Reasonable: using spatial features to learn the reasoning in computer vision

1st Maxim Mametkulov
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
maxim.mametkulov@nu.edu.kz

2nd Abay Artykbayev
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
abay.artykbayev@nu.edu.kz

3rd Aidyn Assan
Department of Mathematics
Nazarbayev University
Nur-Sultan, Kazakhstan
aidyn.assan@nu.edu.kz

4th Bibissara Taktarbekova
Department of Mathematics
Nazarbayev University
Nur-Sultan, Kazakhstan
bibissara.taktarbekova@nu.edu.kz

***Abstract*—Modern deep learning is a powerful tool for numerous tasks in science and industry. However, despite whole description of pipeline creating, neural networks are not able to provide the reasoning behind their decisions. As the mathematical conception, neural networks build non-linear hyperplane, that account for its outputs. We suppose that the hyperplane that distinguishes the reasoning is lying in different spatial dimensions than the original input. We provide techniques and approaches that determine the "reasoning" space and the decision hyperplane.**

***Index Terms*—Reasonable AI, neural networks, learning, computer vision, convolutional neural networks**

I. Introduction

Artificial neural networks are tools or hardware models inspired by the neural structure and activity of the human nervous system. As a powerful learning tool, many large-scale information processing systems are gradually introducing neural networks, but there is no set of clearly defined criteria for choosing a neural network. Artificial neural networks are well-known massively parallel computation models that demonstrated excellent behavior in solving complex problems in artificial intelligence. The ability to approximate any function is the main advantage of artificial neural networks. That is, they can approximate to any desired degree of accuracy any real-valued continuous function or one with a countable number of discontinuities between two compact sets. Some of the biggest complaints, though, are that they are black boxes as argued since there is no clear description of their behavior, which means they record the relationship between inputs and outputs with high accuracy, but there is no single answer to the question of how they function. For example, information stored in a neural network is a set of numerical weights and connections that do not give clear clues about how a function is performed, or what is the relationship between inputs and outputs. This limits the use and implementation of ANN, since use methods based on analytic functions that you can understand and test are needed in many science and technology applications. Therefore, many papers say that AI neural networks are black boxes. It is a major drawback, for it is difficult to trust the efficiency of networks addressing real-world problems without the ability to make comprehensible decisions. Therefore, dealing with approximative tools like artificial neural networks needs more research. Thus, in our project, we have tried to train a model that makes black box components visible. In other words, explain and give the reason why machine learning algorithm gives such kind of result. We experimented with the MNIST dataset to open a black box to possibly reveal the basic relationships between input and output of a neural network. In this report, we will try to understand the structure and reason for obtaining such a conclusion from artificial neural networks and how this will help us in the process of using neural networks to solve problems such as the classification task.

II. Methodology

A. Using spatial features reasoning in semisupervised learning

Vanilla neural networks theory proposes that neural network with 1 hidden layer can model any continuous mathematical function.

$$\hat{y} = f(x) \tag{1}$$

The reasoning for humans is drawn from spatial features of the image. As an example of a person, percepting digit 8, one can say that it is 2 circles drawn one above another. Circles are spatial features of the image, and the automaton does not see it unless the image is processed. We suggest a method of semisupervised learning that connects the idea of reasoning and the classification problem, making it a wider classification problem. The aim of this wider classification problem is not to outperform already existing SOTA on test sets, but rather to outperform it on the new data points that were not present in either test or train set. Wider classification will include a label corresponding to an image that does not belong to the

following set(e.g. some noise). The architecture of the net, that will process an image and draw unsupervised reasoning, that will make an inference on whether an image belongs to the set of previously seen or not, is picked to be the combination of feed-forward neural network and vanilla convolutional neural network. The architecture itself is presented on the Fig. 1 The mathematical background of this methodology is based

