Deep Learning

Lisa

Deep Learning = Learning Hierachical Representation

LeCun

REVIEW

doi:10.1038/nature14539

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

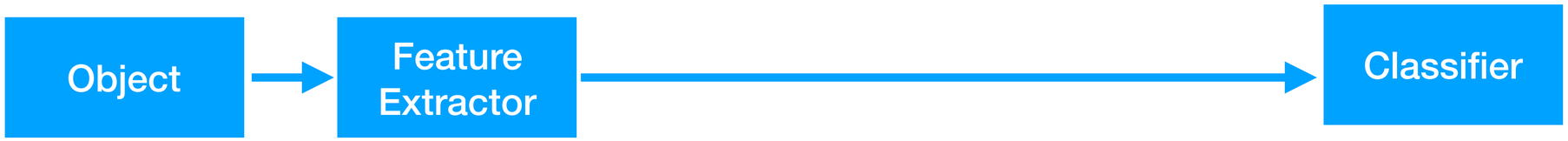
Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning.

Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a pattern-recognition or machine-learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector

intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition[1–4] and speech recognition[5–7], it has beaten other machine-learning techniques at predicting the activity of potential drug molecules[8], analysing particle accelerator data[9,10], reconstructing brain circuits[11], and predicting the effects of mutations in non-coding DNA on gene expression and disease[12,13]. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding[14], particularly topic classification, sentiment analysis, question answering[15] and language translation[16,17].

We think that deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are

Traditional Pattern Recognition



Modern Pattern Recognition (Mainstream)

Deep Learning

hierachical representation

- Image recognition

Pixel → edge → texton → motif → part → object

- Text recognition

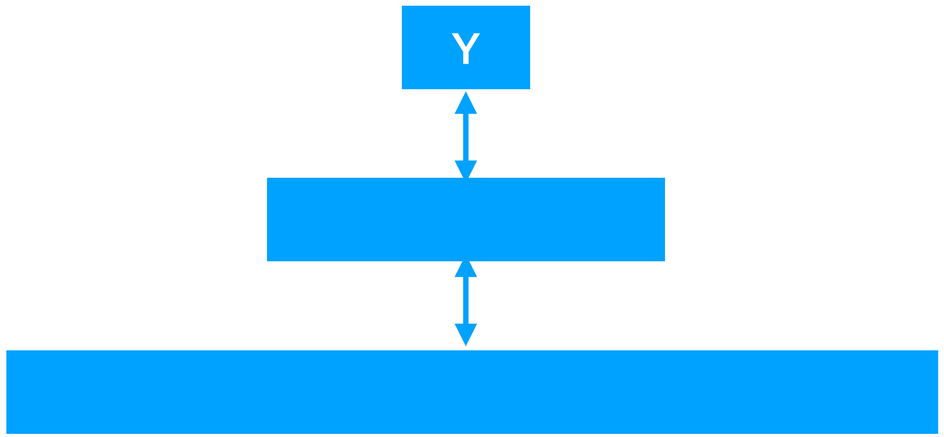
Character → word → word group → clause → sentence → story

- Speech recognition

Sample → spectral band → sound → … → phone → phoneme → word

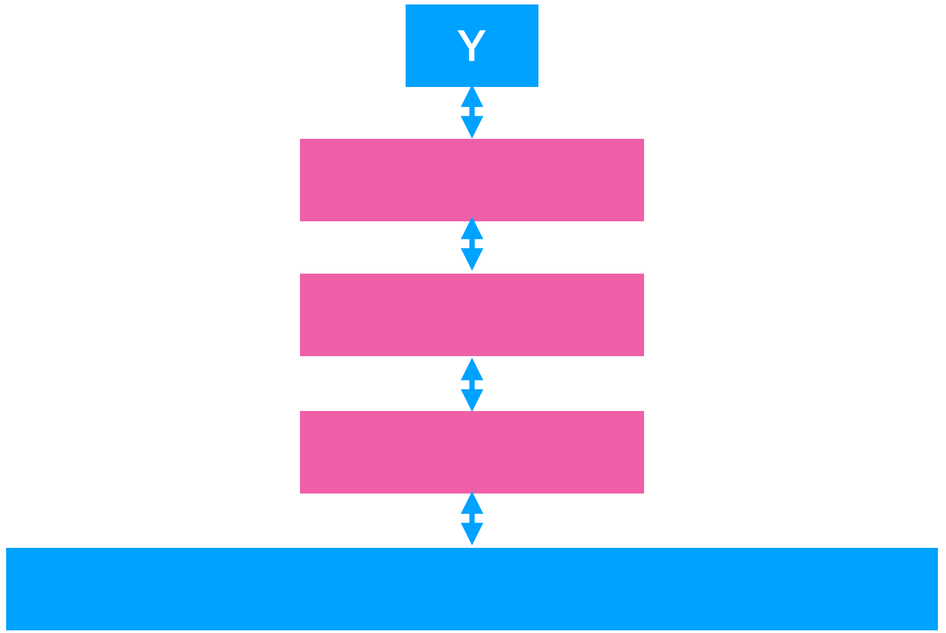
“Shallow & Wide” vs “Deep & narrow”

“more memory” vs “more time”



$$f_1(f_2(x))$$

complicated

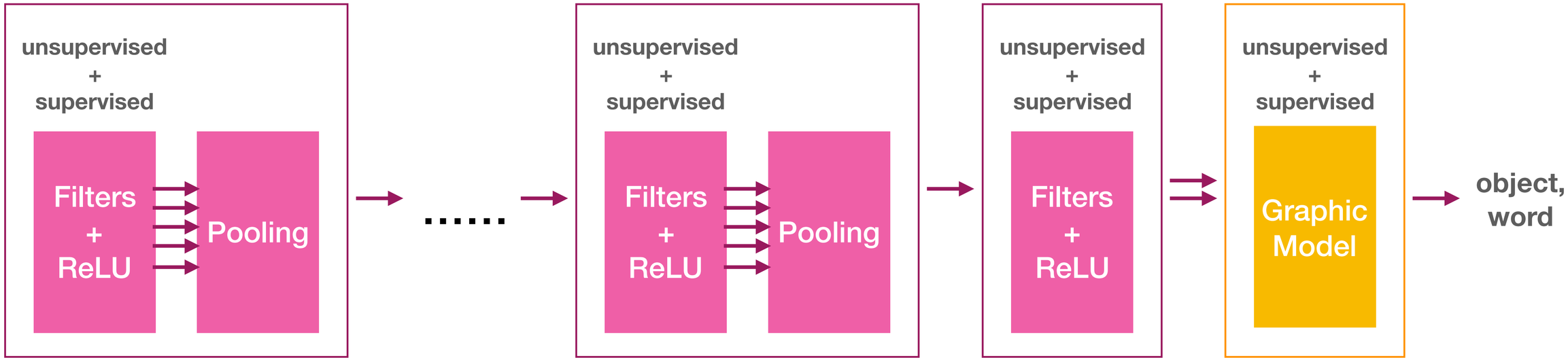


$$f_1\left(f_2\left(f_3(\dots)\right)\right)$$

simple(simple(…))

