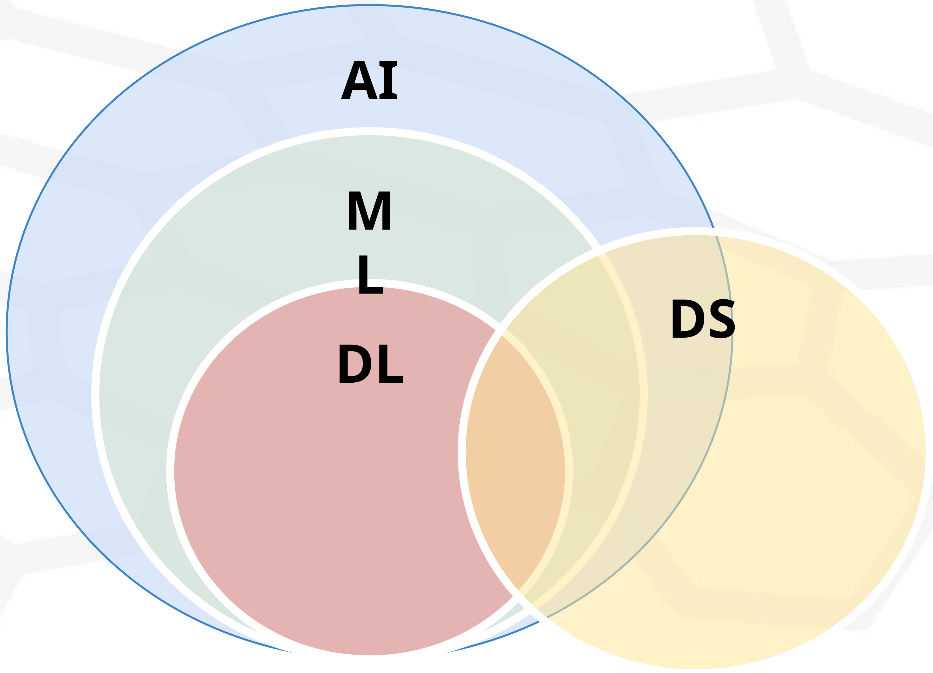
Deep Learning

What is Deep Learning?



Artificial Intelligence (AI)
The development of a machine that can simulate human behavior.

Machine Learning (ML)
The process of a machine learning from data and making decisions without explicit programming.

Deep Learning (DL)
A term associated with a ML algorithm involving artificial neural networks that work to mimic neurons in the brain.

Data Science (DS)
The process of using data analytics, statistics, and programming to solve business problems.

Video

Top 5 Uses of Neural Networks! (A.I.) | ColdFusion

https://www.youtube.com/watch?v=i9MfT_7R_4w

Pros & Cons of Deep Learning

Pros

- Deep learning automates much of the feature extraction process, which can be particularly useful for things like image recognition.
- Deep learning is flexible and can be applied to many different datasets.
- Deep learning models can generally outperform than other types of machine learning models.

Cons

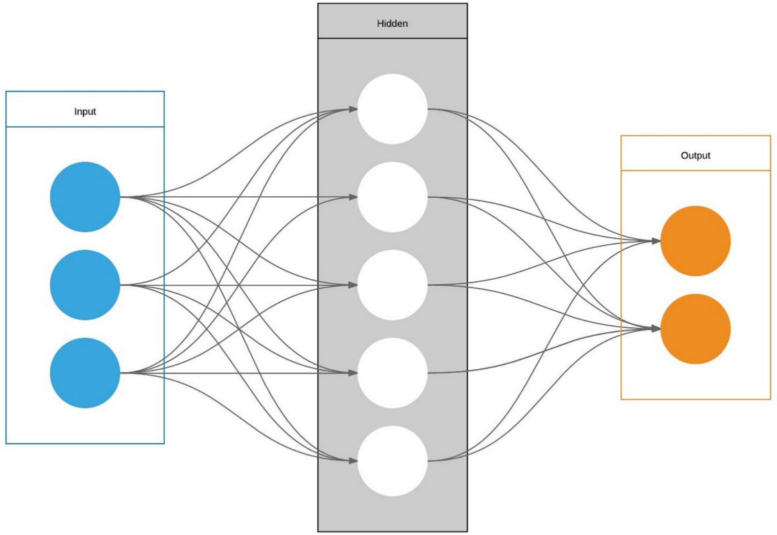
- Deep learning requires large datasets to train the model.
- Deep learning can be very computationally expensive to train.
- There are many modeling decisions that need to be made.
- Results from a deep learning model are often uninterpretable. It is often difficult to understand why it is making its prediction.

