УДК 004.89
МРНТИ 28.23.37

HOW CHANGES IN THE DATASET AFFECTS THE ACCURACY OF THE BODY CLASSIFIER

ILCHUBAYEVA D.

Kazakh-British Technical University

Abstract: *Convolutional neural networks have revolutionized computer vision and pattern recognition. They are used to recognize speech, generate various images, process audio signals, process time series, and analyze the meaning of texts. An increasingly complex and deep architecture of neural networks is being developed, along with the undoubted advantages of the common problems of this approach, there are some disadvantages. One of them is the hidden internal principle of the neural network. A properly trained network does not provide researchers with information about identified data dependencies and the structure of the problem. A trained neural network is a set of weight matrices. From this point of view, neural networks are only a tool for solving a specific machine learning problem, but they do not provide experts with analytical information to study the problem. As has long been known, the principle of neural network operation was taken from the principle of neurons in our brain. Inside the brain, we learned to look through ultrasound, PET, MRI and fMRI. And for convolutional neural networks, such indicators as accuracy, precision and heat maps will be used for visualization. The purpose of the work is to find out the effect of a training dataset on the accuracy of a neural network. And how much data will significantly change the stability of the neural network. First of all, they were selected by hyperparameters: learning speed, batch size and number of eras, as well as image size. Then a series of trainings were carried out using the original data, and only then were the images that had been pre-processed added. As the results of the work showed, the dataset, which contains about 15% of the pre-processed data, has a positive effect on the accuracy of the model. When using more data, there was no significant increase in accuracy.*

Key words: *Convolutional neural network, deep learning, classification, dataset, accuracy*

КАК ИЗМЕНЕНИЯ В БАЗЕ ДАННЫХ ВЛИЯЮТ НА ТОЧНОСТЬ КЛАССИФИКАТОРА

Аннотация: *Сверточные нейронные сети произвели революцию в компьютерном зрении и распознавании образов. Их используют для распознавания речи, генерирования различных образов, обработки аудиосигналов, обработки временных рядов, для анализа смысла текстов. Разрабатывается все более сложная и глубокая архитектура нейронных сетей, наряду с несомненными преимуществами общих проблем этого подхода существуют некие недостатки. Одним из них выступает скрытый внутренний принцип нейронной сети. Правильно обученная сеть не предоставляет исследователям информацию о выявленных зависимостях данных и структуре проблемы. Обученная нейронная сеть представляет собой набор весовых матриц. С этой точки зрения нейронные сети являются лишь инструментом для решения конкретной проблемы машинного обучения, но они не предоставляют экспертам аналитическую информацию для изучения проблемы. Как давно известно, принцип работы нейронной сети был взят из принципа работы нейронов в нашем мозгу. Внутрь мозга мы научились заглядывать посредством УЗИ, ПЭТ, МРТ и фМРТ. А для сверточных нейронных сетей будут использоваться такие показатели как точность и для визуализации тепловые карты. Цель работы – выяснить влияние тренировочного датасета на точность нейронной сети, и какое количество данных в значительной сте-*

пени изменит устойчивость нейронной сети. В первую очередь были подобраны гиперпараметрами: скорость обучения, размер партии и количество эпох, а также размер изображений. Затем были проведены ряд тренировок с использованием оригинальных данных и лишь потом добавляли изображения, которые прошли предобработку. Как показали результаты работы датасет, в котором содержится примерно 15% предобработанных данных, положительно сказывается на точности модели. При использовании большего количества данных не было выявлено значительного увеличения точности

Ключевые слова: *сверточная нейронная сеть, глубокое обучение, классификация, данные, точность*

