Data Training in Deep Learning

Siu Sai Cheong

About Me

SIU Sai Cheong (scsiu@hsu.edu.hk)

- Programme Director, Bachelor of Translation with Business Programme
- Programme Director, Master of Arts in Translation (Computer-aided Translation) Programme
- Former Director, Deep Learning Research and Application Centre

My Research Areas

Translation Technology

Artificial Intelligence for Professional Language Services and Creative Industries

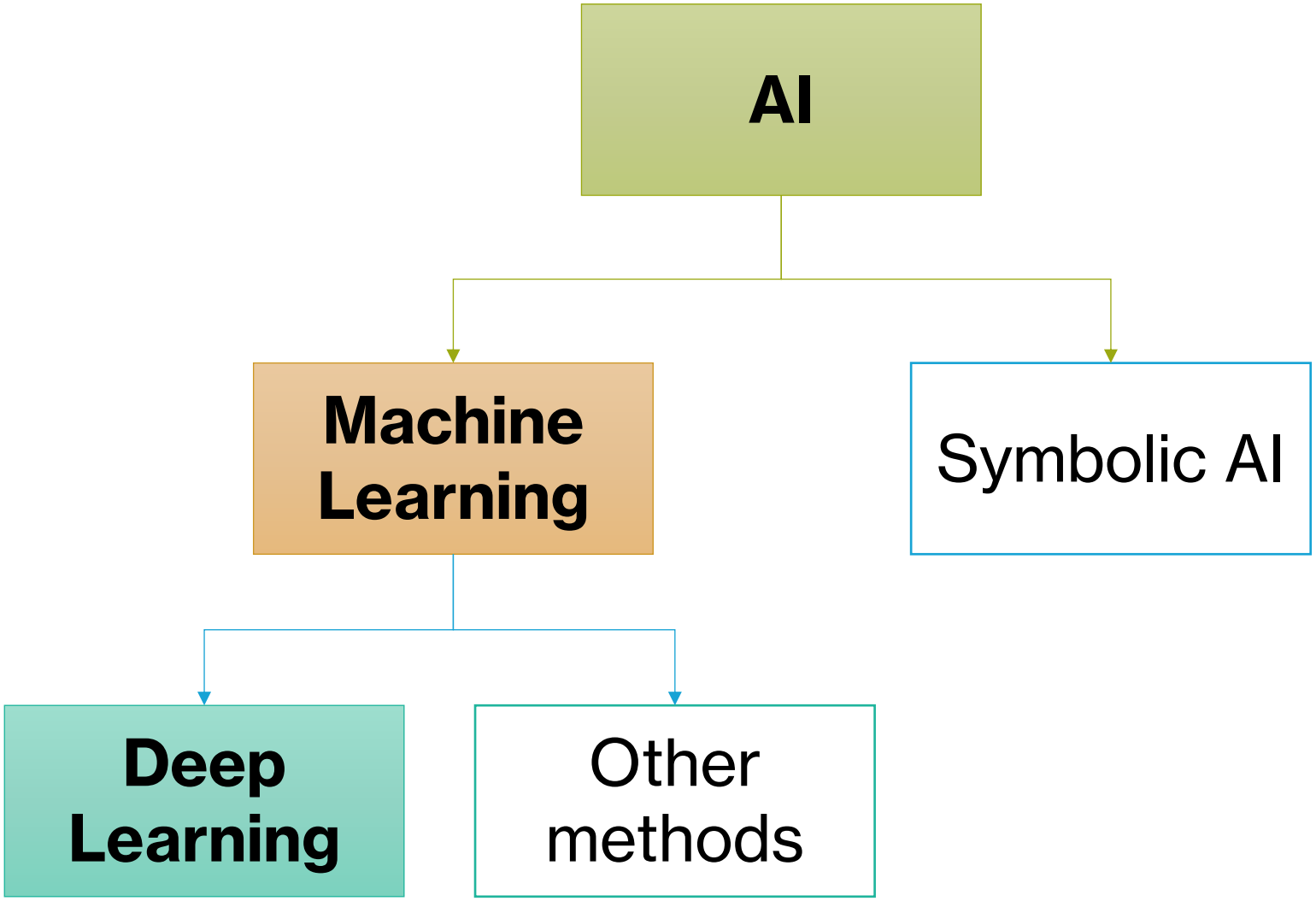
Outline

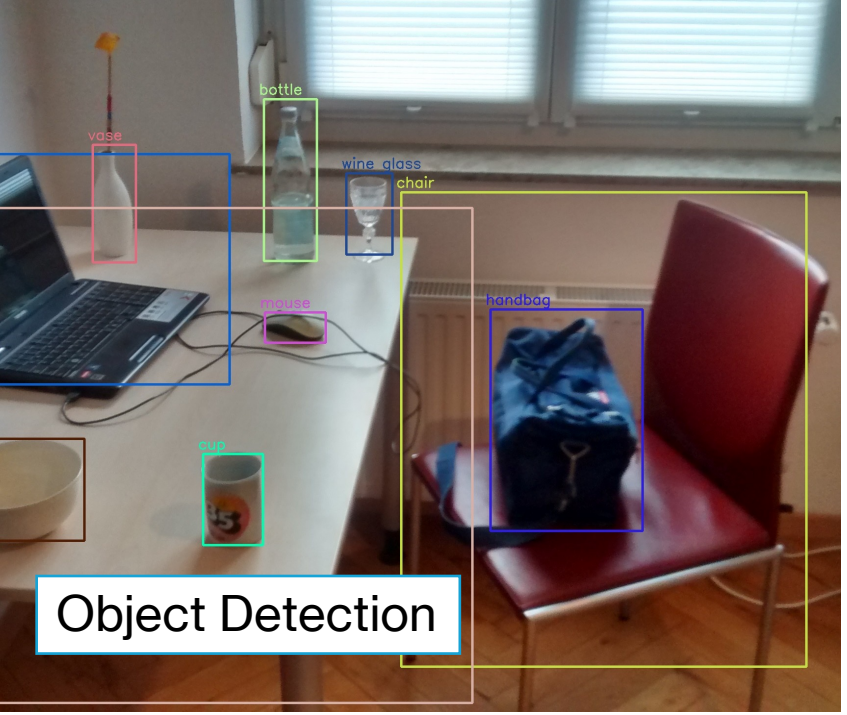
1. Deep Learning: A Brief Introduction

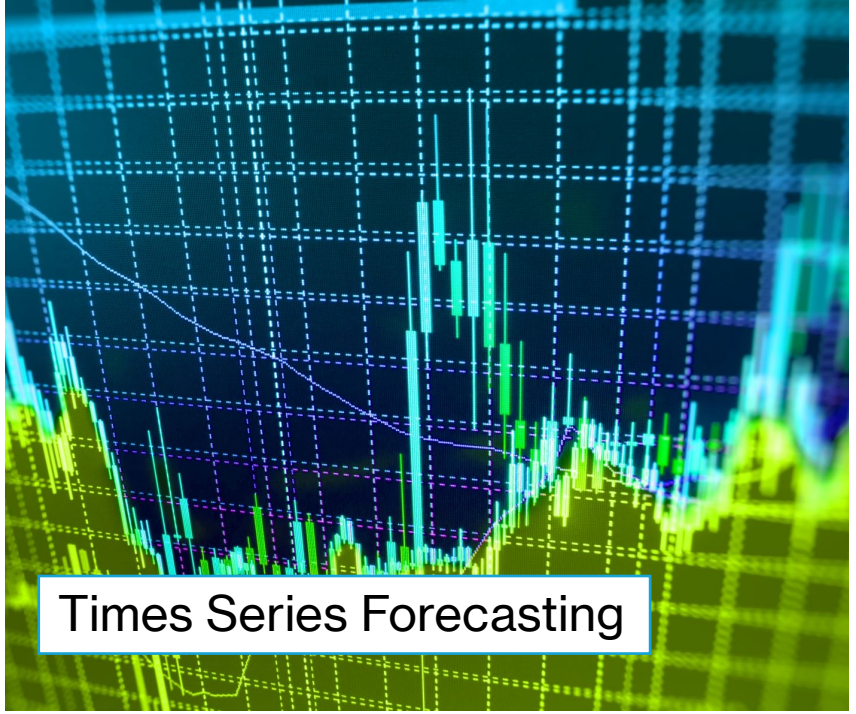
2. Model Training: Key Steps

3. Other Interesting Issues

Deep Learning and Artificial Intelligence


AI
Machine Learning
Symbolic AI
Deep Learning
Other methods


bottle
vase
wine glass
chair
mouse
handbag
cup
Object Detection


Times Series Forecasting


AlphaGo

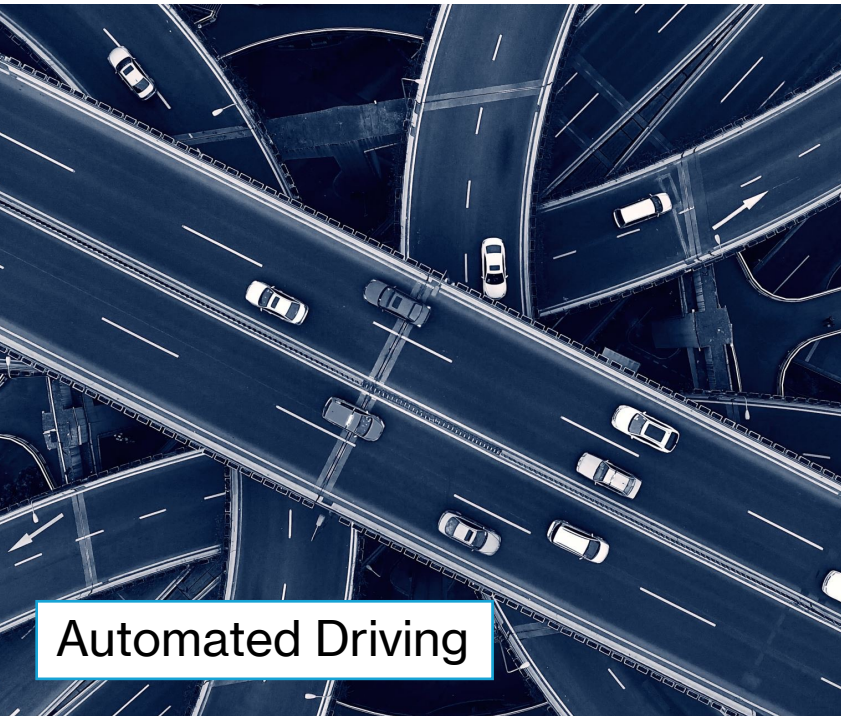
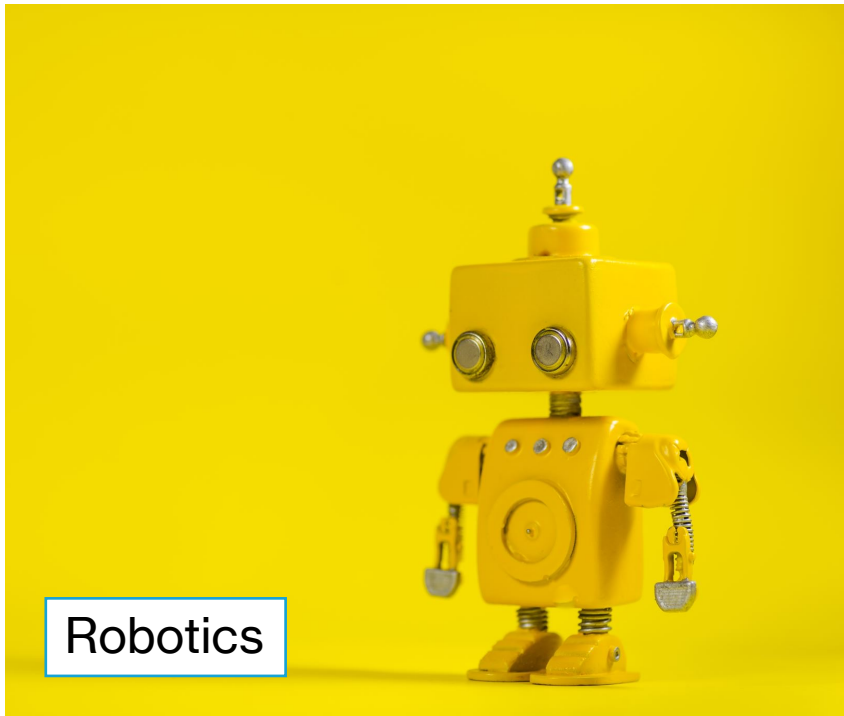

Automated Driving


Image Classification


Robotics

My Deep Learning Projects

- Text and Speech Translation
- Image Translation
- Other Natural Language Processing (NLP) Tasks

IPO Translate (English-Chinese Translation)

Source Text

Hong Kong Exchanges and Clearing Limited, The Stock Exchange of Hong Kong Limited and Hong Kong Securities Clearing Company Limited take no responsibility for the contents of this Prospectus, make no representation as to its accuracy or completeness, and expressly disclaim any liability whatsoever for any loss howsoever arising from or in reliance upon the whole or any part of the contents of this Prospectus.

A copy of this Prospectus, having attached thereto the documents specified in "Appendix V — Documents Delivered to the Registrar of Companies and Available for Inspection" to this Prospectus, has been registered by the Registrar of Companies in Hong Kong as required by section 342C of the Companies (Winding Up and Miscellaneous Provisions) Ordinance (Chapter 32 of the Laws of Hong Kong). The Securities and Futures Commission and the Registrar of Companies in Hong Kong take no responsibility for the contents of this Prospectus or any other document referred to above.