Deep Learning & Neural Networks

Deep learning is a subfield of machine learning.

Neural networks are the foundation of deep learning algorithms. The “deep” in deep learning refers to the number of layers in the neural network (we will discuss this shortly).

Neural networks mimic the brain through the use of algorithms.



Video

Neural Network In 5 Minutes | What Is A Neural Network? | How Neural Networks Work | Simplilearn

https://www.youtube.com/watch?v=bfmFfD2RIcg

Artificial Neural Networks

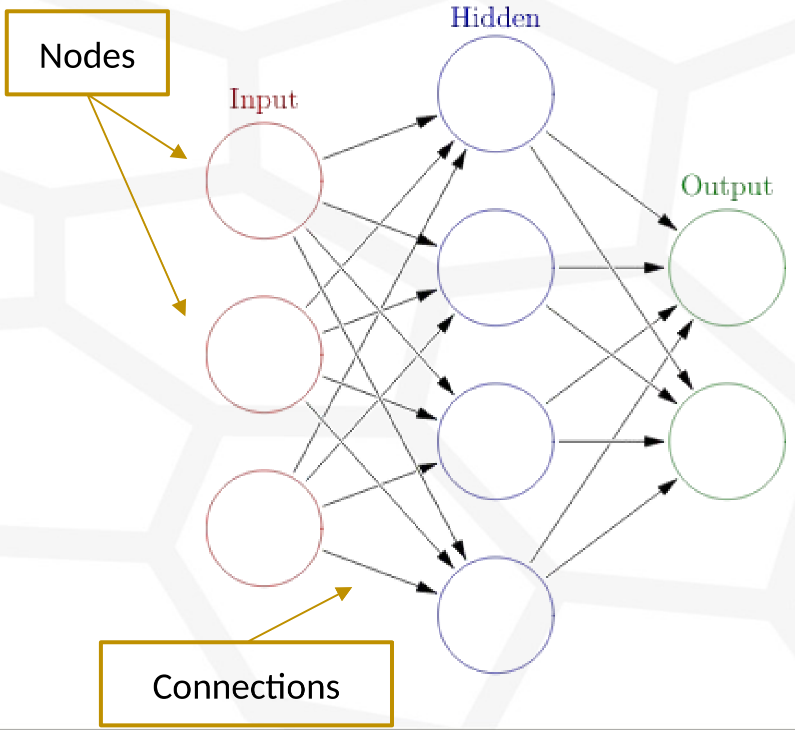
Artificial Neural Networks (ANNs) are a system made up of a collection of connected neurons (also called nodes).

Neurons are able transmit a signal to each other through connections.

Neurons are organized into layers. There are three types of layers

1) Input
2) Hidden
3) Output

Different layers perform different transformations on the data.



