Fundamentals

Deep learning  
 model  
 training

Before the proposed smart farming system is explained in detail, some basic terms and concept used in this paper should be first comprehend so that the step used in the system’s development can be clearly understood. Three main topics will be explained in this section namely smart farming, deep learning, neural network in deep learning.

**Smart Farming**

The present issues facing agriculture cannot be resolved by using traditional methods to fertilize or harvest land. As a result, strategies for digitizing and automating agriculture—also referred to as "Smart Farming"—that are based on cutting-edge ICT are needed \cite{smartfarm}. Agriculture can be simultaneously made more resilient, efficient, and sustainable thanks to smart farming. Smart agricultural technology investments are appealing due of lower sensor technology prices. Because of this, there is presently an oversupply of services from various producers on the market and they are incorporated into the Internet of Things for the Farm to guarantee the interoperability of these services (Farm IoT) \cite{smartfarm}.

**Deep learning**

Deep Learning or known as deep structured learning \cite{deng\_overview\_2011}, which are modeled after the structure and operation of the brain, are a subfield of machine learning approaches that deal with algorithms that use numerous layers to gradually extract higher-level properties from the raw input. Another frequently mentioned advantage of deep learning models, in addition to scalability, is their capacity for automatic feature extraction from unprocessed data, also known as feature learning \cite{advance}.

**Neural network / model**

Neural network is created to make prediction base on given input where neuron is the basic building block. The neuron, depict in ``Fig. \ref{single\_input}”, can have several input and output just like human neuron except this neuron is a set of mathematical function \cite{design}. The multiple input can be visualize as shown in ``Fig. \ref{multiple\_input}”.

\begin{figure}[ht]

\centering

\includegraphics[width=0.2\textwidth]{single\_input}

\caption{Neuron with single input \cite{design}}

\label{single\_input}

\end{figure}

\begin{figure}[ht]

\centering

\includegraphics[width=0.2\textwidth]{multiple\_input}

\caption{Neuron with multiple input \cite{design}}

\label{multiple\_input}

\end{figure}

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

Every neuron has their own transfer function or activation function to satisfy some specification of the problem that the neuron is attempting to solve \cite{design}. ``Fig. \ref{tf\_linear}”shows the example of transfer function.

\begin{figure}[ht]

\centering

\includegraphics[width=0.2\textwidth]{tf\_linear}

\caption{Linear transfer function \cite{design}}

\label{tf\_linear}

\end{figure}

Chart

Description automatically generated

Linear transfer function is used when the output should be linearly proportional to its input. Neurons with this transfer function are used in the ADALINE networks \cite{design}.

Another transfer function is log sigmoid, which constricts the output to the range of 0 to 1 from the input that can have any value between plus and minus infinity and hard limit transfer function, that changes the neuron's output from 1 to 0 depending on whether the function parameter is greater than or equal to 0. This process produces neurons that categorize inputs into two different groups. \cite{design}.

As mentioned before, the neuron is basic building block of neural network. This building block can be grouped into the form called layer as depict in ``Fig. \ref{layer}”. Every layer has their own functionalities and name. For instance, convolutional layer where the neuron acts as a kernel or filter, pooling layer, dropout layer and fully connected layers. These specific layers will be discussed more in proposed method section.

\begin{figure}[ht]

\centering

\includegraphics[width=0.2\textwidth]{layer}

\caption{A layer in neural network\cite{design}}

\label{layer}

\end{figure}

Diagram, engineering drawing

Description automatically generated

The specific layers are then arranged in sequential way to form a neural network or network model. For example, Convolutional neural network (CNN) is widely used in neural network for image recognition that consist at least one layer of convolution layer \cite{practical}.

Aforesaid, neuron in computer science is nothing more than a set of mathematical function that have input and output. That means a layer of neuron have a set of input and output as shown in ``Fig. \ref{layer}”. These variables are handled in matrix form where input matrix will be multiplied with weight and matrix of activation function and added with bias before supplied to the activation function. The output could be scalar or vector depending on the input matrix and activation matrix. Since the input and output could be vector that have more than two dimensions, they are called tensor of input or output \cite{practical}.

Just like human, before the model could make any decision, it should learn how to make the right decision. This phase is called training phase after the neural network, or the model is completely built. This phase will be detailed in proposed method section.

**Training**

There are two main process in training phase which is training and inference. Aforesaid that input from every neuron is multiplied by its weight and added with its bias to be an input to the activation function. In training phase, these parameters are adjusted so that the final output is same as expected value.

Before further discuss on training phase, the terms and data used for the training phase should be first comprehend. This data can be image, text, sound, time series and video and called dataset \cite{ advance }. The data set will be split into two set, one for training process and another one for inference process.

The dataset used in this project and how it is distributed will be explained more detail in section \ref{trainig}

To handle the huge amount of input, the images are split into batches. The batch size is the number of samples that are passed to the network at once \cite{practical}. This is to increase the speed of training because the number of errors that need to be stored is the number of the batch size instead of the error for each images \cite{ practical }.

To increase the accuracy of the model, the data set will be passthrough the model a few times. This iteration is called epoch \cite{epoch}.

Training a network means nothing more than solving a complex optimization problem \cite{advance}. At first, the value of weight and bias for every neuron is randomly assigned. After that, an image is given to the model as an input. The final output value is compared with an expected value, where the difference value or known as loss value is recorded. The loss values are used by loss function to improve the value of the weight in the way that the loss value can be reduced in the future. The lost function that is used in proposed method is categorical cross entropy. In training process, it could happen that the model is “overfitting”. Means that the model has high accuracy on trained data but low accuracy in real environment. To reduce the overfitting, the learning rate is reduced. The algorithm used to achieve that is called optimizer. For the proposed model, the optimizer used is Adam. The detail about Adam optimization can be found at \cite{intelligent}.

The trained model then proceed to the inference process where the model accuracy is evaluated, where the remaining data set is used.

**Development Environment**

In this project, the system was develop using Jupyter notebook environment where python kernel is used. To easy the development, fast ai is used because it have much simpler API to build a neural network model and to train them.