Load Sample | Translate

Hong Kong Exchanges and Clearing Limited, The Stock Exchange of Hong Kong Limited and Hong Kong Securities Clearing Company Limited take no responsibility for the contents of this Prospectus, make no representation as to its accuracy or completeness, and expressly disclaim any liability whatsoever for any loss howsoever arising from or in reliance upon the whole or any part of the contents of this Prospectus.

Translation 1	Translation 2
香港交易及結算有限公司、香港聯合交易所有限公司及香港證券結算有限公司就本招股章程的內容概不承擔任何責任，概不就其準確性或完整性表示任何聲明，並明確概不就本招股章程所載的全部或任何部分產生的任何虧損而產生的任何損失承擔責任。	香港交易及結算所有限公司、香港聯合交易所有限公司及香港中央結算有限公司對本招股章程的內容概不負責，對其準確性或完整性概不發表任何聲明，並明確表示概不就因本招股章程全部或任何部分內容而產生或因依賴該等內容而產生的任何損失承擔任何責任。

A copy of this Prospectus, having attached thereto the documents specified in "Appendix V — Documents Delivered to the Registrar of Companies and Available for Inspection" to this Prospectus, has been registered by the Registrar of Companies in Hong Kong as required by section 342C of the Companies (Winding Up and Miscellaneous Provisions) Ordinance (Chapter 32 of the Laws of Hong Kong).

Translation 1	Translation 2
本招股章程的副本連同本招股章程「附錄五－送呈公司註冊處處長及備查文件」所載文件所附的文件，已根據公司（清盤及雜項條文）條例第342C 條（香港法例第32章）的規定於香港公司登記。	本招股章程副本連同本招股章程「附錄五－送呈公司註冊處處長及備查文件」所述文件，已根據公司（清盤及雜項條文）條例（香港法例第32章）第342C 條的規定由香港公司註冊處處長登記。

The Securities and Futures Commission and the Registrar of Companies in Hong Kong take no responsibility for the contents of this Prospectus or any other document referred to above.

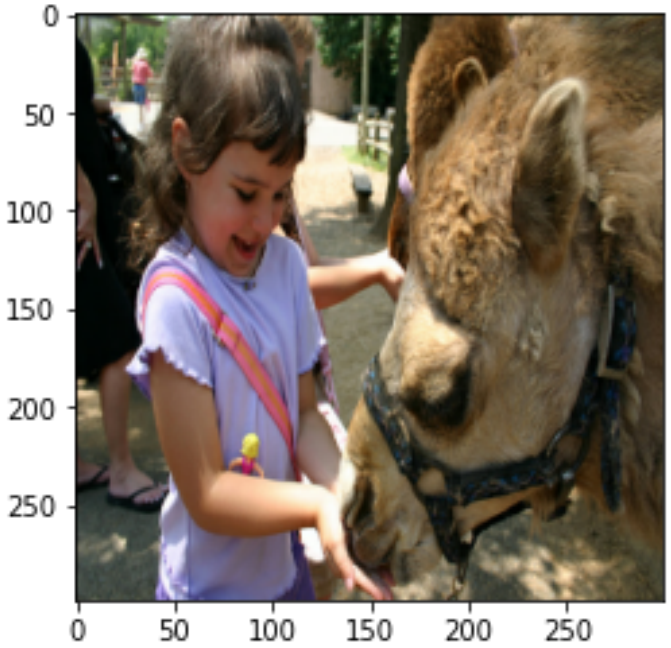
Translation 1	Translation 2
證券及期貨事務監察委員會及香港公司註冊處處長概不就本招股章程或上文所述任何其他文件的內容負責。	證券及期貨事務監察委員會及香港公司註冊處處長對本招股章程或上述任何其他文件的內容概不負責。

Technology for the Translation of Government Press Releases

Website for the Project "A Hybrid Approach to the Translation of Government Press Releases: Integration of Translation Memories and Neural Machine Translation"

The work described in this website was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS14/H16/18).

