Module-IV
Multimodal Fusion & Co-Learning

November 9, 2024

Fusion Strategies in Multimodal AI

Overview:

Fusion strategies combine information from multiple modalities to improve learning and decision-making in AI systems.

- **Why Fusion?**
  - Enhance performance by leveraging complementary information.
  - Capture richer representations of complex data.

Types of Fusion Strategies

- **Early Fusion:**
  - Combines raw data from different modalities before processing.
  - Suitable for scenarios where modalities are closely related.
- **Late Fusion:**
  - Integrates outputs from separate models for each modality.
  - Effective when modalities provide independent insights.
- **Mid Fusion:**
  - Combines features at intermediate layers of a model.
  - Balances early and late fusion by allowing partial cross-modal interaction.

Early Fusion

Definition:
Early fusion involves concatenating or aggregating raw data from multiple modalities before processing them through the model.

- **Example:** In audio-visual tasks, RGB frames and audio waveforms are combined into a single input tensor.
- **Pros:**
  - Captures joint information.
  - Simplifies model architecture.
- **Cons:**
  - Increases input dimensionality.
  - May dilute modality-specific features.

Late Fusion

Definition:
Late fusion involves processing each modality independently and then combining their outputs for final predictions.

- **Example:** Separate models for text, images, and audio produce distinct outputs that are aggregated for classification.
- **Pros:**
  - Preserves modality-specific information.
  - Flexible to changes in modalities.
- **Cons:**
  - Ignores potential inter-modal relationships.
  - Requires careful design of fusion mechanism.

Mid Fusion

Definition:
Mid fusion allows interaction between modalities at intermediate layers of a model, facilitating cross-modal learning.

- **Example:** Combining features from different modalities at various points in a neural network architecture.
- **Pros:**
  - Enables information exchange while maintaining modality integrity.
  - Can leverage strengths of both early and late fusion.
- **Cons:**
  - More complex architecture.
  - Requires careful tuning of interaction points.

Conclusion

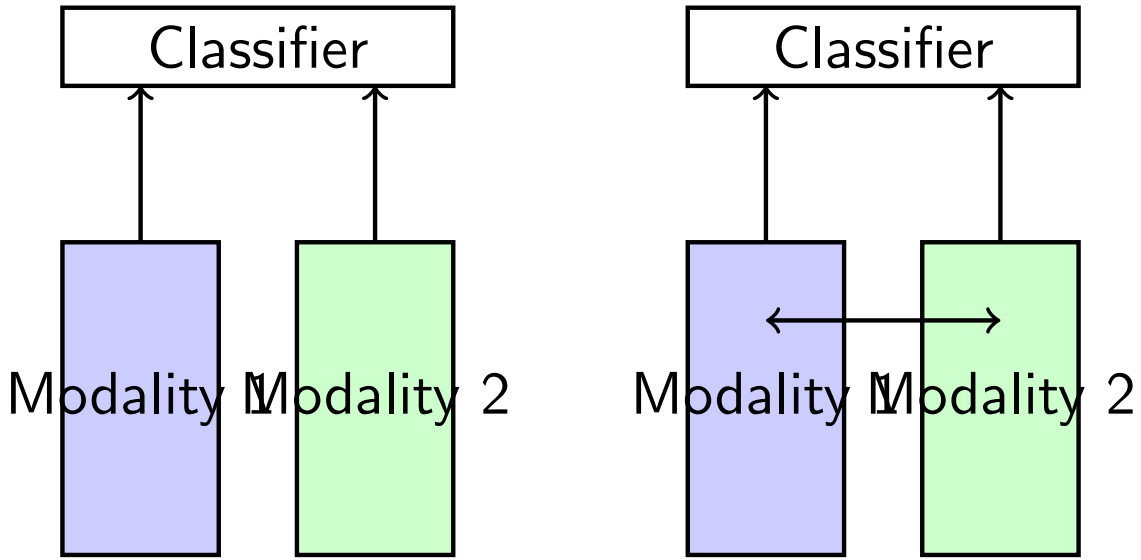
- Fusion strategies are crucial in multimodal AI for leveraging diverse data sources.
- Selection of fusion strategy depends on the task, data characteristics, and model architecture.
- Ongoing research explores novel fusion techniques to enhance multimodal learning.

