Yearbook Dataset Deep Learning Report

Yanyao Shen
University of Texas at Austin
shenyanyao@utexas.edu

Yingying Wu
University of Texas at Austin
ywu@math.utexas.edu

Zhuode Liu
University of Texas at Austin
zhuode.liu@gmail.com

Abstract

Labelling a photo's time information is a challenging task due to its difficulty of summarizing useful features. On the other hand, an accurate and fast time labelling method for images is useful and may provide important information in our real life. In this paper, we start exploring this task by looking at the visual historical record of American high school yearbooks. More specifically, we use the grayscale facial images to predict the year each graduation photo was taken. In this paper, we trained our own deep convolution neural networks and do sanity check with pilot projects. Meanwhile, we fine-tuned a VGG-19 network [6] on the Yearbook dataset [3] and tried both classfication and regression loss, and applied simulated annealing algorithm to tune hyperparameters, including dropout rate, learning rate, and ADAM epsilon parameter, to see if it helps find a better set of hyperparameter more efficiently than manual tuning.

1. Introduction

Time series data is everywhere and people are always interested in gaining information from these data: predicting tomorrow's weather, buying/selling on the stock market day after day, analyzing the performance of your favorite sports team, etc.. People gain insights from all sources of time series data in practice, but the understanding is still limited, and generalization to machine-owned "insights" still has a long way to go.

In the computer vision domain, it is also challenging to deal with time series data, even time related data (the time series data has time-dependent labels for the whole set of training data, while time related data may only have time-dependency within each data input, e.g., videos). For example, compared to super-human performance on image classification, video problems are, in general, very hard to achieve human-level performance. One of the reasons is that, people exploit much more prior knowledge when analyzing/classifying, but for machine, training is burdensome, or even infeasible, if it wants to grasp all the prior information.

A recently released dataset [3] tries to provide a way of learning visual time series data, limited to one certain type of images – graduation photos. The dataset greatly decreases the complexity of images (all photos are students in the same year), and focus on the possibility of exploiting time information for labelling in computer vision, which is novel. Although classification is still able to function, it might not be the ideal way to exploit time information. Then, would regression give better performance? This is the question that we would like to find out.

In this paper, we explore several learning methods for this dataset, and try to understand more about the time information gained from different models. More specifically, we trained both classification models and regression models using deep learning architecture, and our best model is the one using regression method. Apart from that, we also implement a small convolution network, and try several methods for hyperparameter selection.

2. Related Work

There have been a lot of work related to facial tasks done by researchers in computer vision domain. Facial detection has achieved robust and accurate results in both static images and videos [7], and with a finer granularity, facial feature detection and facial expression capturing are solved successfully with real-time methods. In recent years, with the resurgence of neural network, machine learning models are able of solving facial recognition tasks with extremely high accuracy. Recognizer of face images has achieved performance better than average human being, and has widely been used in industry [4].

However, the study of time-series data in computer vision is relatively limited, even in the general image domain. The difficulty of this problem is mainly due to the challenge from both time-series data and image recognition and understanding. In this paper, we consider the time-series facial images in a novel inference task. We view this problem from a deep feature learning perspective and our goal is to

learn the weights of our network efficiently and accurately.

Discussing possible ways of finding suitable hyperparameters during training highly effects the final out come of learning. Simulated Annealing has shown to be an effective method of finding hyperparameters in non-deep learning algorithms, e.g., gradient boosted trees. Simulated Annealing is proposed by Kirkpatrick [5] and Cerny [2] as a probabilistic method for finding the global minimum of a cost function that may possess several local minima. Extensions of simulated annealing for continuous functions have been studies by varies scientists, however, due to floating-point precision, in this work, we follow the framework set up by Bertsima [1] with cost function defined on a finite set.

3. Approach

In this section, we first formally formulate the yearbook labelling problem and introduce the metrics for performance measuring. After the basic setting, we first introduce a novel task-specific convolution network designed for this problem in section 3.2. Then, with the help of powerful image feature generating model, we modify the VGG-19 network architecture, and build both classification model and regression model using cross entropy loss and l1 loss respectively. Finally, we introduce the idea of dealing with hyperparameter selection in our network.

3.1. Problem Formulation

Our dataset is separated into two parts: (X_{train}, y_{train}) and (X_{valid}, y_{valid}) (Xs are the image data and ys are the labels). The raw input of each image is a 171×186 grayscale image, and the label is the year of the image.
We will train our model on (X_{train}, y_{train}) and evaluate on (X_{valid}, y_{valid}).

We use l_1 loss to evaluate the performance. We believe l_1 is more robust than classification accuracy, since the historical image data may not be sensitive to a small sliding window of years. For example, predicting an image taken in 2005 as 1905 would be a very bad prediction, yet classification accuracy would not reveal this problem.