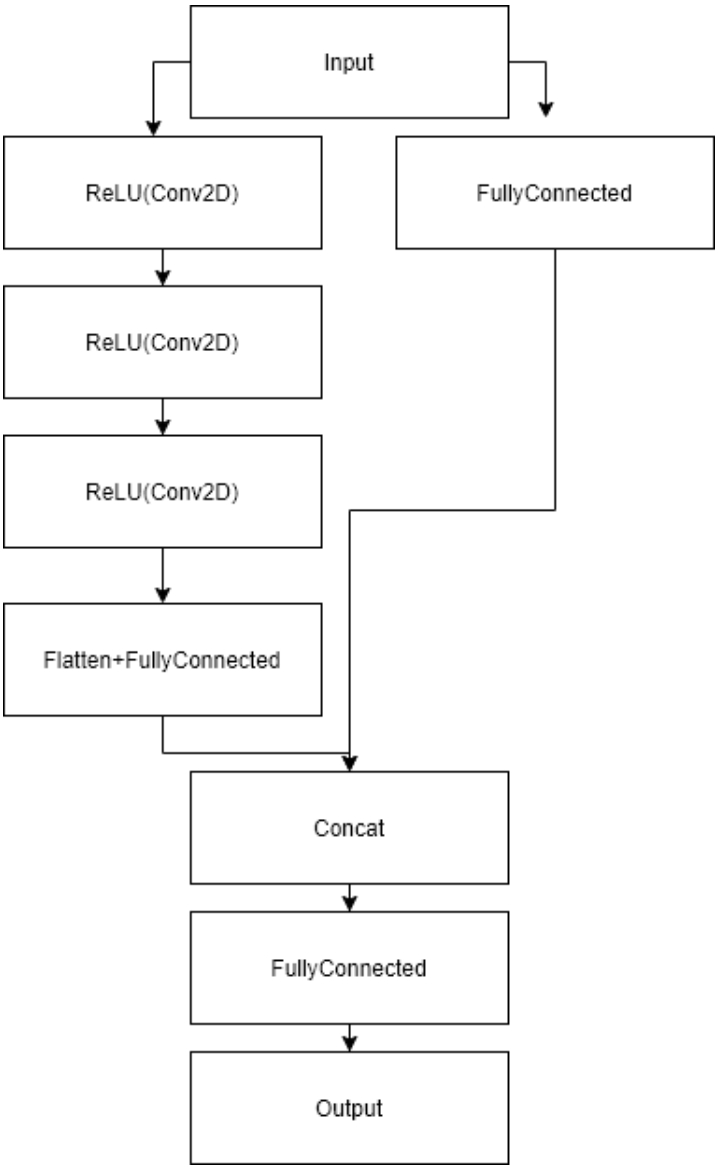

Fig. 1. Architecture of Neural Net in Methodology A

on convolution as an operator to find spatial features. Since all weights are learned with back-propagation, convolutional networks can be seen as synthesizing their own feature extractor[1]. We use convolutional net as a feature extractor and hence, do the mapping

$$f : \mathbb{R}^{28\times 28} \longrightarrow Y \tag{2}$$

Obtained set of features is advised for a semisupervised training that will account for the image being a noise or not. As we solve non-trivial task of obtaining better performance on new data, we introduce error functional, and it will measure the quality of the algorithm given loss to be Cross entropy loss.

$$Q(a(x), y) = CrossEntropy(y, a(x)) \tag{3}$$

We want to minimize the loss, hence

$$Q(a(x), y) \longrightarrow \min \tag{4}$$

The (4) is the core of learning theory, and applying our methodology to functional definition (3), we expand the functional algorithm by having convolutional reasoning to serve as a regularizer. Noise fed into the net will have low score for reasoning, hence, will be classified as a noise and the digit will be classified for the digit itself. By applying the methodology of constructing unsupervised reasoning we add regularization to the error functional presented as a core of learning theory. The following methodology was named ReasonNet.

B. Building a supervised reasoning network

As it was previously mentioned, we supposed that the solution of reasoning should lie in a different space than the input itself, as the processing is required to obtain it. The intuition of convolutional networks as feature extractors is followed in that part of methodology as well. Extracted features are supposed to be **net-reasoning**, flat vector representing the reason of the following label, it was drawn from the last layer of vanilla CNN. Mathematical notion supporting picking this vector is in the fact that the output of a neuron is a linear function to the outputs of the previous layer[2]. The largest value of output unit will correspond to a larger values of input units. Vectors that were selected are used in training the network that will output a reasoning vector. The input of this network is an image and the output is a reasoning vector. The network itself serves as a feature extractor

$$f : \mathbb{R}^{28\times 28} \longrightarrow \mathbb{R}^{256\times 1} \tag{5}$$

