UW DSC.

Neural Networks and Computer Vision Workshop

Presentation: https://bit.ly/2U0U8Jr
Notebook: https://bit.ly/32pKlRJ

Presented by Jack Douglas

Workshop Overview

UW DSC.

1. Machine Learning Refresher
2. Neural Networks Introduction
    a. Neurons & Layers
    b. Biases & Weights
    c. Activation Function
    d. Loss & Optimizers
    e. Terminology
3. Applications w/ ANN's
    a. Simple feedforward neural network from scratch
    b. Classification ANN w/ TensorFlow
4. Convolutional Neural Networks
    a. Kernel & Filters
    b. Convolution Layer
    c. Pooling Layer
    d. Fully Connected Layer

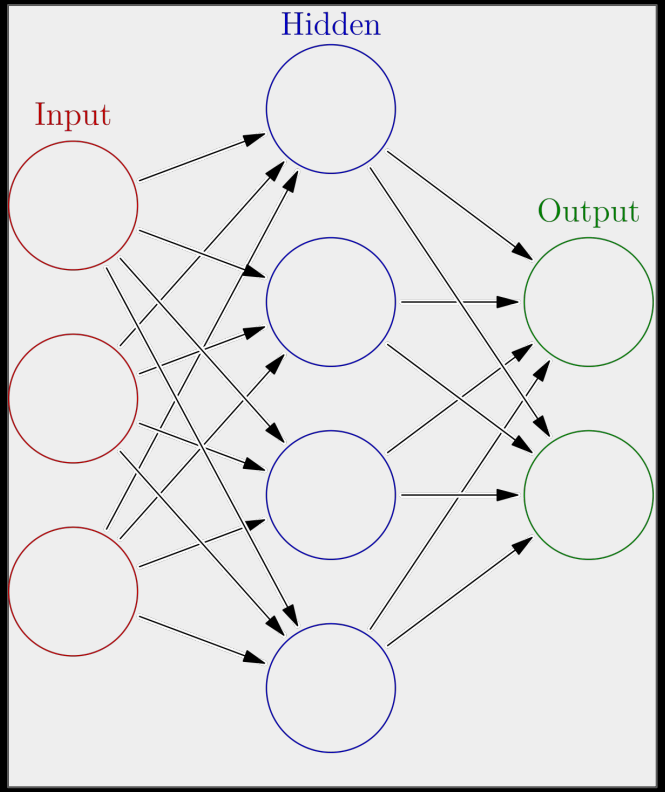
Machine Learning Refresher

UW DSC.

1. Types of Learning
    a. **Supervised Learning:** Finds a correlation between given inputs and outputs (labels)
    b. **Unsupervised Learning:** Finds how to structure unlabelled inputs
    c. **Reinforcement Learning:** Performs a task and improves by maximizing a reward
2. Types of Supervised Learning Problems
    a. **Classification:** Predicting a label
        i. Ex. Distinguish between a cat and dog, given a labelled dataset with photos of both
    b. **Regression:** Predicting a quantity
        i. Ex. Predict the price of house, given a labelled dataset of housing prices along with other factors (lot area, year built, etc.)

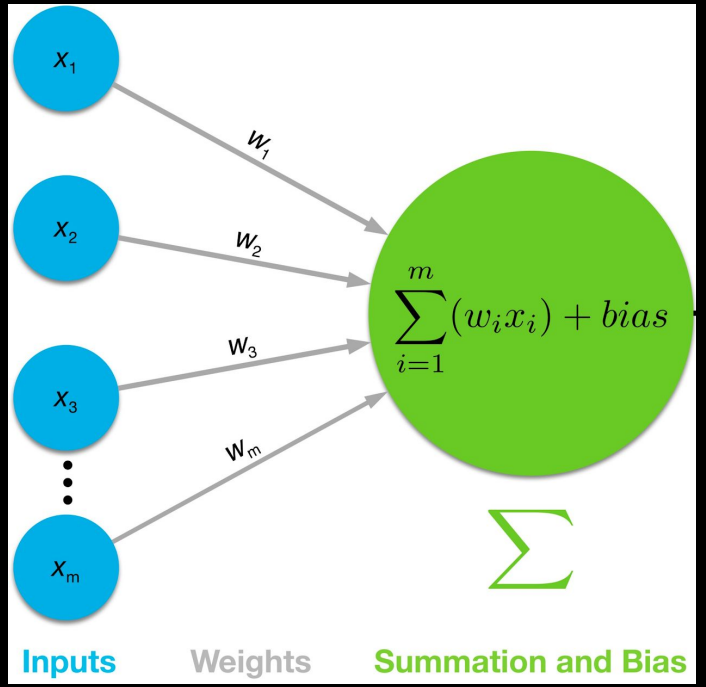
Neurons & Layers

1. **Neurons**
    a. Def'n: Elementary units of a neural network
    b. Can receive one or multiple inputs
    c. The output (or **activation**) of a neuron is calculated using a formula which I will talk about in the following slides
2. Types of Layers
    a. **Input Layer:** Refers to the first layer of the ANN where all the inputs are received
    b. **Hidden Layer:** Refers to layers in between the input and output layers
    c. **Output Layer:** Refers to the last layer of the ANN where the outputs are received



