Adaptive Fusion Techniques for Effective Multimodal Deep Learning

by

Gaurav Sahu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Effective fusion of data from multiple modalities, such as video, speech, and text, is a challenging task due to the heterogeneous nature of multimodal data. In this work, we propose fusion techniques that aim to model context from different modalities effectively. Instead of defining a deterministic fusion operation, such as concatenation, for the network, we let the network decide how to combine given multimodal features more effectively. We propose two networks: 1) Auto-Fusion network, which aims to compress information from different modalities while preserving the context, and 2) GAN-Fusion, which regularizes the learned latent space given context from complementing modalities. A quantitative evaluation on the tasks of multimodal machine translation and emotion recognition suggests that our adaptive networks can better model context from other modalities than all existing methods, many of which employ massive transformer-based networks.

Acknowledgements

I would like to thank all the people who made this thesis possible. I am fortunate to have Prof. Olga Vechtomova as my advisor, and am very grateful for all her advice, and support. She has always been supportive of my ideas, and gave me freedom to pursue my interests independently. I thank her for believing in me, and being patient throughout my Masters program.

I would also like to thank my committee members, Prof. Robin Cohen, and Prof. Pascal Poupart, for agreeing to read my thesis, and provide feedback.

A special token of thanks goes to Prof. Dara Lane, who has been selflessly supporting my creative half behind the scenes.

Finally, I would like to take this opportunity to thank everyone else who has helped me reach where I am until now.

Dedication

This is dedicated to my parents – the true constants of my life. They have blessed me with their unconditional love, no matter the circumstances. Nothing would have been possible without their constant support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multimodal deep learning is an active field of research where, for a single event, one is presented with information across multiple modalities, such as video, speech, and text, so that they may be combined to gain a better contextual understanding. Combining, or more precisely, *fusing* information from multiple modalities is, thus, a vital step for any multimodal task. Better fusion results in richer combined representation of inputs from multiple modalities. However, multimodal data is highly heterogeneous, making fusion a challenging task. Moreover, the extent to which signals from complementing modalities are helpful for a downstream task is not always clear. In such a case, using a convoluted fusion method would only add to computation with little guarantee of improvement in the performance. On the other hand, using a simple fusion method may not capture the context even if sufficient information exists. Therefore, for a challenging NLP task such as machine translation, where we need to combine the unimodal features and extract complementary signals from other modalities, we need to model context better, while avoiding significant computational overhead.

The most common fusion technique used in the literature is concatenation, which involves the concatenation of representations from all the modalities. However, this results in a shallow network [91], and the network focuses more on learning intra-modal features, instead of learning inter-modal features. Later, Zadeh et al. [140], proposed Tensor Fusion Network (TFN), in which, the unimodal, bimodal, and trimodal interactions are modelled using a 3-fold Cartesian product. TFN is shown to be better than simple concatenation; however, it imposes high computational requirements as information from all the modalities is projected as-is, without any prior information extraction. The computational overhead grows exponentially as the dimensions in unimodal features increase. Liu et al. [79], then, proposed a low-rank multimodal fusion technique (LMF) to address the previous problem. Such fusion techniques are useful but often result in a complex architecture with much computation. Moreover, the aforementioned fusion methods focus only on combining individual unimodal features rather than combining *and* extracting useful information simultaneously.

1.1 Contributions

In this thesis, we propose adaptive fusion techniques that allow the model to decide "how" to combine multimodal data for an event in the best possible manner. The first technique, Auto-Fusion, aims to compress multimodal information while preserving as much meaning as possible. The second technique, GAN-Fusion, employs an adversarial network that regularizes the learned latent space for a target modality (text, in our case) according to information presented by the remaining complementary modalities. Since our models are generic, the need to specify a pre-determined fusion operation such as concatenation or Cartesian product is alleviated, and this further incentivizes the network to model multimodal interactions by itself. Moreover, our techniques involve simple components such as linear transformation layers, thereby checking unnecessary computational load, when compared to a heavy component such as the transformer network [127].

We evaluate our models on three benchmark datasets: 1) the How2 dataset [111] with multimodal input for English-Portuguese translation 2) the Multi30K dataset [37], which contains parallel corpora for multimodal machine translation, and 3) the IEMOCAP dataset [14] which contains multimodal data for emotion detection. The quantitative evaluation shows that our models outperform the existing state-of-the-art methods in terms of BLEU scores [96] for machine translation and Precision, Recall, and F1-score for emotion recognition.

Fusion is a vital core aspect of multimodal deep learning. Addressing it will pave ways for a more thorough and robust integration of multimodal communication into embodied or non-embodied AI systems. It may also prove helpful in improving existing systems by replacing rule-based decision-making modules with an adaptive and more robust fusion module. For instance, Simsensei [34], the first application of multimodal learning for healthcare, can benefit immensely from better fusion modules. It also holds exciting avenues for development in the future, like a better understanding of unimodal contribution. We hope the lessons learned in this work serve as a guiding light and bolster research in multimodal fusion.

The rest of the thesis is structured as follows: Chapter 2 introduces required background, Chapter 3 covers relevant work, Chapter 4 discusses the proposed methodologies, Chapter 5 describes the experimental setup and quantitative results, Chapter 6 shows analysis, and Chapter 7 contains our concluding remarks.

Chapter 2

Background

Multimodal Deep Learning (MMDL) involves relating features from multiple modalities – the different sources of information – such as images, audio, and text. The goal is to learn a shared representation of the inputs from different modalities, which a neural network may exploit to make intelligent decisions for the desired task. The earliest attempts to develop such a system involve the work by Ngiam et al. [91], where sparse RBMs and deep autoencoders were employed to demonstrate the improvement of introducing information from different sources. Section 3 covers various such frameworks in much more detail.

The subsequent sections in this chapter are divided as follows: Section 2.1 - 2.7 briefly discuss different components from the machine learning and deep learning literature. They are recommended for a thorough understanding of the proposed work; however, if the reader is already comfortable with the concepts and terminologies discussed, they can skip to Section 2.8, which briefly touches upon multimodal frameworks in the literature.

2.1 Multi-layer Perceptron (MLP)

A multi-layer perceptron (MLP) is the simplest type of feedforward artificial neural network (ANN). For the same reason, they are also sometimes known as “vanilla” neural networks. The simplest MLP has three layers: 1) an input layer, 2) a hidden layer, and 3) an output layer. MLPs are fully-connected, i.e., each node in one layer is connected to every node in the subsequent layer with a certain weight. The nodes, also known as *neurons*, in those layers except the input layer, use a non-linear activation function. This means that a non-linear transformation function exists in every node that maps the weighted inputs to each

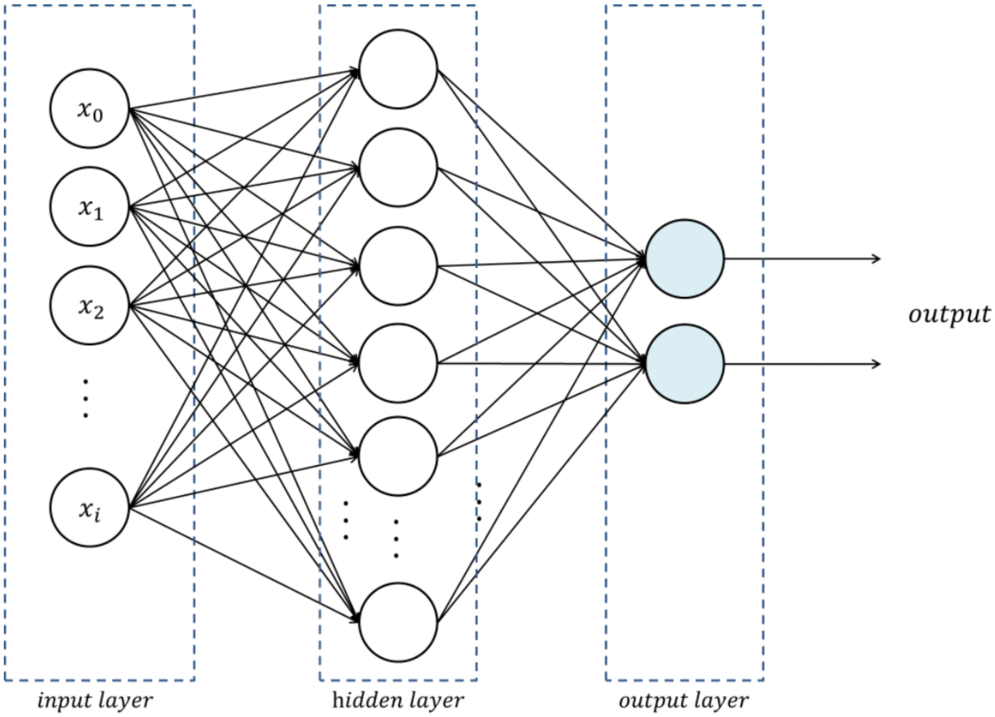


Figure 2.1: Architecture of a multi-layer perceptron (MLP) [1].

neuron's output. A non-linear activation function, such as sigmoid or ReLU [90], allows the network to model complex data distributions using a lesser number of neurons, thereby also allowing them to distinguish between data that is *not* linearly separable. Figure 2.1 shows the different components of an MLP.

The learning process of an MLP is straight-forward – it learns by updating the connection weights based on the accumulated error difference between generated and expected output, after processing a single data sample or a batch of data samples. This process is carried out through the back-propagation algorithm [107], which is widely used in the supervised-learning paradigm. It is also simply known as *backprop*. Backprop computes the gradient of the network with respect to individual connection weights, given a loss function, such as the mean-squared error (MSE) loss.

Theoretically, an MLP with a single hidden layer and sufficient hidden nodes is proven to be a universal approximator [56, 32]. This means that it can approximate any continuous function mapping $f : \mathbb{R} \to \mathbb{R}$, arbitrarily closely and is, therefore, able to model any data distribution. However, empirical results suggest otherwise, and recurrent neural networks

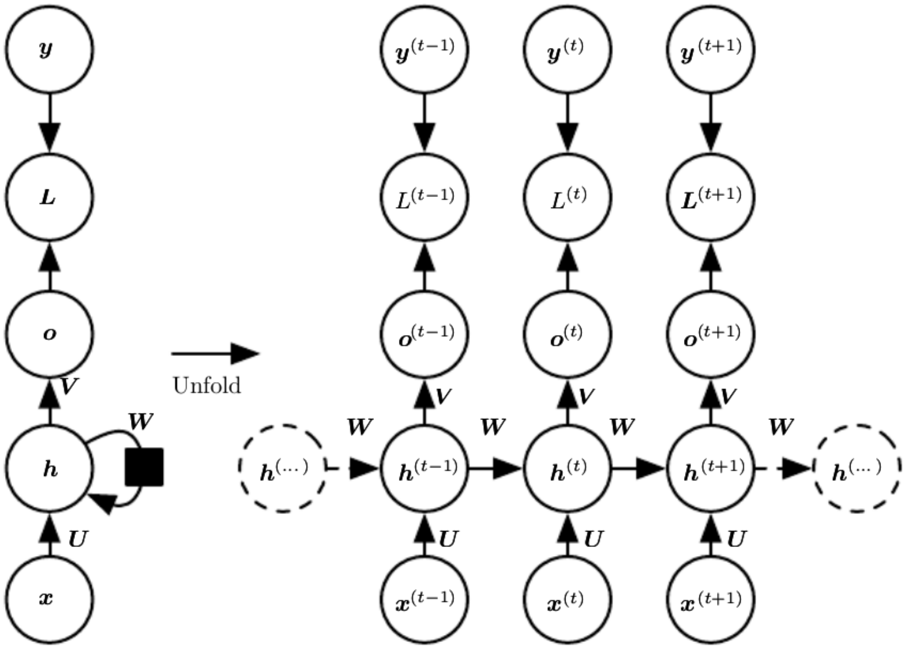


Figure 2.2: Computational graph of a recurrent neural network (RNN). Input sequence $\boldsymbol{x}$ is mapped to its corresponding output sequence $\boldsymbol{o}$. L represents the loss function, which measures the distance between $\boldsymbol{o}$ and $\boldsymbol{y}$, the target sequence [42].

(RNNs) have been proven to be more suitable for more convoluted tasks, such as text-generation, and time-series modelling.

2.2 Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a class of ANN where the nodes are connected to form a directed graph along a temporal sequence. Their internal state, better known as "memory," allows them to process inputs of variable length; hence, they are more suitable for complex tasks such as speech recognition, and time-series analysis.

A basic RNN is simply a network of *neurons* arranged into successive layers. The connections are uni-directional, and every neuron has a real-valued activation function. Figure 2.2 shows the computational graph of a simple RNN along with its different components.

The learning process of an RNN in a supervised, discrete temporal setting involves processing real-valued input vectors. At each time step, the non-input nodes apply appropriate non-linear activation on the weighted inputs, and the final generated output is compared to the expected output value for loss calculation. Standard optimization methods for training RNNs involve gradient-based "backpropagation through time," or simply BPTT, which is a generalization of the backprop algorithm for feed-forward networks [89, 106, 131]. A significant problem with using such gradient-based methods for training basic RNNs is the vanishing/exploding gradient problem [53], where the error gradient vanishes/explodes exponentially quickly with increasing time steps. Many variations of RNNs were proposed in order to mitigate this issue. We discuss one such relevant variant next in this chapter.

2.3 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) [54, 40] is a unique RNN architecture which, unlike standard RNNs, consists of feedback connections. An LSTM unit consists of three different gates: 1) the input gate, 2) the output gate, and 3) the forget gate. The three gates are enclosed within a "cell" that remembers relevant information over different time intervals, while the three gates regulate the flow of information into and out of the cell. The gating mechanism helps the LSTMs to capture distant temporal dependencies.

Figure 2.3 shows an overview of an LSTM cell. We enumerate the equations for those components below for the sake of completeness:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2.1a}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2.1b}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.1c}$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \tag{2.1d}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{2.1e}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{2.1f}$$

Here, $x_t \in \mathbb{R}^d$ is the input vector to the LSTM unit, $f_t \in \mathbb{R}^d, i_t \in \mathbb{R}^h$, and $o_t \in \mathbb{R}^h$ are the outputs of the forget gate, the input gate, and the output gate, respectively. $h_t \in \mathbb{R}^h$ is the hidden vector of the LSTM unit (also known as the output of the LSTM unit,) $\tilde{c}_t \in \mathbb{R}^h$ is the cell input activation vector, $c_t \in \mathbb{R}^h$ is the actual cell state vector. Finally, $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times d}$, and $b \in \mathbb{R}^h$ are the weights and bias matrices, respectively, to be learned during the training process and σ represents the activation functions.

