INFO 7375 - Neural Networks & AI

Homework to Chapter - 7

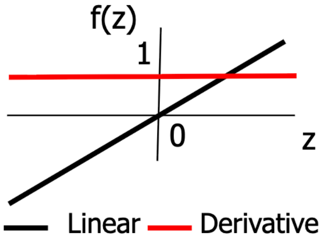
Submitted By:

Abdul Haseeb Khan
NUID: 002844724
khan.abdulh@northeastern.edu

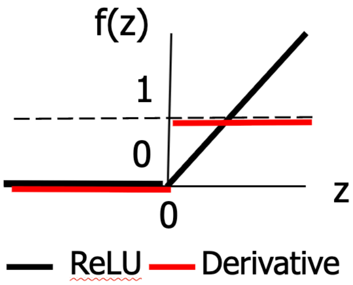
Describe linear, ReLU, sigmoid, tanh, and softmax activation functions and explain for what purposes and where they are typically used.

The following are the activation functions with their purposes:

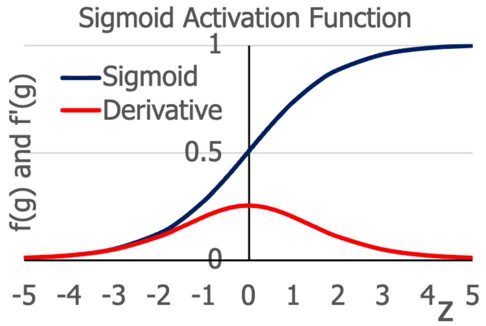
1. **Linear.** The linear (identity) activation implements $a = f(z) = z$, i.e., an affine transformation with no nonlinearity at the output unit. It is standard at the output layer for real-valued regression targets, often interpreted as the mean of a conditional Gaussian so that maximum likelihood corresponds to minimizing mean squared error. Hidden layers should not all be linear because stacking linear maps collapses to a single linear map and removes the network's nonlinearity and representational power.



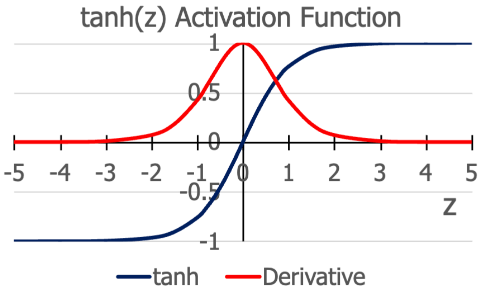
2. **ReLU.** The Rectified Linear Unit computes $f(x) = max(0,x)$ and is piecewise linear with a non-saturating positive region that preserves gradient flow when $x > 0$. ReLU is computationally efficient and in practice accelerates convergence compared with sigmoid/tanh, making it the default recommendation for hidden layers in modern feedforward networks. A known drawback is "dying ReLU," where units stuck with negative inputs yield zero gradients; this motivates careful optimization and variants such as leaky ReLU to maintain nonzero negative-side slopes.



3. **Sigmoid.** The logistic sigmoid $\sigma(x) = \dfrac{1}{1 + e^{-x}}$ squashes real inputs to the interval (0,1), historically motivated by a firing-rate interpretation. It is appropriate for binary outputs, where the network models $P(y = 1 \mid x)$ with a Bernoulli likelihood and is trained via cross-entropy under maximum likelihood. As a hidden activation it saturates at both tails and is not zero-centered, which hampers gradient flow and learning dynamics, so it is rarely used in hidden layers in modern practice.



4. **tanh.** The hyperbolic tangent tanh $tanh(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = 2\sigma(2x) - 1$ maps to [−1,1] and is zero-centered, which often makes it preferable to sigmoid as a hidden nonlinearity when one of these two must be used. Like sigmoid, tanh saturates at large magnitudes, so gradients can still vanish when activations enter the flat regions. Tanh remains common in recurrent architectures for the state nonlinearity, for example in Elman RNNs that use tanh or ReLU as the cell activation.



5. **Softmax.** Softmax converts a vector of logits y to probabilities via $softmax(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$, producing nonnegative outputs that sum to one on the chosen dimension. It is the standard final-layer activation for multiclass classification, modeling a categorical (multinoulli) distribution and trained with cross-entropy under maximum likelihood. Implementations compute along a specified dimension and typically pair with log-softmax or fused losses for numerical stability in training.