Deep Learning Introduction

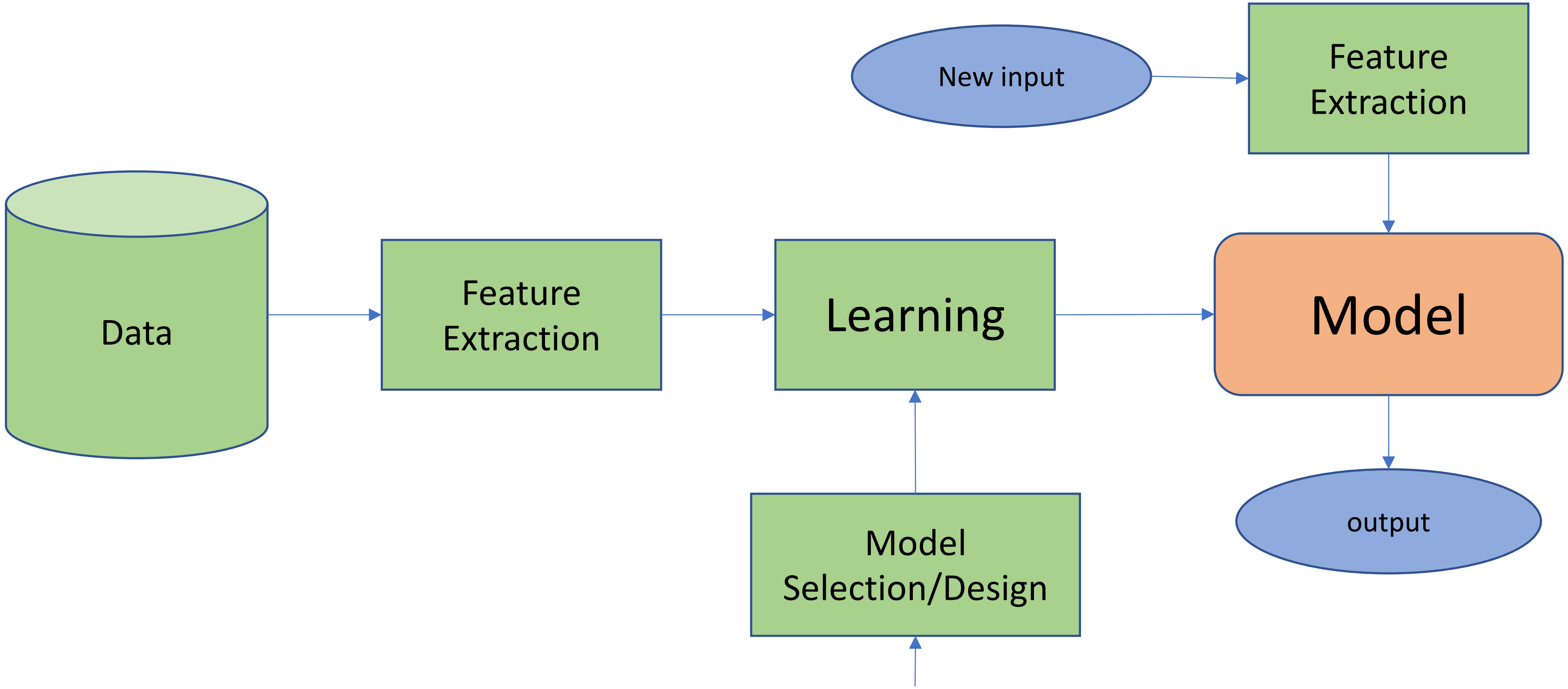
LÊ ANH CƯỜNG

Ton Duc Thang University

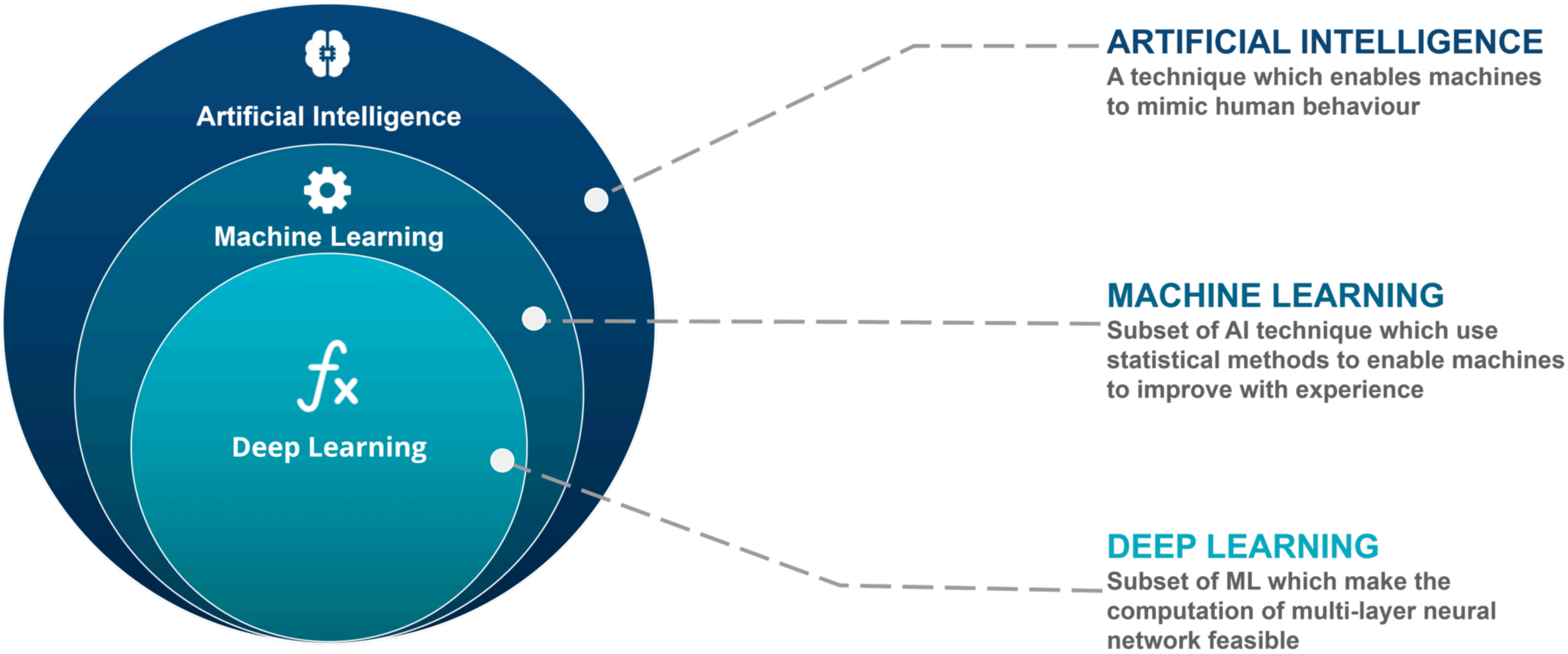
Outline

1. Deep Learning & Machine Learning
2. Review of Basic Neural Networks
3. Tensorflow vs Pytorch
4. Topics of this course

Tradictional Machine Learning Diagram

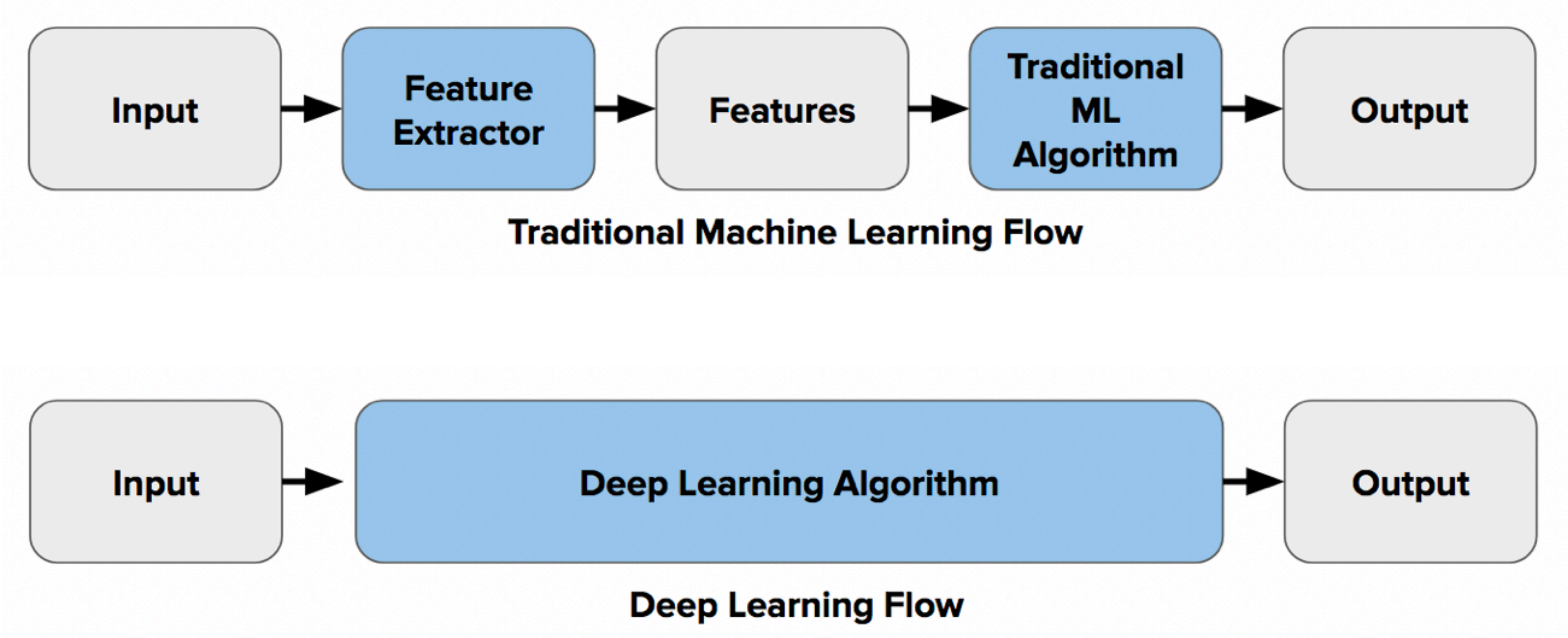


SVM, NB, DT, LR, NN, kNN,…

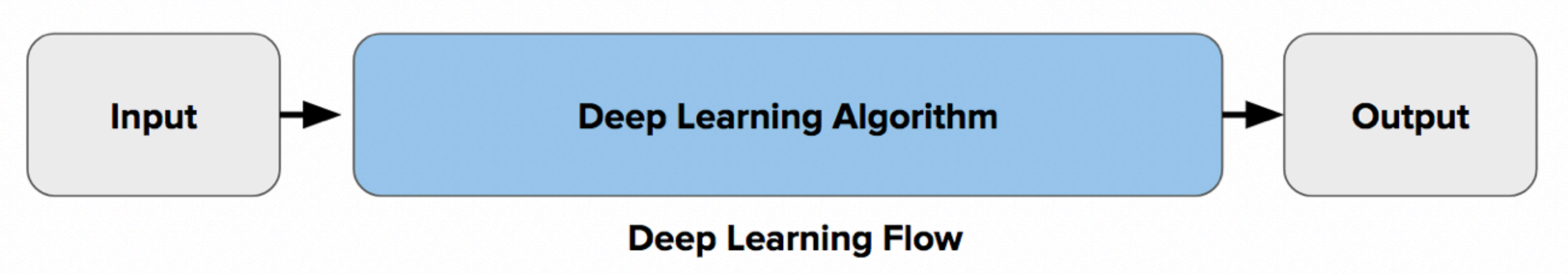


https://ospreydata.com/ai-ml-models-101-what-is-a-model/

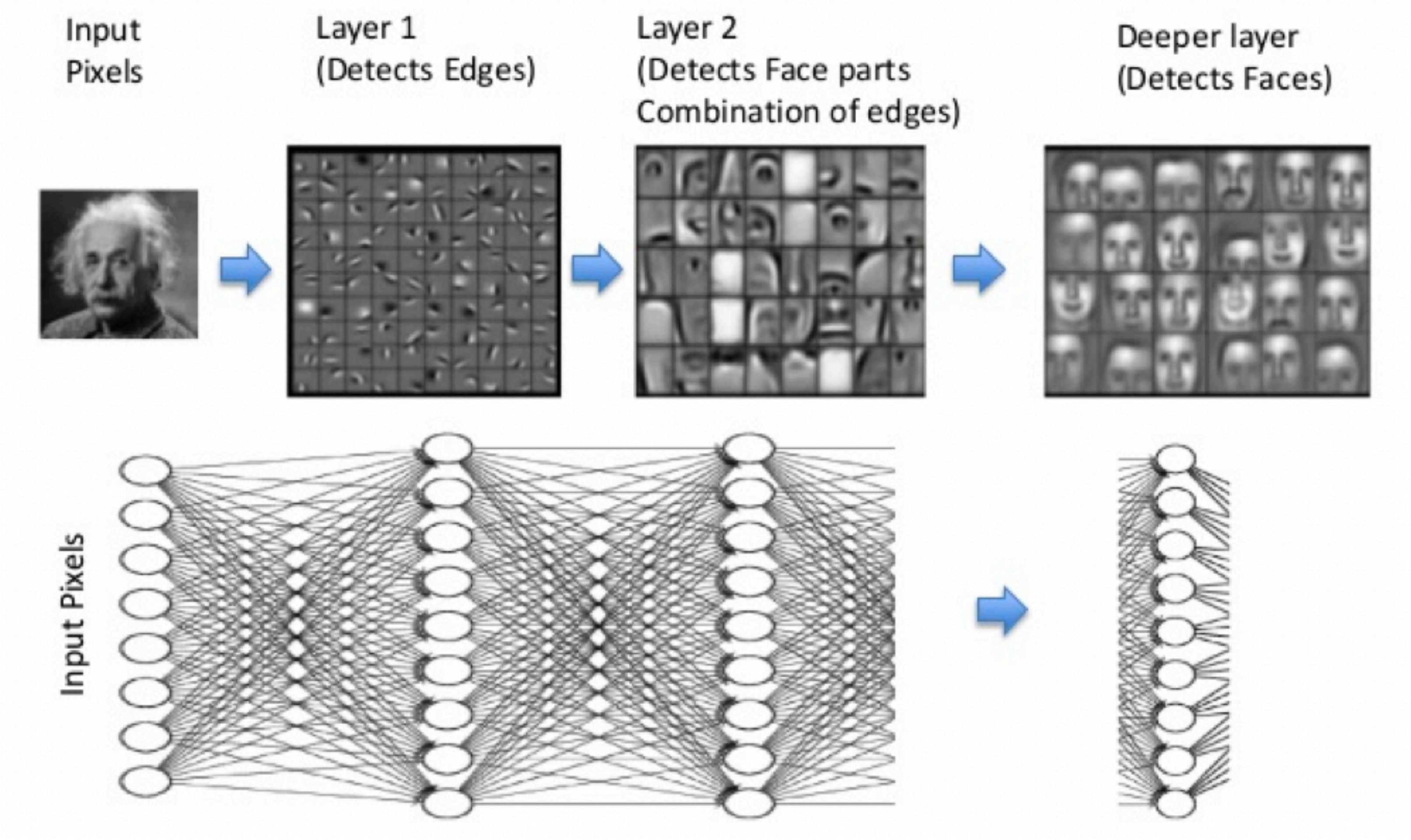
Conventional ML vs Deep Learning



End-to-End Model

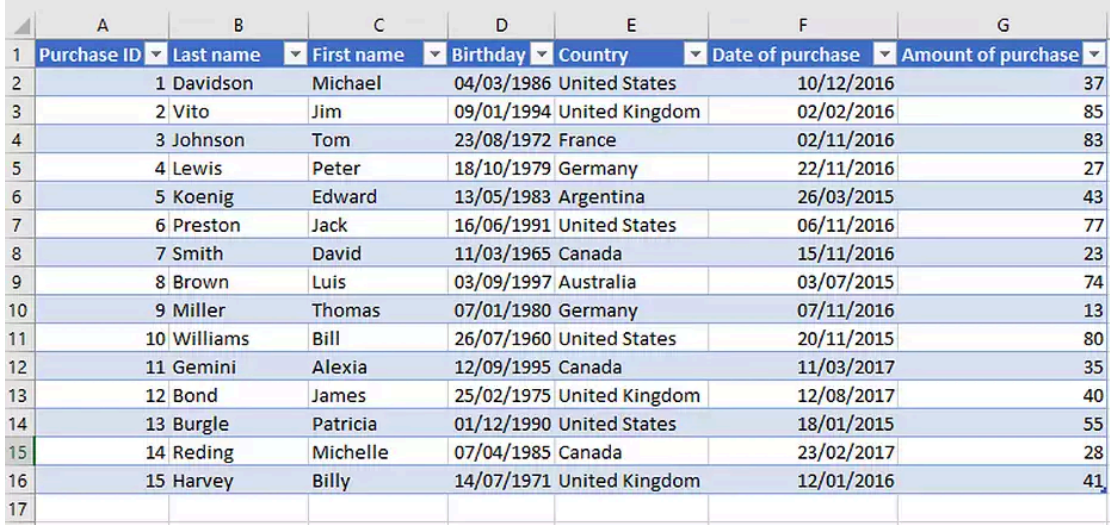


DL as Representation Learning

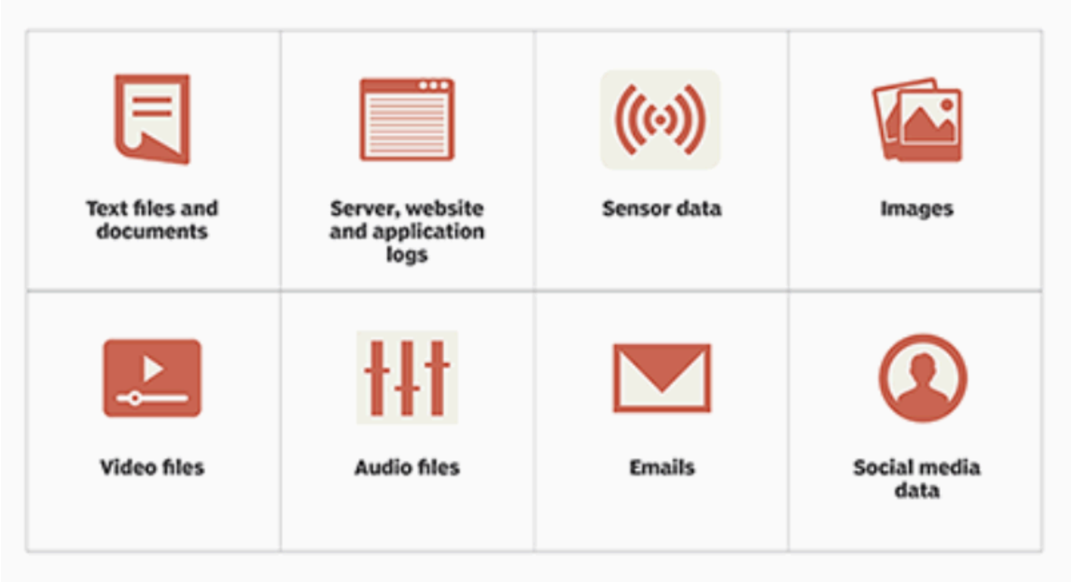