3.2. Classification and Regression

Both classification and regression are common methods to use in deep learning domain, and it is hard to tell which one is better. Therefore, we build both of them and compare. Our last layer design is a fully connected layer with weight of size $T \times 109$ for classification and $T \times 1$ for regression. Here, T is the dimension of the output from the previous layer. For VGG-19, $T = 4096$.

Denote $y_1, y_2, \cdots, y_{109}$ as the output for classification, the loss metric is defined according to cross entropy loss as follows:

$$L_c = \log \frac{\sum_i e^{y_i}}{e^{y_{i^*}}}, \tag{1}$$

where i^* is the correct label.

For regression, the output does not represent the probability, but directly represents the final label, hence the loss function is:

$$L_r = |y - i^*|. \tag{2}$$

Based on the above metrics, we can use back-propagation to train our finetuned networks.

3.3. Novel Convolution Network Setup

First we reshape our image from 171×186 pixels to 171×180 pixels. Our convolutions uses a stride of 3 and are zero padded with max pooling over 3×3 blocks. The first convolutional layer consist of convolution with 32 features for each 5×5 patch followed by max pooling with 1 input channels and 32 output channels.

The second layer have 64 features as output for each 5×5 patch, with 32 input channels. This reduce our image from 171×180 to 19×20. Then we add a fully-connected layer with 10240 neurons to allow processing on the entire image, reshape the tensor from the pooling layer into a batch of vectors, multiply by a weight matrix, add a bias, and apply a ReLU. Finally, we apply dropout before the readout layer to reduce overfitting, and add a softmax layer.

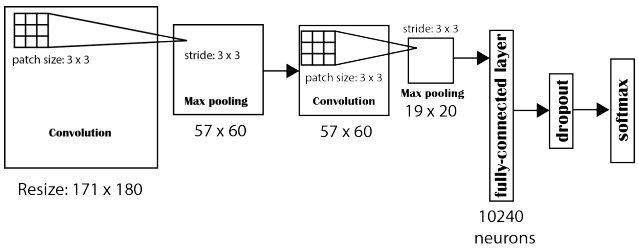


Figure 1: Novel Convolution Network

3.4. VGG Network

We use the VGG-19 network architecture up to the last fully connected layer. The VGG network was introduced by [6] in 2014. Their model achieved first and second places in the localisation and classification tasks, and implies the capability of finding features from general images. Therefore, we utilize their model to create finetuned models for our task. Their deepest model, i.e., VGG-19, includes 16 convolution layers, all consist of 3×3 filters, among which

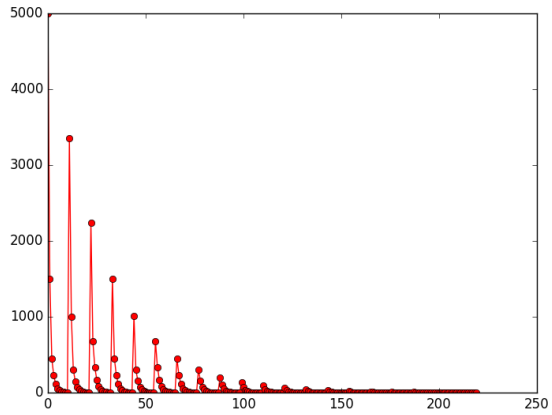


Figure 2: Cooling Scheme

5 max-pooling operation are conducted and "downsample" the image by $2^5 = 32$. An output of size $7 \times 7 \times 512$ is reshaped to a vector and fed sequentially into three fully-connected layers with final output dimension 1000, representing 1000 classes of images.

3.5. Simulated Annealing

We use simulated annealing for hyperparameter tuning. We will explain the method in the following two parts.

3.5.1 Cooling Scheme

We select the temperature initialized as $T_0 = 5000K$, and reduce by half each annealing if the temperature is below 800, otherwise by 0.3. After each scheme, we reheat the temperature back to $\frac{2}{3}T_0$. An illustration is provided below, and in our annealing, we reduce the temperature for five times and then reheat it.

Node	I 0	I 1	A 0	A 1	A 2
1	0.05	0.02	0.01	0.01	0.02
2	0.03	0.03	0.04	0.01	0.03
3	0.04	0.03	0.02	0.02	0.02
4	0.04	0.02	0.00	0.02	0.02
5	0.07	0.02	0.01	0.01	0.02
6	0.02	0.03	0.02	0.03	0.02
7	0.03	0.04	0.02	0.01	0.03
8	0.05	0.02	0.02	0.00	0.00
9	0.02	0.01	0.00	0.04	0.02