Perceptron

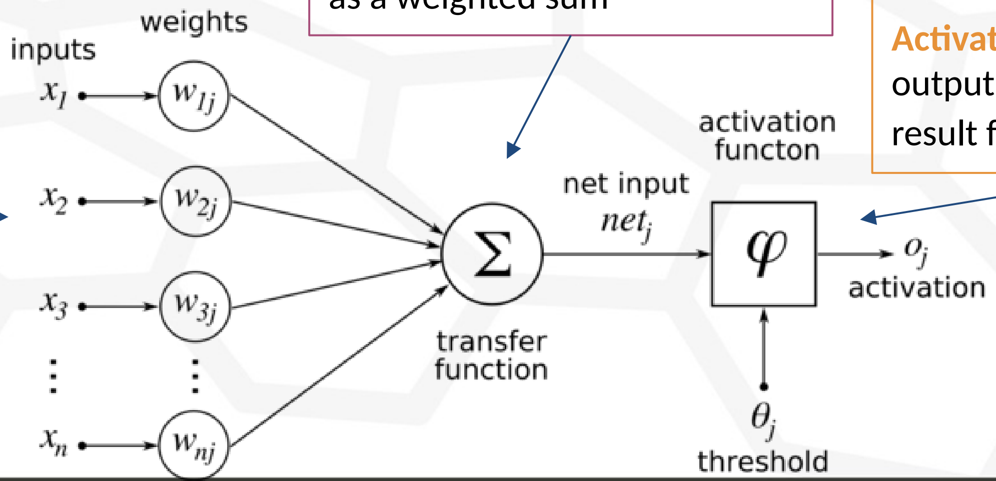
The most basic neural network is called a perceptron. It is broken up into the following pieces:

Weights - each input has a weight associated with it (can be any real number)

Transfer Function - inputs and weights are combined, usually as a weighted sum

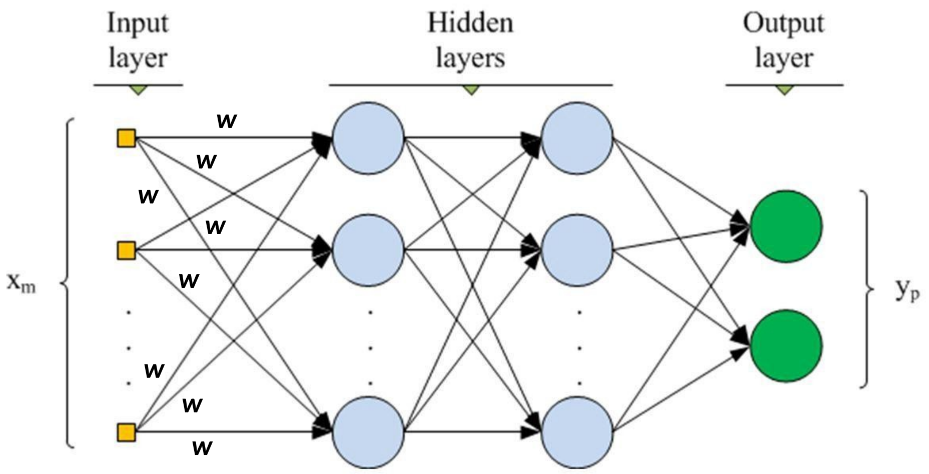
Activation Function - defines the output of the neuron given the result from the transfer function

Inputs - the data (in numerical form)