МӘЛІМЕТТЕРДЕГІ ӨЗГЕРІСТЕР КЛАССИФИКАТОРДЫҢ ДӘЛДІГІНЕ ӘСЕР ЕТУІ

Аңдатпа: *Төңкерілген нейрондық желілер компьютерлік көру және үлгіні тану төңкерісін жасады. Олар сөйлеуді тануға, әртүрлі бейнелер жасауға, дыбыстық сигналдарды өңдеуге, уақыт қатарларын өңдеуге және мәтіндердің мағынасын талдауға қолданылады. Нейрондық желілердің барған сайын күрделі және терең архитектурасы дамытылуда, сол сияқты бұл тәсілдің жалпы проблемаларының артықшылықтары мен кемшіліктері де кездеседі. Олардың бірі – нейрондық желінің жасырын ішкі қағидасы. Дұрыс дайындалған желі зерттеушілерге деректерге тәуелділік туралы және мәселенің құрылымы жайында ақпарат бермейді. Оқытылған нейрондық желі – салмақ матрицаларының жиынтығы. Осы тұрғыдан алғанда, нейрондық желілер – машинамен оқытудың белгілі бір мәселесін шешудің құралы, бірақ олар мәселені зерттеу үшін сарапшыларға аналитикалық ақпарат бермейді. Нейрондық желінің жұмыс істеу принципі біздің миымыздағы нейрондар принципінен алынды. Мидың ішіне осы ультрадыбыстық, ПЭТ, МРТ және фМРИ арқылы қарауды үйрендік. Ал конвульсиялық нейрондық желілер үшін визуализация үшін дәлдік және жылу карталары сияқты көрсеткіштер қолданылады. Жұмыстың мақсаты – оқыту мәліметтерінің нейрондық желінің дәлдігіне әсерін анықтау. Ал қанша деректер нейрондық желінің тұрақтылығын айтарлықтай өзгертеді. Ең алдымен, оларда гиперпараметрлер таңдалды: оқу жылдамдығы, пакеттің мөлшері және дәуірлер саны, сонымен қатар кескін өлшемі. Содан кейін бастапқы мәліметтер көмегімен бірқатар жаттығулар өткізілді, сосын оған алдын ала өңделген суреттер қосылды. Жұмыстың нәтижелері көрсеткендей, алдын ала өңделген мәліметтердің шамамен 15%-дан тұратын мәліметтер базасы модельдің нақтылығына оң әсер етеді. Қосымша деректерді пайдаланған кезде дәлдікте айтарлықтай өсу байқалмады.*

Түйінді сөздер: *Конволюциялық жүйке жүйесі, терең оқыту, жіктеу, мәліметтер базасы, дәлдігі*

I. Introduction

At the moment, convolutional neural networks and its modifications are considered the best algorithms for finding objects on the scene in terms of accuracy and speed. Since 2012, neural networks have been at the forefront of ImageNet's renowned international pattern recognition competition. First of all neural networks won this competition and then they captured more and more computer vision tasks and showed that neural networks work much better than traditional approaches.

With the advent of GPU and TPU[1], it became possible to spend less time for training the model. Also, the advent of large image databases has made it possible to train neural networks with many hidden layers. So in this regard, the amount of data growing every year is an advantage.

We understood that as more hidden layers as more difficult problems can be solved by neural network. Signals from these input layers are transmitted from layer to layer using synapses, each of the layers has its own specific coefficients. The internal principle of the neural network is hidden from us. It must be recognized

that we have a problem with the interpretation of the models [2].

For human looking at picture and say what it is a cat is a simple task. Human brains have been trained in this for a thousand years. And each person can describe and guess about the plot of the image through accumulated experience. However, until 2010, for machine, this was an incredibly difficult task. Human vision is not fooled by small changes in the image, and interference such as blur, snow, changing the shape of an object, poor image quality or abstraction, styles is not a problem[9]. It is added to this that often individuals, like the same cat tend to take completely different poses. Also it can be hidden partially by surrounding objects.

We needed to do this in some geometric way, describe the object, describe the relationship of the object, how these parts can relate to each other, then find this image on the object, compare them and get what we recognized poorly. Usually it was a little better than tossing a coin.

II. Classification

Classification of images is a task where you need to decompose images into several categories. For example, a binary classification, when it is necessary to decide into which two classes pictures belong or it is necessary to expanded pictures into many categories, for example, determine cars and airplanes. For example, ImageNet contains 10 million images from the Internet. Pictures are quite complex, varied, and they are manually marked out for belonging to a particular class.

It should be noted that there are errors and controversial cases in the database. For example, this picture is marked as “helicopter”, and if you answer that the picture is car, then it will be wrong. Classification competitions are periodically held on this ImageNet [3],[8] base, and a million pictures from this database participate in the classification task. It is necessary for each picture from this million to answer the question to which of the thousand classes it belongs. And you can give more than one answer, and if the top five is the correct answer, then it is considered that we answered correctly.

Classification algorithms

Many classification algorithms are currently available, but it will be difficult to say which one is superior to the other. It all depends on the application and the type and complexity of the data that is available. For example, if classes are linearly separable, linear classifiers, such as logistic regression, Fisher linear discriminant can be many times better, faster than complex models, but could be an opposite.