Deep Learning is useful for Unstructured Data

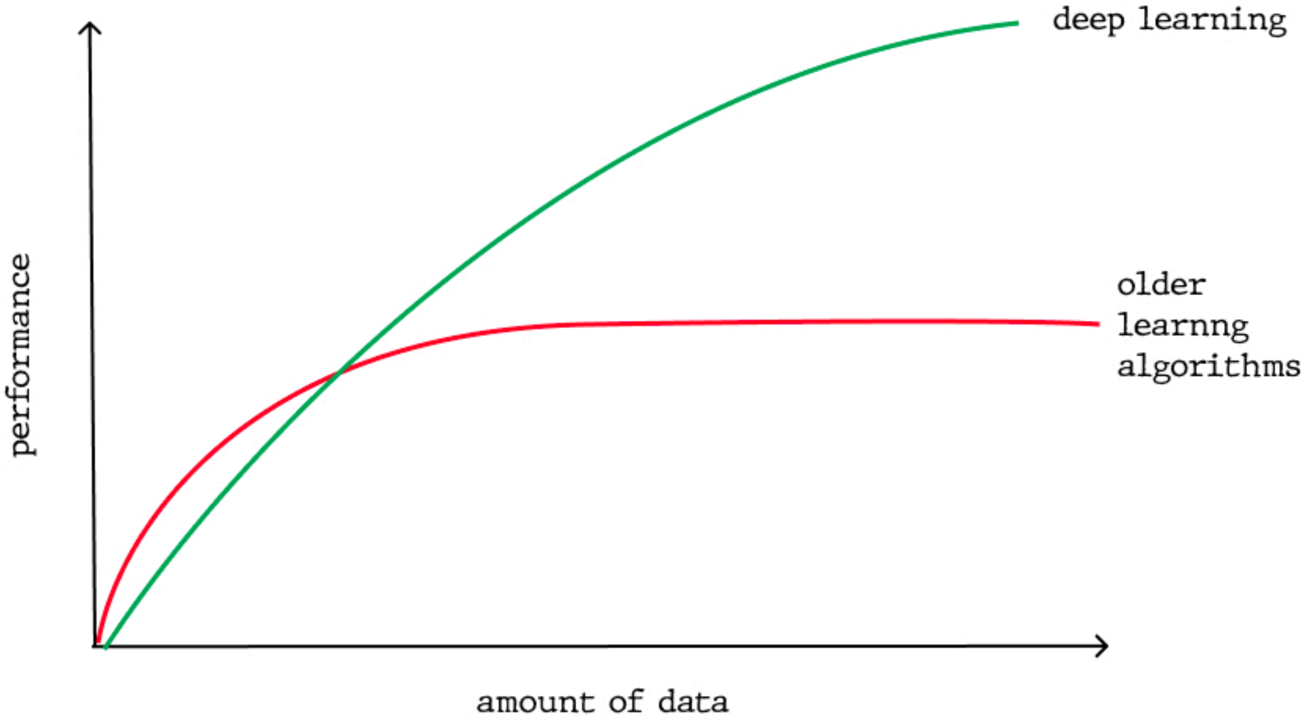
- Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts.



	A	B	C	D	E	F	G
1	Purchase ID	Last name	First name	Birthday	Country	Date of purchase	Amount of purchase
2	1	Davidson	Michael	04/03/1986	United States	10/12/2016	37
3	2	Vito	Jim	09/01/1994	United Kingdom	02/02/2016	85
4	3	Johnson	Tom	23/08/1972	France	02/11/2016	83
5	4	Lewis	Peter	18/10/1979	Germany	22/11/2016	27
6	5	Koenig	Edward	13/05/1983	Argentina	26/03/2015	43
7	6	Preston	Jack	16/06/1991	United States	06/11/2016	77
8	7	Smith	David	11/03/1965	Canada	15/11/2016	23
9	8	Brown	Luis	03/09/1997	Australia	03/07/2015	74
10	9	Miller	Thomas	07/01/1980	Germany	07/11/2016	13
11	10	Williams	Bill	26/07/1960	United States	20/11/2015	80
12	11	Gemini	Alexia	12/09/1995	Canada	11/03/2017	35
13	12	Bond	James	25/02/1975	United Kingdom	12/08/2017	40
14	13	Burgle	Patricia	01/12/1990	United States	18/01/2015	55
15	14	Reding	Michelle	07/04/1985	Canada	23/02/2017	28
16	15	Harvey	Billy	14/07/1971	United Kingdom	12/01/2016	41
17							



ML vs DL in Performance



Why Deep Learning?

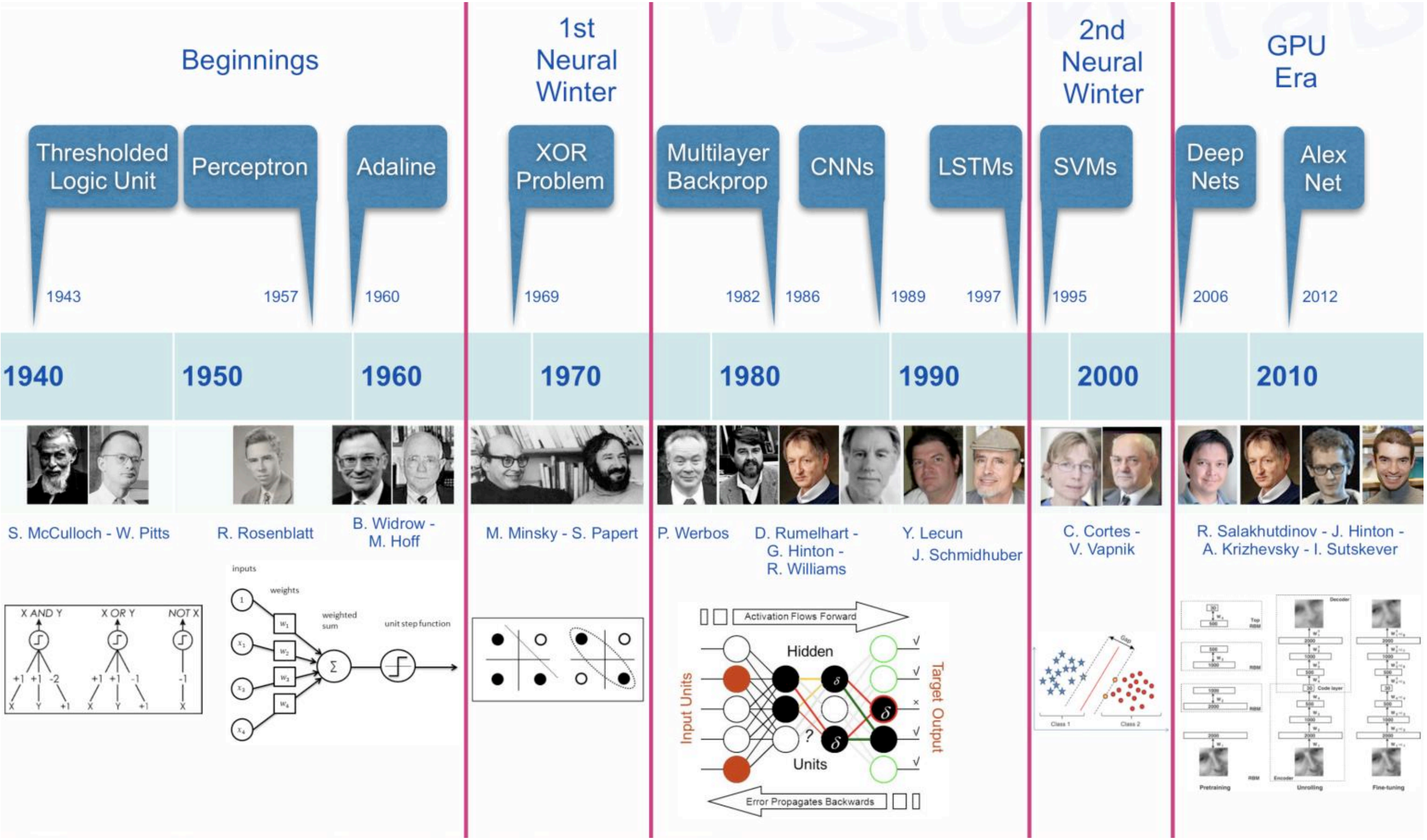
1. Feature Learning and Representation
2. Hierarchical Representation
3. Adaptability to Different Data Types
4. Scalability
5. End-to-End Learning
6. State-of-the-Art Performance
7. Transfer Learning

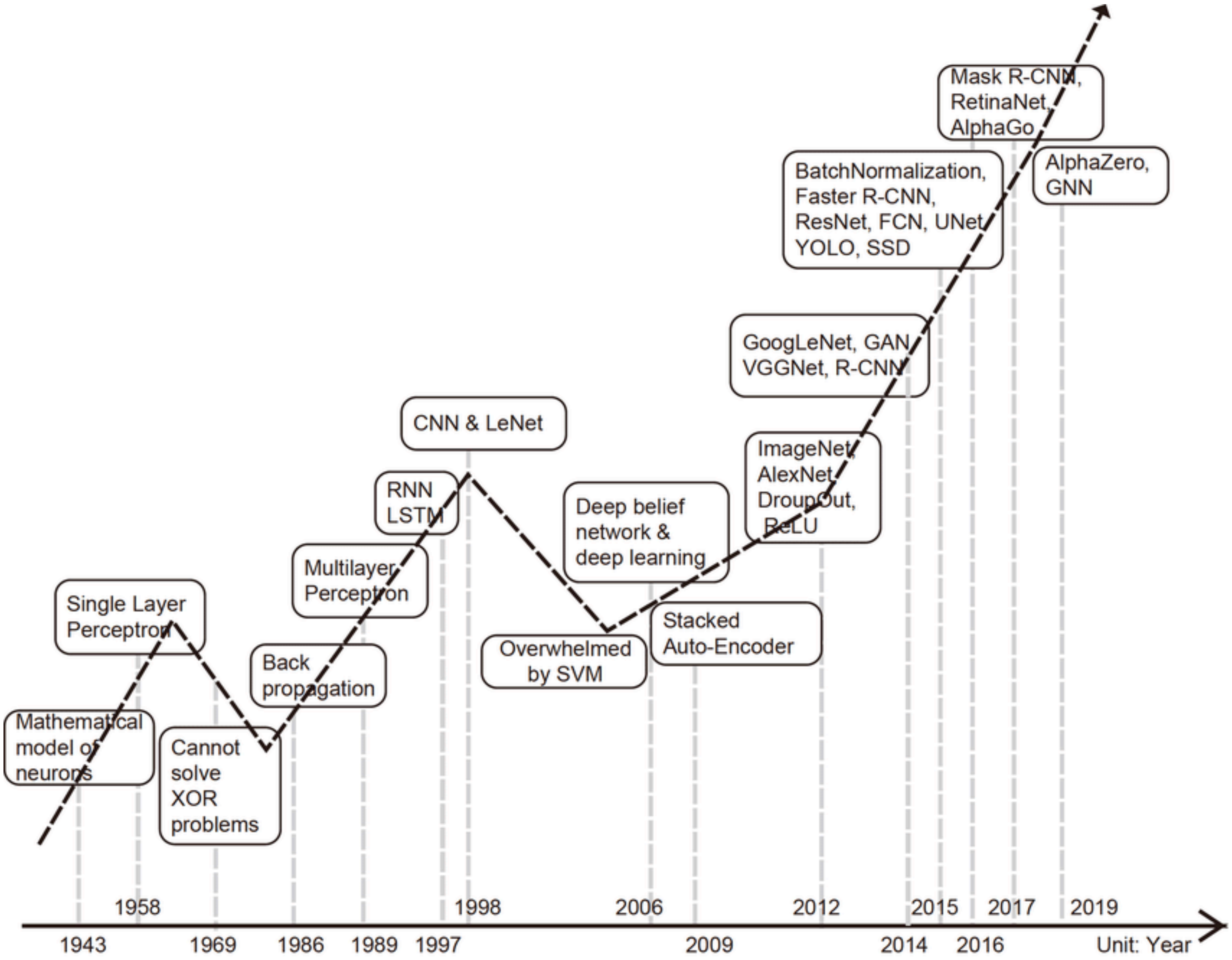
(reference: ChatGPT)