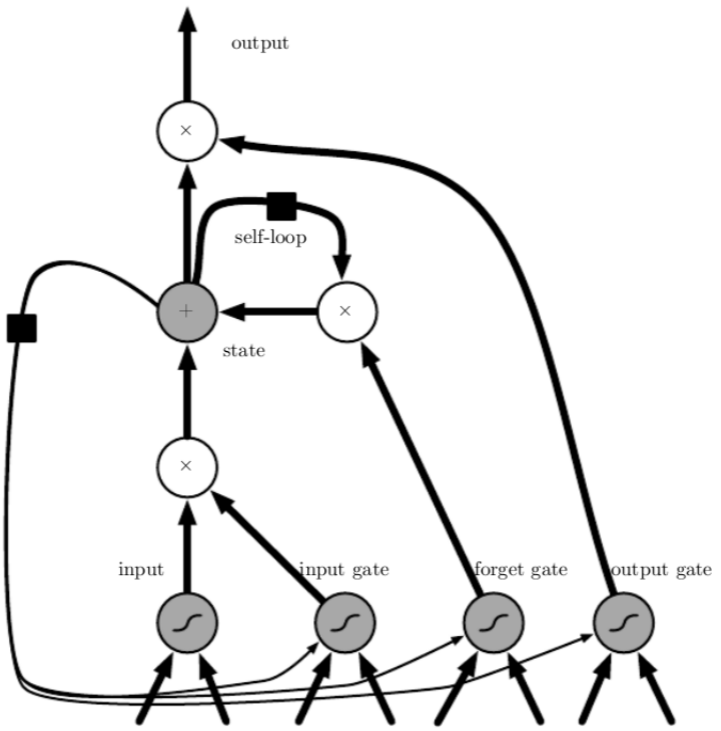
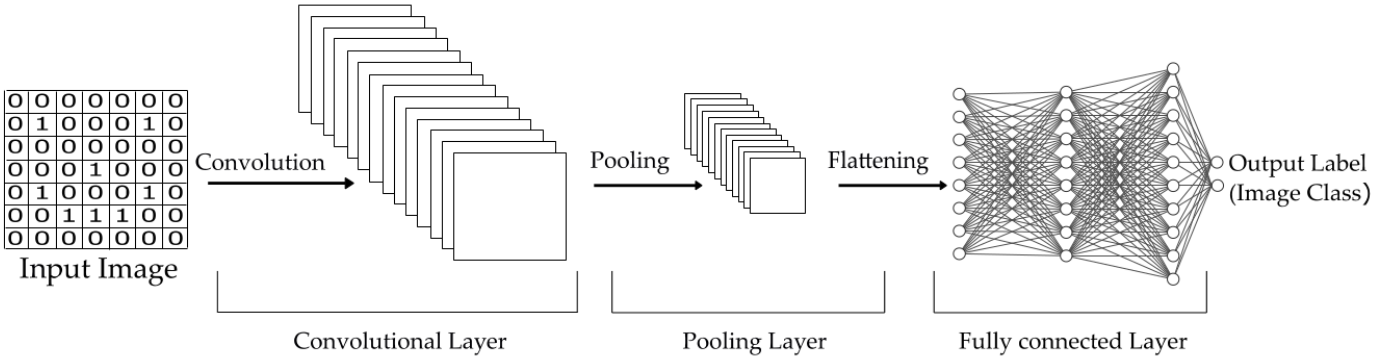


Figure 2.3: General overview of an LSTM cell. Multiple cells are connected recurrently to each other, unlike through hidden units in vanilla RNNs. First, an input feature is obtained through a regular neuron, and its value is accumulated based on whether the sigmoid gate allows for it. The state unit has a linear self-loop whose weight is controlled by the forget gate. The output gate controls the final output of the cell. The black box in the self-loop denotes delay of one time-step [42].

LSTMs were initially introduced to counter the vanishing gradient problem in RNNs discussed earlier because they allow some gradients to flow unchanged. However, they still suffer from the exploding gradient problem.

Since their introduction, LSTMs have revolutionized many application domains with their astounding improvement. They have been extensively employed for various tasks, such as speech recognition, language modelling, and improved machine translation. LSTMs have also been combined with other types of neural networks, such as convolutional neural networks (CNNs), to improve automatic image captioning [128].



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figure 2.4: Overview of a Convolutional neural network (CNN) [11].

2.4 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are a class of ANNs most popularly used for visual analysis. In very simple terms, CNNs are regularized MLPs. As the name suggests, CNNs use the convolution operation in at least one of their layers instead of general matrix multiplication. While MLPs are prone to over-fitting due to their fully-connected nature, CNNs use small and simple patterns to learn bigger, more complex patterns in data. This results in lesser connections and complexity.

A CNN has three types of layers: 1) an input layer, 2) an output layer, and 3) hidden layers. The hidden layers consist of a series of convolution operations, which convolve with an operation such as multiplication or dot product. These layers are commonly followed by additional convolutional layers, such as multiple pooling and fully-connected layers. The pooling layers reduce data-dimension by reducing outputs from a group of data points to a single value. Different pooling operations result in different types of features. *Global pooling* [27, 71], *max pooling* [133, 26], and *average pooling* [86] are the most common pooling operations employed. Figure 2.4 shows an overview of a simple CNN.

Even though CNNs were primarily introduced in the computer vision community, they have been extensively explored for many natural language processing (NLP) tasks [28]. For instance, they have shown promising results in tasks such as semantic parsing [46] and text classification [66]. They are also being explored in combination with RNNs to process multimodal data [128].

2.5 Generative Adversarial Network (GAN)

Generative adversarial networks (GANs) [43] are a class of deep generative models, which take a slightly different approach to generate novel data samples. A GAN can generate new data samples whose characteristics match the dataset it was trained on. For instance, a GAN trained on a dataset of anime character images can generate novel anime characters [20]. GANs were originally introduced as an unsupervised algorithm for synthesizing realistic images that looked authentic, at least superficially. However, they have been successfully adapted for semi-supervised learning [110], fully supervised learning [60], and even reinforcement learning [52].

A GAN consists of two main components: 1) a generator, and 2) a discriminator. The task of the generator is to generate (fake) candidate samples, while the discriminator tries to guess if a given sample is real or fake. The generator eventually learns a latent space mapping to the given data distribution. Formally, the objective of the generative network is to increase the error rate of the discriminator, i.e., generate images such that the discriminative network is unable to determine if the generative network generated the image or it was a part of the real data distribution, thereby, "fooling" the discriminator. The discriminator is usually a binary classifier with a CNN, and the generator is generally a deconvolutional network. GANs are trained using backprop, and the overall objective function of a GAN, V, is given by Equation 2.2, where D and G represent to the discriminator and the generator networks, respectively; $\mathbf{z}$ represents the latent variable, and $\mathbf{x}$ represents a data sample. As evident from the equation, it uses cross-entropy as the loss function, which may sometimes lead to a vanishing gradient problem. To tackle this issue, Mao et al. [83] proposed an alternative objective function using least squares. Figure 2.5 shows an overview of vanilla GAN architecture.

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z}))] \tag{2.2}$$

GANs have been successfully applied to many applications, such as modding video games [130], motion analysis in video [129], and super-realistic image generation [64], to name a few. They have even been employed for many text generation tasks such as text style transfer [61, 135]. Despite the high-quality generation, GANs suffer from "mode-collapse" problem, wherein they fail to capture entire modes in the real data distribution. For instance, a GAN trained on the MNIST dataset – a collection of handwritten digits from one to ten – might neglect a subset of digits from its output. Many solutions have been proposed to tackle this issue [74, 73, 78]. Some of these solutions also lay the foundation of future directions for the proposed work in this thesis.

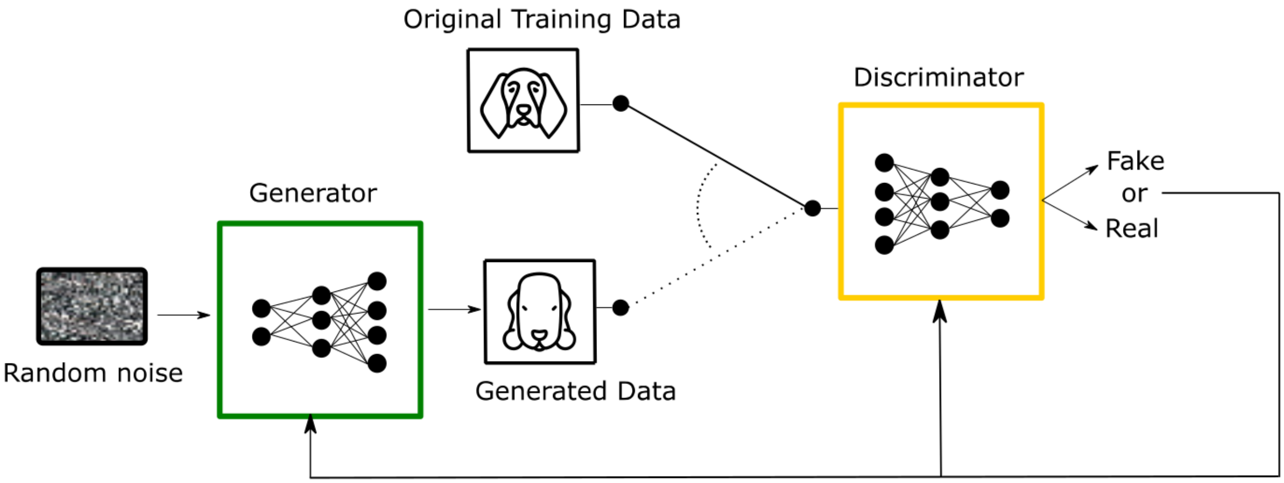


Figure 2.5: Overview of a Generative Adversarial Network (GAN) [11].

The subsequent sections touch upon the application of the architectures discussed until now. For instance, how embeddings are learned for data from different modalities (Section 2.6), and how to extract meaningful information from raw data for a given task (Section 2.7 – 2.8).

2.6 Embeddings

In the context of machine learning, embeddings refer to a mapping of discrete variables to a vector in continuous space. These vectors partially represent the semantics of the raw input data, which otherwise may not be easily comprehensible to a neural network. For instance, training a neural network to classify an image into one of the target classes using just raw pixels is not plausible without first transforming it into an embedding space. We discuss embeddings for inputs from different modalities in the following subsections.

2.6.1 Word Embeddings

In the context of NLP, word embeddings are a vector representation of all the words in the vocabulary. Most techniques to learn word embeddings are unsupervised in nature; however, many supervised and semi-supervised techniques to learn word embeddings have been proposed. The embeddings are learned such that words appearing in similar contexts

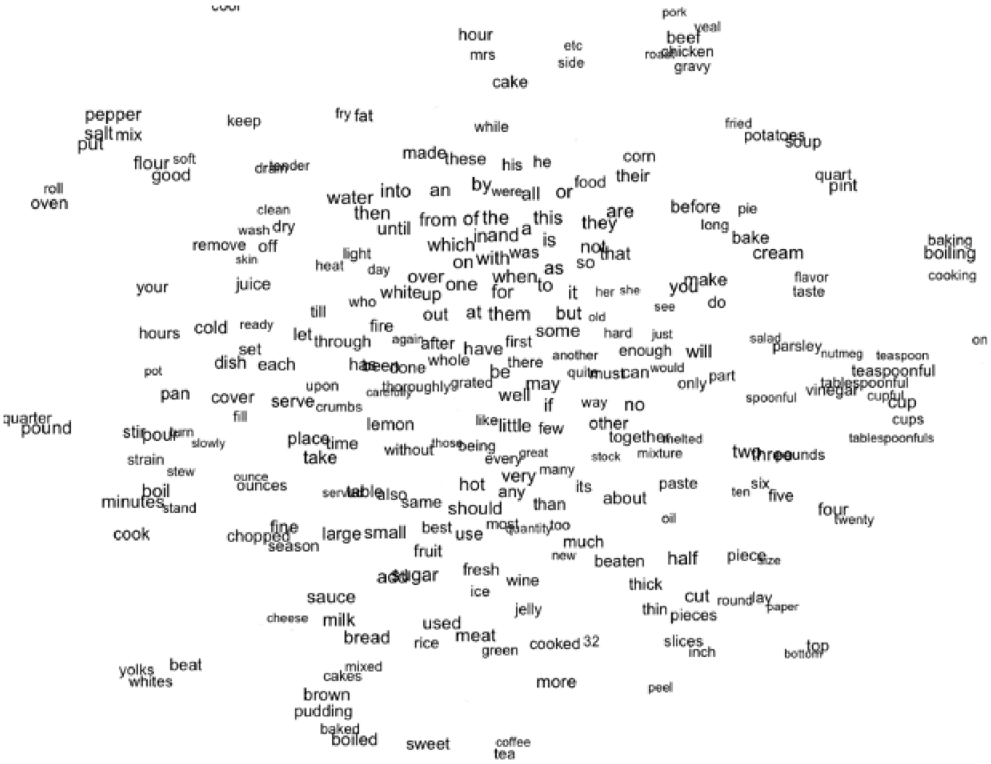


Figure 2.6: Mapping of words in the embedding space. It can be observed that similar words such as “coffee,” and “tea” are mapped closer to each other. [2]

are close to each other. For instance, words “apple” and “mango” will appear close to each other as they belong to the same family of entities – fruits. Figure 2.6 shows such a plot. Different techniques to learn word embeddings include Word2vec [85], Glove [100], ELMo [101], FastText [62], Skip-thoughts [69], Quick-thoughts [80], InferNet [29], and Google’s universal sentence encoder [17].

2.6.2 Speech Embeddings

Similar to word embeddings, speech embeddings are a rich representation of sound waves. They, too, are learned such that sonically similar sounds will end up near each other in the embedding space. Inspired by Word2vec [85], Chung et al. [24] proposed Speech2Vec, which learns acoustic embeddings based on neighbouring acoustic regions. It splits an audio segment by word, learns a fixed embedding for that audio segment. Later, Haque et al. [48] proposed to learn similar embeddings on a sentence-level.

Researchers are still exploring the benefits of using speech embeddings over classical features like Mel-frequency cepstral coefficients (MFCC), or zero-crossing rate (ZCR). However, they have been shown to contain richer information than plain word-embeddings [24] and are being used to learn better cross-modal embeddings [25].

2.6.3 Visual Embeddings

Pre-trained embeddings for visual modality aim to learn a meaningful representation of images; however, employed methods to learn such embeddings are not entirely unsupervised. Most pre-trained visual embeddings used are simply feature-vectors from the hidden layers of a (convolutional) neural network trained on a specific task. For instance, the most popular choices for obtaining pre-trained image embeddings, such as, Inception [123], VGG [114], SqueezeNet [59], and DeepLoc [70], are all trained on an image classification task. Embeddings from such pre-trained networks are now being used for visual grounding in many NLP tasks, such as question answering (QA) [6, 15, 139, 136, 132].

Learning a joint representation of visual and other modalities is still an active field of research. We will discuss more about such frameworks in Section 2.8 and Chapter 3.

2.7 Unimodal Frameworks

This section discusses popular unimodal frameworks for each modality and should serve as a primer for multimodal frameworks to be discussed later in this thesis. Furthermore, proposed multimodal techniques in this thesis also borrow from these frameworks.

2.7.1 Text

NLP includes a wide range of subtasks, such as grammar induction, lemmatization, parsing, word segmentation, text classification, machine translation, natural language generation, and natural language understanding. Techniques proposed to solve these subtasks can be broadly divided into two categories: rule-based and statistical. We will focus on the latter. More concretely, we will limit our discussion to neural network based natural language generation and understanding.

NLP practitioners have leveraged recent advancements in deep learning. RNNs have set performance records on numerous NLP tasks, outperforming previous rule-based approaches significantly [45, 121, 63, 8, 81, 41]. They are still widely employed for a variety of NLP tasks, despite the introduction of supposedly more powerful transformer networks [127]. One reason for preferring RNNs over transformers is that transformers are very heavy architectures imposing much computational overhead. In fact, reducing the parameters in transformers is an active field of research currently [112].

In the following subsections, we will discuss two pieces from NLP literature most relevant to the proposed methodologies – sequence-to-sequence models and attention mechanism.

