Presentation Transcript

Welcome to our presentation. In the presentation, we are going to discuss Neural Network Models for Object Recognition.

Object recognition has already been a research topic for years. Object recognition or object detection is part of the research in computer vision, deep learning area, or in artificial intelligence (Deng et al, 2020). There is multiples method used for object recognition, one of them is Artificial Neural Networks (ANN). ANN imitates the neuron system in the human brain, which is connected to around 86 million neuron networks in the human brain (Herculano-Houzel, 2012). Similar to the human brain, each node is connected with the other node, then the activation function calculates the probability based on the weight and input value to find the closest decision (Géron, 2019).

The presentation will discuss how machine learning can identify the object. The method selected in this presentation is using Artificial Neural Networks (ANN). Artificial neural networks are divided into three main parts, input layer, multiple hidden layers, and output layers (Dertat, 2017). Input layer act as the recipient of the object and it has value. Hidden layer act as the activation function to calculate based on the input value multiplied by the weight of each node. The output layer is the layer where it has the value resulting from the activation function.

The presentation will explain in detail what is ANN and how it works on object recognition. The experiment is using CIFAR-10 data, a collection of 60,000 images with the size of 32x32 classified into 10 classifications, where each has 6,000 images (Krizhevsky, 2017). The Keras library has its own library to load the dataset of CIFAR-10 directly. The dataset is divided into train data and test data. Train data will consist of 50,000 images, and test data will consist of 10,000 images. Train results will be evaluated against the test data, to get the accuracy of the model.

Before we explain the ANN, first we will discuss how the neuron works. Neurons consist of dendrites, cell nuclei, body cells, axons, and synapses. In the image, the orange part near the cell nucleus is the body cell. The input signal is received by the dendrites. Dendrites may receive the input signal from the previous neuron system. Body cells will process all input information and transfer it thru the axon. The synapse is the link between one neuron to another neuron. In the synapse, the processing output will be transferred as the result. As explained earlier, the neuron systems in the human brain comprise 86 millions neuron. The beauty of the human brain, in these large numbers of neuron networks, the information can be processed and transferred in a very short time (Herculano-Houzel, 2012).

Imitating the neuron systems in the human brain, researchers found a method called Artificial Neural Networks. How it works is very similar to what the human brain does. Input information received by the nodes. The input is transferred into one or multiple hidden layers. As seen in the image, three inputs passed into the nodes and transferred into four nodes in the hidden layer. Each node has weight and is calculated in the activation function. The activation function is the sum of the multiplication of the input value, multiply by the weight plus the bias.

The activation function for a single node can be represented in a linear equation:

where is the input, is the weight, while is the bias, and is the output (Nadeem, 2022).

The activation function or known as the transfer function is used to process the input signal into the output signal of the node (Sharma, 2022). There are two types of activation functions, they are linear activation functions and non-linear activation functions. A linear function has the linearity of the result depending on the value of the input. It could range from minus infinite to plus infinite. Non-linearity function, easier to adapt with the input condition. The non-linear activation function is divided into the sigmoid function, tanh function, rectified linear unit (Relu), and Leaky Relu.

As already mentioned earlier, the non-Linear Activation function has a few types such as Sigmoid, Tanh, Rectified Linear Unit (Relu), or Leaky Relu. The sigmoid function has ranged between 0 and 1. The sigmoid function has the capability to predict the probability where if the value is close to 1, the result is most likely true, and if the value result close to 0, the result is most likely false. One of the most used sigmoid functions is softmax, used to classify multiple classes (Sharma, 2022). Unlike the sigmoid function, the Tanh function or called as hyperbolic tangent activation function has a range between -1 and 1. Similar to the sigmoid function, both Tanh and sigmoid is s-shaped function. Tanh activation function is good for comparison between two classifications (Sharma, 2022).

Nowadays, Rectify Linear Unit (Relu) become the most popular activation function used in the world. The range of this activation function is between 0 and infinite. Any negative value in the Relu is changed to 0 immediately. If the function is above zero, then the result is also above zero. The downfall of this method is for any minus result, is changed to zero immediately, which reduces the ability to fit the result during the training. This downfall was tried to fix by the Leaky Relu activation function method. There is an additional constants value “a”, where normally the “a” value is 0.01. It is increasing the range to fit the result during training. The range result will be minus infinite and plus infinite (Sharma, 2022).

The loss function is used to see how well the predicted result is against the target value (Yathish, 2022). There are a few types of the loss functions, such as cross-entropy, binary cross-entropy, categorical cross-entropy, sparse categorical cross-entropy, or mean squared error.

One of the most popular lost functions is Mean Squared Error (Seb, 2022). Mean Squared Error only allows predicted and expected values to be real numbers (Yathish, 2022). The total difference between the predicted and the target value is squared. The total differences are calculated as the average of the total error cost (Seb, 2022).

There are two types of Artificial Neural Networks. Feedforward ANN and feedback ANN. Feed-forward ANN only transfers the information into nodes and passes it as the output. Unlike feed-forward ANN, feedback ANN allows feedback from the output back to the initial nodes (Ali, 2019). Feedforward ANN, only processes in one direction. And it is the simplest form of neural network.

Now my fellow student Bharadwaj will explain how a neural network is implemented.

Thank you, Indra.

CIFAR 10 is a dataset that primarily consists of images of 10 different categories/classes that are used in building machine learning models. There are classes like airplanes, horses, cats, deer, and so on. The dataset comprises 60,000 images using which we performed machine learning techniques like ANN, and CNN, respectively. Further in this presentation, we shall see how we have used the data to train, validate and test the data on ANN and CNN.

