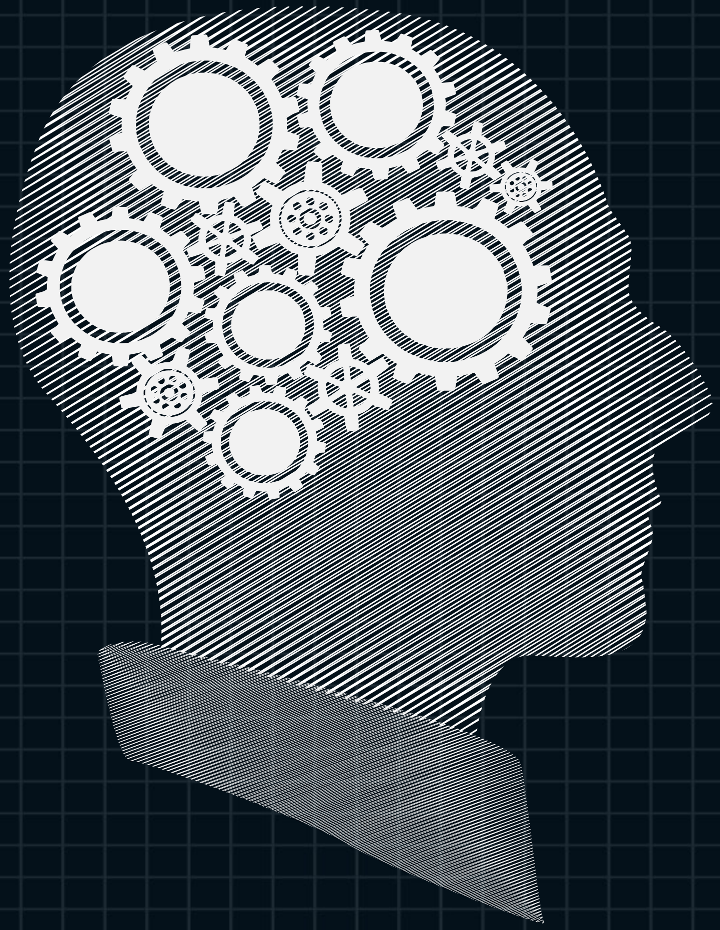
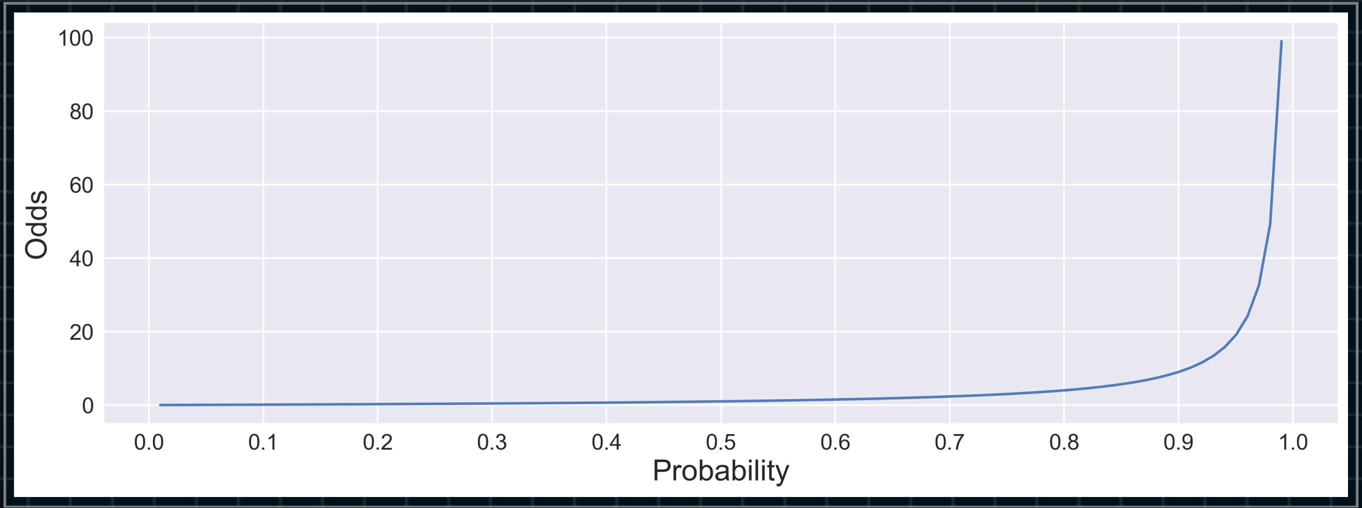