Deep Learning = Deep structured prediction

multi-layer unsupervised supervised

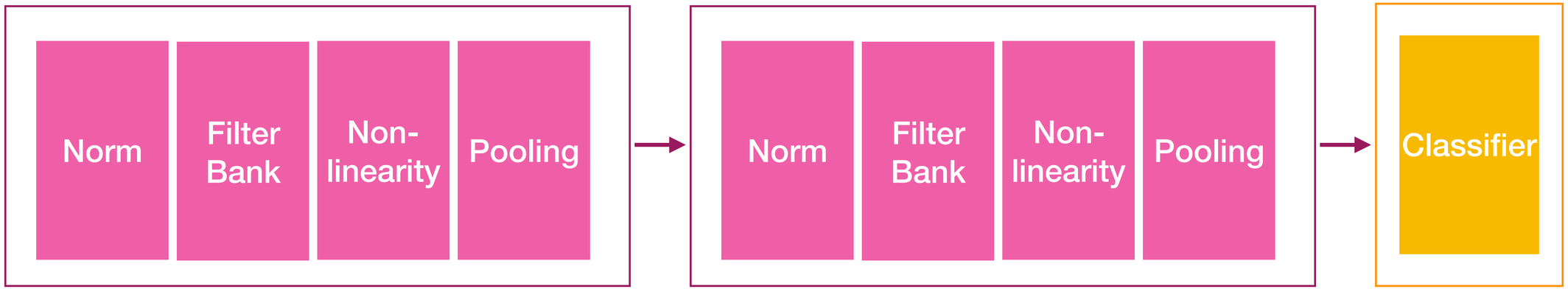


Overall architecture

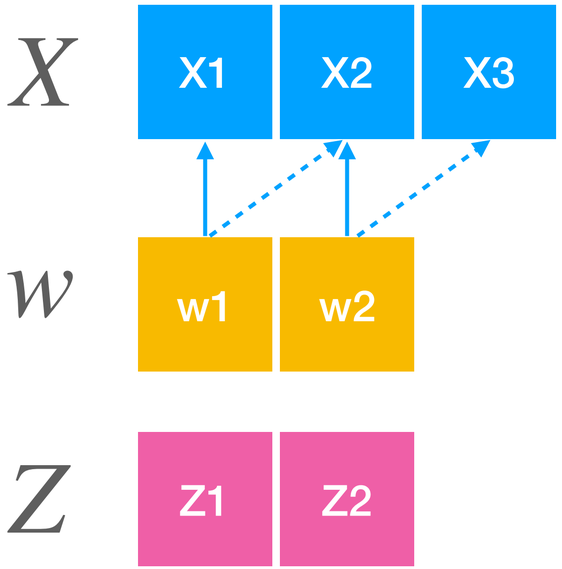
- Convolution (Filter Bank): dimension expansion, project on overcomplete basis
- Non Linearity: sparsification, saturation, lateral inhabition
  ReLu (Rectified linear unit), component-wise shrinkage, tanh
- Pooling or Sub Sampling: : aggregation over space or feature type

$$MAX : Max\{X_i\}, \quad L_p : \sqrt[p]{x_n^p}, \quad log : \frac{1}{b} log(\sum e^{bX_i})$$

- Classification (Fully Connected Layer)



Convolution (Similarity)



Similarity

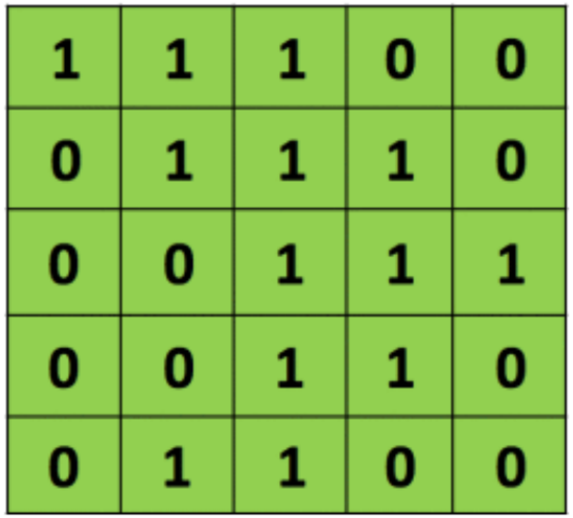
$$\begin{cases} z_1 = w_1 x_1 + w_2 x_2 \\ z_2 = w_1 x_2 + w_2 x_3 \end{cases}$$

Convolution

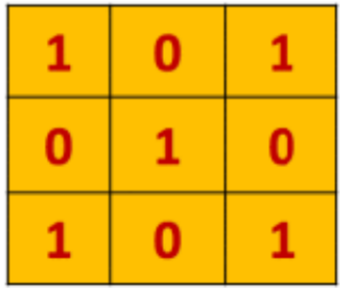
$$\begin{cases} z_1 = w_2 x_1 + w_1 x_2 \\ z_2 = w_2 x_2 + w_1 x_3 \end{cases}$$

$\overline{w}$

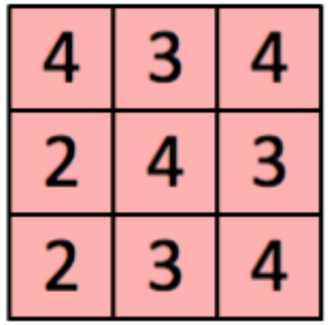
Filter detect different feature: edge detection, sharpen, blur, …



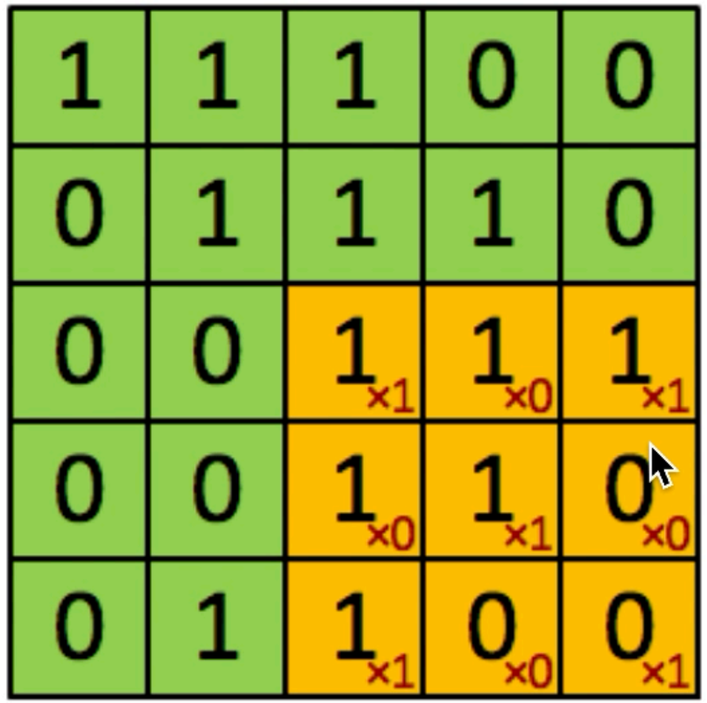
Image



Filter



Convolved Feature



Image

4	3	4
2	4	3
2	3	4

Convolved Feature

The size of the Feature Map (Convolved Feature)

- **Depth:** number of filters
- **Stride:** number of pixels filter matrix jump

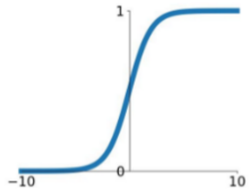
- **Zero-padding:** pad with 0 around the border

Introduce Non Linearity

Activation Functions

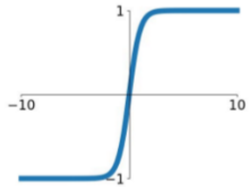
Sigmoid

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



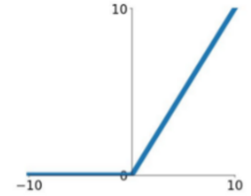
tanh

$$\tanh(x)$$



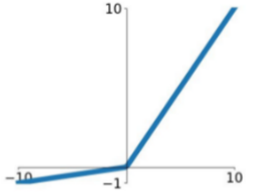
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

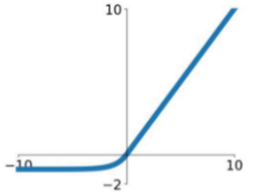


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

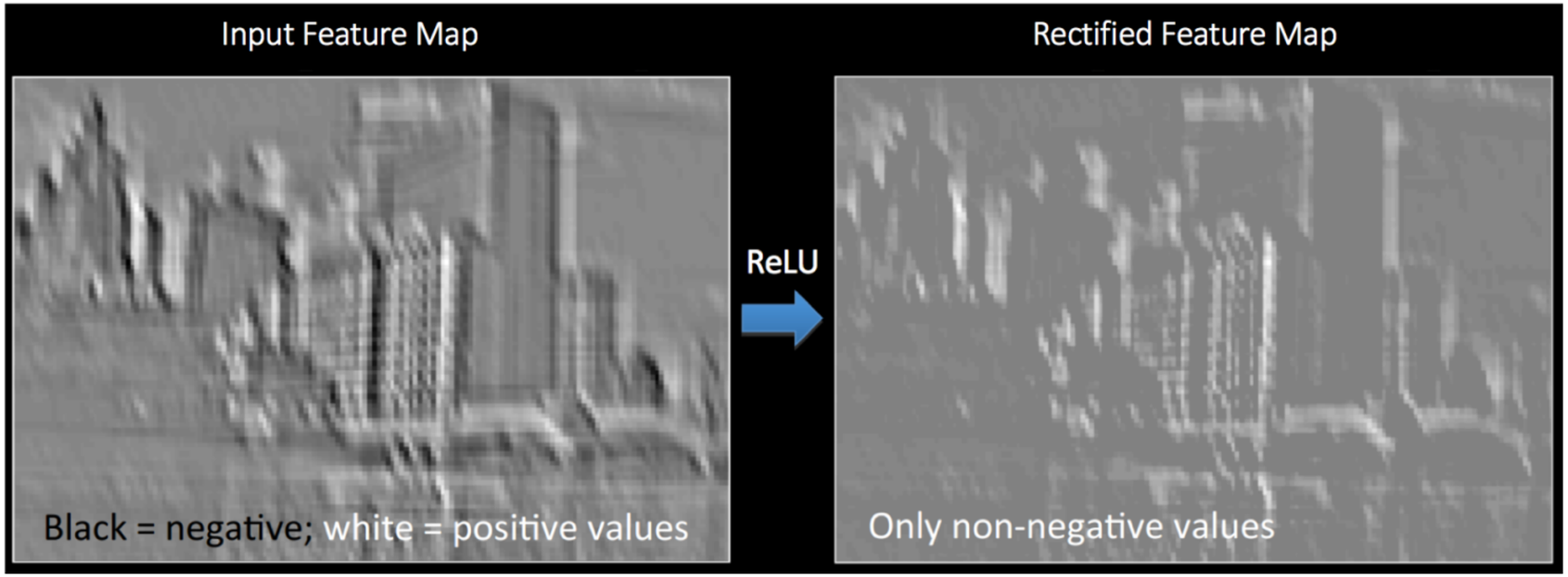
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



