# Requirements: - 23/5

1) Tìm hiểu và trình bày về mô hình phân loại văn bản sử dụng Neural Network

2) Tìm hiểu bài toán phân loại văn bản đa nhãn Multi-label classification và các phương pháp giải quyết nó, tức là 1 input có thể thuộc về nhiều nhãn

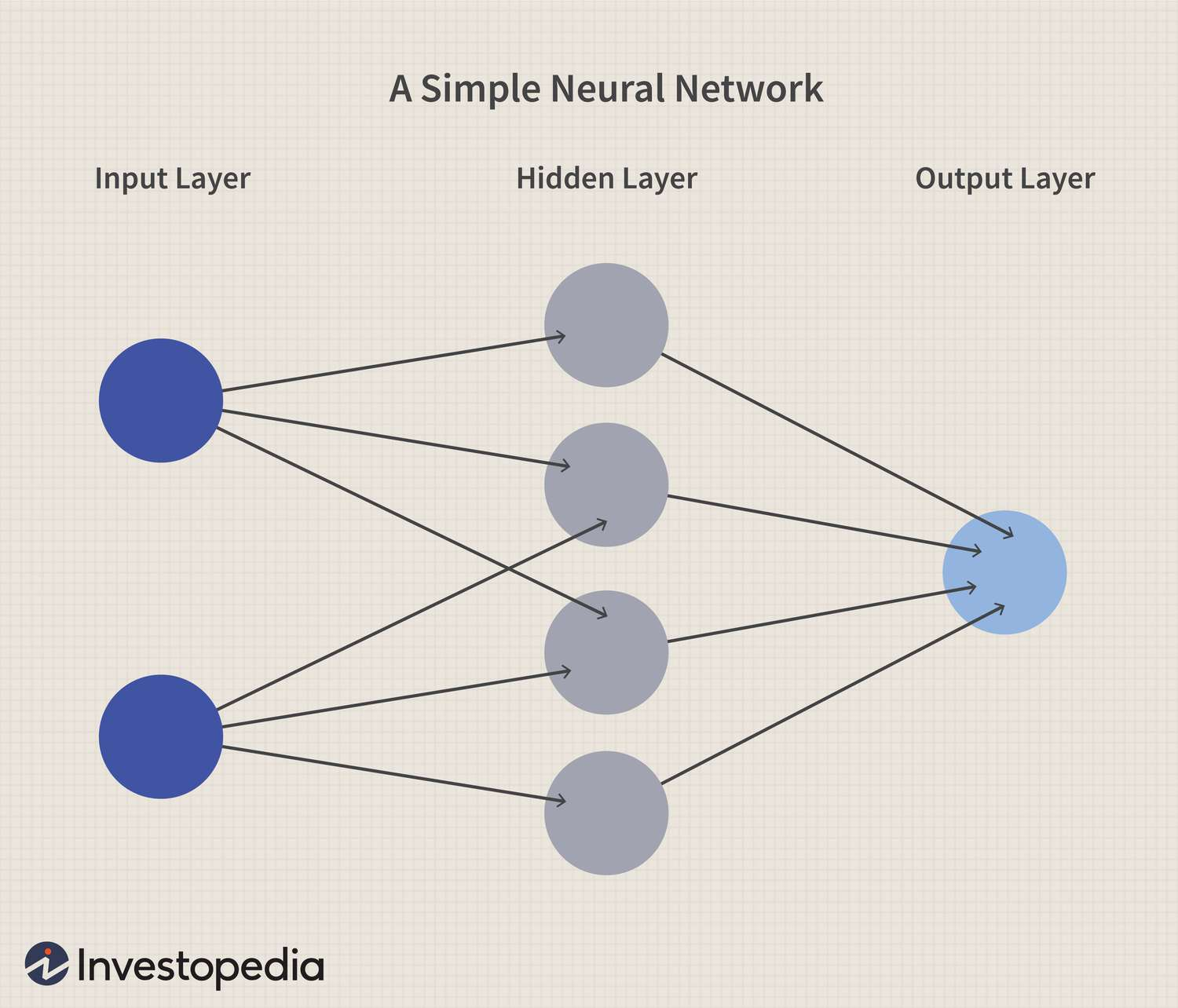
3) Thực nghiệm, trình bày các giải pháp bằng coding

Dữ liệu tự collect

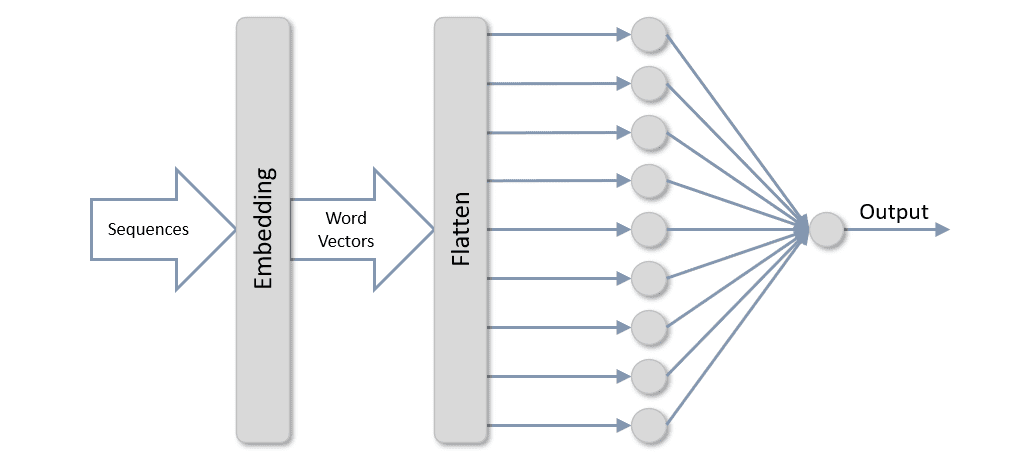
## Chị Trang

### Introduction to Neural Networks

Neural networks are computational models inspired by the human brain's structure and functionality. They consist of interconnected layers of nodes (neurons), where each node processes input and passes the output to the next layer. Neural networks are particularly powerful for complex pattern recognition tasks, including text classification.

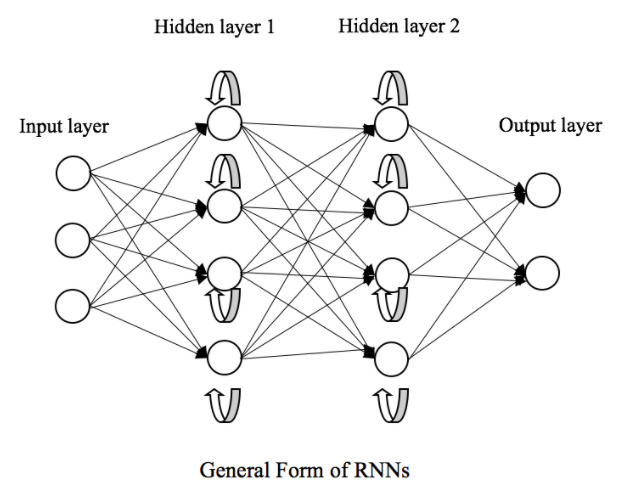


A neural network for text classification has two additional layers: an embedding layer and a flatten layer. The embedding layer converts words into vectors, which encode information about the relationships between the words. The flatten layer transforms the word vectors into a format suitable for the dense layer.



### RNN for Text Classifications in NLP

In Natural Language Processing (NLP), Recurrent Neural Networks (RNNs) are a potent family of artificial neural networks that are crucial, especially for text classification tasks. RNNs are uniquely able to capture sequential dependencies in data, which sets them apart from standard feedforward networks and makes them ideal for processing and comprehending sequential information, like language. RNNs are particularly good at evaluating the contextual links between words in NLP text classification, which helps them identify patterns and semantics that are essential for correctly classifying textual information. Because of their versatility, RNNs are essential for creating complex models for tasks like document classification, spam detection, and sentiment analysis.



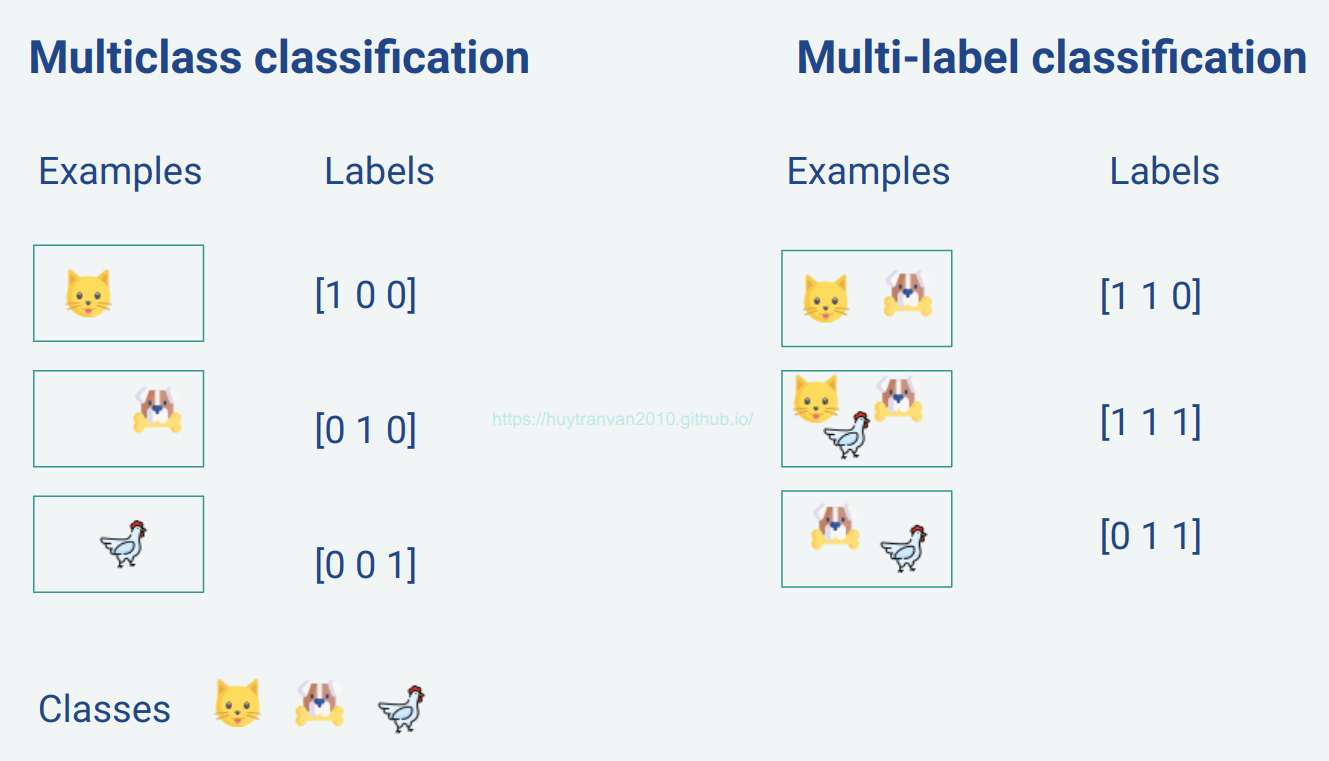
Here's a general approach:

1. **Text Preprocessing:** Convert raw text into numerical representations that the neural network can process. Common techniques include tokenization (splitting text into words or subwords), removing stop words, and lemmatization/stemming.
2. **Embedding Layer:** Transform tokens into dense vectors (embeddings) that capture semantic meaning. Pre-trained embeddings like Word2Vec or GloVe can be used, or embeddings can be learned during training.
3. **RNN Layer:** Process the sequence of embeddings using RNN, LSTM, or GRU layers to capture the sequential nature of the text.
4. **Dense Layer:** Use fully connected layers to map the RNN's output to the desired number of classes.
5. **Output Layer:** Apply an activation function like softmax (for multi-class classification) or sigmoid (for binary classification) to produce the final classification probabilities.

## Vũ

Link: https://huytranvan2010.github.io/Multi-label-classification/

### Multi class and Multi-label

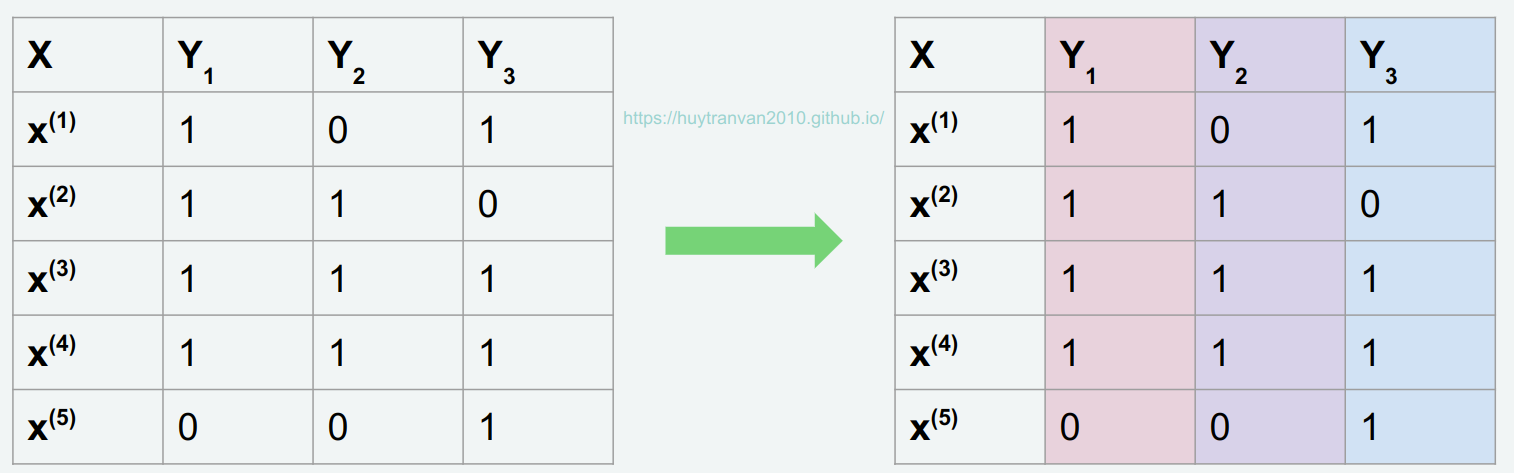
* In multi-class classification, each instance (e.g., document, image, data point) belongs to exactly one class out of a set of mutually exclusive classes. This means that for a given instance, the model must choose one and only one class from multiple possible classes.
* Examples:
* Classifying types of fruits: apple, banana, orange, etc. Each fruit belongs to one and only one type.
* Multi-label text classification problem: This is a problem in the field of text classification where an input document can belong to multiple labels. For example, an online newspaper can be classified as political information, social, or both.
* 

### Technique to solve this problem:

* Problem Transformation
* Adapted Algorithm
* Ensemble approaches
* Neural Network

### Problem transformation

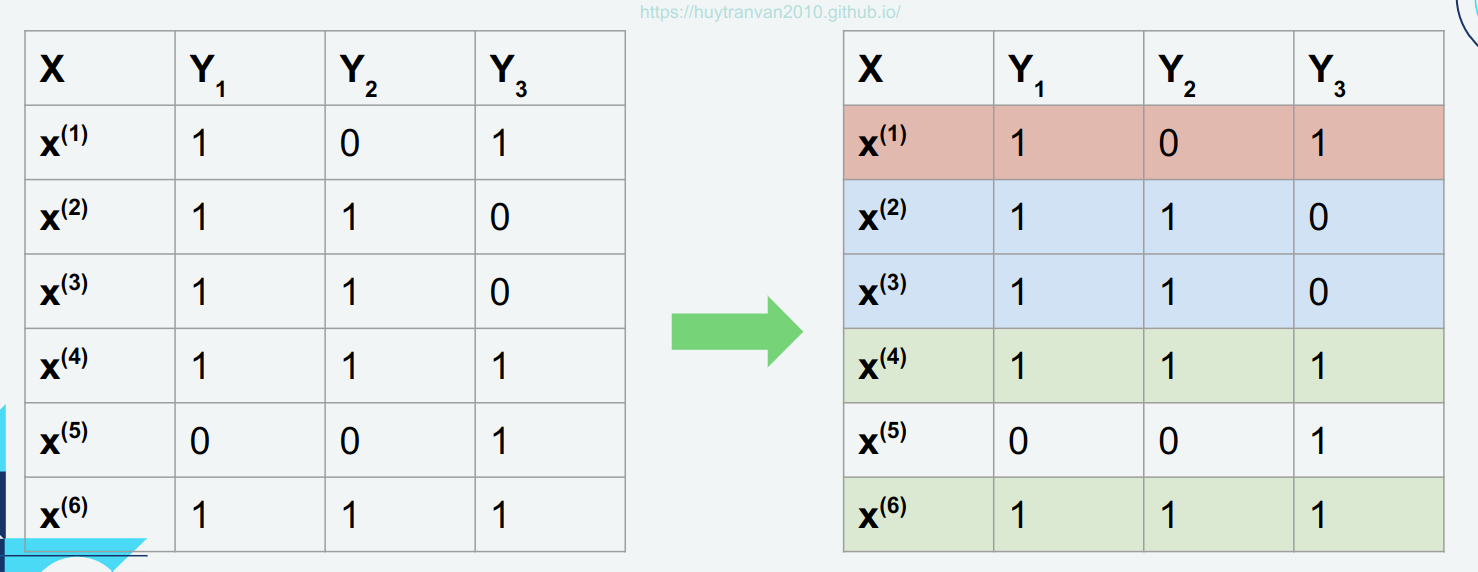
#### Binary Relevance

* With this method, we try to convert a multi-label problem into a single-label problem. This method can be implemented through some of the following ways: Binary Relevance, Classifier Chains, Label Powerset
* 
* With binary relevance, we will divide the problem into 3 single class classifications (since there are 3 classes here). From there, it is easy to handle each individual problem, and then the final results can be combined into one.

#### Classifier Chains

* Classifier Chains (CC): Arrange the labels in a specific order and each label will be predicted based on the previous labels in the chain.
* Original data
* 
* In classifier chains, we also have 3 single label problems corresponding to 3 classes.
* 
* Attention:
* Classifier 1: **Feature**: X —- **target**: Y1
* Classifier 2: **Feature**: X, Y1 —- **target**: Y2
* Classifier 2: **Feature**: X, Y1, Y2 —- **target**: Y3

#### Label Powerset

* Label Powerset (LP): Treats every unique combination of labels as a single class, converting the problem into a multi-class problem.
* 
* As shown in the figure above, data points x2 and x3 are considered to have the same label,
* x4 and x6 are considered to have the same label. x1 has its own label, x5 has its own label.
* At this point, we have a total of **4 new classe**s for the new model. This model serves for **multi-class** classification problems. It is noticed that the **multi-labe**l classification problem has now been **transformed** into a **multi-class** classification problem.