Inhaltsverzeichnis

Abbildungsverzeichnis

List of Equations

Lecture 4: Dense Neural Networks

Date: 15/05/2020

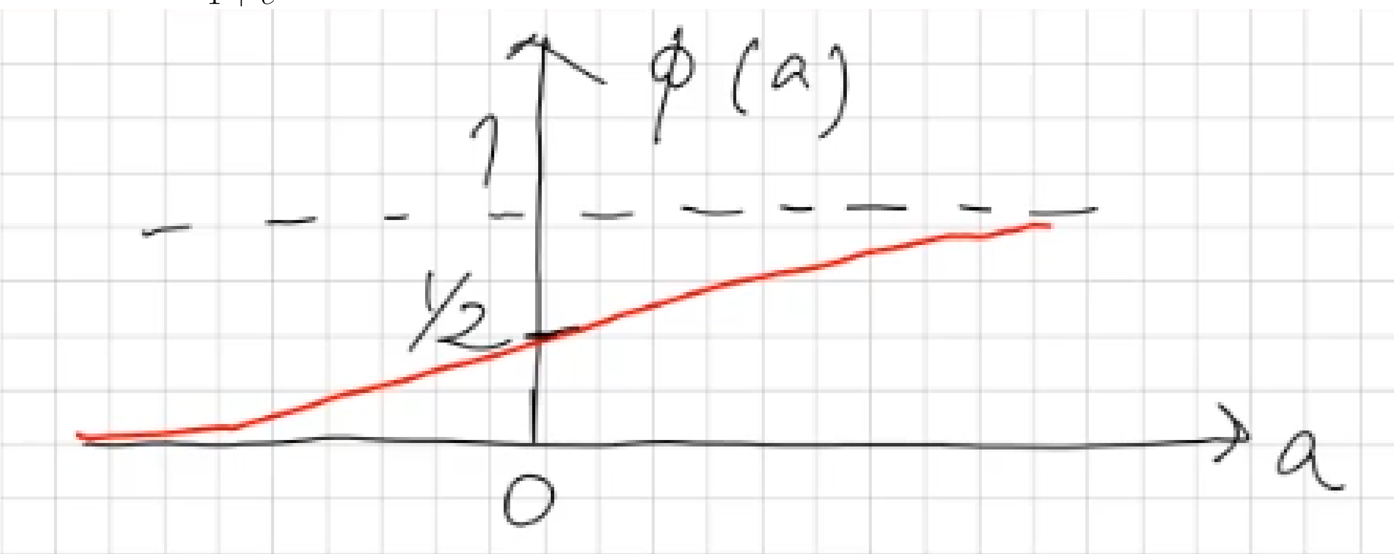
Lecturer: David Silver *By: Nithish Moudhgalya*

1.1. Activation function

Mild requirements on $\phi()$:
•nonlinear in general $\rightarrow$ fundamental
•smooth, differentiable $\rightarrow$ for training
•simple calculation $\rightarrow$ low complexity
•Slides 4-6; 4-7 activation function types