Image-to-text translation



女孩餵駱駝

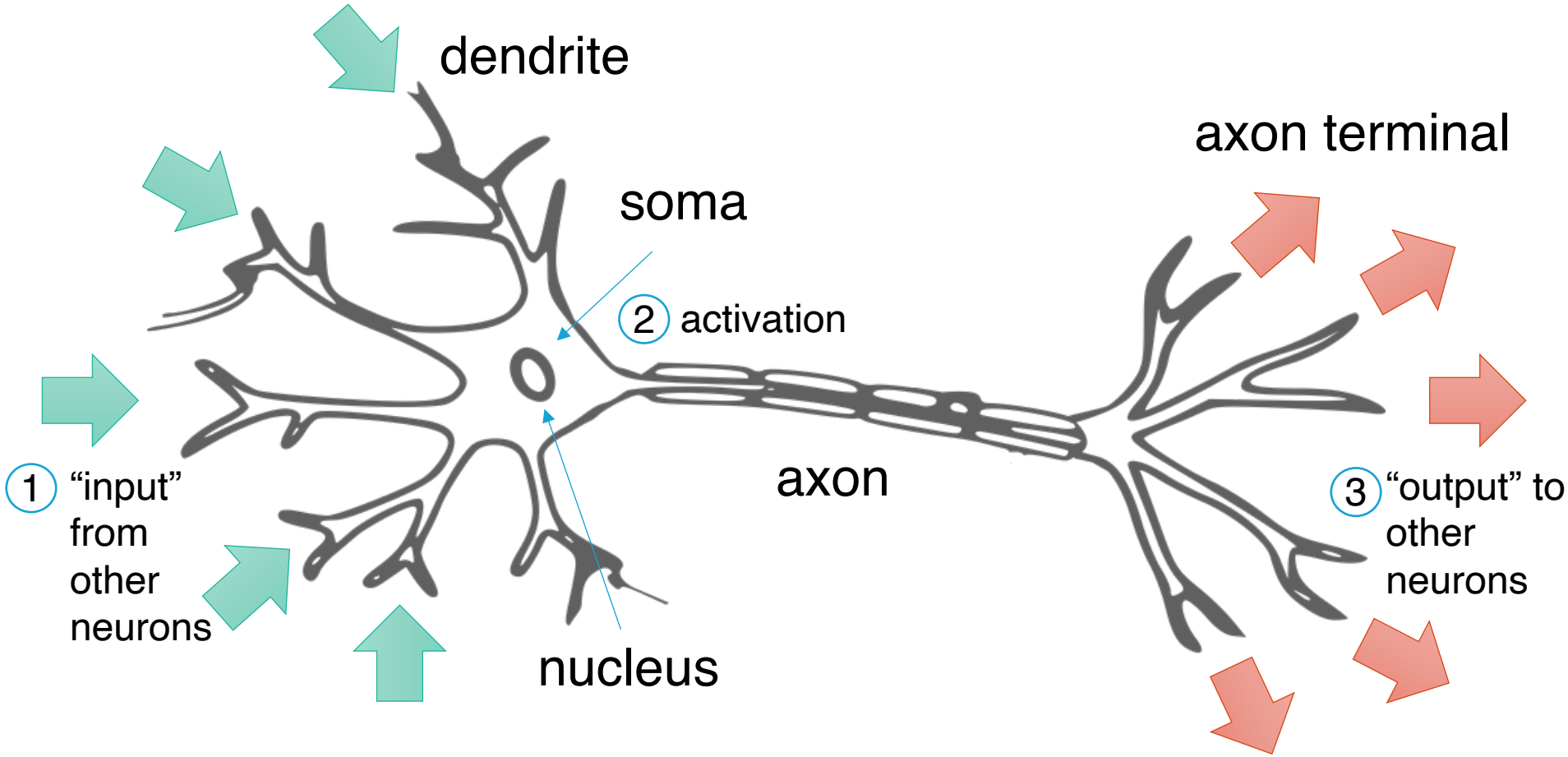
How does Deep Learning work?