Forward Propagation of Neural Networks

Lecture.3
Sigmoid and Softmax

Lecture.3 Sigmoid and Softmax

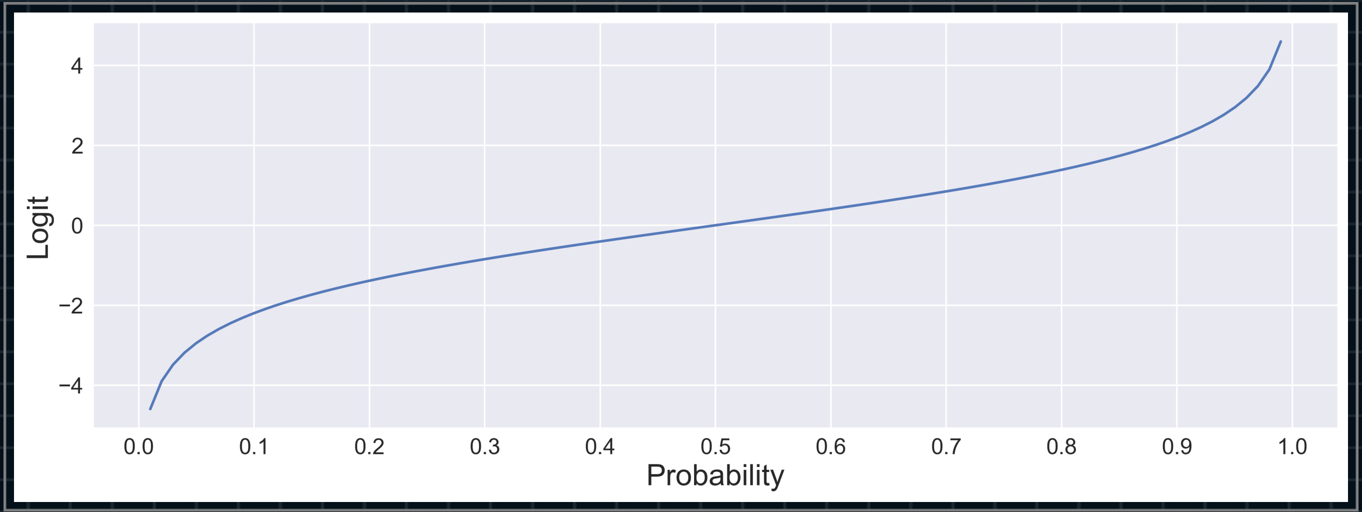
- Odds



$$o = \frac{p}{1-p}$$

Lecture.3
Sigmoid and Softmax

- Logit



$$l = log(\frac{p}{1-p})$$

Lecture.3
Sigmoid and Softmax

- Logit and Sigmoid

$$l = log(\frac{p}{1-p})$$

$$l = log(\frac{p}{1-p})$$

$$e^l = \frac{p}{1-p}$$

$$\frac{1}{e^l} = \frac{1-p}{p} = \frac{1}{p} - 1$$

$$\frac{1}{e^l} + 1 = \frac{1}{p}$$

$$\frac{1+e^l}{e^l} = \frac{1}{p}$$

$$p = \frac{e^l}{1+e^l}$$

$$p = \frac{1}{1+e^{-l}}$$

$$p = \sigma(l) = \frac{1}{1+e^{-l}}$$

- Logit and Sigmoid

$$l = log(\frac{p}{1-p})$$

$$p = \frac{1}{1+e^{-l}}$$

$$\begin{array}{c} l \\ \uparrow \\ \boxed{l = log(\frac{p}{1-p})} \\ \uparrow \\ p \end{array} \qquad\qquad \begin{array}{c} p \\ \uparrow \\ \boxed{p = \frac{1}{1+e^{-l}}} \\ \uparrow \\ l \end{array}$$

Lecture.3 Sigmoid and Softmax - from Logit to Probability

$$0 \leq p \leq 1$$

$$p = \frac{1}{1 + e^{-z}}$$

$$-\infty < z < \infty$$

$$z = (\overrightarrow{x})^T \overrightarrow{x} + b$$

$$(\overrightarrow{x})^T$$

Lecture.3 Sigmoid and Softmax

- from Logit to Probability

$\vec{p} \in \mathbb{R}^{N\times 1}$

$$\vec{p} = \frac{1}{1 + e^{-\vec{z}^{[1]}}}$$

$\vec{z}^{[1]} \in \mathbb{R}^{N\times 1}$

$$\vec{z}^{[1]} = X^T \vec{w}^{[1]} + b^{[1]}$$

$\vec{w}^{[1]} \in \mathbb{R}^{l_I}$

$b^{[1]} \in \mathbb{R}$

$$X^T = \begin{pmatrix} \longleftarrow & \left(\vec{x}^{(1)}\right)^T & \longrightarrow \\ \longleftarrow & \left(\vec{x}^{(2)}\right)^T & \longrightarrow \\ & \vdots & \\ \longleftarrow & \left(\vec{x}^{(N)}\right)^T & \longrightarrow \end{pmatrix}$$