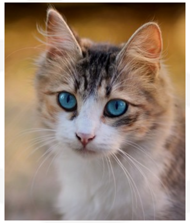
Multi-Layer Perceptron aka ANN

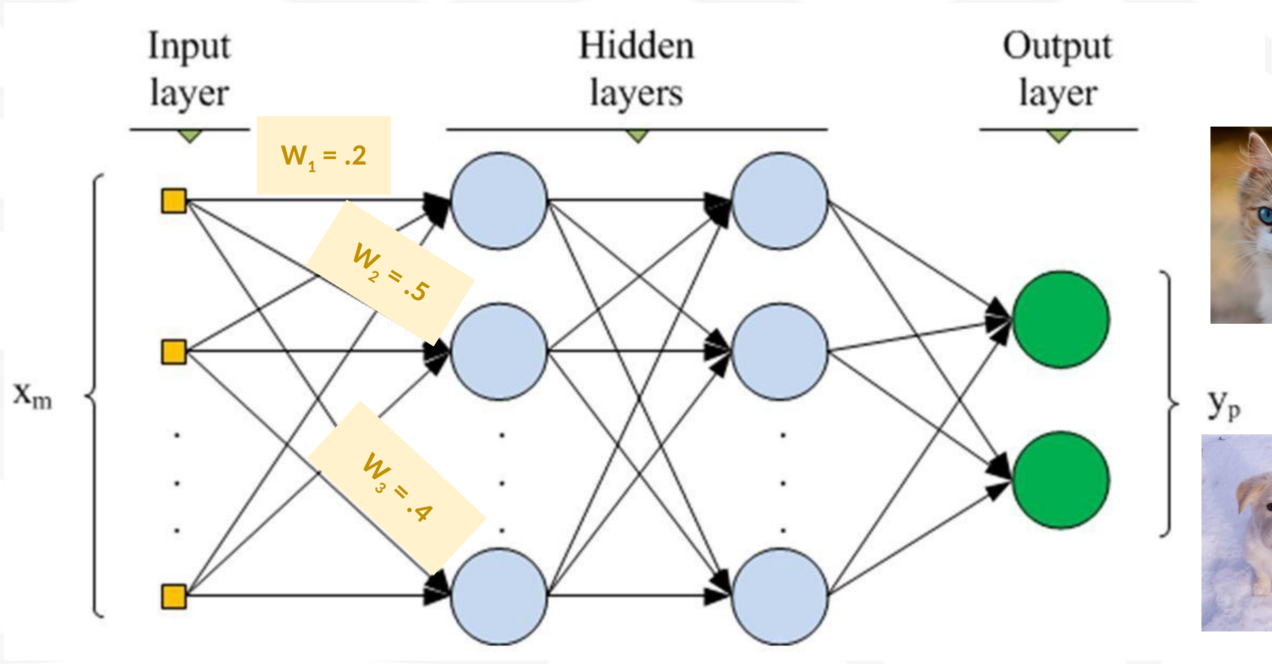
- Input layer = the data (numerical form)
- Output layer = probability that each class is the correct class
- Hidden layer = layer of “neurons” each outputting a result based on a sum of the inputs
- The “magic” lies in tuning all the weights (***w***) so that the network classifies input data correctly

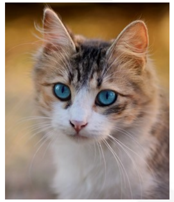


Each path between layers has a weight ***w*** associated with it

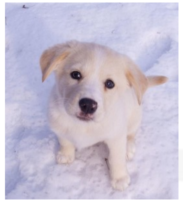
How does a Neural Network Learn?