Cross-modal Fusion

- Unlike **Late Fusion**, where modalities are combined only after classification, cross-modal information can be exchanged at earlier stages.
- We investigate two pathways for **Cross-modal Fusion**:
  - **Mid Fusion**: Pairwise self-attention across hidden units in later layers.
  - **Fusion Bottlenecks**: Tight latent units restricting attention flow within a layer.

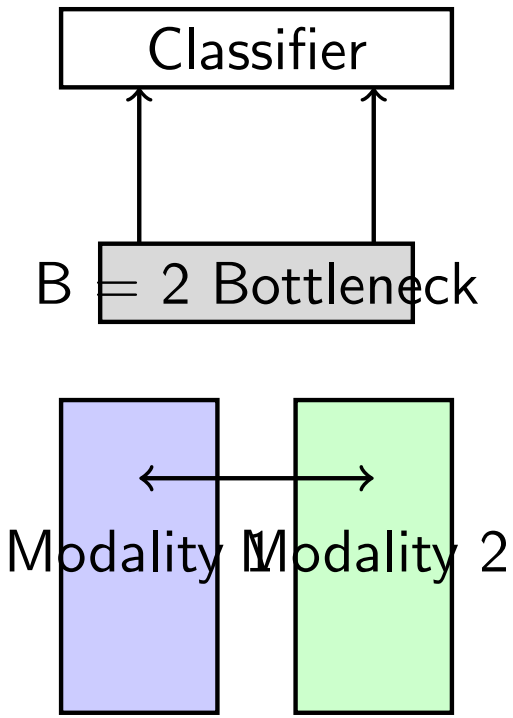
Late Fusion vs. Mid Fusion

- **Late Fusion (Left)**: No cross-modal information is exchanged until after the classifier.
- **Mid Fusion (Right)**: Pairwise self-attention applied across all hidden units in later layers.



Fusion Bottlenecks

- **Fusion Bottlenecks**: Restricts attention flow through tight latent units.
- Can be applied in conjunction with mid fusion for optimal performance (**Bottleneck Mid Fusion**).



Bottleneck Mid Fusion: Key Takeaways

- **Fusion Bottlenecks**: Control the flow of attention through restricted latent units.
- **Bottleneck Mid Fusion**: Combines bottlenecks and mid fusion to optimize cross-modal interactions.
- Grey boxes indicate tokens receiving attention flow from both modalities.

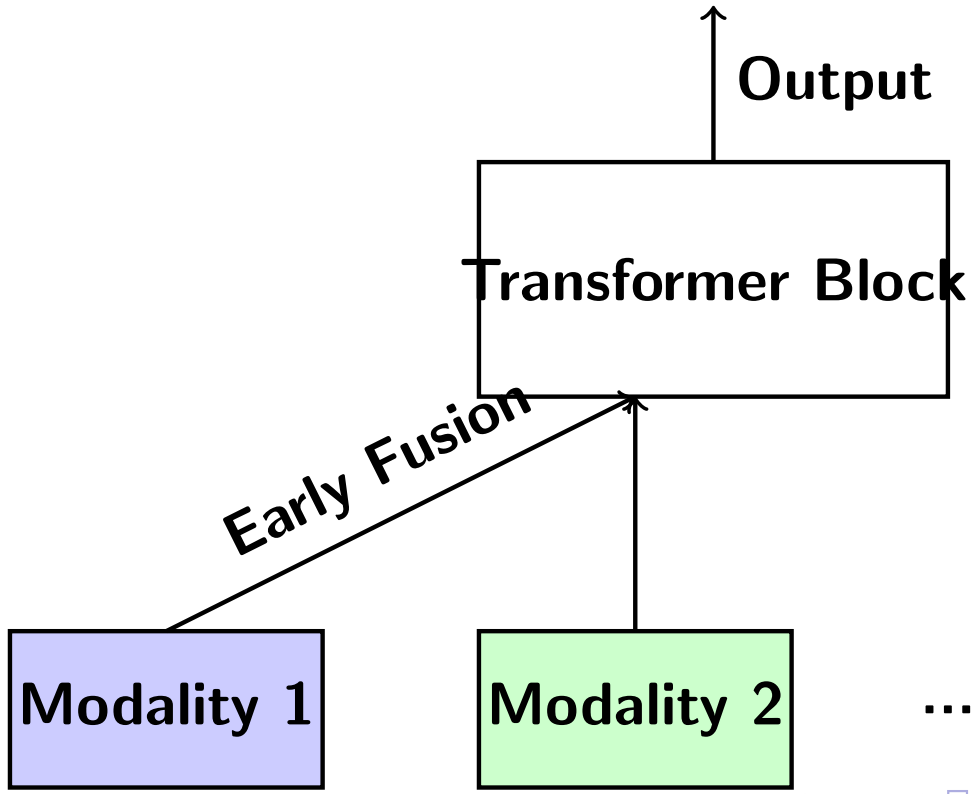
Multimodal Transformers with Self-Attention[1]

