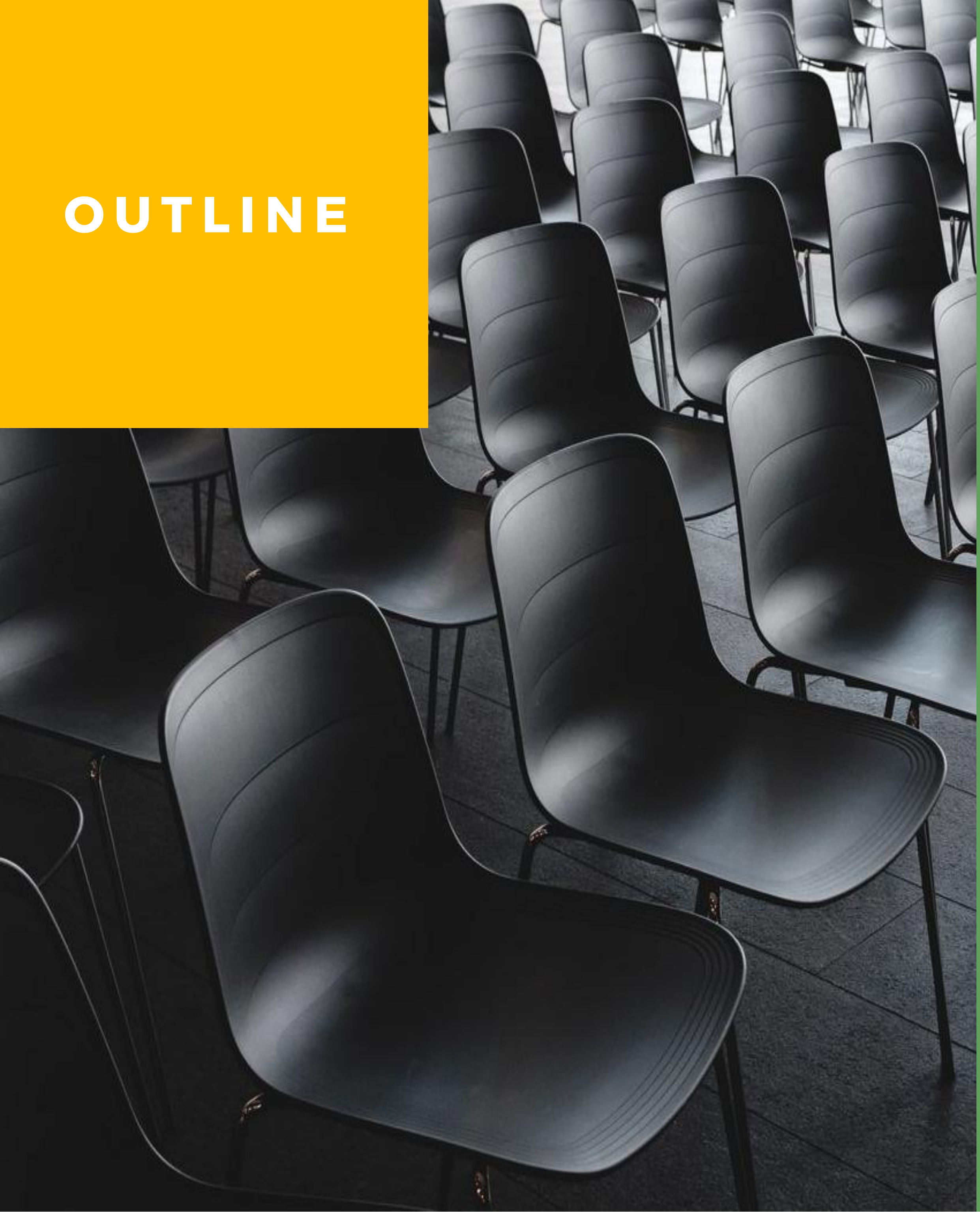
RESPONSIBLE ARTIFICIAL INTELLIGENCE LAB (RAIL)

MLP: Neural Networks and Deep Learning

7th December 2022

ARTIFICIAL INTELLIGENCE FOR DEVELOPMENT AFRICA

RESPONSIBLE ARTIFICIAL INTELLIGENCE LAB



OUTLINE

1. Basic principles of deep learning
2. How to train deep neural networks (DNN) using Tensorflow
3. How to train convolutional neural networks (CNN) using Tensorflow
4. How to use transfer learning to train a CNN Tensorflow

Introduction

- Deep learning is an advanced form of machine learning that tries to emulate how the human brain learns.

- In your brain, you have nerve cells called neurons, connected by nerve extensions that pass electrochemical signals through the network.

Introduction

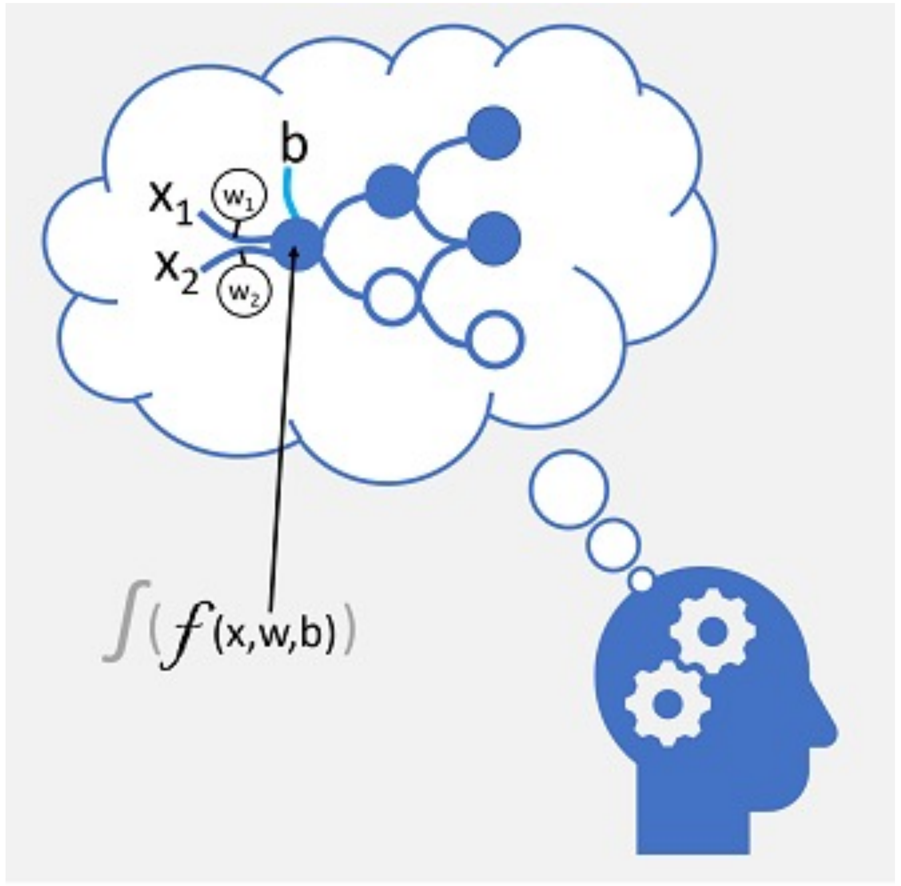
- When the first neuron in the network is stimulated, the input signal is processed.
- If it exceeds a particular threshold, the neuron is activated and passes the signal on to the neurons to which it is connected.
- Over time, the connections between the neurons are strengthened by frequent use as you learn how to respond effectively.
- Deep learning emulates this biological process using artificial neural networks that process numeric inputs rather than electrochemical stimuli.