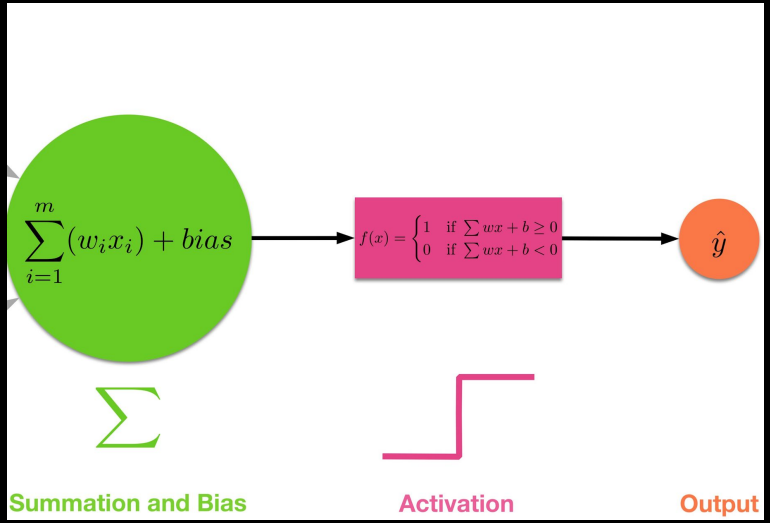
Weights & Biases

1. **Weights**
   a. Def'n: Controls the signal strength between two neurons
2. **Biases**
   a. Def'n: A constant which is added to the linear combination of weights and signal
   b. It helps with learning by shifting activation function left and right
3. These are called the **training parameters**



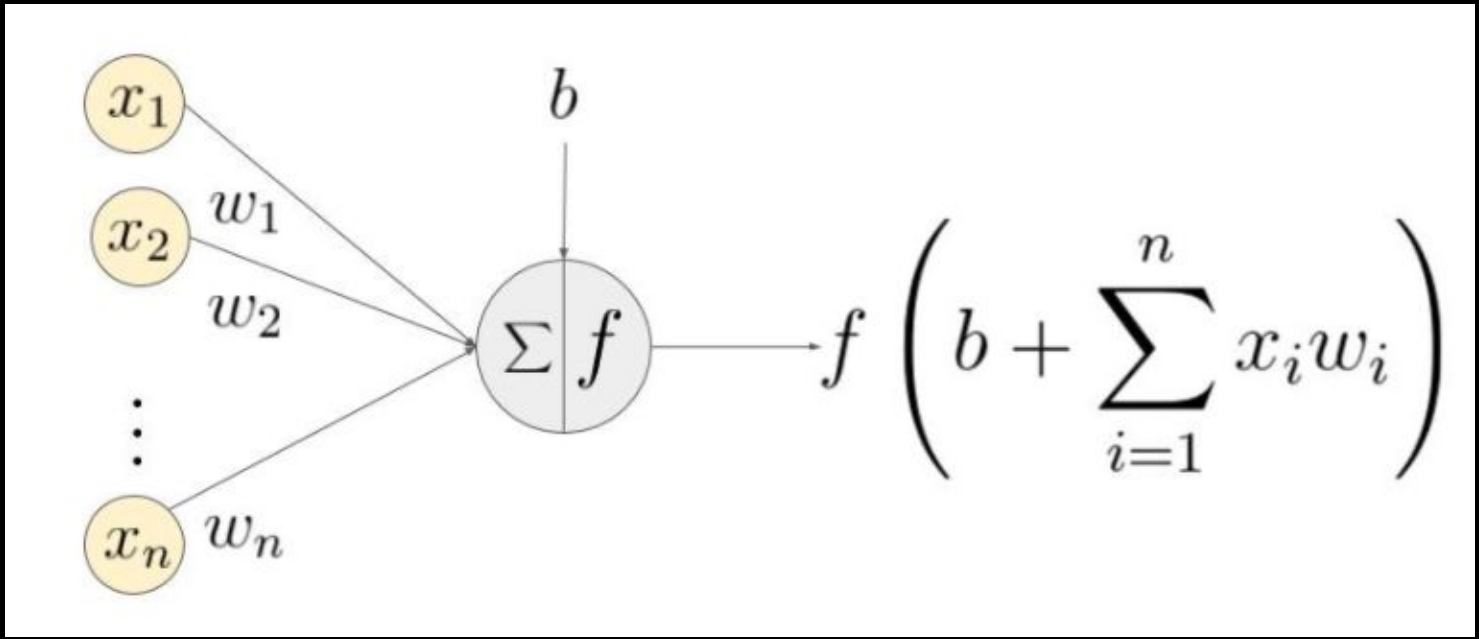
Activation Function

1. **Activation Function**
   a. Refers to the function which determines the output of a neuron
2. Motivation
   a. Can be thought of as turning a neuron OFF (0) and ON (1)
   b. Allows neural networks to learn complex "functions"

