REPRESENTING INFORMATION AS LINEAR COMBINATIONS OF SIGNALS

David Banh
Data Science
AskExplain
St Lucia, Brisbane, Australia
david.b@askexplain.com

ABSTRACT

Here it is proposed that a linear set of parameters learned when adhering to informative properties can enhance signal extraction in neural networks. The parameters would have to adhere to certain properties such as maximally representing signal from observed data - such as orthogonality. Coupling a linear model that adheres to certain structure with a convolutional layer in a deep learning system also gives comparable results to a sequential layer of linear layers with activation functions.

1 Introduction

A linear model finds the optimal function between observed data and predicted estimates of the response variables. When transitioning to matrices of datasets from vectors, a linear model would need to learn a transformative projection with certain properties [1]. Ideally, the projection would have to optimally represent the data to the maximal extent, usually done via orthogonal signals when data is limited.

When applied to images, this would mean a set of orthogonal image filters that can recompose the observations depending on the total amount of dimensions chosen by the data scientist.

The concepts of transformations via encoding and decoding are used often in the machine learning literature on neural networks - especially in autoencoders and transformers [2], and a previous encoding-decoding concept based on Singular Value Decomposition and Procrustes analysis [3]. The main idea is to find an efficient and effective way to learn relevant signals from associative structures across features.

2 Model

Given the response variable Y to be estimated using the main data X as the observed data by learning a projection β, there would be properties instilled into the model that can enable it to extract relevant signals from the residuals.

To transform the signals in the observed data into an estimation of the response variable, β would need to represent the properties of orthogonality, that is

$I = \beta^T\beta$

For example, to put this into the linear model, $Y = X\beta$, the expression can include the orthogonal property by

$$Y = X\beta(\beta^T\beta) = X\beta(\beta^T\beta)(\beta^T\beta)$$

Note that this is a property of the identity.

Furthermore, drawing inspiration from the ideas in Generative Encoders [3], each dataset is decomposed into a common latent code, and the transpose of the transformation projection parameter.

$$X = Z\beta^T$$

this implies the following:

$$X\beta = Z\beta^T\beta$$

Which attempts to find the parameters β such that the covariance of the signals represented by β when composed with the latent sample space Z reformulates the expression for $X\beta$ via $Z\beta^T\beta$.

2.1 Properties of Orthogonality

For more details on orthogonal signals, it is possible to extract relevant signals from the data by looking at key properties important for the model. Theoretically the signals would have to cover as much of the information space as possible. For this to occur, an orthogonal set of signals would not overlap while also attempting to cover as much as the information space as possible.

When parameters α, β, or u are orthogonal it is possible to rewrite an expression that can be included in a neural network without need for generalised inverse operations. When parameters are orthogonal, the fixed effects part can be expressed as:

$$\beta^T\beta = I$$
$$K = K\beta^T\beta = X\beta$$
$$X\beta = X\beta(\beta^T\beta) = X\beta(\beta^T\beta)(\beta^T\beta) = X\beta(\beta^T\beta)(\beta^T\beta)(\beta^T\beta)$$

Here a series is created where the orthogonality property of β creates a series of equalities that implies the structure of a loss function ready for a neural network. For approximation and appropriate runtime with adequate convergence, in classification image tasks of labels Y and images X with convolution $f(X)$, the loss function would then be:

$$Cross\ \ Entropy\ \ (Y, f(X)\beta)\ +$$
$$Cross\ \ Entropy\ \ (Y, f(X)\beta(\beta^T\beta))\ +$$
$$Mean\ \ Squared\ \ Error\ \ (f(X)\beta, f(X)\beta(\beta^T\beta))$$

This also prevents prohibitive inverse operations over dimension sizes of approximately greater than 1000 components and allows neural networks to extend to sizes limited only by computational time of matrix multiplication (or standard neural network operations).

3 Results

Two figures are given, one for a base convolutional neural network and the other using a VGG neural network comparing a sequence of linear layers with activation functions and a summarisation layer.

3.1 Base convolutional neural network

Results show that a base convolutional neural network outperforms a summary layer on CIFAR10. This shows that summarisation can still be extended or improved.

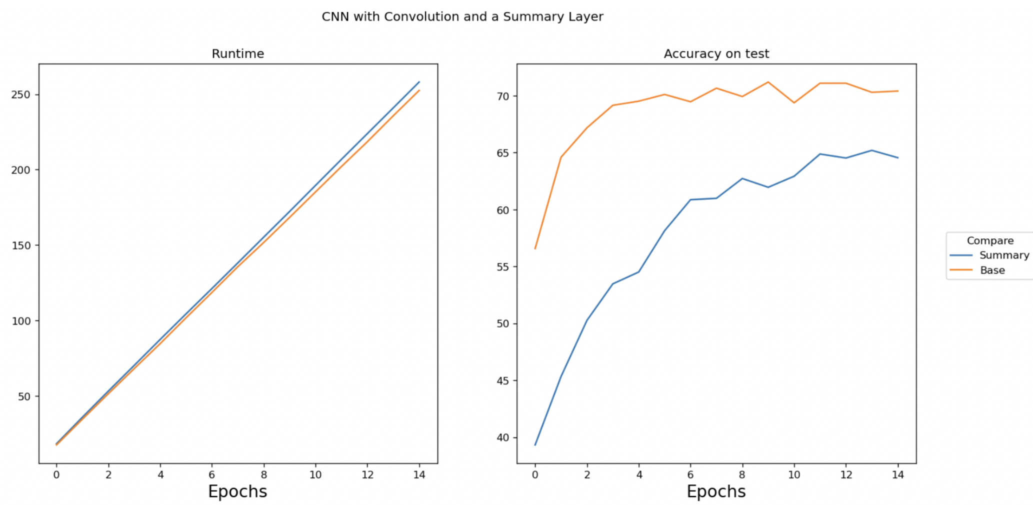

Figure 1: A base convolutional neural network comparing a nonlinear sequence of layers with a linear layer of summarisation

3.2 Base VGG neural network

Results show that a base VGG neural network performs comparably to a summary layer on CIFAR10. This shows that the convolution layer structure affects the ability of a summarisation layer.

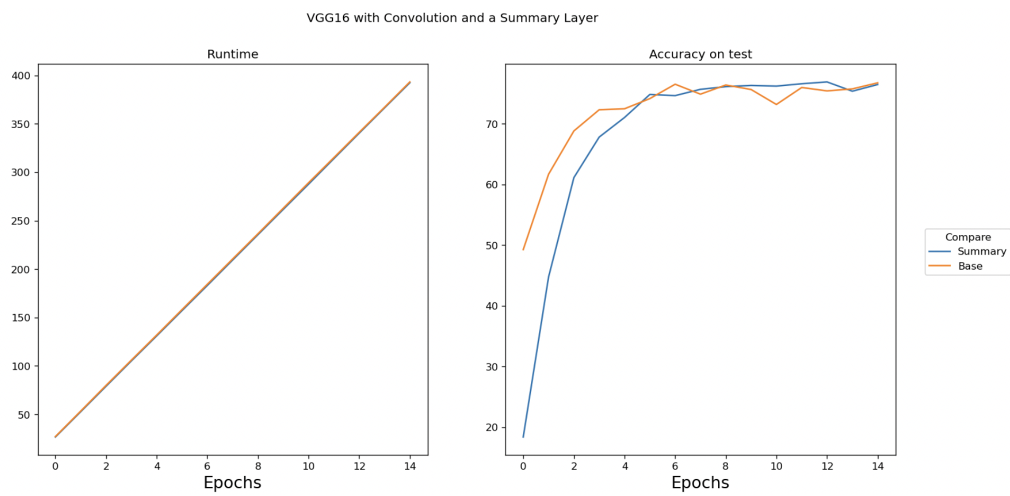

Figure 2: A VGG neural network comparing a nonlinear sequence of layers with a linear layer of summarisation

4 Acknowledgements

David Banh would like to acknowledge:

Alan Huang for relevant discussions of Generative Encoders, and work relating to linear mixed models. Also for being open and generous enough to discuss ideas, even when it was just a small one.

Ryan Deslandes (University of Queensland, Australia) for assistance with software development of the genecode dashboard (no longer in use): `https://board.askexplain.com/genecode`. This idea was presented in the NanoString Hackathon earning third place and bonus prize for best images (first and second place went to the teams from Dr Omer Bayraktar's Lab from the Wellcome Sanger Institute, one of which presented cell2location).

Cameron Gordon and Olivia Ou (University of Adelaide, Australia and University of Queensland, Australia) for consistent support and advice throughout the project.

Alex Alsaffar (University of Queensland, Australia) for relevant discussions on the model of Generalised Canonical Procrustes and Generative Encoding as well as work on a preliminary Python version of gcproc (a prior version of gcode): `https://github.com/thisismygitrepo/gcprocpy`.

Dr Quan Nguyen (University of Queensland, Australia) for a brief discussion on the model of corevec (a prior version of gcproc, which is an earlier version of gcode) regarding imputation and alignment in early 2021 https://github.com/AskExplain/corevec

References

[1] Cinzia Viroli. On matrix-variate regression analysis. *Journal of Multivariate Analysis*, 111:296–309, 2012.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] David Banh and Alan Huang. Scalable parametric encoding of multiple modalities. *bioRxiv*, 2022.