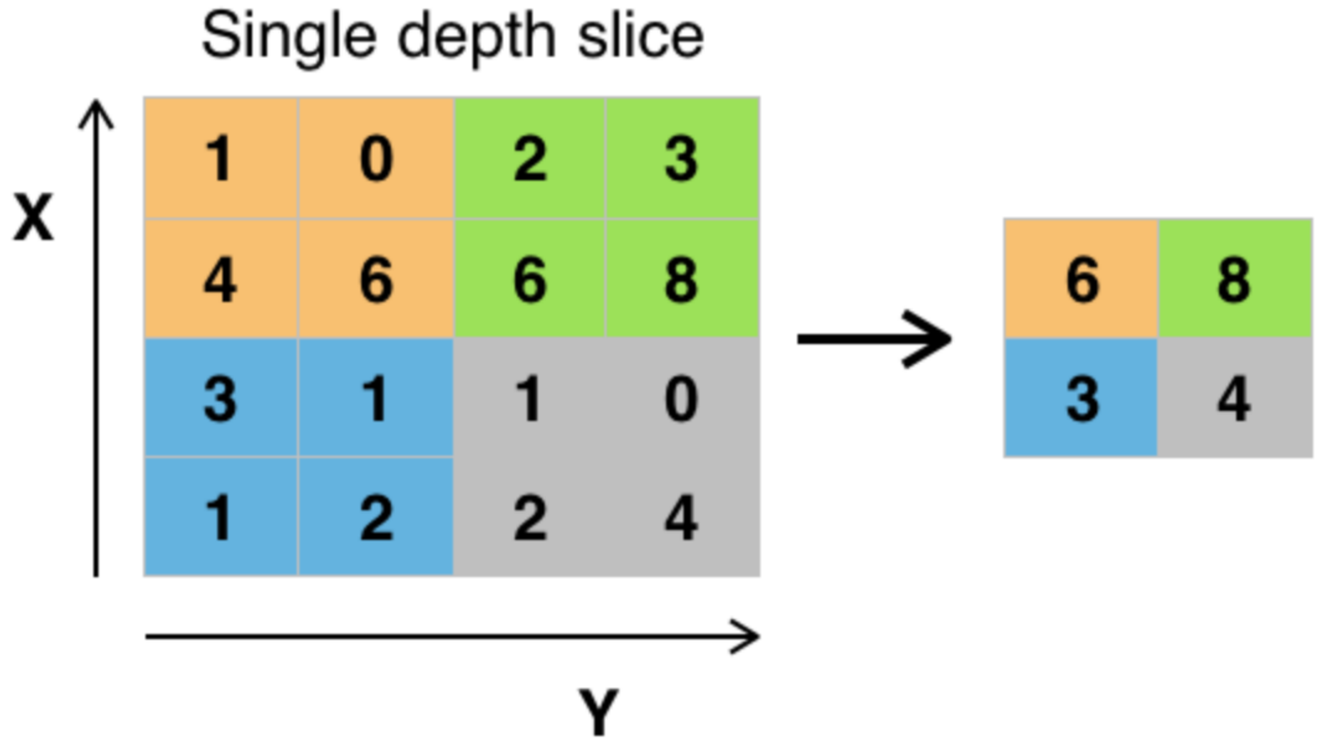
ReLU

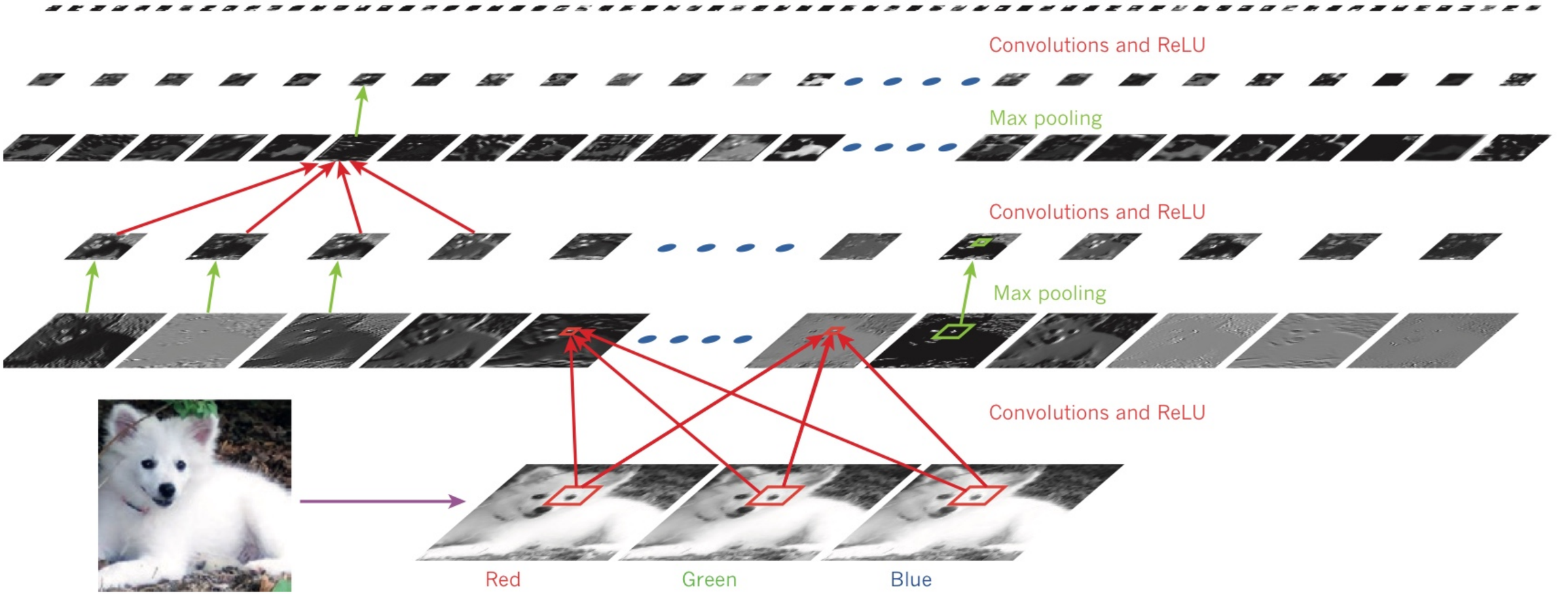
Rectified linear unit



Pooling

reduces the dimensionality and control overfitting

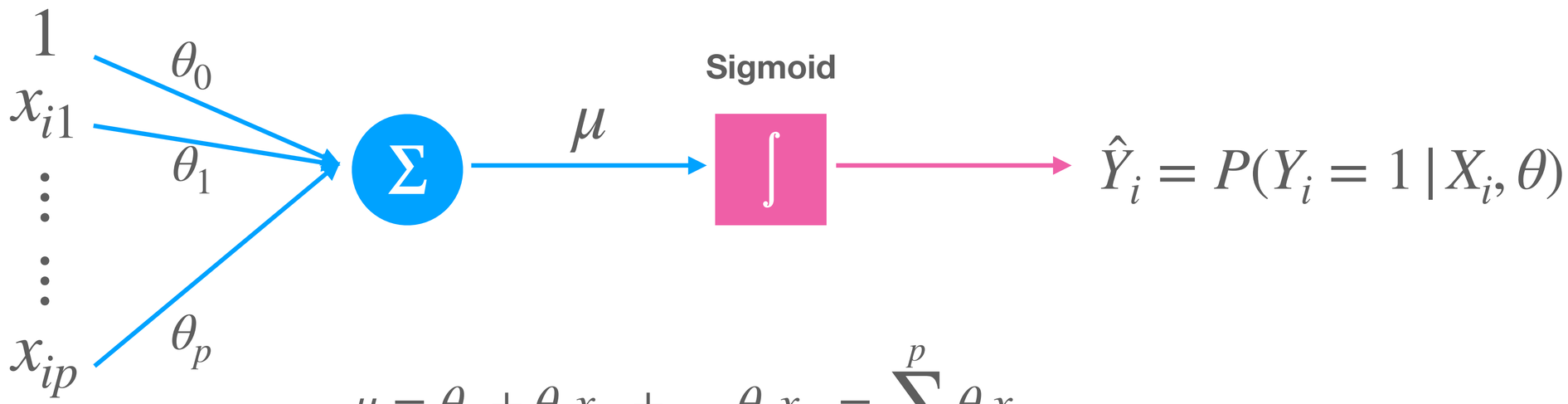



Convolutions and ReLU
Max pooling