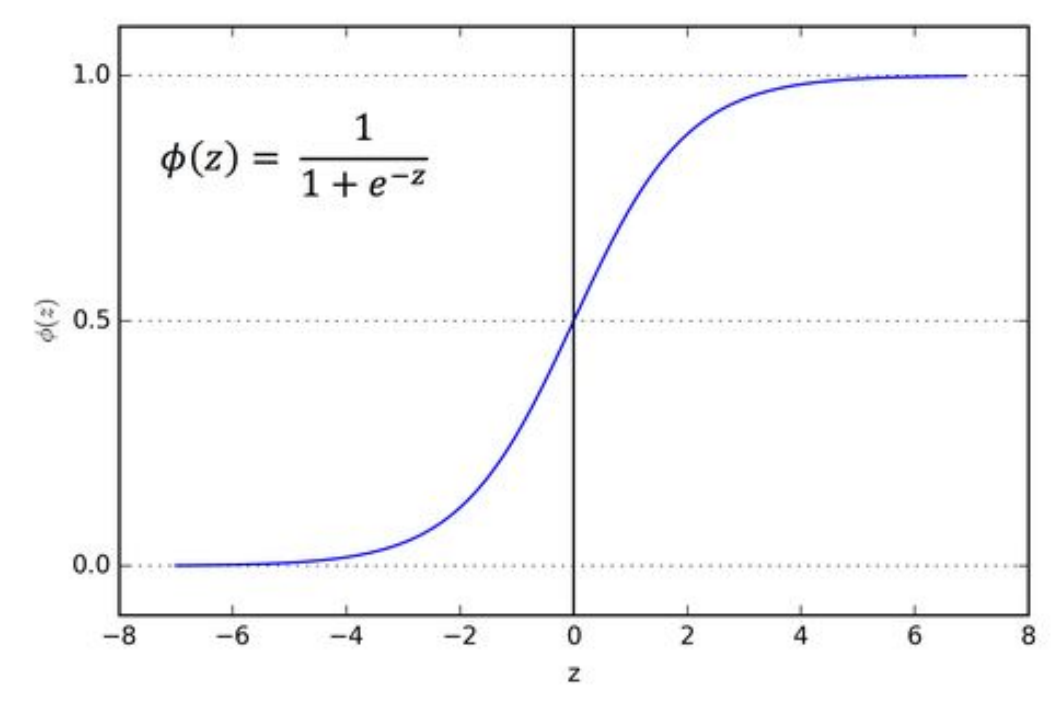


Note: there are many different activation functions, and the Heaviside function is the simplest.