Deep Learning

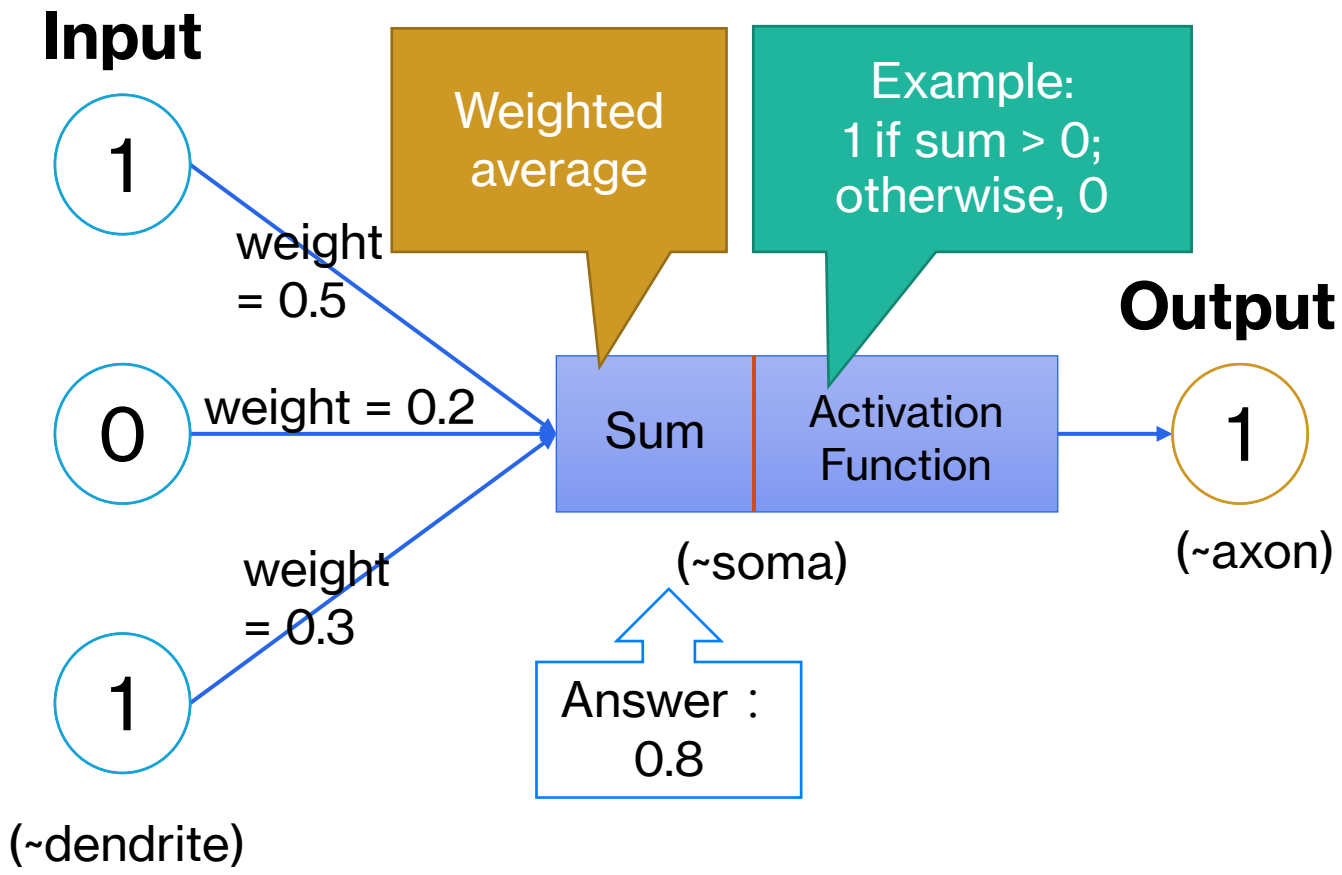
Artificial Neural Network

Artificial Neuron

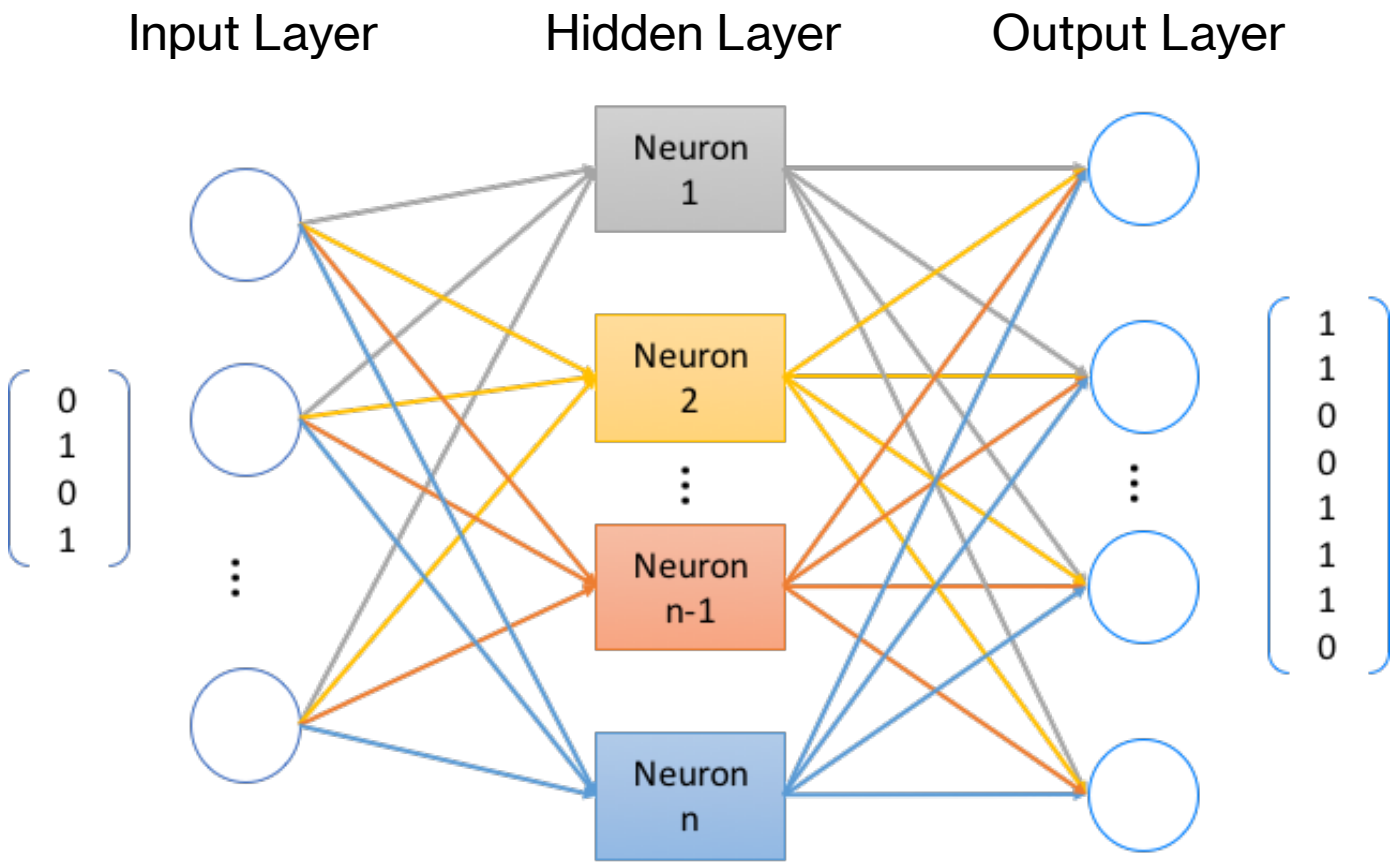
Biological Neuron



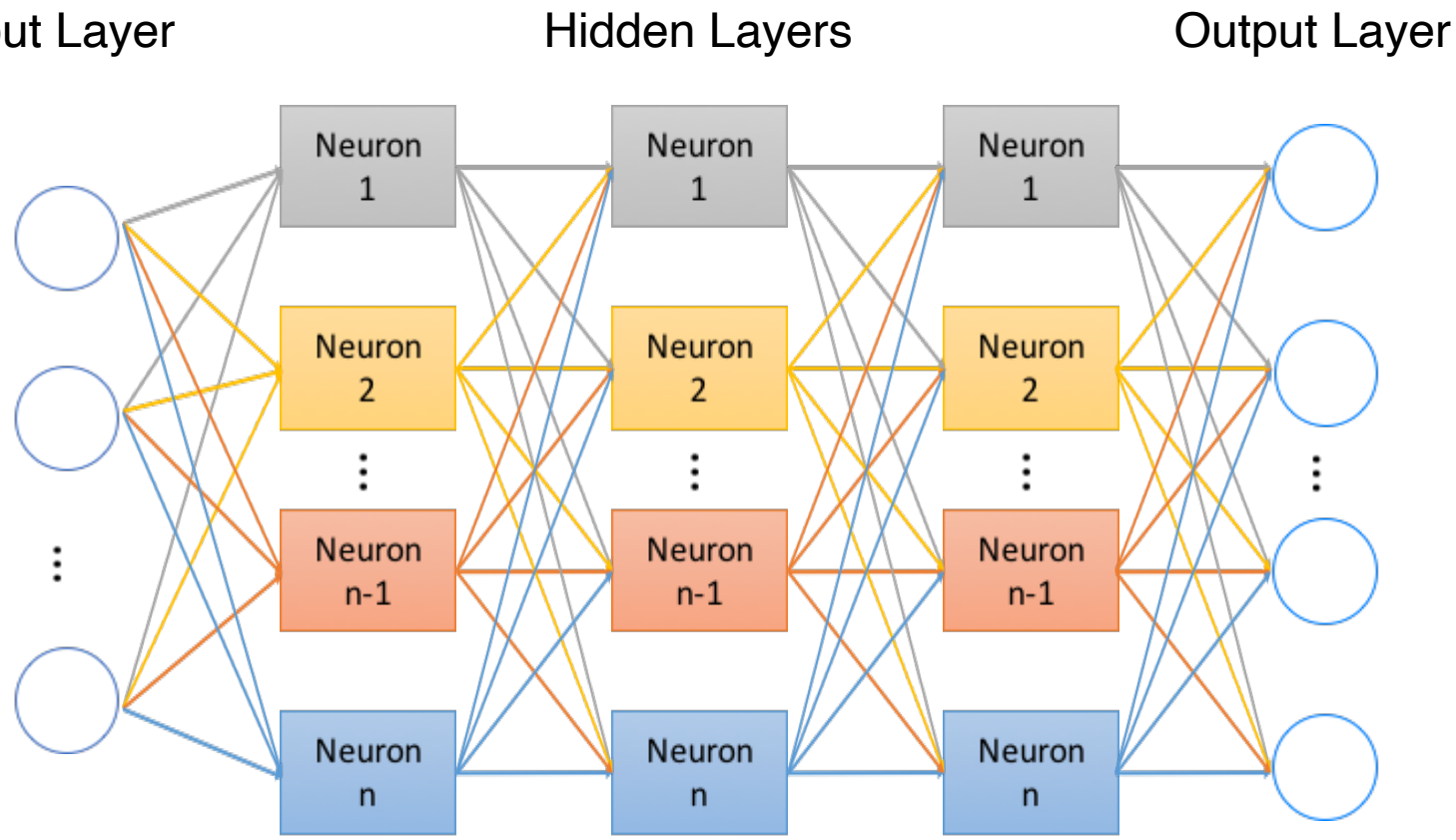
Artificial Neuron



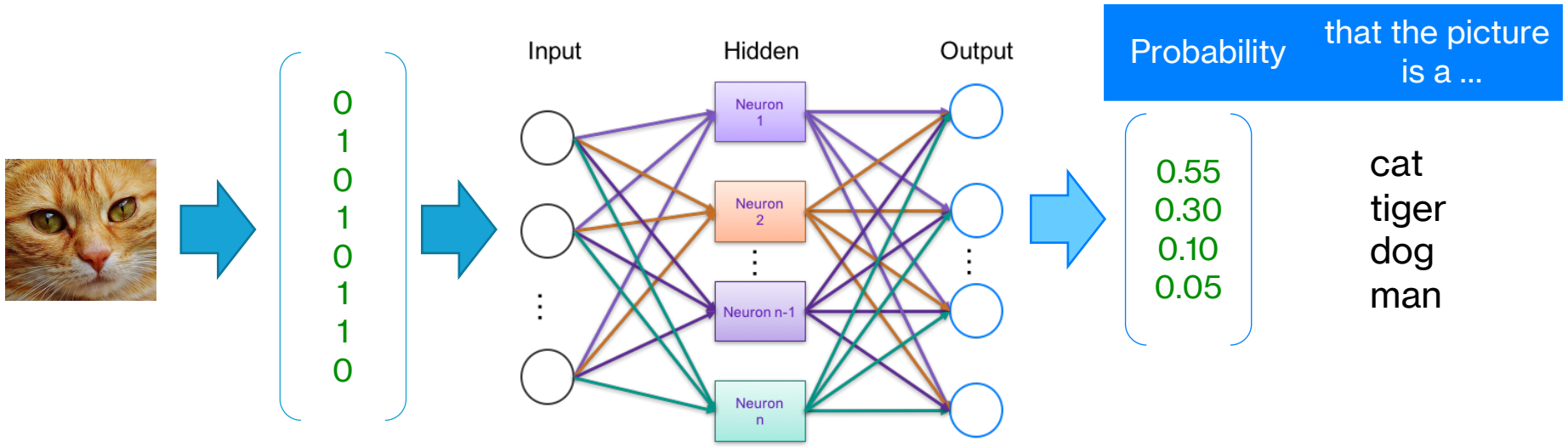
Artificial Neural Network



Deep Neural Network



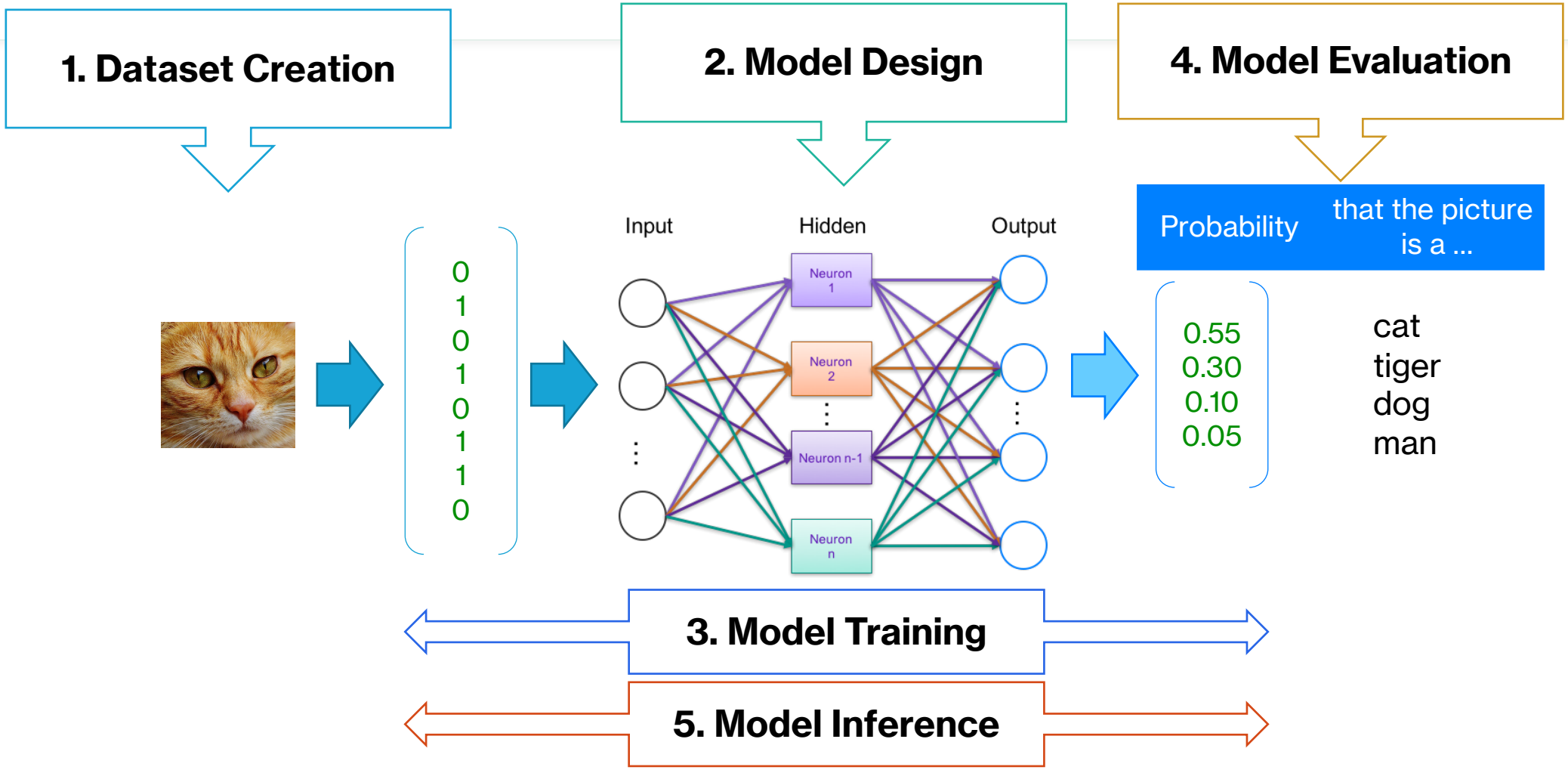
Example: Image Classification



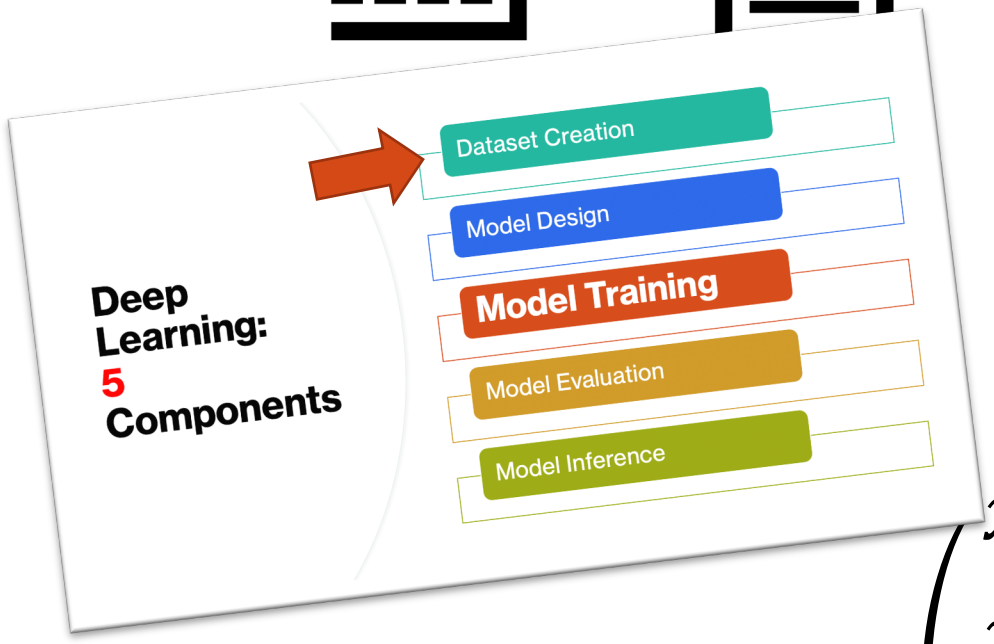
Deep Learning: 5 Components

- Dataset Creation
- Model Design
- **Model Training**
- Model Evaluation
- Model Inference

Example: Image Classification



How can we train a model?

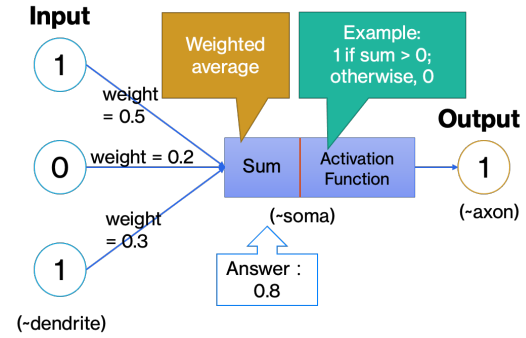


$$x = x_1 \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \quad \boldsymbol{X} = \begin{pmatrix} X_{11} & \cdots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{m1} & \cdots & X_{mn} \end{pmatrix}$$

Manipulation of numbers: scalars, vectors, matrices, and tensors

To get started, let’s revisit the following

A Simple Neuron



Given the input

$$\boldsymbol{x} \in \mathbb{R}^d$$

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$$