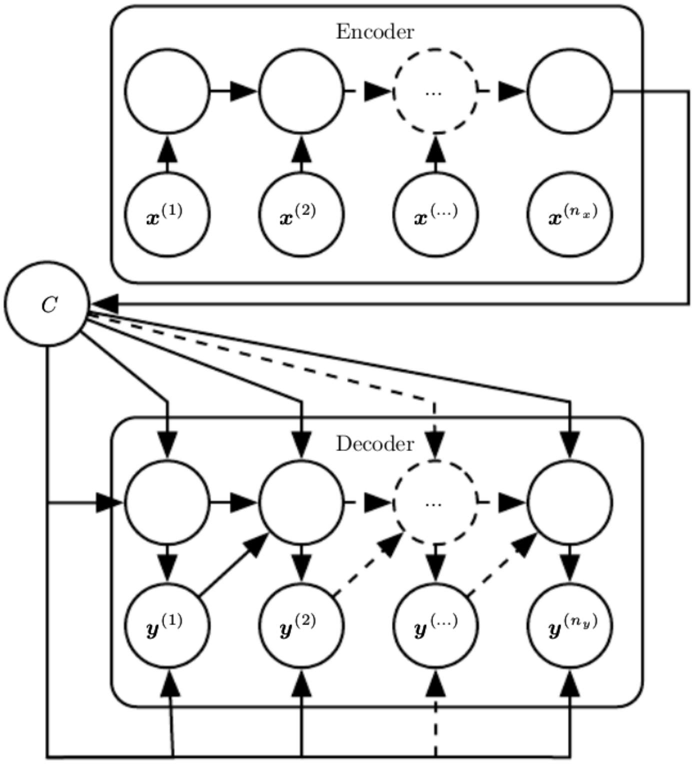


Figure 2.7: Architecture of a Sequence-to-Sequence (Seq2Seq) network. The network learns to generate the output sequence $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n_y)}$ given an input sequence $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n_x)}$. The final hidden state of the encoder, C, is passed through the decoder RNN to generate a fixed-length output sequence. Here, n_y and n_x denote the number of time steps in the output and input sequence, respectively [42].

Sequence-to-Sequence (Seq2Seq) models

Since their introduction, the Sequence-to-Sequence (Seq2Seq) model [121] has been widely adopted for a variety of generation tasks. Seq2Seq refers to the family of generative models that take as input, a sequence of *fixed*-length and outputs a *fixed*-length sequence as well. The input sequence could be a single sentence or a complete document, with words being treated as individual *tokens*, or a word could be treated as an individual sequence as well. In the latter case, characters serve as the *tokens*. If the input and target output sequences are exactly the same, the model is known as an **autoencoder**.

A vanilla Seq2Seq model has two primary components: an **encoder**, and a **decoder**. Both the components comprise an RNN, such as LSTM (to avoid the vanishing gradient problem). Given an input sequence $(x_1, x_2, x_3 \cdots x_T)$, the encoder RNN first learns a vector/hidden representation of the fed input. The hidden representations are computed iteratively, for every time-step, using the Equation 2.3. As evident from the equation, the hidden vector for a given time-step is obtained by applying appropriate weights to the previous step's hidden vector and the current time-step's input.

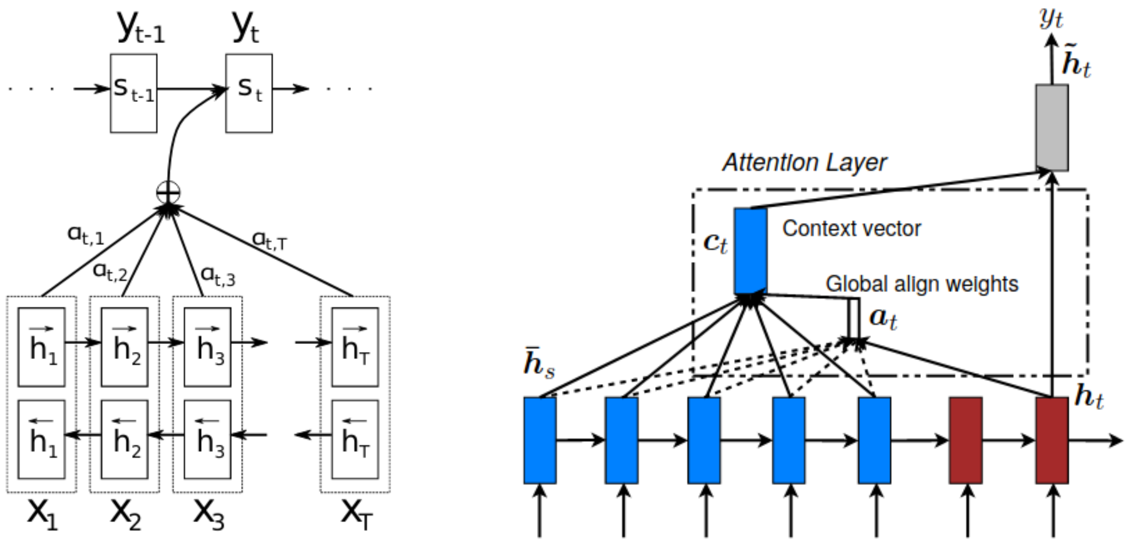
$$h_t = \sigma(W^{(hx)}x_t + W^{(hh)}h_{t-1}) \tag{2.3}$$

The decoder RNN, on the other hand, does the reverse of this process by generating an output sequence of tokens using the vector/hidden representation learned by the encoder RNN. More concretely, it generates an output sequence $(y_1, y_2, y_3 \cdots y_{T'})$, whose length may not be the same as the input sequence's length. Tokens for every time-step are calculated by iteratively using Equation 2.4. It is notable in the equation that h_t represents *decoder* RNN's hidden state from the last time-step. It is initialized by the encoder RNN's hidden vector for the first time-step. Also, the *softmax* operation is used to generate a probability vector, which is then used to find the next most probable token y_t in the output sequence. Figure 2.7 shows the complete architecture of the Seq2Seq model.

$$y_t = softmax(W^{(hy)}h_t) \tag{2.4}$$

The goal of a decoder RNN comprising of LSTM may be alternatively described as the estimation of the conditional probability $p(y_1, \ldots, y_{T'}|x_1, \ldots, x_T)$ given by Equation 2.5, where v is the vector representation of the input sequence $(x_1, \ldots, x_T)$ as learned by the encoder RNN.

$$p(y_1, \ldots, y_{T'}|x_1, \ldots, x_T) = \Pi_{t=1}^{T'} p(y_t|v, y_1, \ldots, y_{t-1}) \tag{2.5}$$



(a) Bahdanau attention [8]

(b) Luong attention [81]

Figure 2.8: Overview of Bahdanau and Luong attention mechanisms.(a) **Bahdanau attention**: The t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$ is generated by computing context vector, which is a weighted sum of the alignment vector α and the hidden vector h. (b) **Luong Global attention**: At each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

The Seq2Seq architecture was originally proposed for machine translation [121], but it has been adopted for numerous other tasks such as solving differential equations [72]. Very recently, Google released an open-domain chatbot, Meena [5], which employs Seq2Seq architecture with over 2.6 billion parameters.

Attention mechanism

A Seq2Seq model learns vector representation of the input sequence; however, if the input sentence is very long, the learned representation may not capture all the relevant information in the input sentence. Attention mechanisms were proposed to solve such issues [8]. The basic idea of the Attention mechanism is to pay more focus on specific parts in the input vector instead of learning a single vector for each entire sentence. It does so using

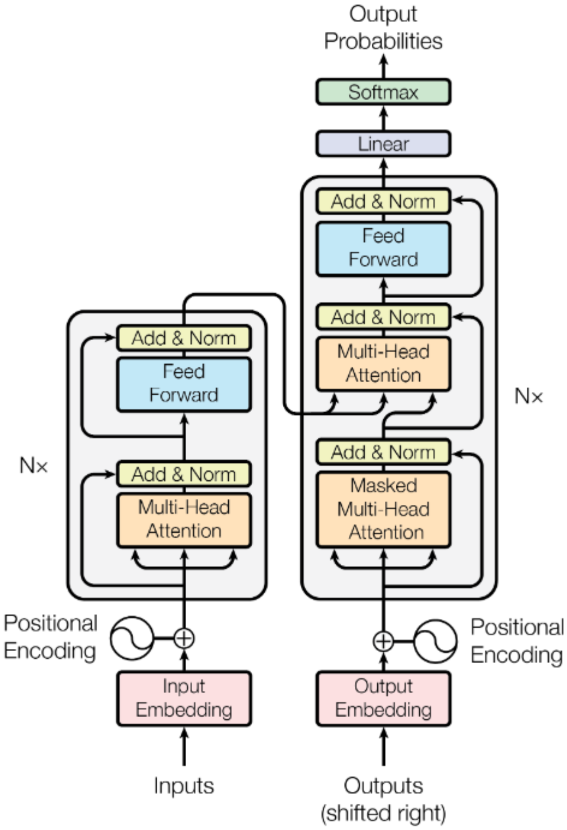


Figure 2.9: Model architecture of the Transformer network [127]

attention weights learned during the training process.

Multiple types of attention mechanisms have been proposed, but all of them have at least three primary components in common: 1) an alignment layer, 2) an attention weights matrix, and 3) a context vector. Figures 2.8 shows architecture diagrams of the two most famous attention mechanisms used in NLP. The transformer networks proposed recently also comprise of multiple self-attention heads [127]. Figure 2.9 shows the outline of a transformer network.

2.7.2 Audio

Before applying deep learning in the audio domain, techniques involved classic temporal and non-temporal audio features such as MFCC and ZCR. However, that changed drastically once deep generative networks were applied successfully for acoustic modelling. One of the first such networks were restricted Boltzmann machines (RBMs), and hidden Markov models (HMMs) [50]. More recently, Google released WaveNet [93], which established new benchmarks for speech recognition. It uses specially designed CNNs to encode raw audio. The next subsection discusses such CNN-based models in more detail.

CNN-based models

In this section, we will discuss WaveNet [93] and JukeBox [35], the two most successful CNN-based models for encoding audio. Both models generate raw audio indicating that one-second of generated audio clip involves thousands of prediction steps, like 44.1k for generating a 44.1kHz audio clip.

WaveNet: WaveNet is a deep generative model designed for raw audio, which can generate a more natural-sounding human voice. It reduced the performance gap between existing text-to-speech systems and human performance by a significant margin of more than 50% [93]. It is also able to synthesize realistic musical scores.

WaveNet uses PixelRNN [94] and PixelCNN [125] as its building block. The two networks were originally proposed to generate images one pixel at a time. They are entirely autoregressive in nature and generate pixels such that pixel-prediction at a given step is conditioned on previously predicted pixels. Moreover, they could not only generate images one pixel at a time but also one colour-channel at a time. These properties make them suitable for encoding raw audio waveforms, which have thousands of samples per second. The two-dimensional PixelNets had to be simply adapted to one-dimensional WaveNet.

JukeBox: JukeBox [35] is a deep generative model recently proposed by OpenAI to generate music at the audio level. It uses a quantization-based approach, VQ-VAE-2 [104], a simplified variant of VQ-VAE [126], to compress audio to a discrete audio space. The used VQ-VAE is hierarchical in nature with three levels and compresses a 44kHz audio by 8x, 32x, and 128x, respectively [35]. It then uses a stack of sparse transformers [22, 127] to ultimately generate raw audio. Figure 2.10 shows the complete architecture of JukeBox.

Music generated by JukeBox is more realistic than prior methods [1], but human-generated music is still significantly better. For instance, generated music still has low music coher-

[1] Many non-cherry picked samples can be found here: https://jukebox.openai.com/

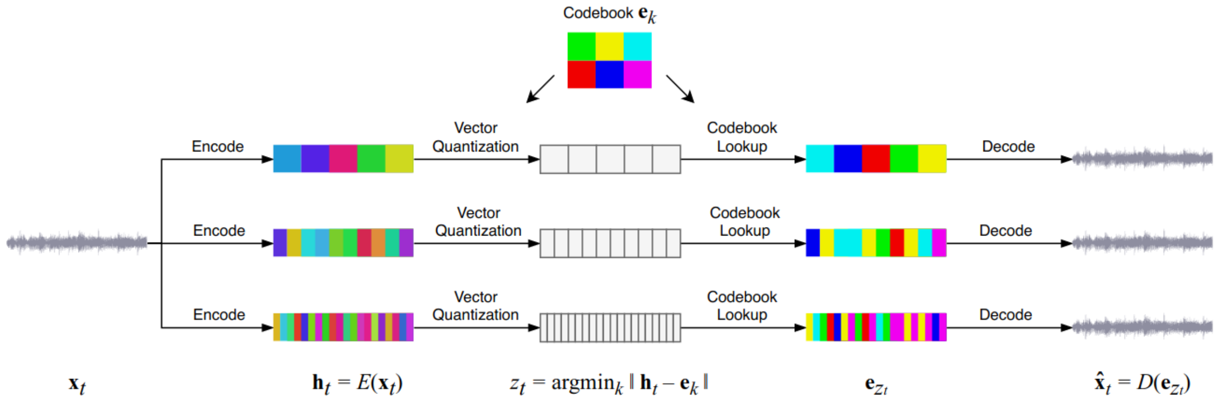


Figure 2.10: Model architecture of JukeBox. First, three separate VQ-VAEs with different temporal resolutions are trained. The top-most level learns the most abstract representation since it is encoding longer audio per token. Notably, audio can be reconstructed at any of the abstraction levels, where the bottom-most level generates the highest quality audio samples [35].

ence. Since the model uses heavy transformers and deep hierarchical VQ-VAEs, it poses immense computational requirements; it can take as long as 9hrs to generate a one-minute audio clip. Distilling such models is still an active field of research.

2.7.3 Vision

Computer Vision (CV) aims at understanding and automating the functioning of human visual systems. It includes methods for processing and understanding digital images by representing the high-dimensional data in images symbolically. CV also has many sub-domains such as scene reconstruction, object detection, motion estimation, 3D scene modelling, and image restoration [88].

Vision has perhaps benefited the most by the success of deep learning. Currently, CNNs serve as state-of-the-art for most CV tasks. The performance of these networks is even close to human performance in some cases [108]. In the subsequent subsections, we will discuss two types of models that have set benchmarks on different CV tasks.

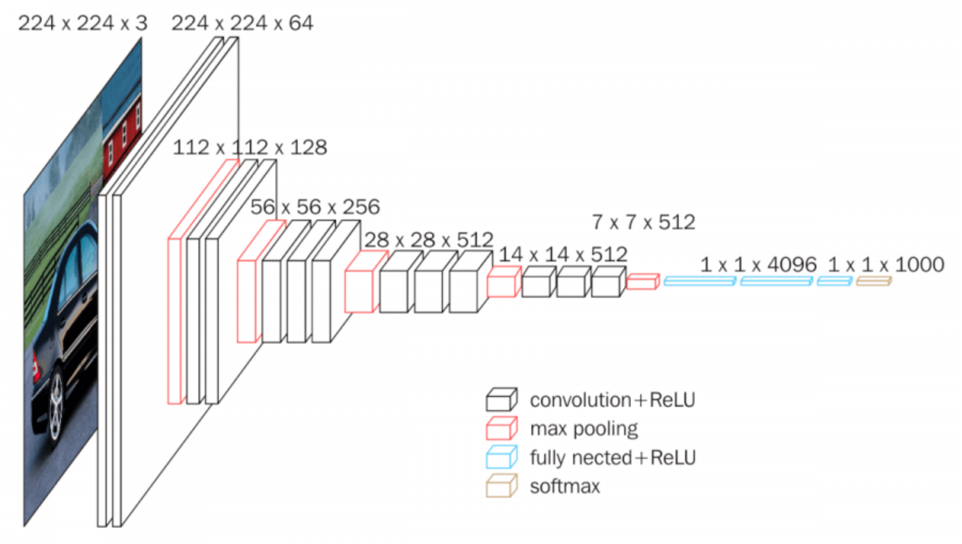


Figure 2.11: Overview of the VGG architecture [3].

CNN-based models

CNNs have revolutionized many tasks in CV by setting incredible benchmarks. We will briefly discuss the four most popular CNNs.

VGG [114]: VGG was one of the first deep neural networks to be proposed. It has a simple architecture, using convolutional layers whose size varied incrementally with depth. It has two variants: VGG16 and VGG19. The numbers 16 and 19 correspond to the number of weighted layers in the network, excluding dense and pooling layers. Figure 2.11 shows the complete VGG architecture.

