Multi Task Learning Based Framework for Multimodal Classification

Danting Zeng
Stanford University, Stanford, USA
dandan_9817@hotmail.com

Abstract

Large-scale multi-modal classification aim to distinguish between different multi-modal data, and it has drawn dramatically attentions since last decade. In this paper, we propose a multi-task learning-based framework for the multimodal classification task, which consists of two branches: multi-modal autoencoder branch and attention-based multi-modal modeling branch. Multi-modal autoencoder can receive multi-modal features and obtain the interactive information which called multi-modal encoder feature, and use this feature to reconstitute all the input data. Besides, multimodal encoder feature can be used to enrich the raw dataset, and improve the performance of downstream tasks (such as classification task). As for attention-based multimodal modeling branch, we first employ attention mechanism to make the model focused on important features, then we use the multi-modal encoder feature to enrich the input information, achieve a better performance. We conduct extensive experiments on different dataset, the results demonstrate the effectiveness of proposed framework.

1 Introduction

With the easy-access of mobile devices, the world has witnessed the explosion of multimedia data, which contains various modalities, such as image, audio and text. Generally speaking, different modality can provide complementary information. However, many previous attempts focus on one single modality, as the multimodal data is more complex. The applications of multimodal data analysis seem to evident in several fields, such as, emotion recognition, medical diagnosis. Recently, the development of multimodal machine learning approaches has witnessed growing interest (Ngiam et al., 2011). On the other hand, deep learning has witnessed dramatically progress in various fields: ranges from computer vision, natural language processing and speech recognition (Oramas et al., 2018). Due to the great success of deep learning in single modality, great interests have been given for the multimodal deep learning framework (Xu et al., 2016; Radu et al., 2016). Despite of sustainable efforts have been made, multimodal deep learning is still far from been fully solved, using deep learning. Moreover, traditional approach train the classifiers on different modal and weighted average to generate the predictions, which is time-consuming and cannot model the interaction between different modal.

In this short paper, a general multimodal data classification task is proposed, leveraging multi task-based deep learning. The framework consists of two branches: multi-modal autoencoder branch and attention-based multi-modal modeling branch. The framework takes the interaction between different modals into consideration. To demonstrate the efficacy and robustness of proposed method, we conduct extensive experiments on different dataset and the results support our claims.

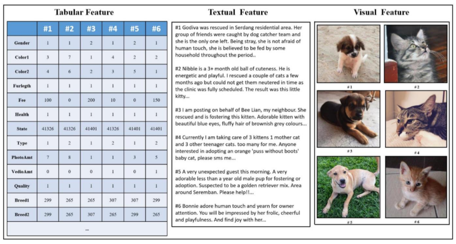
2 Dataset and Evaluation

In this paper, we use the Adoption Prediction Dataset from Kaggle[1] to do our research, which is a real world and challenging dataset. The dataset is composed of three different modal features: tabular features (the basic information about each pet), textual features (the pet description written by English) and visual features (the photo of each pet), and it aims to predict how quickly a pet is adopted. There are 14993 instances in this dataset, and the label is the categorical speed of adoption, there are five different classes from 0 to 4, in details, 0 means this pet was adopted on the same day as it was listed, 1 means this pet was adopted between 1 and 7 days after being listed. Figure 1 shows some example instances. Besides, in this classification task, due to the number of classes is balanced, we use accuracy

[1] https://www.kaggle.com/c/petfinder-adoption-prediction

Proceedings of the Third Workshop on Multimodal Artificial Intelligence, pages 30–35
June 6, 2021. ©2021 Association for Computational Linguistics

to evaluate different models' performance.



Tabular Feature

	#1	#2	#3	#4	#5	#6
Gender	1	1	2	1	2	1
Color1	3	7	1	4	2	2
Color2	4	6	2	3	5	1
Furlegth	1	1	1	1	1	1
Fee	100	0	200	10	0	150
Health	1	1	1	1	1	1
State	41326	41326	41401	41326	41401	41401
Type	1	2	1	2	1	2
PhotoAmt	7	8	1	1	3	5
VedioAmt	0	0	0	1	0	1
Quality	1	1	1	1	1	1
Breed1	299	265	265	307	307	299
Breed2	299	265	307	265	299	265
...						