1. **Feature Learning and Representation:** Deep learning models automatically learn relevant features from the data, eliminating the need for manual feature engineering. This ability is particularly useful in tasks where the underlying patterns are complex and not easily captured by handcrafted features.
2. **Hierarchical Representation:** Deep neural networks can learn hierarchical representations of data. Each layer in a deep network extracts increasingly complex features, allowing the model to understand intricate relationships within the data.
3. **Adaptability to Different Data Types:** Deep learning models can be applied to a wide range of data types, including images, text, audio, and structured data. Convolutional Neural Networks (CNNs) are effective for image data, Recurrent Neural Networks (RNNs) for sequential data like text, and Transformer models for attention-based tasks.

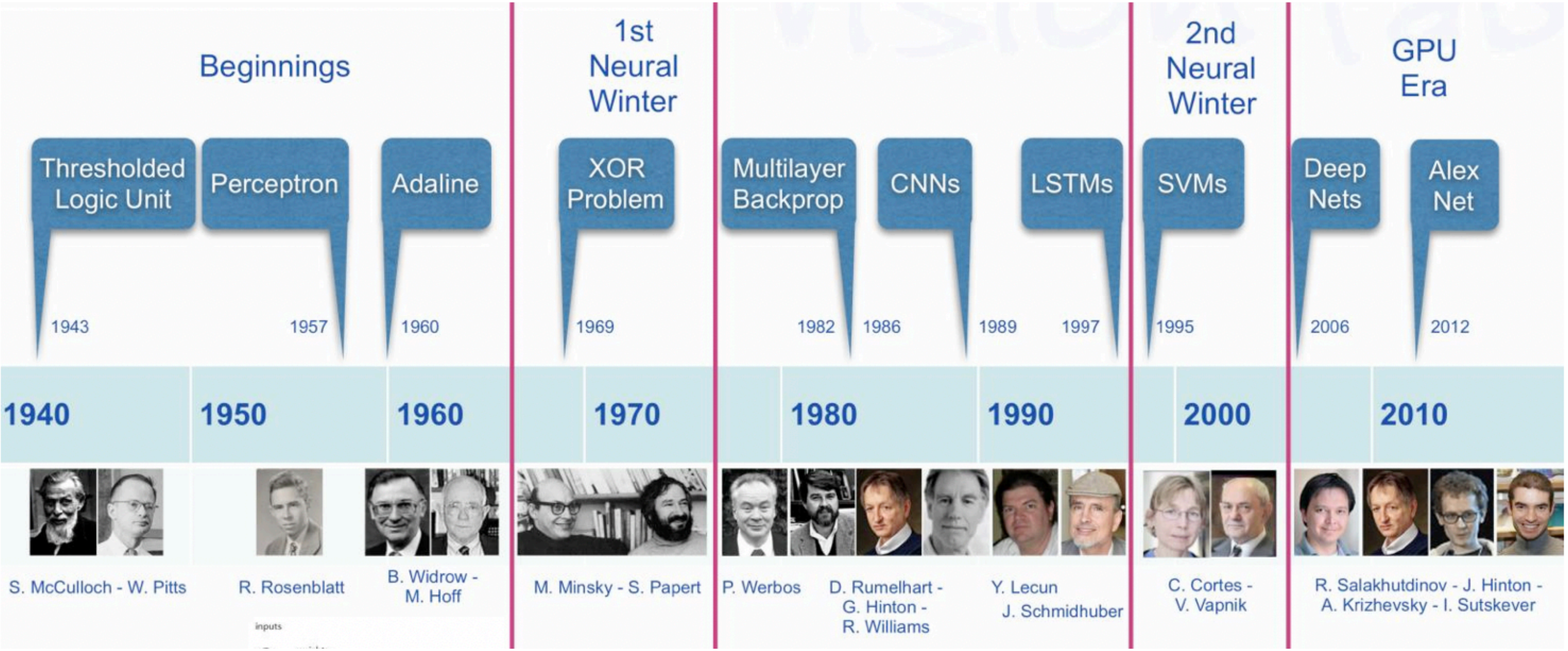
5. **End-to-End Learning:** Deep learning models can perform end-to-end learning, meaning they can learn to map raw input data directly to the desired output without relying on intermediate representations. This is particularly advantageous in tasks where a clear mapping between input and output exists.
6. **State-of-the-Art Performance:** Deep learning has achieved state-of-the-art results in various domains, including computer vision, natural language processing, and speech recognition. This makes it a go-to choice for many applications where high accuracy is crucial.
7. **Transfer Learning:** Pre-trained deep learning models can be fine-tuned for specific tasks, leveraging knowledge gained from large datasets. This is especially valuable when labeled data for a specific task is limited.

History of ML




Mathematical model of neurons
Single Layer Perceptron
Cannot solve XOR problems
Back propagation
Multilayer Perceptron
RNN LSTM
CNN & LeNet
Overwhelmed by SVM
Deep belief network & deep learning
Stacked Auto-Encoder
ImageNet, AlexNet, DroupOut, ReLU
GoogLeNet, GAN, VGGNet, R-CNN
BatchNormalization, Faster R-CNN, ResNet, FCN, UNet, YOLO, SSD
Mask R-CNN, RetinaNet, AlphaGo
AlphaZero, GNN
1943
1958
1969
1986
1989
1997
1998
2006
2009
2012
2014
2015
2016
2017
2019
Unit: Year

History of Deep Learning



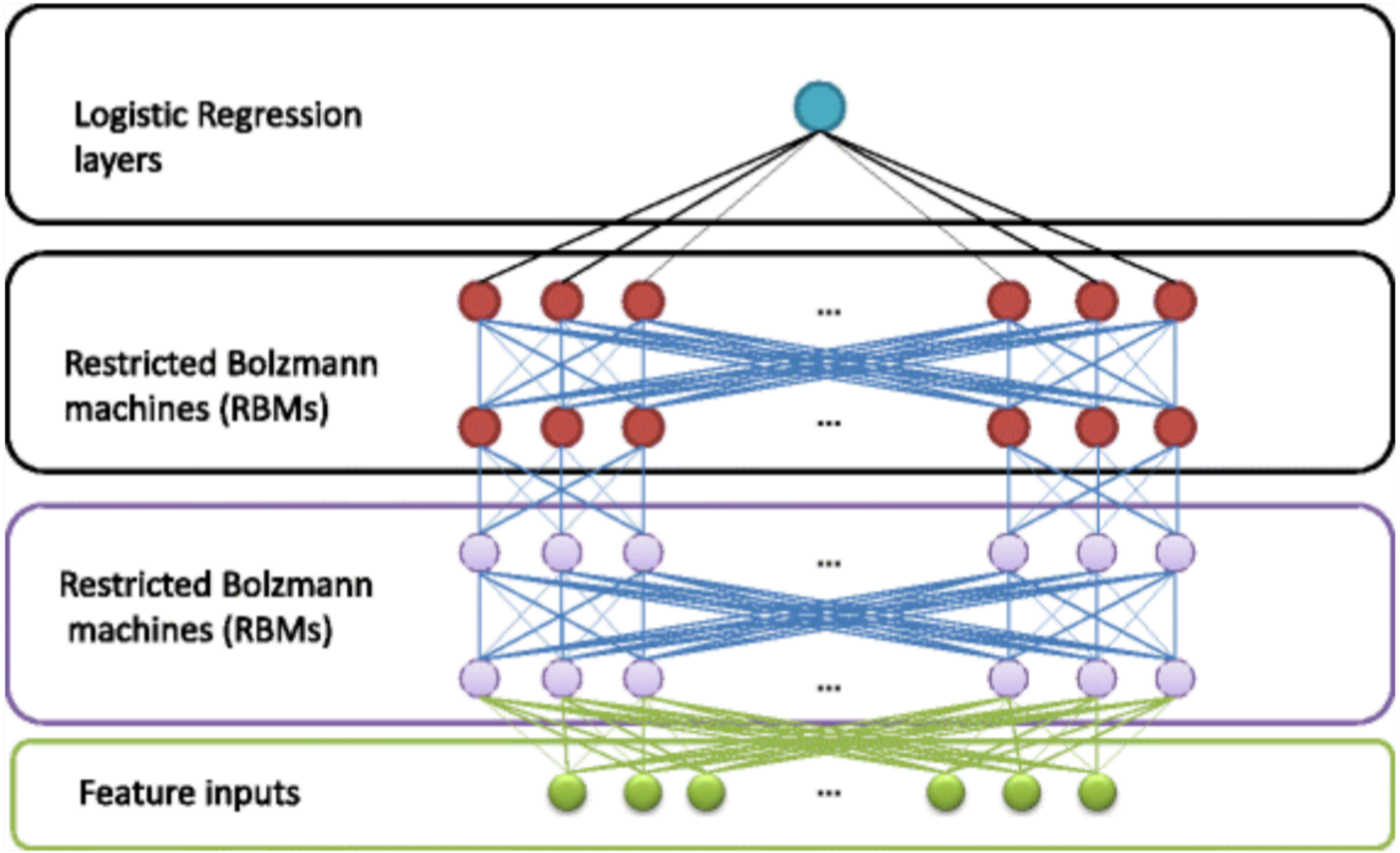
When the term Deep Learning?

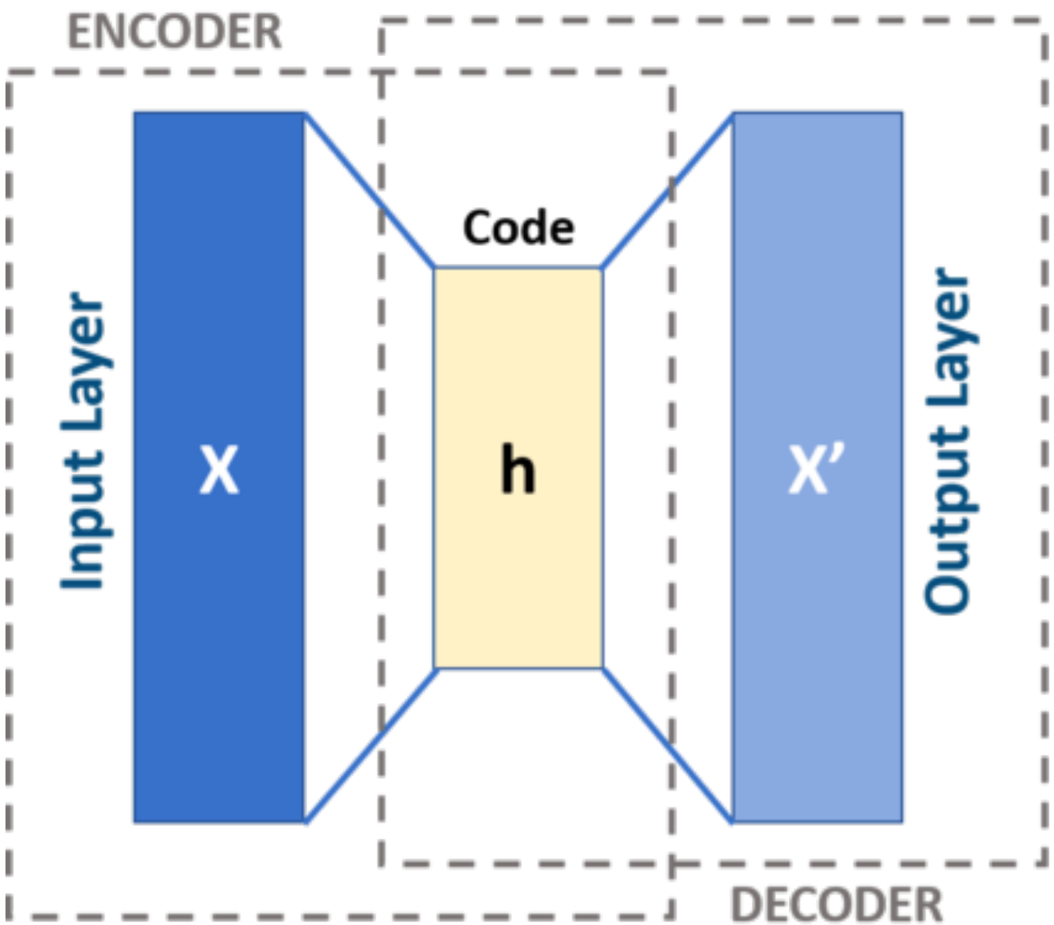
Geoffrey Hinton is a pioneer in the field of artificial neural networks and co-published the first paper on the backpropagation algorithm for training multilayer perceptron networks.

He may have started the introduction of the phrasing "*deep*" to describe the development of large artificial neural networks.

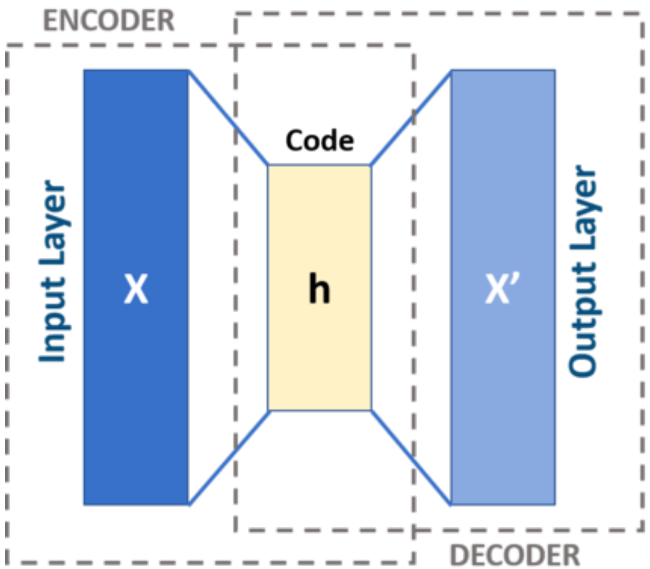
He co-authored a paper in 2006 titled "A Fast Learning Algorithm for Deep Belief Nets" in which they describe an approach to training "deep" (as in a many layered network) of restricted Boltzmann machines.

Deep Belief Network





An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner.[1] The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction



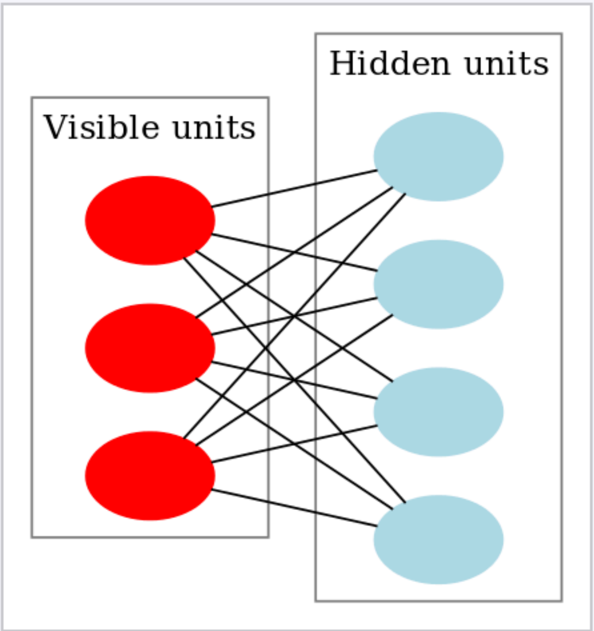


Diagram of a restricted Boltzmann machine with three visible units and four hidden units (no bias units).

An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner.[1] The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction

A **restricted Boltzmann machine** (**RBM**) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs.

Recognition of DL

- Deep: i.e. many layers/levels of data representation
- Big Data (large enough): for learning good features
- High Performance Computing: for doing with complex networks and big data

Topics in this course

1. Review of Neural Networks: FFNN and RNN
2. Learning: optimization methods
3. Regularization methods
4. Convolutional Neural Networks
5. Long Short Term Memory networks
6. Deep Belief Network
7. Stacked Auto-Encoder network
8. Transformers and various its variants
9. Generative Pretrain Transformer (for ChatGPT)
10. Generative Adversarial Network
11. Reinforcement Learning and Deep Reinforcement Learning

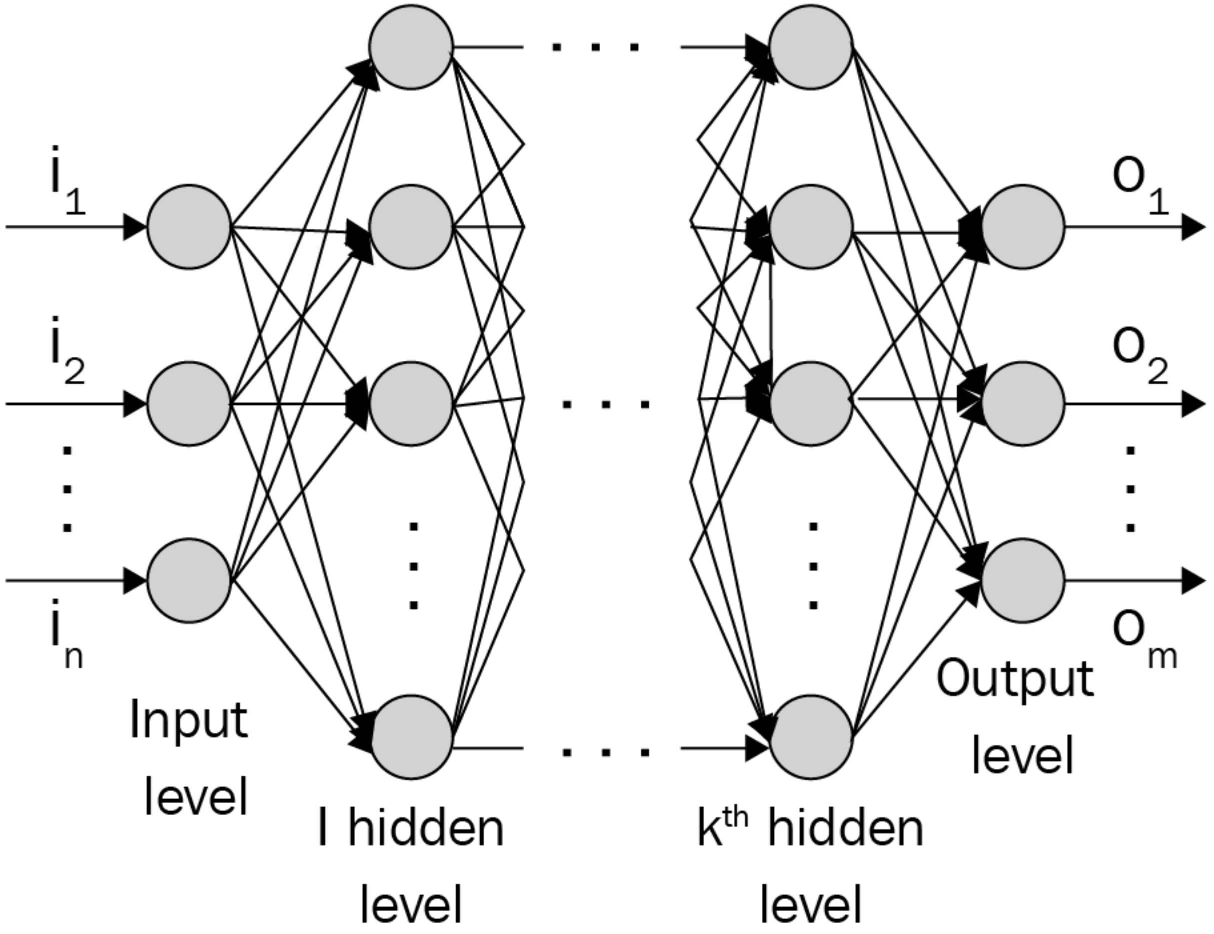
Course assessment

- Middle exam (20%)
  - Project and presentation
- Final exam (50%)
  - Project and presentation
- Progress exercises (30%)

