Machine Learning for Economists

Class 14: Artificial Neural Network (ANN)

葛雷

中国人民大学 - 数量经济

2025 年 5 月 8 日

中國人民大學
RENMIN UNIVERSITY OF CHINA

What is ANN

From OLS to ANN

Core: Activation Functions

Perception to ANN

Output Types

Complex ANN models

What is ANN

From OLS to ANN

Core: Activation Functions

Perception to ANN

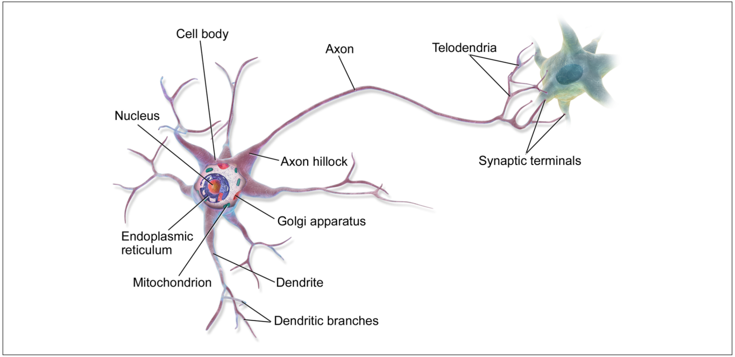
Output Types

Complex ANN models

ANN is popular algorithm in every industry

- Natural Language Processing (ChatGPT, Bert, LLaMA, GLM, Deepseek)
- Code generation (Cursor, Copilot, Marscode, codewhisperer, Webpage generation, UI design)
- Visual recognition and generation (YOLO, VIT, Diffusion)
- Stock Trading, Housing Valuation, Risk Management and wealth management

Emulation of the human neuron



What is ANN

From OLS to ANN

Core: Activation Functions

Perception to ANN

Output Types

Complex ANN models

From OLS to ANN

$$Y = \beta_1 X + \beta_0 + \mu \tag{1}$$

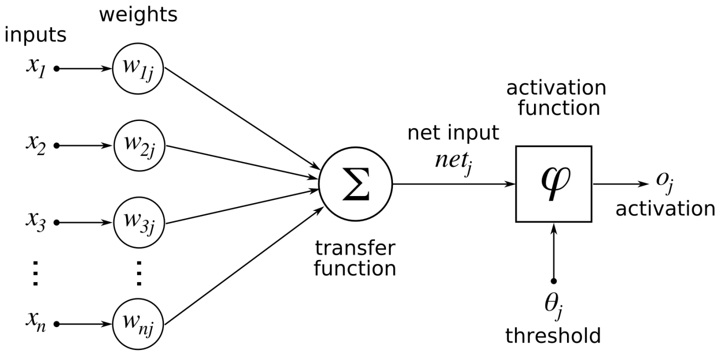
- OLS is a simple neural network
- With only input and output layer
- Without activation function

From Logit to ANN

$$Y = \frac{1}{1 + exp(-(\beta_1 X + \beta_0 + \mu))} \tag{2}$$

- OLS is a simple Neural network
- With only input and output layer
- With **Sigmoid** activation function

Perception: threshold logic unit (TLU)



Perception: threshold logic unit (TLU)

- Perception = Linear Transformation + Activation Function
- It is single layer of the ANN

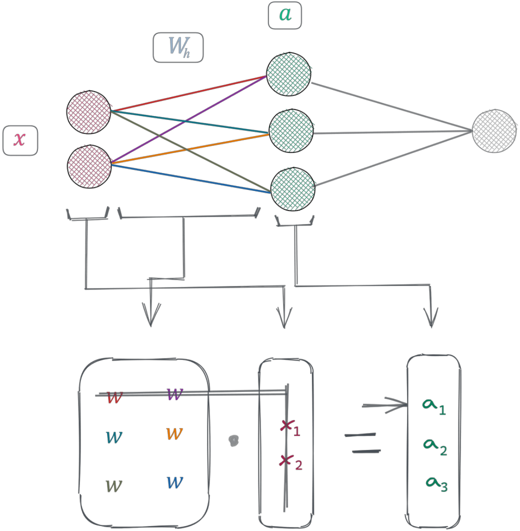
Do we have other types of activation function? Yes, please wait

Linear Transformation

$$y = xA^T + b.$$

- Please check pytorch torch.nn.linear
- OLS is one output linear transformation
- linear transformation also can have multiple outputs, but how to do that?