The decision tree builds classification or regression models in the form of a tree structure. It uses an if-then rule set that is mutually exclusive and exhaustive for classification. In the end, questions that are forked in a tree may seem strange, but this method has proven to be effective. For example, in banking, when you need to make a decision to give a person a loan.

The tree is built in a descending, recursive “divide and conquer” way. All attributes must be categorical. Otherwise, they must be sampled in advance. Attributes at the top of the tree have a greater impact on classification and are identified using the concept of obtaining information.

An overloaded model can show impressive results on training data, but in practice fail. This can be avoided by pre-pruning, which stops the construction of the tree at an early stage, or by pruning, which removes the branches from a fully grown tree.

Naive Bayesian method is a probabilistic classifier based on the Bayes theorem under the simple assumption that the attributes are conditionally independent.

Classification is carried out by deriving the maximum posterior value.

The naive Bayesian algorithm is very simple to implement, and in most cases good results are obtained. It can be easily scaled for large datasets, since it requires linear time rather than the expensive iterative approximation used for many other types of classifiers.

Naive Bayes may suffer from a problem called the zero-probability problem. When the conditional probability is zero for a particular attribute, it cannot give a reliable forecast.

An example is the first spam filters in your mail. There was a conditional dictionary where

the most common words used in advertising mailings were put together. But he quickly learned to deceive by adding the words of their "good list" to the end of the letter.

k-Nearest Neighbor is a learning algorithm that stores all instances corresponding to training data points in n-dimensional space. When unknown discrete data is received, it analyzes the closest k stored instances (nearest neighbors) and returns the most common class as a forecast, and for data with real values, returns the average value of k nearest neighbors.

In the distance-weighted nearest neighbors algorithm, it weighs the contribution of each of the k neighbors according to their distance using the following query, which gives more weight to the nearest neighbors. The algorithm gets significantly slower as the number of examples, predictors and independent variables increase.

An artificial neural network is a set of connected input/output blocks, where each connection has a weight associated with it, which was started by psychologists and neuroscientists to develop and test computational analogues of neurons.

Many network architectures are currently available, such as direct communication, convolutional, recurrent, etc. The corresponding architecture depends on the application of the model. In most cases, direct link models give fairly accurate results.

However, when there are many hidden layers, training and adjusting weights takes a lot of time. Another disadvantage is the poor interpretability of the model compared to other models.

The algorithm gets significantly slower as the number of examples and/or predictors/ independent variables increase.

Data preparation

To implement the body classification was selected following software:

1. Python programming language, which created many libraries for work with data and neural networks;

2. “Keras” library, which allows you to implement CNN using TensorFlow at a higher software level.

Labeled Faces in the Wild [4] was used as the training dataset. Plus manually annotated data which were prepared with tool named labelImg [5]. A python library imgaug [6] was used for augmentation [10].

List of applied addition sequence which apply in random order:

1.Horizontal flips.

To make some images brighter and some darker which can end up changing the color of the images.

2. gaussian blur (sigma between 0 and 2.0).

3. average/uniform blur (blur image using local means with kernel sizes between 2 and 3).

4. median blur (blur image using local medians with kernel sizes between 3 and 5).

5. Affine transformations:

- translate by -20 to +20 relative to height/ width (per axis)

- rotate by -10 to +10 degrees

- shear by -16 to +16 degrees

- order: use nearest neighbour or bilinear interpolation (fast)

VGGFace2 [7] was used as a test set. Image size (70,70). Different sizes of margins were used to make the dataset more variable.

Initially, training was carried out and the hyper parameters (learning rate, batch size, epoch) were changed. Then, reaching maximum accuracy = 0.988 with the parameters: batch_ size=128, EPOCHS = 50. Training was held where the dataset changed. At first, where there were 15% of the images were augmented then 30% and 50%. Also, the quality and type of transformation has been changed.

Results

The first step is to look at the test set and analyze. We need to understand the features that we would like to highlight for the neural network in the training dataset, whether it was trained or not. This we will do with the help of the heat map.