*I: Initial value, A: Annealing

Table 1: Annealing Score with Initial Heating

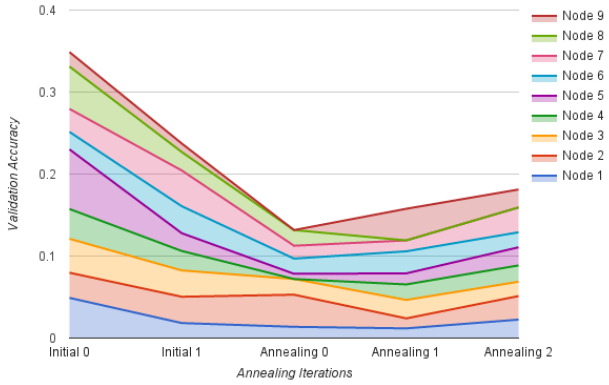


Figure 3: Annealing iteration with multiple nodes.

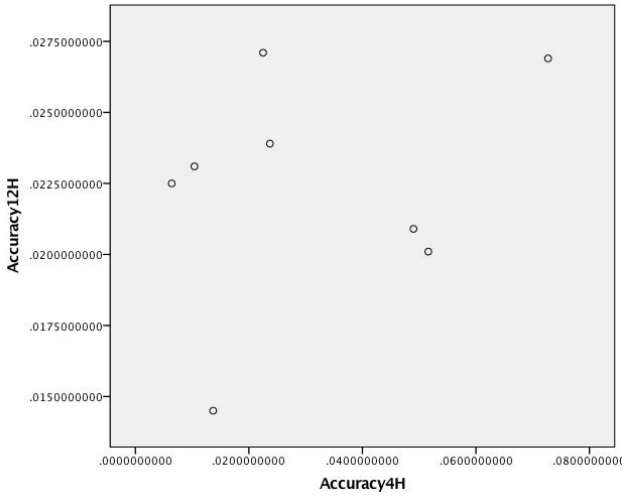


Figure 4: 4 vs. 12 Hour of training Time Results with same hyper-parameters

3.5.2 Simulated Annealing Iterations

We start with two sets of initial values and proceed with 5 annealing scheme followed by 1 heating before continue annealing.

3.6. Generalization of Hyper-parameters

We plot the current data we obtained indicating the relation between 4 hours training time and the extension into 12 hours of training time with the same hyper-parameter. More data is still collecting to be able to conclude results with statistical significance.

4. Experiments

In this section, we show the results based on our ideas described in Section 3. First of all, we summarize the dataset of our problem. Then, we present the result of novel convolution network, by first illustrate its performance on toy examples, and then discuss its training performance on yearbook data. Moreover, we show the result of finetuned VGG network and analyze the pros and cons of simulated annealing based on training results.

4.1. Dataset Description

This dataset contains $14,946$ frontal-facing American high school year-book photos with labels to indicate the years those photos were taken ranging from 1905 to 2013 totalled 104 years with 5 years missing. We use the deep network we developed (described in Section 3.) to predict the year a novel image was taken.

4.2. Novel Model Trained from Scratch

4.2.1 Novel Model Results on Yearbook Data

With batch size 50, we get the following training result on the full yearbook dataset.

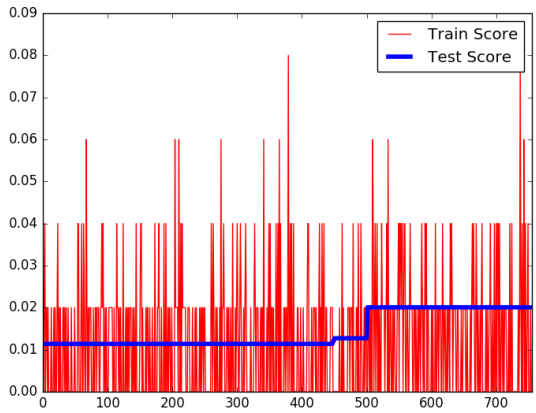
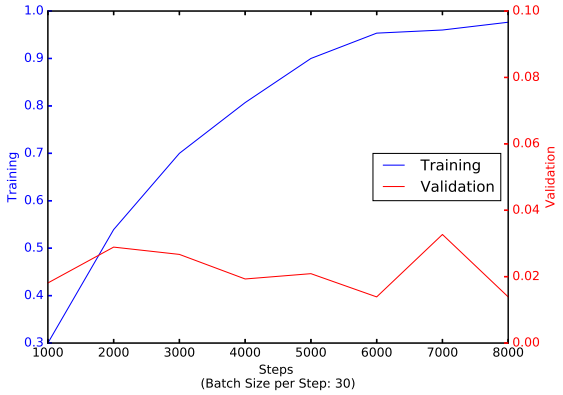