We have

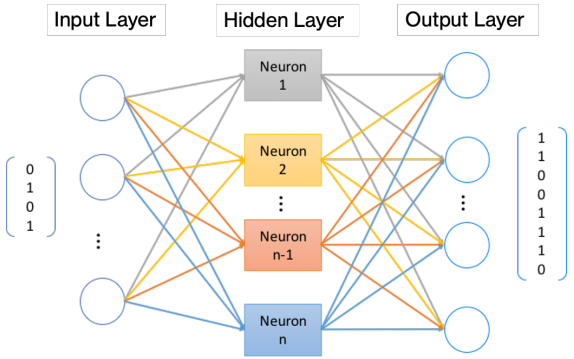
$$\boldsymbol{w} \in \mathbb{R}^d \quad b \in \mathbb{R} \quad \text{and} \quad \sigma(z) = \frac{1}{1+e^{-z}}, \text{ where } z \in \mathbb{R}$$

A sigmoid function as an example

The output is

$$\hat{y} = \sigma(\boldsymbol{w}^T\boldsymbol{x} + b), \text{where } \hat{y} \in \mathbb{R}$$

A Simple Network



Given the input

$$\boldsymbol{x} \in \mathbb{R}^d$$

We have

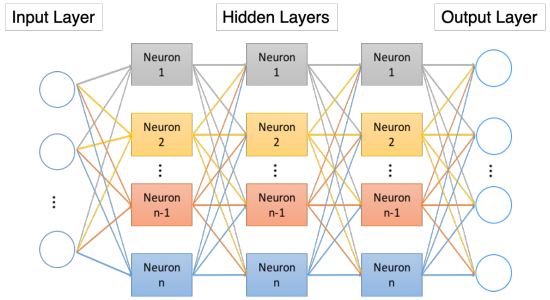
Element-wise operation here

$\boldsymbol{W} \in \mathbb{R}^{d \times h}$ $\boldsymbol{b} \in \mathbb{R}^h$ and $\sigma(\boldsymbol{z}) = \frac{1}{1+e^{-z}}$, where $\boldsymbol{z} \in \mathbb{R}^h$

The output is

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}), \text{where } \hat{\boldsymbol{y}} \in \mathbb{R}^h$$

A Simple Deep Neural Network



Given the input $\boldsymbol{x} \in \mathbb{R}^d$

We have

$\boldsymbol{W_1} \in \mathbb{R}^{d \times h_1}$ $\quad \boldsymbol{b_1} \in \mathbb{R}^{h_1}$

$\boldsymbol{W_2} \in \mathbb{R}^{h_1 \times h_2}$ $\quad \boldsymbol{b_2} \in \mathbb{R}^{h_2}$

... ...