Hence, we train 2 networks in parallel, first to classify and second to support a decision. Net-reasoning vector in this methodology does not infer any output, however, becomes a real reason on why the net picked the label. The net-reasoning vector interpretability is a subject to study, but this work is a proof of concept of constructing a supplemential network, that will support a decision. From the brief description of methodology the effect on (3) is undefined and here we introduce a support to be another form of regularizer. Net-reasoning vector uses an intuition of Ridge regularization[3]. The new error functional is going to be in the form of

$$Q(a(x), y) = CE(a(x), y) + \lambda \|Class - True\|_2^2 \tag{6}$$

$Class$ - stands for net-reason for predicted class, $True$ - stands for true net-reason(weight in vanilla net) of predicted class. Hence, we have to keep the weights of the first nets. If some noise or class, that was not trained on are inputted the $L2$ will grow and huge loss will mean that it does not belong to the classes, that were trained on. Net-reasoning architecture is built on vanilla CNN with some hidden layers.

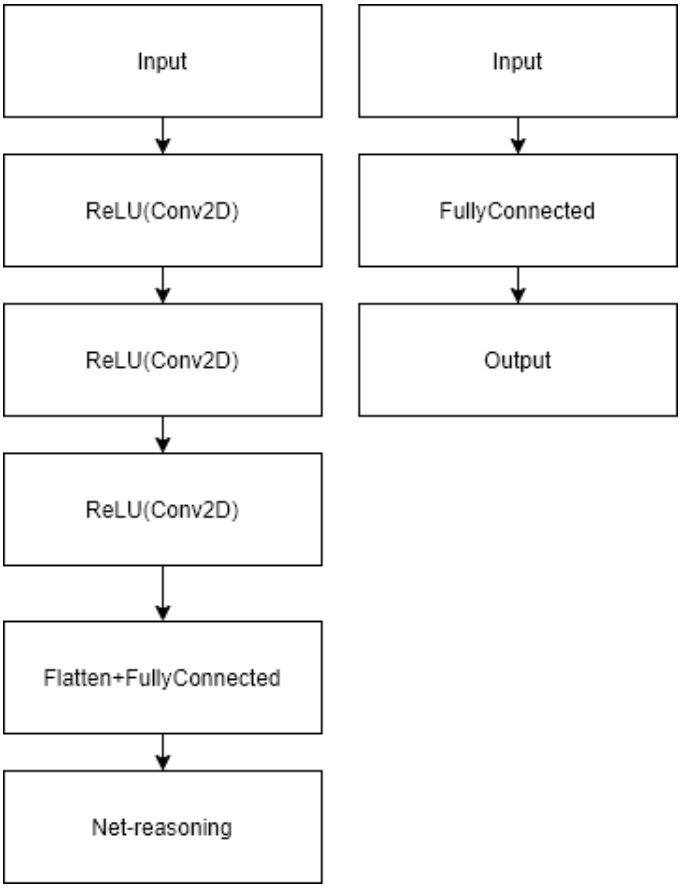

Fig. 2. Architecture of Neural Nets in Methodology B

The architecture is shown on Fig.2. Coming back to (6), the λ here is a significant hyperparameter, that can be tuned and the greater the value of λ the more penalty will be applied as in the case of $\lambda = 0$ the problem simplifies to ordinary classifier. The following method is named Reasonable.

III. Implementation

The models were built and trained using PyTorch[4]. Models were trained on NVIDIA GeForce RTX 2080 Ti. Methodology A was using MNIST dataset as a main dataset and MNIST with noised data[5]. Train data was 85% of the whole dataset and test data was 15% with a single batch, containing 64 items. Models used Adam optimizer with a constant learning rate of 10^{-5}[6]. Methodology A pipeline was trained using CrossEntropy loss as a criterion and Methodology B pipeline used mean-squared error and CrossEntropy. All models were trained for 10 to 30 epochs without learning rate scheduler, so that the learning rate was constant. There were some unit tests on hypothesis and vanilla evaluation of test set. All jupyter notebooks are available on GitHub.

IV. Results

A. Using spatial features reasoning in semisupervised learning

The network from the Methodology A was trained in 10 epochs, and the loss after each epoch is shown in Figure 3. The loss after the last epoch was 0.287, and the loss on the test dataset was about 0.273. This means that this network generalizes well to unseen data, in other words, it is able to tell whether a given image is actually a digit or just a noise. The test set consisted of half of the MNIST and half of the noise data generated by random tensor or shape (28×28). The loss on testset is a result of outperforming vanilla CNN on new datapoints. The results for Vanilla are provided in the next section.
Presented results shows good generalization ability of the proposed method and support mathematical intuition, built in methodology section.

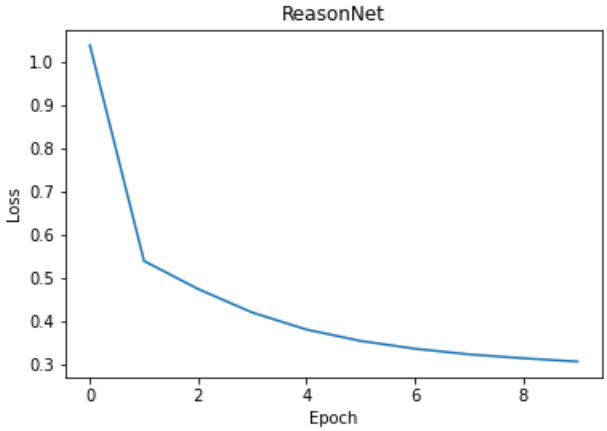

Fig. 3. Loss per epoch during training the Network from Methodology A.

B. Building a supervised reasoning network

For the Methodology B, the Vanilla CNN and the Reasonable network were trained twice, first time with 10 epochs and second time with 30 epochs. Figures 4-7 demonstrate the losses per epoch for each of the cases. As one would expect, the graphs of losses for the 30 epochs are smoother, and the losses are lower. However, even with 10 epochs, the losses are quite low.

The output of the Reasonable network, i.e. vector belonging to $\mathbb{R}^{256\times1}$, is the reason for choosing a digit, and since the losses were low, we can conclude the Reasonable network is able to give a correct reasoning. It is worth to note that this reasoning is supposed to be interpreted by the network, rather than by humans, so for us it may seem meaningless.

Reasonable network showed 0.00027 score after 10 epochs and 0.00023 score after 30 epochs. The losses shows, that even after 30 epochs ,the net is not overfitting. Results on testsets shows the proof-of-concept of the hypothesis of a machine percepting a reason consisting of vectors. Some unit tests(tests

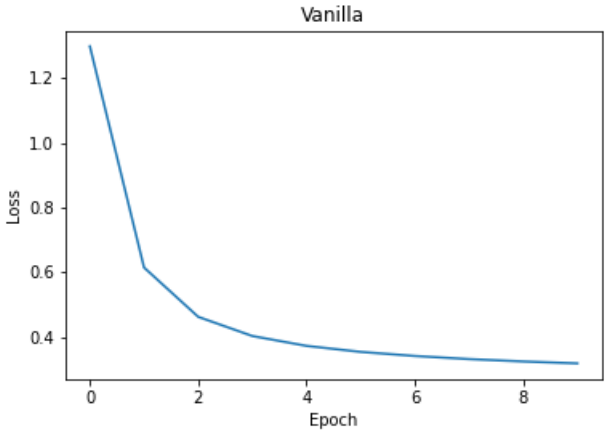

Fig. 4. Loss per epoch during training the Vanilla CNN with 10 epochs.

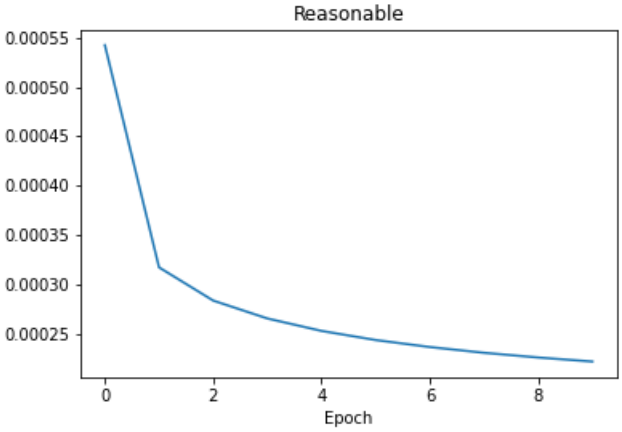

Fig. 5. Loss per epoch during training the Reasonable network with 10 epochs.