In Figure 11 we see the processed results that belong to the model that was trained on the “clean” dataset. That is, as the redder the area, as clearer we understand that this is why the neural network considered that the image is

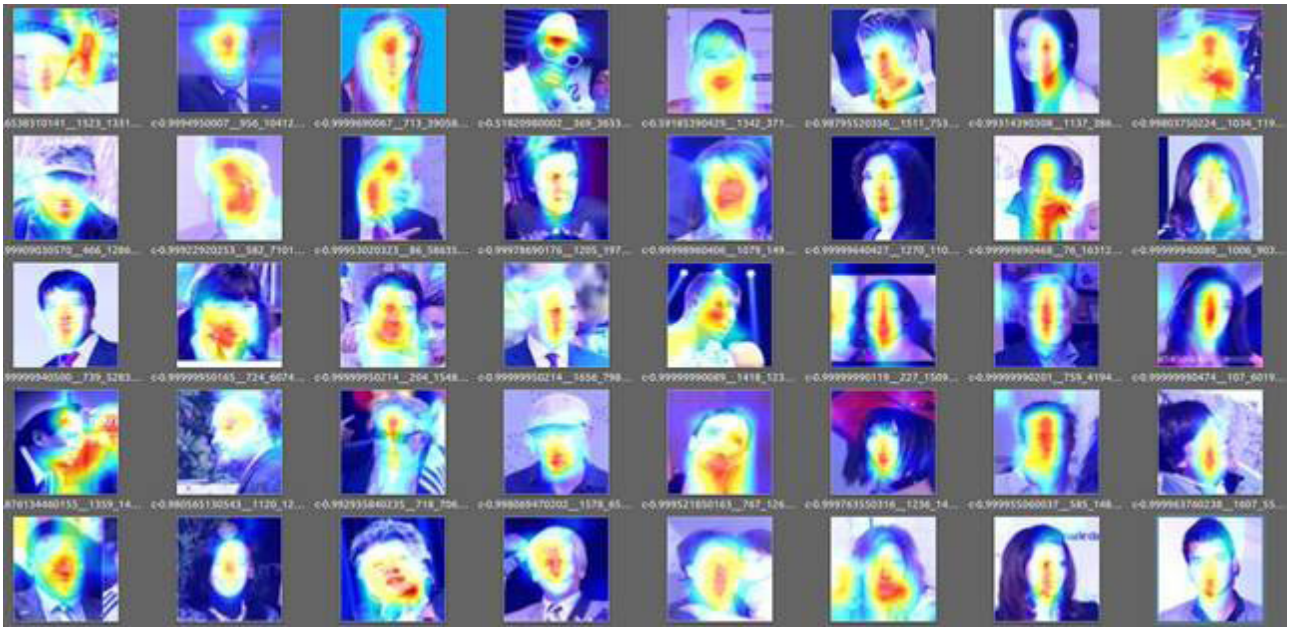

Figure 11 – Trained model results

human. Since the main signs for us are the head and shoulders, the head dominates the faces here, which is good. We will compare the last layers. But for the shoulders to come in as the main feature, we should supplement the dataset.

A large amount of reasoned data may not entirely positively affect the finding of the main features of the object. So the model where 50% of the reasoned data was used has a tendency to expand the red area. If we compare the number of images that are actually false positive. It can be said that their number has decreased, but still finds people where they are not. From this may follow: it is necessary to supplement the negative class with similar images.

III. Conclusion and Future of Threat Intelligence

In general, the hypothesis was confirmed by adding artificially created data to the dataset that has a good effect on the model. This gives her variability and gives a small boost in numbers. But since we live in a world where good and annotated data can be transferred to man's hours and accordingly to money, it's not bad to have tools that will help in training the next model. Now, to the amount of this data, we see that you should not get carried away and make half of your training sample augmented, enough is 10-15%.

For future work it is planned to try the following enhancements:

1. Add data and also take open datasets.
2. Conduct tests with more diverse data augmentation.
3. Experiment with neural network architecture.

Table 1 – Results

	Precision	Recall
Model without augmented data	0.97	0.94
15% augmented data	**0.98**	0.98
30% augmented data	0.94	**0.99**
50% augmented data	0.96	0.98

REFERENCES

1. Tensor processing unit. Available at: https://cloud.google.com/blog/products/gcp/cloud-tpu-machine-learning-accelerators-now-available-in-beta
2. Christoph Molnar. Interpretable Machine Learning : A Guide for Making Black Box Models Explainable, 2018.
3. ImageNet. Available at: http://www.image-net.org/
4. Labeled Faces in the Wild. Availabe at: http://vis-www.cs.umass.edu/lfw/
5. LabelImg. Available at: https://github.com/tzutalin/labelImg
6. Augmentation sequence. Available at: https://imgaug.readthedocs.io/en/latest/source/examples_basics.html
7. VGGFace2. Available at: http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/
8. Alex Krizhevsky. ImageNet Classification with Deep Convolutional Neural Networks
9. Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations?, 2018
10. Dan Hendrycks, Thomas G. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations, 2019.