

- Neural Networks + Machine learning becoming extremely important and there are many types
 - Convolutional Neural Network \rightarrow Images
 - Long short-term memory Network \rightarrow Speech

- Plain Vanilla (aka "multi layer perceptron")

Neurons

- Things that hold 0 or 1
- In the image example, the number in neuron contains activation number
- For output, higher number activations means more sure accuracy

Layers

- In a perfect world, you could break down some number in the image to sub components in the second to last layer before output.
- But how could you even recognize the components?
 - Recognizing one sub component can break down even further. in the example, edges in a drawn number.
- Being able to recognize patterns break into layers of abstractions
- What parameters does it need to recognize a pattern?
 - Use weights (numbers) and combine them with each numbers activation in a region
 - We want activations between 0-1
 - Squeezing function (Sigmoid)
 - How positive is the weighted sum? (0-1)

• You also want some bias before sigmoid function (one for each neuron)

• 13002 different ^(weights/biases) behaviors or settings in the number example

• 'learning': getting computer to find a valid setting

• Notation

$$\text{sigmoid}(\text{weight} \cdot \text{activation} + \text{bias}) = \text{result?}$$

$$\left(\begin{bmatrix} w_0 & w_{0,1} \dots \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \begin{bmatrix} ? \\ ? \end{bmatrix} \right)$$

• More accurate to see neurons as function, and system as function.