1 Deep Learning Prognostic Tools

1.1 Introduction

In recent years research a subcategory of machine learning called 'deep learning' has rapidly emerged to become the state-of-the-art in field of artificial intelligence. Deep learning is able to achieve superior performance where previous approaches have fallen short, and has facilitated the development of many useful applications [1]. Deep learning is a broad term covering a large swathe of mathematical models all comprising a multi-layered ANN of some form. The recent surge in interest in ANNs has been driven by a number of factors. In the last twenty years computing power, storage, and the availability of data have increased dramatically [2]. This has facilitated a paradigm shift in the way that artificial intelligence systems are created; modern algorithms have been developed which learn from data very efficiently, rather than being explicitly programmed by a human to accomplish a given task. Due to the highly parameteric nature of deep ANNs, large quantities of data are required to obtain optimal model configurations, this data-focused approach is responsible in large part for the recent success of ANNs [2].

ANN CNN DEFINED?

To evaluate the potential of CNNs as a prognostic model for oral cancer, a one-dimensional CNN was created to establish if any improvement was gained by adding spectral information. The network follows a similar structure to a 2D CNN; the objective is that the network would extract higher level features from

raw absorbance values — potentially correlating to levels of known biochemicals. A crucial factor in the transmission of FTIR microscopy into a clinical setting is its ability as a technique to be universally adopted. To do this factors including sample preparation, measurement environment, and measurement technique must be as uniform as possible. Preprocessing data is an attempt to mitigate the effects of potential inconsistencies in measurement practice but the process is never perfect. It would be desirable for any analysis method to be able to obviate the need for preprocessing all together by being robust to invariances introduced by experimental practice alone. A CNN is able to manage this to some extent due to the convolution layer [3]; by effectively scanning the entire spectrum and learning how to recognise patterns spanning multiple wavenumbers, the model is robust to slight alterations in wavelength-dependent absorbance. Furthermore, generating artificial data with these measurement aberrations applied randomly could potentially encourage the learning process towards more robust configurations. For example, adding an offset to each spectra would simulate different sample thicknesses; artificial Mie scattering effects could be added to allow the network to learn to ignore the effects of scattering.

Existing work

Existing work [4] leveraged 2D CNNs to dicriminate between different grades of breast cancer from a cohort of 96 patients. The study followed a similar routine to what was carried out previously in **??** where a series of datasets were drawn without replacement to obtain distributions of classification scores. It was found that the addition of spatial information from convolutional layers improved the model performance considerably over pixel-level predictions of a large range of models including: SVM, random forest, and an RBF kernel. Adding spatial information increased the overall accuracy of the predictions by $\sim$20%; the

SVM RBF DEFINED?

specificity and sensitivity increased considerably with one class improving by $\sim$60%.

A one-dimensional neural network was used to classify FTIR, Raman, and near infrared (NIR) data derived from a variety of food samples [3]. The ANN developed by the authors was a shallow CNN, which improved on overall accuracy scores of other classifier methods on preprocessed data from 62% 86%; and from 89% to 96% on raw data. CNNs were used with success on ATR-FTIR data in forensics to detect synthetic cannabinoids [5]. The authors found that CNNs were capable of identify synthetic cannabinoids, achieving 98.7% accuracy and an F1 score of 98.5% — meeting the standards of a forensic screening system. CNNs applied to vibrational spectroscopy have enjoyed further success in forensic applications where they employed to identify amphetamines with an accuracy of over 90% [6].

MLPs have already been utilised with vibrational spectroscopy to detect breast cancer in ATR-FTIR data [7]. The authors used three ANNs in a 10-fold cross validation scheme to discriminate between FTIR spectra collected from 78 malignant and 88 benign breast tumours. The authors found that ANNs had superior performance across many classification statistics in comparison to many other classifiers. The ANN models however did not perform as well as a SVM classifier on the same data.

1.2 Materials and Methods

Convolutional neural networks

As outlined in **??** an ANN consists of a number of layers, each of which comprise a number of nodes. Nodes in a NN are a representation of a relatively simple equation known as the perceptron equation (**??**); which are combined

in complex ways and used as a highly parametric model of a particular inference problem. A neural network trained using labelled data will optimise free parameters within the network to minimise a loss function constructed for the problem at hand. In particular, neural networks have allowed for advances in applications where data contains temporal or spatial information. Convolutional neural networks (CNNs) are a type of network containing specialised layers capable of extracting spatial information from data. This is accomplished with the use of a kernel which is convolved over the input data. A kernel comprises a number of parameters which are refined during a training phase to capture the most relevant spatial information. The first few layers of a CNN are utilised as a method of feature extraction; values of the convolution kernels are refined progressively to extract useful spatial features in the data; these features are then fed into a standard MLP network for further feature extraction, and later classification or regression.

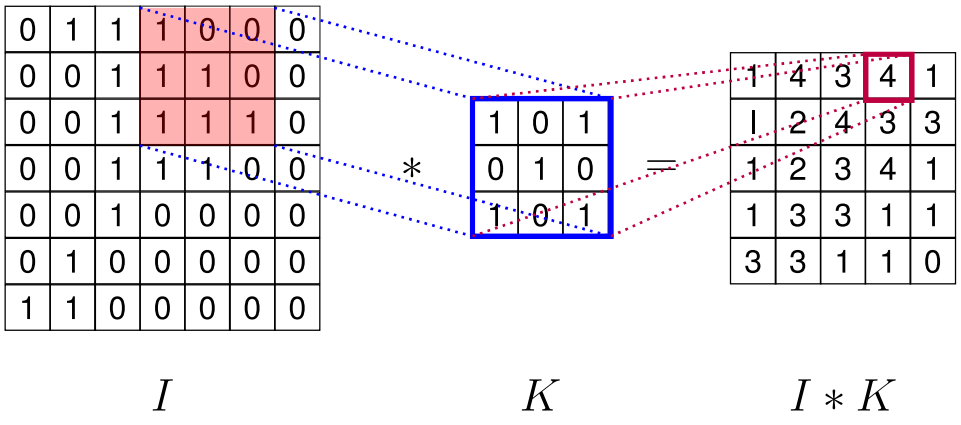

FIGURE 1.1: A 2D convolution layer showing a simple (3×3) kernel K convolved over an input image I of size (m×n). The resulting convolution $I * K$ is effectively a spatial map of where in I most closely resembles K.

This convolution operation can be expressed formally as:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \tag{1.1}$$

The kernel K is moved across the dimensions of the data where it is multiplied by the values in the kernel. A mapping of the similarity of the data to the kernel at that point is obtained then aggregated, typically a maximum or mean is taken and the values are pooled and assigned to an output. The behaviour of a kernel layer is dictated by its size, stride — the number of elements the kernel shifts per iteration, and dilation — the mapping of kernel elements to non contiguous elements of the data. Like other layer types convolutional layers can be altered by many parameters but will not be discussed further here. Depending on the dimensionality of the data the kernel can be combined with other kernels to find correlations between them; allowing colour information in 2D images to be used. A bias term is added to the pooled value and subjected to an activation function like in a normal perceptron layer. The next layer is obtained by convolving over the preceding layer using another kernel, this continues for a number of layers. The input is flattened into a one-dimensional vector where it is passed into a normal MLP where it reaches a softmax layer output. This softmax layer eq. (1.2) outputs scores which sum to one and can be loosely interpreted as probabilistic predictions of a given class.

$$softmax_k(x) = \frac{e^{W_k^T x}}{\sum_{i=1}^{n} e^{W_i^T x}} \tag{1.2}$$

Intermediate steps between these layers can be introduced to assist with regularisation such as dropout layers, batch normalisation layers [8], and many others. Dropout layers [9] are often included in ANN architectures as they provide a strong regularisation effect during the training phase. A simplified representation of a CNN showing the sequential nature of the described layers is shown in fig. 1.2. Dropout layers are typically implemented in a similar way to regular hidden layers, however they differ in that when any forward pass occurs in the training stage a node may become inactive, preventing any change in the

weights of connected nodes for that particular pass. Each dropout layer typically has a probability associated with it which dictates the chance of becoming inactive. Dropout works effectively by discouraging weights from converging towards similar values — encouraging redundancy in the network structure.

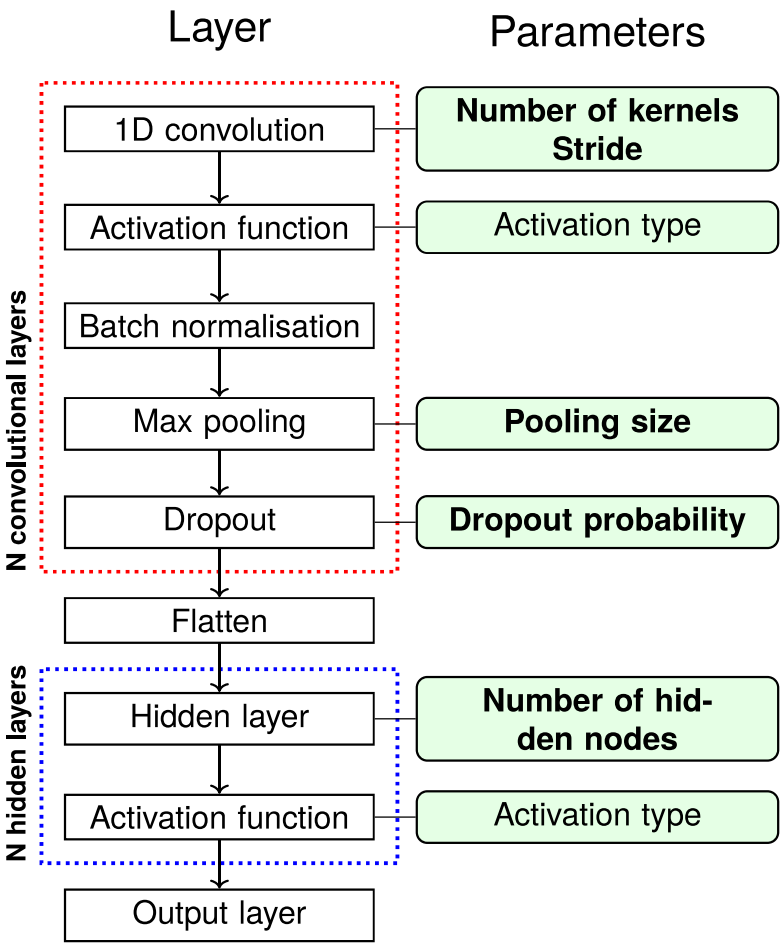

FIGURE 1.2: An typical convolutional network example pipeline; parameters associated with each step are shown in green; extra parameters are shown in bold.

Batch normalisation layers were included in the CNN model to increase the speed and efficiency of the training procedure. Batch normalisation layers work by normalising the distribution of values flowing from one set of nodes to the next, by scaling to a given range. This helps to prevent issues associated with exploding/vanishing gradients whereby update values for nodes increase or decrease rapidly to the detriment of the training process.

Furthermore, many parameters are associated with aspects of the training procedure of the network itself. The objective of the optimisation procedure used to train neural networks is to maximise or minimise a score with respect to the network parameters. Number of optimisation techniques are used but stochastic gradient descent (SGD) was the chosen method for both ANNs described here. SGD works by computing this objective score on a randomly selected subset of the training data. This is beneficial for large datasets where computing on the entire set may be computationally infeasible. When an update occurs a coefficient called the *learning rate* is used to dictate the weighting of the new value. Another parameter used to influence the optimisation strategy is the *weight decay*; a coefficient used to alter the effect of the gradient value on the objective function. A thorough description of SGD and its associated parameters is available from [10]. Another commonly used technique to improve convergence during the training stage is to initialise neural network weights. Weights were intialised using the method described by Kaiming [11]; initialising layer weights has been shown to decrease convergence times and improve the stability of the optimisation procedure.

To evaluate the benefit of utilising convolution layers to analyse spectra, a comparison of a CNN model was made with a MLP. Both ANN models were constructed, trained, and evaluated using an open-source python libary PyTorch [12]. Additional packages [13, 14, 15] were leveraged to implement the evaluation procedure along side other common machine learning operations. The same procedure as discussed in **??** was followed; out of bag sampling was utilised to obtain distributions of classification values, inverse weighting was used to mitigate the effects of dataset imbalance.

The dataset comprised FTIR spectra taken from primary tumour sites of 29 patients with a diagnosis of OSCC. Inclusion criteria for this study were as previously described in **??**: a diagnosis of OSCC; the presence of OSCC in the

TMA core; the ability to co-register adjacent H&E stained and FTIR imaged sections; a follow-up period after surgery of at least 24 months; HPV negative.

Images were acquired at a resolution of 6cm^{-1} over a spectral range of 990cm^{-1} to 3800cm^{-1} using a co-addition of 128 scans. Attenuator and integration time of the focal plane array (FPA) were chosen to gain the maximum signal-to-noise ratio. Background scans were acquired using a blank CaF_2 disk situated within the perspex box before each session of measurements.

The preprocessing steps required for each type of network are slightly different. Given that the convolutional layers in the CNN model are used to extract features from multiple wavenumbers simultaneously, the only preprocessing step is to normalise the data. For the MLP model vector normalisation was used in order to account for sample thickness; wavenumber absorbance features were mean-centered; and variance scaled to one; before a final PCA step to reduce dimensionality of the dataset.

1.2.1 Optimisation of network structure

The sheer number of tunable parameters associated with ANNs necessitates a hyperparameter search similar to that described in **??**. The open source optimisation framework Optuna [16] was used to determine and optimal network structure and associated hyperparameters. The hyperparameters in bold in **??** were chosen for optimisation for the CNN network; in addition the learning rate and weight decay parameter were included as optimal values are task-dependent and have a large impact on the training efficiency of ANNs [17, 18]. The median AUROC value was calculated across a five-fold cross validation of data subsets to determine the general suitability of the network configuration. Fifty sequential trials were chosen to allow sufficient exploration of the parameter space.

Convolutional network

A summary of the configuration determined by the procedure is given in table 1.1

TABLE 1.1: Optimal convolutional neural network hyperparameters and values.

Parameter name	Value
N convolutional layers	5
N kernels in convolution layer 1	96
N kernels in convolution layer 2	128
N kernels in convolution layer 3	32
N kernels in convolution layer 4	80
N kernels in convolution layer 5	128
Maxpool 1 size	3
Maxpool 2 size	3
Maxpool 3 size	7
Maxpool 4 size	5
Maxpool 5 size	3
N fully connected nodes	80
Dropout probability	0.45
Learning rate	8×10^{-5}
Optimum value	0.84

A simplified diagram of the CNN network fig. 1.3 configuration determined by the optimisation procedure is shown below. The CNN network contains five convolutional and maxpooling layers of varying sizes. Deeper network designs are typically better at extracting high-level structural information in data [19].

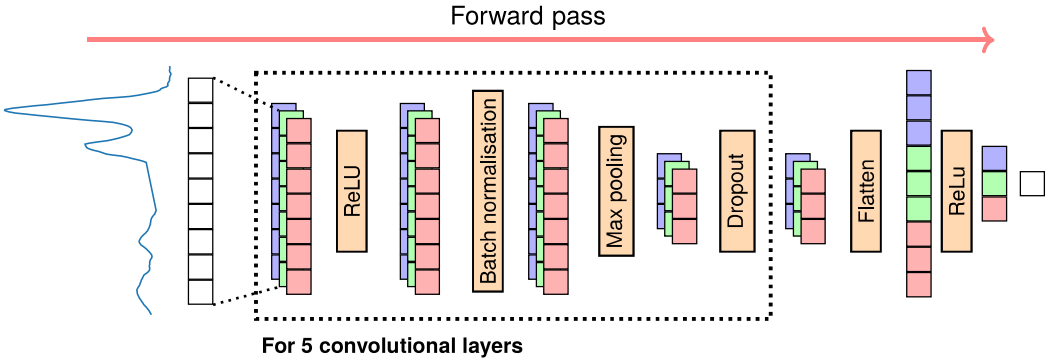

FIGURE 1.3: A simplified schematic of the optimal one-dimensional CNN architecture. The shape of the data as it passes through each layer is represented by vectors; the colour of each vector represents a different kernel. Intermediate layer activations, regularisation steps etc are represented by orange boxes. The final element represents the probability of a poor prognosis for that spectrum.

Multilayer perceptron network

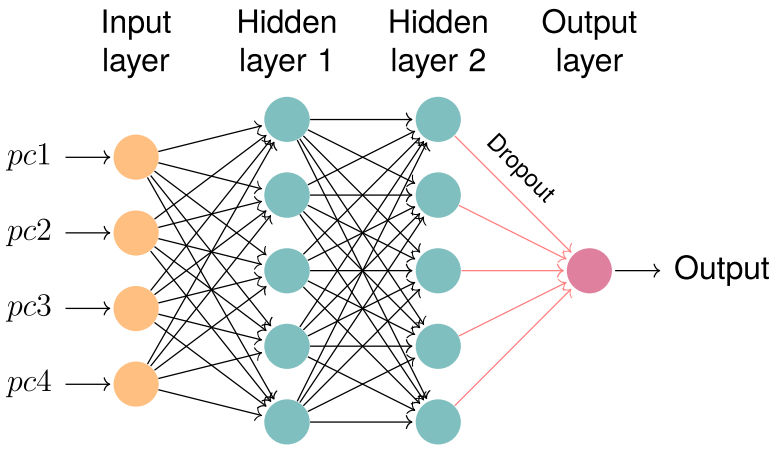

FIGURE 1.4: A multilayer perceptron neural network with an input layer consisting of four input variables $x_0 \dots x_4$, three hidden layers of five nodes each, and a single output layer.

TABLE 1.2: Optimal MLP network parameters.

Parameter name	Value
N hidden layers	2
N nodes in hidden layer 1	169
N nodes in hidden layer 2	10
Dropout layer 1 probability	0.28
Dropout layer 2 probability	0.29
Learning rate	2×10^{-5}
Weight decay	1.1×10^{-3}
Optimum value	0.77

1.3 Results

As dicussed in detail in **??**, the objective was to predict which risk group a patient falls into, risk groups were determined by a GA optimisation routine seeking to achieve the maximum prognostic information. Predictions of risk groups for each patient were taken as the median probability predicted across all spectra for any given patient. The threshold used to dichotomise probabilistic predictions was set by maximising the MCC score. The MCC score considers all possible prediction outcomes and is a well-rounded measure of performance for discrete predictions.