CNN with residual learning extensions in neutrino high energy physics

Miroslav Kubu and Petr Bour
Department of Mathematics, FNSPE, Czech Technical University in Prague, Trojanova 13, 12000 Praha 2, Czech Republic
E-mail: kubumiro@fjfi.cvut.cz (corresponding author)

Abstract. As many reconstruction steps in neutrino high energy physics (HEP) are similar to image pattern recognition tasks, we explore the potential of Convolutional Neural Networks (CNN) combined with residual machine learning algorithm. Characteristic features from neutrino track image pixelmaps are extracted at different scales and these features are used for classification of the type of neutrino interaction. In this contribution, we summarize observed performance of the residual neural networks (ResNet) for neutrino charged current (CC) interaction detections using image-like Monte Carlo simulated data for muon and electron neutrinos. The two topologies depicted at the neutrino detectors differ, muon neutrino CC interaction is dominated by a slowly ionizing muon, while electron neutrino CC interaction is usually recorded as a wide shower. For the ResNet performance evaluation, we use area under ROC curve (AUC) as the evaluation metric. We observe an improvement while using residual learning compared to general CNN architecture, which is caused by a more stable training with lesser vulnerability to the vanishing gradient of the ResNets. Moreover, stacking other hidden layers within our ResNet model greatly increased the AUC value on the test neutrino dataset without the signs of unstable training or overfitting.

1. Introduction
CNNs have been frequently used in computer vision community to solve complex image recognition tasks. For example, CNNs are commonly used for reconstruction purposes within the neutrino experiments [1]. The ability to extract important features from raw data images can be used for charged current (CC) interaction classification and particle identification [2]. Deeper neural networks are necessary for better performance. However, deep models are more difficult to train. In [3], a residual neural network framework is presented to ease the training of networks that are notably deeper than those used previously. Empirical studies show that these residual neural networks (ResNets) are easier to optimize, and can gain accuracy from considerably increased depth [3]. In this paper, we summarize observed performance of the residual neural networks for neutrino CC interaction classification. We use provided image-like Monte Carlo simulated data for muon neutrino $\nu_\mu$ CC and electron neutrino $\nu_e$ CC interactions. The topologies of $\nu_\mu$ and $\nu_e$ CC interactions depicted at the neutrino detectors differ. While the topology of $\nu_\mu$ CC interaction is dominated by a slowly ionizing muon, $\nu_e$ CC interaction is usually recorded as a wide shower. Thus, this classification task can be effectively solved by image recognition techniques and serves well for the comparison of different classification models. Alternatively, the standard classification statistical methods can be employed [4, 5].
needed, such HEP data can be pre-processed by various statistical methods before a classification procedure is carried out (e.g. [6]). To obtain better understanding of data structure and their distributions, statistical estimates or tests by means of appropriately selected metrics should be performed [7, 8, 9].

2. Dataset and CNN classification principle

We compare our CNN models on a Monte Carlo simulated dataset of neutrino CC interactions [10, 11]. A balanced dataset was used for the binary classification task. Because of computational costs, a training set of only 30k observations in total was used. The models were trained on a set of 24k training samples. The remaining subset was used for hyper-parameter tuning and evaluation of the classifiers. The intensity of each pixel in the pixelmaps is proportional to the energy deposition in each scintillator column of the detector. Each pixel is represented by a value from 0 to 255. Thus, the projections could be interpreted as grayscale images.

In addition to the standard CNNs, we solve the binary classification task using the CNNs with residual learning extension. The basic idea of residual learning is to explicitly let every few stacked layers fit a residual mapping instead of a desired underlying mapping. Let $H(\vec{x}) : \mathbb{R}^d \to \mathbb{R}^d$ be the desired underlying mapping of an input $\vec{x} \in \mathbb{R}^d$, $d \in \mathbb{N}$. Then we let the stacked layers fit residual mapping $F(\vec{x}) = H(\vec{x}) - \vec{x}$. It is hypothesized that it is easier to optimize the residual mapping $F$ than to optimize the original desired mapping $H$ as hidden layers do not perform superbly for identity mapping [3]. The formulation can be realized by so-called feed-forward shortcut connections. As indicated in Figure 1, the shortcut connections are the connections skipping one or more stacked layers. The shortcut connections perform identity mapping $x$, which is later element-wise added to the outputs of the stacked layers $F(x)$ forming so-called residual units. This way, we obtain the desired mapping $H(\vec{x}) = F(\vec{x}) + \vec{x}$.

Concerning the parameter learning, the mini-batch gradient descent optimisation process was used for the training. It allows more robust convergence and avoiding local minima [12]. Formally written, let $n \in \mathbb{N}$ be the size of the training set $\vec{x}_1, \ldots, \vec{x}_n$ with corresponding labels $y_1, \ldots, y_n \subset \{0, 1\}$. Then, we minimize the empirical risk function $R$ defined as

$$ R(\vec{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L\left( f(\vec{x}_i; \vec{\theta}); y_i \right), $$

Figure 1. Residual unit with 2 stacked layers and activation functions $\sigma$ and $f$. 

(1)
for free parameters $\vec{\theta} \subset \mathbb{R}$ and output $f(\vec{x}_i; \vec{\theta})$. Function $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_{+}^{\mathbb{R}}$ is a chosen loss function. The binary cross entropy function is used as the loss function for binary classification. Let $\mathcal{B} = (\vec{x}_{i_1}, \ldots, \vec{x}_{i_m})$ be a training mini-batch and $i_1, \ldots, i_m \in \{1, \ldots, n\}$, $m \in \mathbb{N}$ for $m < n$ and for $k \neq l$ let $i_k \neq i_l$ for all $k, l \in \{1, \ldots, m\}$. Then, we minimize empirical risk function using mini-batch learning rule

$$\vec{\theta}^{(k+1)} = \vec{\theta}^{(k)} - \varepsilon \frac{1}{|\mathcal{B}|} \nabla_{\vec{\theta}} \sum_{\vec{x}_i \in \mathcal{B}} L \left( f(\vec{x}_i; \vec{\theta}^{(k)}), y_i \right),$$

(2)

where $|\mathcal{B}| = m$ is the cardinality of the mini-batch $\mathcal{B}$ and $\varepsilon > 0$ is the learning rate.

3. CNN architecture testing

We aim to compare the performance of the classical CNNs to the performance of the residual neural network classifiers. Thus, we firstly build a standard CNN model. A twin tower structure is used for two points of view. In both towers, the input pixelmaps are transformed by two convolutional layers with $32 \times 3$ convolutional kernels before merging. After merging the feature maps from both towers, two convolutional layers with $64 \times 3$ kernels and a fully connected layer with 512 neurons are followed. Then, we aim to extend the standard CNN model by using residual learning techniques. Our ResNet model is based on the same architecture as the first CNN model. However, the shortcut connections for every two stacked hidden layers are used. The model architectures are illustrated in Figure 2. Furthermore, we build a deeper ResNet model to explore whether adding additional layers can improve performance. For the deeper ResNet model, we use additional layers with $32 \times 5$ convolutional filters applied directly to the inputs. Then, we double the residual units in the inner part of the network. For all the architectures tested, ReLU activation function is used for all the hidden layers, while the sigmoid function $f(x) = (1 + e^{-x})^{-1}$ serves as the output activation function. We used mini-batch stochastic gradient descent with mini-batch size of 32. The Xavier method \[13\] was used for the parameter initialization.

4. Classification results and conclusions

For the comparison of the classifiers, we use a test subset containing 10% samples of the original dataset. We use the area under ROC curve (AUC) as the evaluation metric. As Figure 3 shows,
all the three models performed remarkably. Furthermore, we observe an improvement while using residual learning. That may be caused by a more stable training with lesser vulnerability to the vanishing gradient of the ResNets. A complete overview of the AUC values is presented in Table 1. In Figure 4, we show chosen metrics such as signal efficiency $\varepsilon_S$, background efficiency $\rho_B$, precision and F1 score for different discriminant values. In this case, $\nu_e$ CC interaction class corresponds to the signal class. The optimal discriminant value $\delta^*$ was chosen to maximize F1 score. Thus we were able to rapidly increase the performance of the model by using techniques of residual learning. It is obvious that stacking other hidden layers within our ResNet model greatly increased the AUC value on the test dataset without the signs of unstable training or overfitting.

**Acknowledgments**

These results were supported by the research grants LTT180001, LM2018068, CAAS EF16_019/0000778, and SGS18/188/OHK4/3T/14.

**References**

[1] NOvA Collaboration 2016 A Convolutional Neural Network Neutrino Event Classifier *Journal of Instrumentation* 11 (P090001)

[2] Psihas F, Niner E, Groh M, et al 2019 Context-Enriched Identification of Particles with a Convolutional Network for Neutrino Events, ArXiv:1906.00713v1 [physics.ins-det].

[3] He K, Zhang X, Ren S, Sun J 2016 Deep Residual Learning for Image Recognition *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

[4] Stěpánek M, Franc J, Kůš V 2014 Model based clustering method as a new multivariate technique in high energy physics *Journal of Physics: Conference Series* 490 (012225)

[5] Bour P, Kůš V, Franc J 2016 Statistical classification techniques in high energy physics (SDDT algorithm) *J. Phys.: Conf. Series* 738 (012034)
Figure 4. Overview of the chosen metrics and discriminant $\delta$ values for the CNN (top), ResNet (center) and deep ResNet (bottom) classifiers.

[6] Kus V, Finger R 2019 Unifying approach to score based statistical inference in physical sciences *Journal of Physics: Conference Series* 1391 (012124)

[7] Hrabáková J, Kůs V 2013 The Consistency and Robustness of Modified Cramer-Von Mises and Kolmogorov-Cramer Estimators *Communication in Statistics-Theory and Method* 42 (20) 3665-3677

[8] Hrabáková J, Kůs V 2017 Notes on consistency of some minimum distance estimators with simulation results *Metrika* 80 (2) 243-257

[9] Kůs V, Morales D and Vajda I 2008 Extensions of the parametric families of divergences used in statistical inference *Kybernetika* 44, 95-112

[10] Acero MA, et al. 2018 New constraints on oscillation parameters from $v(e)$ appearance and $v(\mu)$ disappearance in the NOvA experiment *Physical Review D* 98(3)

[11] Acero MA, et al. 2019 First measurement of neutrino oscillation parameters using neutrinos and antineutrinos by NOvA *Physical Review Letters* 123(15)

[12] Goyal P, et al. 2017 Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour *CoRR Journal* 1706 (02677)

[13] Glorot X, Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks *Proceedings of the International Conference on Artificial Intelligence and Statistics*