Coupled Deep Learning for Multi-Modal Retrieval

Student Name: Sumit Keswani
Roll Number: 2013111

BTP report submitted in partial fulfillment of the requirements for the Degree of B.Tech. in Computer Science & Engineering

BTP Track: Research

BTP Advisor
Dr. Richa Singh
Dr. Mayank Vatsa

Indraprastha Institute of Information Technology
New Delhi
Student’s Declaration

I hereby declare that the work presented in the report entitled “Coupled Deep Learning for Multi-Modal Retrieval” submitted by me for the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of Dr. Richa Singh and Dr. Mayank Vatsa. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

.............................. Place & Date: ............................
Sumit Keswani

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

.............................. Place & Date: ............................
Dr. Richa Singh
Work Distribution

Semester 1: August 2016 - December 2016

• Understood the problem and read about past approaches.
• Collected available datasets for experiments.
• Implemented previous papers and reproduced the state of the art results.
• Implemented a sensible baseline that will be considered as a benchmark for future work.

Semester 2: January 2017 - April 2017

• Proposed a novel coupled deep learning architecture for the problem.
• Proposed a new training method for the proposed joint network and addressed various intricacies faced during training.
• Implement the proposed architecture and try different variations.
• Analyse the experiments on an open large scale dataset and compare it’s performance against existing models.
Abstract

In past few years, cross-modal information retrieval has drawn much attention due to significant growth in the multimodal data. It takes one type of data as the query to retrieve relevant data of multiple modalities. For example, a user can use a text to retrieve relevant pictures or videos. Since the query and its retrieved results can be of different modalities, how to measure the content similarity between different modalities of data remains a challenge. The existing solutions try to project data from different modalities into a common latent space and then learn a independent mapping from one modality to another. In this paper, we propose a novel fully-coupled deep learning architecture that can effectively exploit the inter-modal and intra-modal associations from heterogeneous data. The proposed learning objective can capture the correlations between the cross-modal data while preserving the intra-modal relationships. We also propose a training method that uses expectation maximization for learning the mapping function from one modality to other. The proposed training method is memory efficient and large training datasets can be split into mini-batches for parameter updations.

Keywords: Coupled Deep Learning, Alternate Minimization, Information Retrieval, Machine Learning, Auto-encoder
Chapter 1

Introduction

In this section, we will begin with the motivation of our problem and the impacts it can have over a variety of applications. Then, we move on to formally define our target problem with some understanding of complexity involved in the process. The following sections cover some basic deep learning architectures and other techniques that will be required to understand some of the related work explained in chapter 2 of the report.

1.1 Motivation

Over the last few years, the volume of different types of media data such as text, images, videos etc. have grown significantly due to the prevalence of social-media and other media platforms. For instance, Twitter recently reported that over 340 million tweets were sent each day\footnote{https://blog.twitter.com/2012/twitter-turns-six} while Facebook reported that around 300 million photos were created each day\footnote{https://goo.gl/UK85DH}. This explosive growth have created endless opportunities of exploiting insights and contextual observations in the data. The heterogeneity of the data, however, has come out as a barrier to successful retrieval of cross-media data. There has been an extensive research on the intra-modal relationships within the data but it hardly captures the context information that can be utilized for other types of media.

Automated description of images is one of the ultimate goals of computer vision which require learning models over text and its corresponding images. But the limited availability of such paired data is a problem for a progressive research. Obtaining very high correlation between the existing independent datasets of text and images can make such existing datasets useful for the research.

The traditional approaches learn the connection between multiple modalities by direct utilization of hand-crafted low-level heterogeneous features and the learned correlation are
merely constructed in terms of high-level feature representation. To well exploit the intrinsic structures of multimodal data, it is essential to build up an interpretable correlation between multimodal data which is very difficult to get from low-level features. Therefore, there is a need to come up with an efficient solution that can construct abstract high-level features out of heterogeneous data and can harness it to obtain correlation between data belonging to different modalities and hence, serving our motivation.