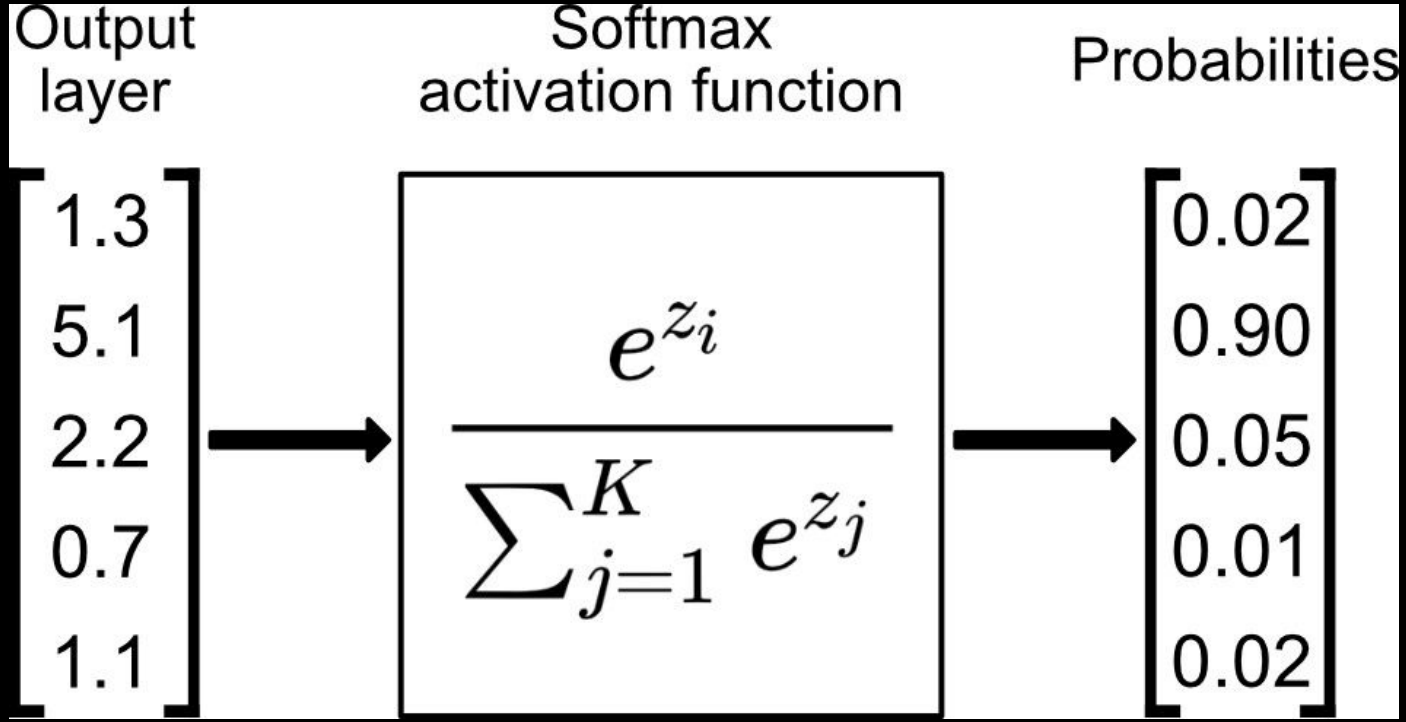
Activation Function



Sigmoid Activation Function

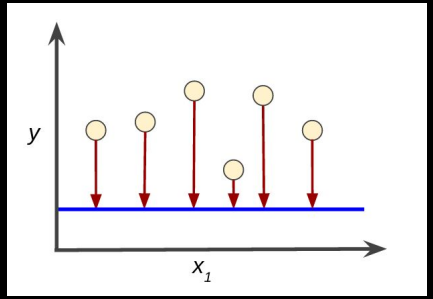


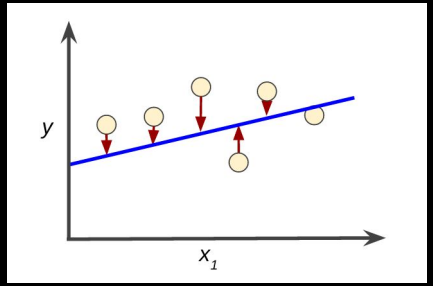
Softmax Activation Function



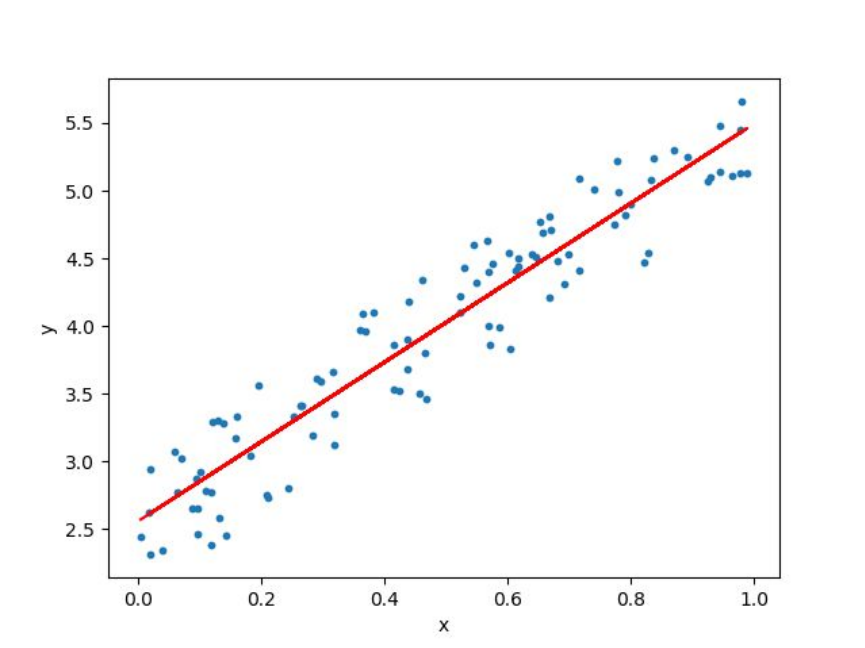
Loss & Optimizers

1. **Loss**
   a. An indicator of how bad the model's prediction was on a single sample
2. **Loss/Cost Function**
   a. A function which calculates the error in the model across all samples
3. **Optimizers**
   a. Algorithms which aim to minimize loss by changing the training parameters



