Name: Umair Ahmed Younas
Reg : SP22-BCS-047
AI Assignment

Table 1: Activation Functions in Deep Learning

Reference	**Function**	**Explanation**	**Benefits**	**Limitations**	**Data Type**
Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Berlin, Germany: Springer, 1998, pp. 9–50.	Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$, maps to (0,1).	Smooth, probabilistic output, differentiable.	Vanishing gradients, not zero-centered, costly.	Binary classification
Y. LeCun et al., 1998 (same as above)	Tanh	$f(x) = \tanh(x)$, maps to (-1,1).	Zero-centered, stronger gradients than sigmoid.	Vanishing gradients, computationally intensive.	Recurrent networks
V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 807–814.	ReLU	$f(x) = \max(0, x)$, zero for negatives.	Simple, avoids vanishing gradients, fast.	Dying ReLU, not zero-centered.	Image data
A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, 2013, vol. 30, no. 1, p. 3.	Leaky ReLU	$f(x) = \max(\alpha x, x)$, $\alpha \approx 0.01$.	Prevents dying ReLU, allows negative gradients.	Needs tuning α, not zero-centered.	Sparse data

Continued on next page

Table 1 – *Continued from previous page*

Reference	Function	Explanation	Benefits	Limitations	Data Type
D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, 2016.	ELU	$f(x) = x$ if $x > 0$, else $\alpha(e^x - 1)$.	Zero-centered, smooth negative region.	Slower, needs tuning α.	Continuous data
G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 971–980.	SELU	Scaled ELU, $f(x) = \lambda x$ if $x > 0$, else $\lambda\alpha(e^x - 1)$.	Self-normalizing, robust to noise.	Needs specific initialization, sensitive to hyperparameters.	Dense networks
D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.	GELU	$f(x) = x \cdot \Phi(x)$, Φ: Gaussian CDF.	Smooth, combines ReLU and dropout.	Computationally complex, less interpretable.	NLP, transformers
P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.	Swish	$f(x) = x \cdot \mathrm{sigmoid}(x)$.	Smooth, outperforms ReLU in some tasks.	Computationally costly, less intuitive.	Deep networks
D. Misra, "Mish: A self regularized non-monotonic neural activation function," *arXiv preprint arXiv:1908.08681*, 2019.	Mish	$f(x) = x \cdot \tanh(\ln(1 + e^x))$.	Smooth, non-monotonic, preserves negatives.	Costly, less studied.	Generative models

Continued on next page

Table 1 – *Continued from previous page*

Reference	Function	Explanation	Benefits	Limitations	Data Type
H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, 2015, pp. 1–8.	Softplus	$f(x) = \ln(1 + e^x)$, smooth ReLU.	Smooth, fully differentiable.	Vanishing gradients, costly.	Regression tasks
X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256.	Softsign	$f(x) = \frac{x}{1+\|x\|}$, maps to (-1,1).	Smooth, bounded, simpler than tanh.	May have vanishing gradients.	General tasks
K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 1026–1034.	PReLU	$f(x) = \max(\alpha x, x)$, α learned.	Adapts to data, improves Leaky ReLU.	More parameters, risks overfitting.	Image data
Y. LeCun et al., 1998	Linear	$f(x) = x$, direct pass-through.	Simple, no computation cost.	No non-linearity, limited use.	Regression output
F. Rosenblatt, "The perceptron: A probabilistic model," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.	Binary Step	$f(x) = 1$ if $x > 0$, else 0.	Fast, simple for binary tasks.	Not differentiable, unsuitable for deep learning.	Basic classification

Continued on next page

Table 1 – *Continued from previous page*

Reference	Function	Explanation	Benefits	Limitations	Data Type
A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.	Softmax	$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$.	Outputs probabilities for multi-class.	Costly for many classes.	Multi-class classification