Representation Models

Text

Statistical vs ML Methods

- Statistical methods: [适用] small data [优点] efficient and interpretability
- ML-based models: [适用] large data [优点] capture complex linguistic patterns .

Word2Vec vs Transformer models

- Word2Vec: assumes that a word’s meaning is determined by its context. It trains a NN to learn the relationship between words and their surrounding context, generating word embeddings.

  [缺点] (1) static embeddings (struggle with words with multiple meanings); (2) only captures local context (unable to model long-range dependencies); (3) not retain word order information, (4) cannot handle out-of-vocabulary (OOV) words.
- Transformer: using self-attention to capture relationships between words across an entire sequence, generating contextual embeddings.

  [缺点] (1) high computational complexity, (2) large-scale training data, (3) no interpretability.

CBOW vs Skip-gram

- Both are Word2Vec
- CBOW (Continuous Bag of Words): Predicts the target word based on its surrounding context words.
- Skip-gram: Predicts context words given a target word. [缺] large data [优] work for rare words.

1. BERT

架构

1. Input Representation: each token is mapped to a dense vector: captures semantic and positional.
2. Transformer Encoder Layers: (a) Self-Attention: analyze relationships between tokens across the entire sequence, understanding each word in context; (b)Feed-Forward Neural Networks (FFN): further transform, capturing higher-level patterns.
3. Output Laye: generate task-specific predictions.

训练

1. Masked Language Modeling: randomly masks tokens and trains model to predict, cross-entropy loss
2. Next Sentence Prediction: every sentence has two segments, predict the second segment is the actual next segment, cross-entropy loss.

1. Cross-Encoder BERT

架构

1. Input：[CLS] Query Text [SEP] Item Text [SEP].
2. Output: [CLS] representation input to fully connected layer (MLP), produce a similarity score

训练

1. Pointwise : encourages the model to assign higher probabilities to relevant documents.
    - Data Point: query + one document + label (relevant/irrelevant)
    - Output: one similarity score $\longrightarrow$ (sigmoid) $\longrightarrow$ relevance score

    - Loss: binary cross-entropy
2. Pairwise (contrastive) : encourages the model to give higher scores to the more relevant document.
    - Data Point: query + two documents + label (which doc is relevant)
    - Output: two similarity scores s_1 and $s_2 \rightarrow$ (softmax) $\rightarrow$ probabilities p_1 and p_2
    - Loss: if doc 1 is label, $\mathcal{L} = -\log p_1$

2. BoW

<u>**核心思想**</u>

Bag of Words converts text/document into fixed-dimensional vectors by counting word frequencies

<u>**做法**</u>

1. form a fixed-size vocabulary: extract unique words from all documents.
2. create frequency vector: for each document, counts the frequency of each word.

<u>**缺点**</u>

1. not consider the order
2. not capture the semantic meaning
3. vector is sparse

3. TF-IDF

<u>**核心思想**</u>

Some words appear frequently but no much meaning. Replace frequencies in BoW by metric reflecting importance of a word in a document/sentence.

<u>**做法**</u>

1. calculates a word's frequency in a document
2. calculate how common each word is across all documents
3. calculate importance by division.

<u>**缺点:**</u>

1. Recompute how common when a new sentence is added.

4. CBOW

<u>**架构**</u>

1. If window size 5, the input includes the two words before and after the target word.
2. Input Layer: words are converted to one-hot vectors.
3. Embedding Layer: each one-hot vector is mapped to a word embedding, and the embeddings of 5 context words are then averaged to obtain a context representation.
4. Output Layer, the context representation is projected back to the vocabulary space, and softmax is applied (to compute the probability distribution over the vocabulary).

<u>**缺点**</u>

1. Softmax function in output layer is expensive when the vocabulary size is large.

> [解决方法] Negative Sampling :
>
> 1. samples k negative examples;
> 2. uses the Sigmoid function to compute the probability for positive/negative samples;

> 3. calculate Negative Log-Likelihood (NLL) loss;
> 4. updates only the weights related to the current samples
>
> $\mathcal{L} = -\log\sigma(v_{w_c}^T v_{w_t}) - \sum_{i=1}^{k}\log\sigma(-v_{w_{n_i}}^T v_{w_t})$
>
> - $\sigma(x) = \frac{1}{1+e^{-x}}$ is Sigmoid function.
> - $v_{w_t}, v_{w_c}, w_{n_i}$ is the vector representation of center word w_t, positive, negative samples.