Introduction

- The incoming nerve connections are replaced by numeric inputs that are typically identified as **x**.

- When there's more than one input value, x is considered a vector with elements named **x_1**, **x_2**, etc.

- Associated with each **x** value is a weight (**w**), which is used to strengthen or weaken the effect of the **x** value to simulate learning.

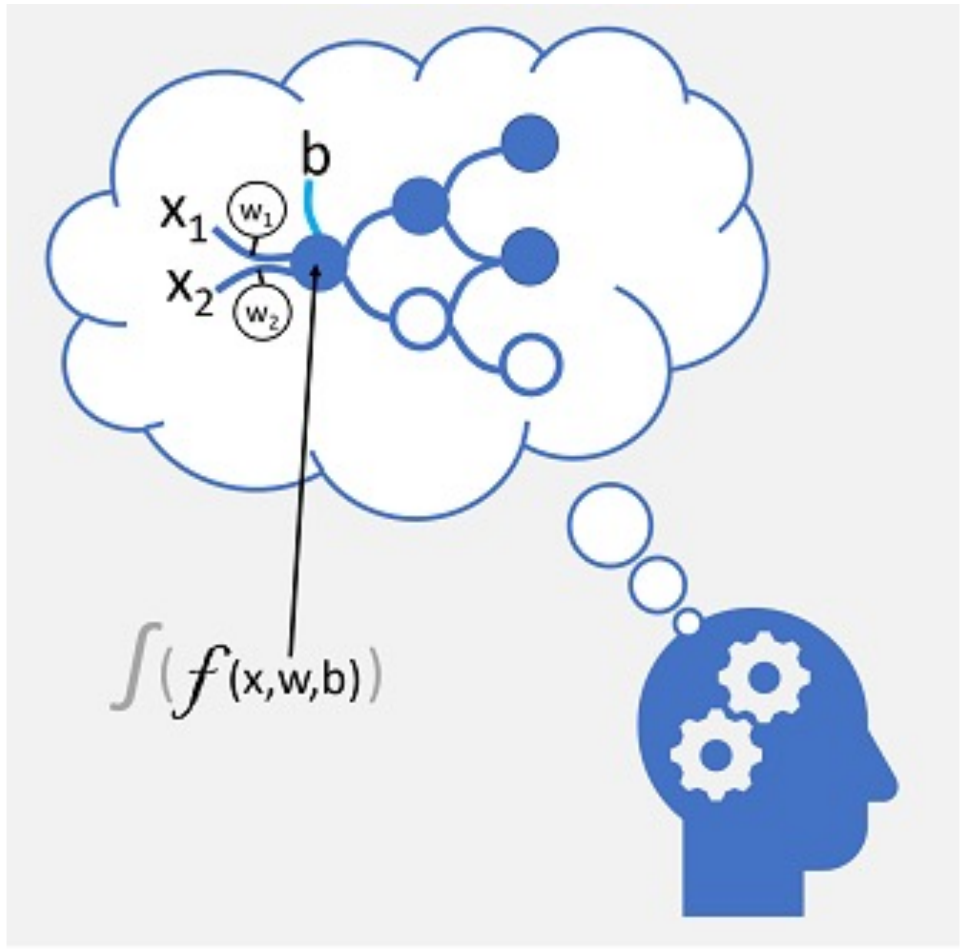


Introduction

- Additionally, a bias (**b**) input is added to enable fine-grained control over the network.

- The neuron itself encapsulates a function that calculates a weighted sum of **x**, **w**, and **b**.

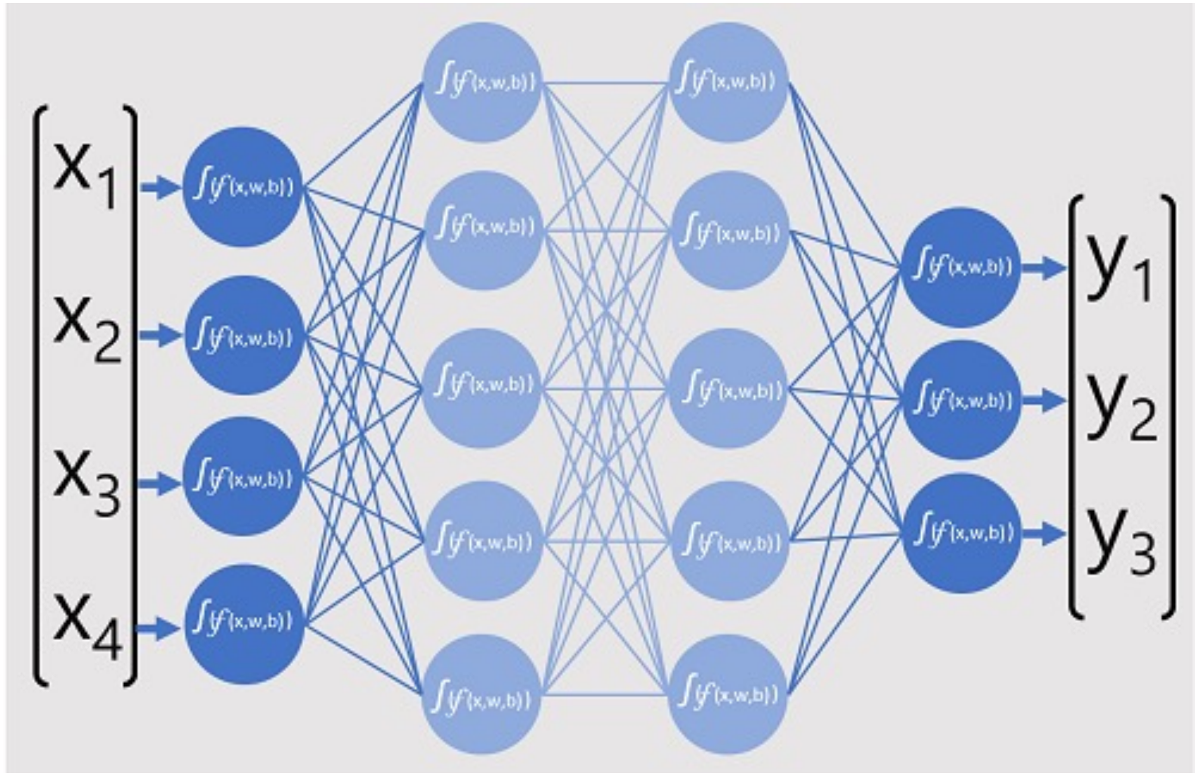
- This function is in turn enclosed in an activation function that constrains the result (often to a value between 0 and 1)
  - to determine whether or not the neuron passes an output onto the next layer of neurons in the network.