ResNet [49]: Residual networks or ResNets were proposed slightly later than VGGs. They consist of residual connections in the network, making the architecture modular, i.e. it had a network within a network. The most popular variant of ResNet is the ResNet50, where 50 corresponds to the number of layers excluding pooling and fully connected layers. They showed a remarkable improvement over VGGs, and despite having more layers than VGGs, they needed much less memory (nearly five times).

Inception [122]: The Inception model was introduced around the same time as the VGG. However, unlike the VGG, it uses blocks with filters of *different* sizes, which are ultimately concatenated to extract features at various levels. It was further optimized by Szegedy et al. [123], but the underlying architecture remains the same. This model further reduced the computational requirements when compared to VGG and ResNet. Figure 2.12 shows the originally proposed Inception module.

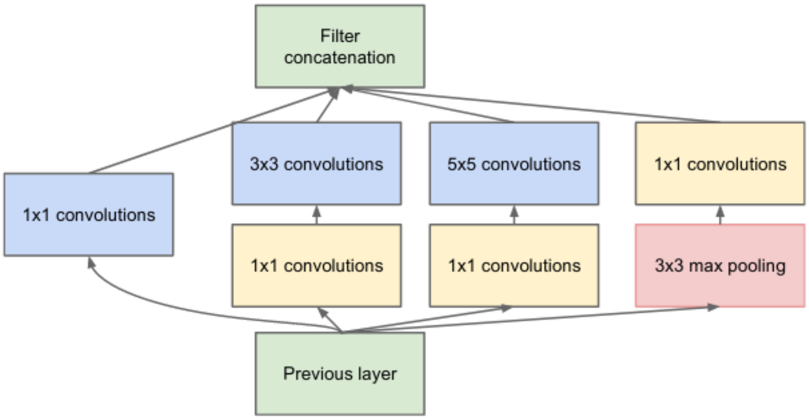


Figure 2.12: Overview of the Inception module with dimensionality reduction [122].

Xception [23]: The Xception model is very similar to Inception, except that it requires even lesser memory for computation. It optimizes convolutions by separating 2D convolutions in the Inception model with two 1D convolutions. The optimization allowed for performance gains on the task of image classification, as well as an increased capacity of the model.

GAN-based models

GANs have set benchmarks on numerous generative tasks in CV such as style transfer [60], image inpainting [98], and super-resolution [134]. They have shown promising results in realistic image generation. In this section, we will discuss some interesting GAN architectures.

CycleGAN [141]: CycleGANs are special types of GANs primarily used for image style transfer. For instance, they can learn to transform a human's face into a different age group. The critical difference between a CycleGAN and a vanilla GAN is that, in addition to the standard adversarial loss, CycleGAN introduces a cycle-consistency loss, which enables the learning of a transformation, which is an inverse mapping of adversarial loss's learned transformation. Figure 2.13 shows the architecture of a CycleGAN.

StyleGAN [64]: StyleGAN is a unique formulation of GAN, allowing it to generate very high-quality images. The underlying idea is to stack up layers incrementally, where

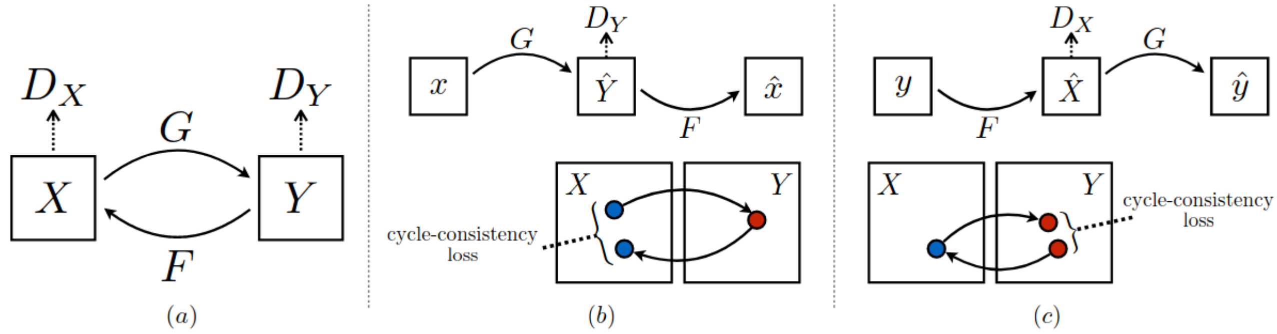


Figure 2.13: CycleGAN architecture [141]. (a) The model learns two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and the adversarial discriminators D_X and D_Y. Two cycle-consistency losses are introduced to regularize the mapping: (b) Forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) Backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$.

initial layers learn to generate low-resolution images (2×2) and the resolution increases gradually. Figure 2.14 compares the style-based generator of StyleGAN with a traditional GAN generator.

text-2-image [105]: The text-2-image network made significant progress in generating meaningful images based on an explicit textual description. In this formulation, in addition to the noise, the generator is also fed with the vector representation of the textual description as input.

DiscoGAN [65]: DiscoGAN was proposed to infer cross-domain relationships in an unsupervised manner. In terms of the fundamental idea, it is very similar to CycleGAN. The only difference lies in the loss function - while CycleGAN uses a single cycle-consistency loss, DiscoGAN introduces *two* reconstruction losses, one for both the domains.

2.8 Multimodal Frameworks

This section briefly discusses some generic multimodal frameworks proposed in the literature. We will limit our discussion to generative frameworks. Multimodal deep learning is an active field of research, and many aspects of it can be individually targeted. For instance, one could focus on fusion techniques or explore the interpretability of the shared latent space. In this thesis, we will focus more on different fusion techniques.

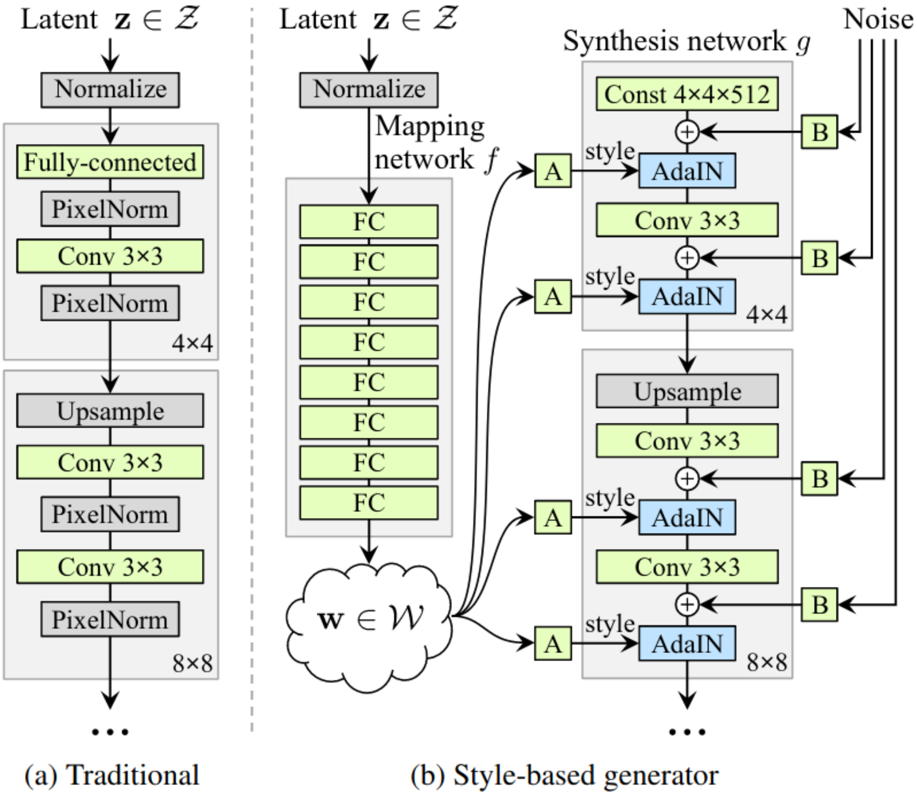


Figure 2.14: Comparison between (a) Traditional generator network in a GAN, and (b) Style-based generator used in the StyleGAN architecture [64].

The subsequent subsections discuss some basic multimodal frameworks and fusion techniques. Towards the end, we introduce the two tasks used for evaluating proposed methodologies in this thesis.

2.8.1 Generative Multimodal frameworks

We will now briefly discuss some generative multimodal frameworks.

Deep Restricted Boltzmann Machines (Deep RBMs)

Restricted Boltzmann machines [116] were one of the first generative frameworks to be used for multimodal deep learning. RBMs are a variant of Boltzmann machines [4], which may also be interpreted as the stochastic, generative counterpart of Hopfield networks [55]. A Boltzmann machine consists of two types of units: visible and hidden and allows connections between any units. This leads to exponential learning time, thereby leading to the development of restricted Boltzmann machines where a connection is allowed only between a hidden and a visible unit. Figure 2.15 shows the architecture of an RBM.

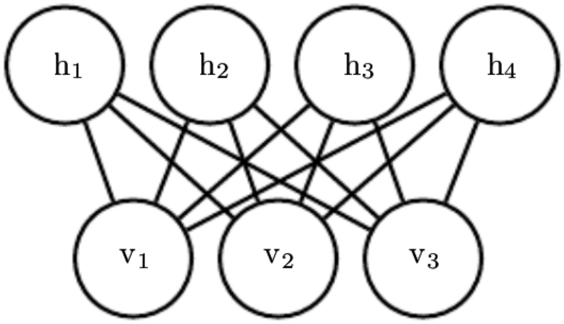


Figure 2.15: RBM architecture. It is a bipartite graph with visible nodes (v_i) on one side and hidden nodes (h_i) on the other. [42].

In Ngiam et al. [91], the authors use RBMs for multimodal learning in two different configurations. The first configuration involves *separately* learning posteriors of hidden units for each modality, and then concatenating vectors from each modality to construct a shared representation. This results in a shallow network as the RBM cannot learn the non-linear relationship between vectors from the two modalities. The second configuration resolves these issues by using a deep autoencoder initialized using bimodal DBN weights. A DBN, or a deep belief network, can be interpreted as a collection of RBMs where the hidden units of the current layer serve as visible units for the next.

These networks were also used for cross-modal reconstruction, i.e., at inference time, input from only one modality was present in the input vector. It showed some promising results for the task of classification as well.

The next section briefly discusses *fusion* in the context of multimodal deep learning.

2.8.2 Fusion

Fusion refers to the process of combining inputs from various modalities to construct a joint representation so as to model the interaction from all the modalities [36]. It helps in a better learning procedure; information absent from one modality may be compensated by another modality, thereby, helping the model. Intuitively, too, this follows as a human brain is continually processing multimodal input. Fusion can be broadly divided into three types:

Early fusion: In early fusion, *output* vectors from each modality's learner (or encoder) are merged to make a decision.

Late fusion: In late fusion, the *intermediate semantic information* from each modality's encoder is combined.

Hybrid fusion: In hybrid fusion, the integration procedure to combine multimodal inputs involves a mixture of early and late fusion across different levels.

Multimodal fusion is an active field of research, and we will discuss more advanced fusion techniques in Chapter 3. We will now discuss the two tasks we use to measure the effectiveness of the proposed methods.

2.9 Multimodal Emotion Recognition

Emotion recognition in a multimodal setting refers to the task of understanding an individual's emotional state [103]. Inputs from different modalities such as facial expressions, raw speech, and textual description of an utterance, are fed to the model during inference.

Formally, emotion recognition is a classification problem, and multiple methods have been proposed to tackle this problem [103, 140, 79, 138, 137, 91]. They all use different fusion techniques, but the most important take-away from these prior works is that a better fusion method leads to better performance. It can also be inferred from these works that a complex fusion technique will not always yield better results.

Identifying emotion autonomously is challenging because the information from the different modalities is heterogeneous in nature, and they need not be aligned. Perhaps the most challenging aspect of emotion recognition is resolving ambiguity. For instance, the sentence, "I am feeling surreal." would generally indicate content in an individual, but the same sentence could also be spoken in a sarcastic manner. Proposed multimodal frameworks aim to handle such ambiguities; however, there is still room for improvement as the models still have a hard time distinguishing between closely related but different emotions, such as happiness and calmness.

2.10 Multimodal Machine Translation

Multimodal machine translation is an extension of standard machine translation task in NLP, which involves translating input sentences written in one language to a different language. Here, too, we have inputs from multiple modalities; however, in this case, these inputs are more diverse than for emotion recognition. For instance, we may have *any* type of visual cue from the visual modality, ranging from a facial expression to an image of a football ground.

In addition to resolving ambiguity, the network now needs to develop an understanding of the presented scenario to generate better translations. For instance, knowing that the text is about football may help the model in translating jargon words more easily, as such words may not occur as frequently as “normal” words in the aligned dataset. Moreover, multimodal machine translation is a generative task, which further increases the complexity involved. These challenges make multimodal machine translation an arduous task.

Multiple methods have been proposed in the literature for this task; however, there is room for significant improvement as the methods are still unable to exploit multimodal information as effectively [119, 92, 10]. We will discuss more such models in the next chapter.

Chapter 3

Related Work

In this chapter, we will briefly review some previous works related to our task. Most earlier works in multimodal deep learning focused on traditional shallow classifiers such as support vector machines [31] and Naive Bayes classifiers [87] to exploit bimodal data. Inspired by the success of deep learning over the last decade across multiple tasks, Ngiam et al. [91] train end-to-end deep graph neural networks to reconstruct missing modalities at inference time. They demonstrate that better features for one modality can be learned if relevant data from different modalities is available at training time; however, they employ simple concatenation for fusion. Hence, the joint representation learned is shallow and is not guaranteed to model inter-modal interactions. Their findings were later verified by Srivastava et al. [120], who use a Deep Boltzmann Machine [109] to generate data from the image and text modality. Huang et al. [57] construct a multilingual common semantic space to achieve better machine translation performance by extending correlation networks [18]. They use multiple non-linear transformations to reconstruct sentences from one language to another repeatedly and finally build a common semantic space for all the different languages.

In an attempt to mitigate issues presented by shallow fusion methods such as concatenation, techniques such as the Tensor Fusion Network (TFN) [140], and Low-rank Multimodal Fusion (LMF) [79], were proposed; however, the problem of effectively modelling context in multimodal samples remains unsolved. The subsequent sections discuss different fusion/alignment strategies from the literature.

3.1 Concatenation

Concatenation is the simplest way of constructing a joint representation of feature vectors from multiple modalities. This method requires individual learners (encoders) for all input modalities. Each encoder aims to learn feature vector (generally, the hidden representation) for the provided input. The concatenation operation, depending on the type of fusion, is executed at an appropriate stage of the learning process for the desired entities. For instance, if using early fusion, output vectors of individual learners are concatenated. Therefore, concatenation of the vectors occurs *after* the learners encode their respective inputs. This is not an effective way to fuse multimodal inputs as modelling non-linear inter-modal interaction becomes difficult in such scenarios [91]. In contrast, if using late fusion, input vectors from individual learners are concatenated, i.e., concatenation of feature vectors occurs *before* the learners have encoded their respective inputs [140]. Therefore, late fusion can not adequately model inter-modal interactions.