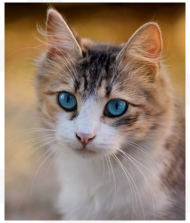


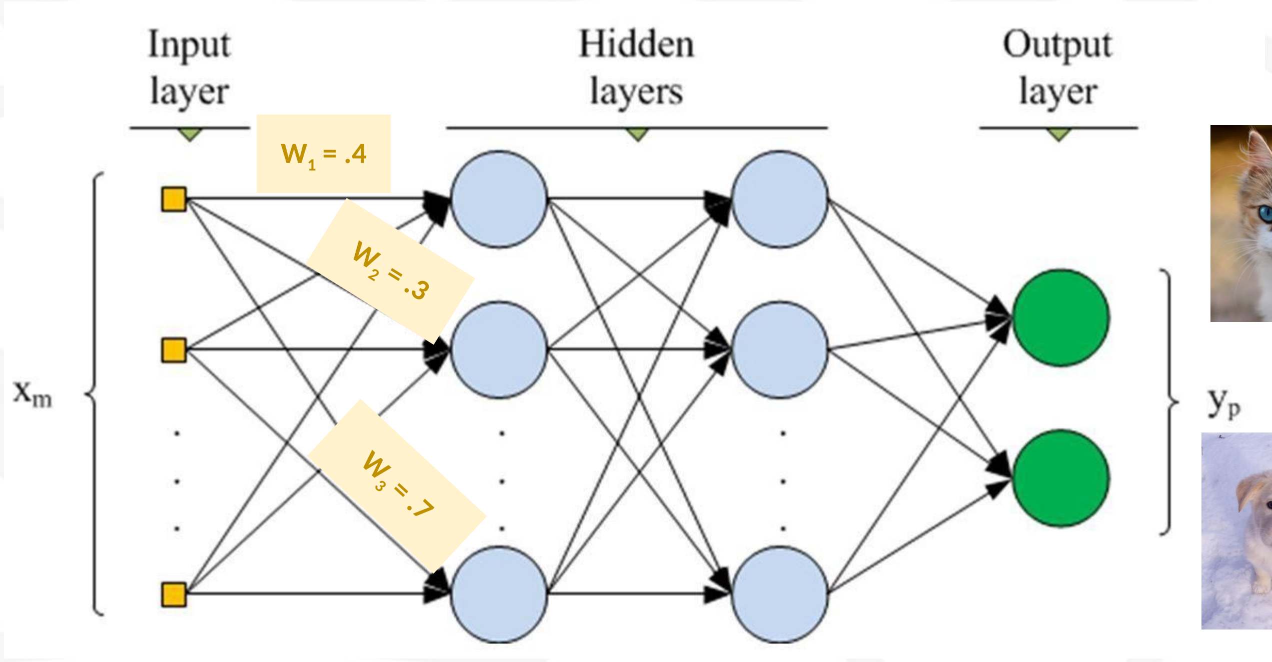
.25

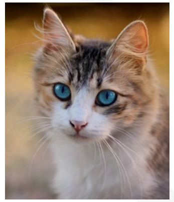


.75

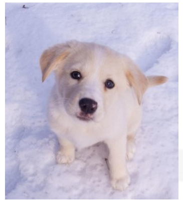
How does a Neural Network Learn?





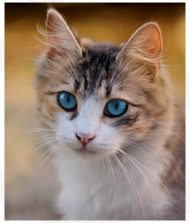


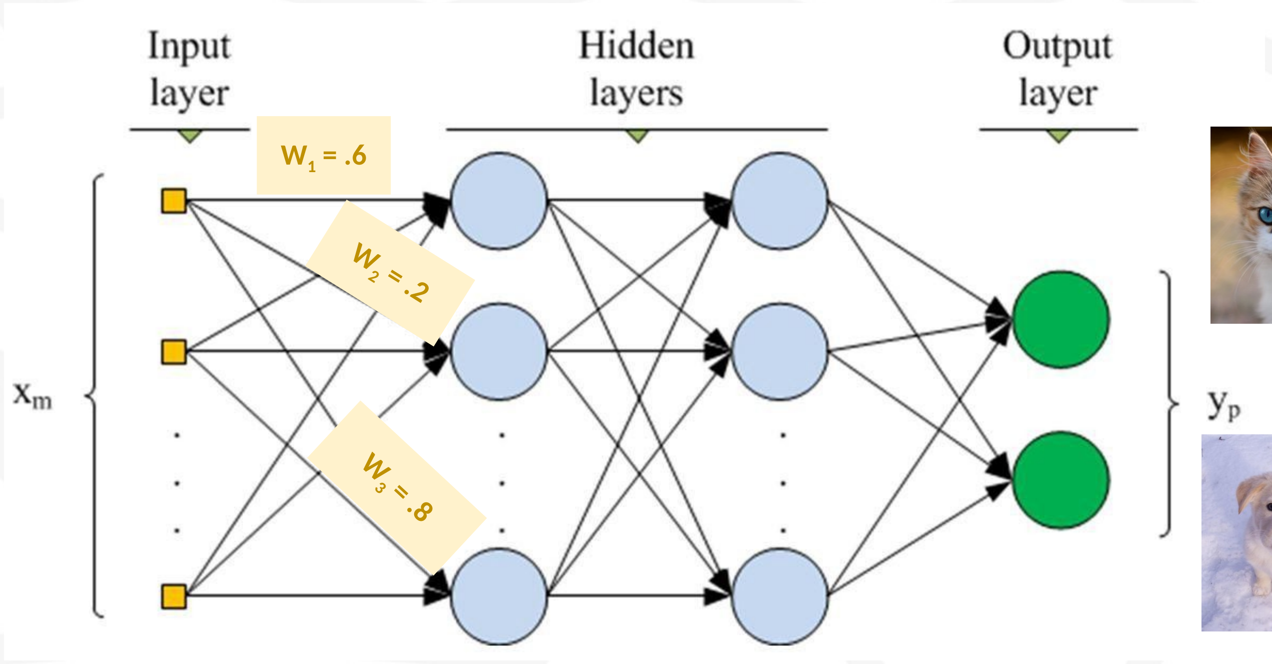
.65

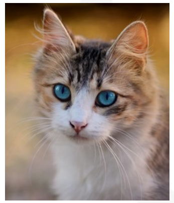


.35

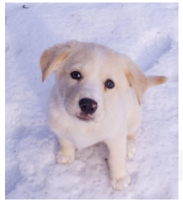
How does a Neural Network Learn?