$\boldsymbol{W_n} \in \mathbb{R}^{h_{n-1} \times h_n}$ $\quad \boldsymbol{b_n} \in \mathbb{R}^{h_n}$

and $\sigma(\boldsymbol{z}) = \frac{1}{1+e^{-\boldsymbol{z}}}$, where $\boldsymbol{z} \in \mathbb{R}^{h_i}$

We usually have more than one $\sigma(x)$

The output is

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{W_n^T}\left(\ldots\sigma\left(\boldsymbol{W_2^T}\left(\sigma\left(\boldsymbol{W_1^T x} + \boldsymbol{b_1}\right)\right) + \boldsymbol{b_2}\right)\ldots\right) + \boldsymbol{b_n}), \text{ where } \hat{\boldsymbol{y}} \in \mathbb{R}^{h_n}$$

$$\hat{y} = \sigma(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b), \text{where } \hat{y} \in \mathbb{R}$$

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{b}), \text{where } \hat{\boldsymbol{y}} \in \mathbb{R}^{h}$$

$$\hat{\boldsymbol{y}} = \sigma(\boldsymbol{W}_{\boldsymbol{n}}^{\boldsymbol{T}}\left(\ldots\sigma\left(\boldsymbol{W}_{\boldsymbol{2}}^{\boldsymbol{T}}\left(\sigma\left(\boldsymbol{W}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{b}_{\boldsymbol{1}}\right)\right) + \boldsymbol{b}_{\boldsymbol{2}}\right)\ldots\right) + \boldsymbol{b}_{\boldsymbol{n}}), \text{where } \hat{\boldsymbol{y}} \in \mathbb{R}^{h_n}$$

As we'd like to focus on model training rather than model design, we have

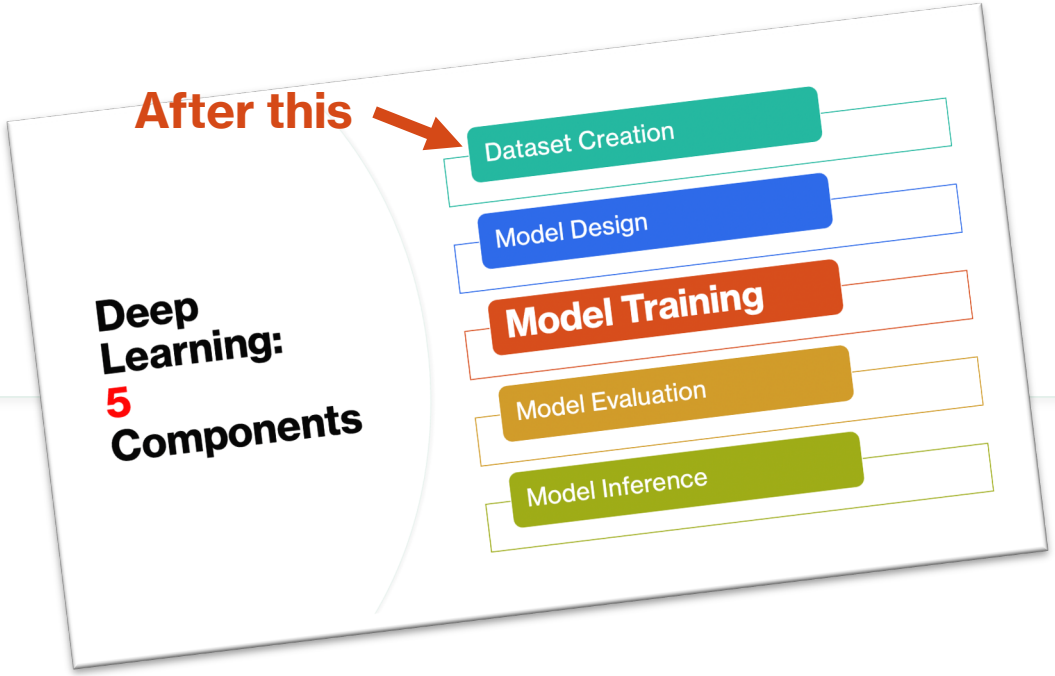
$$\hat{\boldsymbol{y}} = f_{\theta}(\boldsymbol{x}), \text{where } \theta = (\boldsymbol{W}_{\boldsymbol{1}}, \boldsymbol{b}_{\boldsymbol{1}}, \boldsymbol{W}_{\boldsymbol{2}}, \boldsymbol{b}_{\boldsymbol{2}}, \ldots, \boldsymbol{W}_{\boldsymbol{n}}, \boldsymbol{b}_{\boldsymbol{n}})$$

Model Design:
How is f like

Model Training:
Find our parameters ☺

We’re ready to train our model!

Data First!



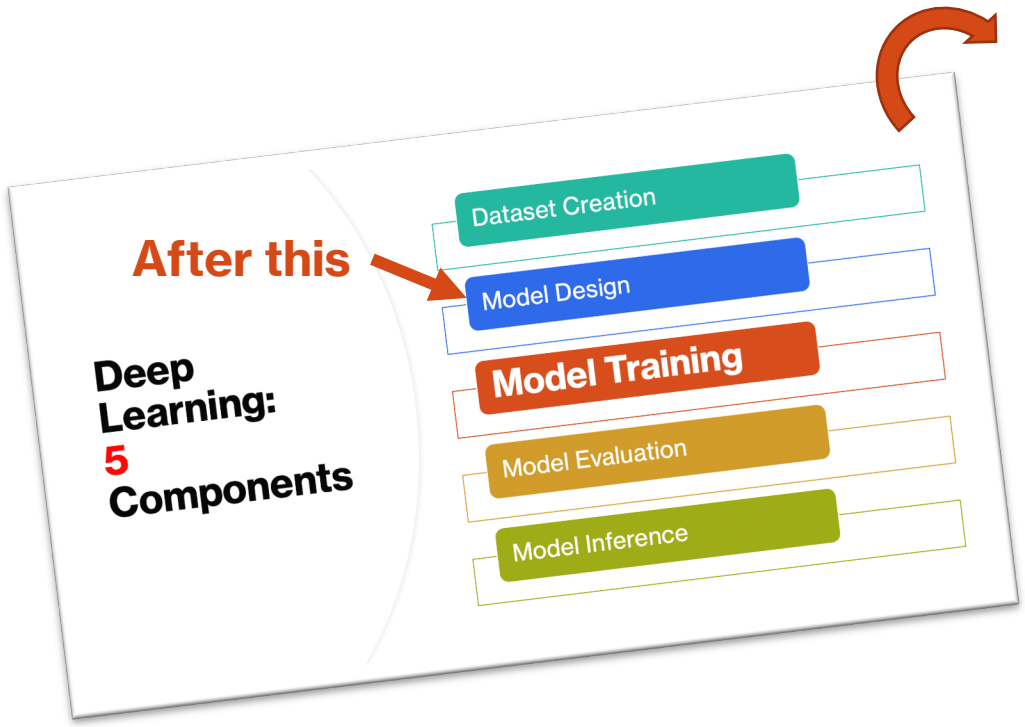
For example, we have

$$\{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^{m} = \{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), \dots, (\boldsymbol{x_m}, \boldsymbol{y_m})\}$$

(e.g., $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^h$ for all i)

We can also have other options like $x_i \in [0,255]^d$ and $y_i \in \{0, \dots, |class|\}$, depending on the nature of our task.

