Fully Connected Neural Network

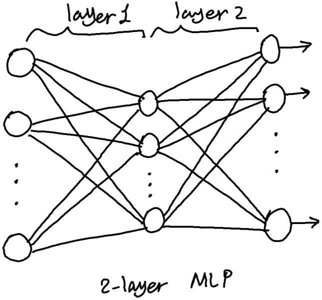
Deep Learning (ENGG*6600*07)

School of Engineering,
University of Guelph, ON, Canada

Course Instructor: Benyamin Ghojogh
Fall 2023

MLP

- A **fully connected neural network** is a stack of layers of neural network where in every layer, all the neurons of the previous layer are connected to all the neurons of the next layer.
- Every layer of the fully connected neural network is called a **fully connected layer** or a **dense layer**.
- Each neuron in the fully connected neural network is a Perceptron neuron. That is why this network is also called the **Multi-Layer Perceptron (MLP)**.
- MLP was proposed by **Rosenblatt** in 1958, in the same paper as Perceptron [1]. In that paper, he proposed an MLP with three layers.



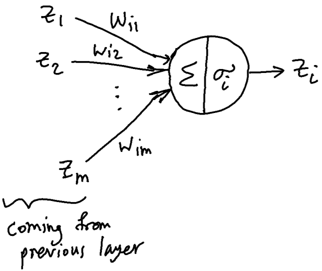
Neuron

- Each neuron in the fully connected neural network is a Perceptron neuron. So it has:
  - Summation of the outputs of previous layer multiplied by the weights of previous layer:

$$a_i = \sum_{\ell=1}^{m} w_{i\ell} z_\ell. \tag{1}$$

  - Activation function:

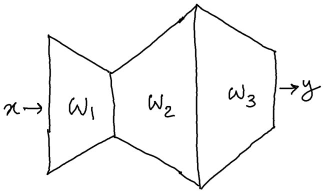
$$z_i := \sigma_i(a_i). \tag{2}$$



Layer as a Projection

- Every layer in the fully connected network can be seen as a linear projection followed by an activation function.

$$\boldsymbol{y} = \sigma_3\Big(\boldsymbol{W}_3^\top \sigma_2\big(\boldsymbol{W}_2^\top \sigma_1(\boldsymbol{W}_1^\top \boldsymbol{x})\big)\Big). \tag{3}$$



- The activation function is usually a nonlinear function because if all activation functions are linear in the network, the entire network is collapsed to be one linear projection.

$$\boldsymbol{y} = \boldsymbol{W}_3^\top \boldsymbol{W}_2^\top \boldsymbol{W}_1^\top \boldsymbol{x} = \boldsymbol{V}^\top \boldsymbol{x}, \tag{4}$$

where:

$$\boldsymbol{V} := \boldsymbol{W}_1 \boldsymbol{W}_2 \boldsymbol{W}_3. \tag{5}$$

Activation Function

There exist various activation functions. Some of them are:

- Linear (identity) function:

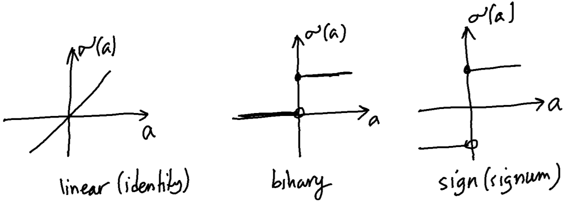
$$\sigma(a) = a, \quad \sigma(a) \in (-\infty, \infty), \quad \sigma'(a) = 1. \tag{6}$$

- Binary step:

$$\sigma(a) = \begin{cases} 0 & \text{if } a < 0 \\ 1 & \text{if } a \geq 0 \end{cases}, \quad \sigma(a) \in [0,1], \quad \sigma'(a) = \begin{cases} 0 & \text{if } a \neq 0 \\ \text{undefined} & \text{if } a = 0. \end{cases} \tag{7}$$

- Sign (signum) function:

$$\sigma(a) = \begin{cases} -1 & \text{if } a < 0 \\ 1 & \text{if } a \geq 0 \end{cases}, \quad \sigma(a) \in [-1,1], \quad \sigma'(a) = \begin{cases} 0 & \text{if } a \neq 0 \\ \text{undefined} & \text{if } a = 0. \end{cases} \tag{8}$$



Activation Function

- Logistic (sigmoid) function:

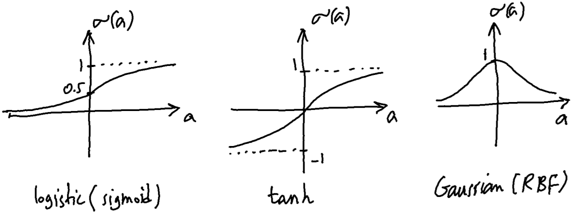
$$\sigma(a) = \frac{1}{1+e^{-a}}, \quad \sigma(a) \in [0,1], \quad \sigma'(a) = \sigma(a)(1-\sigma(a)). \tag{9}$$