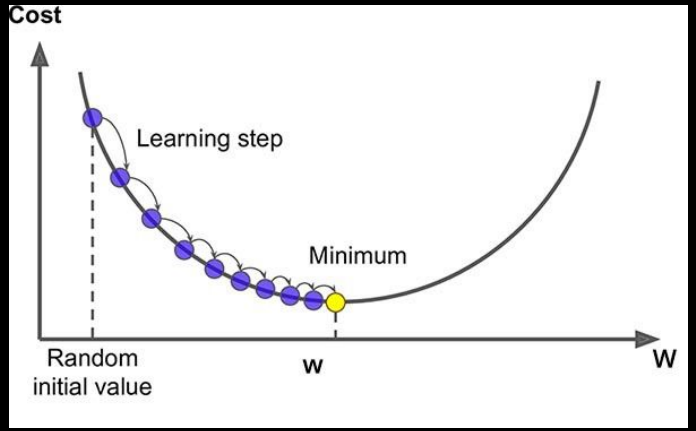


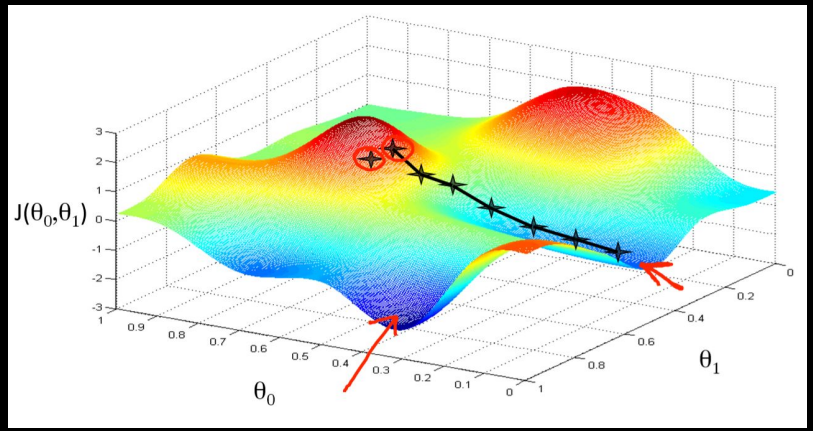
Mean Squared Error (Loss Function)



$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Gradient Descent (optimizer)





repeat until convergence {

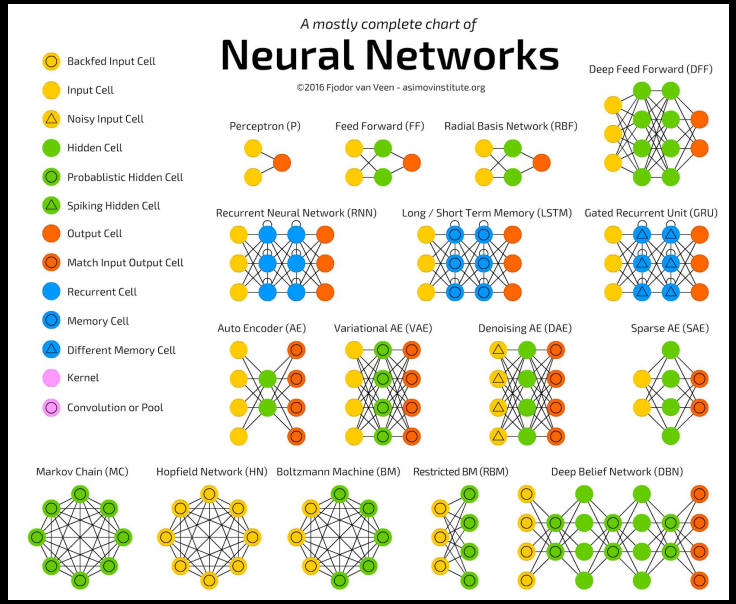
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \text{(for } j = 0 \text{ and } j = 1\text{)}$$

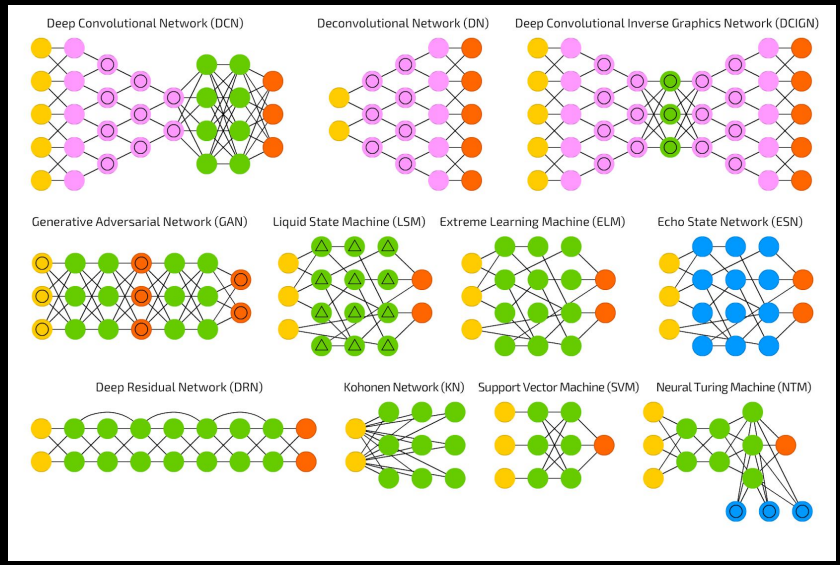
}

Terminology

1. **Batch Size**
   a. Refers to the number of training samples in one iteration before the training parameters (ie. weights and biases)
2. **Epoch**
   a. Refers to the number of cycles through the training data
3. **Fit/Train**
   a. Process of changing weights and biases by methods mentioned
4. **Overtrain**
   a. The model has created a “function” that is too specific to training data
5. **Undertrain**
   a. The model has created a “function” which is too general
6. **Learning Rate**
   a. The hyperparameter which says how much the weights are updated during training
7. **Class**
   a. A category or label
8. **Backpropagation**
   a. An algorithm which calculates the gradient efficiently in the gradient descent optimizer

Neural Network Architectures



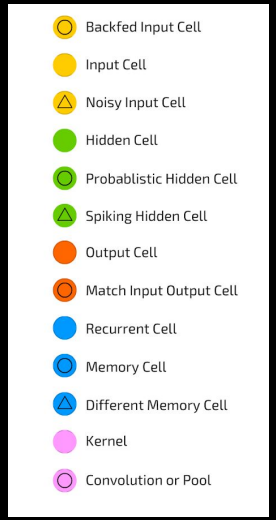


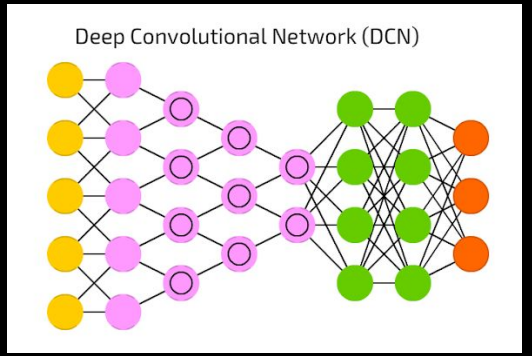
UW DSC.

Applications with ANN’s

Convolutional Neural Networks

1. Computer Vision
    a. CNN's are used in computer vision because many of the building blocks within a CNN can extract high and low level features
2. Building blocks
    a. Kernels & filters
    b. Convolutional layer
    c. Pooling layer
    d. Fully convolutional layer