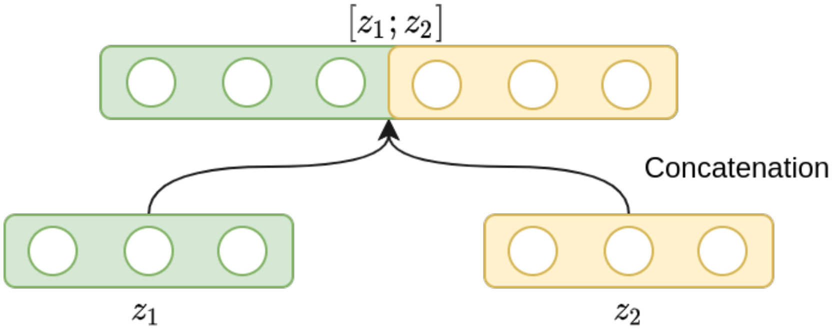


Figure 3.1: Simple concatenation fusion. **Note:** ; denotes concatenation

Despite several disadvantages, concatenation serves as a good litmus test as it is the most straightforward fusion technique to implement amongst all. It may not be the best way to learn a joint multimodal representation, but it does boost a unimodal framework's performance on a given task. This is helpful in quickly estimating the effectiveness of introducing multiple modalities. For this reason, simple concatenation is used as a baseline in all major works. Figure 3.1 shows simple concatenation for fusion.

3.2 Tensor Fusion Network (TFN)

Tensor Fusion Network (TFN) [140] was one of the first fusion techniques to show a significant improvement over simple concatenation. Unlike previously proposed multimodal frameworks, it aims to capture *both* intra- and inter-modal dynamics simultaneously, in an end-to-end fashion. The network has two essential components: 1) a Modality Embedding Subnetwork for each modality to capture intra-modal dynamics, and 2) a Tensor Fusion module to capture inter-modal dynamics. It has an additional module for performing inference, but we will focus only on the two mentioned components. The Modality Embedding Subnetworks are tasked with outputting a rich modality embedding when fed input unimodal features as input. The Tensor Fusion module, on the other hand, explicitly aggregates uni-, bi-, and tri-modal interactions by performing a 3-fold Cartesian product from previously obtained modality embeddings. Figure 3.2 shows the complete architecture of the TFN in detail.

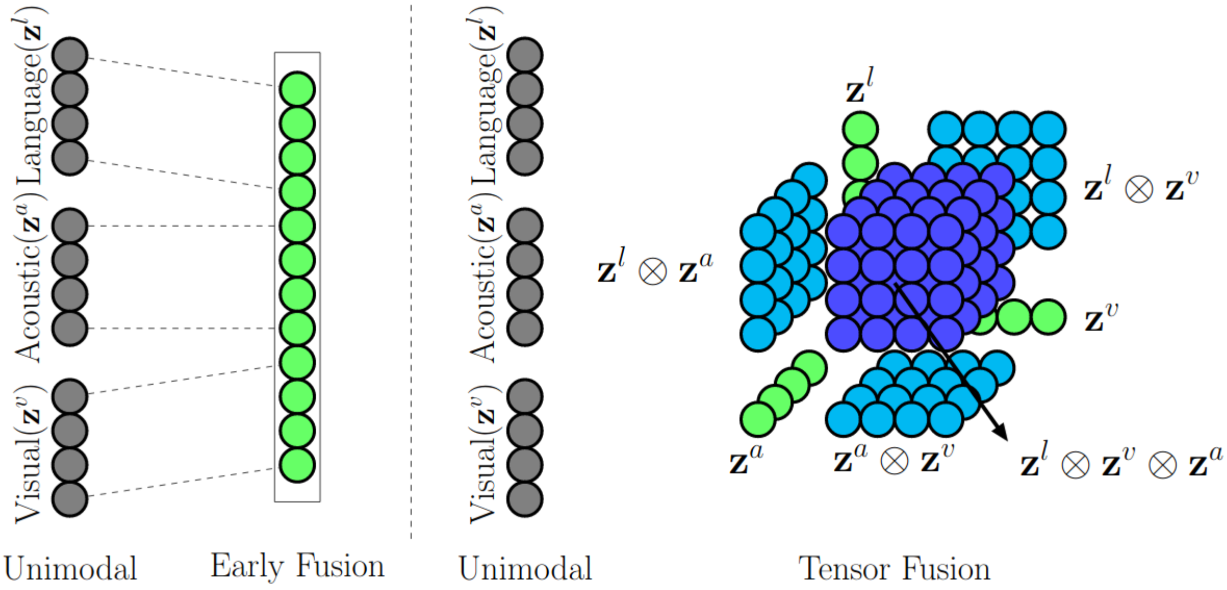


Figure 3.2: **Left:** Commonly used Early Fusion (multimodal concatenation) **Right:** Tensor Fusion Network with uni-, bi, and tri-modal subtensors [140].

TFN showed significant improvements for multimodal sentiment analysis, outperforming previously existing techniques by a large margin. However, it had a major drawback: the Cartesian product operation in the Tensor Fusion module makes the algorithm very inefficient, thereby leading to massive computational overhead. The memory costs exponentiate, especially when modelling trimodal interactions. Next, we will look at a method, which tries to mitigate this issue.

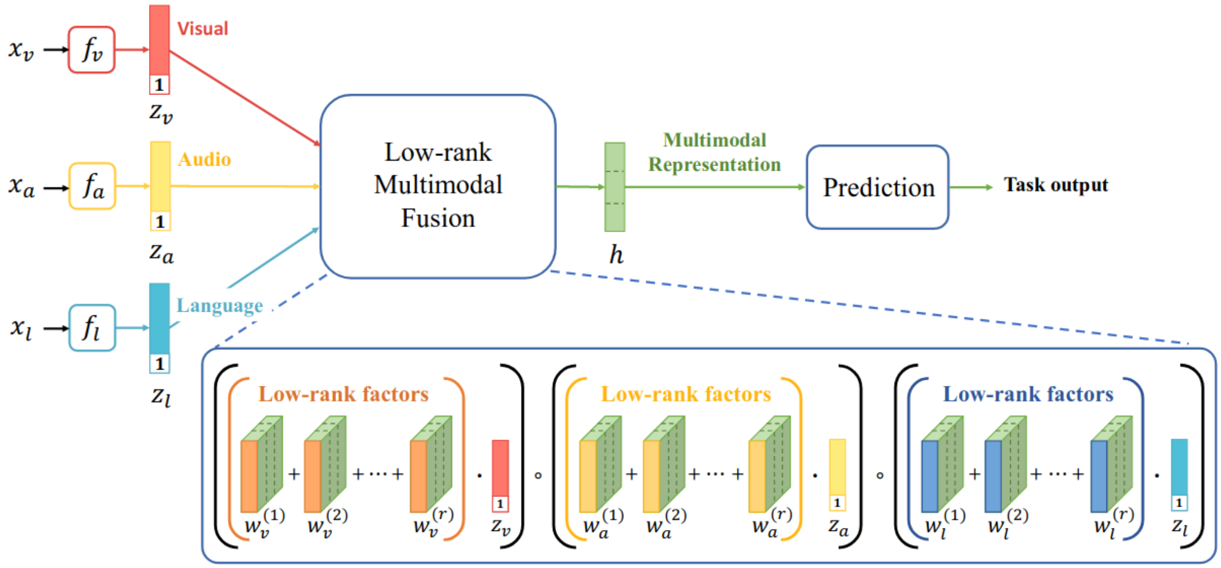

x_v
f_v
Visual
1
z_v
x_a
f_a
Audio
1
z_a
x_l
f_l
Language
1
z_l
Low-rank Multimodal Fusion
h
Multimodal Representation
Prediction
Task output
Low-rank factors
$w_v^{(1)}$ + $w_v^{(2)}$ + ⋯ + $w_v^{(r)}$ · z_v
∘
Low-rank factors
$w_a^{(1)}$ + $w_a^{(2)}$ + ⋯ + $w_a^{(r)}$ · z_a
∘
Low-rank factors
$w_l^{(1)}$ + $w_l^{(2)}$ + ⋯ + $w_l^{(r)}$ · z_l

Figure 3.3: Overview of Low-rank Multimodal Fusion (LMF). First unimodal representations z_a, z_v, z_l are obtained by passing the unimodal inputs x_a, x_v, x_l into three sub-embedding networks f_v, f_a, f_l respectively. Then, low-rank multimodal fusion with modality-specific factors is performed to obtain fused multimodal representation, which can later be used for prediction [79].

3.3 Low-rank Multimodal Fusion (LMF)

Low-rank Multimodal Fusion (LMF) [79] was proposed in an attempt to scale up prior fusion techniques while maintaining reasonable model complexity, and without compromising model-performance. It was an improvement over prior methods such as Zadeh et al. [140] and Fukui et al. [38], whose computation and memory costs increase exponentially. LMF addresses these issues by decomposing the weight tensors into low-rank factors, thereby, lessening parameters in the model. The decomposition process is further optimized by leveraging parallel decomposition of low-rank weight tensor and input tensor to compute tensor-based fusion [79]. Figure 3.3 shows a general overview of the LMF network.

LMF was able to reduce the parameters by several-folds ($\sim$ 11 times when compared against TFN) while maintaining competitive performance across different tasks such as multimodal emotion recognition, and multimodal sentiment regression. Additionally, it scales linearly with increasing modalities, which is significantly better than an exponential increase in prior networks.

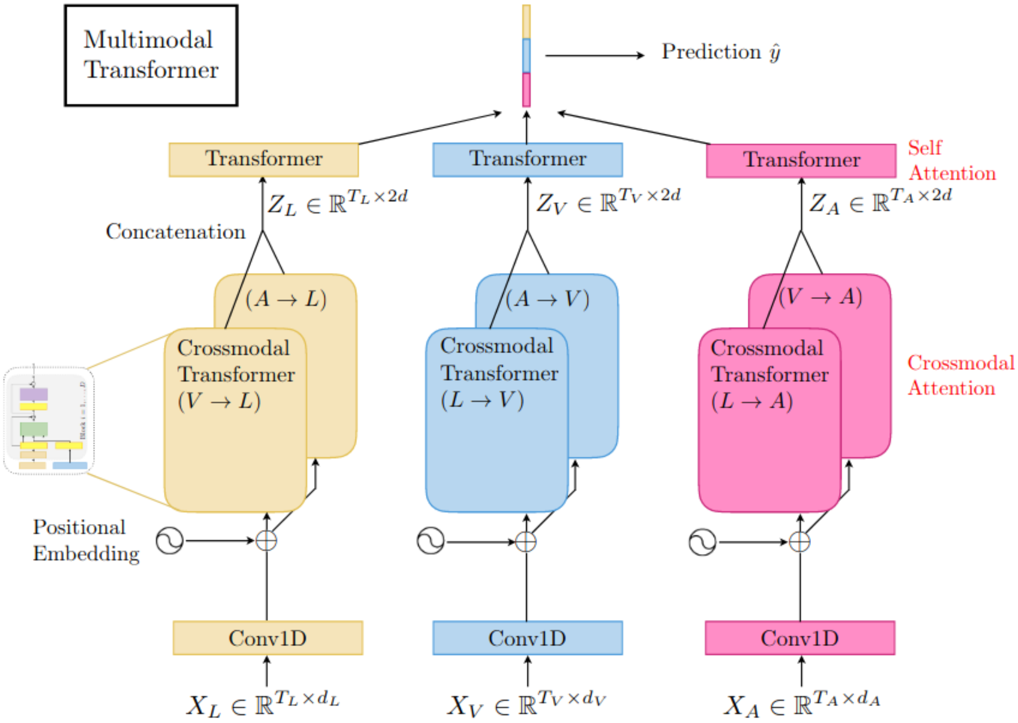

Multimodal Transformer
Prediction $\hat{y}$
Transformer
Transformer
Transformer
Self Attention
$Z_L \in \mathbb{R}^{T_L \times 2d}$
$Z_V \in \mathbb{R}^{T_V \times 2d}$
$Z_A \in \mathbb{R}^{T_A \times 2d}$
Concatenation
$(A \rightarrow L)$
$(A \rightarrow V)$
$(V \rightarrow A)$
Crossmodal Transformer $(V \rightarrow L)$
Crossmodal Transformer $(L \rightarrow V)$
Crossmodal Transformer $(L \rightarrow A)$
Crossmodal Attention
Positional Embedding
Conv1D
Conv1D
Conv1D
$X_L \in \mathbb{R}^{T_L \times d_L}$
$X_V \in \mathbb{R}^{T_V \times d_V}$
$X_A \in \mathbb{R}^{T_A \times d_A}$

Figure 3.4: Architecture of Multimodal Transformer (MulT) for modalities (L, V, A) [124].

3.4 Multimodal Transformer (MulT)

Multimodal Transformers (MulT) [124] were recently proposed to align data from different modalities implicitly. On a high-level, MulT has cross-modal attention modules for each modality, whose outputs are merged through a feed-forward fusion mechanism. Each cross-modal attention module iteratively learns the alignment (or attention) vector between its target modality and the remaining two modalities' feature vectors. For instance, the cross-modal attention module with target modality as the text will learn the attention vector between text, and combined visual and speech features. Figure 3.4 shows the complete model architecture of MulT, which also contains self-attention transformers for prediction.

Due to its cross-attention modules, MulT can latently adapt to streams from one modality to another without explicit data alignment. It has shown encouraging results on the tasks of emotion recognition and multimodal sentiment analysis. Moreover, the generic nature of the model makes it suitable for a variety of other tasks. For instance, it could be potentially adapted for Visual Question Answering (VQA), where the input signal is a mixture of static and time-evolving signals. Figure 3.5 exemplifies the task of VQA.

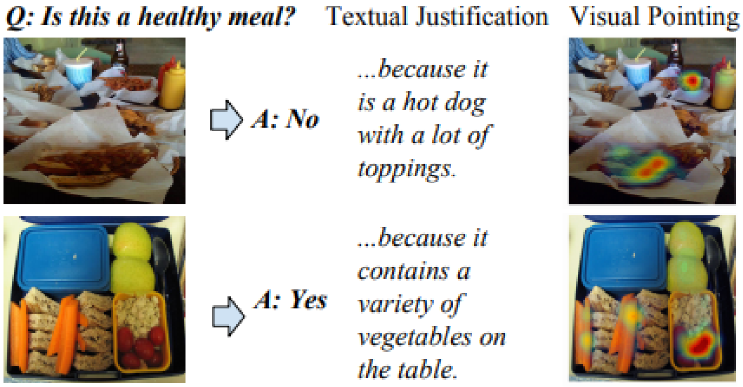

Q: Is this a healthy meal?
Textual Justification
Visual Pointing
A: No
...because it is a hot dog with a lot of toppings.
A: Yes
...because it contains a variety of vegetables on the table.

Figure 3.5: Example of visual question answering (VQA) task [58]. Here, additional information from an image is used during input to answer the given question along with its visual and textual justification.

3.5 Variational Mixture-of-Experts Autoencoders

Variational Mixture-of-Experts Autoencoders [113] are a class of deep generative multimodal frameworks that aim to learn a synergic shared representation for multiple modalities. They follow the objective of importance weighted autoencoder [13] along with a K-sample estimator, which results in a tighter lower bound than originally proposed VAEs. Additionally, they model the joint multimodal posterior as a mixture of unimodal posteriors.

This method shows promising results for cross-modal generation, and the model is able to generate correlated images for given captions, and vice versa. However, the scaling of the Mixture-of-Experts method still needs to be studied thoroughly as we increase input modalities for the network.

Multimodal generation is an active field of research with a lot to achieve. However, prior works to tackle this issue establish foundational stones to build-up on. In Chapter 4, we will discuss our proposed methodologies, focussing primarily on fusion techniques that can easily be plugged into a generative/discriminative framework for better generation/understanding.