- Softmax Layer

$$p_i = P(C_i),\ 1 \le i \le K$$

$(\vec{p})^T$ $\quad p_1 \quad p_2 \quad \cdots \quad p_K$

$$S_j\big((\vec{l})^T\big) = p_j = \frac{e^{l_j}}{\sum_{k=1}^{K}\left[e^{l_k}\right]}$$

$(\vec{l})^T$ $\quad l_1 \quad l_2 \quad \cdots \quad l_K$

Lecture.3
Sigmoid and Softmax

- Softmax Layer

p_1 $\quad p_2 \quad \cdots \quad$ p_K

$(\vec{p})^T \in \mathbb{R}^{1\times K}$

$$S_j\left((\vec{z}^{[O]})^T\right) = p_j = \frac{e^{z_j}}{\sum_{k=1}^{K}\left[e^{z_k}\right]}$$

$\cdots$

$(\vec{z}^{[O]})^T \in \mathbb{R}^{1\times l_O},\ \ l_O = K$

$$(\vec{z}^{[O]})^T = (\vec{a}^{[O-1]})^T W^{[O]} + (\vec{b}^{[O]})^T$$

$(\vec{a}^{[O-1]})^T$

Lecture.3
Sigmoid and Softmax

- Softmax Layer

$$P^T = \begin{pmatrix} \longleftarrow & (\overrightarrow{p}^{(1)})^T & \longrightarrow \\ \longleftarrow & (\overrightarrow{p}^{(2)})^T & \longrightarrow \\ & \vdots & \\ \longleftarrow & (\overrightarrow{p}^{(N)})^T & \longrightarrow \end{pmatrix} \in \mathbb{R}^{N \times K}$$

$$S_j\big((\vec{z}^{[O](i)})^T\big) = p_j^{(i)} = \frac{e^{z_j^{(i)}}}{\sum_{k=1}^{K}\left[e^{z_k^{(i)}}\right]}$$

$$\cdots$$

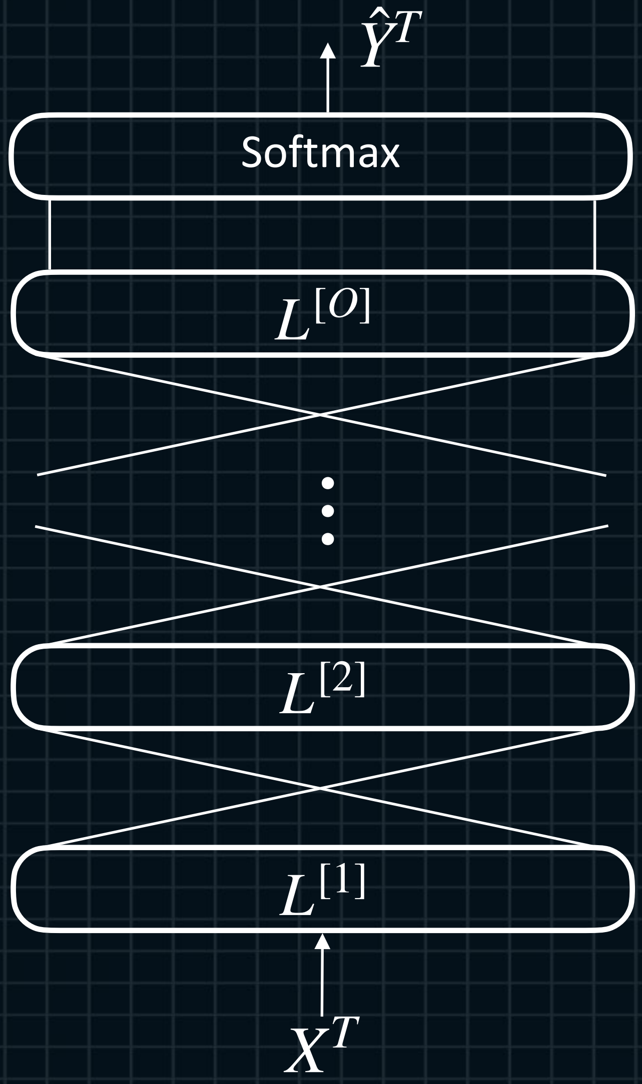
$$(Z^{[O]})^T = \begin{pmatrix} \longleftarrow & (\vec{z}^{[O](1)})^T & \longrightarrow \\ \longleftarrow & (\vec{z}^{[O](2)})^T & \longrightarrow \\ & \vdots & \\ \longleftarrow & (\vec{z}^{[O](N)})^T & \longrightarrow \end{pmatrix} \in \mathbb{R}^{N \times l_O}, \ \ l_O = K$$

