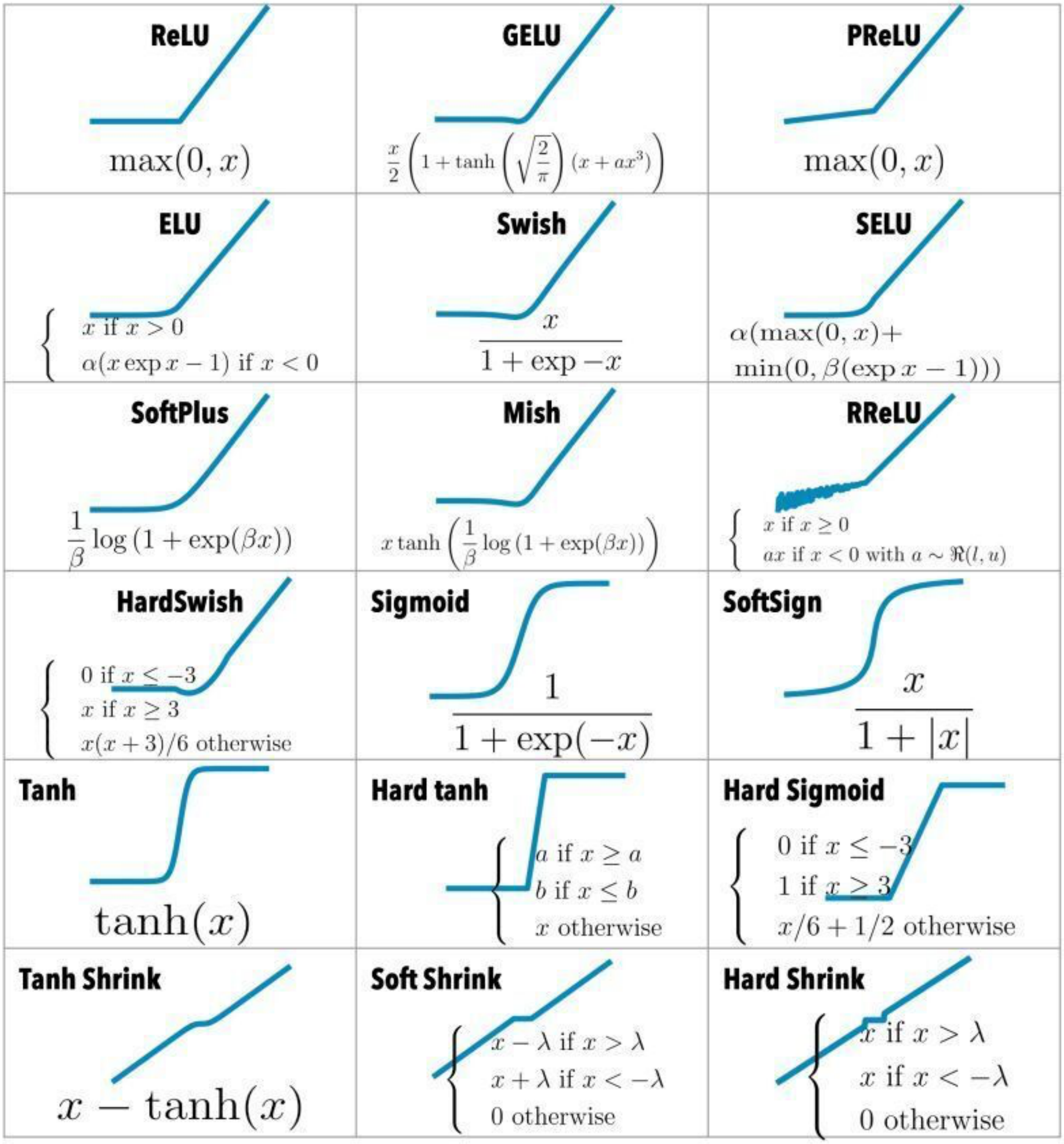
Neural Network Activation Functions: a small subset!



Neural Network Activation Functions:

1️⃣ ReLU (Rectified Linear Unit):
Formula: $f(x) = max(0, x)$
Pros: Simple, computationally efficient, helps mitigate the vanishing gradient problem.
Cons: Can lead to "dying ReLUs" where neurons become inactive and stop learning.

2️⃣ GELU (Gaussian Error Linear Unit):
Formula: $f(x) = x * P(X \le x)$, where P is the cumulative distribution function of the standard normal distribution.
Pros: Smooth, differentiable, combines properties of ReLU and dropout, improves performance in NLP tasks.
Cons: Computationally more intensive compared to ReLU.

3️⃣ Sigmoid:
Formula: $f(x) = 1 / (1 + exp(-x))$
Pros: Outputs probabilities, useful for binary classification.
Cons: Prone to vanishing gradient problem, slow convergence.

4️⃣ Tanh (Hyperbolic Tangent):
Formula: $f(x) = (exp(x) - exp(-x)) / (exp(x) + exp(-x))$
Pros: Zero-centered, less likely to saturate than sigmoid.
Cons: Still suffers from vanishing gradients, slower training.

5️⃣ Leaky ReLU:
Formula: $f(x) = x$ if $x > 0$ else $alpha * x$
Pros: Addresses the "dying ReLU" problem, allows a small gradient when inactive.
Cons: Introduces a small negative slope, which may not always be optimal.

6️⃣ ELU (Exponential Linear Unit):
Formula: $f(x) = x$ if $x > 0$ else $alpha * (exp(x) - 1)$
Pros: Smooth, reduces the bias shift by pushing mean activations closer to zero.
Cons: More computationally intensive than ReLU, introduces an additional hyperparameter.

7️⃣ Swish:
Formula: $f(x) = x / (1 + exp(-x))$
Pros: Smooth, differentiable, improves performance in deep networks.
Cons: More computationally intensive than ReLU.

8️⃣ Softplus:
Formula: $f(x) = log(1 + exp(x))$
Pros: Smooth, always positive, avoids the problem of zero gradients.
Cons: Computationally intensive, can lead to vanishing gradients for large negative inputs.
Choosing the right activation function can significantly impact your model's performance.