

	VIETTEL AI RACE	Public 481
	GIỚI THIỆU VỀ MẠNG NƠ-RON NHÂN TẠO	Lần ban hành: 1

## 1. GIỚI THIỆU VỀ MẠNG NƠ-RON NHÂN TẠO

Mạng nơ-ron nhân tạo (Artificial Neural Networks – ANN) là nền tảng cốt lõi của trí tuệ nhân tạo hiện đại. Ý tưởng dựa trên cách bộ não sinh học xử lý thông tin thông qua các nơ-ron liên kết.

## 2. MẠNG NƠ-RON NHÂN TẠO

### 2.1 Yêu cầu trước khi làm thí nghiệm

Yêu cầu trước khi thực hàn
----------------------------

### 2.2 Mục đích của phần thí nghiệm

Mục đích của phần thí nghiệm:

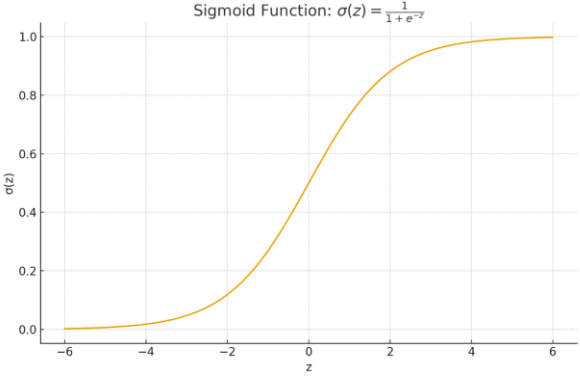
- Hiểu nguyên lý hoạt động của ANN.
- Làm quen với các công thức toán học mô tả quá trình huấn luyện.
- Ứng dụng ANN trong bài toán thực tế: phân loại, dự đoán, xử lý ảnh, ngôn ngữ.

### 2.3 Tóm tắt lý thuyết

#### 2.3.1 Nơ-ron nhân tạo

Định nghĩa	Công thức
Nơ-ron nhân tạo thực hiện phép biến đổi tuyến tính–tịnh tiến trên vector đặc trưng đầu vào, sau đó qua hàm kích hoạt phi tuyến để tăng năng lực biểu diễn; bias giúp dịch chuyển biên quyết định.	$z = \sum_{i=1}^n w_i x_i + b$ $y = f(z)$
Sigmoid nén giá trị về (0,1), thường dùng cho đầu ra nhị phân; nhưng dễ bão hòa gradient ở vùng biên.	$\sigma(z) = \frac{1}{1 + e^{-z}}$

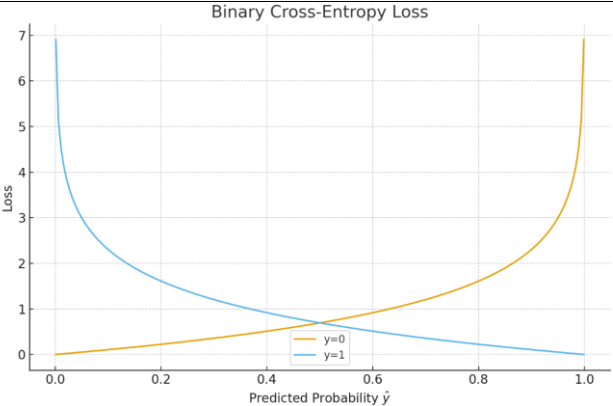
	VIETTEL AI RACE	Public 481
	GIỚI THIỆU VỀ MẠNG NƠN NHÂN TẠO	Lần ban hành: 1

	
Tanh là phiên bản tịnh tiến/scale của Sigmoid, đầu ra (-1,1), trung bình bằng 0 giúp hội tụ tốt hơn trong một số mạng	$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
ReLU giữ thành phần dương, triệt tiêu âm, giúp mạng sâu hội tụ nhanh; có biến thể Leaky ReLU khắc phục “neuron chết”.	$ReLU(z) = \max(0, z), \quad LeakyReLU(z) = \max(\alpha z, z)$

### 2.3.2 Mật mát và phân phối

Softmax chuyển vector logit thành phân phối xác suất trên K lớp; nhạy cảm với chênh lệch lớn giữa các logit.	$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$
Cross-Entropy đa lớp đo độ lệch giữa nhãn thật 1-hot và phân phối dự đoán; kết hợp Softmax cho huấn luyện phân loại.	$L = - \sum_{i=1}^K y_i \log \left( \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \right)$
Binary Cross-Entropy dùng cho nhị phân/đa nhãn; khuyến nghị dùng logit-stable triển khai để tránh tràn số.	$L = - \frac{1}{N} \sum_{i=1}^N (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$

	<b>VIETTEL AI RACE</b>	Public 481
	<b>GIỚI THIỆU VỀ MẠNG NƠN NHÂN TẠO</b>	Lần ban hành: 1

	
<i>MSE</i> thường dùng cho hồi quy; nhạy cảm với ngoại lai do bình phương sai số.	$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$
<i>Hinge loss</i> dùng trong SVM/NN phân biệt biên cứng; khuyến khích lề phân tách lớn.	$L = \sum_{i=1}^N \max(0, 1 - y_i \hat{y}_i)$

### 2.3.3 Tối ưu hoá

#### 2.3.3.1. Gradient Descent/SGD

<b>[CT1]</b>	$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t)$
--------------	--

#### 2.3.3.2. Gradient Descent/SGD

$$v_t = \mu v_{t-1} + \eta \nabla_{\theta} L_t, \quad \theta_{t+1} = \theta_t - v_t$$

	VIETTEL AI RACE	Public 481
	GIỚI THIỆU VỀ MẠNG NƠN NHÂN TẠO	Lần ban hành: 1

### 2.3.3.3. Adam

[CT2]	$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \widehat{m}_t = \frac{m_t}{1 - \beta_1^t}, \widehat{v}_t = \frac{v_t}{1 - \beta_2^t}, \theta_{t+1} = \theta_t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}$
-------	--

### 2.3.3.4. Weight Decay (L2)

[CT3]	$\theta_{t+1} = (1 - \eta\lambda)\theta_t - \eta \nabla_{\theta} L(\theta_t)$
-------	---

## 2.3.4. Regularization & Normalization

L1/L2 Regularization lần lượt khuyến khích thưa (sparsity) và nhỏ hoá tham số; tác động khác nhau lên giải pháp tối ưu.	$\Omega_{L1} = \lambda \sum_i  w_i , \Omega_{L2} = \lambda \sum_i w_i^2$
Dropout vô hiệu ngẫu nhiên đơn vị trong huấn luyện để phá đồng thích nghi; scale ở suy luận để bảo toàn kỳ vọng.	$\tilde{h}_i = h_i \cdot d_i, \tilde{h}_i \sim \text{Bernoullip}$
BatchNorm (train) chuẩn hoá theo mini-batch rồi affine transform với $(\gamma, \beta)$ ; cải thiện ổn định và tốc độ học. tránh tràn số.	$\mu_B = \frac{1}{m} \sum_i x_i, \sigma_B^2 = \frac{1}{m} \sum_i (x_i - \mu_B)^2, \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \hat{x}_i = \gamma \hat{x}_i$
MSE thường dùng cho hồi quy; nhạy cảm với ngoại lai do bình phương sai số.	$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$

	<b>VIETTEL AI RACE</b>	Public 481
	<b>GIỚI THIỆU VỀ MẠNG NƠON NHÂN TẠO</b>	Lần ban hành: 1

<p><i>Hinge loss</i> dùng trong SVM/NN phân biệt biên cứng; khuyến khích lề phân tách lớn.</p>	$L = \sum_{i=1}^N \max(0, 1 - y_i \hat{y}_i)$
--	---

2025-10-19 02.31.38\_AI Race

2025-10-19 02.31.38\_AI Race

2025-10-19 0