Cat and dog classification

My project in LaTeX

Dinh The Kiet

2022-07-01

Table of content

1 Introduction

Classify cat and dog using common architecture in deep learning and transfer learning using inception v3 model to compare the different.

2 Dataset

The dataset used is google api mledu dataset which contains cats and dogs images.

3 CNN architecture

3.1 Architecture

3.1.1 Common architecture

The convolution neural network start with a chain of convolution layers and maxpooling layers to extract images's feature.
It's then followed by a flatten layer which is used to convert all multi-dimensions matrix from maxpooling layer to single long linear vector.
The model is finished by 2 dense layers for learning the pattern from feature vector by forward and backward propagation

Figure 1.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 150, 150, 3)]	0
conv2d (Conv2D)	(None, 148, 148, 16)	448
max_pooling2d (MaxPooling2D)	(None, 74, 74, 16)	0
conv2d_1 (Conv2D)	(None, 72, 72, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_2 (Conv2D)	(None, 34, 34, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 64)	0
conv2d_3 (Conv2D)	(None, 15, 15, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 512)	1606144
dense_1 (Dense)	(None, 1)	513

Figure 1: CNN architecture.

The total number of parameters are 1,667,169. While the trainable params are 1,667,169, the non-trainable params are 0.

3.1.2 Transfer learning

Pretrained model Inception Modules are used in Convolutional Neural Networks to allow for more efficient computation and deeper Networks through a dimensionality reduction with stacked 1×1 convolutions. The modules were designed to solve the problem of computational expense, as well as overfitting, among other issues. The solution, in short, is to take multiple

kernel filter sizes within the CNN, and rather than stacking them sequentially, ordering them to operate on the same level.

Custom model Inception v3 model is frozen to the layer mixed7 and then added flatten layer which is followed by dense layer. A drop out layer is continued to drop not very important node then it can avoid overfit. The model is finished by a dense layer with sigmoid acttivation function to classify cat and dog.

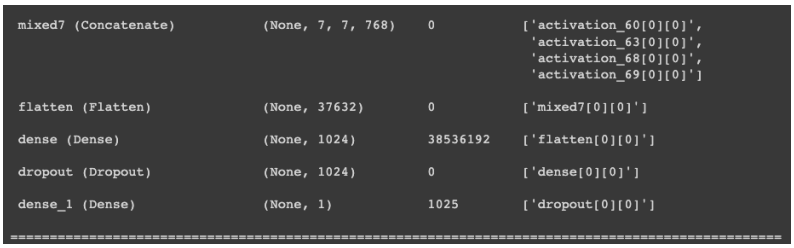


Figure 2: Transfer learning architecture.

The total number of parameters are 47,512,481. While the trainable params are 38,537,217, the non-trainable params are 8,975,264.

3.2 Loss function

Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events.

$$\mathcal{L}(\gamma, y) = -(y * \log(\gamma) + (1 - y) * \log(1 - \gamma)) \tag{1}$$

As we can see the cross-entropy is simply a way to measure the probability of a model. The cross-entropy is useful as it can describe how likely a model is and the error function of each data point. It can also be used to describe a predicted outcome compare to the true outcome.

3.3 Optimization method

3.3.1 Gradient descent

Gradient descent is probably the most popular and widely used out of all optimizers. It is a simple and effective method to find the optimum values for the neural network. The objective of all optimizers is to reach the global minima where the cost function attains the least possible value.

3.3.2 Learning rate

Learning rate is probably the most important aspect of gradient descent and also other optimizers as well. The learning rate choosen is 0.001

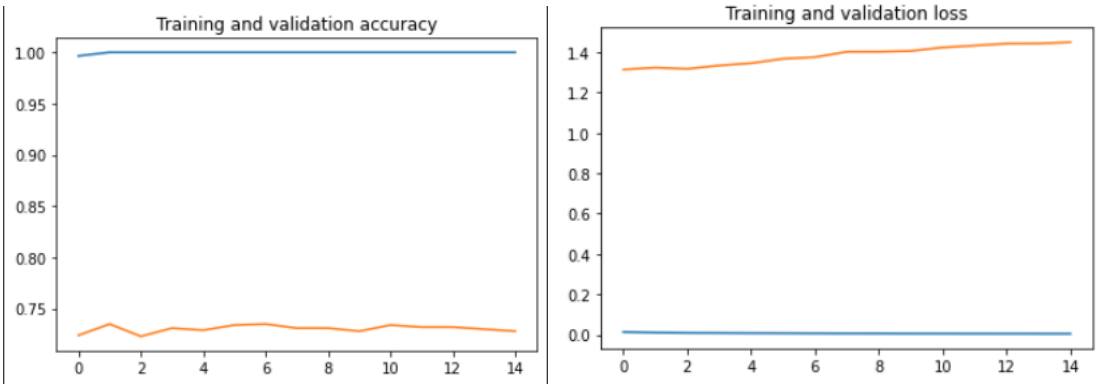
4 Preprocessing

The dataset will be rescaled by 1./255. Validation, train images is flowed in batches of 20 using datagen.generator

5 Evaluation

5.1 Common architecture

The blue lines represent for model's accuracy and loss over epoches in train phases. The yellow line represent for model's accuracy and loss over epoches in test phase.

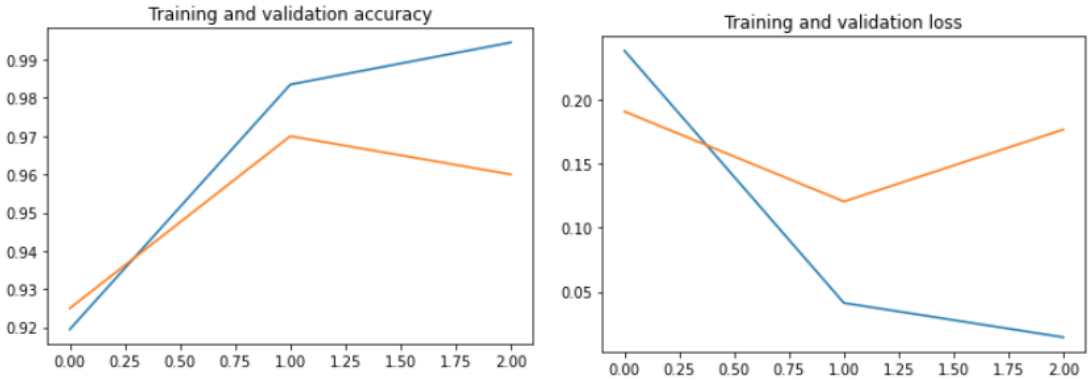


(a) accuracy (b) loss

The model is overfit while it learn perfectly the training set while the accuracy in test set is lower than the the accuracy in train test significantly.

5.2 Pretrained model

The blue lines represent for model's accuracy and loss over epoches in train phases. The yellow line represent for model's accuracy and loss over epoches in test phase.



(a) accuracy (b) loss

The model seems to be good in both train and test set.