1 Literature Review

1.1 Designing Neural Networks

Parameter Initialization Strategies When we start optimizing parameters with gradient descent, a method we adopt to initialize parameters significantly affects convergence rate to global optimum, model performance measured by standard metric, and vanishing or exploding gradient problem. Therefore, how to initialize parameters in a neural network has been one of the most controversial issues in the history of neural network research area regardless of its dataset modality[1], model architecture, and task specification.

Rumelhari et al. (2004) embraced random initialization when they introduced backward propagation, explaining that any constant numbers, including zero initialization which initializes all parameters to zero, does optimize the same amount of information for each iteration, there is no meaning in using them. Glorot and Bengio (2010) later suggested scaling up initialize parameter concerning dimensionality of layers considering their changes in variance after backward propagation[2]. He et al. (2015a) was published by the 2015 winner of ImageNet Large Scale Visual Recognition Challenge(ILSVRC) competition, one of the most famous computer vision competitions that occasionally have brought some of the most important innovations in the neural network, and one of their groundbreaking solutions, which is generally called *Xavier initialization*, is most commonly used initialization method up to now.

However, this does not mean that the Xavier initialization can accomplish the best result regardless of the task, modality, model architecture, and other indefinite conditions, hence there have been many alternatives such as layer-wise sequential unit variance (Mishkin and Matas, 2016), fixup initialization (Zhang et al., 2019), and zerO initialization (Zhao et al., 2021).

The Type of Hidden Unit Generally, each hidden layer of neural networks is composed of function composition that a non-linear function is one the top of an affine transformation. The non-linear function enables the model to solve complex non-linearly separable problems and has a crucial role for the model to sustain consistent parameters while remaining adequate gradients.

The most typical non-linear functions, which are the logistic sigmoid function and the hyperbolic tangent function were used until the rectified linear unit(ReLU) was devised to compensate limitations of a sigmoidal function (Glorot et al. 2011; Nair and Hinton 2010; Jarrett et al. 2009). That is, the sigmoidal function sensitively reacts only for a value around zero, which prevents the model from stepping forward. As of 2022, the default recommendation is to use variants of ReLU, such as a parametric ReLU(PReLU) He et al. (2015b) or a leaky ReLU (Xu et al., 2015).

Nonetheless, although the critical limitation of sigmoidal activation function, other settings such as recurrent networks and some autoencoders occasionally befit to the sigmoidal function. That is to say, the design of the hidden unit is an extremely active field, and thereby deciding an adequate hidden unit consists

[1] In this paper, what we refer to modality is limited to computer vision(image), natural language processing(text), and speech language processing(speech).

[2] See Eq.16 of the paper for the detailed explanation.

of trial and error of various types of hidden units and evaluating the actual performance of the model.

choosing a learning rate As the modern deep neural networks model adopts numerical analysis to solve the target problem, an iterative optimization process is carried out using a derivative of the parameter for object function. In process of optimization, researchers can decide how much amount of derivative should be used to update parameters. This is called the *learning rate* or the size of the step. A popular approach is to evaluate $f(x - \epsilon\Delta_x f(x))$ for several values and select one that gives a minimal cost, where f and ϵ are an objective function and the learning rate respectively. Although a small constant value is recommended to the first choice, an optimized function can be a poor local minimum, where the point is lower than its local neighbors but far different from a global minimum. Thus one should experiment with a wide range of scalar values ranging from 0 to 1 and decide the one that performs best on an (unseen) dataset.

1.2 Reciprocal fields of Neural Networks and Visual Analytics

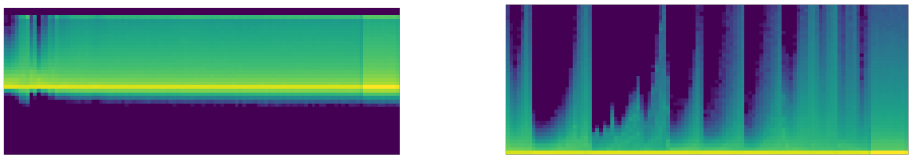
The History of Visual Analytics The term visual analytics has been termed by Wong and Thomas (2004) as a field of research. Keim et al. (2010) defined visual analytics later as the science that combines automated systems with interactive visualizations for an effective understanding, reasoning, and decision making based on large and complex datasets. The goals of visual analytics, in summary, is represented as to enable users to obtain insight that directly supports an assessment, planning, and decision making by amplifying human cognition with abstract information using an interactive visual interface (Card et al. 1999; Thomas and Cook 2006; Keim et al. 2006; Keim et al. 2008)

Visual Analytics and Neural Networks Given that the results and procedures of the solution from modern neural networks models are rarely understood by a human, and therein lies the concept of the "black box", incorporation of visual analytics into neural networks deems to be apparent. According to Keim et al. (2010), it is only possible through visual analytics that these days large-scale and complex problems become solvable since neither automated analysis nor visualization alone can provide.

The first systematic study performed by Zeiler and Fergus (2014) found that visualization of parameters reconstructed by Deconvolutional Network Zeiler et al. (2011) illustrates how exactly each receptive filter shares the pattern as an intermediate level of representation of images. This brilliant way of approach not only verifies the very intuition behind the Convolutional neural network (LeCun et al., 1999) matches the actual behavior of the model, also sheds a light on cultivating visualization to demystify features inside models. More recent evidence (Li et al., 2018) reveals that on the smoother landscape of visualized loss function rest the reasons for the different model performance.

1.3 What The Paper is About

limitations of the previous study So far, we have discussed difficult and time-consuming issues of designing neural networks and an emerging and promis-



(a) $y = max(0, x)$ (b) $y = max(0.1x, x)$

Figure 1: The histogram of 4 3x3d ConvNet on MNIST measured after 1 epoch with 0.5 constant learning rate and vanilla stochastic gradient descent (i.e., without momentum, dampening, and weight decay). The hidden unit with leaky ReLU shows more well-balanced distribution.

ing new kind of science combined with visual analytics and neural networks. However, to our best knowledge, few researchers have addressed the issue of incorporating visual analytics into neural networks *in the process of* designing the system, and hence as only for clarifying the relationship between input and output of the system, are those that are currently likely to be investigated.

visualizations towards better training Let us briefly explain two examples illustrating the possible way of beneficiary cooperation of two fields. Figure 1 shows distributions of a randomly selected activation layer where the model is composed of a stack $N = 4$ two-dimensional convolutional layers with a max-pooling and a final linear layer on the top of the convolutional layers. All other model specifications are set to be equal except the type of hidden unit. You might be able to observe that (a) the left one keeps a more stable tendency than (b) the right one. In other words, the left one is training smoothly while the right one is incessantly struggling from both underflow and overflow problems.[3] Note that these histograms can be obtained at any time in the process of training models. That is to say, we can estimate the appropriate option without ever training each model through to the end.

Another example can be seen in Figure 2, which depicts a loss of the model. The loss function (or objective function) is that we are supposed to minimize, where each loss is calculated through the batches respectively without optimization process. In other words, after calculating the loss, we directly skipped to the next batch iteration. Here for our experiment, two blocks of the QRNN Encoder-Decoder model are used to translate French to German of WMT'15 [4]. Again we can easily select the most optimal learning rate around 0.001 where the steepest gradient before the lowest loss resides without experimenting with all possible values.

In conclusion, this study has the potential to provide an alternative way of incorporating visual analytics into neural networks hence helping researchers to extrapolate the model performance based on visualized model behavior. Though we could not assure the best result in this way which might have been obtained from numerous trials and errors, we believe that this method guides you to explore an enormous abyss of neural network experiments.

[3] An explanation of what it means will be provided in the extended version of the thesis.
[4] The dataset can be downloaded here: `https://nlp.stanford.edu/projects/nmt/`

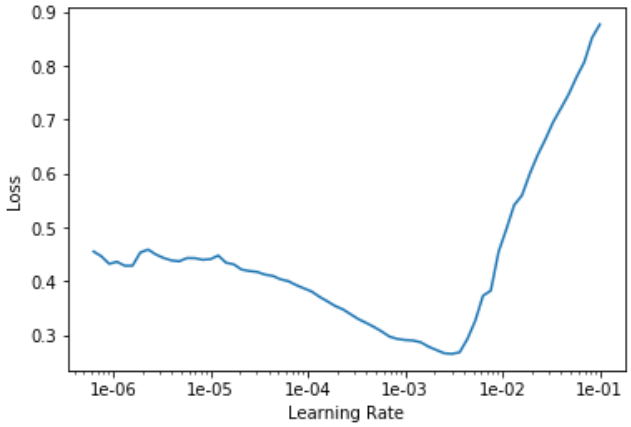


Figure 2: The line chart of loss through batches of one epoch. Note that here Cross Entropy is used for the loss function.

References

David E. Rumelhari, Geoffrey E. Hintont, Ronald, J., and Williams. Learning representations by backpropagating errors. 2004.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015a.

Dmytro Mishkin and Jiri Matas. All you need is a good init. *CoRR*, abs/1511.06422, 2016.

Hongyi Zhang, Yann Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *ArXiv*, abs/1901.09321, 2019.

Jiawei Zhao, Florian Schäfer, and Animashree Anandkumar. Zero initialization: Initializing residual networks with only zeros and ones. *ArXiv*, abs/2110.12661, 2021.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, 2009.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015b.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *ArXiv*, abs/1505.00853, 2015.

Pak Chung Wong and James J. Thomas. Visual analytics. *IEEE computer graphics and applications*, 24 5:20–1, 2004.

Daniel A. Keim, Florian Mansmann, and James J. Thomas. Visual analytics: how much visualization and how much analytics? *SIGKDD Explor.*, 11:5–8, 2010.

Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. Readings in information visualization - using vision to think. 1999.

James J. Thomas and Kristin A. Cook. A visual analytics agenda. *IEEE Computer Graphics and Applications*, 26:10–13, 2006.

Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in visual data analysis. *Tenth International Conference on Information Visualisation (IV'06)*, pages 9–16, 2006.

Daniel A. Keim, Gennady L. Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. In *Information Visualization*, 2008.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *2011 International Conference on Computer Vision*, pages 2018–2025, 2011.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, 1999.

Hao Li, Zheng Xu, Gavin Taylor, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.