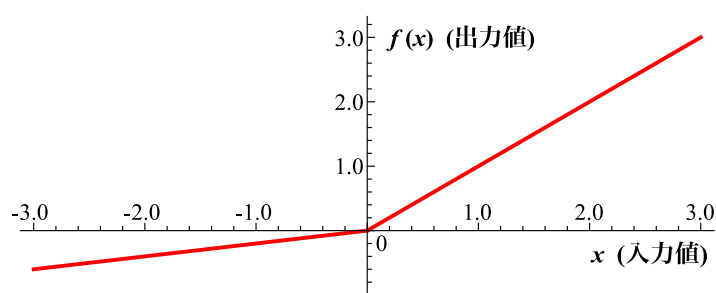


LeakyReLU関数

LeakyReLU (リーキーレルー) 関数は、次の式で定義される関数です。

$$f(x) = \begin{cases} x & (x > 0) \\ \alpha x & (x \leq 0) \end{cases} \quad [\alpha \text{は定数, 基本的には} \alpha = 0.01 \text{とする}]$$

なんだか難しそうな名前の関数ですが、単に傾きの異なる2つの直線を $x = 0$ でくっつけただけの関数です。



LeakyReLU関数 (見やすくするために、 $\alpha = 0.2$ としています)

♣ 名前の由来

LeakyReLU関数は、ReLU関数の拡張版です。「Leaky (漏れている)」という言葉の通り、ReLU関数では入力値 x が0以下の場合は出力値が常に0となりますが、LeakyReLU関数では入力値が0より小さい場合、出力値は下に漏れ出す (leak) ように0より下の値となります。

♣ ReLU関数との比較

現在のニューラルネットワークでは、隠れ層 (中間層) の活性化関数としてReLU関数を用いることが一般的となっていますが、より良い結果を求めて考えられたのが、このLeakyReLU関数です。いくつかの論文では、LeakyReLU関数を用いた方が精度が良くなると報告されていますが、あまり精度の向上が見られないと結論づけている論文もあります。実際に使用する際には、最適な α 値 (基本的には0.01) を試行錯誤しながら探しだすことで、精度を向上させることができると考えられます²。

² α 値を固定せず、学習で決定する場合は、特に PReLU (Parametric ReLU) 関数と呼ばれます。