Figure 5: Novel Convolution Network Results

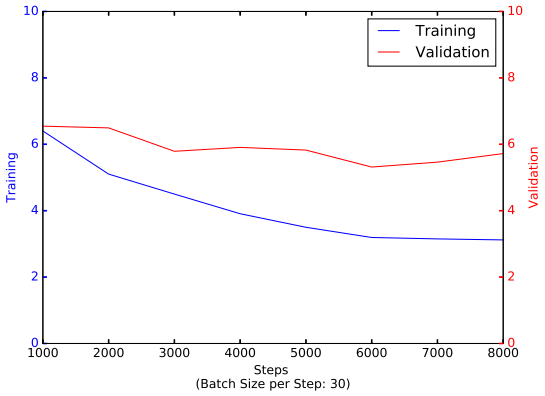
4.3. Finetuned Models Based on VGG Network

4.3.1 Experimental Setup

For networks trained without using simulated annealing for hyper-parameter tuning, our parameter setting is as follows: weight_decay $\in \{1e-1, 1e-2, 1e-3, 1e-4\}$, learn_rate $\in \{1e-3, 1e-4, 1e-5, 1e-6\}$, dropout = 0.5, eps = $1e-8$. When doing regression and classification, we also tried to add a ReLU layer after fc8. However, during finetuning, this method would give worse results than models without final ReLU layers.



(a) accuracy
learn_rate=1e-5, weight_decay=1e-4



(b) ℓ_1 distance

Figure 6: Learning curves using classification loss and regression loss

4.3.2 Experimental Results

Below shows our result on the yearbook validation set in terms of ℓ_1 distance between the predicted year and the true year:

- VGG-19 fine-tuned with classification loss(softmax) achieves l_1 loss of 7.3.
- VGG-19 fine-tuned with regression loss(ℓ_1 distance) achieves l_1 loss of 4.2.

- VGG-19 fine-tuned with classification loss, with hyper-parameter tuned by simulated annealing achieves l_1 loss of 7.4.

The above results show that among all our training models, regression method achieves the best performance. This implies that the time information hidden in the dataset is worth exploiting, and naively treating each year as a different class loses (does not focus on) this information.

5. Discussion

One major problem we found with our network is that it suffers from heavy over-fitting. The training data size is only 22,840, but VGG-19 has 143,667,240 parameters, which makes it capable of perfectly fitting the training set.

Figure 6a shows the learning curve of one of our models trained using classification (softmax) loss. We can see the training data accuracy is already very high at the 4000-th iterations (80%); while the validation accuracy only fluctuate around 3.5% through out 8000 iterations, which is 20x smaller than the training accuracy.

Actually we spent lots of time tuning our parameters, and do find increasing weight_decay helps training a better model, but even with a high weight_decay value (0.1), the training accuracy is still easily be 10x higher than the validation accuracy which is at most 5%.

What about the over fitting problem when using a regression loss (ℓ_1 distance) on this task? Figure 6b shows the learning curve of our best performing regression model which achieves an average ℓ_1 distance of 4.2 on validation data. The graph shows that the performance gap between training and validation set is smaller than in the classification setting (Figure 6a). This may be due to the regression network has fewer parameters — the last layer of the regression network has size 4096×1 whereas the last layer of the classification network has size 4096×109. So the single neuron in the last layer of the regression network gets trained by all the training data, but each neuron in the last layer of the classification network only sees the data belonging to the class which that neuron is responsible for classifying.

Therefore, a possible extension for the classification task would be 'smoothing' the target probability distribution. Concretely, in classification we encode the target label as an one-hot vector, which corresponds to a distribution where the target label has probability 1.0 and all else being 0.0. Then, the optimizer seek to minimize the cross-entropy between this distribution and the distribution output by the classification network. But we could 'soften' the target distribution by assigning neighboring years of the true year a small probability, so that the neurons responsible for those years also get trained. However, this is assuming images from neighboring years also 'look' similar. When this is not true, we might need a way to identify years that looks similar in the data set.

6. Conclusion and Future Work

In this paper, we try to solve the task of visual historical data by delving into the graduation yearbook dataset. We implemented both novel convolution networks and finetuned VGG networks, and achieved our best results using finetuned VGG regression model, which is reasonable. Also, we tried out the simulated annealing method in hyper-parameter selection and analyzed the pros and cons. In our future work, we will try to understand the novel architecture better by not only using pilot examples, but also simplified training data cropped from the yearbook dataset. On the other hand, finding possible methods to decrease training time is also important, methods such as curriculum learning, i.e., the model is first trained based on decade label, then finetune based on year label, may be helpful.

References

[1] D. Bertsimas, J. Tsitsiklis, et al. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[2] V. Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.

[3] S. Ginosar, K. Rakelly, S. Sachs, B. Yin, and A. A. Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–7, 2015.

[4] M. Inc. Face++ research toolkit. www.faceplusplus.com, Dec. 2013.

[5] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.