Deep Neural Network Model

- The deep neural network model for a classifier consists of multiple layers of artificial neurons.

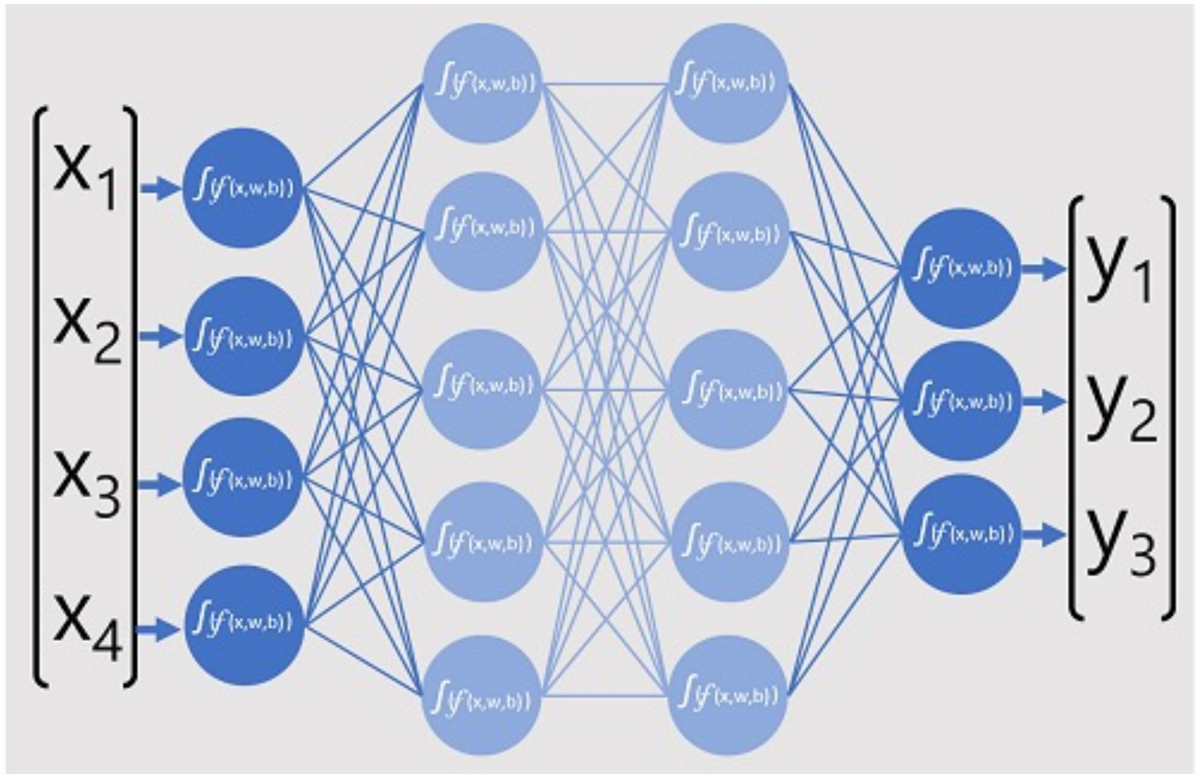
- In this case, there are four layers:
    - An **input layer** with a neuron for each expected input (x) value.
    - Two **hidden layers**, each containing five neurons.
    - An **output layer** containing three neurons - one for each class probability (y) value to be predicted by the model.



Deep Neural Network Model

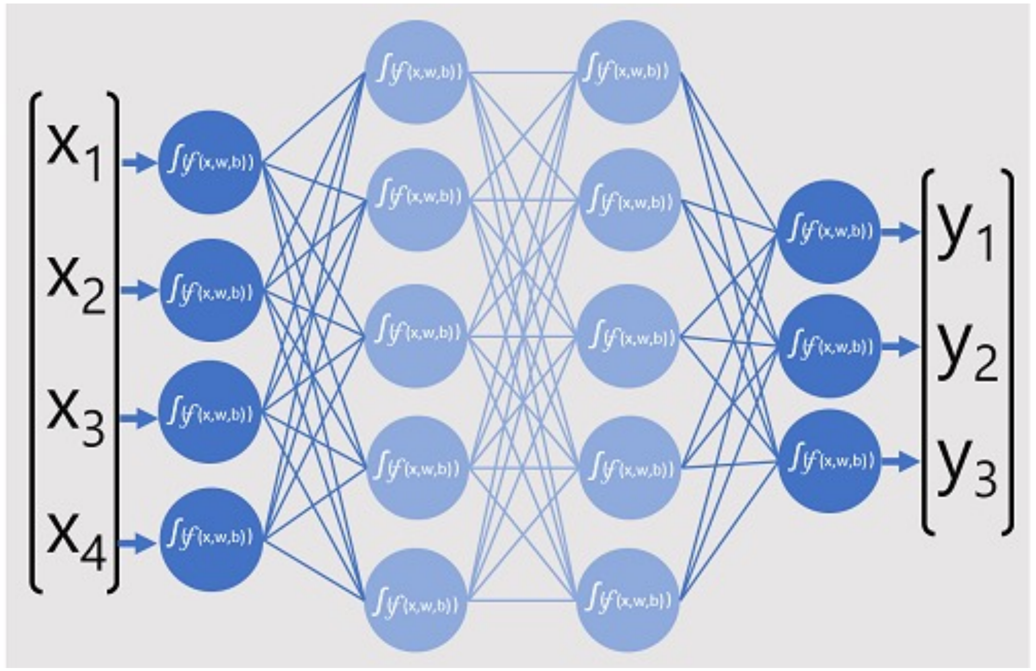
- Because of the layered architecture of the network, this kind of model is sometimes referred to as a multilayer perceptron.

- Additionally, notice that all neurons in the input and hidden layers are connected to all neurons in the subsequent layers - this is an example of a fully connected network.



Deep Neural Network Model

- When you create a model like this, you must define an
  - input layer that supports the number of features your model will process
  - output layer that reflects the number of outputs you expect it to produce.
- You can decide how many hidden layers you want to include and how many neurons are in each of them;

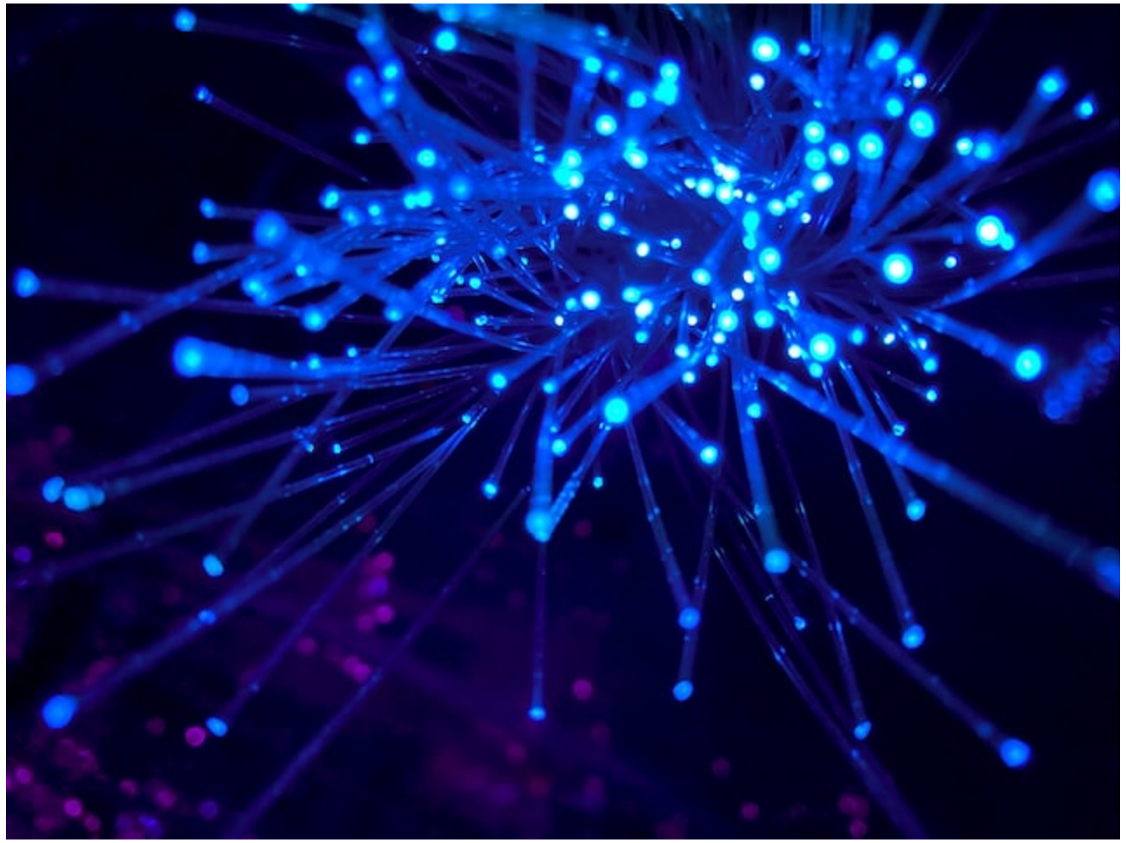


- you have no control over the input and output values for these layers - these are determined by the model training process.

Training a Deep Neural Network

- The training process for a deep neural network consists of multiple iterations called epochs.

- For the first epoch, you start by assigning random initialisation values for the weight (w) and bias b values.



Training a Deep Neural Network

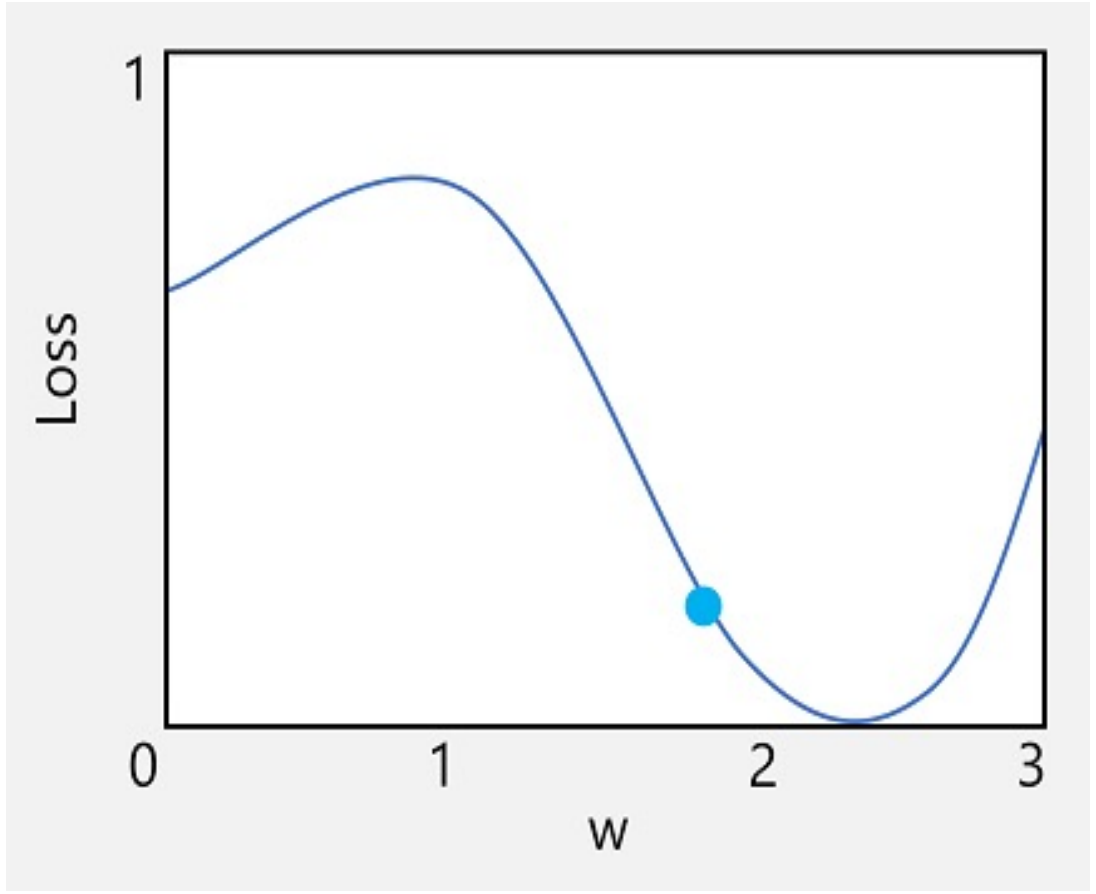
- Then the process is as follows:
  - Features for data observations with known label values are submitted to the input layer.
    - Generally, these observations are grouped into batches (often referred to as mini-batches).
  - The neurons then apply their function and, if activated, pass the result onto the next layer until the output layer produces a prediction.

