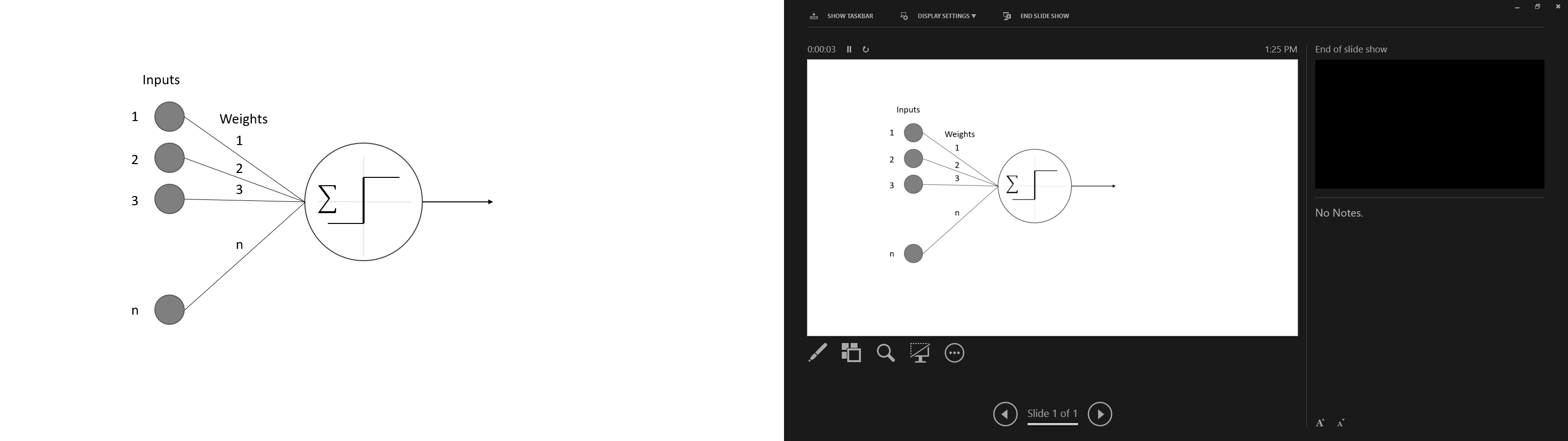
**4.2 Neural Networks**

Neural networks have a long history in the machine learning realm. In 1958 Frank Rosenblatt theorized the perceptron to mathematically model a neuron in the human brain, Figure x.1. In his theory the neuron takes multiple inputs and multiples the input by a weighted value. The neuron then sums all the inputs and weights, if the sum is above a threshold then the neuron is activated or if the sum is below the threshold the neuron stay dormant.