1.2 Query Based Information Retrieval

Content based retrieval has been a core component for many media platforms that provide different services across the Internet. As soon as a query is made, the ranked data is examined against the query for the relevance and is accordingly returned as the result. The hypothesis is that the query should fetch the appropriate data independent of the media type. So, the search within the data can belong to one of the following two categories:

1. **Intra-modal Search**: There has been a lot of study for searching data belonging to the same modality as the query. This is a relatively simple problem as both the query and data lie in the same latent space. Several efficient techniques are already available for various media types and is therefore not the main concern of this project. Examples of this type of search include web document retrieval based on keyword query, content based image retrieval and song retrieval against a sound query.

2. **Cross-modal search**: This type of search aims at retrieving the relevant information from the data belonging to a different modality. This is a hard problem as searching across the data from different latent space needs context of the query in being able to fetch similar results. Hence, coming up with a novel approach to solve the problem of cross-modal search will be the prime objective of this project. Examples for this type of search includes voice based search of images, searching images based on text query etc.

1.3 Problem Statement

As stated above, cross-modal data retrieval needs to learn the context of the query in order to search for the relevant information. This is true for all types of media data but images and text are two of the most prevalent media types and account for most of the data volume across the internet. So, for this project, we will narrow our scope down to two modalities - images and text.
Now, we are ready to formally define our problem statement:

Given two databases $\text{db}^T$ and $\text{db}^I$ of text and images respectively with $T_i$ denoting $i^{th}$ text in $\text{db}^I$ and $\text{im}_i$ denoting $i^{th}$ image in $\text{db}^I$ for $i \in \{1, 2, 3, \ldots\}$.

We define a function $\eta$ that takes 2 arguments - text and image and returns the measure of the closeness of the two entities. By closeness, we mean how much related the two entities are. For example, the text "A dog is walking in the park" will be highly related to an image that shows a dog walking in a park.

Our objective is to learn two mapping function $\phi_{IT}$ from image to text and $\phi_{TI}$ from text to image that can be defined as:

$$\phi_{IT} (\text{im}) = \text{text}$$
$$\text{s.t}, \text{text} = \max \arg T_i \{ \eta(\text{im}, T_i) \}$$

$$\phi_{TI} (\text{text}) = \text{image}$$
$$\text{s.t}, \text{image} = \max \arg T_i \{ \eta(\text{text}, \text{im}_i) \}$$

where, $T_i$ is the $i^{th}$ text in $\text{db}^T$, $\text{im}_i$ is the $i^{th}$ image in $\text{db}^I$, for $i \in \{1, 2, 3 \ldots\}$.

In other words, we want to search the closest text from a query image and vice-versa. The complete problem can be broken down in four following steps:

1. Identify the various known objects of interest in the image. Known objects here refer to those objects that have been seen in paired data.

2. Identifying the various phrases of interest within a text. The phrases should have been observed paired with the images of similar contexts.

3. Learning a good representation of both textual and image data and be able to recreate the original data from it.

4. Identifying the correlation between the language contexts and image contexts.
1.4 Convolutional Neural Networks

Convolutional Neural Networks (and especially deep-CNN) have shown great success since the first deep-CNN (called AlexNet) was introduced by Alex [1]. This architecture has been able to extract the spatial information within an image to recognize the objects of interest within an image. The architecture, as any other deep net, is able to learn more abstract features at every level and therefore able to reduce the significant features within the image.

It has specialized connectivity structure, which usually consists of multiple convolutional layers followed by fully connected layers. These layers form stacked, multiple-staged feature extractors, with higher layers generating more abstract features from lower ones. After every layer, the volume obtained can also be sampled for reduction in dimension. On top of the feature extractor layers, there is a classification layer that can be fine-tuned for data-specific classification.