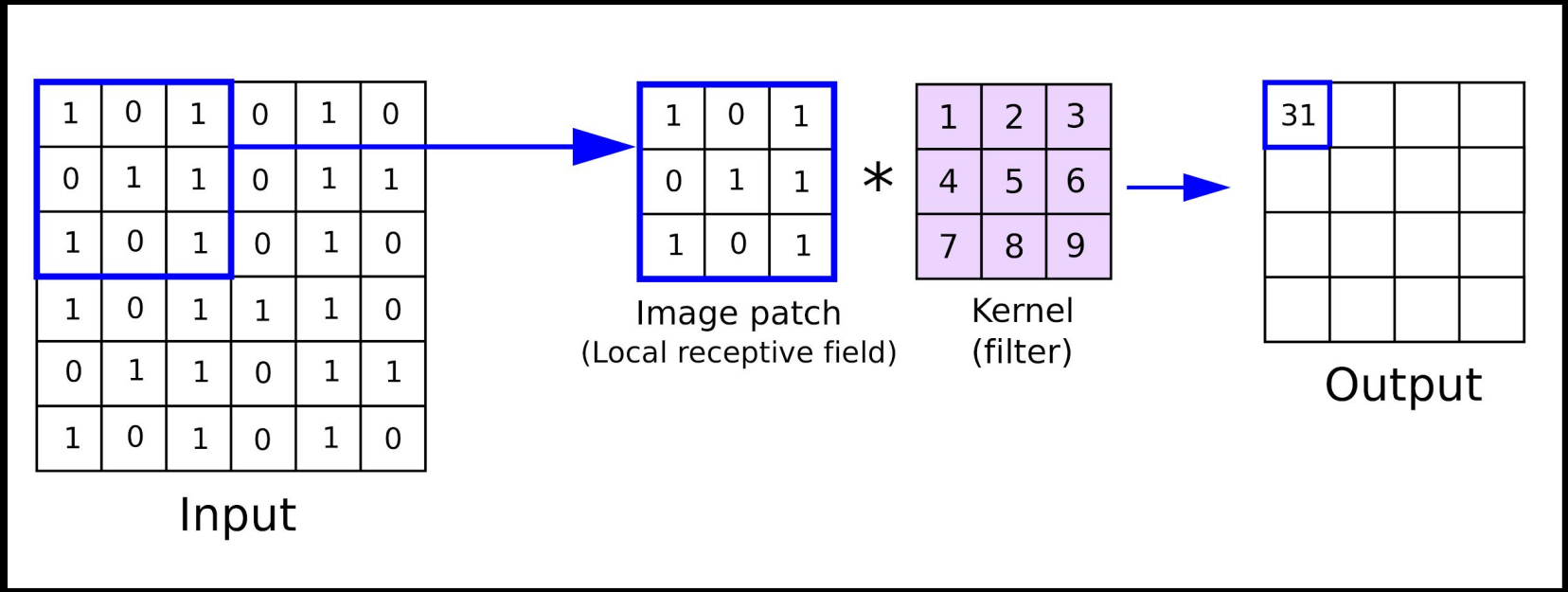




Kernels & Filters

1. **Kernel**
   a. Def'n: A matrix which is applied to an image to extract features
2. **Filter**
   a. Def'n: Multiple kernels stacked together
3. **Convolve**
   a. Def'n: The process of a kernel being applied across an entire image
4. **Stride**
   a. Def'n: The number of pixels the kernel moves before being applied again

Kernels & Filters

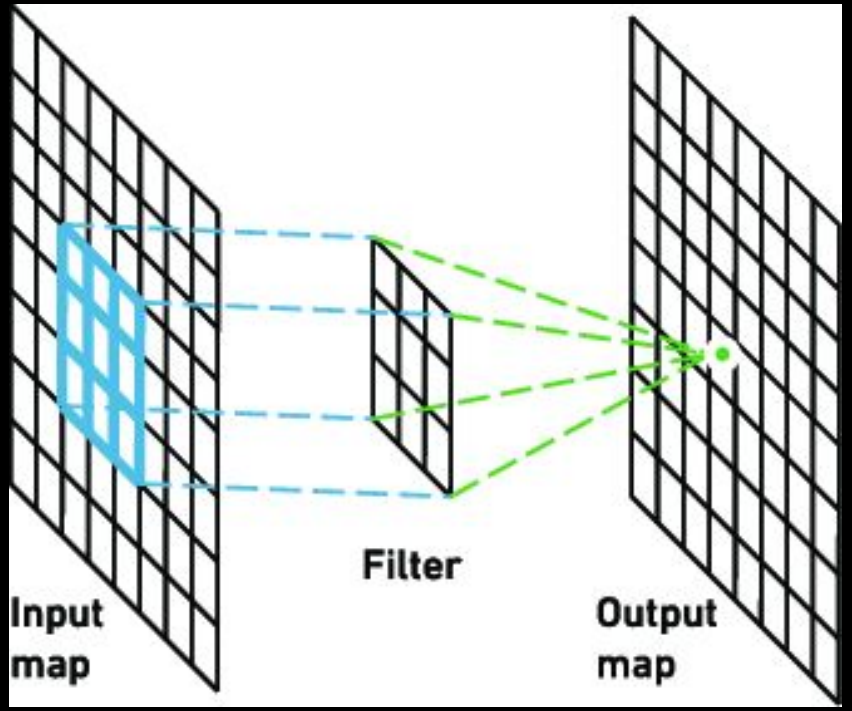


Note that the stride in this example is 1 and that since there is a single kernel being applied, the filter and kernel are the same thing.

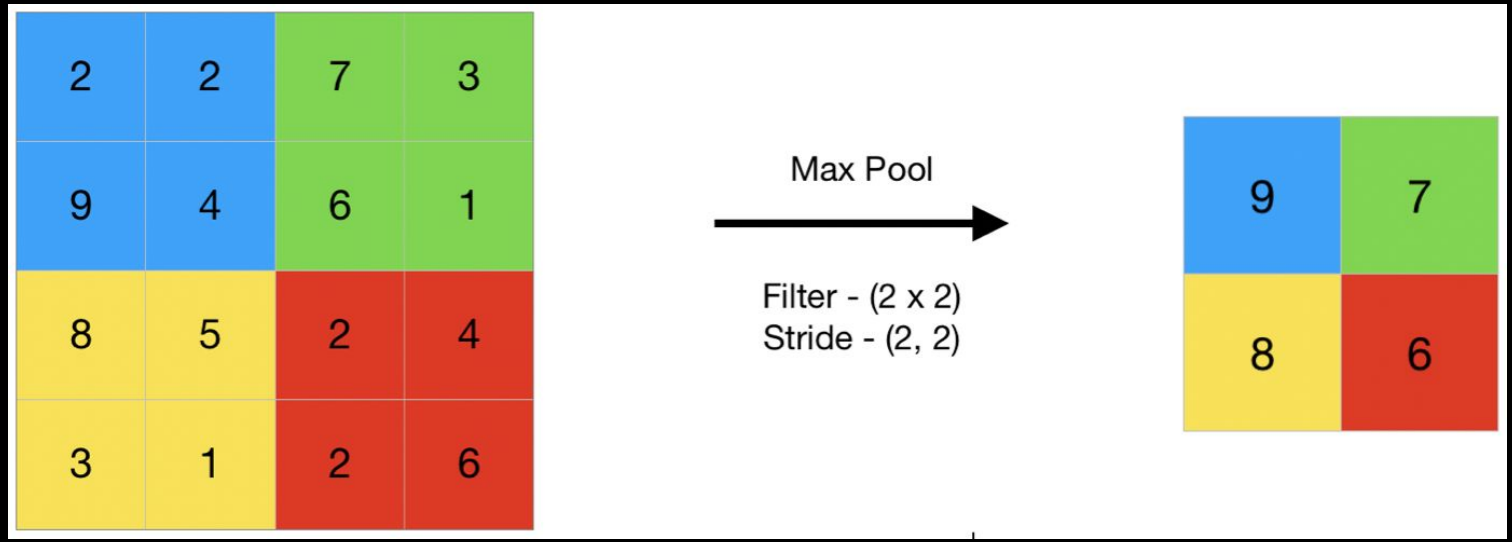
Types of CNN Layers

1. **Convolutional Layer**
    a. Def'n: A layer where a kernel/filter is applied to an image
    b. Used to extract features from an image (ie. edges of an object, particular shapes, etc.)
2. **Pooling Layer**
    a. Def'n: A layer which reduces the spatial size of the image to help lower the number of parameters
    b. You can think of it as globbing together areas of an image that are similar
3. **Fully Connected Layer**
    a. Def'n: A layer in which all the neurons are connected to every neuron in the following layer
    b. Takes the features extracted and the information from pooling layers to classify an image

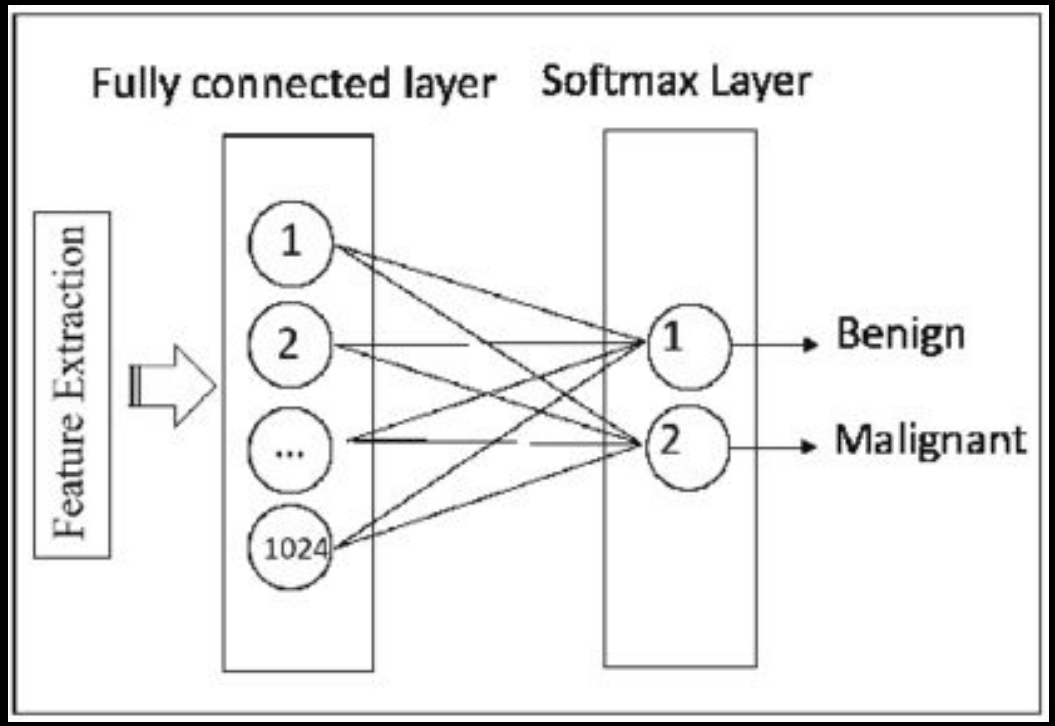
Convolutional Layer



Pooling Layer



Fully Connected Layer



UW DSC.

Questions