1.1.1. Sigmoid activation function

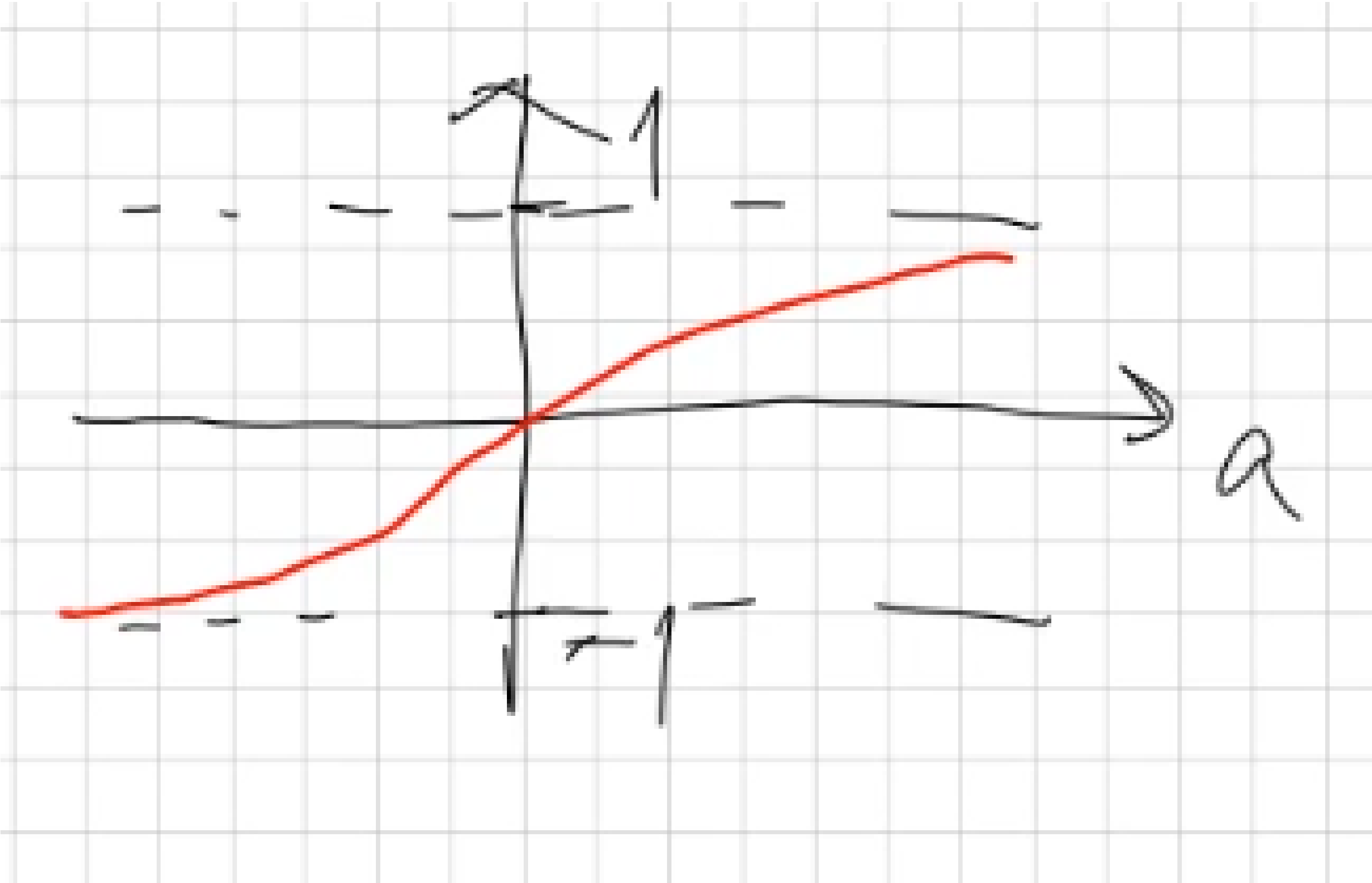
$\phi(a) = \sigma(a) = \dfrac{1}{1+e^{-a}}$



•$0 < \phi < 1 \,\hat{=}\,$ prob.
•symmetry: $\phi(-a) = 1 - \phi(a)$
•derivative: $\dfrac{d\phi(a)}{da} = ... = \dfrac{e^{-a}}{(1+e^{-a})^2} = \phi(a) \cdot \phi(-a) = \phi(a)(1-\phi(a)), \in (0,1)$
easy calculative
•widely used in conventional NN (shallow)

1.1.2. hyperbolic tangent activation function

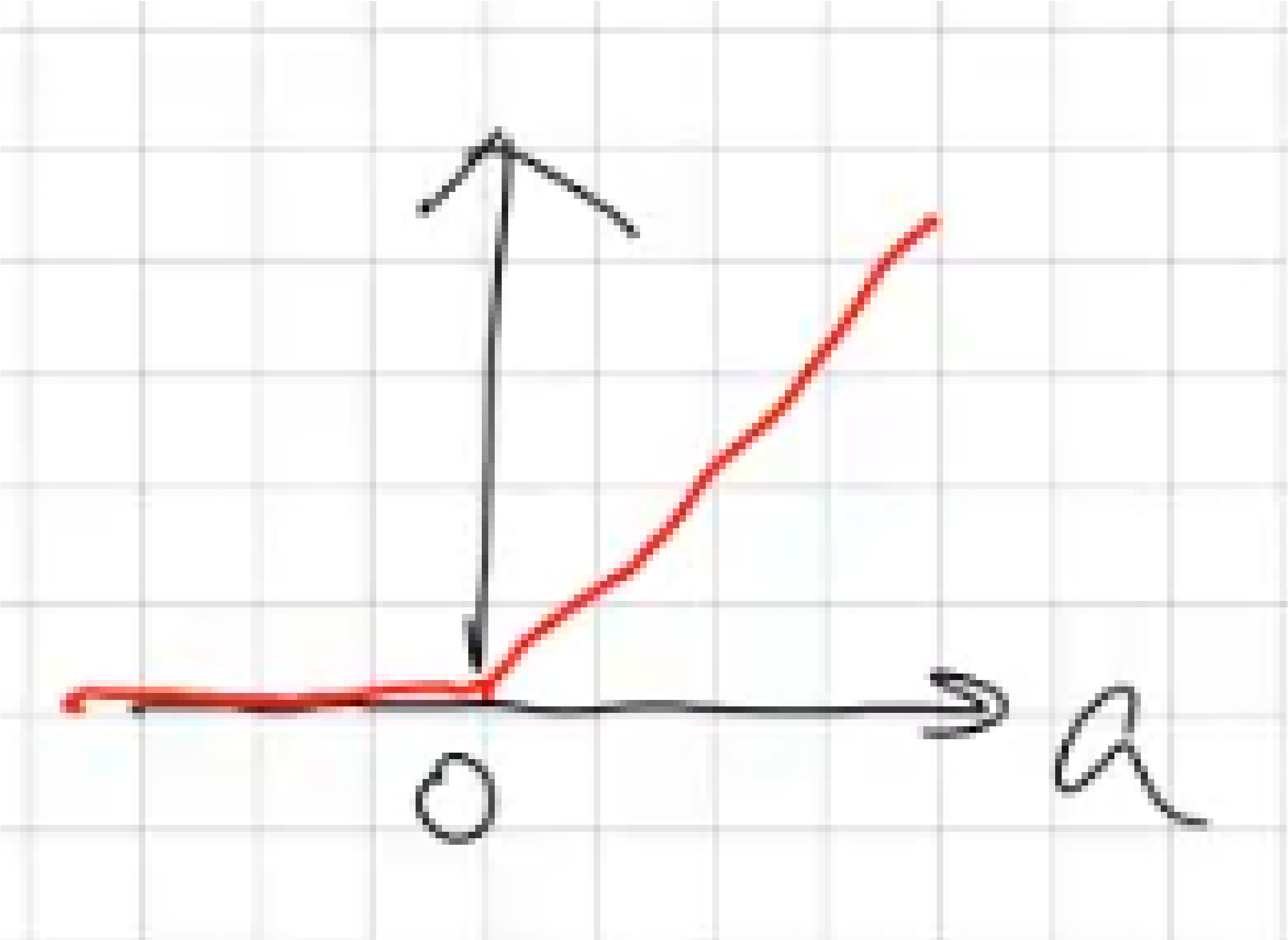
$\phi(a) = tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$



