**Slide 1 [Title]**

Good morning. My name is Dylan Siegler and my Strnad project is on using medical literature to improve neural network diagnoses.

**Slide 2 [What is a Neural Network]**

My project revolves around a type of computer program called a neural network. Invented in the 1950s, initially to try to model how human brains think, the neural network has exploded in popularity in the past ten years. There are many variations of neural networks, but the most basic networks consist of an input layer, hidden layers, and an output layer. Each yellow circle, called a neuron, holds a number in it. Each black arrow represents a connection between two neurons, and while passing through this connection numbers are multiplied by the weight associated with each individual connection. Data is put into the neurons in the input layer in the form of numbers and the data is then propagated through the network. After some calculations, the “answer” can be found in the output layer, again in number form.

**Slide 3 [Example Network]**

This is a concrete example of a neural network. The goal of the network is to classify if an image is of a cat or a dog. The input would be the color value of every pixel of a given image (in reality the network would be way taller than shown in the picture). After these values go through the network, the output layer then contains whether the network thinks the input image is a cat or a dog.

**Slide 4 [How is a Neural Network “Trained”]**

The process I just described is fairly vague and begs the question: how does the neural network “know” what a cat or a dog looks like. This knowledge comes from the training process. If you recall, each connection between neurons has a weight associated with it; during training, these weights are slowly adjusted in order to make the network more accurate by looking at one example at a time, seeing what changes would make the network correct for that example, move the network slightly in that direction, then repeating many times. At the bottom of the screen you can see a graph of a training run from one of my neural networks; as time on the x-axis goes on, the loss, or error, decreases, approaching zero. This means that the training process relies on a large amount of data; this is why neural networks have become so prominent recently. Additionally, this training process is exceptionally robust, meaning that neural networks are able to tackle many problems with very good results.

**Slide 5 [Project Goals]**

I had two main goals for this project. First, I wanted to investigate a type of neural network that has largely been unexplored – a network that is capable of taking in two completely different data types, as opposed to traditional networks which take in data all of the same type. Second, I wanted to test if this type of neural network could at all help in predicting diagnoses for diseases. There is extensive research on using neural networks to provide diagnoses for diseases; however, none of them integrate medical research directly, rather relying purely on patient data to try to predict a diagnosis.

**Slide 6 [What Neural Networks am I Using?]**

In my project, I am using two types of neural networks. First, I am using an autoencoder network to convert the text in medical literature to numerical form. The output of this first network, along with patient data (information such as blood pressure, cell density, etc.) will be fed into a classification network that looks very similar to the dog and cat network previously shown.

**Slide 7 [The Autoencoder]**

This is a picture of an autoencoder neural network. It looks similar to the traditional neural network, again with neurons and connections, but with one important difference: the shape. The “bottleneck” in the middle of the network forces it to compress the input data. The “goal” of an autoencoder is simply to recover the input. This means that the network is forced to come up with the most efficient way to store the input data in the small amount of space given to it in that bottleneck area it must pass through.

**Slide 8 [Why Is Such a Complex Algorithm Used to Convert Text to Numbers]**

There are many ways to convert text to numbers, but the autoencoder has some unexpected benefits. The autoencoder is able to pick up on the semantics, or meanings of words. This means that, when fully trained, the autoencoder knows that, for example, “well” and “good” have similar meanings, or “Cleveland” and “Pittsburg” also have similar meanings, and thus the network generates very similar outputs for these respective pairs of words. This interesting property is not explicitly coded in, but rather it arises due to the statistical nature of the training process and the relatively predictable patterns found within language.

**Slide 9 [Result of the Autoencoder]**

The output of the autoencoder can be thought of specially, with the autoencoder assigning every word to a point in space. Similar words such as “good” and “well” would be close in this space. The output of my autoencoder can be seen here in 3D space. Each of the thousands of points in space represents a word. I have highlighted the word “mathematical” and all words that are “close” to it in space. You can see that, logically, words such “calculus,” “logic,” and “theoretical” are closest to “mathematical” in space. It may appear that these aren’t the closest points, but this is due to the processing done to visualize the output.

**Slide 10 [The Classification Network]**

The second half of my project is a classification network. This network looks and acts in an almost identical manner to the dog and cat network shown earlier with only one difference: the input. Instead of taking in a picture as an input, this network takes in two different types of inputs: patient data as well as the output from the autoencoder, which is the encoded medical literature. The neural network would not be able to handle these two different types of data well, so I placed a small neural network between the autoencoder’s output and the input to the classification network. This mini network learns to make the data compatible, thus allowing the classification network to train like any other network.

**Slide 11 [Results and Conclusion]**

In order to see if the addition of medical literature helped the neural network, I first ran the classification network without any medical literature. When just using the patient data, the network had an accuracy of 56%. However, when run with the medical literature the network achieved an accuracy of 59%. Although both of these numbers are low compared to the state of the art, around 80-90% accuracy, what is most notable is the increase when adding medical literature to the network. These results show potential for this method, and may offer a way in which state of the art networks could improve their accuracy further.

**Slide 12 [Thank you to…]**

I would like to extend a huge thank you to my parents, Mrs. Axelrod, Mr. Sweeney, Dr. Laux, and Dr. Dyke as well as the Strnad family for providing me with this opportunity and helping me at every step of the process. Thank you and good morning.

**Introduce Maheep**