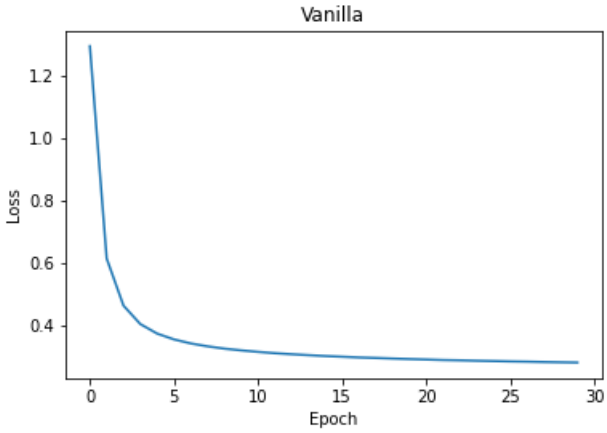

Fig. 6. Loss per epoch during training the Vanilla CNN with 30 epochs.

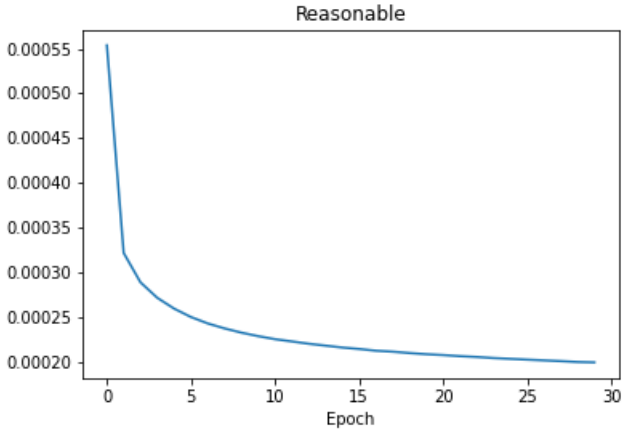

Fig. 7. Loss per epoch during training the Reasonable network with 30 epochs.

for a single digit/noise image) showed, that the selection of λ is highly essential.

V. Conclusion and Future Work

A. Future work

The following work in proof-of-concept has a lot of applications to a real-world deep learning. Method described in this paper can be used in the evaluation of aggressive data augmentations. Sometimes, the augmented data can have no meaning, yet will be fed to neural network, resulting in drop of the loss. Proposed methodology helps in obtaining only meaningful augmentations. This work can also have applications to Graph neural networks and in developing the theory of reasonable AI by applying some of the methods proposed in Bach et al. work[7]. The Graph neural network can extend the opportunities of the vector by presenting more complex mathematical structure. Bach et al. methods can be applied by creating lighter version of the pixel-wise reasoning. Future work includes work in advanced computational tasks for CV, such as Segmentation and detection. The method was also only for CV, but the methodology can be expanded and tested for modern NLP SOTAs.

B. Conclusions

Proposed methodology proved the hypothesis on outperforming vanilla nets on new data points. The ReasonNet from the Methodology A is able to find spatial features and instantly use them as a reasoning making an inference from it to determine whether a given image belongs to the trained classes or not. The Reasonable network from the Methodology B can successfully output a net-reasoning vector, that can be used as a regularizer parameter in solving complex classification tasks. The performance of the models, that are built on methodology is presented in the “Results“ section and the losses prove the concept, proposed in methodology. This work is dedicated to be the starting point in considering some more applications and tests for large-scaled modern SOTA. In this work, some

methods discussed produce outputs, that are supposed to be the reason of a single net choosing a label.

ACKNOWLEDGMENTS

We would like to thank our professors, Dr. Siamac Fazli and Dr. Anh Nguyen Tu, for teaching us and motivating us to research in Deep Learning. We would also like to thank our families and friend who supported us during these times.

REFERENCES

[1] Yann LeCun et al., "Gradient-based Learning Applied to Document Recognition", proc. of the IEEE, November 1998

[2] Rumelhart, D., Hinton, G. & Williams, R. "Learning representations by back-propagating errors." Nature 323, 533–536 (1986). https://doi.org/10.1038/323533a0

[3] Arthur E. Hoerl & Robert W. Kennard (2000) "Ridge Regression: Biased Estimation for Nonorthogonal Problems", Technometrics, 42:1, 80-86, DOI: 10.1080/00401706.2000.10485983

[4] Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library", Advances in Neural Information Processing Systems 32, 8024-8035, 2019

[5] LeCun et al. "MNIST handwritten digit database", ATT Labs [Online], 2, 2010

[6] Diederik P. Kingma and Jimmy Ba "Adam: A Method for Stochastic Optimization", arXiv, 1412.6980, 2014

[7] Bach et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation", https://doi.org/10.1371/journal.pone.0130140, 2015