Convolutions and ReLU
Max pooling
Convolutions and ReLU
Red
Green
Blue

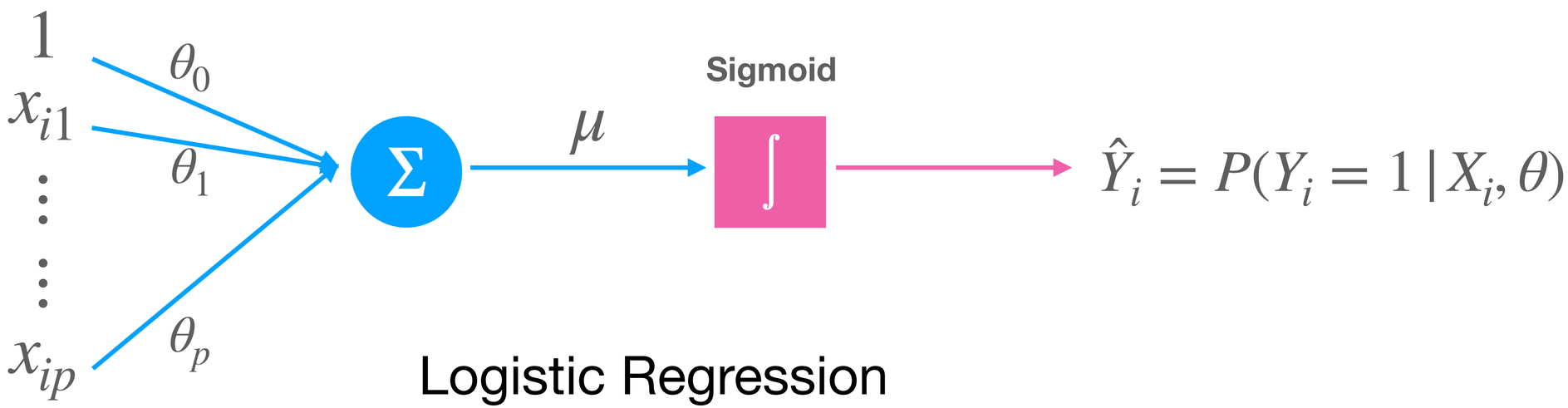
How it works

MLP (1 neuron example)

Multilayer perceptron



$$\mu = \theta_0 + \theta_1 x_{i1} + \ldots \theta_p x_{ip} = \sum_{j=1}^{p} \theta_j x_{ij}$$