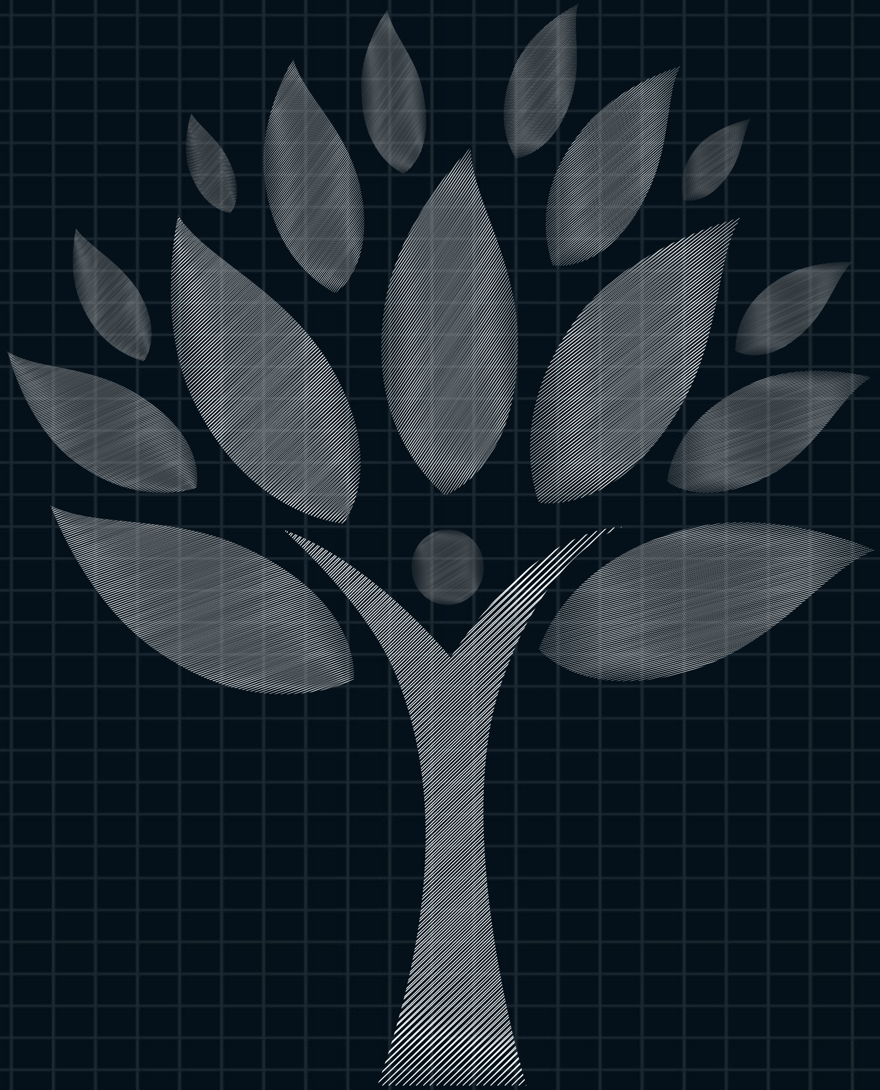
$$(Z^{[O]})^T = (A^{[O-1]})^T W^{[O]} + (\overrightarrow{b}^{[O]})^T$$

$$(A^{[O-1]})^T = \begin{pmatrix} \longleftarrow & (\overrightarrow{a}^{[O-1](1)})^T & \longrightarrow \\ \longleftarrow & (\overrightarrow{a}^{[O-1](2)})^T & \longrightarrow \\ & \vdots & \\ \longleftarrow & (\overrightarrow{a}^{[O-1](N)})^T & \longrightarrow \end{pmatrix} \in \mathbb{R}^{N \times l_{O-1}}$$

Lecture.3 Sigmoid and Softmax

- from Feature to Prediction





Forward Propagation of Neural Networks

Lecture.3
Sigmoid and Softmax