- The **self-attention** mechanism of transformers provides a natural way to connect multimodal signals.
- Applications include:
  - **Audio enhancement**, **Speech recognition**
  - **Image segmentation**, **Cross-modal sequence generation**
  - **Image/video retrieval**, **Visual navigation**
  - **Image/video captioning**, **Classification**
- Inputs to transformers often use the output representations of **single modality CNNs**.

[1] https://arxiv.org/abs/2107.00135

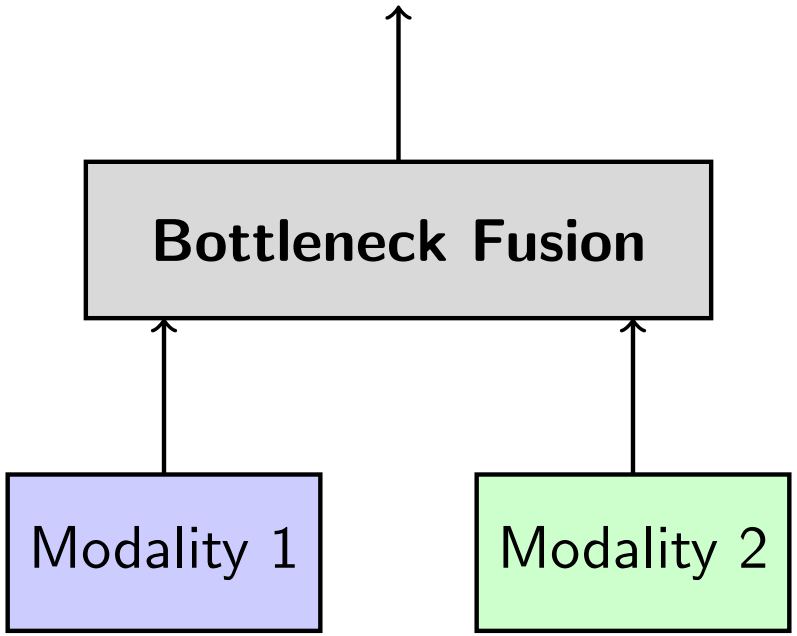
Multimodal Transformer Architecture

- Unlike traditional approaches, **transformer blocks** are used throughout the network, with only a single convolutional layer to rasterize 2D patches.
- **Tokens from different modalities** are combined directly as inputs to the transformer (Early Fusion).



Multimodal Bottleneck Transformer (MBT)

- We propose the **Multimodal Bottleneck Transformer (MBT)**, which channels cross-modal connections through bottlenecks.
- These bottlenecks restrict the flow of attention between modalities, enhancing efficiency and accuracy.



Vision Transformer (ViT) and Audio Spectrogram Transformer (AST)

Key Insights:

- ViT and AST adapt the Transformer architecture for 2D inputs.
- Extract N non-overlapping patches from RGB image (or audio spectrogram), $x_i \in \mathbb{R}^{h \times w}$.
- Convert patches into 1D tokens $z_i \in \mathbb{R}^d$:

$$z = g(x; E, z_{\text{cls}}) = [z_{\text{cls}}, Ex_1, Ex_2, \ldots, Ex_N] + p$$

where E is a linear projection, z_{cls} is a special classification token, and $p \in \mathbb{R}^{(N+1) \times d}$ is a learned positional embedding.

