Deep Learning with PCA Wavelet Networks

Report Name	Project Outline
Author (User Id)	Samuel Spurling (sas90)
Supervisor	Bernie Tiddeman (bpt)
Module	CS39440
Degree Scheme	GG47 (Computer Science and Artificial Intelligence with integrated year in industry)
Date	February 5, 2022
Revision	0.1
Status	Draft

1.   Project description

My project is developing extensions to the paper Principal component wavelet networks for solving linear inverse problems [1]. This paper proposes a new technique for solving linear inverse problems, with impressive results on computer vision tasks. For example, the PCAW network has a PSNR score of 29.98 (std of 2.2) for super resolution tasks, which makes it competitive with GAN based models. However, this model has the advantage of shorter training time, and is less prone to overfitting.

A linear inverse problem is where we have a set of measurements and are trying to predict the state of the object from which we obtained the measurements. In most cases the object is something we can't directly observe. The inverse problems of interest to us are ill-posed, which means they violate at least one of the following conditions [2]

1) There exists one solution (Existence)
2) There is at most one solution (Uniqueness)
3) The solution depends continuously upon the data (Stability)

One example of a linear inverse problem is super resolution of images, where given some image we attempt to accurately reconstruct the image in a higher resolution. The forward process is obtaining a low-resolution image from a higher resolution image, which is easily done by removing pixels. The reverse process is much harder.

There are several natural extensions of the research, such as editing the architecture of the network. I can also look at evaluating the model on novel applications. This will be discussed in Proposed Tasks.

The project will use several datasets. The two I am using at the time of writing are the CelebFace Attribute and ImageNet datasets. These are the two datasets used in the original paper. I may also use graph and text datasets, as I believe that the network could be extended for problems in graph-based machine learning and natural language programming (NLP).

I will be writing the code in python as a notebook. The platforms I will be using will be Jupyter Notebooks, Google Colab and spyder. The device I am using for most of the work will be my personal laptop (8.00 GB of RAM, Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz). I may also use a free GPU or TPU in google Colab, and a GPU cluster provided by the university.

For version control I will be using GitHub, for issue tracking I will be using either Gitlab or click up for issue tracking.

2.   Proposed tasks

The first task will be to reimplement this project. There is code from the original authors which I will reuse. After getting the original code to work on my local machine I can begin building new features. This is clearly a primary task, as no other work can be completed without the completion of this task.

After this initial tasks, I will explore the effect of basic edits to the network architecture. This may include, but will not be limited to, adding regularisation, adapting network size,

experimenting with activation functions, and changing how the network is trained. This is another primary task.

I will also find a novel application of the technique, outside of the applications proposed in the original paper. Two problems I have discussed with my supervisor is crowd counting and image segmentation. There are a number of linear inverse problems outside the field of computer vision. For example, predicting missing words in text. It would be interesting to explore problems like this too. This will be a primary task.

The original network is a Convolutional Neural Network (CNN), with hand-derived filters. A lot of state-of-the-art models for computer vision actually have a transformer architecture. Another task could be to see if I can apply PCA wavelets to transformers and see what effect it has. This will be a secondary task, as the transformer architecture is quite different to CNNs and there could be difficulties in getting the PCA decomposition to work.

I will explore the effect of replacing PCA with other dimension reduction techniques. The first substitution will be a linear technique, Independent Component Analysis (ICA). The second substitute will be a non-linear technique, Uniform Manifold Approximation and Reduction. In particular, I will be exploring UMAP and other techniques to deal with non-linearities in the datasets. This will be a secondary task.

The original code tested the network for multiple tasks: inpainting, super-resolution and compressive sensing. I will be evaluating any new architectures I create on these tasks and comparing them to the original model's performance and the other reported baselines.

Throughout the project there will be a lot of research into different parts of the project, such as network architectures and dimensionality reduction techniques.

3.   Project deliverables

Software - At the end of this project, I will deliver a number of python files. Some will be in a notebook (.ipynb) format, and some will be in a standard (.py) format. The software will fulfil the previously stated tasks. This will include source code and a demonstration of model performance through experiments.
I will also generate files for the models I create, saving the weights and architectures.

Documentation – There will be documentation to explain the code I write and help a user to run and interpret the experiments. I will also detail the architecture of my models.

Blog – I will be blogging about the progress of my project weekly. I will aim to post every Sunday. The blog will concern research I have undertaken, and the progress of the software I am creating.

Report – The report will be in a research template. It will detail the processes I have taken when experimenting with architectures. I will document my design process and experimental results in a final report.

Demonstrations – There will be 2 demonstrations throughout the project, mid-project and final. The mid-project demonstration will be indicative of the progress of the project to that point, and the final demonstration will be a display of the final piece of software.

4.   Initial annotated bibliography

[1]   B. G. M. Tiddeman, "Principal component wavelet networks for solving linear inverse problems" *Symmetry,* vol. 13, no. 6, 2021.

This is the paper my project will be based on. The paper details the architecture of PCWNs and discusses experiemntal results for the network compared to other models.

[2]   [LectureNotes2020.pdf (cam.ac.uk)](LectureNotes2020.pdf)

The lecture notes found here give a very in-depth explanation of an inverse linear problem and discuss common methods of solving them.

[3]   L McInnes, J Healy, J Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction" 2018

Details the UMAP algorithm, a dimension reduction technique for handling non-linear data with roots in Riemannian geometry and algebraic topology. Has better global structure preservation than T-SNE