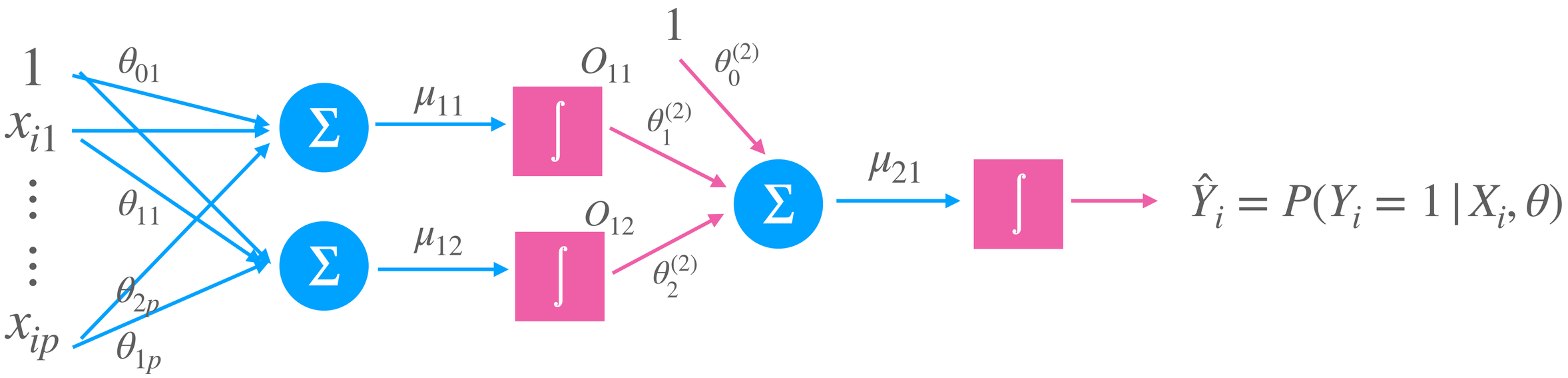


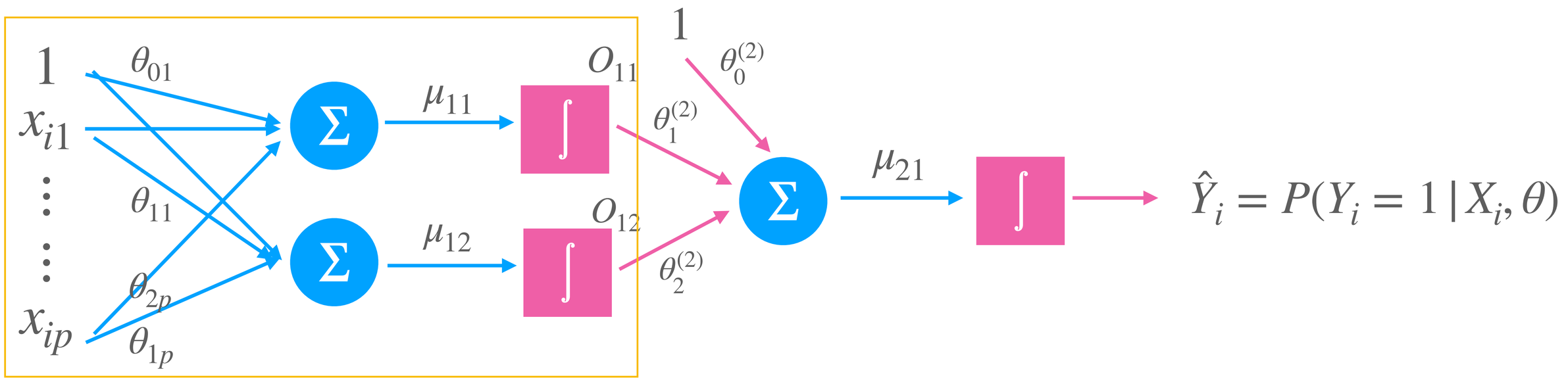
Logistic Regression

$$\mu = \theta_0 + \theta_1 x_{i1} + \ldots \theta_p x_{ip}$$

$$\hat{Y}_i = \frac{e^{\mu}}{1 + e^{\mu}} = \frac{1}{1 + e^{-\mu}} = \frac{1}{1 + e^{-\theta_0 - \theta_1 x_{i1} - \ldots - \theta_p x_{ip}}} = P(Y_i \mid X_i, \theta)$$

MLP (3 neurons, 2 layers)

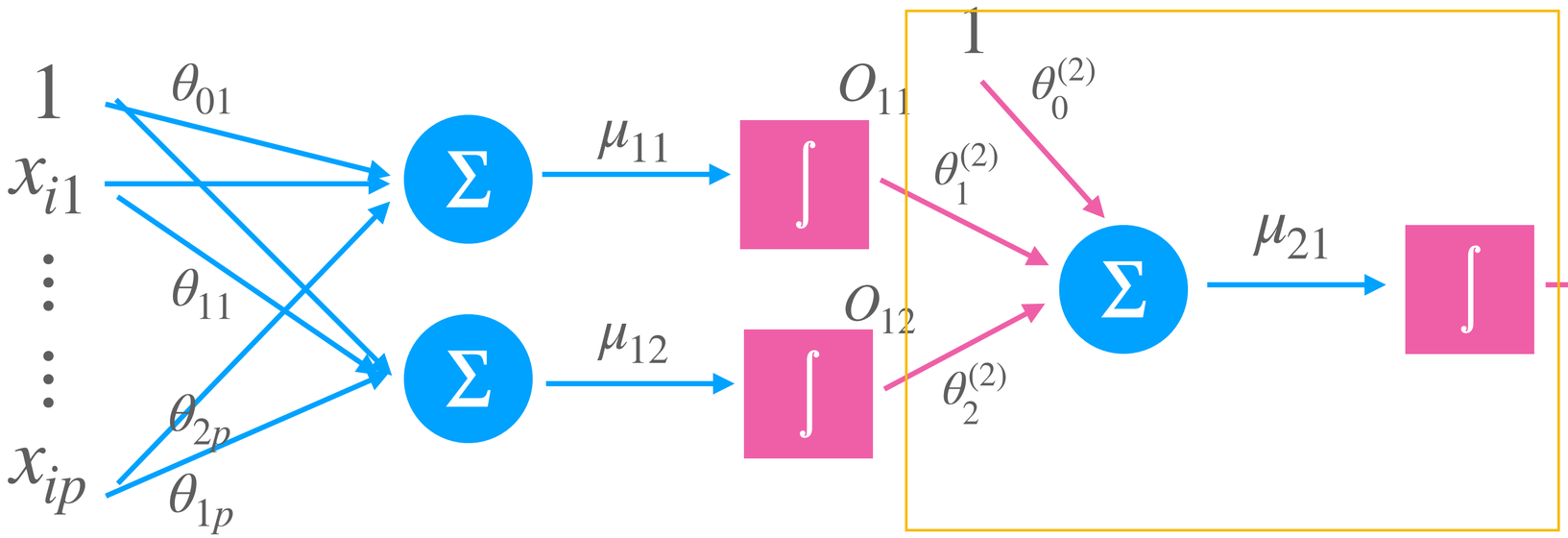




$$\begin{cases} \mu_{11} = \theta_{10} + \theta_{11} x_{i1} + \ldots \theta_{1p} x_{ip} \\ \mu_{12} = \theta_{20} + \theta_{21} x_{i1} + \ldots \theta_{2p} x_{ip} \end{cases}$$

Layer 1

$$\begin{cases} O_{11} = \dfrac{e^{\mu_{11}}}{1 + e^{\mu_{11}}} = \dfrac{1}{1 + e^{-\mu_{11}}} \\ O_{12} = \dfrac{e^{\mu_{12}}}{1 + e^{\mu_{12}}} = \dfrac{1}{1 + e^{-\mu_{12}}} \end{cases}$$