Contd...

- **Transformer Encoder:**
  - Tokens are passed through a sequence of L Transformer layers.
  - Each layer consists of Multi-Headed Self-Attention (MSA), Layer Normalization (LN), and MLP blocks:

$$y_l = \text{MSA}(\text{LN}(z_l)) + z_l$$

$$z_{l+1} = \text{MLP}(\text{LN}(y_l)) + y_l$$

Multi-Headed Self-Attention and Cross-Attention

Multi-Headed Self-Attention (MSA):

$$\text{MSA}(X) = \text{Attention}(W_Q X, W_K X, W_V X)$$

where queries, keys, and values are linear projections of the input.

Multi-Headed Cross Attention (MCA):

$$\text{MCA}(X, Y) = \text{Attention}(W_Q X, W_K Y, W_V Y)$$

where X forms the query and Y forms the keys and values, enabling cross-modal interactions (used in multimodal cases).

Multi-Headed Cross-Attention

Concept:

- Relates two different datasets: queries (Q) and context (K, V).
- Multiple attention heads learn different relationships.

Attention Calculation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Example:

- **Text Queries (Q):** "A dog is playing in the park."
- **Image Features (K, V):** Features from an image of a dog in a park.

Attention Heads:

- Head 1: Focuses on the relationship between "dog" and dog features.
- Head 2: Focuses on the relationship between "playing" and action features.

Vanilla Fusion Model

Overview:

- A straightforward fusion model utilizing a regular transformer for multimodal inputs.
- Tokenization of video clips involves:
  - Uniformly sampling F RGB frames.
  - Converting the audio waveform into a single spectrogram.

Contd...

- **Tokenization Process:**
  - Each frame and spectrogram are embedded independently, following the encoding in ViT.
  - Total RGB patches: N_v from all F sampled frames: $x_{\text{rgb}} \in \mathbb{R}^{N_v \times d}$.
  - Total spectrogram patches: N_a: $x_{\text{spec}} \in \mathbb{R}^{N_a \times d}$.
  - Concatenated token sequence:

$$z = [z_{\text{rgb}} \parallel z_{\text{spec}}]$$

where

$$z_{\text{rgb}} = g(x_{\text{rgb}}; E_{\text{rgb}}, z_{\text{cls-rgb}}) \quad \text{and} \quad z_{\text{spec}} = g(x_{\text{spec}}; E_{\text{spec}}, z_{\text{cls-spec}}).$$

Multimodal Encoder

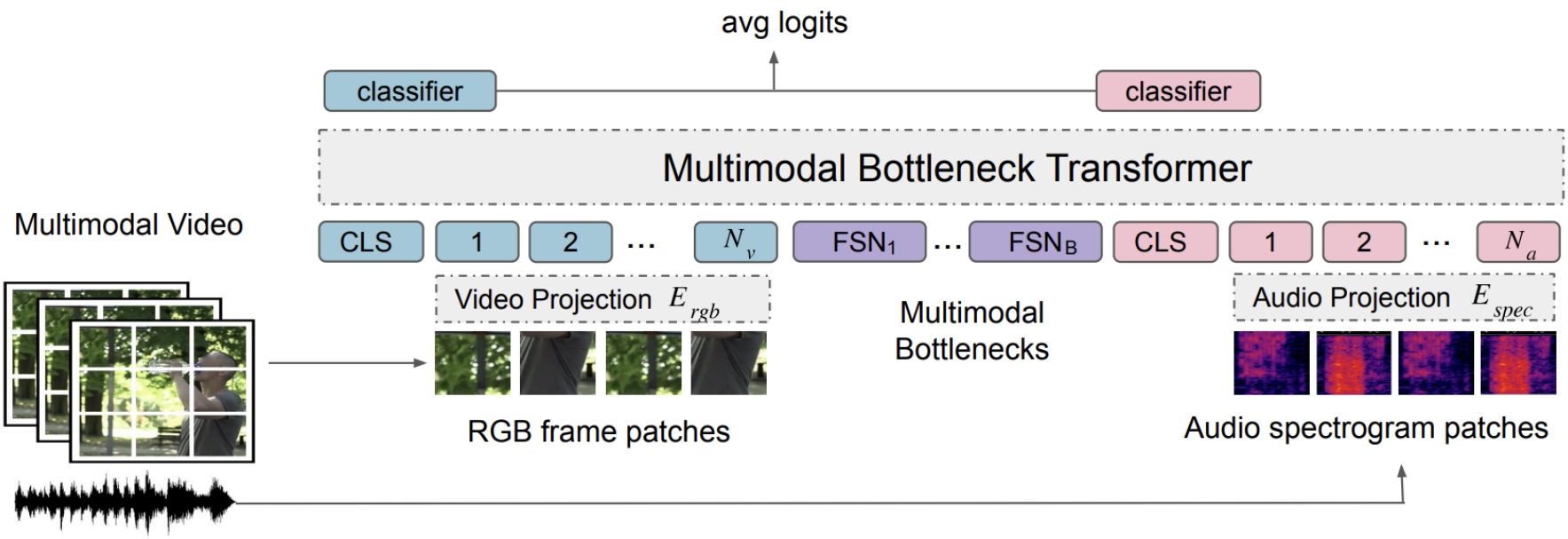
Encoder Architecture:

