודלם אינם ליניאריים, משמע שלא תמיד בית גדול יותר יהיה יקר יותר, זה תלוי במיקום הנכס, האם הוא משופץ וכו' ... אם לרשתות עצביות לא היו Activation Functions, הן לא היו מצליחות ללמוד את התבניות הלא-ליניאריות המורכבות הקיימות בנתונים מהעולם האמיתי.

Activation Functions מציגות אי-ליניאריות, המאפשרות לרשתות עצביות ללמוד מיפויים מורכבים ביותר בין קלט ופלט.
בחירת פונקציית ההפעלה הנכונה היא חיונית לאימון רשתות עצביות שמכלילות היטב ומספקות תחזיות מדויקות יותר.

Sigmoid Activation -



פונקציית ההפעלה Sigmoid, המיוצגת לעתים קרובות כ-σ(x), היא פונקציה חלקה, ולה יש חשיבות היסטורית בפיתוח רשתות הנוירונים. לפונקציית ההפעלה Sigmoid יש את הנוסחה הבאה:

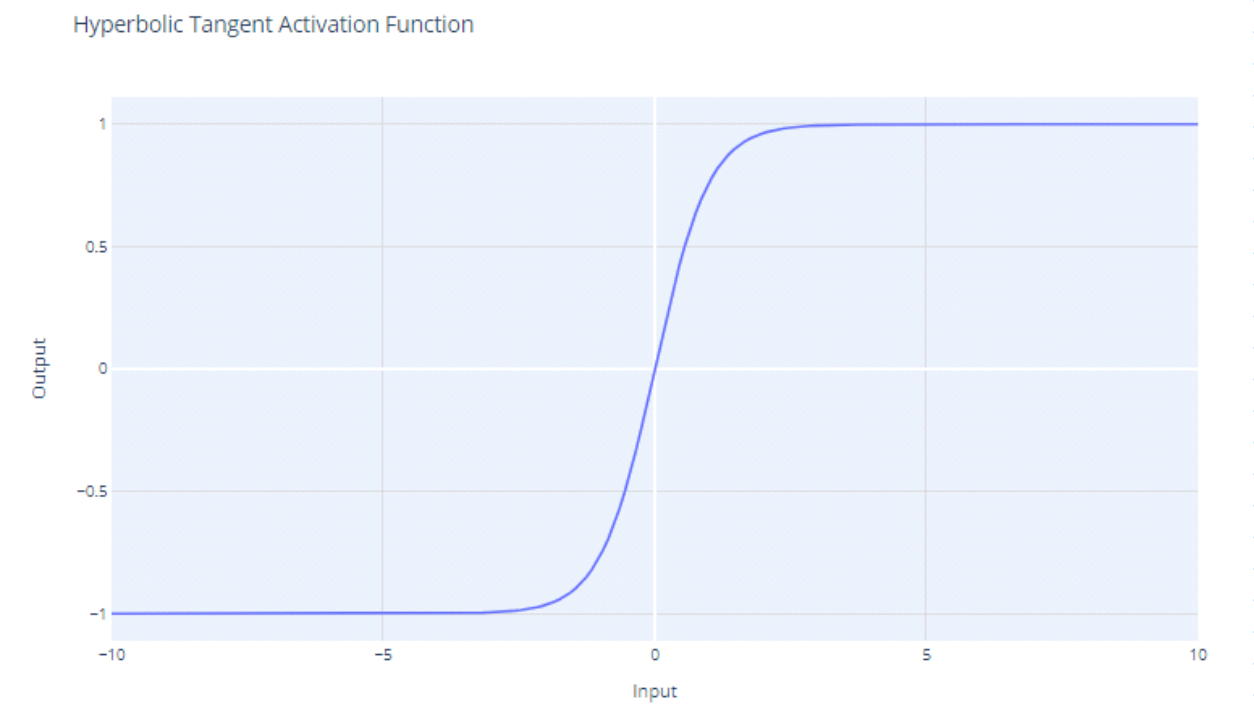
$$f(x) = \frac{1}{(1 + e^{-x})}$$

היא לוקחת קלט בעל ערך אמיתי "ומועבת" אותו לערך שבין 0 ל-1. לפונקציה Sigmoid יש עקומה בצורת "S" שיש לה אסימפטוטות ב-0 עבור מספרים שליליים גדולים ו-1 עבור מספרים חיוביים גדולים. ניתן לפרש בקלות את הפלטים כהסתברויות, מה שהופך אותו לטבעי לבעיות סיווג בינארי, לדוגמא האם הודעת הדוא"ל שנשלחה היא ספאם או לא - תשובה של כן או לא.

יחידות Sigmoid היו פופולריות ברשתות עצביות מוקדמות מכיוון שה-Gradient הוא החזק ביותר כאשר הפלט של היחידה קרוב ל-0.5, מה שמאפשר אימון יעיל ל-Backpropagation. לעומת זאת, יחידות Sigmoid סובלות מבעיית "Vanishing Gradient" הפוגעת בלמידה ברשתות עצביות עמוקות.

"Vanishing Gradient" זה כאשר ערכי הקלט הופכים לחיוביים או שליליים באופן משמעותי, הפונקציה מלאה בערכים 0 או 1, עם שיפוע שטוח במיוחד. באזורים אלה, השיפוע קרוב מאוד לאפס. זה גורם לשינויים קטנים מאוד במשקלים במהלך ה-Backpropagation, מה שגורם ללמידה להאט מאוד או אפילו לעצור אותה. זה מכונה בעיית "Vanishing Gradient" ברשתות נוירונים.

- Tanh (Hyperbolic Tangent) Activation



ה-Tanh Activation Functions מוגדרת כך:

$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

הפונקציה Tanh מוציאה ערכים בטווח של 1- עד 1+. המשמעות היא שהוא יכול להתמודד עם ערכים שליליים בצורה יעילה יותר מאשר הפונקציה Sigmoid, שיש לה טווח של 0 עד 1.

בניגוד לפונקציה Sigmoid, Tanh הוא Zero-Centered, כלומר הפלט שלו סימטרי סביב המקור של מערכת הקואורדינטות. זה נחשב לעתים קרובות ליתרון מכיוון שהוא יכול לעזור לאלגוריתם הלמידה להתכנס מהר ובכך למודל ללמוד מהר יותר.

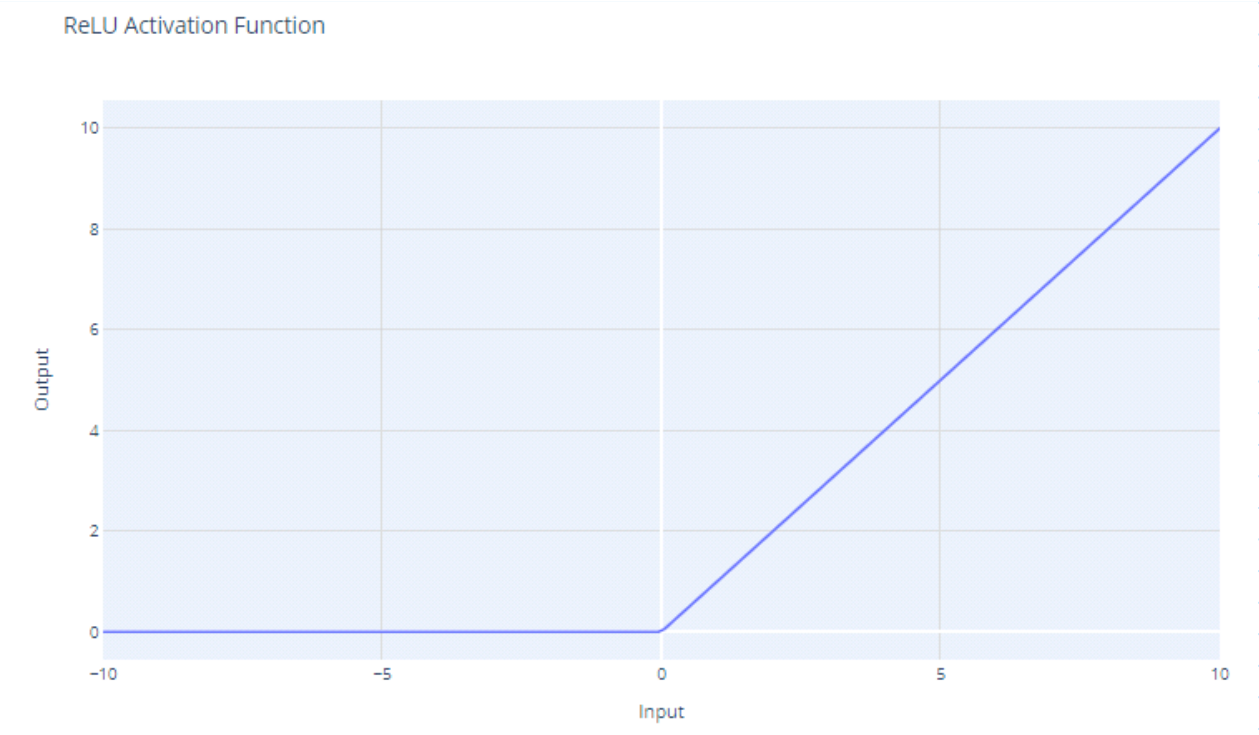
מכיוון שהפלט של Tanh נע בין 1- עד 1+, יש לו Gradient חזקים יותר מהפונקציה Sigmoid. שיפועים חזקים יותר מביאים לעיתים קרובות ללמידה והתכנסות מהירה יותר במהלך האימון מכיוון שהם נוטים להיות עמידים יותר מול בעיית שיפועים נעלמים בהשוואה לשיפועים של הפונקציה Sigmoid.