.95



.05

Types of Neural Networks

There are many types of neural networks, each used in different circumstances. We will focus on three common types:

- Artificial Neural Networks (ANNs) - also known as feed-forward neural network. Does not account for sequential nature of data and can result in a large number of predictors in image classification.

- Recurrent Neural Network (RNN) - captures sequential nature of data. Usually used for time series and text data.

- Convolutional Neural Networks (CNNs) - uses kernels to extract relevant features from the input. Also captures spatial features of the data. Usually used for image classification.

How to Create Neural Network?

When fitting a neural network, you have to decide on several things:

- The number of layers
- The number of nodes in each layer
- The type of layers (see next slide)
- The activation function in each layer (see later slides)

How do you decide on the number of layers and nodes? The most common way is just experimentation (trial and error). There is no simple way to determine the best number of layers or nodes for your problem. You will likely need to just try different options and see what gives you the best model performance.

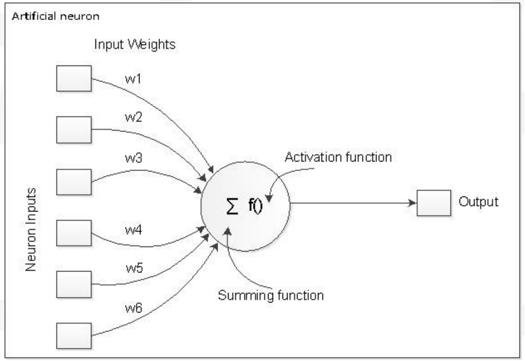
Types of Layers

There are several types of layers that can be used in a neural network. We will focus on two common types:

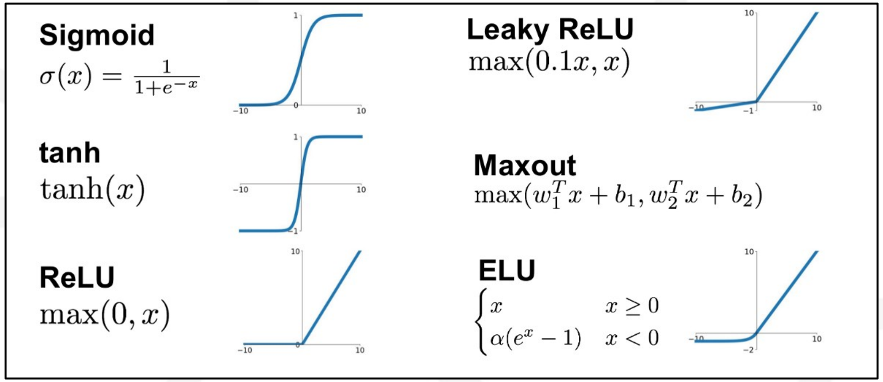
- **Dense Layers** - Dense layers are used when connections can exist among any feature to any other feature in a data set. This corresponds to an ANN.
- **Convolutional Layers** - Convolutional layers can be used to detect patterns in an image. Early convolutional layers can detect things like edges or geometric shapes. Deeper convolutional layers can detect more sophisticated patterns like eyes, noses, etc. Convolutional layers are used when nearby connections among the features are important, but farther connections are not. This corresponds to a CNN.

Activation Functions

- All weighted input values are summed and then processed with an activation function
- That output is then passed along to the next layer or to the output
- Activation functions are inspired by neurons in our brain that fire based on different stimuli.