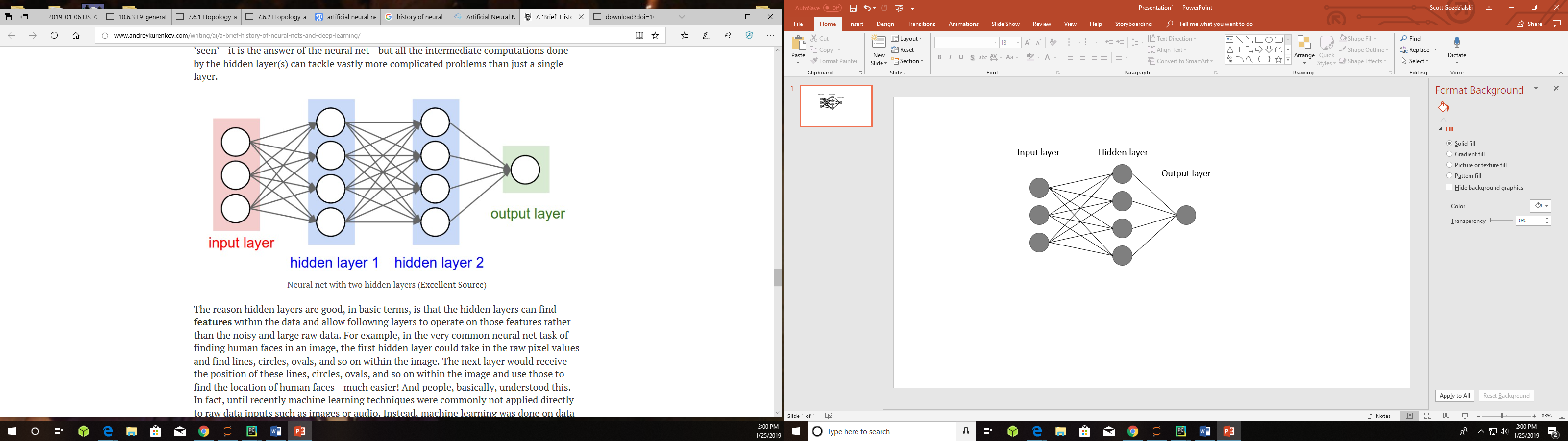
Figure x.1. Perceptron



Over time the perceptron was shown to be able to handle some simple functions, but more complex problems need more than one perceptron to calculate the solution. The next step was to build networks of perceptrons to work in conjunction, call neural networks. The first neural networks we made up of only a single input layer and a single output layer, this was caused by the need for multiple outputs for a simple problem. An example of the would be if you wanted to classify a picture of a vehicle as a truck, a car, or motorcycle; then, you could have multiple inputs of the images, but a single perceptron could only output one response. For this example, we would need three different outputs, one for truck, one for car, and one for motorcycle.

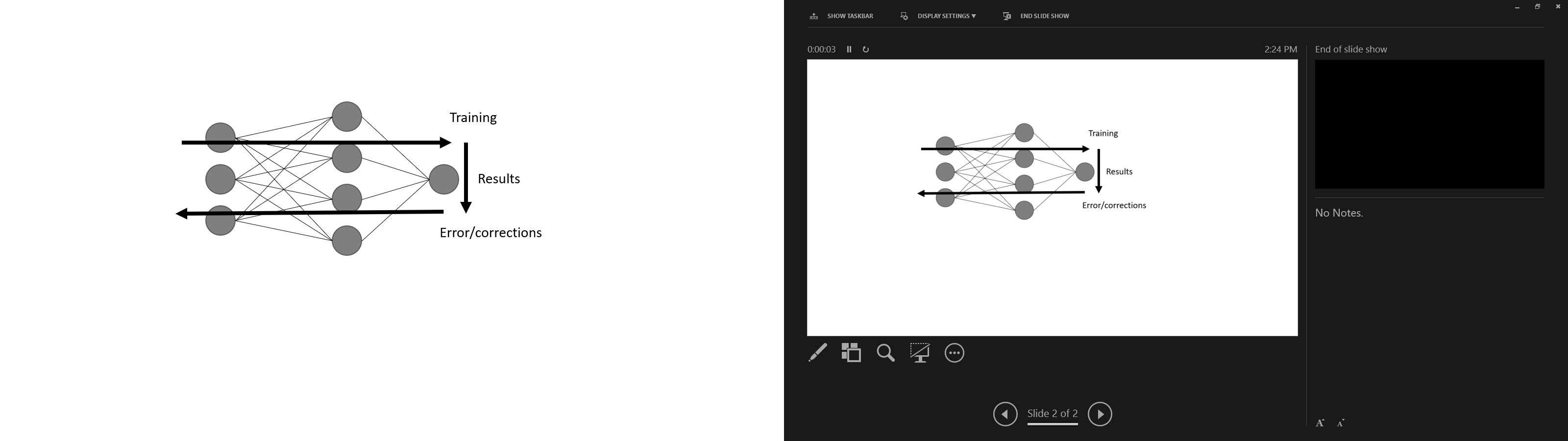
Neural networks continued to become more and more complex as the problems increased in difficulty. We learned that having a single layer of neurons and needed to have hidden layers, Figure x.2. The hidden layers are fully connected to the neurons in the previous layer and the next layer, but they do not have an input or output connection outside of the neural network.

Figure x.2. Hidden Layer



As the layers of neural networks grown to include hidden layers the way we trained the models had to change. Since the output did not have a direct connection to the output layer, we had to find a way to push the corrections to the weights of the connections of layers further back. The way we achieve this is with backpropagation, Figure x.3. As network is trained it outputs the results, the results are calculated for an error. From the error we build a correction to the weights of the neurons. We backpropagate those corrections through the network adjusting the weights until the loss is minimized within the loss function. Giving us the best results.