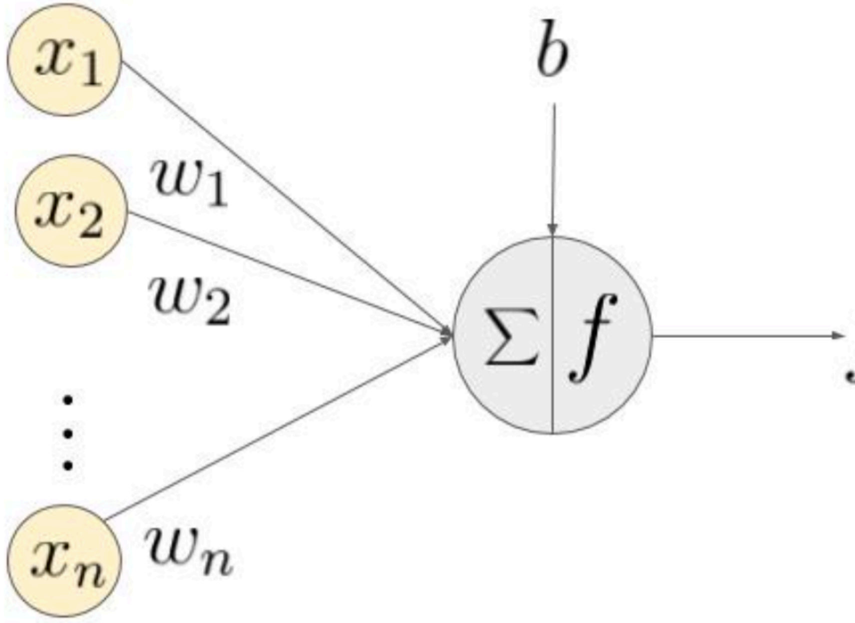
Outline

1. Review of Basic Neural Networks
    - Feed Forward Neural Network
    - Gradient Descent Algorithm
    - Backpropagation Algorithm
    - Recurrent Neural Network
    - Backpropagation Through Time
2. Keras of Tensorflow vs Pytorch
    - Keras and Tensorflow
    - Pytorch

Feed Forward Neural Network



A Neural

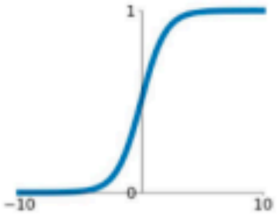


$$f\left(b+\sum_{i=1}^{n} x_i w_i\right)$$

Activation functions

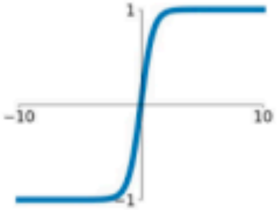
Sigmoid

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



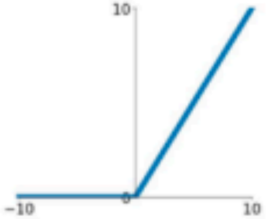
tanh

$\tanh(x)$



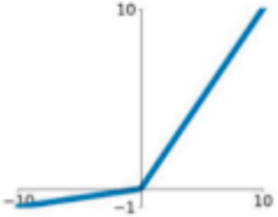
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

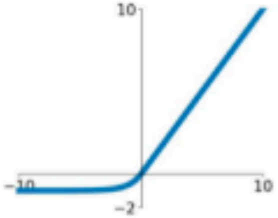


Maxout

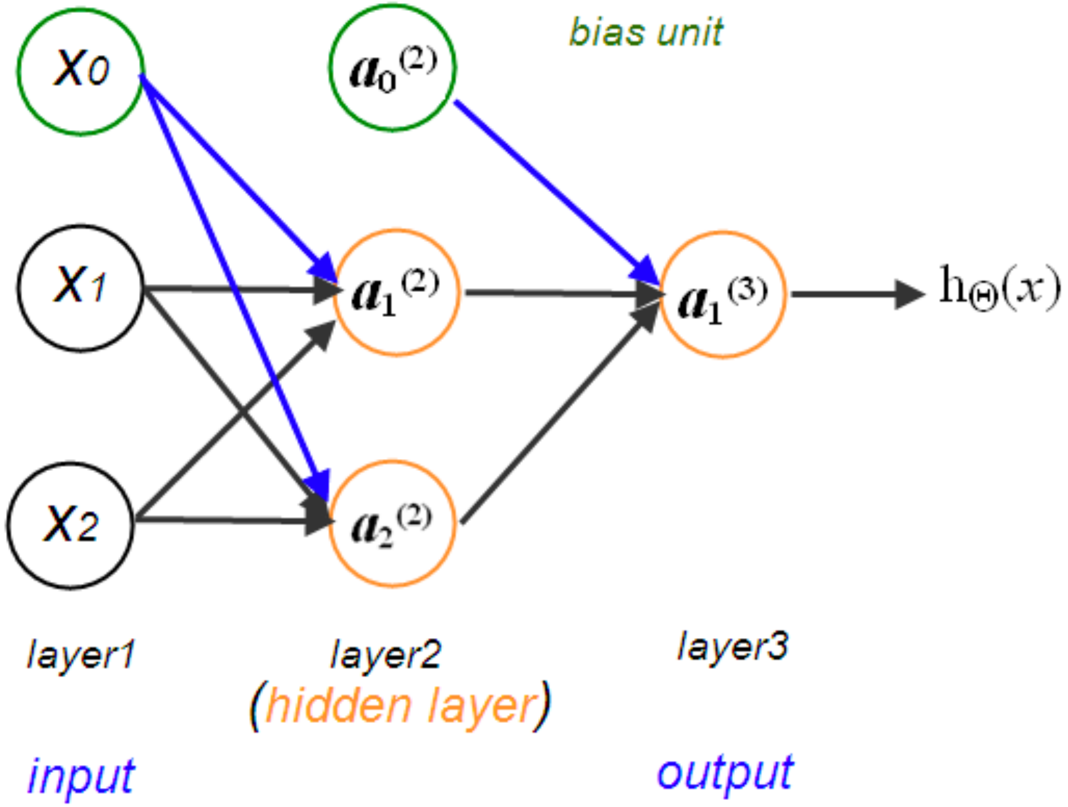
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