The first and foremost step is to import all the required python libraries and the CIFAR 10 dataset into our python Integrated Development Environment (IDE). The image shows the same. The predominant library among all is Keras and its subclasses. Once we have imported all the requirements, we move on to the next step, which is, data preprocessing.

After loading the dataset, we begin the data preprocessing step, firstly, by splitting the dataset into train and test sets, respectively. From the image, the shape of the train set implies that the training set has 50,000 images of 32 heights by 32 widths with a 3-color scheme. The target variable y has 50,000 records containing the names of the classes. The test samples are shown as 10,000 which will be used for testing our neural network model.

Furthermore, we check the distribution of each class in both train and test sets so as to avoid any imbalances in the classes. We can deduce from the graph that there is an equal amount of data in the train and test set for all 10 classes. In the coming slides, we are going to perform more data preprocessing tasks and eventually get to the model building.

The next step in our process is to normalize the data and start building the neural net. The purpose of normalization is to ensure the similarity of data across all records. This is done by converting the train and test samples into float data types and then dividing by 255 (owing to the grayscale 0-255) to make the model building even simpler. The target variable is also converted into a vector for both train and test samples. This process is called one hot encoding.

We now start building the neural net. Our neural network consists of 9 layers out of which 2 are convolution layers, 2 are pooling layers, one flattening layer, and 4 are dense layers, thereby completing the neural network.

A convolution layer is the building block of a neural network. It contains a set of filters and parameters that are to be learned to gauge the neural network’s performance. The filters are smaller in size than the image, which convolves with the image and produces an activation map. Then by iteration of the parameters, the activation map gets precise.

A max-pooling layer is similar to a convolution layer; instead, the layer does not do a product between the image and the filter used but determines the maximum value in the region overlapped by the kernel on the image (Patel, 2019). The layer then strides across the whole image by doing so and generates an activation map. This layer helps in reducing the overfitting of the data.

After obtaining the pooled layer, we then flatten the activation map by passing it through the flattened layer of the network. The flattened layer converts the feature map, which is basically a matrix into a vector. This ensures faster processing of the data. (SuperDataScience, 2018).

Finally, we pass out outputs from all previous layers to the dense layers, which are used to classify the outputs from previous layers. It uses a nonlinear function called the activation function and classifies the output. The result of the dense layer is generally considered as the final output of the neural network (Dumane, 2020).

This is what the summary of the neural network looks like. It gives us brisk information about the layers used, and the number of trainable and nontrainable parameters.

After building the neural net, we move on to compile the neural network using certain input parameters. Optimizers are algorithms used to monitor the network's weights and other parameters to reduce the loss incurred in the model (Kumar, 2020). Once the required parameters are set, we move on to training the neural net with those input parameters and of course, our data.

We now train our neural network using the training set of the CIFAR10 data and observe how it does “model.fit” is the command generally used to train any neural network with the input parameters as shown in the image.

This is how the neural network performs on the training set of the CIFAR10 data. The model's accuracy gradually increases while the loss gradually decreases. The key thing in the compilation is the val\_accuracy parameter which is about 0.6799 ~ 68% when rounded off.

The graphs show how has the model progressed over each iteration. The left graph is the model’s accuracy. With each increasing iteration, the training accuracy ( blue line) has steadily risen. The validation accuracy has however peaked after the 5th epoch, which is about 68%, as seen earlier. The test scores are also like the validation accuracy which, in our case, is good. The loss graph shows how the loss has decreased with each iteration. The validation loss has not been consistent, but a decreasing trend could be observed. This happened because we had limited the number of iterations for reasons like avoiding overfitting, saving time, etc.

This is how we generate the metrics for each class in the CIFAR10 data. The classification\_report command is used to generate it. It gives metrics that are calculated using the confusion matrix like precision, recall, etc. Without the hassle of calculation the matrix would be of 10x10 size. By observing the table, class 1 has the highest scores of all.

We have verified whether our model has made correct predictions for the testing data or not These images here show the greatest number of correct predictions (6/10) like the cat, frog, the airplane, and the ship. The subsequent slides show the wrong and extremely wrong predictions made by the neural net.

This slide highlights the wrong predictions made by the neural net. In these predictions, the model was able to recognize the image partially but, in the end, the classification was made wrongly. For instance, in the horse-cat classification, the model recognized that the image had 4 legs, just like horses and cats do, but made the wrong prediction as a cat when the true image was of a horse. The next slide highlights the most wrong predictions made by the neural net.

This is where the neural net has failed to recognize an image correctly and classify it. This could have happened because the number of layers in our network was less which gives no room for the model to analyze all images correctly, train the model on the data and classify the images. Also, poor labeling of input images must have impacted the model's prediction capability (Franky, 2021).

We have now come to the end of our object recognition presentation. The key takeaways here are how a naturally occurring human brain and a similar design which is a neural network function, with the help of input parameters, and how, by updating the parameters, one can make the neural network perform more efficiently. The Artificial Intelligence module, as a whole was very rich in knowledge, thanks to which, we were able to accomplish this task of creating a neural network and talking about it. The fascinating part is that the human brain is much faster than an artificially designed neural network. But we are inching close to matching the speed of the brain. Thank you to everyone who has helped us gain knowledge regarding this powerful and mesmerizing concept. We sure look forward to applying this knowledge, whenever and wherever possible. Thank you.