Linear Transformation in Matrix



What is ANN

From OLS to ANN

Core: Activation Functions

Perception to ANN

Output Types

Complex ANN models

Activation Function (!!!Source of Non-linearity!!!)

- An activation function is a function that determines the output of a neuron in an artificial neural network, based on its inputs and weights.

- It is a **non-linear transformation** that can help the network learn complex patterns and features

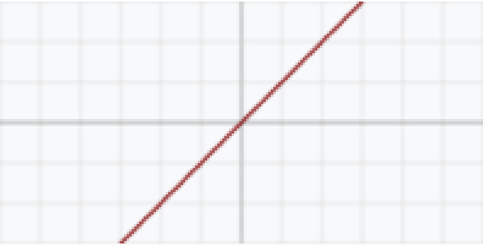
Ideal Activation function properties(Not in practice)

- Nonlinear: A nonlinear activation function allows the network to approximate any function, as proven by the universal approximation theorem
- Continuous and differentiable: A continuous and differentiable activation function enables the use of gradient-based optimization methods, such as backpropagation, to update the weights of the network
- Bounded: A bounded activation function limits the range of the output values, which can prevent issues such as exploding or vanishing gradients
- Monotonic: A monotonic activation function preserves the order of the inputs, which can facilitate the learning

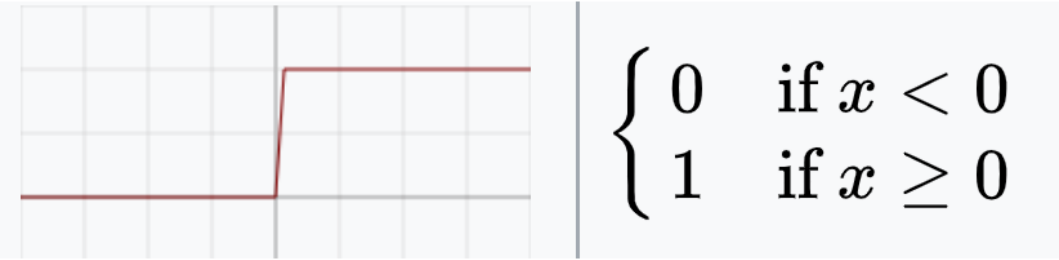
Types of activation functions

- identity, step
- sigmoid, tanh
- ReLU,Leaky ReLU, GELU, SILU (ChatGPT and Bert use GELU)
- All create the **non-linearity** in the neural network

Activation 1: Identity function (no activation)

	x

Activation 2: Step function

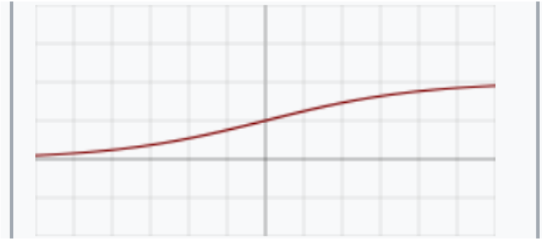


$$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

Comments: identity and step activation fcn

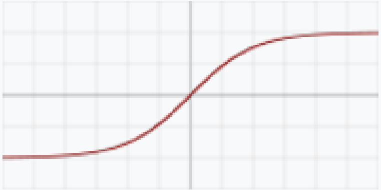
- simple
- lack non-linearity
- Identity activation used by ols, word embedding (What is Word Embedding?)

Activation 3: Sigmoid, or soft step



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

Activation 4: tanh, or Hyperbolic tangent



$$\tanh(x) \doteq \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Comments on sigmoid and tanh

sigmoid and tanh similar, but tanh has larger gradient

$$tanh(x) = 2sigmoid(2x) - 1$$

Comments on sigmoid and tanh

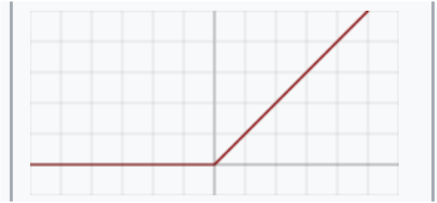
Advantages:

- 1) smooth gradient
- 2) output [0,1] or [-1,1]
- 3) sigmoid can put into the last layer to output probability of classification

Disadvantages:

- 1) vanishing gradient when x is large
- 2) output [0,1] or [-1,1]
- 3) computational expensive

Activation 5:Rectified linear unit (ReLU)



$$(x)^+ \doteq \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$
$$= \max(0, x) = x\mathbf{1}_{x>0}$$