As depicted in Figure 1.2, the input to a CNN is raw image pixels such as an RGB vector, which is forwarded through all feature extractor layers to generate a feature vector that is a high-level abstraction of the input data. The training data of a CNN consists of image-label pairs. Let \( x \) denote the image raw feature and \( f_I(x) \) the feature vector extracted from CNN. \( t \) is the binary label vector of \( x \). If \( x \) is associated with the \( i \)th label \( l_i \), \( t_i \) is set to 1 and all other elements are set to 0. \( f_I(x) \) is forwarded to the classification layer to predict the final output \( p(x) \), where \( p_l(x) \) is the probability of \( x \) being labelled with \( l_i \). Given \( x \) and \( f_I(x) \), \( p_l(x) \) is defined as:

\[
p_l(x) = \frac{e^{f_l(x)_i}}{\sum_j e^{f_l(x)_j}}
\]
which is a softmax function. Based on Equation 9, we define the prediction error, or softmax loss as the negative log likelihood:

\[ L_i(x, t) = - \sum_i t_i \log p_i(x) \]

### 1.5 Stacked Autoencoder

As explained in [10], an autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. I.e., it uses \( y^{(i)} = x^{(i)} \). A simple autoencoder often ends up learning a low-dimensional representation of data very similar to PCAs and therefore is a popular model for unsupervised dimension reduction. Stacked Autoencoder stacks these vanilla autoencoders and tries to learn a deep encoding of the input feature vector. Each layer of an autoencoder is backpropagated assuming that the reconstructed feature is the same as input feature.

The stacked autoencoder tries to learn a function \( h_{W,b}(x) \approx x \). In other words, it is trying to learn an approximation to the identity function, so as to output \( \hat{x} \) that is similar to \( x \). The identity function seems a particularly trivial function to be trying to learn; but by placing constraints on the network, such as by limiting the number of hidden units, we can discover interesting structure about the data.
Chapter 2

Related Work

In this section we discuss some of the previous approaches taken for multi-modal data retrieval. We will begin with a Topic model that has been applied to specific cross-modal problems - Latent Dirichlet Allocation (LDA). LDA can be extended to learn the joint distribution of multi-modal data and thus can be used to find correlation between image and text. We will then move to some deep learning methods that are currently the state-of-the-art. The first one is an auto-encoder based deep learning architecture that first finds the latent representation of intra-modal data and then tries to correlate that data in different modality. The second architecture introduces a trace-norm objective function that is used to optimize the correlation between image and text.

2.1 General Framework

Figure 2.1 presents the general framework of cross-modal retrieval as described in [9], in which, feature extraction for multimodal data is considered as the first step to represent various modalities of data. Based on these representations of multimodal data, cross-modal correlation modeling is performed to learn common representations for various modalities of data. At last, the common representations enable the cross-modal retrieval by suitable solutions of search result ranking and summarizing.

2.2 Topic Model - LDA

Latent Dirichlet Allocation is a topic model that has been widely studied and used in various multi-modal problems. It has also been used as a supervised model for dimensionality reduction. [2] introduces Correspondence LDA that can achieve simultaneous dimension-
Figure 2.1: General Framework for cross-modal retrieval

Reducing the representation of region descriptions and words, while also modeling the conditional correspondence between their respective reduced representations.

The model can be viewed in terms of a generative process that first generates the region descriptions and subsequently generates the caption words. In particular, it first generates N region descriptions \( r_n \) from an LDA model. Then, for each of the M caption words, one of the regions is selected from the image and a corresponding caption word \( w_m \) is drawn, conditioned on the factor that generated the selected region.

Figure 2.2: General Framework for cross-modal retrieval
2.3 Deep Learning - Identifying latent features

As we mentioned above, it is common that different types of data are used for description of the same events or topics in the web. This makes it very challenging for traditional methods to obtain a joint representation for multimodal data. Inspired by the recent progress in deep learning, [3] provides an end-to-end framework that first finds the latent representation of intra-modal data and then tries to map different modality in a common latent space.

Figure 2.3: Flowchart of training. Relevant images (or text) are associated with the same shape. In single-modal training, objects of same shape and modality are moving close to each other. In multi-modal training, objects of same shape from all modalities are moving close to each other.

Considering the Figure 2.3, data belonging to each modality goes through 2 stage training process:

1. **Training Stage I**: Unsupervised intra-modal latent feature representation of image and text in which each latent features are learned using stacked auto-encoders separately for each modality. This helps in mapping each data to a common latent space and thus making it possible to compare with the data from other modality.

2. **Training Stage II**: Multi-modal learning of the model that tries to minimize the Euclidean distance between the data between 2 modalities i.e. image and text for our case.
By taking into account both intra-modal and cross-modal training, the general objective function for the framework is:

\[ \mathcal{L} = \beta_I \mathcal{L}_I + \beta_T \mathcal{L}_T + \mathcal{L}_{I,T} + \xi(\theta) \]

Here, \( \mathcal{L}_I \): loss in intra-modal training for images, \\
\( \mathcal{L}_T \): loss in intra-modal training for text, \\
\( \mathcal{L}_{I,T} \): loss in cross-modal training

### 2.4 Deep Correlation for cross-modal entities

Another deep learning framework proposed by [4] introduces a trace-norm objective function to find the correlation between an image and its corresponding text. The overall framework is given in the Figure 2.4:

![Deep Correlation architecture for Matching Images and Text](image)

The image is fed into a deep Convolutional Neural Network (CNN) with layers of convolution, ReLU as activation function and followed by a fully connected layer. The text on the other hand is converted to an TFIDF feature vector and then fed into a network consisting of stacked triplets of fully connected layer, ReLU layer and dropout layer.

Finally, Given two sets of \( m \) random vectors \( X \in \mathbb{R}^{d_{X \times m}} \) and \( Y \in \mathbb{R}^{d_{Y \times m}} \), let their covariances be \( \Sigma_{xx} \) and \( \Sigma_{yy} \) respectively, and let the cross covariance be \( \Sigma_{xy} \). Canonical correlation analysis (CCA) seeks pairs of linear projections that maximise the correlation of the two views as:

\[
(w_x^*, w_y^*) = \arg\max_{w_x, w_y} \text{corr}(w_x^T X, w_y^T Y)
\]

\[
= \arg\max_{w_x, w_y} \frac{w_x^T \Sigma_{xy} w_y}{\sqrt{w_x^T \Sigma_{xx} w_x w_y^T \Sigma_{yy} w_y}}
\]

The gradients are then finally computed and propagated down along the two branches of the given network.
Chapter 3

Coupled Deep Learning and Latent Feature Space Learning

This chapter describes the coupled deep learning framework and how the network is trained to learn a mapping between heterogeneous data. In section (3.1), we present the formulation of our defined problem and explain how each component of it addresses the part of the problem we are trying to optimize. Then in section 3.2.1, we explain how the network is jointly trained to simultaneously learn latent representation of each data type along with the mapping between their latent space representation. Further, training initialization and regularization is discussed in sections (3.2.2) and (3.2.3) respectively. Finally, we address the problem of Network Poisoning during training and suggest a regularization that can be added to prevent it.

Figure 3.1: Block diagram for problem formulation
### 3.1 Problem Formulation

Let image set $X = [x_1, x_2, x_3, ..., x_n] \in \mathbb{R}^{d_1 \times n}$ and $Y = [y_1, y_2, y_3, ..., y_n] \in \mathbb{R}^{d_2 \times n}$ be $n$ unlabelled pairs of data points extracted from two different modalities, with dimensions $d_1$ and $d_2$ respectively. We try to formulate the mapping as the following overall Loss function:

$$Loss = \alpha \cdot \{L_{\text{img}}(X, Z_X) + L_{\text{txt}}(Y, Z_Y)\} + \beta \cdot \{L_{\text{coupled}}(Z_X, Z_Y)\} + \epsilon(\theta) \quad (3.1)$$