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

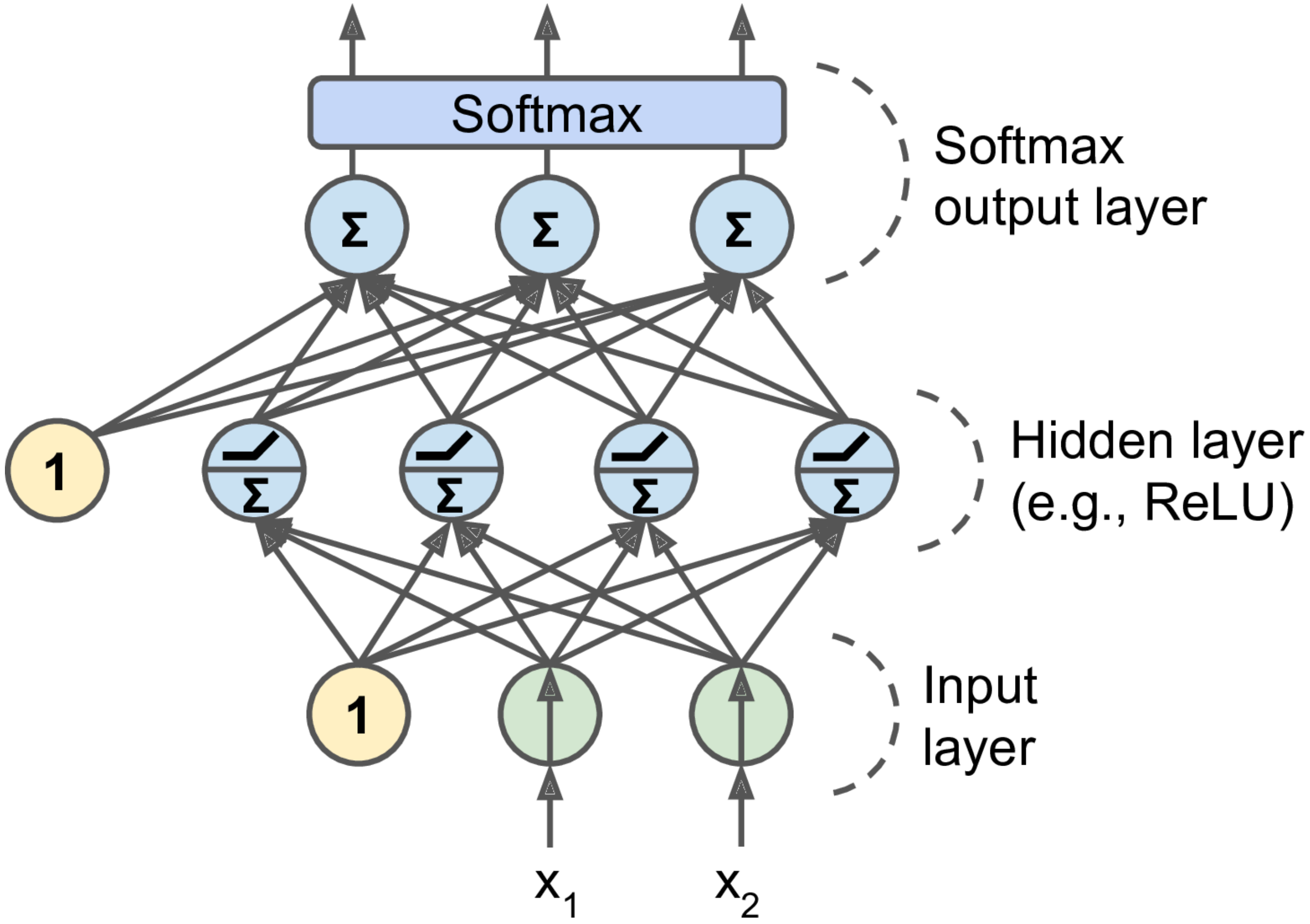


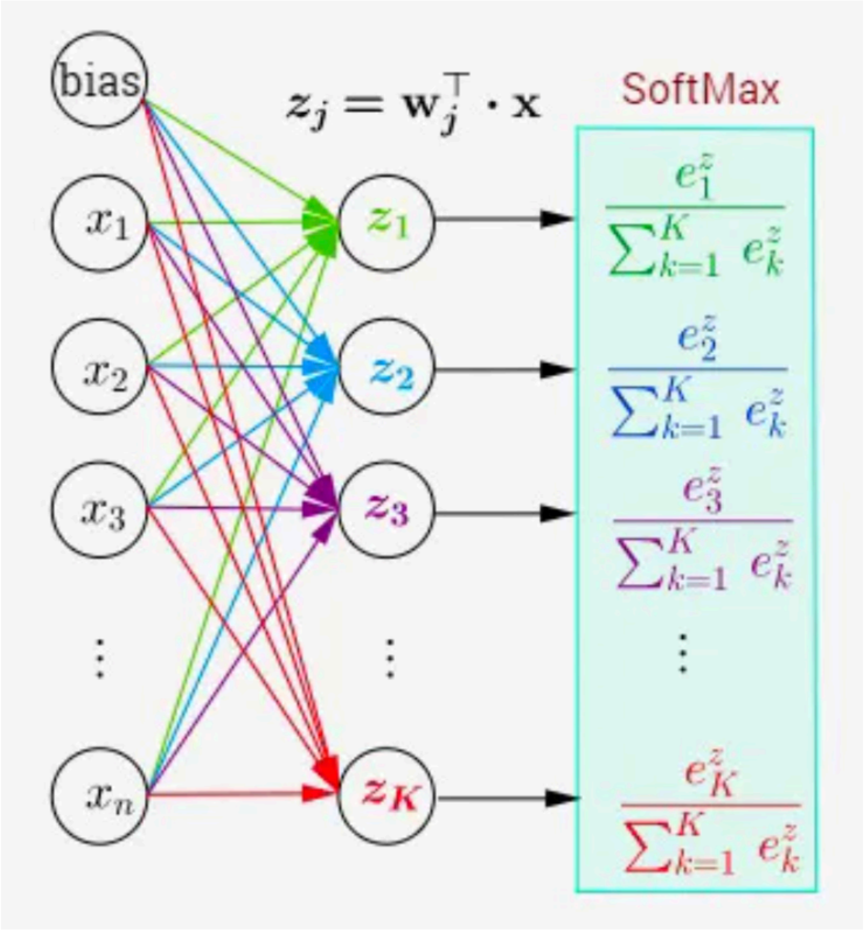
Example



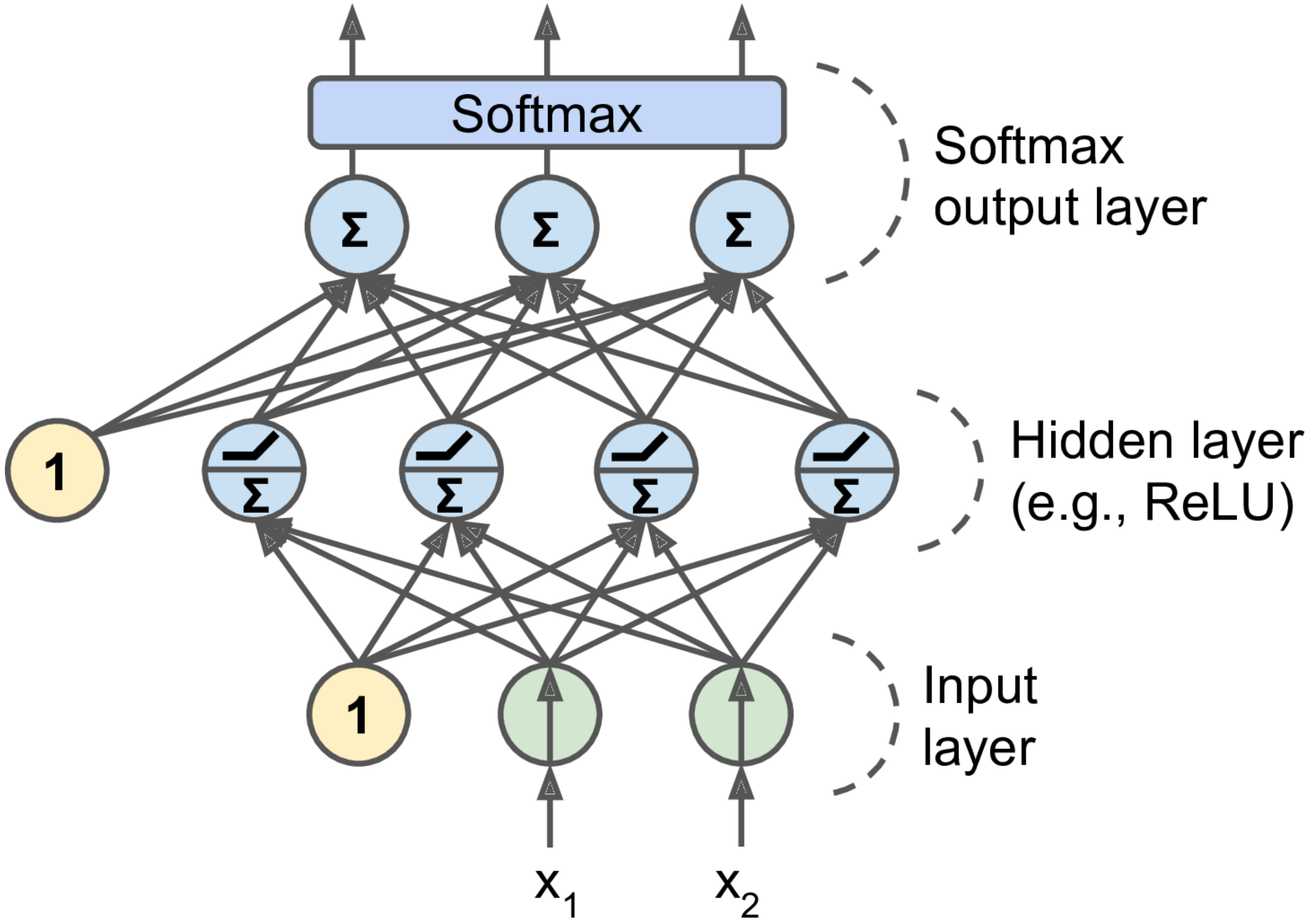
Fill in values and computing?

FFNN for classification





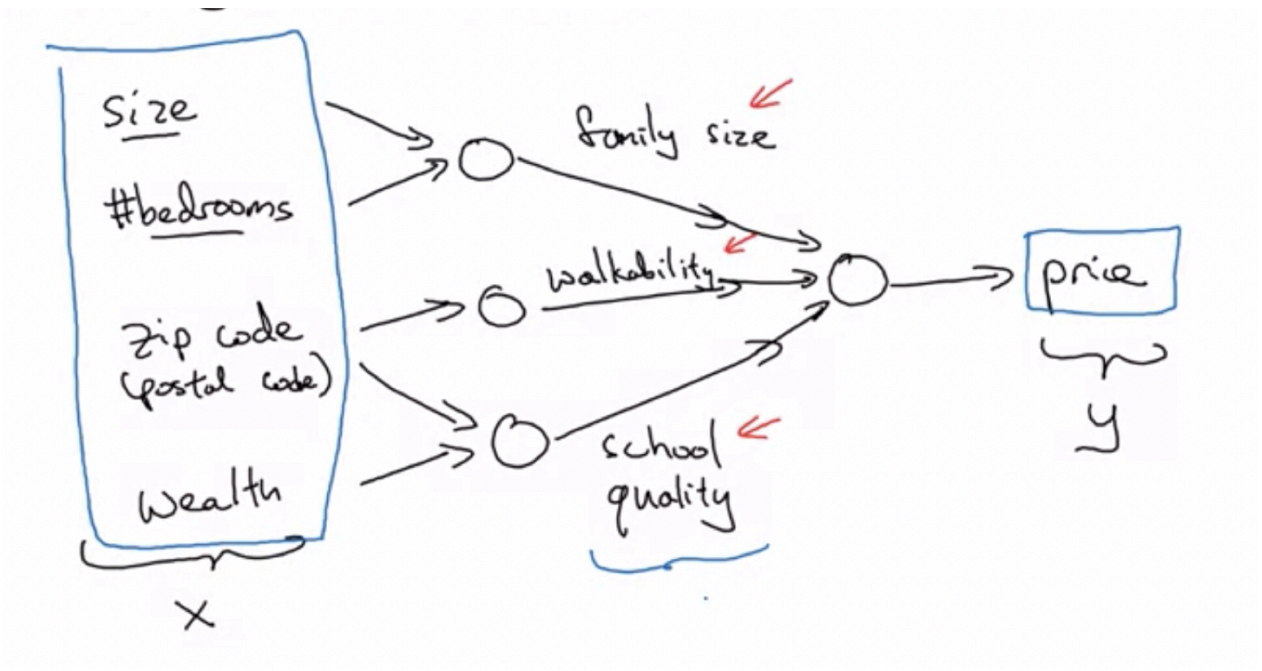
FFNN for classification



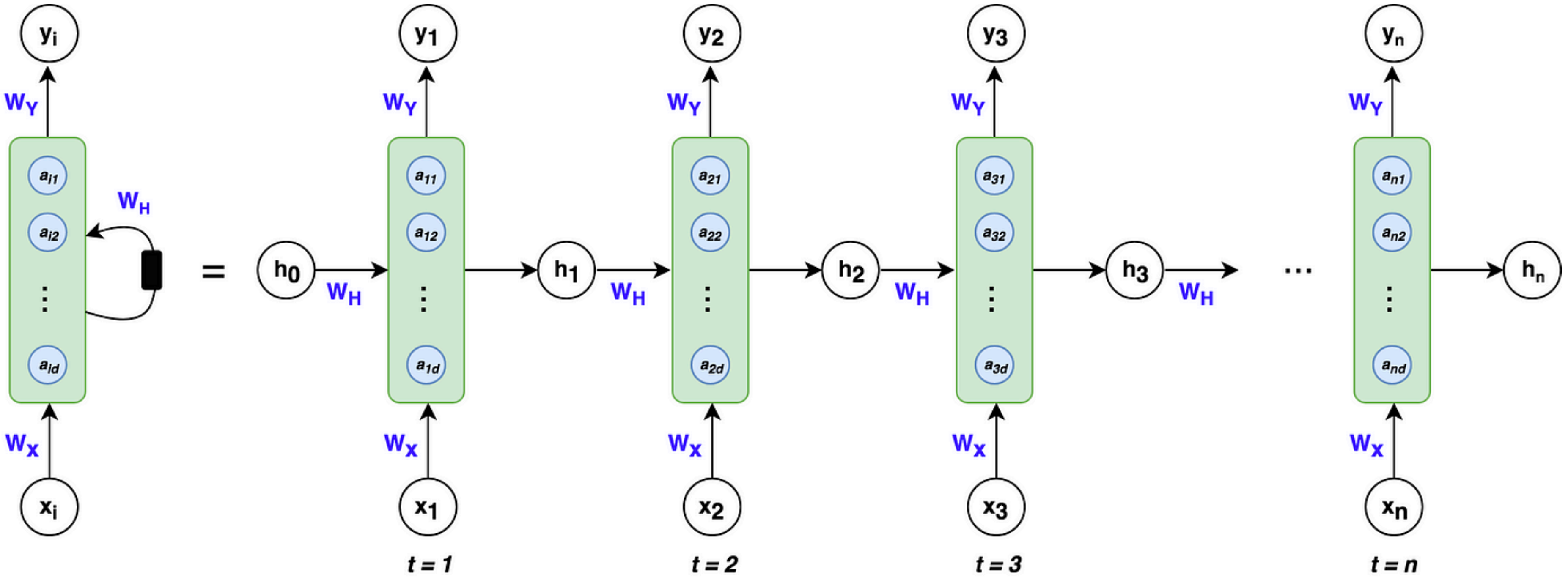
- Example?

FFNN for Regression
house pricing problem?

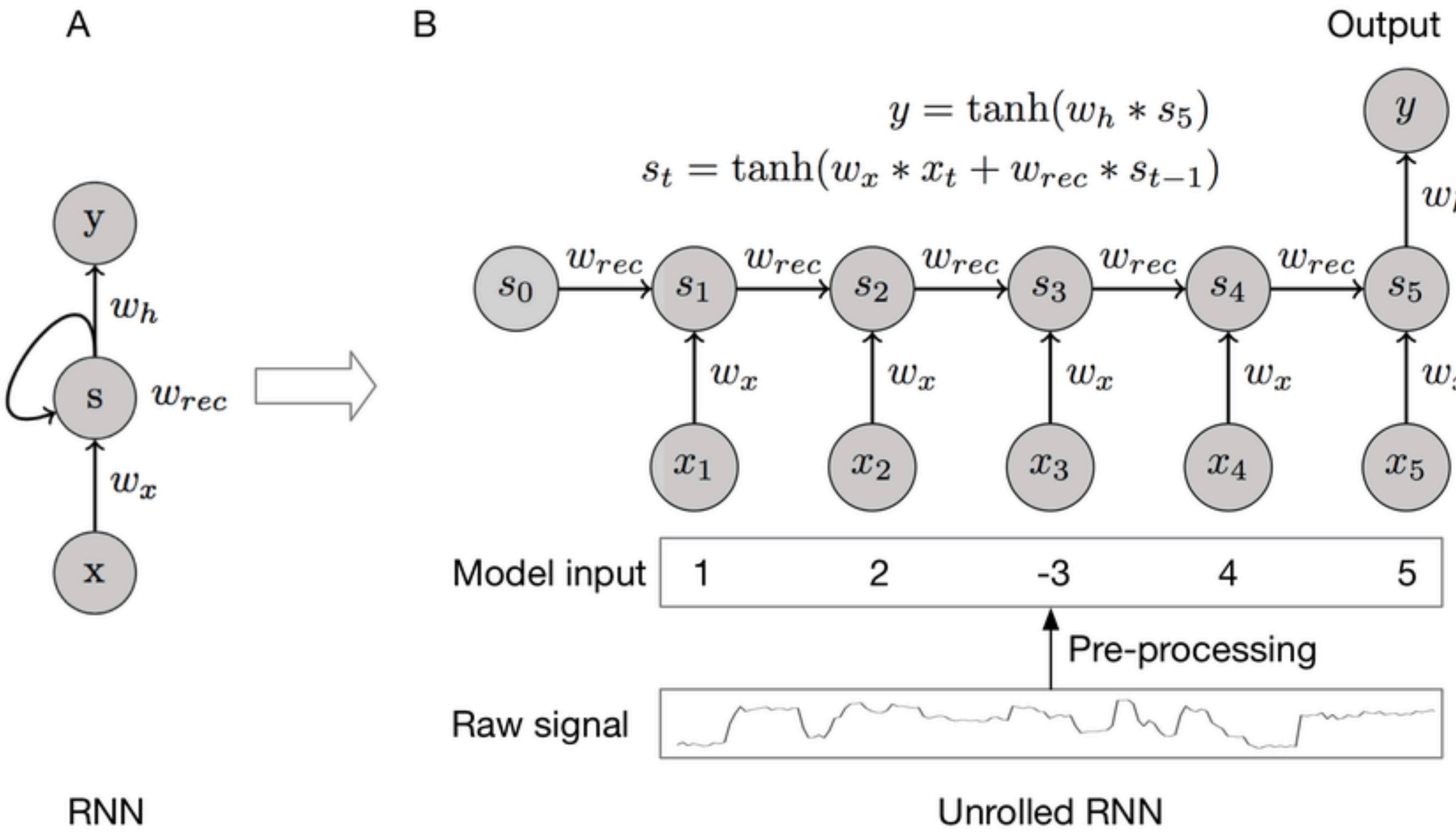
- Design new structure?
- Example?



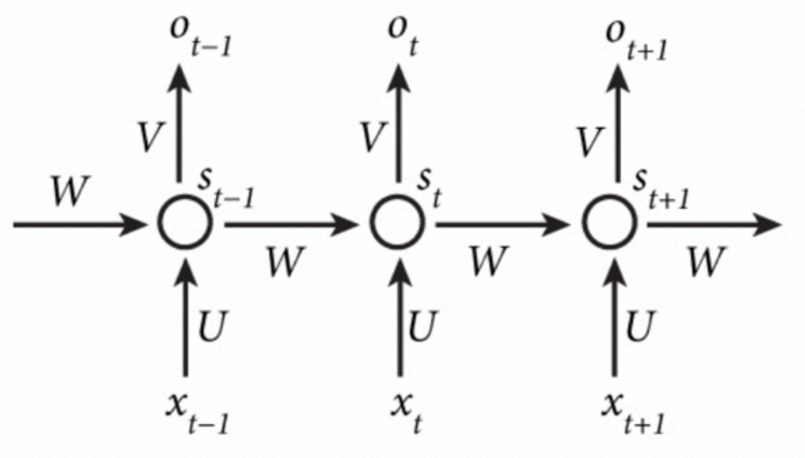
Recurrent Neural Network (RNN)



RNN


A
B
Output
$y = \tanh(w_h * s_5)$
$s_t = \tanh(w_x * x_t + w_{rec} * s_{t-1})$
y
s
x
w_h
w_{rec}
w_x
y
s_0
s_1
s_2
s_3
s_4
s_5
x_1
x_2
x_3
x_4
x_5
Model input
1
2
-3
4
5
Pre-processing
Raw signal
RNN
Unrolled RNN

RNN example



$$s_t = f(Ux_t + Ws_{t-1}).$$

$$o_t = \mathrm{softmax}(Vs_t).$$

$$a_1 = \begin{pmatrix} 0.1 & 0.5 & 0.1 \\ 0.5 & 0.9 & 0.3 \\ 0.3 & 0.2 & 0.1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.6 & 0.8 & 0.4 & 0.8 \\ 0.2 & 0.2 & 0.8 & 0.7 \\ 0.9 & 0.8 & 0.1 & 0.2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.2 \\ 0.9 \end{pmatrix}$$

$$h_1 = tanh(\begin{pmatrix} 0.6 \\ 0.2 \\ 0.9 \end{pmatrix}) = \begin{pmatrix} 0.54 \\ 0.20 \\ 0.72 \end{pmatrix}$$

$$y_1 = softmax(\begin{pmatrix} 0.9 & 0.8 & 0.3 \\ 0.2 & 0.3 & 0.4 \\ 0.6 & 0.9 & 0.1 \\ 0.5 & 0.0 & 0.3 \end{pmatrix} \begin{pmatrix} 0.54 \\ 0.20 \\ 0.72 \end{pmatrix}) = \begin{pmatrix} 0.32 \\ 0.21 \\ 0.24 \\ 0.22 \end{pmatrix}$$