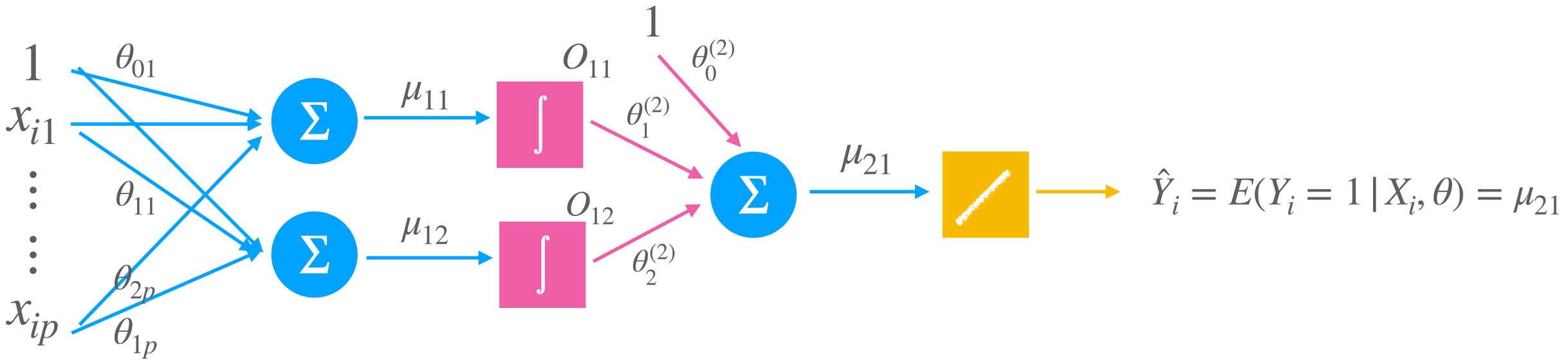


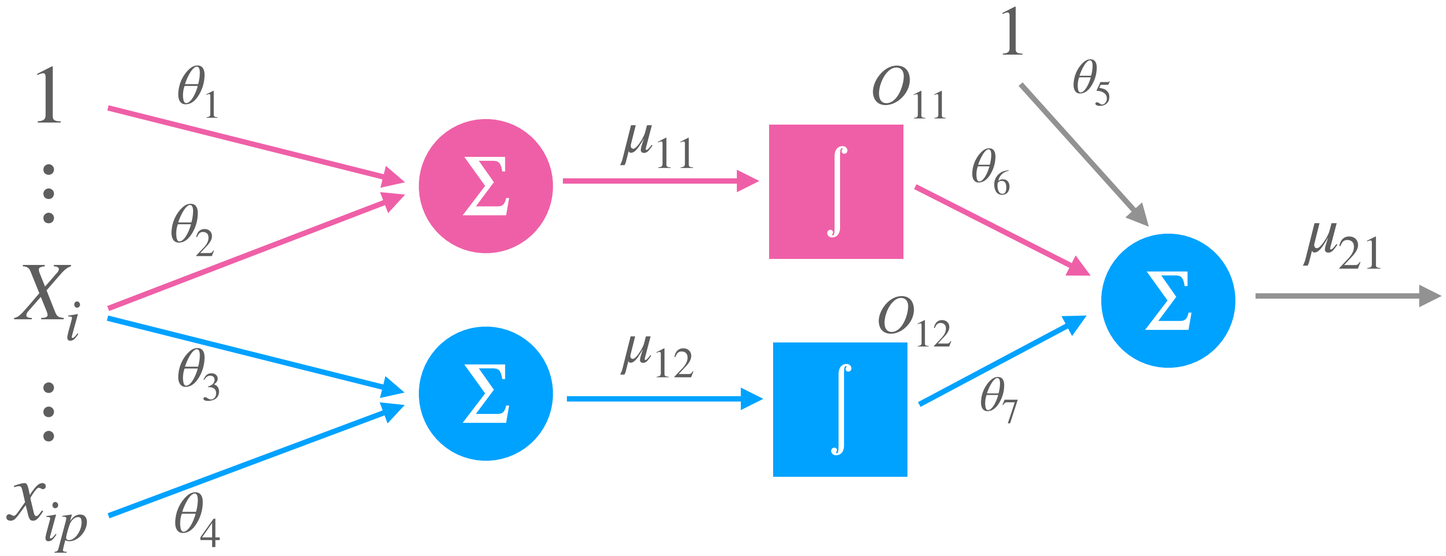
Layer 2

$$\mu_{21} = \theta_0^{(2)} + \theta_1^{(2)} O_{11} + \theta_2^{(2)} O_{12}$$

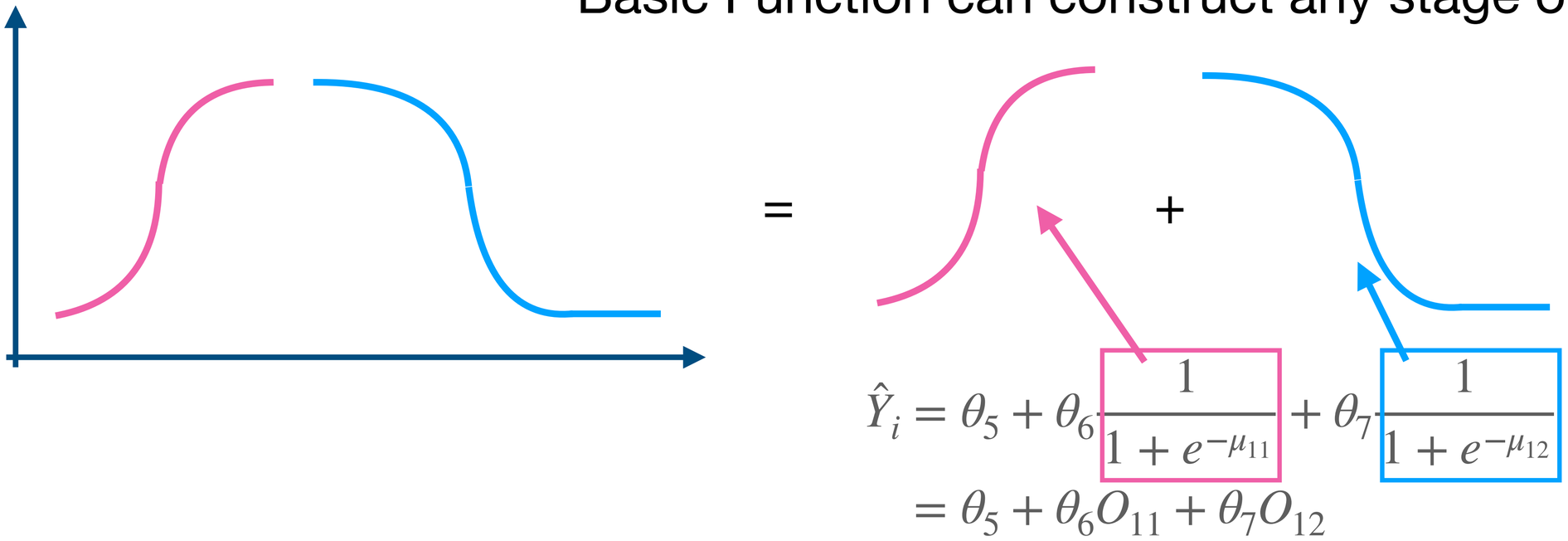
$$\hat{Y}_i = \frac{1}{1 + e^{-\mu_{21}}}$$