like sigmoid but another output range

1.1.3. rectifier linear unit(ReLU

$\phi(a) = ReLU(a) = max(a, 0) = \begin{cases} a & a \geq 0 \\ 0 & a < 0 \end{cases}$



•$\equiv$ diode
simple calculation
•$\dfrac{d\phi}{da} = \begin{cases} 1 & a > 0 \\ 0 & a < 0 \end{cases} = u(a), u(0) = 0$ typically used
•most popular in DNN
•Details on 4-7

1.1.4. Softmax activatoin function(classification problem)

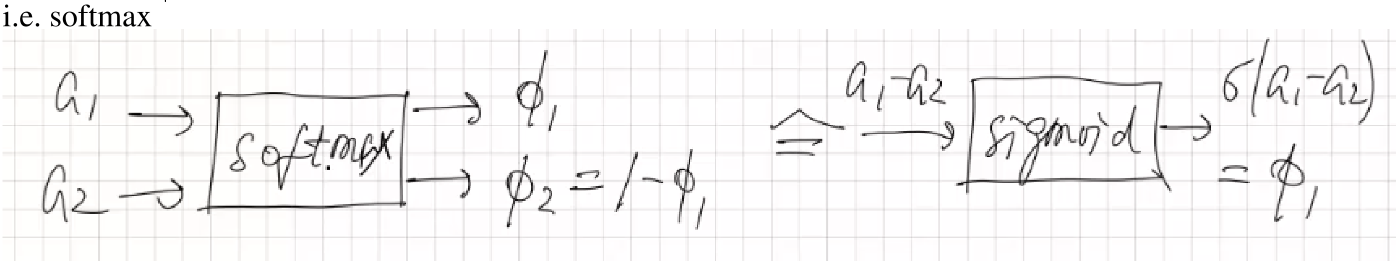
$\phi(\underline{a} : \underline{a} = [a_i] \in \Re^c \to \Re^c$

$\phi(\underline{a}) = softmax(\underline{a}) = \begin{bmatrix} \phi_1(\underline{a}) \\ \vdots \\ \phi_c(\underline{a}) \end{bmatrix}$

$\phi(\underline{a}) = \dfrac{a^{a_i}}{\sum_{j=1}^{c} e^{a_i}}, \in (0, 1), \sum_{i=1}^{c} \phi_i(\underline{a}) = 1$
•maps $\underline{a} \in \Re^c$ to a categorical PMF with c classes
•a_i large$\to$ $\phi_i(\underline{a})$ close to 1
•a_i small $\to$ $\phi_i(\underline{a})$ close to 0
•used in the output layer for classification problems

1.2. Special case c=2, binary classification problem

: $\phi_1(\underline{a}) = \dfrac{e^{a_i}}{e^{a_1} + e^{a_2}} = \dfrac{1}{1 + e^{-(a_1 - a_2)}} = \sigma(a_1 - a_2)$
$\phi_2(\underline{a} = \dfrac{e^{a_2}}{e^{a_1} + e^{a_2}} = 1 - \phi_1(\underline{a}) = \sigma(a_2 - a_1)$
i.e. softmax



one sigmoid output is sufficient for binary classification instead of 2 output softmax!
Derivative of softmax:
$\dfrac{\partial \phi_i(\underline{a})}{\partial a_j} = ... = \begin{cases} \phi : i(\underline{a})(1 - \phi_i(\underline{a}) & i = j \\ -phi_i(\underline{a})\phi_j(\underline{a}) & i \neq j \end{cases}$