Machine Learning Milestone 3

Yu-Hung Yeh [*1] **ZheTing Liu** [*1]

Abstract

This is a paper teaching you how to implement DeViSE(Frome et al., 2013) that is a model from Google. Our dataset is Cifar-10 that has 10 class images(ex: airplane, automobile..etc)from Toronto university. We will separate two parts to discuss: the first focus on implementing DeViSE's similar baseline model, and the other part provides improvement method about baseline model. The DeViSE is comprised of visual model and language mode. We promote visual model accuracy via decreasing the number of convolution layer and increasing the sample numbers, then accuracy dramatically improve from 54% to 80%. Finally, the DeViSEs hit 10 accuracy improve from 42% to 81% in 10 epochs.

1. Introduction

In the Baseline model, The visual model structure is Alexnet using the Cifar-10 as dataset, Alexnet is made up of five convolution layer, two fully-connected layer and final layer is softmax, Alexnet's accuracy is 54%. The other is language model using the wiki's all day English word as dataset and accuracy is 70%. We prefer to change the visual model structure and tune parameters, because the language model training time is absolutely longer than visual model and Alexnet's accuracy is very low. We want to let Alexnet's accuracy up and describe details about two steps to improve accuracy method in the following.

First, We use image pre-processing, flip the same amount of original images and then add to dataset creating 100000 dataset that 50000 is from original and the other 50000 is from mirror images. Result show that visual model accuracy is from 54% to 70%. Finally, We change the Alexnet model to 3 convolution and 2 fully-connected layer including softmax, since Cifar-10 images is 32x32x3 that is a very low pixel image. Alexnet deals with ImageNet that size is 224x224x3, so it is too complex to get high accuracy for Cifar-10.

[*]Equal contribution [1]National Tsing Hua University. Correspondence to: <>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

2. Baseline model

2.1. Dataset

2.1.1. Environment

- Ubuntu 16.04
- python : version 3.5.2
- CPU : AMD Ryzen 5 1600 Six-Core Processor
- Memory : Kingston DDR4-2133 8g
- GPU : GTX 1060 TI 6G

2.1.2. Pre-trained Visual Model(Alexnet)

- Dataset link: CIFAR-10 - Object Recognition in Images
- The dataset included 50000 train data and 10000 test data.
- We need to download the python3 version of the CIFAR-10, using the function that website provided could get the data from batch. The data size is 10000x3072 numpy that means 10000 samples, each row of the array stores a 32x32 color image and first 1024 is red and the next 1024 is green, the final 1024 is black.

2.1.3. Language Model(Gensim)

- Dataset link:enwiki-20171001-pages-articles.xml.bz2
- Before starting to train the model, we have to install gensim library. We use the instruction, ***pip3 install gensim***, to do it.
- We can use the functuin in gensim to import data file. After that, we have to do some preprocess on the data.
- ***prepocessing.py*** this program use to split each sentence and clear sign.

- ***gensim_train.py*** this program use to tain the skip-gram model.
- ***model_test*** thi program use to test the skip-gram model with the dataset from google.

2.2. Training Process

2.2.1. Pre-trained Visual Model(Alexnet)

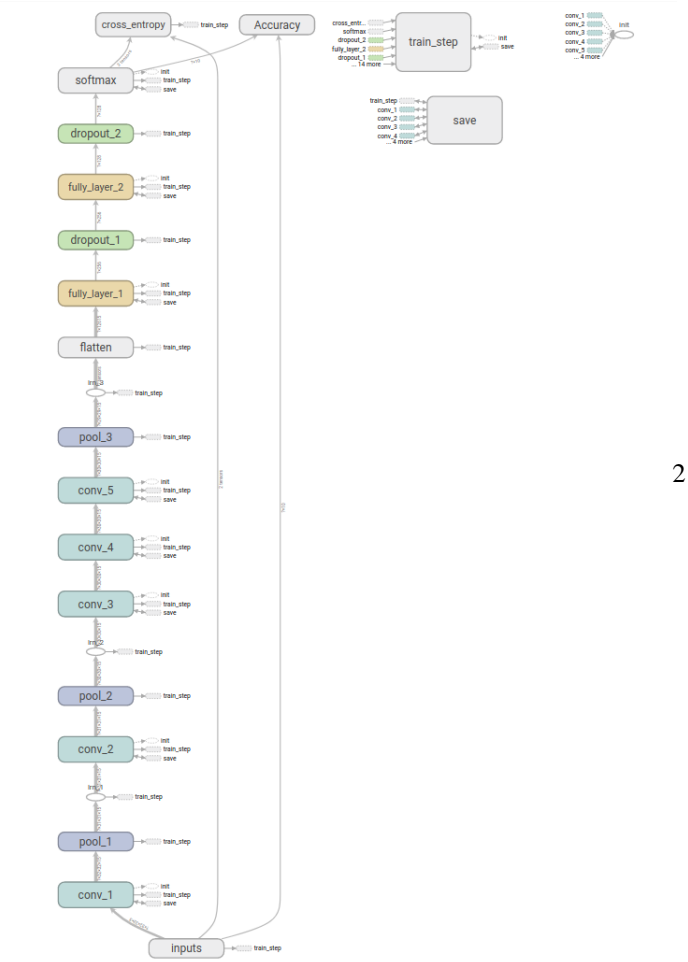


Figure 1. Pre-trained Visual Structure

- We create five Convolutional Neural Networks and three fully-connected layer, one of the three fully-connected layer is softmax layer, in the course of the creating the structure, we add the max pools and dropout, more detail in the Figure 1 Pre-trained Visual Structure.

2.2.2. Language Model(Gensim)

- First, we have to clear something not important like sign. In gensim, we can use this funtion, ***WikiCorpus***, and then use ***get_texts()*** to get each article. It will only return the article that its length over 50 words.
- Second, we set some parameters, includes embedding size, training_epoch, sg. Our embedding size is 250, training_epoch is 5 and sg is 1, it means we use skip-gram not cbow.
- Finally, we use the dataset ***questions-words.txt*** from google to test our word2vec model. It will use 14 different classes to test our model, and we get ***70%*** in average.

```
capital-common-countries: 95.3% (482/506)
capital-world: 92.2% (2227/2415)
currency: 4.7% (4/86)
city-in-state: 63.0% (1428/2265)
family: 91.6% (348/380)
gram1-adjective-to-adverb: 27.8% (181/650)
gram2-opposite: 44.6% (107/240)
gram3-comparative: 71.7% (711/992)
gram4-superlative: 42.5% (215/506)
gram5-present-participle: 54.3% (441/812)
gram6-nationality-adjective: 98.7% (1353/1371)
gram7-past-tense: 46.2% (649/1406)
gram8-plural: 73.3% (727/992)
gram9-plural-verbs: 63.2% (379/600)
total: 70.0% (9252/13221)
```

Figure 2. Accuracy for word2vec model

2.2.3. DeViSE

- Frozen the visual model
  - We want to combine the visual and language model, So we discard the softmax layer from the visual model, according to ***alexnet_frozen.py***.
- Embedding vector
  - We store the Embedding vector of the true label by ***word2vec.py***. It use for new loss function to evaluate the loss.
- Evaluate the loss
  - $loss(image, label) =$

    $$\sum_{j \neq label} max[0, margin - \vec{t}_{label} M\vec{v}(image) + \vec{t}_j M\vec{v}(image)]$$
  - We create three matrix, including margin, $\vec{t}$, and $M\vec{v}(image)$, margin is constant matrix which value is 0.1, $\vec{t}$ is selected form embedding vector that relative to the label, $M\vec{v}(image)$ is from output of the transformation, using the matrix multiplication to get the matrix that we mentioned in the previously, we can get the loss.
- Combine skip-gram model and visual model
  - Our batch size is 256. We train the model for ten epochs and twenty epochs. In each epoch, we take a batch vector into embedding space.

- We calculate the similarity between each vector and its vector of the true label in the embedding space.
- We have four different cases, first is flat hit@1, second is flat hit@2, third is flat hit@5, last is flat hit@10. In this process, we calculate average accuracy in total epochs. We will show the table in Section4.

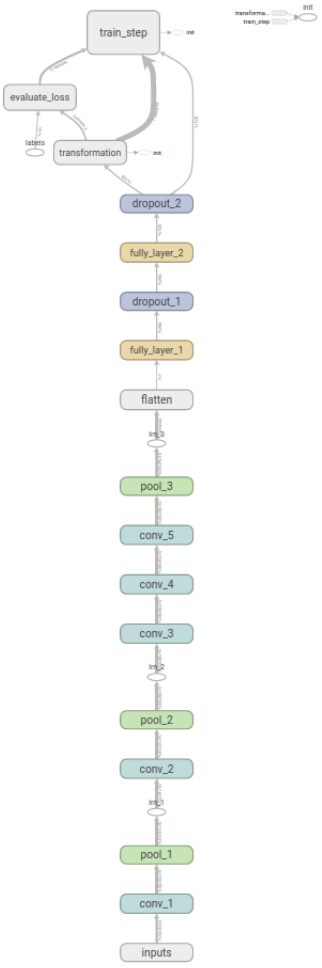


Figure 3. DeViSE Structure

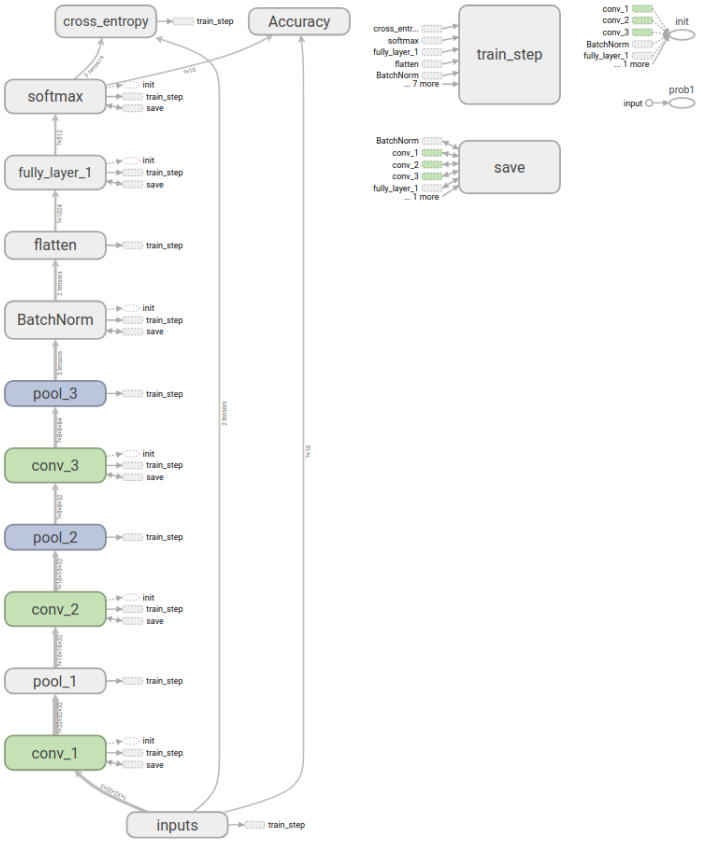


Figure 4. New Visual Structure

3. Improved Model

3.1. New Visual Model

- AlexNet uses for ImageNet. The image in ImageNet is 224x224x3. But our image is from CIFAR10. Its size is 32x32x3. We think the AlexNet model is too complex to train for CIFAR10, so we can't get high accuracy. In order to solve this problem, we decide to simple our model. We decrease the layers of both convolutional layer and full connect layers from five to three and from three to two, respectively. We also set different filter in each convolutional layer. We use 32 filters in first and second layer. The filters in last convolutionl layer is 64. In the full connect layer we change the number of outputs, too. We set our outputs to 512. The reason we choose 512 is that the dimensions in first full connect layer is 256, we have to let large space map to small space. One of the two fully-connected layer is softmax layer, in the course of the creating the structure, we also add two kinds of pooling layers and batch normalization, more detail in the Figure 1 Visual Structure. By adding batch normalization, we can solve the internal covariate shift. Briefly, it can accelerate the speed of training.

3.2. DeViSE

- Improve visual model
  - In addition to change model structure, we use different method on our image. For increase more data, we use some methods on our image to get more features. The following are three different

image processing we use.

  * First, we only catch 24x24 image from our image. That is we make the feature more obvious. But it is not useful for our model.
  * Second, we use openCV to let our image be more bright. But it is also not useful for our model.Above the two methods, they can't improve our model. The accuracy is the same as previous model.
  * Last, we mirror our image. The result is shown as Figure 5. This method improve our model very much. We increase accuracy from 54% to 70%.

– After combining change model structure and mirror image, our accuracy up to 80%.

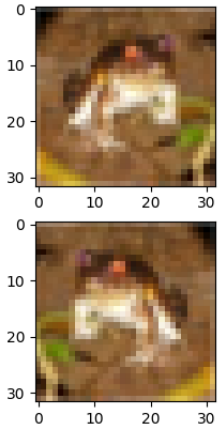


Figure 5. mirror image

- Combine skip-gram model and visual model

  – In training phase, our batch size is 256. We train the model for ten epochs and twenty epochs. Then, in testing phase, we divide the data into 100 batches, 100 data per batch. After that we input the testing data into the model, get the vectors which are predicted by model.

  – we calculate the similarity between each vector and its vector of the true label in the embedding space.

  – We have four different cases, first is flat hit@1, second is flat hit@2, third is flat hit@5, last is flat hit@10. In this process, we calculate average accuracy in total epochs. We will show the table in Section4.

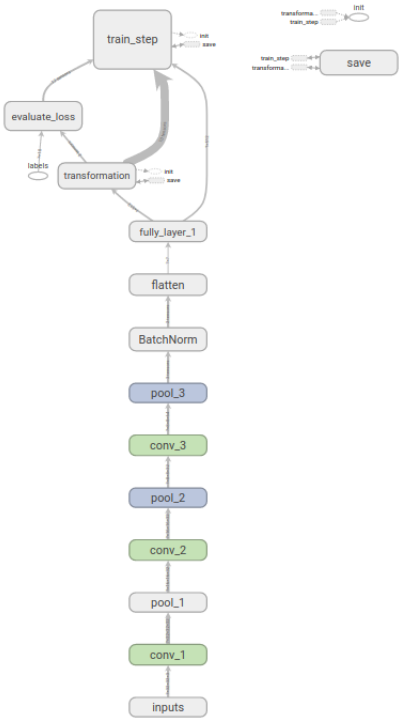


Figure 6. DeViSE Structure

4. Result

- The following grid form is the accuracy of Language Model, Alexnet model and new visual model.

Baseline Model:

	Language model	Pre-trained Visual Model(Alexnet)
Accuracy	70%	54%

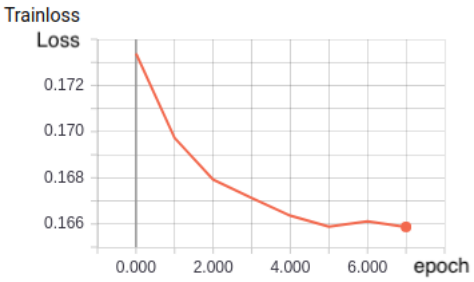
Improved Model:

	Language model	New visual model(New)
Accuracy	70%	80%

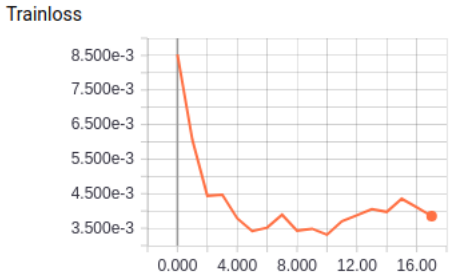
- The following grid form is the Flat hit for old model and new model. The accuracy of the new model is up to two times grater than old one.

Model	Epochs	dim	Flat hit@k(%)			
			1	2	5	10
Alexnet	10	250	22%	24%	34%	42%
	20	250	23%	25%	31%	36%
New	10	250	72%	76%	80%	81%
	20	250	71%	77%	79%	83%

- The following grid form is the loss trending.



(a) old loss



(b) new loss

Figure 7. Loss

5. Conclusion

- We propose two models for language and image. Also, We combine these two model, then use a new loss function to evaluate loss. Furthermore, calculate the similarity between embedding vector and predict vector from model. In baseline model, our visual model's accuracy is 54%. Because of the low accuracy in visual model, we can only get 30% accuracy in DeViSE model. After we improve visual model, from 54% to 80%, DeViSE model up to 75% in average. Compare with MS2 model, we increase accuracy over 45%.
- In future work, we have two methods to do.
  - First, we want to change our data set from CIFAR10 to CIFAR100. After that, we can implement hierarchical and zero shot.
  - Second, the dimensions in our embedding space is 250. We try to adjust dimensions to high value beacuse in the paper, the authors think that the dimensions in embedding space is 500 or 1000. They are better than other values. But we are limited by our hardware. We will improve our hardware to get the higher dimensions.

References

Frome, Andrea, Corrado, Greg S, Shlens, Jon, Bengio, Samy, Dean, Jeff, Mikolov, Tomas, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pp. 2121–2129, 2013.