MLP Regression

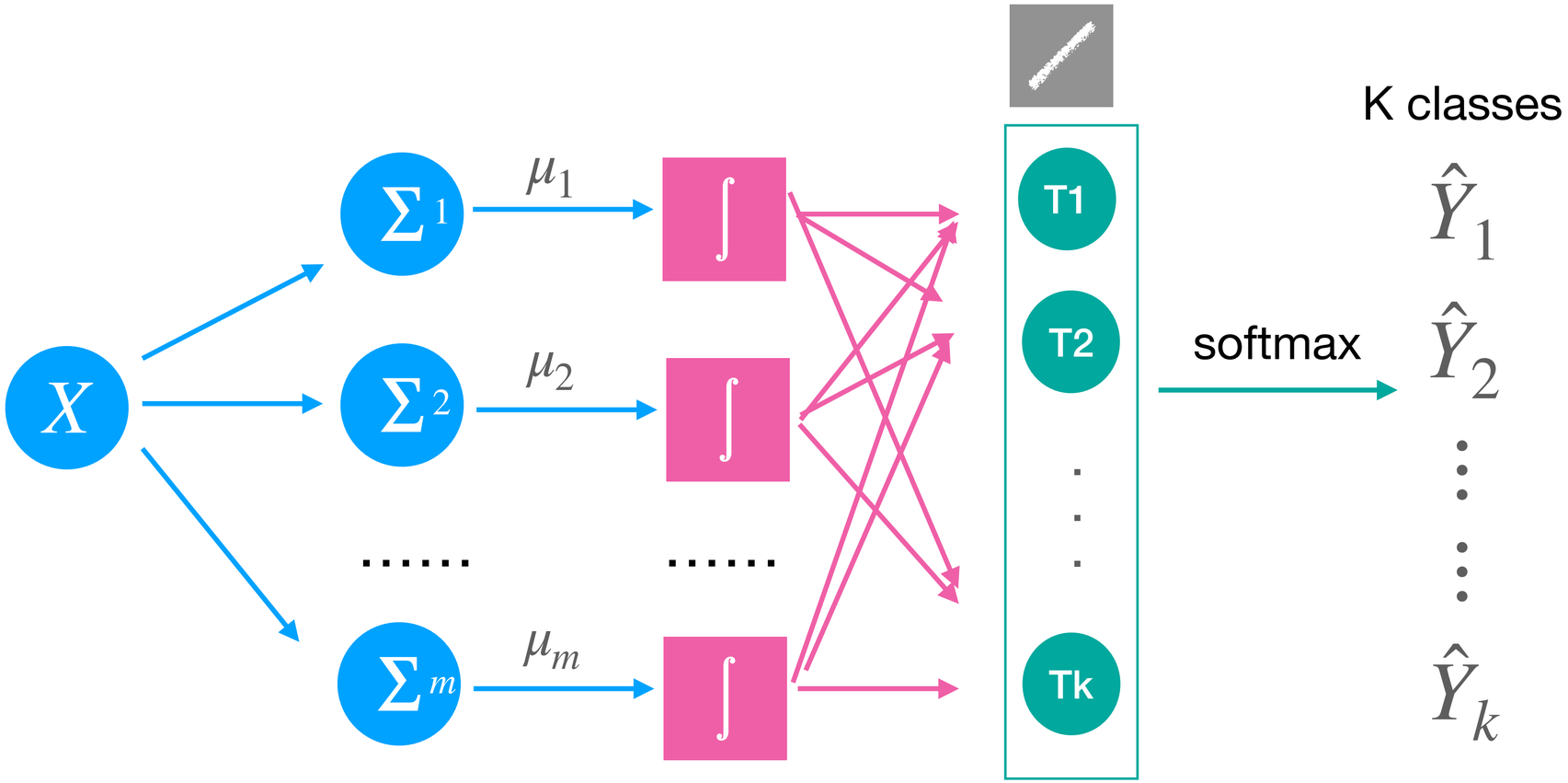




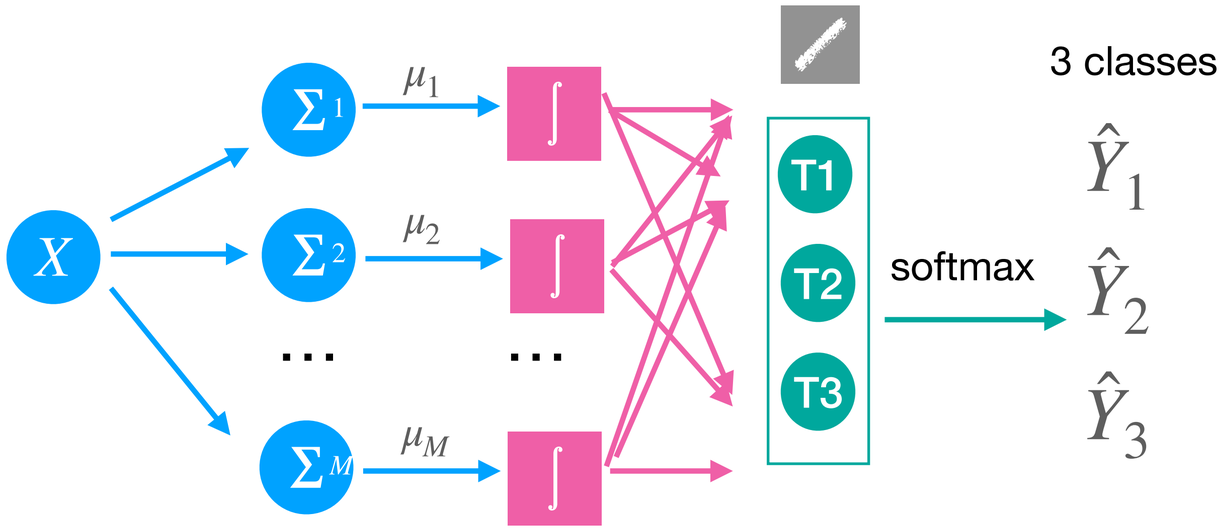
Basic Function can construct any stage of classifier !



Multi-class classification



Multi-class classification



$$\mu_m = \theta_{m0} + \theta_{m1}x_{i1} + \ldots \theta_{mp}x_{ip} \quad \textbf{where} \quad m = 1,2,\ldots M$$

$$Z_m = P(Y_i \mid X_i, \theta) = \frac{1}{1 + e^{-\mu_m}}$$

$$T_k = \beta_{0k} + \beta_{1k}z_1 + \ldots + +\beta_{1k}z_M$$

Softmax

$$\hat{Y}_k = \frac{e^{T_k}}{\sum_{k=1}^{3} e^{T_k}} \quad \textbf{where} \quad k = 1,2,3$$

e.g. $$P(Y_i = 2 \mid X_i, \theta) = \hat{Y}_2 = \frac{e^{T_2}}{e^{T_1} + e^{T_2} + e^{T_3}}$$

Merci


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP