Development Individual Project: Presentation
Developing a Neural Network with the CIFAR-10 image dataset
Written Audio Transcript, References, Appendix

By William Bolton
24.05.2024
Word count: 2443 (Aim for word count range: 2000-2500)

Please hit 'Slide Show' and 'Play from the Start' to listen and watch the presentation.

Slide 1: Title slide.

Hello, my name is Will Bolton, and I am a postgraduate certificate in artificial intelligence student at the University of Essex online. My presentation today is about an analysis of The CIFAR-10 dataset (Canadian Institute For Advanced Research), comprising of 60,000 coloured images, distributed across 10 object categories with 6,000 images for each category. The structure of my presentation will cover a brief introduction and overview of artificial intelligence (or AI) and object recognition, and then moving on to discuss neural networks and their architectures, with description of an example of methods and a model used for this dataset in his assignment.

Slide 2: AI and Object Recognition

AI represents a transformative force across numerous global industries, fundamentally reshaping how we interact with technology. At its core, AI involves creating systems that can perform tasks which typically require human intelligence. Among these tasks, object recognition stands out as a critical function, empowering systems to perceive and interpret the world around them just as humans do.

Cutting-edge applications include:

Autonomous Vehicles: AI enables these vehicles to 'see' and navigate safely by identifying obstacles, road signs, and pedestrians.
Medical imaging and diagnosis: AI can aid recognition and diagnosis of diseases or abnormalities in medical imaging and assist the human doctor in treating patients faster and more accurately.
Inventory management: AI can help human workers using digital devices to scan shelves, which automatically identify and highlight various products, helping automate and accelerate inventory management.

Slide 3: Neural Networks: A backbone of AI

Neural networks involve collections of algorithms designed to identify relationships in data through a process that is similar to the function of the human brain. They are often called artificial neural networks or ANNs.

Neural networks consist of layers of interconnected nodes or 'neurons', called perceptrons, each capable of performing simple calculations. These consists of an input layer, numerous hidden layers, and an output layer, and the connections between these layers have 'weights' that adjust as the network learns from data, which is fundamentally how it improves and makes increasingly accurate predictions over time.

Other key components of neural network architectures include the transfer function which allows an activation function to be applied. The activation function permits non-linearity in the working to the perceptrons which is essential for the functioning of the network. Finally, a loss function is required to guide the optimisation of the process and help the network learn and improve over time.

Slide 4: Architecture of the Artificial Neural Network

The perceptron is the simplest fundamental unit of neural network architectures. They are combined in different ways to form bigger architectures that have different strengths and weaknesses.

Because the CIFAR-10 image dataset requires an architecture that can handle image classification problems efficiently and simply, a convolutional neural network (CNN) was chosen. A CNN is a type of feed-forward neural network that has hidden convolutional layers that aim to detect patterns in image data via feature extraction.

I developed a standard feed-forward convolutional neural network for this task with the following architectures. The hidden convolutional layers applied convolutional operations using the ReLU or Rectified Linear Unit activation function to extract features from the input images and used a sparse categorical cross-entropy loss function. Pooling layers were used to reduce the spatial dimensions of the feature maps and aided in the reduction of computational complexity. They also help reduce the risk of over fitting. I had a flatten layer that converts two-dimensional feature maps into a one-dimensional vector. These layers perform the final classification based on the extracted features from the convolutional layers.

Strengths of this architecture include its simplicity, being easy to implement and computationally efficient. This makes it suitable for image classification tasks where detailed spatial information is less critical and computational resource is limited. The model is also versatile, meaning it can be adapted with extra layers or units as required.

Main disadvantages include limited spatial context in comparison to more advanced architectures. There is also an increased risk of overfitting with small numbers of layers, especially when applied to more complex datasets.

Alternative more complex architectures were considered including ResNet or residual networks and U-net models. Each architecture has its unique strengths including deeper networks and denser connections, but they generally require more resources and tuning. For simplicity and efficiency, a standard CNN was the architecture taken forward for this project.

Slide 5: Imports, setup, and preparing the dataset

The CIFAR-10 dataset is one of the most widely used datasets for machine learning research and was published in 2009. Before the model was developed using the dataset, a number of preprocessing steps had to be taken first.

The programming language used for this task was Python. I then imported a number of packages and libraries including tesnorflow, numpy and matplotlib. I then loaded the CIFAR-10 dataset into batches and reshaped them into 4D arrays and transposed them to ensure they were compatible with tensorflow. Along with this, labels were converted into numpy array format to also ensure compatibility with tensorflow.

Next, normalisation of pixel values was performed. Normalisation is the process of scaling the values of data to a standard range, typically between 0 and 1 or -1 and 1. This is a common preprocessing step in machine learning and deep learning. We normalise image data for a number of reasons, including stabilising the training process as neural networks perform better and more efficiently. As images are typically represented as arrays of pixel values, each ranging from 0 to 255, dividing each pixel value by 255 scales these values to a range between 0 and 1.

Slide 6: Partitioning the validation set from training data and its importance

There are three datasets used in this model development exercise: The training set; the validation set; the testing set. An essential element to partitioning data into these sets is that there is representative splitting, ensuring each subset contains a variety of images to try and reduce bias in model training and evaluation. The training set is a large portion of the total data from images in the CIFAR-10 dataset. Diversity of the data is absolutely crucial for training models in this context. I used this for training the neural network and the model was taught using this data to recognise and classify the different objects of interest. A training subset in this way allows the model to learn pattens and features that help with object recognition and classification.

A validation subset is separate from the training data and is used to fine tune the model’s hyperparameters including checks against over or underfitting. Hyperparameters are parameters that need setting before the learning process begins and include the learning rate or the number of layers in a neural network, for example. In this way it is used to evaluate the model and allows the AI engineer to understand how well the model can generalise to new data and make the necessary adjustments to optimise the model accordingly.

The testing set is the final subset and consists of brand-new data that the model has not seen during training or validation. The purpose of this subset is to evaluate the final model’s performance on entirely new data and confirms how well the model will perform in future real-world scenarios.

The method I used for partitioning was the representative split function to define the training validation sets and also printed the shapes to confirm the split. The proportions I used 75% of the data for training samples (45,000 samples), 8.3% of the data for validation samples (5,000 samples), leaving 16.6% of the data test samples (10,000). These proportions ensure that the dataset is appropriately divided for training, validation, and testing purposes.

I then visualised the first 5 images from training set ensured that data had been processed, partitioned, and loaded correctly.

Slide 7: Activation and Loss Function

An activation function is a mathematical function used in artificial neural networks to introduce non-linearity into the model. This non-linearity is crucial because it allows the network to learn from and represent complex patterns or relationship in the data. In this task, I used the ReLU for my activation function within the hidden layers of the model. This was chosen because of its computational efficiency as it involves simple functions. It also mitigates the vanishing gradient problem during backpropagation. Disadvantages of this activation function include the dying ReLU problem which is more of a problem if the learning rate is too high.

I used softmax in the output layer of the model to output a probability distribution over the 10 classes within CIFAR-10. This meant that outputs could be interpreted as probabilities which is useful for classification tasks and ideal for multi-class classification problems. Downsides of this is that it is relatively more computationally intensive.

Other activation functions available include the sigmoid function, which is useful for binary classification and output layers, but is more prone to vanishing gradients in deep networks.

A loss function is a mathematical function that quantifies the difference between the predicted values produced by the machine learning model and the actual target values from the training data. Its purpose is to provide a measure of how well or poorly the model is performing. During the training process the model aims to minimise this loss by adjusting's parameters like weights and biases through optimisation algorithms.

The loss function chosen in this exercise is the sparse categorical cross-entropy function that is appropriate for multi-classification problems where the target labels are integers, as is the case in this dataset. Strengths of this choice include the simplicity and efficiency. Weaknesses include a lack of interpretability as the values themselves are not always intuitive.

Other loss functions include mean squared error or MSE which is useful in regression problems, and therefore not suitable for classification problems as it doesn't handle probabilities and categorical outcomes well.

Slide 8: Epochs in the modelling process

The number of epochs when developing a neural network model is chosen based on experimentation. In the development of the model for this presentation, I initially chose 20 epochs. The performance here is summarised by the graph on the left of the slide. When assessing the model's performance from epochs 11-20, the model began to overfit, indicated by the increasing validation loss and fluctuating validation accuracy. Training accuracy did continue to improve, but validation accuracy stopped showing significant improvement. Main problems with too many epochs also include increased training time and computational resources.

To address overfitting initially, I added used techniques like regularisation and data augmentation, but these resulted in reduced accuracy and over-complicated models. With these extra elements, the model was potentially too complex for the data, where it appeared to memorise the training data but not generalise well to unseen data. I removed these and went back to the simpler architecture presented in the next slide but reduced the number of epochs to 12 in the final model and produced a good accuracy with less overfitting. The performance is summarised in the graph on the right of the slide. Ultimately the choice of number of epochs needs to provide a good balance between learning while avoiding excessive overfitting.

Slide 9: Neural Network's Design Strategy

The specific CNN model I used in this project was defined as a simple architecture designed to extract features through convolutional layers followed by pooling layers to reduce dimensionality. I chose the number of convolutional and pooling layers to try and strike the balance between complexity captured and performance versus the computational cost and risk of overfitting. More layers can improve performance for complex data, permitting deeper representations, but it was felt to be unnecessary for the complexity of dataset in this project.

The model compiler was chosen for its adaptive learning rate and ability to handle multi-class classification tasks with integer labels as I previously mentioned. The model was trained for 12 epochs as following experimentation this was allowed improved accuracy with reduced effects of overfitting. The training data was used to update the model weights and the validation data was used to monitor performance and reduce overfitting.

Slide 10: Reflecting and conclusions on learnings acquired

I evaluated the model using a test set and the metrics I used to assess the model performance were test accuracy, precision, recall and F1 score. As you can see, the model correctly classified over 70% of the samples in the test set. Test precision aims to answer the question: Of all instances that were predicted as a certain class, how many were actually of that class? This model was correct in over 71% of cases. Recall is the sensitivity of the model and was also above 70%. Finally, the F1 score is the weighted average of precision and recall that provides a single metric that aims to balance both measures. The F1 score of above 70% indicates that the model has a balanced performance.

Critically analysing the results, the model produced is balanced and provides a good level of accuracy that is comparable to similar machine learning experiments with this dataset. Key limitations of the model include its simplicity and effects of overfitting. The model could be improved by more advanced model architectures, more comprehensive evaluation methods. However, given the complexity of the data and computational resources at my disposal, I am satisfied with the model developed and the results I was able to generate.

Reflecting on the new knowledge I have gained throughout this project; I am inspired by how far I have come. I am a total novice to data science, machine learning and programming, and therefore I am very proud to see what I have been able to achieve. I am grateful to my course tutors and peers who have helped and inspired me throughout. I now feel more confident to analyse an unmet need that an AI system could help with, design the appropriate model based on the needs and available data, with an understanding of everything from preparing the dataset, building, testing training, and evaluating the machine learning model such as artificial neural networks. I now look forward to applying this newfound knowledge and confidence to other object recognition projects and real-world problems, particularly in my background area of healthcare.

Slide 11: References

Here are my references.

Slide 12: End slide

Thank you very much for your attention, this concludes the presentation.

References
(Not dictated verbatim)

CNN Explainer: Learn Convolutional Neural Network Webpage by Wang, J; Turko: R; Shaikh, O; Park, H; Das, N; Hohman, F; Kahng, M; Chau, P. Online at: https://poloclub.github.io/cnn-explainer/ Accessed 24.05.2024.

Kubat, M., 2017. An introduction to machine learning. Springer.

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

File: Example of a deep neural network.png, From Wikimedia Commons, the free media repository. Online at: https://commons.wikimedia.org/wiki/File:Example_of_a_deep_neural_network.png Accessed 24.05.2024.

The Essential Guide to Neural Network Architectures, by Pragati Baheti (Microsoft), V7. Online at: https://www.v7labs.com/blog/neural-network-architectures-guide Accessed 24.05.2024.

Techniques for training large neural networks, OpenAI. Online at: https://openai.com/index/techniques-for-training-large-neural-networks/ Accessed 24.05.2024.

Will Cukierski. (2013). CIFAR-10 - Object Recognition in Images. Kaggle. https://kaggle.com/competitions/cifar-10