למרות היתרונות הללו, פונקציית ה-Tanh עדיין סובלת מבעיית ה-"Vanishing Gradient".

פונקציית Tanh משמשת לעתים קרובות בשכבות הנסתרות של רשת עצבית.

אם צריך לבחור בין Sigmoid ל-Tanh ואין סיבה ספציפית להעדיף אחד על פני השני, Tanh היא לרוב הבחירה הטובה יותר בגלל הסיבות שהוזכרו לעיל. עם זאת, ההחלטה יכולה להיות מושפעת גם ממקרה השימוש הספציפי ומההתנהגות של הרשת במהלך ניסויי ההכשרה הראשוניים.

- RELU (Rectified Linear Unit) activation



ה-RELU Activation Functions מוגדרת כך:

$$f(x) = max(0, x)$$

הוא מסמן את הקלט באפס, ומחזיר 0 עבור ערכים שליליים ואת הקלט עצמו עבור ערכים חיוביים.

עבור קלטים גדולים מ-0, RELU פועלת כפונקציה ליניארית עם שיפוע של 1. משמעות הדבר היא שהיא אינה משנה את סולם הקלטים החיוביים ומאפשרת לשיפוע לעבור ללא שינוי במהלך Backpropagation. תכונה זו היא קריטית בהפחתת בעיית ה-"Vanishing Gradient".

למרות ש-RELU הוא ליניארי עבור מחצית ממרחב הקלט שלו, מבחינה טכנית זוהי פונקציה לא ליניארית מכיוון שיש לה נקודה לא ניתנת להבדלה ב-x=0, שם היא משתנה בפתאומיות מ-x. אי-ליניאריות זו מאפשרת לרשתות עצביות ללמוד דפוסים מורכבים.

הפונקציה RELU היא זולה מבחינה חישובית מכיוון שהיא כוללת סף פשוט באפס. זה מאפשר לרשתות להתרחב לשכבות רבות ללא עלייה משמעותית בנטל החישובי, בהשוואה לפונקציות מורכבות יותר כמו Tanh או Sigmoid.

Used in layer	Activation Function	Details	Pros	Cons
Hidden / Output	**Sigmoid**	$\sigma(x) = 1/(1+e^{-x})$ Output range: [0,1]	- Smooth activation that outputs values between 0 and 1, making it suitable for binary classification - Historically popular	- Vanishing gradient problem due to saturated neurons - Output not zero-centered as sigmoid outputs are always positive - exp() operation is computationally intensive
Hidden	**Tanh (Hyperbolic tangent)**	$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ Output range: [-1,1]	- Zero centered outputs that help networks train faster	- Suffers with vanishing gradient problem when saturated
Hidden	**ReLU** **(Rectified Linear Unit)**	$f(x) = max(0,x)$ Output range: [0, ∞)	- Does not saturate: avoids vanishing gradient issues for positive inputs - Computationally efficient since only certain number of neurons are activated at the same time - Much faster convergence compared to sigmoid/tanh	- Output not zero-centered - Prone to a "dying ReLU" problem, where neurons can get stuck during training and never activate again, leading to a dead neuron that doesn't update its weights. © AIML.com Research