- The prediction is compared to the actual known value, and the variance between the predicted and actual values (which we call the loss) is calculated.
- Based on the results, revised weights and bias values are calculated to reduce the loss, and these adjustments are backpropagated to the neurons in the network layers.
- The next epoch repeats the batch training forward pass with the revised weight and bias values, hopefully improving the model's accuracy (by reducing the loss).

Optimisers

- The optimiser determines in which direction we need to adjust **w** and **b** (up or down) to reduce the overall loss in the model.

- There are multiple commonly used optimisation algorithms:
  - Stochastic Gradient Descent (SGD),
  - Adaptive Learning Rate (ADADELTA),
  - Adaptive Momentum Estimation (Adam), etc.

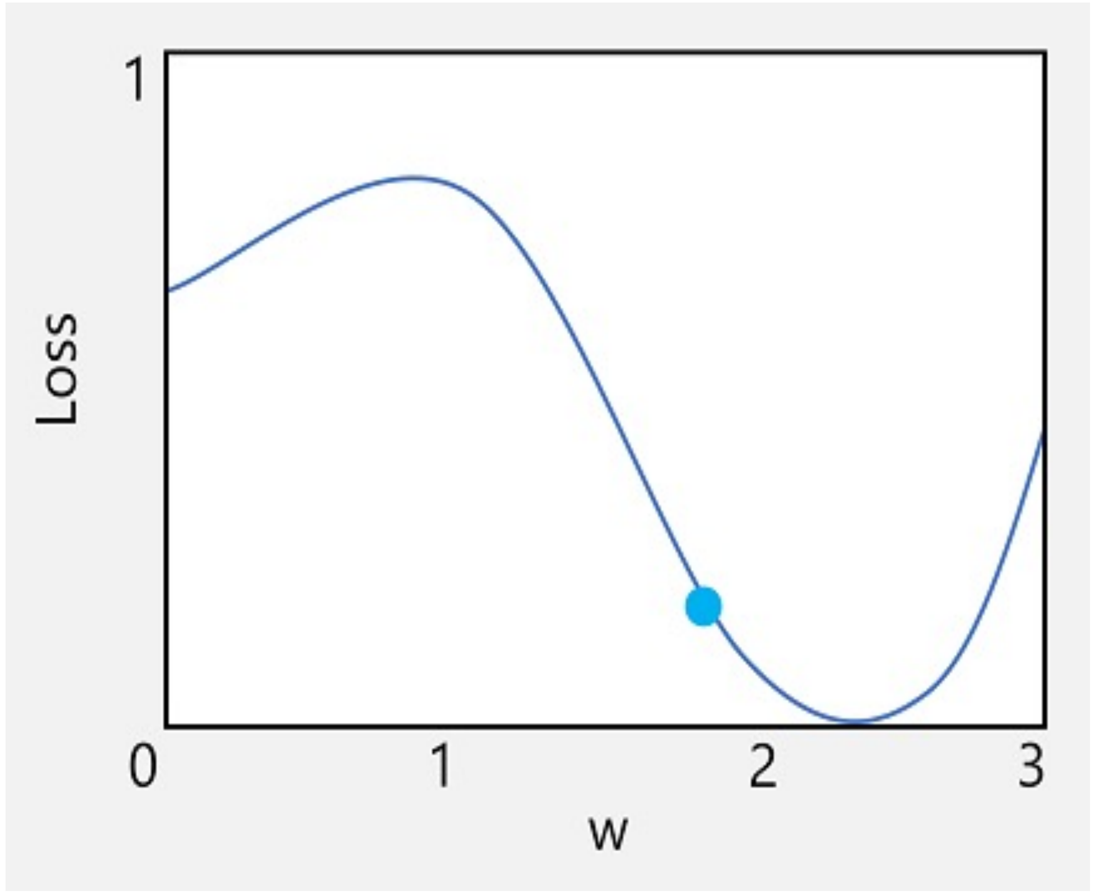


Learning Rate

- The size of the adjustment of the variables is controlled by a parameter that you set for training called the learning rate.

- A **low** learning rate results in **small adjustments**
  - so it can take more epochs to minimize the loss

- A **high** learning rate results in **large adjustments**
  - so you might miss the minimum altogether.



Activity: Deep Learning with TensorFlow

- Go through the Jupyter notebook on deep learning with tensorflow.

