$$ReLU(x) = \max(0, x)$$

Isso significa que:

- Se a entrada x for **maior que zero**, a saída será o próprio x.
- Se a entrada x for **menor ou igual a zero**, a saída será **zero**.

Características Principais da ReLU:

- **Não-linearidade:** Apesar de parecer linear por partes, a função ReLU é não-linear, o que é crucial para permitir que redes neurais aprendam padrões complexos.
- **Eficiência Computacional:** É muito mais rápida de calcular do que funções como Sigmoide ou Tanh, pois envolve apenas uma operação de comparação (max) em vez de exponenciais.
- **Mitigação do Problema do Gradiente Vanishing:** Ao contrário das funções Sigmoide e Tanh (que "espremem" os valores entre 0 e 1 ou -1 e 1), a ReLU não satura para valores positivos grandes. Isso ajuda a evitar que os gradientes se tornem muito pequenos durante a retropropagação, acelerando o treinamento de redes profundas.
- **Problema do "Dying ReLU":** Uma desvantagem potencial é que, para entradas negativas, o gradiente da ReLU é zero. Se um neurônio sempre produzir uma entrada negativa (por exemplo, devido a um grande bias negativo), ele pode "morrer" e nunca mais ser ativado, independentemente da entrada, pois seu gradiente será sempre zero, e seus pesos nunca serão atualizados. Variantes como Leaky ReLU ou ELU foram desenvolvidas para lidar com isso.

Exemplo de Cálculo:

1. **Se a entrada $x = 5$:**
   $ReLU(5) = \max(0, 5) = 5$
2. **Se a entrada $x = -3$:**
   $ReLU(-3) = \max(0, -3) = 0$
3. **Se a entrada $x = 0$:**
   $ReLU(0) = \max(0, 0) = 0$

Aqui está uma imagem que ilustra o gráfico da função ReLU: