Dynamic Analytical Modeling and Optimization of Critical Healthcare Data using Connectionism Systems

Gagana B
Computer Science,
PES University
Bangalore, India
gagana@pesit.pes.edu

H A Ujjwal Athri
Computer Science,
PES University
Bangalore, India
ujjwalathri@gmail.com

Dr. S Natarajan
Computer Science,
PES University
Bangalore, India
natarajan@pes.edu

***Abstract*—Classical Convolutional Neural Networks (ConvNets) have been the ruling benchmarks for most object classification and face recognition tasks despite major limitations such as the inability to capture spatial co-locality between data points and favoring invariance over equivariance mechanisms. Hence, Hinton et al, proposed a layered architecture called Capsule Networks (Capsnets) to overcome these shortcomings by replacing pooling techniques with dynamic routing abilities between lower level and higher level neural units which better capture hierarchical relationships within the data, thus, outperforming traditional systems. By overcoming existing limitations, Capsules have proven themselves to be potential benchmarks in object segmentation, detection and reconstruction. Capsules have achieved state-of-the-art results on the fundamental MNIST dataset by reducing the ConvNets test error benchmark of 0.39% to 0.25%. The two novel aspects inspected in this paper are the augmentation of this error benchmark distinction by optimizing the architecture through five activation units such as sigmoid, e-Swish, Swish, variants of Rectified Linear Units (ReLU) like Parametric ReLU (PReLU) and Scaled Exponential Linear Units (SELU) and the applicability of results obtained on visual data to stochastic numeric healthcare data uncovering newer challenges of predictive neural networks.**

Keywords—Activation function, Stochastic Numeric Data, Capsule Networks

I. INTRODUCTION

Machine Intelligence is deemed to be one of the most prodigious fields of research that has proffered a plethora of intuitive applications. Although, the brain's mechanisms of information processing are significantly different from that of convolutional systems, latest advancements like Capsule networks[1] apply the Hebbian Learning principles which are closer to emulating human capabilities.

The broad architectural framework of Hinton et al, Capsule networks has an encoder comprising of a convolutional layer, a PrimaryCaps layer and DigitCaps layer along with a decoder composed of three fully connected layers, FC#1 (With ReLU activation unit), FC#2 (With ReLU activation unit), and FC#3 (With sigmoid activation unit) which effectively reconstruct the encoded image while dealing with two performance parameters namely accuracy and loss. This loss, in turn, is broken down into margin loss and reconstruction loss. The detailed technical functionality of each of the encoder and decoder layers is explained as follows:

(a) The ReLU convolutional layer ingests the 28 x 28 image with one or more color channels and detects the basic features of the image while forming a feature map of the same in the form of a 20 x 20 x 256 tensor while dealing with 20992 parameters. This segment uses 256 9x9 kernels to generate an output with 256 channels (feature maps). With a stride of 1 and no padding, the spatial dimension is reduced to 20x20.

(b) The PrimaryCaps layers, which is a modified convolutional layer acting as a supporting structure, produces a combination of the above detected features; This accepts the output of the convolutional layer which is a 20 X 20 X 256 tensor and outputs a 6 X 6 X 256 tensor while dealing with 5308672 parameters. The input 28 X 28 image is reduced into 20 X 20 followed by a 6 X 6 in the Primary caps in terms of its spatial dimension. There are 32 capsule units in this layer, and a 8D vector that is generated to capture the position, texture, hue, color, type, and velocity amongst other parameters. PrimaryCaps layer uses 9x9 kernels with stride 2 and no padding to reduce the spatial dimension.

(c) the DigitCaps layers generates the transformation weight matrix W_{ij} and the corresponding loss function by the equation:

$$L_c = T_c \max(0, m^+ - \|v_c\|)^2 + \lambda(1 - T_c)\max(0, \|v_c\| - m^-)^2 \quad (1)$$

while dealing with 1497600 parameters.

The transformation matrix is used to transform the 8-D capsule to a 16-D capsule for each class j. Because there are 10 classes, the shape of DigiCaps is 10x16 (10 16-D vector.) Each vector v_j acts as the capsule for class j. The probability of the image to be classify as j

is computed by $\| v_j \|$.

$$\hat{u}_{j|i} = W_{ij} u_i \quad (2)$$

The final output v_j for class j is computed using the novel squashing function as:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (3)$$

where

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \quad (4)$$

(d) the three fully connected layers of the decoder serve the purpose of calculations for the number of parameters based on bias. The first and the penultimate layer of the decoder have the ReLU activation function while the last layer retains the sigmoid activation unit.

The first fully connected layer accepts the 16 X 10 matrix as input, directed to the 512 neuron units, processing 82432 trainable parameters.

The second fully connected layer accepts the output of the 512 neural units as input, passing the same through a network of 1024 neural units, processing 525312 trainable parameters

The final fully connected layer accepts the output of the second fully connected layer while passing the same through a 784 neural unit system dealing with 803600 trainable parameters which is the final 28 X 28 output.

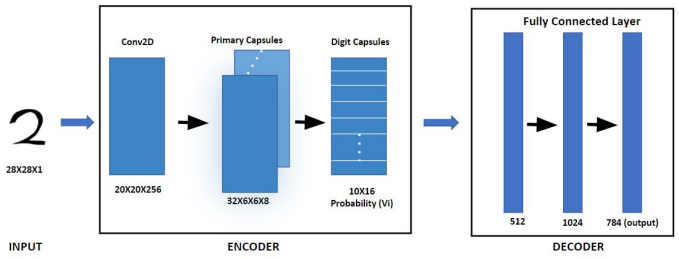

Figure 1. Block Diagram of Capsule Network Architecture

The total number of parameters in the capsule network are: 8238608.

A. *Activation Functions*

Activation functions or Transfer functions, as they are known, refer to non-linear transformations applied to each neural node in the network, enabling it to perform more complex tasks than simple linear regression.

$$Y = \sum (weight * input) + bias \quad (5)$$

Since 'Y' can range from negative infinity to positive infinity, the activation function is a significant feature applied over the input signal which decides whether a neuron fires or not depending upon the aggregate of bias and weight over input. This mechanism together with backpropagation iterates over the bias aggregate to update the gradients resulting in a loss metric which in case of capsules is comprised of reconstruction loss and margin loss. The mathematical definitions of the various activation function units are as follows:

The non-linear ReLU[2], defined as

$$A(x) = \max(0,x) \quad (6)$$

which capsules rely on, is one of the most widely used activation functions as they have proven to work consistently well across networks, easily backpropagating errors while activating multiple neural layers at once but not all, hence making the network efficient and sparse.

The nonlinear sigmoid function[3]

$$A = \frac{1}{1+e^{-x}} \quad (7)$$

which is bound in the range of [0,1] is a smooth, monotonic, real-valued, step like and continuously differentiable function. But however, the vanishing gradients that appear in the sigmoid function, tend to affect the learnability of the system as the rate drastically slows down in this range.

The parameterized ReLU function (PReLU)[4] defined as

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ ax & \text{otherwise} \end{cases} \quad (8)$$

is an improvised version of the leaky ReLU implementation, but ensures faster optimal convergence as parameter 'a' is learnable.

The SELU (Scaled Exponential Linear Units) [5] is defined as (9)

$$\text{selu}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leqslant 0 \end{cases}.$$

where alpha and gamma are fixed parameters learnt from the input and do not iterate through the network. Typically, for standard scaled inputs, the values are: α: 1.6732 and δ: 1.0507.

Recent developments in activation functions have paved way to newer activation units such as the Swish novel activation function defined as:

$$f(x) = x \cdot \text{sigmoid}(\beta x) \quad (10)$$

proposed by Ramachandran et al[6] and the e-Swish activation function[7] with a learnable beta component defined in terms of the sigmoid as:

$$E - swish = \beta x * sigmoid(x). \quad (11)$$

Our experimentation show the E-Swish and PReLU significantly and consistently outperform the ReLU activation function.

B. Healthcare

India's GDP with respect to healthcare remains as low as 1.2% as opposed to China's 5.5% or US's whooping 17%. As the government appraises the healthcare sector which is poised to grow, technological advancements are substantial for outreach in the 280 billion dollar market. To ensure affordability and meet the large scale demand for quality, research on healthcare systems is of atmost importance with accuracy being the key performance parameter. Thus, exploring this transformational space in terms of the latest machine learning systems like predictive neural net frameworks could be an influential progress in this direction.

The stochastic numeric UCI healthcare dataset[13] was mapped to a time series forecast timeline along with appropriate channel labels and fed into the conv2D layer which detects the basic features while forming a feature map in the form of a 20 x 20 x 256 tensor while dealing with 20992 parameters. The PrimaryCaps layers produces a combination of the above detected features and outputs a 6 X 6 X 256 tensor while dealing with 5308672 parameters. The DigitCaps layers generates the transformation weight matrix W_{ij} and the corresponding loss function while dealing with 1497600 parameters. The transformation matrix is used to transform the 8-D capsule to a 16-D capsule for each class. The three fully connected layers of the decoder serve the purpose of calculations for the number of parameters based on bias.

II. Related Work

Hinton et.al.[8] first proposed canonical object reference frames for shape perceptions and detection of spatial disposition which was employed in organising interactions in parallel networks. This further enabled organisation of the interactions between units in any given parallelised network so that the pattern activity in question simultaneously converges on a single representation. Hence, the concept is conceptualised using coordinate hardware, cooperative computation and parallelised frameworks as the processing of the coordinate frame inherently relative affects the environment in which the frame is based. Hence the architecture must implicitly pair the elements of the surroundings as activity in one can profoundly influence another in terms of associated channels like the size, position and orientation. This channel based frame reference stimulates a rough object segmentation.

This idea further evolved into the "Transforming autoencoders"[9] as vector representations of activity and instantiation parameters adapted more efficiently to the features of the domain than scalar references. This model is altered to suit the different viewing conditions of an implicitly defined visual entity where the recognition probability is multiplied element-wise to the capsule output as the implicit routing feature learn feature detections over time and produce explicit vector output representations of instantiation parameters over scalar ones. The probability of visual entity is expected to be invariant as the entity moves over the manifold of possible appearances, while instantiation parameters are equivariant. As the viewing conditions change and the entity moves over the appearance manifold, the instantiation parameters change by a corresponding amount because they are representing the intrinsic coordinates of the entity on the appearance manifold and this hence, is a more promising approach for the same over classical neural networks.

Thus Capsules, which is a nested set of neural layers, performed dynamic routing mechanisms of selected features more efficiently by denoising features at lower lower capsules before hierarchically routing to higher level capsules which would then uncover more intricate patterns within the data. Vector representations perform internal computations at each layer which summarise the activities of the local pool resulting in highly informative concise outputs due to their internal denoising properties.

With respect to improving the efficiency of capsule networks, Hinton et al[10] also proposed EM routing algorithm which is the expectation maximisation logistic unit recursively updating the weighted assignment coefficient matrix clustering probabilities that are closer to each other. The algorithm not only paved way for effective representation of part whole relationships but also increased

capsules robustness towards adversarial attacks proving significantly lesser vulnerability than baseline ConvNets. This technique on the NORB dataset cut the state of the art test error benchmark by nearly 45% .

Capsules in recent times, has been applied to application domains such as healthcare on the following datasets:

(a) Lung disease dataset[11] where the CapsNets architecture was trained on Convnets preprocessed data where the model was first scanned for attributes such as sex, age, noise filtering and then resorts to resizing after which it is tested based on convolutional neural units ability to accelerate the convergence using pre-trained models and optimise them using spatial transformation techniques. CapsNets proved that they can thrive with minimal data while far exceeding the ConvNet benchmark of 71% in terms of the area under the receiver operating characteristic curve.

(b) Brain fMRI images[12] proposes a new CapsNet architecture based reconstruction mechanism to reconstruct image stimuli by comparing with the goal of decoding orientation, position and object category from activities in visual cortex, mainly aimed to answer the open Neuroscience question of how sensory stimuli are encoded by neurons and conversely, how sensory stimuli can be decoded from neuronal activities mainly using fMRI images.

Other applications include niche domains like fluid physics where traditional linearly iterative computation models have been replaced with nested networks to extract complex 3D features while generating a reference map charting out feature types and coordinate grids.

Other systems such as CapsuleGAN [13] have been explored where the discriminator is replaced by a capsule network to model image data with a different objective function evaluated qualitatively and quantitatively on the Generative Adversarial Metric (GAM) and at semi-supervised image classification while maintaining the margin reconstruction loss. This objective of CapsuleGAN can be mathematically summarised as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \quad (12)$$

The aforementioned papers have laid the basis for the proposed methodology articulated below.

III. EXPERIMENTATION

The ReLU activation unit in the capsnet architecture was replaced by the SELU, e-Swish, Swish, PReLU, and the sigmoid in various different separate experimental setups, each of which are executed on the MNIST dataset. This was executed for approximately 62 epochs as we found that the results could be extrapolated since functions fairly stabilised thereafter.

A keras implementation with a tensorflow backend in a jupyter notebook environment run on a tesla k40c GPU configuration was the key framework. However, each of the activation functions were executed in the same framework environment with accuracy measures on the MNIST as the performance parameter with ReLU as the baseline benchmark.

IV. RESULTS AND DISCUSSIONS

It was observed that the sigmoid yields the least results at 11.27% which could be due to the vanishing gradients problem. Sigmoid activation function mathematically, involve inherent expensive operations which ReLU is able to optimise by thresholding. ReLU also deals with sparsity issues than the sigmoid as sigmoid tends to generate non-zero value resulting in dense representations thus, affecting performance.

As far as SELU is concerned, although SELU outputs normalized activations to the next layer, ReLU converges faster than SELU. SELU is expected to perform more effectively under certain conditions such as:

(a) **Theorem 1:** Since SELU is intuitively self normalising, high variance in one layer is mapped to lower variance in the next layer, and this alternating variance works since: SELU decreases the variance for negative inputs and increases the variance for positive inputs.

(b) **Theorem 2:** To prevent gradients from exploding, the mapping of variance is bound at the upper limit.

(c) **Theorem 3:** To prevent vanishing gradients, the mapping of variance is bound at the lower limit.

but is computationally more effectively than ReLU. But the limits defined by Theorem (2) and (3), is the range where SELU perhaps is at the optimal best.

Due to factors such as variance damping of SELU which doesn't fall under the conditions under which most other activation functions in question thrive, SELU isn't batch normalised. However, SELU on an average performed comparably to ReLU at 99.38 while ReLU set the benchmark at 99.40.

The results for the Swish are as follows: 99.06326531 on an average, with maximum accuracy value of 99.24 at beta values of 0.70783865 and 0.7197824 with corresponding loss of 0.165600 and 0.70783865. The least accuracy value is recorded at 97.9 with a loss of 0.468200 at a beta value of 0.822877. Although the beta values have to be learnt which affects the convergence rate, it is observed that higher values

of beta provide faster learning. Hyperparameter optimisation inefficiencies could be a probable reason for the said accuracy rate of Swish as opposed to ReLU.

The results for the e-Swish, and PReLU in terms of accuracy as opposed to ReLU is as tabulated below.

	Maximum Value	Minimum Value	Average Value
ReLU	99.46456453	98.680815	99.36171797
PReLU	99.54000115	99.05999899	99.42440876
e-Swish	99.56	99.34	99.44

Table 1. PReLU and e-Swish accuracy with reference to ReLU baseline

The results show that the e-Swish, Swish and PReLU consistently outperform the ReLU benchmark on the MNIST Dataset. A graphical representation of the same is as follows where the y axis represents the accuracy and the horizontal x axis represents the epoch value.

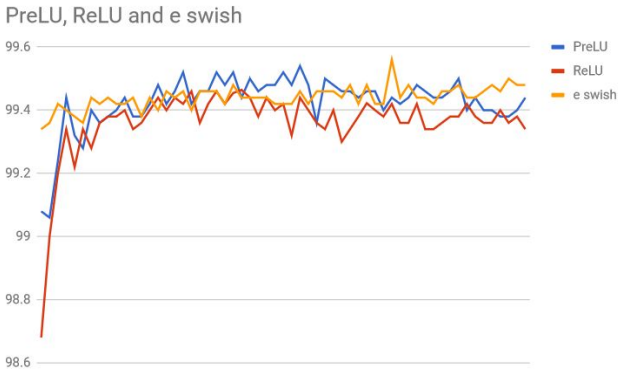

Figure 2. PReLU and e-Swish accuracy with reference to ReLU

The other performance parameter that plays a significant role in this context is loss, as improvements between consecutive iterations are benchmarked against this attribute. A tabulation of the obtained results of the PReLU and e-swish with reference to ReLU are as follows:

	Maximum Loss	Minimum Loss	Average Loss
ReLU	0.4211860299	0.1490184814	0.1653636606
PReLU	0.2821122408	0.1474184841	0.1588650043
e-Swish	0.246235	0.144173	0.1507350678

Table 2. Comparison between ReLU, PReLU and e-Swish with reference to loss parameter

e-Swish outperforms both PReLU and ReLU in terms of the loss metric. A graphical representation of the same is as follows where the y axis represents the loss and the horizontal x axis represents the epoch value.

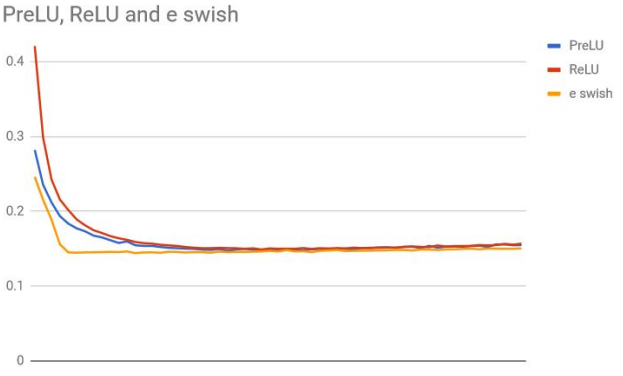

Figure 3. Graphical Representation of ReLU, PReLU and e-swish with reference to loss

As discussed above, the Swish and ReLU variants outperform the benchmark on the MNIST dataset. These activation functions could be experimented with different datasets in terms of volume and complexity amongst other parameters. Other newer functions such as the leaky ReLU implementation or tweaks to the existing sigmoid function would be research avenues worth exploring.