Figure x.3. Backpropagation



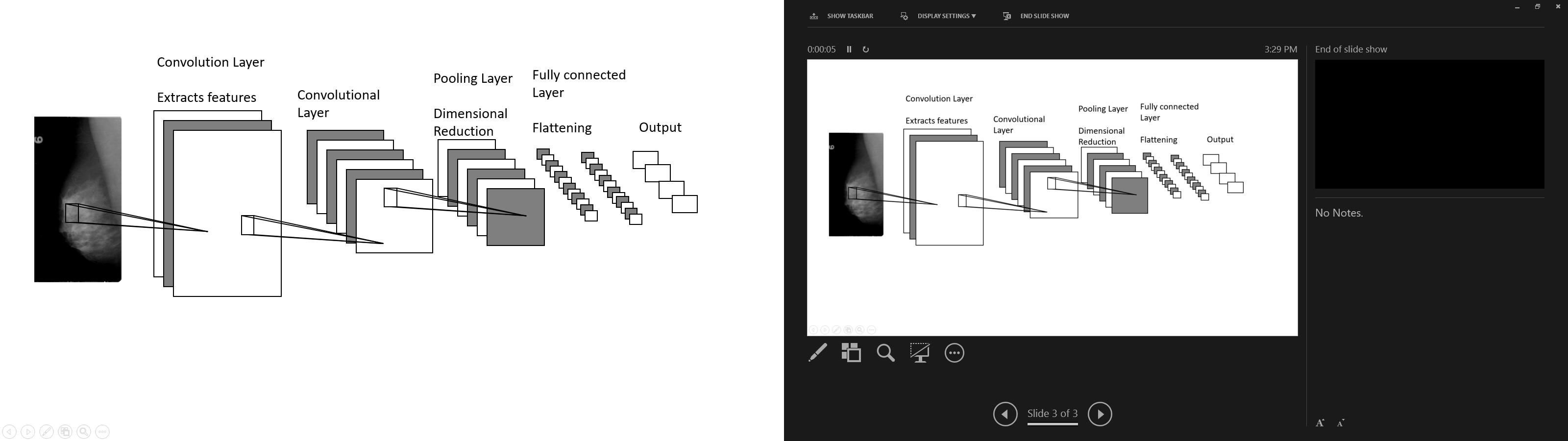
Utilizing this method many problems can be solved by computers.

**4.2.1 Convolutional Neural Networks**

The convolutional neural network (CNN) is a neural network used to identify images. It does this on a high leave by examining the pixel of the image with filters and identifying different features of the image structure. It then compiles the different features extracted and builds a model of the object with in the image. After the model has been properly built an unknown image can be feed into the model and a probability of the object in the image belonging to the different classifications trained in model can be output.

A CNN is made up of different layer performing different functions, Figure x.x.1. The first type of layer is the convolution layer itself, which applies a filter to the image extracting the features from the image. The next type of layer is the pooling layer, the pooling layer reduces the size of the previous layers cutting the computational complexity of the problem. The final type of layer is the fully connected layer which is responsible to transform the outputs from the other layers into the final output.

Figure x.x.1. CNN example



The first type of layer of a CNN is the Convolutional layer. This layer applies a convolutional filter to the image. The filter is used to learn different features and relationships of the image. It will look for structural or gradient on each different channel of the image. The different channels can be three for each color in the red, green, blue image or one on a greyscale image. These filters take the pixels within the filter size and apply a different linear algebra function to them to extract the feature from that section of the image. They then travel around the entire image mapping the entire image. Each of the filters produce different maps of the image.

With the different filters creating the unique maps on each channel of the image we need to reduce the amount of data we are using. The pooling layer of the CCN is used to reduce the size of the data needed to process. The pooling layer completes this by taking a piece of the filtered image and using a function to combine the properties of the piece while maintaining the integrity of the information contained within.

After multiple convolutional and pooling have been applied to the image, we need to send the information to the output layer. Until we have not talked about the dimensions of the CNN, but each of the convolutional and pooling steps change the shape of the information. The original image may be 100x100x3 pixels for the RGB channels, after the first convolutional layer with 10 filters it can be 100x100x3x10. Even with pooling it will not be reduced to a perfect shape to feed the output. The flattening layer is responsible for reducing the response from the previous layers into a single stream of bits. This flatten layer if bit is feed into the output layer to get the probability results.

Each CNN uses a different number of each layer in different orders to achieve their results.

**4.2.2 Object Detection**

Utilizing the CNN built above we can develop a way to detect objects within an image and highlights it location. There are three main ways to do this sliding window, "You Only Look Once, YOLO" and the "Single Shot Multiple Box Detector." The different way to detect an object have some similarities and a few differences, but fundamentality they all work the same. They all work by first, dividing the image into smaller sub-images. Then they, run the image classification model we described above on the sub-images. A probability is retrieved from the image classifier and recorded with the location within the image. If the probability is above a certain threshold, then that section of the image is highlighted in some fashion.

Reference

<http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/s>

This has stuff about Perceptrons and NN and backpropagation.

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Survey of Convolutional Neural Networks -- has no URL

CNN

<http://www.cs.unc.edu/~wliu/papers/ssd.pdf>

SSD: Single Shot multibox detector