- Click on Image to Open the File

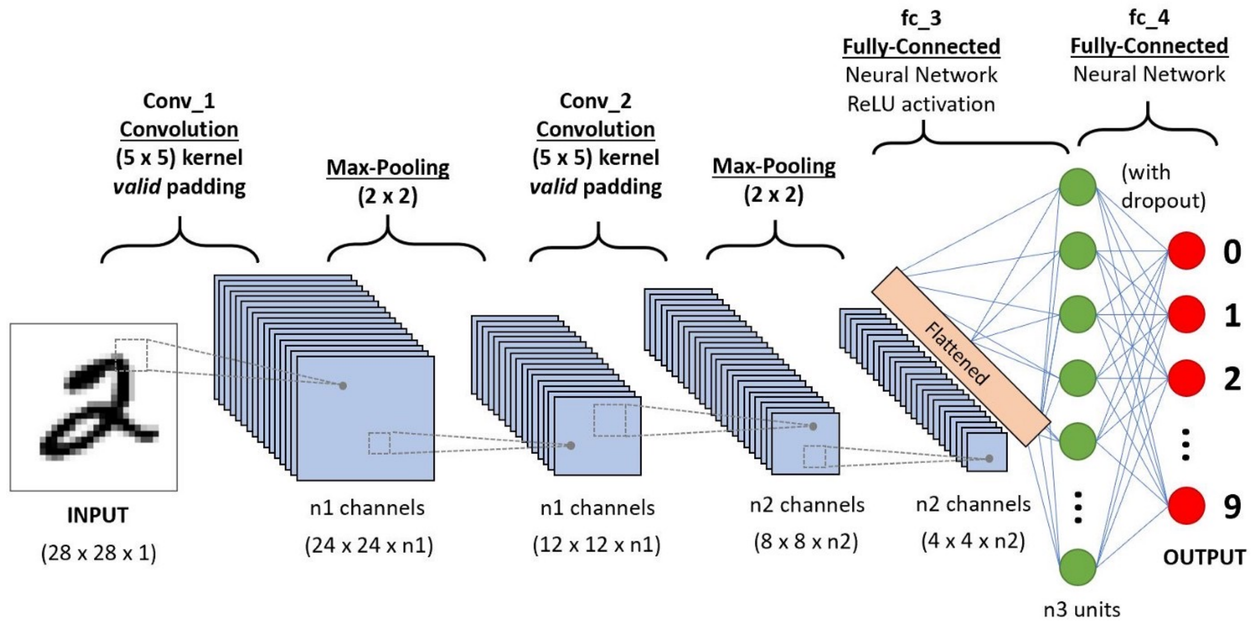
Convolutional Neural Networks

- While you can use deep learning models for any machine learning, they're particularly useful for dealing with data that consists of large arrays of numeric values - such as images.
- Machine learning models that work with images are the foundation for an area of artificial intelligence called computer vision.
- Deep learning techniques have driven remarkable advances in this area over recent years.
- At the heart of deep learning's success in this area is a kind of model called a convolutional neural network, or CNN.

Convolutional Neural Networks

- A CNN typically works by extracting features from images and then feeding those features into a fully connected neural network to generate a prediction.

- The feature extraction layers in the network have the effect of reducing the number of features from the potentially huge array of individual pixel values to a smaller feature set that supports label prediction.



Convolution Process in Action

Pooling Process in Action

ARTIFICIAL INTELLIGENCE FOR DEVELOPMENT AFRICA

RESPONSIBLE ARTIFICIAL INTELLIGENCE LAB

IDRC · CRDI
International Development Research Centre
Centre de recherches pour le développement international
Canadä

Sida
SWEDISH INTERNATIONAL DEVELOPMENT COOPERATION AGENCY

german cooperation
DEUTSCHE ZUSAMMENARBEIT

Implemented by:
giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

FAIR FORWARD
Artificial Intelligence for all.

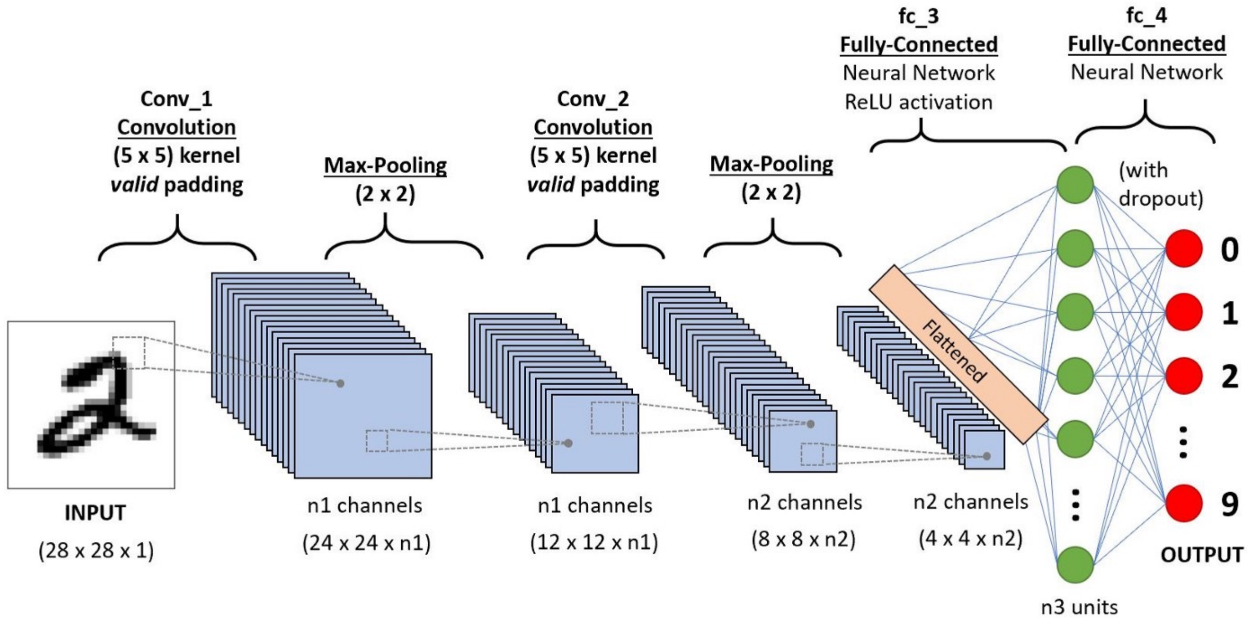
Dropping Layers

- One of the most difficult challenges in a CNN is the avoidance of overfitting,
- One technique you can use to mitigate overfitting is to include layers in which the training process randomly eliminates (or "drops") feature maps.
- This may seem counterintuitive, but it's an effective way to ensure that the model doesn't learn to be over-dependent on the training images.
- Other techniques to mitigate overfitting include randomly flipping, mirroring, or skewing the training images to generate data that varies between training epochs.

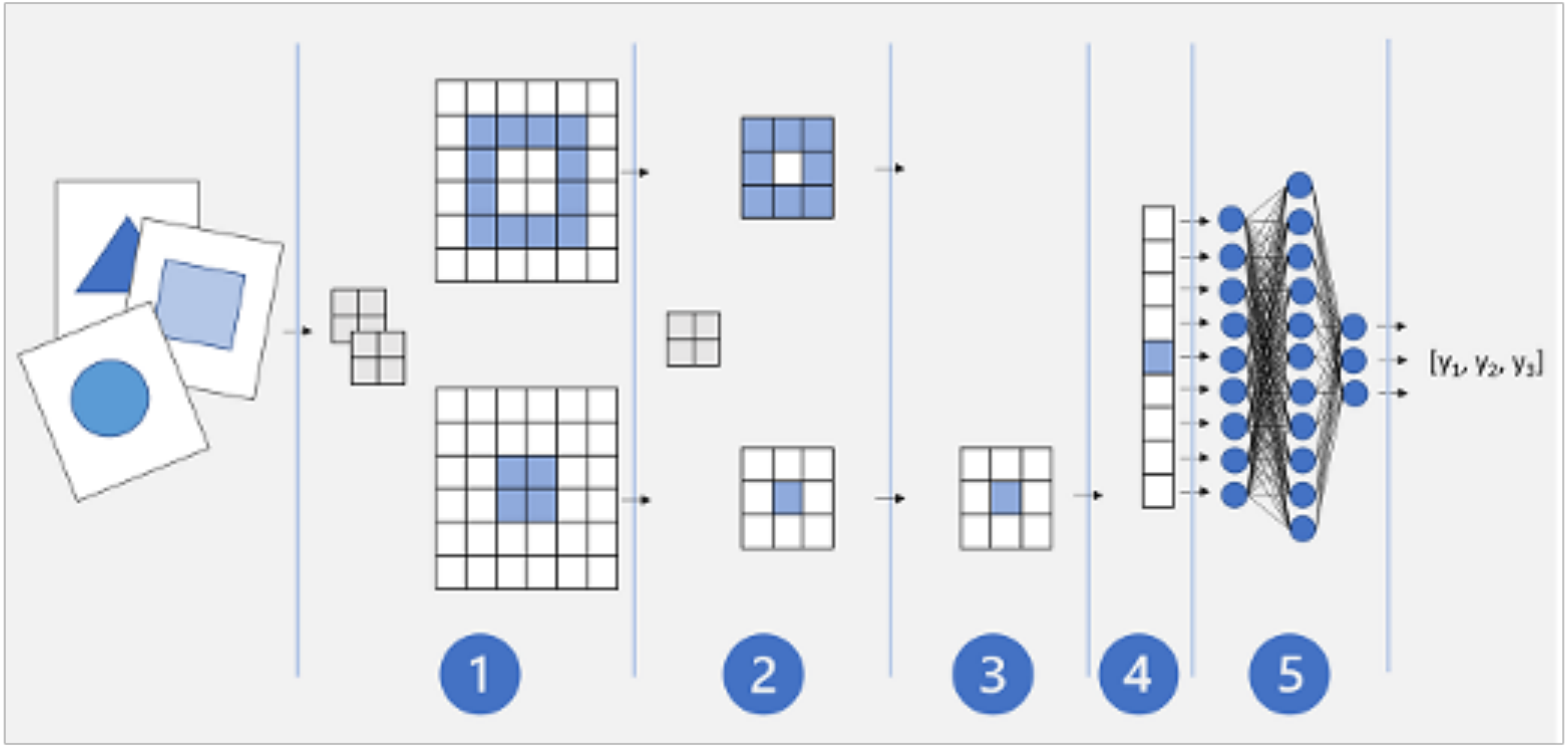
Flattening Layers

- After using convolutional and pooling layers to extract the salient features in the images, the resulting feature maps are multidimensional arrays of pixel values.

- A flattening layer is used to flatten the feature maps into a vector of values that can be used as input to a fully connected layer.



Fully Connected Layers



Activity: Train a Convolutional Neural Network

- Go through the Jupyter notebook on convolutional neural networks with TensorFlow.

- Click on Image to Open the File

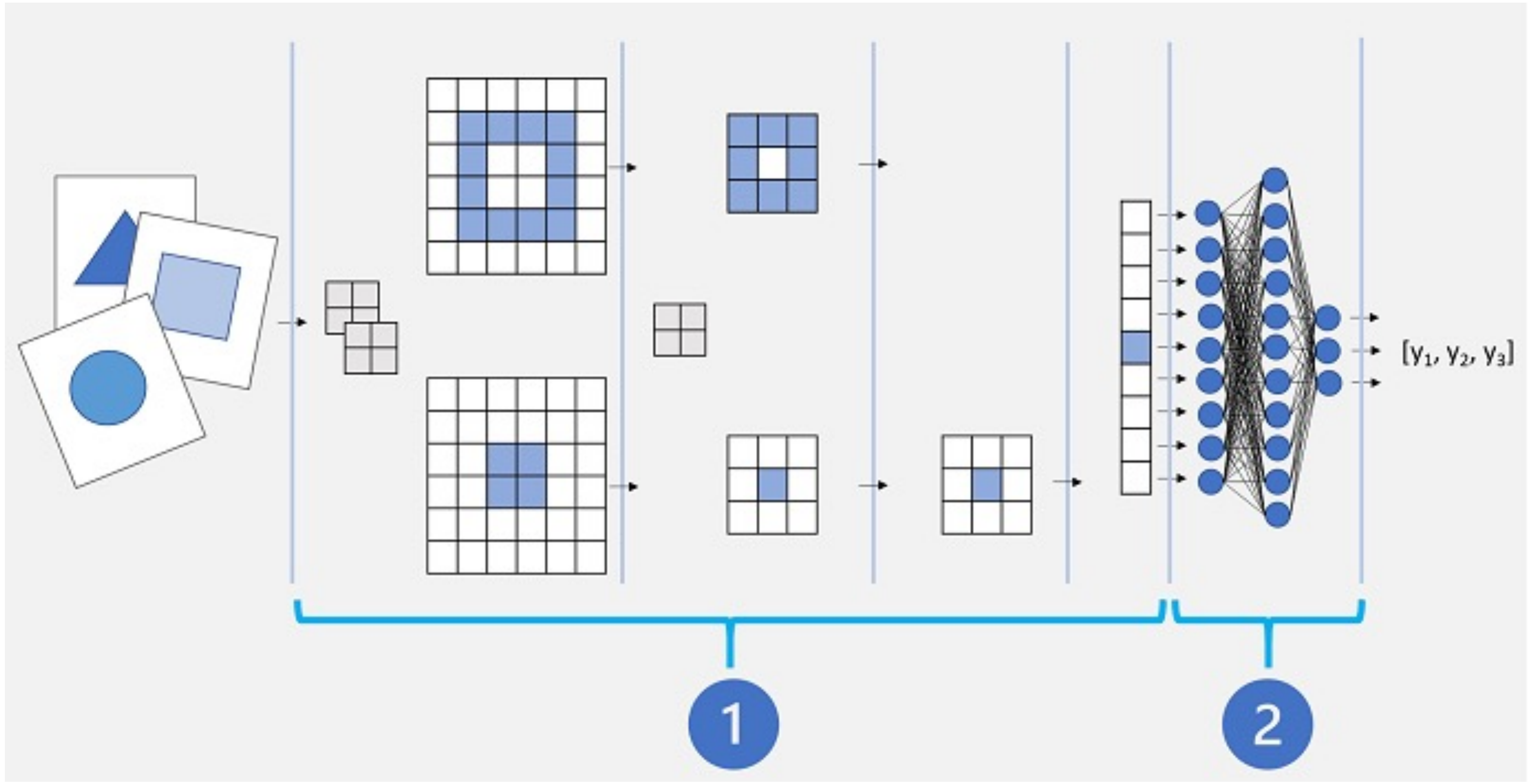
Challenge: Safari Challenge

- Go through the Jupyter notebook on Safari Challenge.
- Click on Image to Open the File

Transfer Learning

- In life, it’s often easier to learn a new skill if you already have expertise in a similar, transferrable skill.
- For example, it’s probably easier to teach someone how to drive a bus if they have already learned how to drive a car.
- The driver can build on the driving skills they've already learned in a car, and apply them to drive a bus.
- The same principle can be applied to training deep learning models through a technique called **transfer learning**.

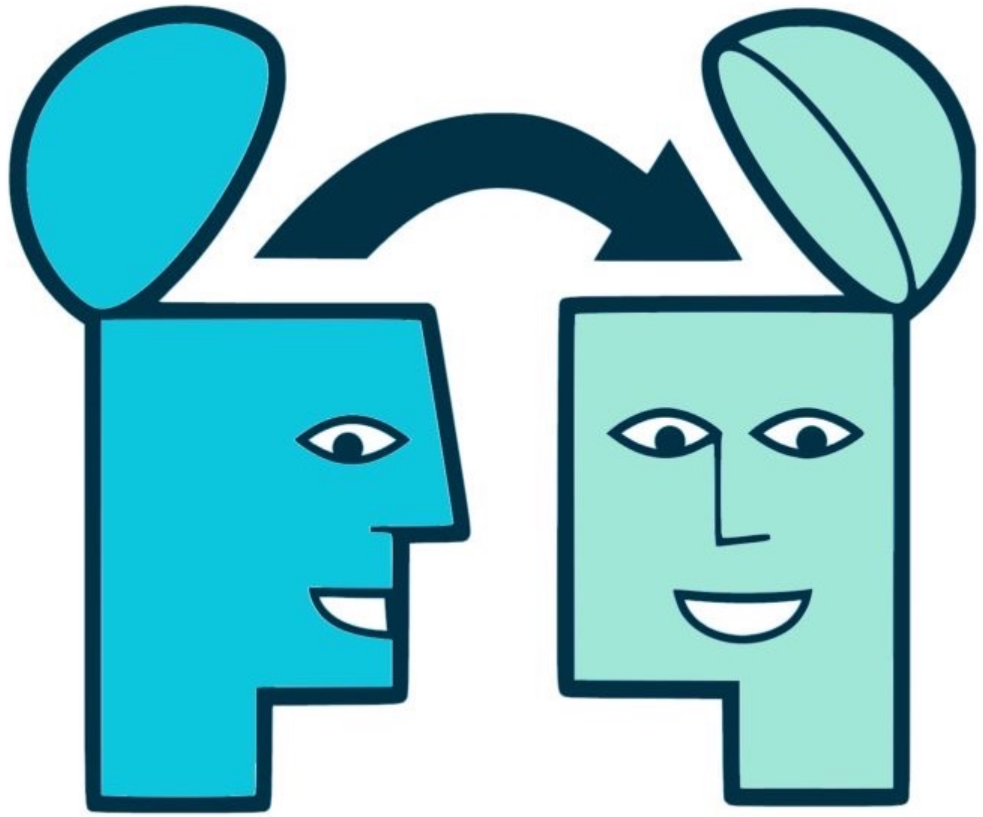
Transfer Learning



Activity: Transfer Learning with TensorFlow

- Go through the Jupyter notebook on transfer learning with TensorFlow.

- Click on Image to Open the File



Knowledge Check - 1

- You are creating a deep neural network to train a classification model that predicts to which of the three classes an observation belongs based on ten numeric features. Which of the following statements is true of the network architecture

A. The input layer should contain three nodes.

B. The network should contain three hidden layers,

C. The output layer should contain three nodes.

Knowledge Check - 2

- You are training a deep neural network. You configure the training process to use 50 epochs. What effect does this configuration have?

A. The entire training dataset is passed through the network 50 times.

B. The training data is split into 50 subsets, passing each subset through the network.

C. The first 50 rows of data are used to train the model, and the remaining rows are used to validate it

Knowledge Check - 3

- You are creating a deep neural network. You increase the Learning Rate parameter. What effect does this setting have?

A. More records are included in each batch passed through the network.

B. Larger adjustments are made to weight values during backpropagation.

C. More hidden layers are added to the network

Knowledge Check - 4

- You are creating a convolutional neural network. You want to reduce the size of the feature maps that are generated by a convolutional layer. What should you do?

A. Reduce the size of the kernel used in the convolutional layer.

B. Increase the number of filters in the convolutional layer.

C. Add a pooling layer after the convolutional layer

THANK YOU FOR YOUR TIME

RAIL@KNUST.EDU.GH

FECE, COLLEGE OF ENG., KNUST, KUMASI GHANA

+233(0)3224-93435