- The multimodal encoder applies a series of transformer layers:

$$z^{l+1} = \text{Transformer}(z^l; \theta)$$

- Attention flows freely through the network:
  - Each RGB token can attend to all other RGB and spectrogram tokens.
  - Transformer refers to a standard transformer layer with vanilla self-attention blocks.

Architecture



Architecture

- A Multimodal Fusion Transformer applied to audiovisual inputs.
- The input sequence consists of image and spectrogram patches. These are then projected into tokens and appended to special CLS (classification) and FSN (fusion bottleneck) tokens.
- The transformer encoder then uses self-attention to model unimodal information, and restricts cross-modal information flow via cross attention to the bottleneck tokens at multiple layers of the network.

Fusion Bottleneck Tokens

Objective:

- To tame the quadratic complexity of pairwise attention in multimodal transformer models.

Introduction of Fusion Bottleneck Tokens:

- Introduce a small set of bottleneck tokens:

$$z_{\text{fsn}} = [z_{\text{fsn}}^1, z_{\text{fsn}}^2, \ldots, z_{\text{fsn}}^B]$$

- The new input sequence becomes:

$$z = [z_{\text{rgb}} \parallel z_{\text{fsn}} \parallel z_{\text{spec}}]$$

Quadratic Complexity of Pairwise Attention in Multimodal Transformers

Self-Attention Mechanism:

- Each token attends to every other token in the sequence.
- Computation of attention scores for a sequence of length n involves the query (Q), key (K), and value (V) matrices:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Q, K, and V are $n \times d$ matrices (where d is the hidden dimension).
- The matrix multiplication QK^T produces an $n \times n$ matrix of attention scores.

Complexity:

- **Time complexity:** $O(n^2 d)$
- **Space complexity:** $O(n^2)$ (for storing attention scores)

Contd...

Multimodal Transformer:

- For multimodal data (text, images, audio), the total sequence length is $n = n_t + n_i + n_a$ (e.g., text tokens n_t, image tokens n_i, and audio token n_a).
- Example: If $n_t = 100$ and $n_i = 196$ (for image patches), $n = 296$.
- The complexity becomes $O(n^2 d) = O(296^2 d)$, which grows quadratically with the number of tokens.

Token Representation and Attention Flow

Cross-Modal Attention Flow:

- For layer l, token representations are computed as:

$$[z_i^{l+1} \parallel \hat{z}_{\text{fsn}i}^{l+1}] = \text{Transformer}([z_i^l \parallel z_{\text{fsn}}^l]; \theta_i)$$

- Bottleneck tokens updated:

$$z_{\text{fsn}}^{l+1} = \text{Avg}_i(\hat{z}_{\text{fsn}i}^{l+1})$$

Modality-Specific Updates:

- Modality-specific bottleneck tokens $\hat{z}_{\text{fsn}i}$ are updated separately with audio and visual information.
- Cross-modal attention must flow through these bottlenecks, forcing the model to condense necessary information.