Initialize our parameters



We have

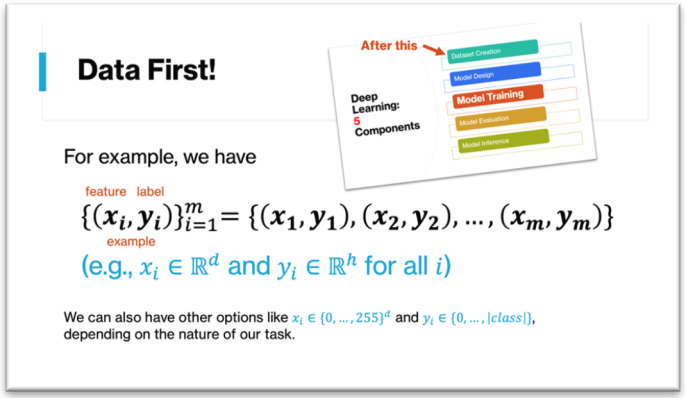
$$\hat{\boldsymbol{y}} = f_\theta(\boldsymbol{x}),$$
where $\theta = (\boldsymbol{W_1}, \boldsymbol{b_1}, \boldsymbol{W_2}, \boldsymbol{b_2}, \ldots, \boldsymbol{W_n}, \boldsymbol{b_n})$

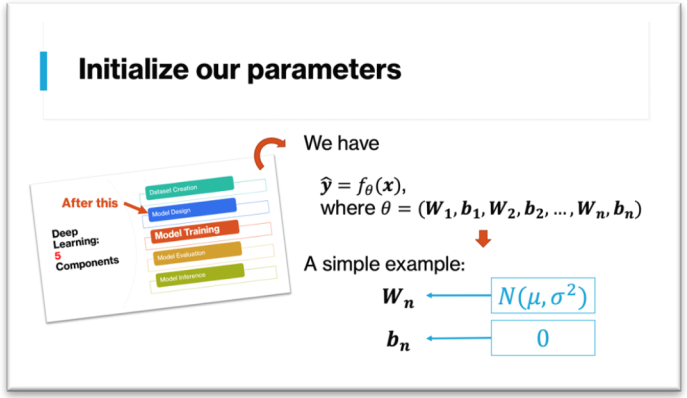
A simple example:

$\boldsymbol{W_n}$ ← $N(\mu, \sigma^2)$

$\boldsymbol{b_n}$ ← 0

Forward propagation





$$\{(x_i, y_i)\}_{i=1}^{m} \Rightarrow f_\theta(x), \text{ where } \theta = (W_1, b_1, W_2, b_2, \dots, W_n, b_n) \Rightarrow \hat{y}_i$$

Initialized

Loss computation

$\{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^{m}$ → $f_\theta(\boldsymbol{x})$, where $\theta = (\boldsymbol{W_1}, \boldsymbol{b_1}, \boldsymbol{W_2}, \boldsymbol{b_2}, \ldots, \boldsymbol{W_n}, \boldsymbol{b_n})$ → $\hat{\boldsymbol{y}}_i$

↓

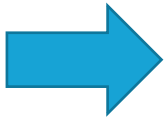
$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{m}\sum_{i}^{m} l(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i)$$

Mean Absolute Error (MAE), Mean Squared Error (MSE), Cross-entropy Loss and more

Parameter adjustment

Let's use this

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

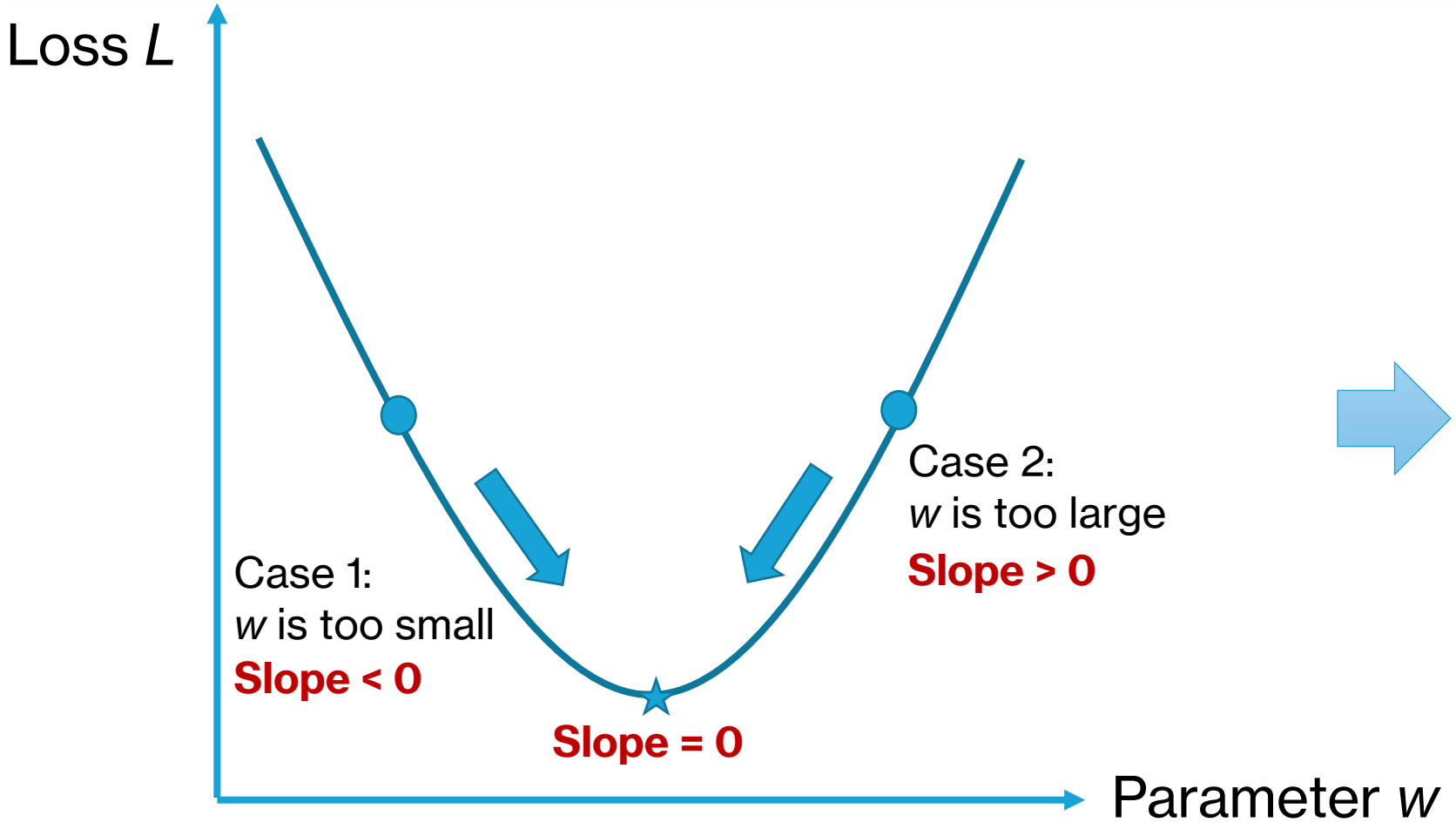


How?

To "improve" our parameters

$$\theta = (\boldsymbol{W_1}, \boldsymbol{b_1}, \boldsymbol{W_2}, \boldsymbol{b_2}, \ldots, \boldsymbol{W_n}, \boldsymbol{b_n})$$

Consider this

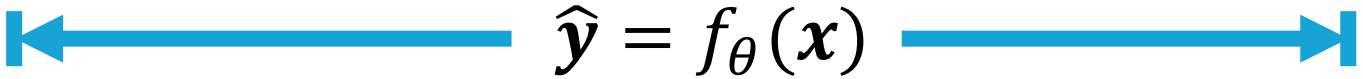


1. Find $\frac{\partial L(\hat{\boldsymbol{y}}, \boldsymbol{y})}{\partial \theta}$

2. Adjust θ

Back propagation

An Example



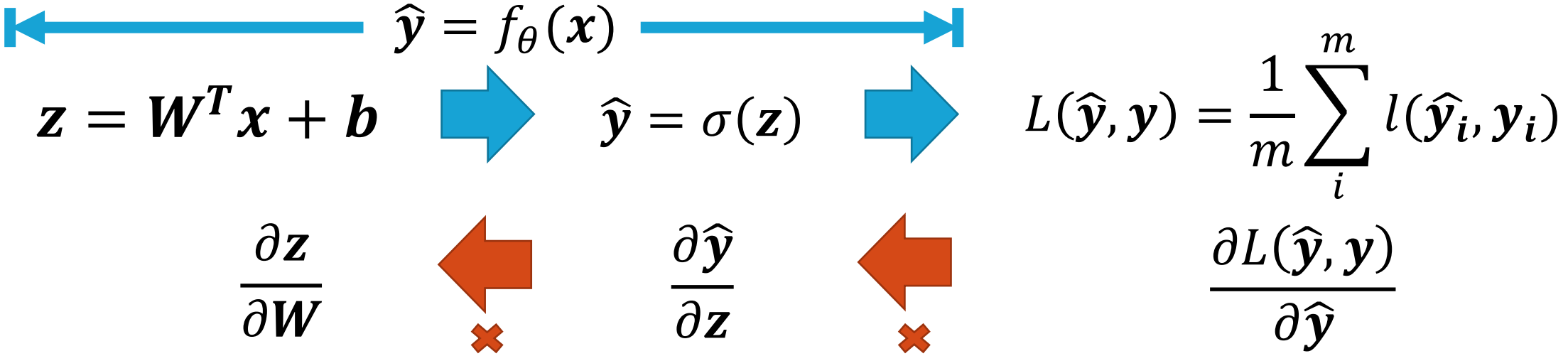
$$z = W^T x + b \Rightarrow \hat{y} = \sigma(z) \Rightarrow L(\hat{y}, y) = \frac{1}{m}\sum_{i}^{m} l(\hat{y}_i, y_i)$$

How can we find $\frac{\partial L(\hat{y}, y)}{\partial W}$?

Chain rule!

Back propagation

An Example



$$= \frac{\partial L(\hat{y}, y)}{\partial W}$$

Optimization

A Simple Example: Stochastic/Mini-batch Gradient Descent

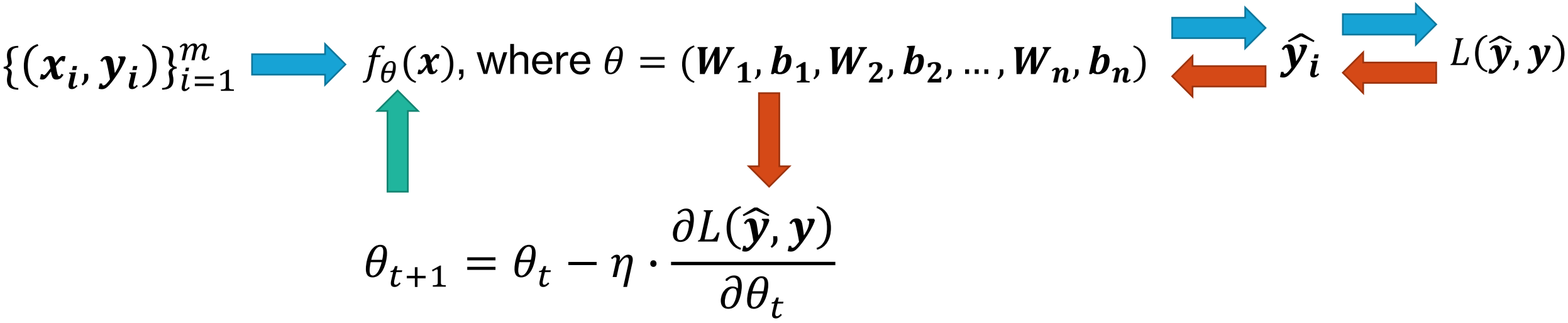
For all $\theta = (\boldsymbol{W_1}, \boldsymbol{b_1}, \boldsymbol{W_2}, \boldsymbol{b_2}, \dots, \boldsymbol{W_n}, \boldsymbol{b_n})$ at step t, we have

Learning rate

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\partial L(\hat{\boldsymbol{y}}, \boldsymbol{y})}{\partial \theta_t}$$

From back propagation

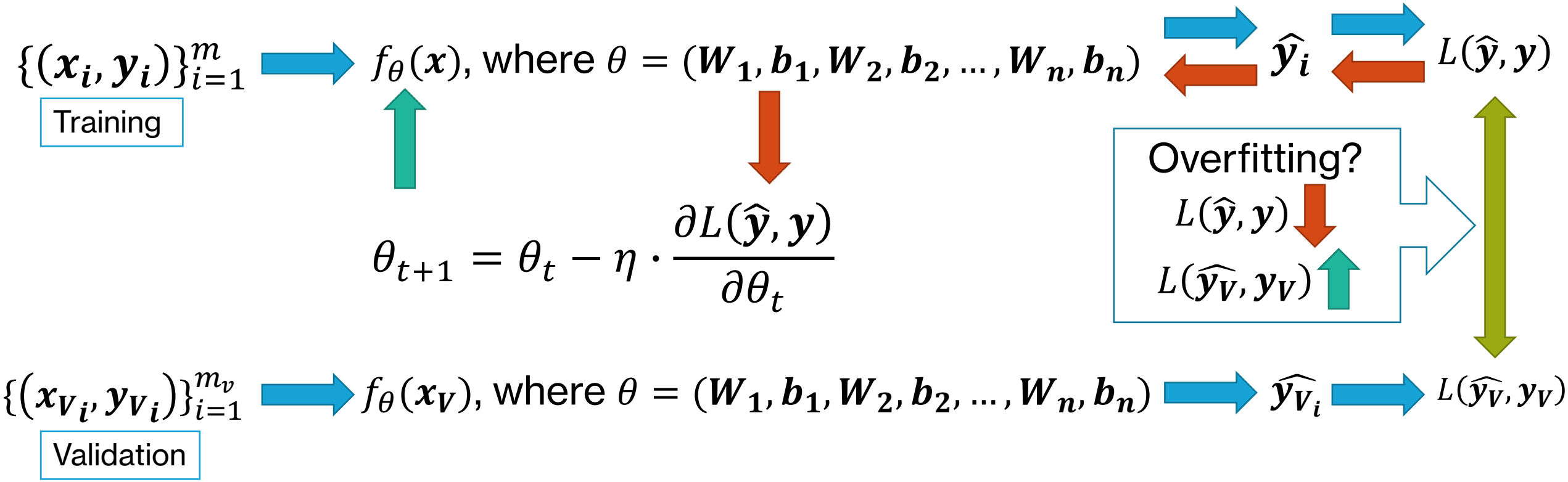
Iteration


$\{(x_i, y_i)\}_{i=1}^{m}$
$f_\theta(x)$, where $\theta = (W_1, b_1, W_2, b_2, \ldots, W_n, b_n)$
$\hat{y}_i$
$L(\hat{y}, y)$
$\theta_{t+1} = \theta_t - \eta \cdot \frac{\partial L(\hat{y}, y)}{\partial \theta_t}$

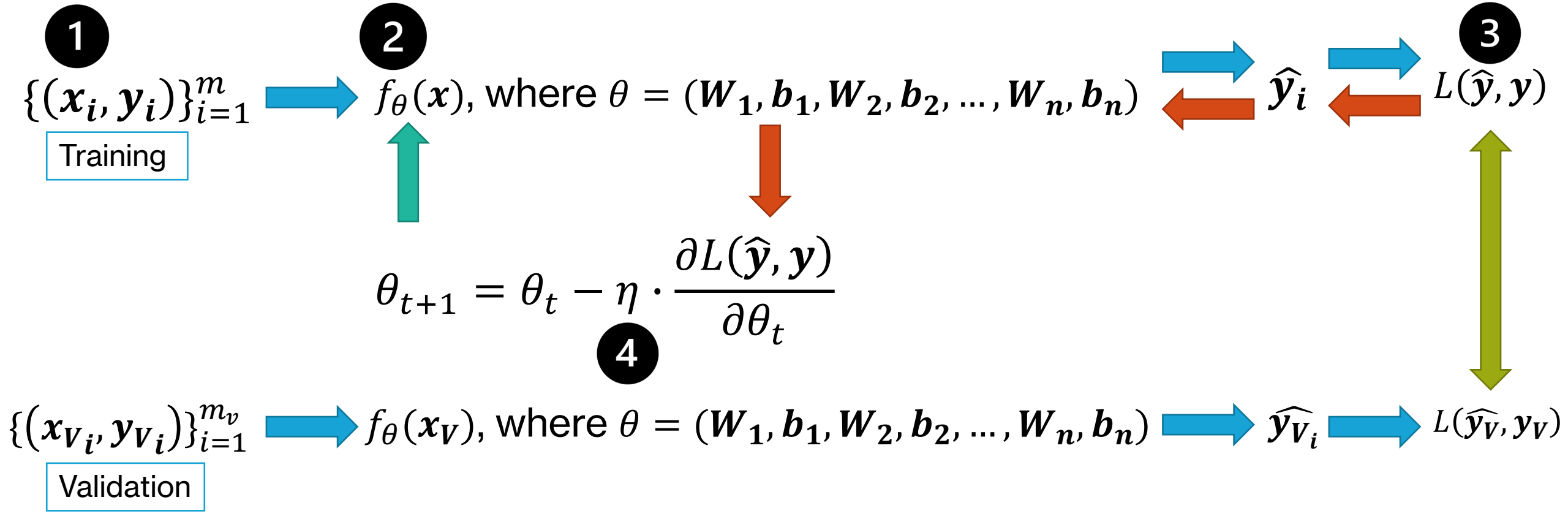
When should we stop training?

- No. of steps
- No. of epochs
- Early stopping

Early Stopping



Deep Learning Tasks



Other Interesting Issues

- Programming for Data Training
- Different Modes of Training: Supervised Training and Beyond
- Optimization Algorithms
- Design of Training Tasks
- Advanced Training Methods (e.g., training using trained models)
- Specific Examples of Training (e.g., translation, image processing, regression, time series forecasting, and more): To be explored in conjunction with dataset creation and model building

Thank you!