Comments on ReLU

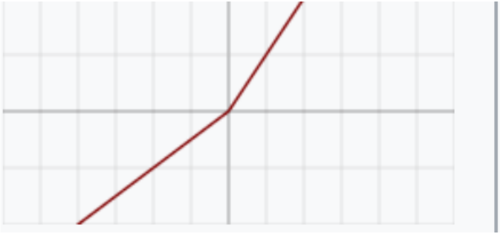
Advantages:

- Mitigates Vanishing Gradient Problem for sigmoid
- Non-linearity
- Computational Efficiency
- Sparse Activation: zeroing out negative values, faster

Disadvantages:

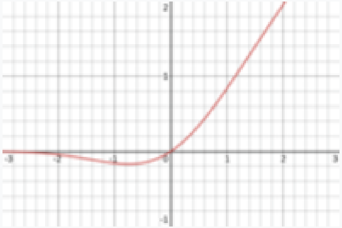
- Sparse Activation: dying ReLU problem
- exploding gradient when x is large

Activation 6: Leaky rectified linear unit (Leaky ReLU)



$$\begin{cases} 0.01x & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

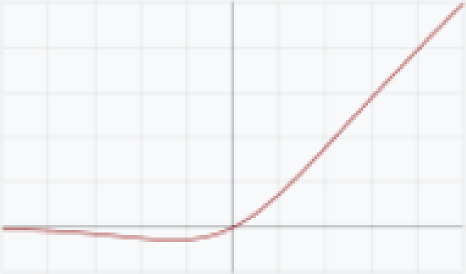
Activation 7: Gaussian Error Linear Unit (GELU)



$$\frac{1}{2}x\left(1+\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

$$= x\Phi(x)$$

Activation 8: Sigmoid linear unit (SiLU or Swish)



$$\frac{x}{1+e^{-x}}$$

Comments on GELU, SiLU and Leaky ReLU

Advantages:

- Enhanced edition of ReLU
- Mitigates Dying ReLU problem of constant zero outputs
- smooth and differentiable for GELU and SiLU
- GELU used by both ChatGPT and Bert

One Catch: computational expensive than ReLU, but still popular among AI models

Questions: Activation fcn vs link fcn

- what is the relation btw activation fcn and link fcn in econometrics class?
- paper pencil to write down the logit regression
- both proposes: create non-linearity

What is ANN

From OLS to ANN

Core: Activation Functions

Perception to ANN

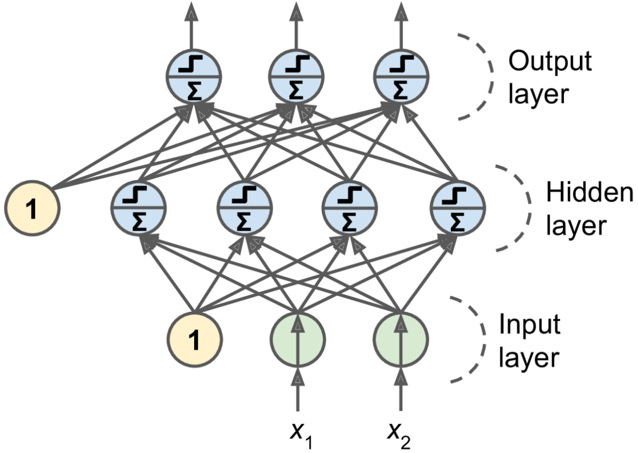
Output Types

Complex ANN models

Perception to ANN

- Perception is the unit of the ANN (Linear Transformation + Activation)
- Perception + Perception + ... + Perception => ANN

Architecture of a Multilayer Perceptron



function of ANN:

- $\hat{y} = F(x;\theta) = f^L(W^L f^{L-1}(W^{L-1} \cdots f^2(W^2 f^1(W^1 x)) \cdots)))$
- $W^l = (w^l_{jk})$: the weights between layer $l-1$ and l, where w^l_{jk} is the weight between the k-th node in layer $l-1$ and the j-th node in layer l
- f^l: activation functions at layer l

What is ANN

From OLS to ANN

Core: Activation Functions

Perception to ANN

Output Types

Complex ANN models

Types of Output

- continuous output (housing price, stock price)
- two classes output (default risk, fraud risk)
- multiple classifications output (ChatGPT,recommendation system)

1. continuous output

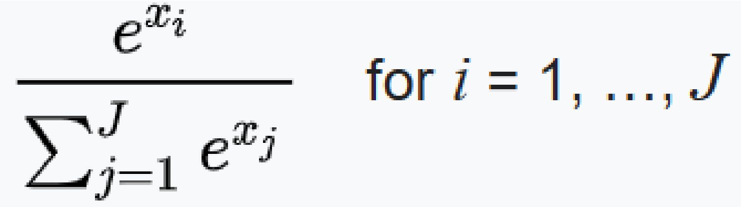
Linear activation (no activation) in the last layer

2. two classes output

sigmoid activation function in the last layer

3. Multiple classification output: Softmax

Softmax activation function in the last layer (looks like sigmoid, right)

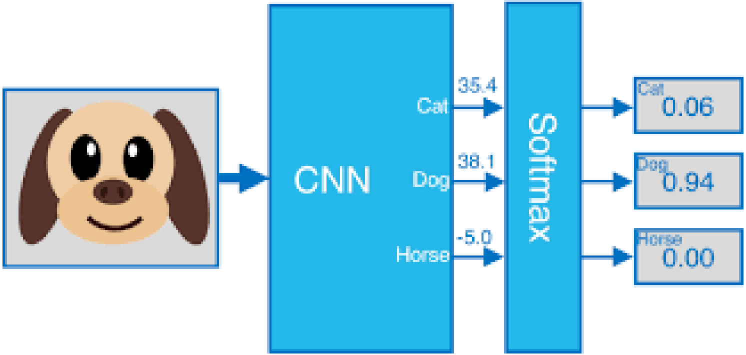


$$\frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}} \quad \text{for } i = 1, \ldots, J$$

Multiple classification output: Softmax

- Softmax: multiple classification:
- extension of sigmoid for multiple classification
- in last layer for transformer model and CNN model (ChatGPT last layer is 100,000 classes classification)

Multiple classification output: Softmax



4. loss function of ANN:

- $Loss(y, \hat{y})$
- $Loss(y, f^L(W^L f^{L-1}(W^{L-1} \cdots f^2(W^2 f^1(W^1 x)) \cdots)))$
- *Loss*: MSE loss for regression problem and Log loss (cross entropy) for classification problem

Training ANN: old friend Gradient Descent

$$W_{t+1}^{l} = W_{t}^{l} - \eta \frac{\partial Loss}{\partial W^{l}}$$

η is the learning rate, l is for any layers in the ANN

Training ANN: Details

- back-propagation
- chain-rule of the derivative
- questions: why called back propagation?

What is ANN

From OLS to ANN

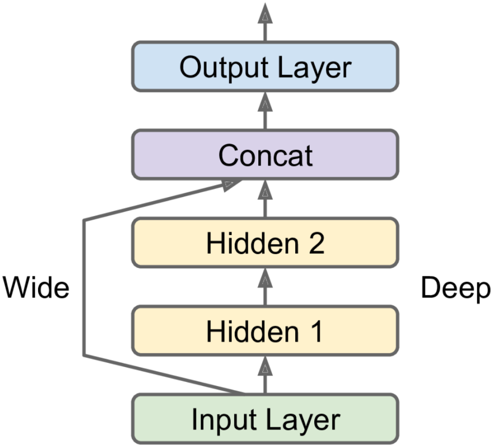
Core: Activation Functions

Perception to ANN

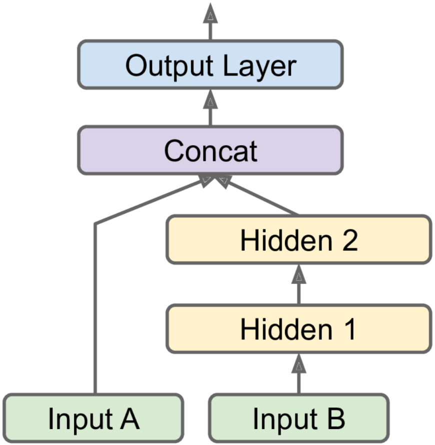
Output Types

Complex ANN models

Wide & Deep neural network (2016 paper by Heng-Tze Cheng)



Wide & Deep neural network 2



Wide & Deep neural network (2016 paper by Heng-Tze Cheng)

- Deep features vs Sallow features
- Useful for 1) recommendation system (Cheng 2016), 2) housing evaluation, 3) stock prediction ...
- why?

Reference

1. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow (3rd edition)
2. Wikipedia
3. geeksforgeeks
4. Kaggle
5. Wikipedia
6. ChatGPT
7. DeepSeek