- Hyperbolic tangent (tanh):

$$\sigma(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}, \quad \sigma(a) \in [-1,1], \quad \sigma'(a) = 1-\sigma(a)^2. \tag{10}$$

- Gaussian (radial basis function):

$$\sigma(a) = e^{-a^2}, \quad \sigma(a) \in (0,1], \quad \sigma'(a) = -2ae^{-a^2}. \tag{11}$$



Activation Function

- Softplus [2] (2011):

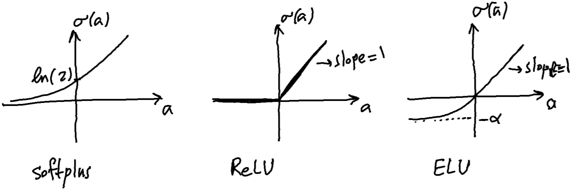
$$\sigma(a) = \ln(1+e^a), \quad \sigma(a) \in [0,\infty), \quad \sigma'(a) = \frac{1}{1+e^{-a}}. \tag{12}$$

- Rectified linear unit (ReLU) [3] (2010):

$$\sigma(a) = \begin{cases} 0 & \text{if } a < 0 \\ a & \text{if } a \geq 0 \end{cases} = \max(0,a), \quad \sigma(a) \in [0,\infty), \quad \sigma'(a) = \begin{cases} 0 & \text{if } a < 0 \\ \text{undefined} & \text{if } a = 0 \\ 1 & \text{if } a > 0. \end{cases} \tag{13}$$

- Exponential Linear Unit (ELU) [4] (2015):

$$\sigma(a) = \begin{cases} \alpha(e^a - 1) & \text{if } a < 0 \\ a & \text{if } a \geq 0 \end{cases}, \quad \sigma(a) \in (-\alpha,\infty), \quad \sigma'(a) = \begin{cases} \alpha e^a & \text{if } a < 0 \\ 1 & \text{if } a = 0, \alpha = 1 \\ 1 & \text{if } a > 0. \end{cases} \tag{14}$$



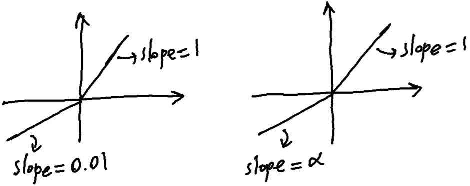
Activation Function

- Leaky rectified linear unit (Leaky ReLU) [5] (2013):

$$\sigma(a) = \begin{cases} 0.01a & \text{if } a < 0 \\ a & \text{if } a \geq 0 \end{cases}, \quad \sigma(a) \in (-\infty, \infty), \quad \sigma'(a) = \begin{cases} 0.01 & \text{if } a < 0 \\ \text{undefined} & \text{if } a = 0 \\ 1 & \text{if } a > 0. \end{cases} \tag{15}$$

- Parametric rectified linear unit (PReLU) [6] (2015):

$$\sigma(a) = \begin{cases} \alpha a & \text{if } a < 0 \\ a & \text{if } a \geq 0 \end{cases}, \quad \sigma(a) \in (-\infty, \infty), \quad \sigma'(a) = \begin{cases} \alpha & \text{if } a < 0 \\ \text{undefined} & \text{if } a = 0 \\ 1 & \text{if } a > 0. \end{cases} \tag{16}$$



Activation Function

- Softmax:

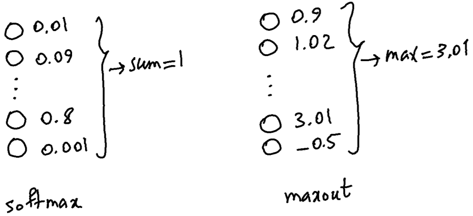
$$\sigma_i(\boldsymbol{a}) = \frac{e^{a_i}}{\sum_{i=1}^{m} e^{a_i}}, \forall i \in \{1, \ldots, m\}, \quad \sigma_i(\boldsymbol{a}) \in (0, 1), \quad \sigma'(a) = \sigma_i(\boldsymbol{a})\big(\delta_{ij} - \sigma_j(\boldsymbol{a})\big), \tag{17}$$

where δ_{ij} is the Kronecker delta:

$$\delta_{ij} := \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases} \tag{18}$$

- Maxout [7] (2013):

$$\sigma_i(\boldsymbol{a}) = \max_j(\sigma_j), \quad \sigma_i(\boldsymbol{a}) \in (-\infty, \infty), \quad \sigma'(a) = \begin{cases} 1 & \text{if } i = \arg\max_j(\sigma_j) \\ 0 & \text{if } i \neq \arg\max_j(\sigma_j). \end{cases} \tag{19}$$



References

[1] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[2] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, JMLR Workshop and Conference Proceedings, 2011.

[3] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[4] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[5] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, Atlanta, Georgia, USA, 2013.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[7] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *International conference on machine learning*, pp. 1319–1327, PMLR, 2013.