Benefits and Generalization

Benefits:

- Reduces computational complexity while maintaining or increasing performance in multimodal fusion.
- The formulation is generic to the type and number of modalities used in the model.

Conclusion:

- Fusion bottlenecks effectively streamline information exchange between modalities, enhancing the efficiency of the transformer architecture.

Co-Learning in Multimodal AI

Definition: Co-learning aims to transfer information learned through one (or more) modalities to tasks involving another. This consists of adding external modalities during training, learning a joint representation space, and investigating how the joint model transfers to unimodal functions during testing.

Why Co-Learning?

- Improves unimodal systems by incorporating external data.
- Useful for low-resource target tasks.
- Provides insights for other multimodal tasks by learning joint representations.

Categories of Co-Learning Methods:

- **Category 1:** Focuses on the type of cross-modal interactions.
- **Category 2:** Explores learning joint representation space.
- **Category 3:** Examines how co-learning models transfer to unimodal tasks.

Category 1: Focuses on Cross-Modal Interactions

Cross-modal interaction types:

- **Early Fusion:**
  - Features from different modalities are concatenated early in the model pipeline.
  - Example: Image and text features combined for Visual Question Answering (VQA).
- **Late Fusion:**
  - Each modality is processed separately, and outputs are combined later.
  - Example: Separate models for text and image with ensemble at the output stage.
- **Cross-Attention:**
  - One modality attends to the features of another modality.
  - Example: In video captioning, text attends to video frames for better description.

Category 2: Joint Representation Space

Learning a shared representation across modalities:

- **Multimodal Embeddings:**
  - Both modalities (e.g., text and image) are projected into the same embedding space.
  - Example: CLIP model for learning shared image-text embeddings.
- **Shared Latent Space:**
  - Data from multiple modalities is transformed into a common latent space.
  - Example: Multimodal Variational Autoencoders (VAEs) for shared space learning.
- **Canonical Correlation Analysis (CCA):**
  - Maximizes the correlation between two modalities in a shared space.
  - Example: Aligning text and image features by maximizing their correlation.

Category 3: Transfer to Unimodal Tasks

Leveraging co-learning models for unimodal tasks:

- **Zero-shot Learning:**
  - Multimodal models are used for unimodal tasks without direct supervision.
  - Example: Image-text model applied to image classification using text labels.
- **Multimodal Pretraining for Unimodal Fine-tuning:**
  - A model is pretrained on multimodal data and fine-tuned on a unimodal task.
  - Example: Pretraining on video-text pairs, fine-tuning for text classification.
- **Knowledge Transfer:**
  - Knowledge from one modality helps in another modality's task.
  - Example: A model trained on video-text can be adapted for text-only summarization.

What is Joint Representation Space?

- A joint representation space is a unified latent space where data from different modalities (e.g., text, image) is projected.
- The goal is to align related elements from different modalities and represent them close together in the same space.
- This allows effective cross-modal reasoning, where the relationships between modalities are captured.

Modality-Specific Feature Extraction

- Each modality is first processed to extract modality-specific features:

$$T = f_{\text{text}}(x_{\text{text}}) \in \mathbb{R}^d \quad \text{(Text features)}$$

$$I = f_{\text{image}}(x_{\text{image}}) \in \mathbb{R}^m \quad \text{(Image features)}$$

where f_{text} and f_{image} are the feature extraction functions for text and image, respectively.

Projecting into Joint Representation Space

- Features from each modality are projected into a shared latent space:

$$z_{\text{text}} = g_{\text{text}}(T) \in \mathbb{R}^k \quad \text{(Text in joint space)}$$

$$z_{\text{image}} = g_{\text{image}}(I) \in \mathbb{R}^k \quad \text{(Image in joint space)}$$

where g_{text} and g_{image} are projection functions into the joint space, and k is the dimension of the joint space.

Contrastive Loss for Aligning Modalities

- The objective is to minimize the distance between related pairs and maximize the distance between unrelated pairs:

$$L_{\text{con}}(z_{\text{text}}, z_{\text{image}}) = \begin{cases} ||z_{\text{text}} - z_{\text{image}}||_2^2 & \text{(positive pair)} \\ \max(0, W_m - ||z_{\text{text}} - z_{\text{image}}||_2)^2 & \text{(negative pair)} \end{cases}$$

- Here, W_m is a margin that separates unrelated pairs.