Common Activation Functions

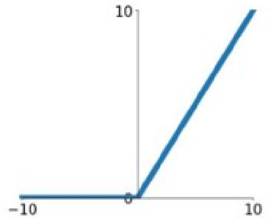


Common Activation Functions

- One thing in common to activation functions is that they are *nonlinear.*
- The baseline function we will use is the ReLU. Sigmoid is often used for binary classification.
- Other activation functions can easily be swapped in to examine if performance is increased.

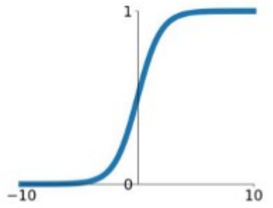
ReLU

$\max(0, x)$



Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



https://towardsdatascience.com/a-quick-guide-to-activation-functions-in-deep-learning-4042e7addd5b

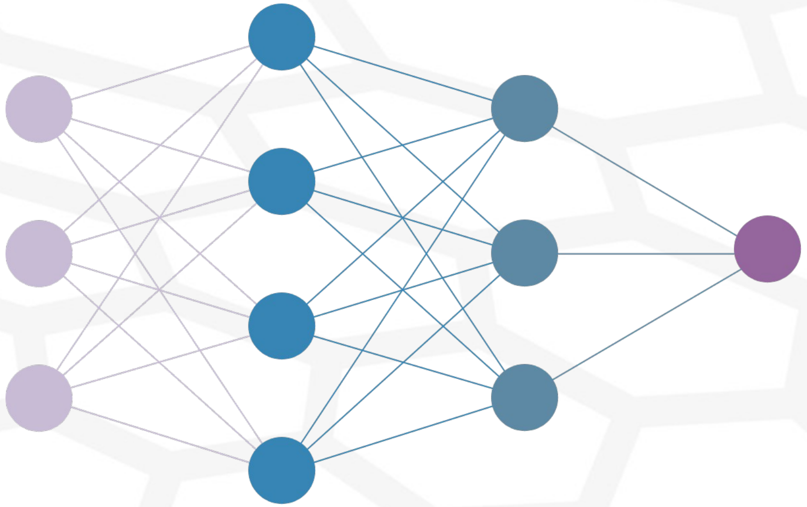
Fitting a NN in Keras

Keras is a Python library that is used to fit neural networks.

The following are the common steps you want to take when fitting a neural network using keras:

1. Load data
2. Define keras model
3. Compile model
4. Fit model
5. Evaluate model
6. Use model for prediction

Define Keras Model - ANN



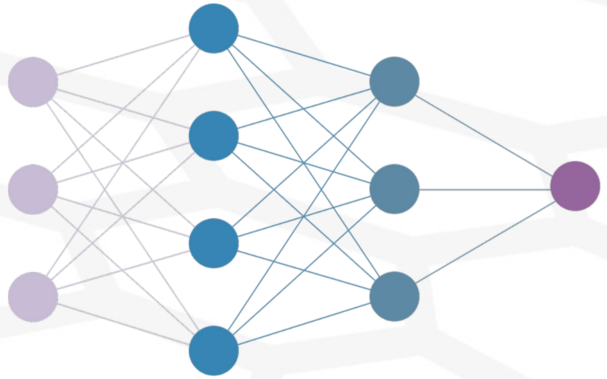
```
model = Sequential()

# Define input layer & first hidden layer
model.add(Dense(4, activation = 'relu', input_dim = 3))

# Add second hidden layer
model.add(Dense(3, activation = 'relu'))

# Define output layer
model.add(Dense(1,activation = 'sigmoid'))
```

Compile & Fit Keras Model - ANN



Optimizer - Algorithm used to minimize the loss function (we will use ‘adam’ for this class).
Loss - Loss function we are trying to minimize. Depends on the type of problem. Functions are listed here.
Metrics - Metric(s) you want to print out.
Epochs - Number of times you want the data to pass through the model.

```
# Using binary cross entropy loss function and accuracy as metric since we are doing a binary prediction
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
# Fit model using training data
model.fit(X_train, y_train, epochs=8)
```

Resources

Deep Lizard tutorials - https://deeplizard.com/learn/video/gZmobeGL0Yg

Neural Network Example - https://playground.tensorflow.org

Keras - https://keras.io/

Tensorflow - https://www.tensorflow.org/