The survey with healthcare data has proved to be phenomenal with 19.5% increased relative correlation as compared to previous benchmarks of linear regression[14] on a subset of the dataset. These results are currently being extrapolated to cancer research models which are expected to surpass the ConvNet accuracy benchmarks leading to implications and inferences on demographic constitutions.

V.. CONCLUSION

From the experiments, it is evident that the e-Swish, and PReLU better optimise the capsule architecture than the currently used ReLU in terms of better accuracy and ensuring faster convergence, hence, lesser training time. These activation units outperform the state of the art accuracy benchmarks on MNIST dataset. Future work may be carried out using newer and novel functions applied to more complex models.

The non-normalized, distributed data with changing behavioral attributes and complex curves often pose new challenges. This ambitious research venture could redefine modern processing with respect to time series analysis and forecasting where age old contemporary systems seem to have failed miserably with techniques that are possibly be profoundly flawed.

With the consideration of the aforementioned ideology, these newfangled architectures are expected to rule and drive systems of the future where technologies are rapidly advancing and landscapes are fast changing. While current technological revelations delineates the aforementioned scenarios, the future could spur out formidable and impressive inroads to advancements to not only effectively tackle current challenges but to create and solve newer

problems in this space that we don't even know exist yet

VI. REFERENCES

[1] Sara Sabour, Nicholas Frosst, Geoffrey E Hinton, "Dynamic routing between Capsules" in Computer Vision and Pattern Recognition (cs.CV), arXiv:1710.09829v2, October 2017.

[2] Krizhevsky, I. Sutskever, and G. Hinton. "Imagenet classification with deep convolutional neural networks" in NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Pages 1097-1105, December 2012.

[3] LeCun, Y., Bottou, L., Orr, G. B., & Muller, K.-R., "Efficient backprop. In Neural networks, tricks of the trade" in Lecture Notes in Computer Science LNCS 1524, Springer Verlag, 1998

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification" in International Conference on Computer Vision, February 2015

[5] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, "Self-Normalizing Neural Networks" in arXiv: 1706.02515, September 2017

[6] Quoc V. Le, Prajit Ramachandran, Barret Zoph. "Swish: a Self-Gated activation function." in Neural and Evolutionary Computing (cs.NE); Computer Vision and Pattern Recognition (cs.CV); Learning (cs.LG), arXiv: 1710.05941, October 2017.

[7] Eric Alcaide, "E Swish: Adjusting Activations to Different Network Depths" in Computer Vision and Pattern Recognition (cs.CV); Learning (cs.LG); Machine Learning (stat.ML), arXiv: 1801.07145, January 2018

[8] GE Hinton, F Cambridge, "Shape Representation in parallel systems" in International Joint Conference on Artificial Intelligence, 1981

[9] Hinton G.E., Krizhevsky A., Wang S.D., "Transforming Auto-Encoders" in Honkela T., Duch W., Girolami M., Kaski S.(eds) Artificial Neural Networks and Machine Learning – ICANN 2011, Lecture Notes in Computer Science, vol 6791. Springer, Berlin, Heidelberg

[10] Sara Sabour, Nicholas Frosst, Geoffrey E Hinton, "Matrix Capsules with EM Routing" in 6th International Conference on Learning Representations, ICLR, February 2018

[11] Leslie H. Sobin M.D.,Irvin D. Fleming M.D. , "TNM classification of malignant tumors" in Wiley online library, 2000

[12] Kai Qiao, Chi Zhang, Linyuan Wang, Bin Yan, Jian Chen, Lei Zeng, Li Tong, "Accurate reconstruction of image stimuli from human fMRI based on the decoding model with capsule network architecture" in Computer Vision and Pattern Recognition (cs.CV); Artificial Intelligence (cs.AI); Neurons and Cognition (q-bio.NC), arXiv: 1801.00602, January 2018

[13] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, Premkumar Natarajan, "CapsuleGAN: Generative Adversarial Capsule Network" in Machine Learning (stat.ML), Learning (cs.LG); arXiv: 1802.06167, March 2018