Joint Space Optimization

- The goal is to minimize the loss function to align related pairs in the joint space:

$$\min_{\theta_{\text{text}},\theta_{\text{image}}} \sum_{(T,I)} L_{\text{con}}(g_{\text{text}}(T), g_{\text{image}}(I))$$

- θ_{text} and θ_{image} are the parameters of the projection functions for text and image.
- The optimization ensures that related data points from different modalities are close in the joint space.

Co-Learning vs. RAG in Multimodal AI

Definitions:

- **Co-Learning**: Integration of multiple modalities (e.g., text, image, audio) to enhance cross-modal learning and improve task performance.
- **RAG (Retrieval-Augmented Generation)**: A framework that retrieves relevant external information to improve generation, incorporating it into a generative model.

Contd...

Similarities:

- **Information Fusion**:
  - Co-learning fuses information from multiple modalities.
  - RAG fuses retrieved external knowledge with generative models.
- **Cross-Modal and Knowledge Integration**:
  - Co-learning integrates data from different modalities (text, image, audio).
  - RAG integrates external knowledge as an additional source of information.
- **Task Augmentation**: Both enhance the task by leveraging complementary sources of information.
- **Handling Ambiguity**: Both methods help resolve incomplete or ambiguous data using external information (RAG) or another modality (Co-Learning).

Enrichment Approaches in Multimodal Learning

Definition: Enrichment approaches involve enhancing the representation space of unimodal models with additional modalities as input. This can provide more information and structure compared to prior unimodal representations.

Examples:

- **Video to Text Transfer:** Training a joint video-based multimodal model and transferring it to text-only classification tasks.
- **Image Classification with Knowledge Graphs:** Structuring the representation space for image classification by integrating knowledge graphs.

Contd...

- **Pros:**
  - Easier tuning and iteration in discriminative tasks.
  - Enables more fine-grained interaction design in the latent space.
  - Prioritizes additive interactions between modalities for better understanding of changes.

- **Cons:**
  - Difficult to determine and understand the learned interactions.
  - Cases exist where multimodal training leads to strong unimodal classifiers without enforcing cross-modal dependencies.
  - Challenges in interpreting missing data in some multimodal setups.

Translating Unimodal Data into Another Modality (Hallucination)

Definition: This approach involves translating unimodal data into another modality or latent space, learning a joint multimodal space via **"hallucination"**. It forces the latent space to handle and recreate another modality, even with limited training data.

Examples:

- **Vokenization:** Mapping contextualized text embeddings into images.
- **Image to Semantic Space:** Projecting image embeddings into semantic word embedding spaces.
- **Language Hallucination:** Translating language into hallucinated video and audio modalities.

Contd...

- **Pros:**
  - More flexible and easier to design by defining a reconstruction loss on the modality's input space.
  - Provides easier visualization of the latent space by reconstructing modalities or measuring reconstruction losses.

- **Cons:**
  - These methods may suffer from sample inefficiency due to the challenge of reconstructing high-dimensional data [**?**].
  - Potential to hallucinate false correspondences in the data, making reconstruction unreliable.

Kernel-Based Data Fusion

- Kernel-based data fusion combines information from different data modalities using kernel functions.
- Modality examples: text, images, audio, video, and sensor data.
- Aim: Create a unified representation by combining kernels from various modalities.

Kernel Functions for Each Modality

- Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m$ represent feature spaces from m different modalities.
- A kernel function $k_i(x_i, x_i')$ maps the input data from modality i into a high-dimensional space where learning occurs.
- For modality i, the kernel is defined as:

$$k_i(x_i, x_i') = \phi_i(x_i)^\top \phi_i(x_i')$$

where ϕ_i is a feature mapping for modality i.

Composite Kernel for Data Fusion

- The goal of data fusion is to combine kernels from different modalities into a single composite kernel.
- The composite kernel $K(x, x')$ is typically a weighted sum of individual kernels:

$$K(x, x') = \sum_{i=1}^{m} \alpha_i k_i(x_i, x_i')$$

- Here, $\alpha_i \geq 0$ are weights that determine the contribution of each modality i to the fusion.
- The weights α_i can be learned from data.

Joint Kernel Matrix

- The kernel matrix K for a dataset with n samples is given by:

$$K = \sum_{i=1}^{m} \alpha_i K_i$$

  where $K_i \in \mathbb{R}^{n \times n}$ is the kernel matrix for modality i, and $K \in \mathbb{R}^{n \times n}$ is the final fused kernel matrix.

- Each element of K represents the similarity between two samples based on all modalities.

Learning the Weights α_i

- The weights α_i can be optimized using a learning objective that maximizes the model's performance.
- One common approach is to minimize a regularized loss function:

$$\min_{\alpha} \sum_{j=1}^{n} \mathcal{L}(y_j, f(K(x_j, \cdot))) + \lambda \|\alpha\|_2^2$$

where $\mathcal{L}$ is the loss function (e.g., squared loss or hinge loss), and λ is a regularization parameter.

- The regularization term $\|\alpha\|_2^2$ ensures that the weights are not too large, leading to a balanced contribution from each modality.

Multimodal Learning Objective

- The model learns from the combined kernel by minimizing the empirical risk:

$$\min_{\mathrm{w},b} \frac{1}{n} \sum_{j=1}^{n} \mathcal{L}(y_j, f(\mathrm{w}^\top \Phi(x_j) + b))$$

  where $\Phi(x_j)$ represents the combined feature map across all modalities.
- f is the function learned from the combined kernel, and w are the model parameters.

Summary

- Kernel-based data fusion allows the integration of multiple data modalities through a unified kernel representation.
- Composite kernels provide a flexible approach to combine heterogeneous data sources.
- Weights for each modality can be learned to maximize predictive performance.

Introduction to Multiple Kernel Learning (MKL)

- MKL is a framework combining multiple kernels for better classification and task performance.
- It allows flexibility in learning different aspects of data by using different kernels.
- Applications: Visual Object Recognition, Text Classification, etc.

MKL for Binary Classification

Given $D = \{x_1, \ldots, x_n\}$ as training data, and $y \in \{-1, +1\}$ as labels, MKL combines s base kernels, $K_j(x, x')$, with coefficients β_j:

$$\kappa(x, x'; \beta) = \sum_{j=1}^{s} \beta_j \kappa_j(x, x')$$

$$K(\beta) = \sum_{j=1}^{s} \beta_j K_j$$

The objective is to find the optimal β by minimizing the regularized classification error:

$$\min_{\beta \in \Delta, f \in H_\beta} \frac{1}{2} ||f||^2_{H_\beta} + C \sum_{i=1}^{n} \max(0, 1 - y_i f(x_i))$$

MKL Optimization

The problem is expressed as a convex-concave optimization:

$$\min_{\beta \in \Delta} \max_{\alpha \in Q} 1^\top \alpha - \frac{1}{2}(\alpha \circ y)^\top K(\beta)(\alpha \circ y)$$

where:

- α are the dual variables.
- $\circ$ denotes the Hadamard product.
- $Q = \{\alpha : \alpha_i \in [0, C]\}$.

Regularization in MKL

- L1 Regularization: Sparsity-inducing

$$\Delta_1 = \left\{ \beta \in \mathbb{R}_+^s : \sum_{j=1}^{s} \beta_j = 1 \right\}$$

  Results in the elimination of irrelevant kernels.

- Lp Regularization: Smooth kernel combination

$$\Delta_p = \left\{ \beta \in \mathbb{R}_+^s : ||\beta||_p \leq 1 \right\}$$

Sequential Minimal Optimization (SMO) for MKL

MKL can be solved using SMO for faster optimization. The objective function is:

$$\max_{\alpha \in Q} \left(1^\top \alpha - \frac{1}{8\lambda} \left(\sum_{j=1}^{s} (\alpha \circ y)^\top K_j (\alpha \circ y) \right)^{\frac{2}{q}} \right)$$

- SMO updates the kernel weights β_j iteratively.
- This ensures computational efficiency.

Application: Visual Object Recognition

- MKL has been extensively used in object recognition.
- Combines kernels based on color, texture, shape, etc.
- Allows a more robust model than using a single kernel.