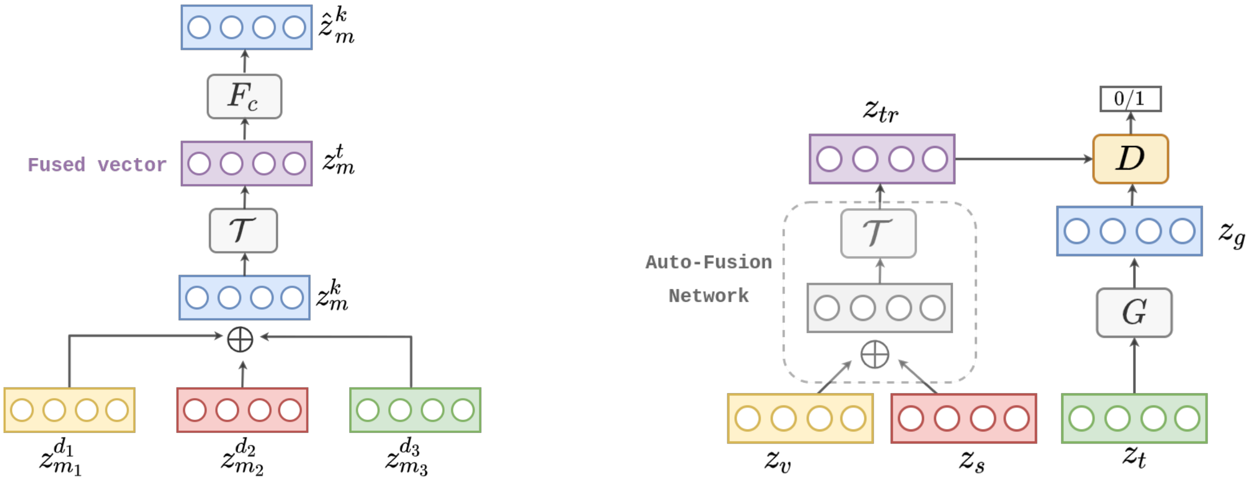
Chapter 4

Approach

In this chapter, we will discuss the proposed methodologies for effectively fusing inputs from multiple modalities. A majority of the fusion techniques proposed in the literature, such as concatenation, and TFN, involve a deterministic operation for constructing the joint multimodal representation. For instance, in concatenation, the model is presented with a concatenation of all unimodal features for making a decision. Similarly, in TFN, the 3-fold Cartesian product of unimodal features is used for prediction. In both cases, the algorithm focuses more on learning rich unimodal features. However, there is no such "learning" procedure for joint representation; they are simply constructed by combining unimodal features in a specific fashion. In this thesis, we will refer to such techniques as *static* fusion techniques. Since there is no special learning procedure for the joint representation, it becomes challenging for the final predictor module to model the complex dynamics of multimodal features. In other words, the model is unable to utilize multimodal information effectively.

On the other hand, fusion methods such as LMF, and MulT are *adaptive* because they involve a cognitive feature processing step to construct the joint representation. In LMF, it is the decomposition module, and in MulT, it is the final feed-forward fusion mechanism, which takes care of constructing the joint representation such that it may only contain the most essential features from each modality. However, these algorithms themselves are either not very straightforward to implement or impose high computational and memory costs. In this thesis, we aim to address the above issues, and propose two *adaptive* fusion techniques that are easy to implement and posit reasonable complexity.

Our fusion methods involve concatenating unimodal embeddings at the first step. Hence, to avoid any conflicts with past works, we will only consider the steps *after* con-



(a) Auto-Fusion network

(b) GAN-Fusion Network

Figure 4.1: Proposed architectures. (a) Auto-Fusion network: Assuming that $\boldsymbol{z}_{m_1}^{d_1}$, $\boldsymbol{z}_{m_2}^{d_2}$, and $\boldsymbol{z}_{m_3}^{d_3}$ represent the video, speech, and text latent vectors respectively, we first concatenate them to obtain $\boldsymbol{z}_m^k$. It is then passed through $\mathcal{T}$ which outputs the "autofused" vector $\boldsymbol{z}_m^t$. We then obtain the reconstructed concatenated vector $\hat{z}_m^k$ by passing the autofused vector through F_c, another transformation layer. Finally, we optimize the loss between $\hat{z}_m^k$ and $\boldsymbol{z}_m^k$. (b) GAN-Fusion module for the text modality: Assuming that $\boldsymbol{z}_s$, $\boldsymbol{z}_v$, and $\boldsymbol{z}_t$ are the latent speech, video, and text vectors, respectively, we first autofuse $\boldsymbol{z}_s$ and $\boldsymbol{z}_v$ to give $\boldsymbol{z}_{tr}$. Simultaneously, we pass $\boldsymbol{z}_t$ through the generator G to get $\boldsymbol{z}_g$. The generator loss tries to match $\boldsymbol{z}_{tr}$ and $\boldsymbol{z}_g$ and discriminator D tries to distinguish between $\boldsymbol{z}_{tr}$ and $\boldsymbol{z}_g$, the two sources of input. **Note:** $\bigoplus$ denotes concatenation.

catenation as a part of our fusion method. This is because we do not use the concatenated vector for final prediction, rather, it is only an introductory step. The next few sections describe the proposed fusion techniques and the end-to-end training process in detail.

4.1 Model Overview

In order to mitigate the "staticness" of existing fusion methods, we propose two adaptive yet simple fusion techniques, *Auto-Fusion* and *GAN-Fusion*. They aim to effectively combine multimodal inputs and mitigate the problem of shallowness and computational overhead in fusion techniques. As described at the beginning of the chapter, most fusion methods proposed in the past, such as concatenation, either result in a shallow network [91], or are computationally expensive, such as tensor fusion [140], and there is no intelligent feature extraction. In both cases, the fusion operation is specified beforehand, and the network does not have the freedom to learn multimodal interaction on its own. In this section, we describe the two methods developed for effectively combining multimodal inputs.

4.2 Auto-Fusion

This method encourages the model to extract intermodal features by maximizing the correlation between multimodal inputs. In this method, we first concatenate individual unimodal features, and then pass them through a transformation layer to get an *autofused* latent vector. We use appropriate learners for individual modalities. We then try to reconstruct the originally concatenated vector from the autofused latent vector. Finally, we minimize the Euclidean distance between the original and reconstructed concatenated vector. This process ensures that the learned autofused vector does not contain arbitrary signals from the input concatenated latent vector. Additionally, the model is incentivized to "compress" information without losing any critical cues as much as possible. In other words, it increases the correlation between the autofused and the concatenated latent vector. This method applies to any scenario where multiple features need to be combined. For example, it can even be used to combine the forward and backward hidden states of LSTMs [54], instead of pooling methods such as 1D pooling, max pooling, sum pooling or even simple concatenation.

We now discuss the Auto-Fusion network in detail. We pose the fusion of multimodal inputs as a compression problem, where we must retain as much information from the individual modalities as possible. Given n (≤ 3 in our case) d-dimensional multimodal latent vectors, $\boldsymbol{z}_{m_1}^{d_1}, \boldsymbol{z}_{m_2}^{d_2}, \ldots, \boldsymbol{z}_{m_n}^{d_n}$, we first concatenate them to obtain a vector, $\boldsymbol{z}_m^k$, where $k = \sum_i^n d_i$. Then, we apply a transformation, $\mathcal{T}$, to $\boldsymbol{z_m^k}$, reducing its number of dimensions to t. Then, we use $\boldsymbol{z^t}$ to reconstruct the originally concatenated vector $\boldsymbol{\hat{z}_m^k}$. Finally, we calculate the loss, J_{tr}, between $\boldsymbol{\hat{z}_m^k}$, and $\boldsymbol{z_m^k}$. The simplest version of Auto-Fusion network employs the mean squared error (MSE) loss function, which aligns with our motivation to compress multimodal features, so as to filter out the less useful signals. These steps could be followed in Figure 4.1(a) and the MSE loss for Auto-Fusion network is given by:

$$J_{tr} = || \, \boldsymbol{\hat{z}_m^k} - \boldsymbol{z_m^k} \, ||^2 \tag{4.1}$$

4.3 GAN-Fusion

In addition to the "staticness" of existing methods, there is also the challenge of distinguishing between ambiguous cases. For instance, the sentence "Kevin, this is hilarious," could be said in a funny, or sarcastic manner. Resolving ambiguity becomes especially important when working on social problems such as hate speech detection. Existing methods, even when fed with the corresponding speech vector, cannot effectively distinguish between similar but different emotions such as happiness and calmness. We hypothesize that this is because they do not learn the conditional distribution of sentiment given an utterance (an utterance includes input from all available modalities).

In order to mitigate this issue, we propose an adversarial training regime that is incentivized to learn the desired conditional distribution. For a task such as emotion recognition, the objective would be sentiment given an utterance. For a more challenging generation task, the model could learn a more complex behaviour, such as the association of different sentences based on how similar they sound and their polarity. In our experiments, we show that our GAN-based approach is better able to learn multimodal dynamics compared to the existing methods.

We now describe GAN-Fusion's architecture in detail. For a given multimodal sample x, we first encode the inputs from each modality (speech, visual and text) to get the respective latent vectors, $\boldsymbol{z_s}, \boldsymbol{z_v},$ and $\boldsymbol{z_t}$. Choosing a target modality such as, text, we pass $\boldsymbol{z_t}$ through a generator to obtain $\boldsymbol{z}_g = G(\boldsymbol{z_t})$ and autofuse the remaining latent vectors, $\boldsymbol{z_s}$, and $\boldsymbol{z_v}$ simultaneously to obtain $\boldsymbol{z_{tr}}$. In the event where we have input from only one modality in addition to text, we do not need an Auto-Fusion, and can simply treat the other modality's vector as $\boldsymbol{z_{tr}}$. Finally, we train the network in adversarial fashion, labelling $\boldsymbol{z_{tr}}$ as positive samples and $\boldsymbol{z_t}$ as negative samples. The adversarial loss, J_{adv}, is given below:

$$\min_{G}\max_{D} J_{adv}(D,G) = \mathbb{E}_{x\sim p_{\boldsymbol{z_{tr}}}(x)}[\log D(x)] + \mathbb{E}_{\boldsymbol{z}\sim p_{z_t}(\boldsymbol{z})}[\log(1-D(\boldsymbol{z}_g))] \tag{4.2}$$

Overall, the generator G tries to align features of the target modality with features from the complementary modalities and the discriminator tries to identify the type of its input. Such a translation between latent vectors has been shown to learn an "intermediate" latent vector denoting their joint representation [102, 39]. Learning the latent space in such an adversarial manner induces a clustering effect on the latent space, where texts associated with similar sounds and visuals are grouped together. We hypothesize that adversarial training helps the model to learn the relative topology of the complementary modalities' latent space, which, in turn improves sampling for the target modality. This is

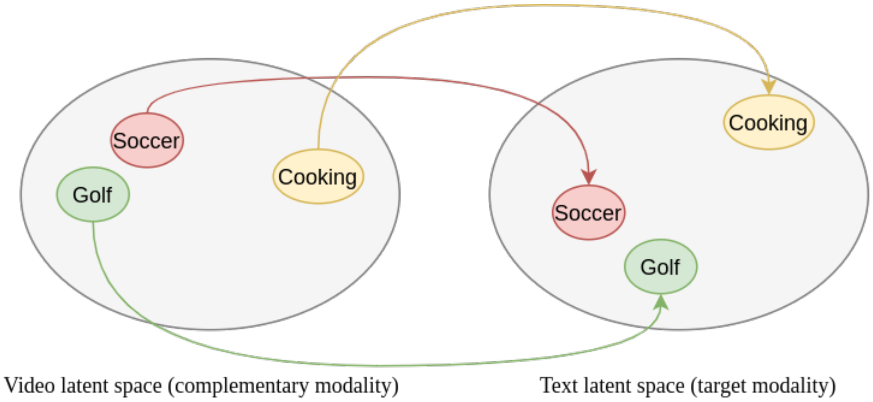

Soccer
Cooking
Golf
Cooking
Soccer
Golf
Video latent space (complementary modality)
Text latent space (target modality)

Figure 4.2: Influence of complementary modality on the topology of the target modality's latent space. For the sake of simplicity, we only have one complementary modality (video) in this example. We hypothesize that adversarial learning in GAN-Fusion incentivizes the model to learn the text latent space such that it is able to preserve the relative topology from the video latent space. For example, latent embeddings for tokens related to Soccer and Golf would follow similar relative positioning in the latent space as present in the video latent space. Furthermore, embeddings for Soccer and Golf – falling under the general category of Sports – will be mapped closer to each other compared to embeddings from an unrelated topic such as Cooking. For multimodal machine translation, this means that if the model is fed with a Golf video along with the source text as input, it may be better able to sample jargon words from Golf due to this topology inheritance, ultimately improving generation quality.

also explained in Figure 4.2. Figure 4.1(b) shows GAN-Fusion module for the text modality. We have one such module for each modality. We concatenate outputs of each module, and pass the concatenated vector through a feed-forward layer. This output represents the fused multimodal representation.

4.4 Training Process

In this section, we describe the end-to-end training process for using the proposed fusion methods for 1) Classification (e.g. speech emotion recognition, hate speech detection) and 2) Generation (e.g. visual question answering, machine translation).

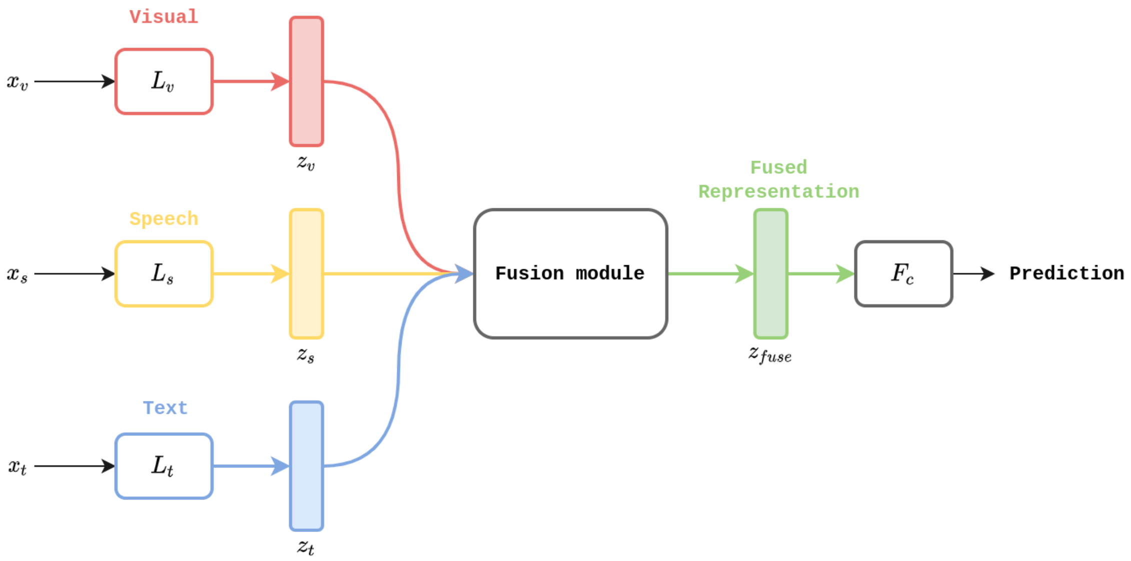


Figure 4.3: Using proposed fusion techniques for classification. Unimodal inputs x_v, x_s, x_t are passed to their respective learners L_v, L_s, L_t to obtain unimodal representations z_v, z_s, z_t. Here, v, s, t correspond to visual, speech, and text modalities respectively. The individual unimodal representations are then passed through the fusion module (either Auto-Fusion or GAN-Fusion,) which outputs the fused multimodal representation z_{fuse}. Finally, z_{fuse} is passed through a fully-connected layer F_c. Predictions can be obtained by applying a $softmax$ on these outputs.

4.4.1 Classification

The training process for classification is straightforward. Figure 4.3 shows the end-to-end model pipeline for using the proposed fusion techniques for classification. Since both Auto-Fusion and GAN-Fusion are modular in nature, using them is as simple as plugging them between the individual learners and the final predictor. The overall loss function can be described as follows:

$$J_{total} = \lambda_1 J_{fusion} + \lambda_2 J_{classification} \tag{4.3}$$

Here, J_{fusion} refers to the loss function of the fusion network. It equals J_{tr} (from equation 4.1) when using Auto-Fusion, and J_{adv} (from equation 4.2) when using GAN-Fusion. Furthermore, $J_{classification}$ refers to the classification loss, for instance, cross-entropy loss. λ_1 and λ_2 are hyperparameters to tune.

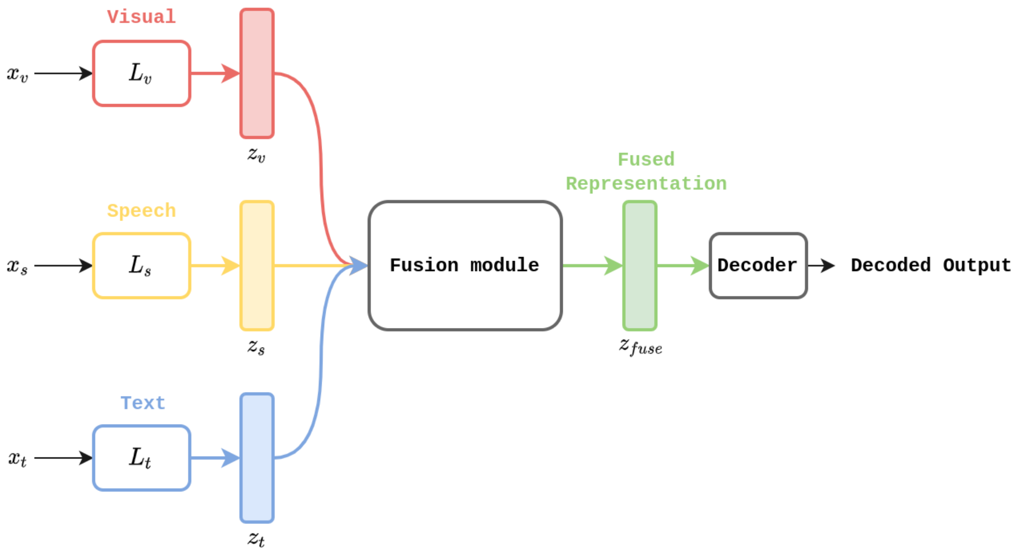


Figure 4.4: Using proposed fusion techniques for classification. Unimodal inputs x_v, x_s, x_t are passed to their respective learners L_v, L_s, L_t to obtain unimodal representations z_v, z_s, z_t. Here, v, s, t correspond to visual, speech, and text modalities respectively. The individual unimodal representations are then passed through the fusion module (either Auto-Fusion or GAN-Fusion,) which outputs the fused multimodal representation z_{fuse}. Finally, z_{fuse} is passed through a decoder, which generates outputs for the desired target modality.

4.4.2 Generation

The training process for generation is similar to that of classification. Figure 4.4 shows the end-to-end model pipeline for using the proposed fusion techniques for generation. It can be seen that the pipeline looks very similar to a vanilla Seq2Seq network. We have just introduced a fusion module between the encoder and the decoder module. We only validate this process for generating text, but this method could very well be used for generating outputs for different target modalities. The overall multi-task loss function, in this case, can be described as follows:

$$J_{total} = \lambda_1 J_{fusion} + \lambda_2 J_{generation} \tag{4.4}$$

Here, J_{fusion} carries the same meaning as described for the classification network, and $J_{generation}$ refers to the generation loss, for instance, cross-entropy loss for a Seq2Seq model.

λ_1 and λ_2 are hyperparameters to tune.

In the next chapter, we discuss the experiments performed to measure the effectiveness of the proposed techniques.

Chapter 5

Experiments

This chapter describes the experimental setup used to evaluate our techniques. We measure the effectiveness of proposed fusion techniques on two tasks: 1) multimodal machine translation, and 2) multimodal emotion recognition.

The subsequent sections describe the datasets used, implementation details, and evaluation results.

5.1 Datasets

To aid multimodal research, numerous datasets have been introduced in the past. We choose three such datasets: 1) the IEMOCAP dataset to test our methods on emotion recognition, 2) the How2 dataset, and 3) the Multi30K dataset to test our methods on multimodal machine translation. We now briefly describe the three datasets used.

5.1.1 IEMOCAP

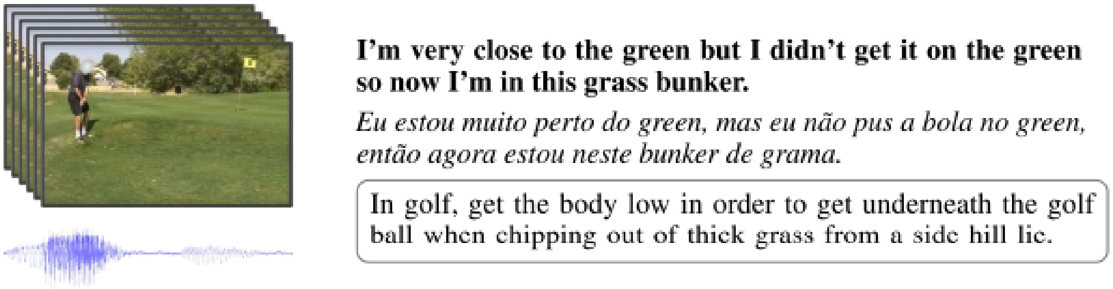
We use the benchmark Interactive Emotional Dyadic Motion Capture (IEMOCAP) database [14] for emotion recognition. The dataset contains emotion-annotated utterances that we split to obtain a wav file for each transcribed sentence. The dataset is already split into multiple utterances for each session, and we further split each utterance file to obtain wav files for each sentence. We split each utterance file to obtain a wav file for every sentence using the start and end timestamp provided for the transcribed sentences. This results in a total of ∼ 10K audio files, which are then used to extract features to predict a given utterance's emotion. Concretely, we identify the task as an emotion recognition problem, where, given a sentence and its audio signal, we aim to infer the correct emotion for that utterance.

5.1.2 How2

We evaluate Auto-Fusion and GAN-Fusion on the multimodal How2 dataset [111], which offers 79,114 instructional videos in addition to word-level time alignments to the ground-truth English subtitles and their respective crowd-sourced Portuguese translations. A brief description of the video clip is also included to encourage future work on image captioning. This dataset was created by scraping videos along with their metadata from YouTube using a keyword-based spider, and manually extracting and processing visual, auditory, and textual features. Figure 5.1 shows a multimodal sample from the How2 dataset.

Unlike other popular datasets frequently featured in the multimodal deep learning literature, such as CUAVE [99] and AVLetters [84], the How2 dataset is, in fact, trimodal, therefore making it suitable to evaluate the contribution of each modality towards different tasks.

Further, as a large-scale multilingual dataset, it enables a convenient medium for neural machine translation in our thesis.



I'm very close to the green but I didn't get it on the green so now I'm in this grass bunker.

Eu estou muito perto do green, mas eu não pus a bola no green, então agora estou neste bunker de grama.

In golf, get the body low in order to get underneath the golf ball when chipping out of thick grass from a side hill lie.

Figure 5.1: A multimodal sample from the How2 dataset [111]

5.1.3 Multi30K

In addition to the How2 dataset, we also run experiments on the Multi30K dataset [37] extended for French, where each sample has an image, its description in the source language and its translated version. We choose the En-Fr version of the dataset to run our experiments. Figure 5.2 shows a data sample from the Multi30K dataset.

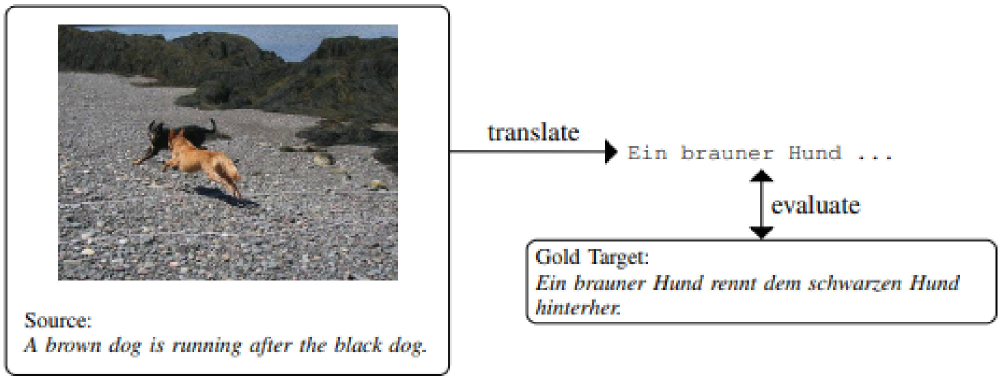

translate
Ein brauner Hund ...
evaluate
Gold Target:
Ein brauner Hund rennt dem schwarzen Hund hinterher.
Source:
A brown dog is running after the black dog.

Figure 5.2: A multimodal sample from the Multi30K dataset [119]

5.2 Implementation Details and Hyperparameters

In this section, we enumerate the implementation details and hyperparameters for our models.

5.2.1 Implementation framework and computational resources

All the networks in our experiments are implemented in PyTorch [97]. To train the different classification networks and generation networks on the Multi30K dataset, we use either an Nvidia GTX 1080Ti GPU with 12GB of RAM or an Nvidia RTX 2080Ti with 12GB of RAM. However, to train our network on the How2 dataset, we use Nvidia P100 with 16 GB RAM[1] as raw video vectors required needed extra memory.

5.2.2 Hyperparameters

We use an LSTM encoder with 256 hidden units as the learner for textual description. To encode audio vectors in the IEMOCAP dataset, we first pre-process the raw audio vectors

[1] These GPUs were accessed through Sharcnet clusters set up at the University of Waterloo

and compute an 8-dimensional feature vector. We then use an LSTM encoder with 50 units as a learner for those feature vectors. For the How2 dataset, raw audio vectors are already present, and we use a LSTM encoder with 256 hidden units for encoding. For the visual modality, we use the Kaldi vectors already provided for videos in the How2 dataset and use a VGG to encode the images in the Multi30K dataset. In all our networks, we use 100 dimensions to represent the latent vectors for all modalities, except for audio in the IEMOCAP dataset, where we use 50 units to express the latent vector.

5.3 Evaluation Metrics

We now enumerate the different benchmark metrics to evaluate the generation and classification quality of our models quantitatively.

5.3.1 Precision

We report Precision, Recall, and F1-score for all our classification experiments. We now describe each of them in detail below.

Precision, also known as positive predictive value, is the fraction denoting the number of relevant instances out of all the retrieved instances. It is expressed as in equation 5.1a, and can be described as the ratio of true positives (TP) and the sum true positives and false positives (FP), i.e., total instances labelled as belonging to the positive class. In our case, the classes are emotions such as happy, angry, sad, and neutral.

In the context of classification, a perfect precision score of 1.0 for a class, say, C means that every item labelled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labelled correctly).

5.3.2 Recall

Recall, also known as sensitivity, is the fraction of relevant instances actually retrieved. For classification, it can be expressed as in equation 5.1b, and can be described as the ratio of true positives and the sum of true positives and false negatives (FN), i.e., total number of instances that actually belong to the positive class.

For classification, a perfect recall of 1.0 means that every item from class C was labelled as belonging to class C (but says nothing about how many items from other classes were incorrectly also labelled as belonging to class C).

Model	**Source**	**BLEU 1**	**BLEU 2**	**BLEU 3**	**BLEU 4**
Sanabria et al. [111]	t	-	-	-	54.4
Sanabria et al. [111]	s-v-t	-	-	-	54.4
Lal et al., (2019)	s-v-t	-	-	-	51.0
Raunak et al., (2019)	t	-	-	-	55.5
Wu et al., (2019)	s-v-t	-	-	-	56.2
Seq2Seq	t	48.32	30.63	20.79	14.60
	s	20.11	7.01	3.12	1.57
	v	19.28	6.35	2.33	1.03
Seq2Seq + attn	t	79.21	67.34	52.67	47.34
Auto-Fusion (Ours)	s-t	56.31	33.82	24.63	21.45
	s-v-t	57.18	34.71	25.15	22.10
Auto-Fusion + attn (Ours)	s-t	80.34	67.83	61.27	55.01
	s-v-t	85.23	71.95	69.54	57.80
GAN-Fusion (Ours)	s-t	60.65	37.43	30.01	28.87
	s-v-t	61.23	38.76	31.23	29.31
GAN-Fusion + attn (Ours)	s-t	82.25	69.43	64.33	56.5
	s-v-t	**89.66**	**74.48**	**71.29**	**59.83**

Table 5.1: Results for machine translation experiments on the How2 dataset. 't', 's', 'v' represent the text, speech, and video modalities, respectively. '+ attn' shows the inclusion of word-level attention [81] to the model. **Note:** Model names in bold denote that the networks were implemented by the author, and the attention module attends to text only.

5.3.3 F-Score

Based on Precision and Recall's description, we can observe that we need to consider *both* the metrics to judge a prediction system in terms of relevance. The F-score (or the F1-score) metric considers both the Precision (P) and Recall (R) to compute a score. It takes the harmonic mean of precision and recall, as shown in equation 5.1c.

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.1a}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.1b}$$

$$F = 2 \times \frac{PR}{P + R} \tag{5.1c}$$

Model	**BLEU 4**	**Meteor**
Seq2Seq baseline	36.3	56.9
Aalto [47]	44.1	**64.3**
Auto-Fusion + attn (Ours)	42.31	61.7
GAN-Fusion + attn (Ours)	**44.23**	63.8

Table 5.2: Results for machine translation on the Multi30K dataset. All methods use 'v' and 't' as the source modalities except the unimodal Seq2Seq baseline, which uses only text. **Note:** Model names in bold denote that the networks were implemented by the author.

5.3.4 Bilingual Evaluation Understudy (BLEU)

Bilingual Evaluation Understudy (BLEU) [96] is a metric for evaluating the quality of machine-translated text from one natural language to another. We use BLEU to evaluate the quality of translated sentences. This metric has shown the highest correlation with human judgements on similar tasks [95, 16]. The calculation of the metric is quite inexpensive; it involves computing n-gram overlap between a generated sentence, and some good quality reference translations. These scores for individual sentences are then averaged over the complete corpus to obtain the final BLEU score. It is a real number between 0 and 1. More formally speaking, BLEU is nothing but a modified version of the precision metric to compare a candidate translation against multiple reference translations. We report BLEU 1, BLEU 2, BLEU 3, and BLEU 4 – which compute 1-gram, 2-gram, 3-gram, and 4-gram overlap between candidate and reference translations – for all our generation experiments. A higher value of BLEU for a task such as machine translation indicates better generation quality as it shows more overlap between the generated and ground truth text. Notably, the significance of improvement in BLEU-n score increases as we increase the value of n. For instance, a 0.04 point improvement in BLEU 4 is much more significant than a 0.04 point improvement in BLEU 1.

5.3.5 METEOR

METEOR [9] is another metric for evaluating machine translation systems that shows a high correlation with human judgement. It was designed to address some of the deficiencies inherent in the BLEU metric. While BLEU is a purely precision-based metric, METEOR uses the weighted harmonic mean of unigram precision and unigram recall, thereby, providing a better indication of translation quality.

Model	P	R	F	A
LSTM + attn (uni)	53.2	40.6	43.4	43.6
LSTM + attn (bi)	66.1	65.0	64.7	64.2
Yoon et al. [138]	-	-	-	71.8
Yoon et al.[137]	-	-	-	76.5
Auto-Fusion + attn (Ours)	75.3	77.4	76.3	77.8
GAN-Fusion + attn (Ours)	**77.3**	**79.1**	**78.2**	**79.2**

Table 5.3: Precision (P), Recall (R), F1-score (F), and Accuracy (A) scores for emotion recognition. The LSTM+attn (uni) baseline uses only text to predict emotions, while LSTM+attn (bi) uses inputs from the speech and text modality, and simple concatenation is used for fusion. **Note:** Model names in bold denote that the networks were implemented by the author.

5.4 Results

Results of our experiments on the How2, Multi30K and IEMOCAP dataset are shown in Tables 5.1, 5.2 and 5.3, respectively. For emotion recognition, we observe that our models perform well across all the evaluation metrics. For the relatively tricky task of machine translation, we note that our best performing model beats the existing methods in terms of BLEU scores and is competitive in terms of METEOR [9], despite being much lighter than the transformer-based baselines. Further, we note that including a word-level attention mechanism consistently improves performance. It is important to note that the attention module only attends to text, and not to the input from any other modality. We discuss some ablation studies in Chapter 6, revealing some interesting insights.

Chapter 6

Analysis

In this chapter, we discuss some ablation studies that reveal some interesting insights.

6.1 Robustness of multimodal features

It is very important that the learned multimodal latent features are robust, i.e., they should work reasonably well even in the presence of some noise. In order to measure robustness of the learned multimodal features, we conduct an ablation test on the How2 dataset. In this test, we randomly replace some tokens in the test sentence with the unknown token, `<UNK>`, and translate the sentence using our best performing model, GAN-Fusion + attn.

The ablation study reveals that the complementary modalities are able to compensate for nearly 30% of the missing text as beyond that, we see a sharp drop in the BLEU scores. This shows that our method does not rely just on the textual description for translation. It also follows that the learned joint representation indeed contains rich information from other modalities, which can compensate for the absence of information from some other modality, and enhance the capability of the system.

6.2 Comparison against other baselines

For the machine translation task on How2 dataset, we use multiple baselines including the pyramidal encoder-decoder framework presented in the How2 paper as well as some

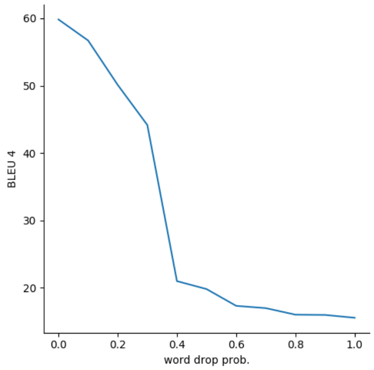


Figure 6.1: Ablation test on the How2 dataset. Word drop probability v/s BLEU 4. A sudden drop in BLEU scores as we move from 0.3 to 0.4 indicates that our model was able to compensate for $\sim 30\%$ of the missing text.

top performing models from the How2-challenge.[1] For the Multi30K dataset, we compare our models against the best performing system on the dataset, Aalto [47], a transformer-based network for machine translation. Despite being significantly less complex than the transformer networks, our method emerges as the best performing model in terms of BLEU scores on both the datasets, and it produces competitive results in terms of METEOR on the Multi30K dataset as well. This shows the benefits of using adaptive fusion techniques for complex multimodal tasks such as machine translation.

6.3 Contribution from each modality

Our extensive set of experiments on the How2 dataset also measure the effect of adding/removing input from one of the modalities. Table 5.1 reports results when using different combinations of source modalities. It reveals that while both acoustic and visual modalities contribute to enhanced translation, the video's contribution is slightly lower. This was also observed in some prior works such as Grönroos et al. [47]; however, it needs further validation as this might be addressed by simply using a better visual learner. Knowing the contribution from each modality will be crucial in understanding multimodal aspects of communication in detail. Our ablation study also suggests that we may have essential

[1] The How2 Challenge has three tasks: Speech Recognition, Machine Translation, and Summarization. We choose our baselines from the Machine Translation task. More information about the competition can be found here: https://srvk.github.io/how2-challenge/

signals from complementary modalities. Therefore, a more thorough set of experiments is needed to derive a concrete conclusion.

Chapter 7

Conclusions and Future Work

This chapter contains a summary of the proposed methodologies in this thesis and throws light on prospective directions of exploration.

7.1 Summary

In this thesis, we propose two adaptive fusion techniques that allow for effective multimodal fusion. Instead of “fixing” the fusion operation beforehand, we let the model decide “how” to extract signals from different modalities. Our results indicate that such adaptive models are more effective than their heavier counterparts, such as transformer networks. Moreover, the joint multimodal representations learned by these models are robust, which allows them to extract complementary signals from other modalities when partially deprived of information from one modality.

Our models achieve state-of-the-art results on many metrics and are highly competitive against existing systems on other metrics. It is, however, essential to remember that there are many less understood aspects as well to these networks. For instance, a more systematic analysis is required to measure the contribution of each modality. Similarly, the degree of information that the methods can extract for each modality is still unknown. The nature of the learned latent space of the joint distribution is also unexplored. Clearly, there is a vast scope of improvement for the current work in terms of understanding the nature of the learned models and multimodality. We discuss some possible directions to explore in the future that may aid in solving the current mysteries.

7.2 Future Directions

Following up on the discussion so far, we identify prospective aspects to multimodal deep learning for exploration. Out of all the exciting aspects of multimodal research, we believe feature alignment, i.e., aligning signals from each modality, and interpretability, i.e., understanding the learned latent space are the most important. While alignment might help address the heterogeneity in data, solving the latter problem may bear us crucial insights into such models' decision-making process. Currently, the dynamics of the learned latent space are not completely known. For instance, there still is not a concrete metric for measuring distance between two data points in the latent space. While most use the Euclidean distance, some argue the use of other metrics such as Riemmanian distance [21]. Unveiling this black-box will not only solve such trivial problems, it may also bear revolutionary insights about the functioning of neural networks, in general.

7.2.1 Better GANs

GANs, in general, exhibit numerous issues in practice. These issues, including inconsistency with originally claimed theoretical guarantees, are well-documented in the literature [7, 115, 30]. Recently, such implicit assumptions about GANs have been probed for better understanding [73]. Perhaps the most critical issue that was addressed was the mode-collapse problem. In Li et al. [75], the authors argue about the need to return to the principle of maximum likelihood, insisting on full recall, as opposed to generation by a network with unknown recall. They propose to use Implicit Maximum Likelihood Estimation (IMLE) for training GANs in order to increase the network's capacity as a whole. The application of IMLE on tasks such as high-resolution image synthesis [77] and multimodal image synthesis [76] achieve excellent results, and it would be interesting to see the effect of using IMLE for training the GAN-Fusion network proposed in this thesis.

7.2.2 Understanding Latent Space

Understanding of the latent space of the learned multimodal joint distribution remains an unsolved problem. A good understanding of the nature of such systems may pave ways for human inputs into the learning algorithms. Moreover, achieving true interpretability will unravel mysteries of the current "black-box" networks. Some initial works explain some interesting insights about the implicit behaviour of multimodal networks [44, 19]; however, real much remains to be explored. It would be interesting to see the results of similar systematic approaches for text generation.

We believe tackling the two aforementioned core tasks will help not only the researchers indulged in multimodal deep learning but the entire deep learning community as a whole. We look forward to advancing towards such a goal, one step at a time.

References

[1] Feedforward neural networks. https://www.cc.gatech.edu/~san37/post/dlhc-fnn/. Accessed: 2020-07-19.

[2] Mapping word embeddings with word2vec. https://towardsdatascience.com/mapping-word-embeddings-with-word2vec-99a799dc9695. Accessed: 2020-07-19.

[3] Vgg16 convolutional network for classification and detection. https://neurohive.io/en/popular-networks/vgg16/. Accessed: 2020-07-19.

[4] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[5] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[7] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl work-*

shop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pages 65–72, 2005.

[10] Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, 2019.

[11] Sonia Barrios, David Buldain, María Paz Comech, Ian Gilbert, and Iñaki Orue. Partial discharge classification using deep learning methodssurvey of recent progress. *Energies*, 12(13):2485, 2019.

[12] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[13] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[14] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335, 2008.

[15] Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1989–1998, 2019.

[16] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[17] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[18] Sarath Chandar, Mitesh M Khapra, Hugo Larochelle, and Balaraman Ravindran. Correlational neural networks. *Neural computation*, 28(2):257–285, 2016.

[19] Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. Do explanations make vqa models more predictable to a human? *arXiv preprint arXiv:1810.12366*, 2018.

[20] Jie Chen, Gang Liu, and Xin Chen. Animegan: A novel lightweight gan for photo animation. In *International Symposium on Intelligence Computation and Applications*, pages 242–256. Springer, 2019.

[21] Nutan Chen, Francesco Ferroni, Alexej Klushyn, Alexandros Paraschos, Justin Bayer, and Patrick van der Smagt. Fast approximate geodesics for deep generative models. In *International Conference on Artificial Neural Networks*, pages 554–566. Springer, 2019.

[22] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[23] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[24] Yu-An Chung and James Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *arXiv preprint arXiv:1803.08976*, 2018.

[25] Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. Unsupervised cross-modal alignment of speech and text embedding spaces. In *Advances in Neural Information Processing Systems*, pages 7354–7364, 2018.

[26] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[27] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[28] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[29] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[30] Robert Cornish, Hongseok Yang, and Frank Wood. Towards a testable notion of generalization for generative adversarial networks, 2018. In *URL https://openreview.net/forum.*

[31] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[32] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[34] David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, et al. Simsensei kiosk: A virtual human interviewer for healthcare decision support. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1061–1068, 2014.

[35] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[36] Arianna DUlizia. Exploring multimodal input fusion strategies. In *Multimodal Human Computer Interaction and Pervasive Services*, pages 34–57. IGI Global, 2009.

[37] D. Elliott, S. Frank, K. Sima'an, and L. Specia. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, 2016.

[38] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[39] Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. Jointly optimizing diversity and relevance in neural response generation. In *NAACL-HLT 2019*, March 2019.

[40] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[41] Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*, 2015.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[44] Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. Towards transparent ai systems: Interpreting visual question answering models. *arXiv preprint arXiv:1608.08974*, 2016.

[45] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[46] Edward Grefenstette, Phil Blunsom, Nando De Freitas, and Karl Moritz Hermann. A deep architecture for semantic parsing. *arXiv preprint arXiv:1404.7296*, 2014.

[47] Stig-Arne Grönroos, Benoit Huet, Mikko Kurimo, Jorma Laaksonen, Bernard Merialdo, Phu Pham, Mats Sjöberg, Umut Sulubacak, Jörg Tiedemann, Raphael Troncy, et al. The memad submission to the wmt18 multimodal translation task. *arXiv preprint arXiv:1808.10802*, 2018.

[48] Albert Haque, Michelle Guo, Prateek Verma, and Li Fei-Fei. Audio-linguistic embeddings for spoken sentences. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7355–7359. IEEE, 2019.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[50] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[51] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[52] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[53] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[55] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[56] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[57] Lifu Huang, Kyunghyun Cho, Boliang Zhang, Heng Ji, and Kevin Knight. Multilingual common semantic space construction via cluster-consistent word embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 250–260, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[58] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8779–8788, 2018.

[59] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[60] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[61] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy, July 2019. Association for Computational Linguistics.

[62] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[63] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[64] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[65] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[66] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[67] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[69] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[70] Oren Z Kraus, Ben T Grys, Jimmy Ba, Yolanda Chong, Brendan J Frey, Charles Boone, and Brenda J Andrews. Automated analysis of high-content microscopy data with deep learning. *Molecular systems biology*, 13(4):924, 2017.

[71] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[72] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*, 2019.

[73] Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *ArXiv*, abs/1809.09087, 2018.

[74] Ke Li and Jitendra Malik. On the implicit assumptions of gans. *ArXiv*, abs/1811.12402, 2018.

[75] Ke Li and Jitendra Malik. On the implicit assumptions of gans. *arXiv preprint arXiv:1811.12402*, 2018.

[76] Ke Li, Shichong Peng, Tianhao Zhang, and Jitendra Malik. Multimodal image synthesis with conditional implicit maximum likelihood estimation. *arXiv preprint arXiv:2004.03590*, 2020.

[77] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4220–4229, 2019.

[78] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS18, page 15051514, Red Hook, NY, USA, 2018. Curran Associates Inc.

[79] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2247–2256, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[80] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

[81] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[82] Jiaxin Ma, Hao Tang, Wei-Long Zheng, and Bao-Liang Lu. Emotion recognition using multimodal residual lstm network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 176–183, 2019.

[83] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[84] Iain Matthews, Timothy F. Cootes, Andrew Bangham, Stephen Cox, and Richard Harvey. Extraction of visual features for lipreading. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24:198 – 213, 03 2002.

[85] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[86] Sparsh Mittal. A survey of fpga-based accelerators for convolutional neural networks. *Neural computing and applications*, pages 1–31, 2020.

[87] Sunil S. Morade and Suprava Patnaik. Comparison of classifiers for lip reading with cuave and tulips database. *Optik - International Journal for Light and Electron Optics*, 126(24):5753–5761, 2015.

[88] Tim Morris. *Computer vision and image processing*. Palgrave Macmillan, 2004.

[89] Michael C Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex systems*, 3(4):349–381, 1989.

[90] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[91] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[92] Bojar Ondrej, Rajen Chatterjee, Federmann Christian, Graham Yvette, Haddow Barry, Huck Matthias, Koehn Philipp, Liu Qun, Logacheva Varvara, Monz Christof, et al. Findings of the 2017 conference on machine translation (wmt17). In *Second Conference onMachine Translation*, pages 169–214. The Association for Computational Linguistics, 2017.

[93] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[94] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[95] Kishore Papineni, Salim Roukos, Todd Ward, John Henderson, and Florence Reeder. Corpus-based comprehensive and diagnostic mt evaluation: Initial arabic, chinese,

french, and spanish results. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT 02, page 132137, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[96] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[97] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[98] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[99] Eric K Patterson, Sabri Gurbuz, Zekeriya Tufekci, and John N Gowdy. Cuave: A new audio-visual database for multimodal human-computer interface research. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–2017. IEEE, 2002.

[100] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[101] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[102] Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6892–6899, 2019.

[103] Hiranmayi Ranganathan, Shayok Chakraborty, and Sethuraman Panchanathan. Multimodal emotion recognition using deep learning architectures. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[104] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14866–14876, 2019.

[105] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[106] AJ Robinson and Frank Fallside. *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering Cambridge, MA, 1987.

[107] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[109] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[110] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[111] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: A large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*, 2018.

[112] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[113] Yuge Shi, N Siddharth, Brooks Paige, and Philip Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In *Advances in Neural Information Processing Systems*, pages 15718–15729, 2019.

[114] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[115] Mathieu Sinn and Ambrish Rawat. Non-parametric estimation of jensen-shannon divergence in generative adversarial network training. In *International Conference on Artificial Intelligence and Statistics*, pages 642–651, 2018.

[116] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986.

[117] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[118] Kihyuk Sohn, Wenling Shang, and Honglak Lee. Improved multimodal deep learning with variation of information. In *Advances in Neural Information Processing Systems*, pages 2141–2149, 2014.

[119] Lucia Specia, Stella Frank, Khalil Simaan, and Desmond Elliott. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, 2016.

[120] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[121] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[122] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[123] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[124] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

[125] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[126] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[128] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[129] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016.

[130] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[131] Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, 1(4):339–356, 1988.

[132] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. *arXiv preprint arXiv:1809.00812*, 2018.

[133] Kouichi Yamaguchi, Kenji Sakamoto, Toshio Akabane, and Yoshiji Fujimoto. A neural network for speaker-independent isolated word recognition. In *First International Conference on Spoken Language Processing*, 1990.

[134] Harry Yang. super-resolution using gan. https://github.com/leehomyc/Photo-Realistic-Super-Resoluton, 2016.

[135] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[136] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in neural information processing systems*, pages 1031–1042, 2018.

[137] Seunghyun Yoon, Seokhyun Byun, Subhadeep Dey, and Kyomin Jung. Speech emotion recognition using multi-hop attention mechanism. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2822–2826. IEEE, 2019.

[138] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 112–118. IEEE, 2018.

[139] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8807–8817, 2019.

[140] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[141] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.