In equation (3.1), $L_{\text{img}}$ and $L_{\text{txt}}$ refer to the intra-modal loss of the network during the training. The $Z_X$ and $Z_Y$ are the learnt latent representations of the images and text respectively. $L_{\text{coupled}}$ refers to the coupling loss between the two learnt representations and controls the mapping from one modality to other. Finally, $\epsilon(\theta)$ is used for the activity regularization during the training. It has been observed that assigning different weights for different losses (both intra and cross modal) provides better performance. For experiments, the modality with lower input quality is assigned lower weight in the objective function. This also enforces inter-modal constraints due to this. After defining the structure of our loss function, we can look at each part of the function separately.

![Proposed coupled deep learning architecture](image)

**Figure 3.2:** Proposed coupled deep learning architecture. (a) The blue and orange figures are the autoencoders to learn the latent representations $Z_X$ and $Z_Y$, of images and text respectively. (b) The transformations between $Z_X$ and $Z_Y$ learns the cross modal mapping.

Fig (3.1) provides a block diagram of the proposed coupled network. Now, from equation (3.1), the intra modal losses $L_{\text{img}}$ and $L_{\text{txt}}$ are the reconstruction errors of the two stacked autoencoders as in the fig (3.1). For experiment purposes, we enforce sparsity in the latent
representations by using L1 regularization for both the auto encoders as in [5]. Moreover, the objective functions for both the autoencoders is mean-squared error. Hence, the intra-modal loss in the equation (3.1) becomes:

\[ L_{\text{img}} = \| X - X' \|^2 + L_1(\theta) \] (3.2)

Here, X is the input image feature vector and X’ is the reconstructed feature vector by autoencoder. Similarly for text, Y and Y’ are the input and reconstructed text features respectively.

\[ L_{\text{txt}} = \| Y - Y' \|^2 + L_1(\theta) \] (3.3)

Now, for the cross-modal loss, we want to couple the two networks in such a way that one tries to train the other and vice versa. Hence, we define the \( L_{\text{coupled}} \) as the mean square loss between \( Z_X \) and transformed \( Z_Y \) and between \( Z_Y \) and transformed \( Z_Z \). Therefore, the \( L_{\text{coupled}} \) is defined as:

\[ L_{\text{coupled}} = \| Z_X - M_Y Z_Y \|^2 + \| Z_Y - M_X Z_X \|^2 + \epsilon(\theta) \] (3.4)

Hence, substituting the values in equation (3.1), we get our overall loss function as:

\[ L = \alpha \{ \| X - X' \|^2 + \| Y - Y' \|^2 \} + \beta \{ \| Z_X - M_Y Z_Y \|^2 + \| Z_Y - M_X Z_X \|^2 \} + \epsilon(\theta) \] (3.5)

The L1 regularizers for stacked autoencoders ensures highly abstract sparse representations of data and coupling loss regulates the mapping between the two latent representations. In the next subsection, we will detail the optimization process at the training stage.

### 3.2 Training Method

This section describes the training method we propose to train the coupled network. Later, different initialization technique for the network is discussed and what we have used for our experimental purposes. Finally, an important problem has been addressed that can occur during training using the proposed training method and how can it be overcome.
3.2.1 Alternate Minimization for Training

The loss in equation (3.4) is responsible for learning a transformation from one modality to another. While the mapping function is not a convex function jointly in $M_Y$, $Z_Y$, $M_X$ and $Z_X$, each direction of transformation loss is convex given the target vector is fixed. Hence, we use an Alternate Minimization approach to train our coupled network. The idea behind alternate minimization is to alternately train the two joint networks by each other. So, after every round, both the learnt representations by the networks gets updated according to the learnt representation of its counterpart.

3.2.2 Pre-training: Initialization for coupled networks

One of the major problems that can occur during the training of such coupled networks is the initialization. Since the cross-modal mapping is learnt only at the intersection of both the networks, a bad initialization can lead the network to learn poor representation of the original data especially when the weight of cross modal loss function is more than than intra-modal loss. Hence, a good initialization for the two networks is extremely important. The idea behind a good initialization will be that the initial latent representation of both the modalities should be of high quality capturing the important abstract features of the data. Hence, initially, we let the two autoencoders train independently only with the loss of reconstruction error. Once a rich representation is learnt by the network, it can be further be updated according to the other network learning a mapping form one feature to other.

Algorithm 1 and Algorithm 2 provide the pre-training and alternate minimization methods proposed in the paper:

Algorithm 1: Pre-training for coupled network

| Result: Initialized $Z_X$, $Z_Y$ |
| imageAE $\leftarrow$ deepModel(input layer $=$ $X$, output layer $=$ $X'$); |
| textAE $\leftarrow$ deepModel(input layer $=$ $Y$, output layer $=$ $Y'$); |
| loss $\leftarrow$ mean_squared_error + L1 regularization; |
| while not imageAE convergence do |
| $\mid$ $Z_X$ $\leftarrow$ imageAE.train(image features, image features); |
| end |
| while not textAE convergence do |
| $\mid$ $Z_Y$ $\leftarrow$ textAE.train(text features, text features); |
| end |
Algorithm 2: Alternate Minimization for coupled network training

Result: Learnt representations $Z_X$, $Z_Y$

$Z_X$, $Z_Y \leftarrow$ initialized through pre-training;
imageModel $\leftarrow$ deepModel(input layer = X, output layers = [X', $Z_X'$]);
textModel $\leftarrow$ deepModel(input layer = Y, output layers = [Y', $Z_Y'$]);
loss $\leftarrow$ [mean squared error, mean squared error];

while convergence is not reached do
    while $k_1$ iterations do
        $A \leftarrow$ image training batch;
        $B \leftarrow$ text training batch;
        init$Z_X \leftarrow$ textModel.predict($B$);
        imageModel.train($A$, [$A$, init$Z_X$])
    end
    while $k_2$ iterations do
        $A \leftarrow$ image training batch;
        $B \leftarrow$ text training batch;
        init$Z_Y \leftarrow$ imageModel.predict($A$);
        textModel.train($B$, [$B$, init$Z_Y$])
    end
end
$Z_X$, $Z_Y \leftarrow$ init$Z_X$, init$Z_Y$

3.2.3 Network Poisoning

This section addresses an important problem that may occur during the training of the proposed coupled network - Network poisoning. We define network poisoning for the alternate minimization updates as when one of the activities/outputs of network tends to zero, due to alternate minimization, the coupled network also starts reducing to zero vector. As a result, with every round of updating, the latent representations of data from both the modalities will end up to produce zero vectors. Something to notice from our overall objective function equation (3.1), the objective will achieve a local minima at $Z_X$, $Z_Y = 0$. As a result, the network may be poisoned. In order to prevent vectors to become zero vectors, negative log det regularization is popularly used as described in [6]. Hence, an activity regularization is added in our objective function as included in equation (3.5).
Chapter 4

Experimental analysis

This chapter describes the experimental setup and results obtained by the implementation of existing architectures for the multi-modal retrieval. The two architectures, as described in the section 2.3 and section 2.4, are used in order to avoid low-level hand-crafted feature extraction from text and images.

4.1 Dataset used

Imageclef IAPR TC-12, as described in [7], is a benchmark dataset for multi-modal retrieval with the image collection of 20,000 still natural images taken from locations around the world. The dataset includes pictures of different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life.

Each image is associated with a manual text caption in up to three different languages (English, German and Spanish). For the purpose of experiments, we have only considered English text description of the images. An example of an image and description is given as following:
There are other larger datasets of images and captions (e.g., Flickr8K, Flickr30K etc.) but IAPR TC-12 Benchmark dataset captions are much more detailed with the average caption length of 28.2 words as compared to 12.9 and 14.4 words in Flickr8K and Flickr30K respectively and hence, serving for our purpose. For IAPR TC-12 dataset, since there is only single pairs of text and images, avg MAP becomes the average of 1/rank of retrieved document.

4.2 Evaluation Metrics

We evaluate the effectiveness of the mapping mechanism by measuring the accuracy of the multi-modal search, i.e., $Q_{q \rightarrow t}$ ($q, t \in T, I$), using the mapped latent features. We use Mean Average Precision (MAP) as described in [8], one of the standard information retrieval metrics, as the major evaluation metric. Given a set of queries, the Average Precision (AP) for each query $q$ is calculated as,

$$AP(q) = \frac{\sum_{k=1}^{R} P(k) \delta(k)}{\sum_{j=1}^{R} \delta(j)}$$

where R is the size of the test dataset; $\delta(k) = 1$ if the k-th result is relevant, otherwise $\delta(k) = 0$; $P(k)$ is the precision of the result ranked at position k, which is the fraction of true relevant documents in the top k results. By averaging AP for all queries, we get the MAP score. The larger the MAP score, the better the performance.
4.3 Using Pre-Trained models

Transfer learning is a popular technique that can utilize the training of a neural network on large datasets available [1]. It can very well deal with the problems of having small dataset and time limitations. In this experiment, we have used pre-trained models of Convolutional Neural Network(VGG19 available at [1] and skip-gram model(model available at [2]) for image and text respectively.

For the MSAE architecture as in section 2.3, that uses autoencoders for dimensionality reduction, an image is passed to the pre-trained CNN model and the output feature vector(FC7 features) are passed to autoencoders which are then used for dimensionality reduction. This comes under the stage-1 of training.

On the other hand, for the deep correlation architecture, the image and text are supplied to the pre-trained model and an additional layer is added at the end of the two streams of neural networks. The objective function, as described in the sections 2.4, tries to maximize the correlation between the two feature vectors. The loss is calculated and gradient is back-propagated in order to finetune the network.

[1] https://gist.github.com/baraldilorenzo/8d096f48a1be4a2d660d
### Table 4.1: MAP table for IAPR TC-12 dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Image Annotation MAP</th>
<th>Image Retrieval MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSAE</td>
<td>0.397</td>
<td>0.391</td>
</tr>
<tr>
<td>Deep CCA</td>
<td>0.412</td>
<td>0.407</td>
</tr>
<tr>
<td>Cosine Proximity</td>
<td>0.402</td>
<td>0.406</td>
</tr>
<tr>
<td><strong>Coupled Deep Learning</strong></td>
<td><strong>0.439</strong></td>
<td><strong>0.426</strong></td>
</tr>
</tbody>
</table>

*The comparison for Coupled Deep Learning against all other methods for other available datasets is also currently going on. This will be updated as soon as the results are available.

### 4.4 Results

As given in Table 4.1, the proposed Coupled Deep Learning performs better than all other existing methods for both image annotation and image retrieval. The proposed architecture is able to learn better transformation using the proposed Alternate Minimization method as compared to learning independent transformations or deep CCA.

Mean average precision (MAP) for Deep Correlation gives the better results than MSAE for both image to text and text to image mapping. Also, image annotation has higher MAP value for most part as compared to Image retrieval. A possible reason is that the feature representation for images from CNN is better as compared to text representation from doc2vec.

We also tried using negative cosine proximity as the objective function instead of CCA in the architecture explained in section 2.4. The idea behind using cosine similarity is to minimize the difference in the distributions (in vectors) obtained from images and text even though the results obtained by DCCA are better than cosine proximity.
Bibliography


[5] Andrew Ng CS294A Lecture notes Sparse autoencoder