Textual Feature

#1 Godiva was rescued in Serdang residential area. Her group of friends were caught by dog catcher team and she is the only one left. Being stray, she is not afraid of human touch, she is believed to be fed by some household throughout the period..

#2 Nibble is a 3+ month old ball of cuteness. He is energetic and playful. I rescued a couple of cats a few months ago but could not get them neutered in time as the clinic was fully scheduled. The result was this little kitty...

#3 I am posting on behalf of Bee Lian, my neighbour. She rescued and is fostering this kitten. Adorable kitten with beautiful blue eyes, fluffy hair of brownish grey colours...

#4 Currently I am taking care of 3 kittens 1 mother cat and 3 other teenager cats. too many for me. Anyone interested in adopting an orange 'puss without boots' baby cat, please sms me...

#5 A very unexpected guest this morning. A very adorable less than a year old male pup for fostering or adoption. Suspected to be a golden retriever mix. Area around Seremban. Please help!!...

#6 Bonnie adore human touch and yearn for owner attention. You will be impressed by her frolic, cheerful and playfulness. And find joy with her...

Figure 1: Six example instances from Adoption Prediction Dataset. The instance numbers are displayed as $\#1$ to $\#6$.

Tabular Features: These features are the basic information of each pet, there are 15 categorical variables and 4 continuous variables.

Textual Features: The textual features are the pet descriptions written by English.

Visual Features: The visual feature of each pet is a image whose size is from 240 pixels to 1024 pixels, in order to train our model, we reshape all the images to 512 × 512.

3 Proposed Approach

In this paper, our proposed approach has two parts: multi-modal autoencoder branch and attention-based multi-modal modeling branch.

3.1 Multi-modal Autoencoder

In the previous work, autoencoders receive a single modal feature and reconstitute it, with a goal to minimize the reconstruction loss between the input and output. However, if a task has multi-modal features, we can build a MMAE which can receive different modal features at the same time. MMAE first learns the encoder representation from each single modal feature, then concatenating them as a multimodal encoder feature, and finally this feature is used to reconstitute all the input. As can be seen in Figure 2, MMAE has two parts:

Input Layer: For the tabular features (represented as $x_{tabular}$), One-Hot Encoding for categorical variables and Max-Min Normalization for continuous variables. As for the visual features (represented as x_{visual}), we first reshape all the images size into 512×512, that is $x_{visual} = x_{visual}/255.0$. As for the textual features, every instance has a paragraph to describe the pet, for the i_{th} input paragraph with n words $w^i_1; w^i_2; :::; w^i_n$, we first padding the paragraph into fixed length $l = 100$.Then we us word embedding layer to transform paragraph into dense matrix X^i. All input paragraphs will be transformed into dense matrices whose size is 100×300, represented as $x_{textual}$. After the data preprocessing, the input layer will put $x_{tabular}$,x_{visual} and $x_{textual}$ into the next layer.

Multi-modal Interaction Layer: For each modal feature, we suppose $f(x)$ is the encoder function, $g(x)$ is the decoder function, in the previous work, we should build three independent autoencoders, each autoencoder can only encode a single modal feature. During encoding, the input data is compressed into a low dimensional vector, which we called encoder feature. During encoding, the autoencoder will reconstitute the input using encoder feature. The mathematical expressions are shown below:

$$h_{tabular} = f^1(x_{tabular}), \hat{x}_{tabular} = g^1(h_{tabular}) \tag{1}$$

$$h_{visual} = f^2(x_{visual}), \hat{x}_{visual} = g^2(h_{visual}) \tag{2}$$

$$h_{textual} = f^3(x_{textual}), \hat{x}_{textual} = g^3(h_{textual}) \tag{3}$$

where $h_{tabular}$,h_{visual} and $h_{textual}$are the encoder features of each modal input, they have the same length k, and during training, we minimize the reconstruction loss to optimize the parameters, the loss function is Mean Square Error (MSE). Take visual features as an example:

$$x_{visual} \approx \hat{x}_{visual} \tag{4}$$

In multi-modal interaction layer, in order to automatically obtain the interactive information between different modal features, we merge all the encoder features into a multi-modal encoder feature to reconstitute each input, rather than directly use corresponding encoder feature. In details, we first concatenate different encoder features to h_{mm}:

$$h_{mm} = [h_{tabular}; h_{visual}; h_{textual}] \tag{5}$$

Then $h_{mm} \in R^{1\times 3k}$ is used to decode all the inputs:

$$\overline{x}_{tabular}, \overline{x}_{visual}, \overline{x}_{textual} = g(h_{mm}) \tag{6}$$

In fact, this could be treated as multi-task learning, and the loss in **MMAE** is shown as bellow:

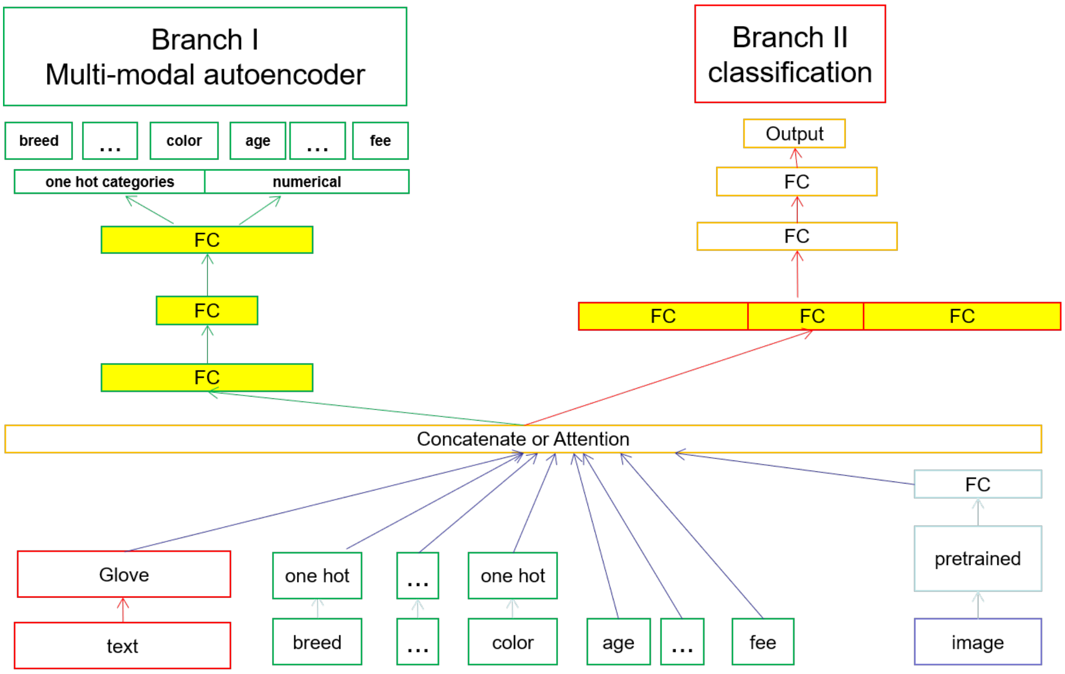


Figure 2: Framework of our solution.

$$Loss = \alpha * loss_{tabular} + \beta * loss_{visual} + \gamma * loss_{textual} \tag{7}$$

where $\alpha * loss_{tabular}$, $\beta * loss_{visual}$ and $\gamma * loss_{textual}$ are the reconstruction losses of different inputs, α, β and γ are the corresponding weights of different losses, they can adjust according to the practical scenario. In our experiments, we find that $\alpha = \beta = \gamma$ yields the best result. Besides, all the autoencoders in **MMAE** are four layers fully-connected neural networks. The multi-modal encoder feature we obtained from **MMAE** will be used in some downstream tasks to improve the performance, such as classification task.

3.2 Attention-based Multi-modal Modeling part:

In the previous work, a multi-modal model first receives different kind of inputs, then handles them separately to obtain high-level features, and do some simply interactions such as concatenate, finally a fully-connected layer is followed to get the prediction. However, in practical scenario, different modal features for a same task may have different importance, so simply concatenate those high-level features is not enough to help the model get key information. Inspired by the attention mechanism used in natural language processing and computer vision,we introduce attention mechanism into multi-modal model,which can make the model focus on the key information. Besides, we also add the multi-modal encoder feature from **MMAE** to enrich our input. The modeling part model mainly composed of four components:

Input Layer: This layer has the same function as the input layer in MMAE, so in this layer, we do the same thing as mentioned above.

Fully-Connected Layer and Convolutional Layer: In this layer, we use different neural networks for different input features. For the tabular features, we use a fully-connected layer to learn the high-level representation v_1, the activation function in each layer is ReLu (Glorot et al., 2011), and a dropout (Srivastava et al., 2014) is followed by each layer to prevent our model from over fitting. For the textual features, after word embedding layer, we use the same model architecture as TextCNN. Finally a fully connected layer is followed to obtain the final representation v_2. As for the visual features, we use the same architecture as DenseNet (Huang et al., 2017). DenseNet has some dense blocks, each layer in a dense block obtains additional inputs from all preceding layers. In our model, we use two dense blocks to obtain the final representation v_3.

Attention Layer: This layer is the core layer of Attention-based Multimodal Model. At the previous layer,we get the high-level one-dimensional features from each modal input: v_1, v_2 and v_3, these three representations have the same dimension $d^{1\times m}$. we employ soft attention mechanism to associate the important information between the given three high-level features. We compute the normalized attention weights as the similarity with Equation 8.

$$e_i = tanh(v_i \odot \mu^T), i \in [1, 2, 3] \quad (8)$$

where v_1 is one of the $v_{tabular}$, $v_{textual}$ and v_{visual}, μ is the weighted vector that we used to compute the similarity, it will randomly initialized and will be adjusted during the training stage. e_i is the un-normalized attention weights, $odot$ is the dot product between the two given vectors. Beside, in this equation, we use $tanh$ as the activation function. Next, we use softmax to get the normalized attention weights. For each element in v_i, it will multiply by its corresponding normalized attention weight to get the final attention output.

$$\hat{v}_i = \sum_{i=1}^{3} \frac{exp(e_i)}{\sum_{i=1}^{3} exp(e_i)} \cdot v_i, i \in [1, 2, 3] \quad (9)$$

where $\hat{v}_i$ is the attention output of each high-level feature. Finally we concatenate every $\hat{v}_i$ vi as this layer's output and pass on it to the next layer.

Merge and Classification Layer: In this layer, we not only use $\hat{v}_1$, $\hat{v}_2$ and $\hat{v}_3$ to predict the results,but also add the multi-modal encoder feature hmm which obtained from **MMAE** to improve model's performance.

$$h = [\hat{v}_1, \hat{v}_2, \hat{v}_3, h_{mm}] \quad (10)$$

where $h \in R^{1\times(3m+3k)}$. Because this is a multi-class classification problem,so we use $softmax$ to get the final results.

$$prediction = softmax(h) \quad (11)$$

4 Experimental settings and Results

In this section, we first introduce some baseline models and their experimental settings. In order to have a fair comparison and reduce the randomness of results, we use five-fold cross-validation. The batch size is set as 32. The neural networks are trained using the RMSprop optimizer with the learning rate 0.001.

4.1 Baseline models and Previous Work

#1 **Tabular Only:** In this model,the input only has tabular features and will do data preprocessing mentioned above. Tabular Only model is a two layers fully-connected neural network,the number of hidden layer units in each layer is 256 and 128,the activation function is $relu$,and a dropout layer is followed to avoid overfitting,the dropout rate is 0.2.

#2 **Textual Only:** This model is an application of **TextCNN**. In this model, we have the same parameter settings as **TextCNN**, the filter windows is 3,4,5 with 100 feature maps each, and dropout rate is 0.5, but we have a full-connected layer at then end before the classification layer.

#3 **Visual Only:** This is an application of **DenseNet**. In this model, we have two Dense Blocks, each Dense Block has the same parameter settings, and we also have a full-connected layer at the end before the classification layer.

#4 **Tabular (Continuous) + Textual + Visual with Concatenatey:** This is a common architecture for multi-modal dataset, this model has three independent parts which used to learn high-level features from different modal inputs. Continuous means the continuous features in tabular features only do Max-Min Normalization before put into the model. The parameters in these three parts are the same as baseline model **Tabular Only**, **Textual Only** and **Visual Only**. For the representations learned from different parts, this model will simply concatenate them before classification layer.

#5 **Tabular (Discretized) + Textual + Visual with Concatenate:** This model is inspired by. The architecture and the parameters are the same as the model #4, but this model will convert the continuous features to a discrete sequence of tokens to reduce the storage and prevent the model from overfitting.

4.2 Experimental Results

#6 **Tabular(Continuous)+Textual+ Visual with Attention:** The architecture and the parameters in this model are the same as the model #4, but we use soft attention mechanism to interactive the representations learned from different modal inputs instead of simply concatenating.

#7 **Tabular(Continuous)+Textual+Visual+AE Feature with Attention:** In this architecture, we add the autoencoder features into our model. The autoencoder features has three parts from tabular features, textual features and visual features, they are trained from three in dependent autoencoders,all the autoencoders are four layers fully-connected neural network, and the hidden

	Model	Operation	Accuracy ± STD
#1	**Tabular only**	-	36.729±0.0061
#2	**Tabular only**	-	29.403±0.0032
#3	**Visual only**	-	29.252 ±0.0031
#4	**Tabular(Continuous)+Textual+Visual**	Concatenate	37.080±0.0055
#5	**Tabular(Discretized) +Textual + Visual**	Concatenate	37.152±0.002
#6	**Tabular(Continuous) + Textual + Visual**	Attention	**37.381±0.0035**
#7	**Tabular (Continuous)+Textual+ Visual+ AE-Feature**	Attention	37.582±0.0032
#8	**Tabular (Continuous)+Textual+ Visual+ MMAE-Feature**	Attention	**37.883±0.0037**

Table 1: Accuracy between our models and some baseline models on different Multi-modal datasets. AE-Feature means the additional features obtained from three independent autoencoders, MMAE-Feature means the additional features learned from Multi-modal Autoencoder. As for the representations learned from different modals, concatenate means they are combined by simply concatenating, attention means they are combined using attention mechanism. Accuracy higher than the best baseline are in bold. Results are displayed as $mean \pm std$.

Feature	MSE (Normalized)
Visual Feature only	0.03786
+ Tabular Feature	0.03557
+ Textual Feature	**0.03468**

Table 2: The image reconstruction loss using different feature combination. Multi-model Autoencoder has a lower loss.

units size is 512-64-64-512. We concatenate them together with the attention output to predict the final results.

#8 **Tabular(Continuous)+Textual+Visual+MMAE Feature with Attention:** In this architecture, we add the multi-modal autoencoder features into our model. As introduced above, the multi-modal encoder feature is obtained from output of MMAE, which learns the interactive information between different modal features. In order to have a fair comparison with #7, the MMAE Feature has the same dimension with AE-Feature. Besides, MMAE also has three autoencoders, and the parameters in each autoencoders are the same as #7.

5 Conclusion

In this paper, we proposed the a novel framework for multimodal data classification. The framework consists of multi-modal autoencoder module and attention-based multi-modal modeling module. We evaluate the model on the large-scale multimodal datasets. Our framework shows an advantage on accuracy with compared to other approaches. In the future, we will try to extract more features, such as the semantic information of images, thus the similarity or dissimilarity between different modality can be calculated. Moreover, our framework could be adapted to other types of multimodal machine learning task, for instance, the detection task. On the other hand, we will conduct more experiments on larger dataset.

References

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *ICML*.

Sergio Oramas, Francesco Barbieri, Oriol Nieto Caballero, and Xavier Serra. 2018. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*

Valentin Radu, Nicholas D Lane, Sourav Bhattacharya, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2016. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 185–188.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Tao Xu, Han Zhang, Xiaolei Huang, Shaoting Zhang, and Dimitris N Metaxas. 2016. Multimodal deep learning for cervical dysplasia diagnosis. In *International conference on medical image computing and computer-assisted intervention*, pages 115–123. Springer.