5. Sentence

Pooling Methos: (1) [CLS] Token Representation， (2) Average (Mean) Pooling，(3) Max Pooling，

(4) Attention- or Weight-Based Pooling，(5) Concatenation of Multiple Approaches

Image

CNN vs ViT

- Convolutional Neural Networks: [适用] small to mid data [优点] efficient [缺点] cannot capture global patterns (struggle with long-range dependencies)
- Vision Transformers: [适用] large data [优点] self-attention to model global patterns [缺点] high computational complexity, no interpreterbility

1. ViT

核心思想

ViT splitting image into small patches, treating these patches like a sequence of tokens, learn both local details and global context (long-range dependencies)

架构

1. Preporcess: divide an image into small, equally sized patches (like tiles in a puzzle).
2. Input representation: each patch is converted to a vector, capturing visual features and position.
3. Transformer Encoder: (1) multi-head self-attention: understand the relationships between different parts of the image, (2) a feed-forward network: further transform, capturing higher-level patterns.
4. Output: perform a specific vision task.

训练

1. 思想：processes positive/negative image pairs, maximize (minimize) the similarity for positive (negative) pairs
2. 训练数据:

   (1) human manually judgment: [优] high accuracy [缺] costly and time-consuming.

   (2) user clicks to infer similarity: [优] scalable [缺] noisy and sparse.

   (3) self-supervision (e.g, transformations-rotation): [优] no manual effort, less noisy [缺] not reflect real-world scenarios.
3. 损失函数:

   Pairwise: binary classification, cross-entropy loss

   Batchwise: InfoNCE Loss, $L = -\log\frac{\exp(sim(z_i,z_j)/\tau)}{\sum_k \exp(sim(z_i,z_k)/\tau)}$.

   > InfoNCE Loss applies a softmax over all candidate pairs in the batch. This maximizes the probability of the positive pair while pushing negative pairs further apart.

2. CNN

核心思想

CNN looks at small parts at a time. First detects simple features like edges and textures, then gradually combines them to recognize more complex patterns and objects.

架构

1. Convolutional layer: small filters (kernels) slide over the input image to detect local patterns such as edges and textures. Each filter produces a feature map that highlights specific characteristics of the image.
2. Pooling layer: reduce the size of the feature while retaining the most important information. E.g., max pooling selects the most significant values.

   > By stacking multiple convolutional and pooling layers, the network gradually learns more complex features. The early layers detect simple edges, while deeper layers recognize object parts and entire objects.

3. Fully connected layer: the extracted features are flattened into a vector, passed through fully connected layers to produce the predicted label.

3. Video

处理方法

1. Frame-based processing：(1) decodes video to frames, (2) samples key frames, as images
2. End-to-end: directly feeds video for spatiotemporal feature learning

视频模型

1. 3D CNN (e.g., C3D, I3D) : convolute along the temporal dimension too, capture spatial and temporal information simultaneously.
2. Transformer (e.g., TimeSformer, ViViT): splits videos into a sequence of spatial × temporal patches, use multiple attention heads for spatial attention and temporal attention.

Multi-modal

CLIP vs BLIP

- CLIP uses contrastive learning to align image and text features, good for image-text retrieval
- BLIP not only uses contrastive learning but also train generation tasks, good for tasks like image captioning and visual question answering.

1. CLIP

架构：Two independent encoders: vision (e.g., ViT or ResNet) and text (e.g., Transformer). Image and text are embedded into a shared space, can directly caluclate similarity measures

训练：Contrastive learning on millions of image-text pairs.

Anomaly Detection

Early vs Late

- Late fusion: each modality is processed independently, and their predictions are combined to make a final prediction.

  [优] This allows independent model training and improvement.

  [缺] Fail when each modality is benign but harmful content arises from the combination.

  [缺] Requires separate training data, time-consuming and expensive.

- Early fusion: the modalities are combined first, and then make a prediction.

  [优] Only need to collect training data for the whole model.

  [优] The model can capture if each modality is benign but combination is harmful.

  [缺] Learning task is difficult.

Multiple Classes Classifer

- One binary classifier per harmful class: each harmful class has its own binary classifier

  [优] Allowing independent model improvements

  [缺] Expensive and time-consuming.

- Multi-label classifier: a single model predicts probabilities for multiple harmful classes

  [优] Reducing training and maintenance costs

  [缺] Input features may not be transformed optimally for each class, potentially affecting accuracy.

- Multi-task classifier: Uses a shared model to capture the fundamental relationship between users and items. This shared representation is used by seperate modules to predict metrics for each task.

  [优] Computationally efficient: one model avoids redundant computations

  [优] Data efficient: training data from one task benefits others

Data Challenge

Imbalanced Data

1. Resampling: Undersampling (delete normal), Oversampling (replicate, data augment)
2. Weighted Loss Function
3. One-Class Learning: train the model on only normal to learn distribution
4. Ensemble Learning: Bagging, Boosting (assigning higher weights to misclassified anomalies)
5. Semi-Supervised and Unsupervised Learning: