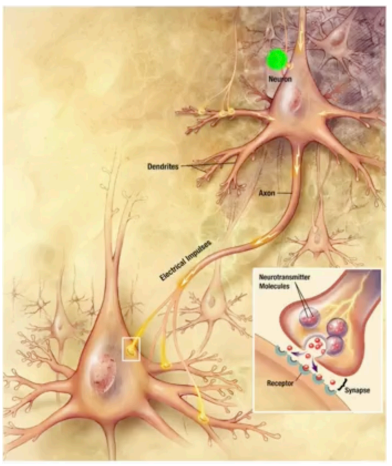
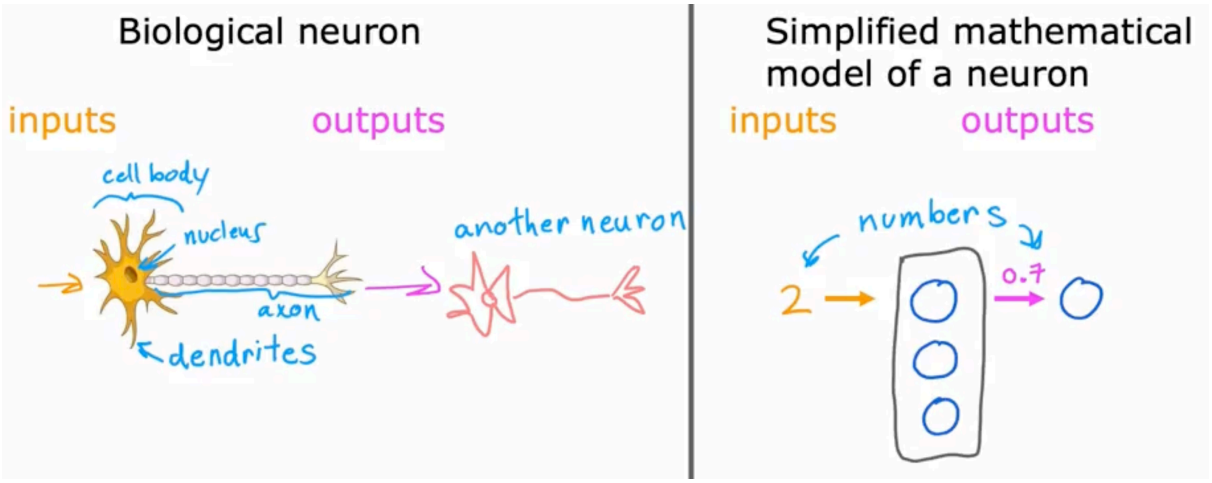
Neural networks intuition

Neural networks were initially designed to mimic the learning processes of the human brain.

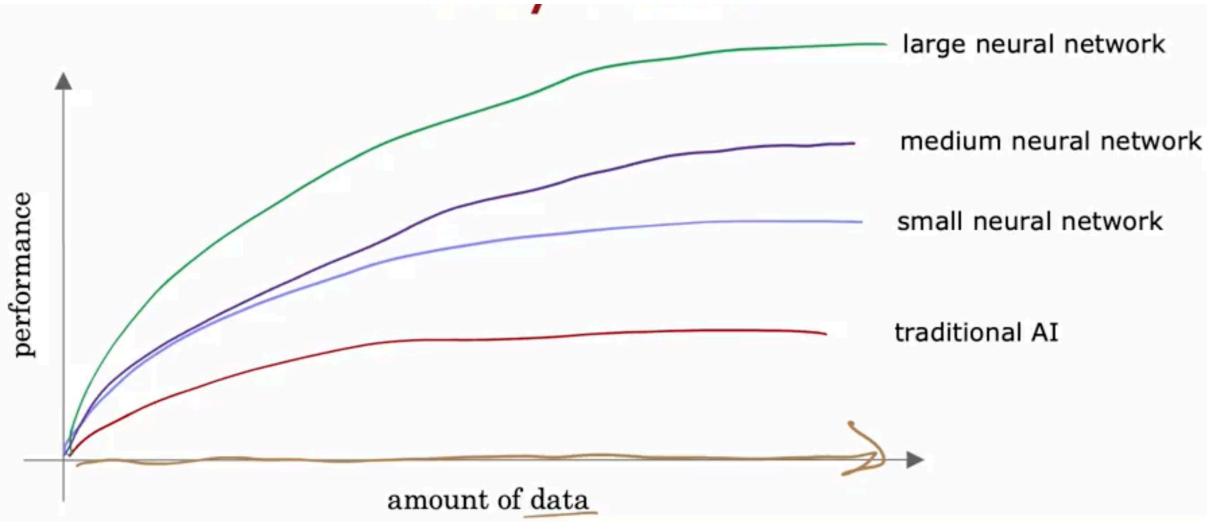
Neuron and the brain

Neurons in the brain





Why now?



Deman Prediction

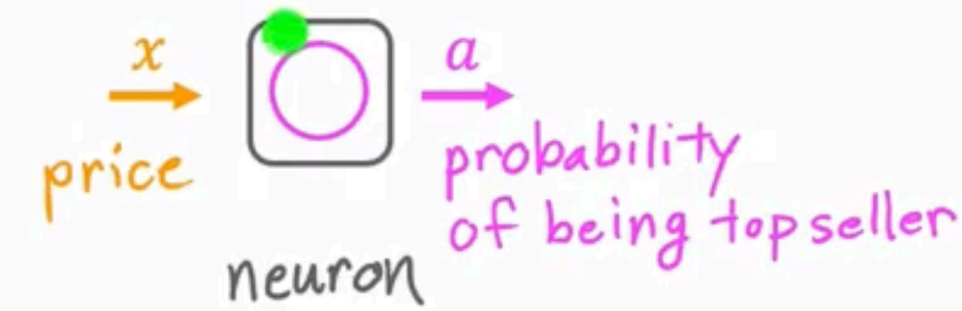
Example

Top seller? ⇒ Yes/No

Using sigmoid function to predict as logistic regression. That's logistic regression's formula is called: **activation**

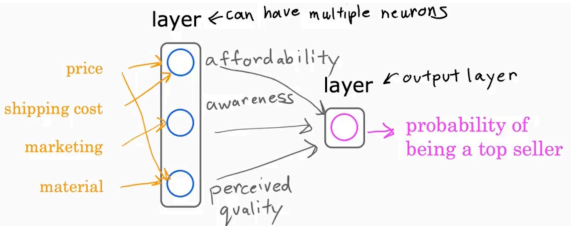
$$a = f(x) = \frac{1}{1 + e^{-(wx+b)}}$$

activation



Complex "Top seller?"

- A more complex model includes multiple features: price, shipping costs, marketing, and material.
- “Affordability, awareness, perceived quality” are also called **“activation”** of the 4 features.
- And the output is also **“activation”** of 3 above activations



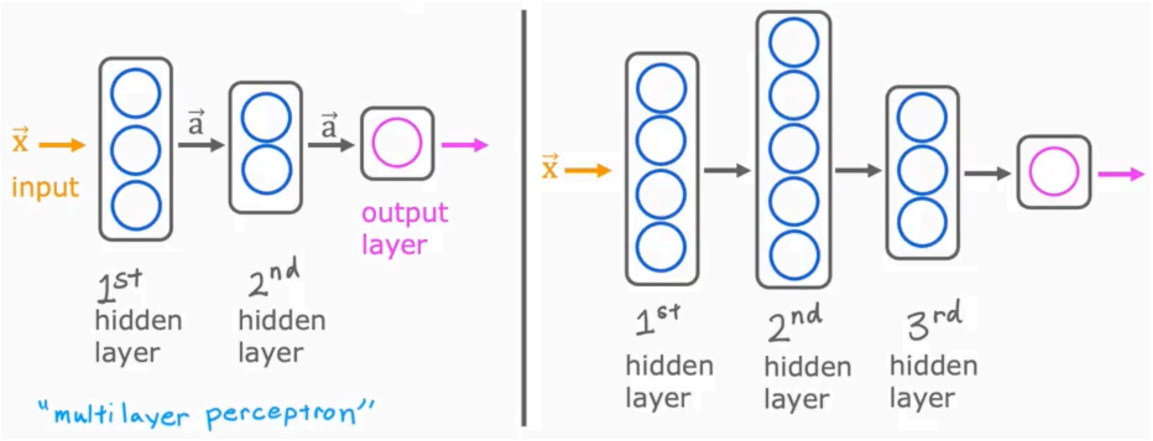
- Neurons are grouped into layers, with the input layer receiving features and the output layer providing predictions

The neuron network uses 4 nums (to computes) ⇒ 3 nums (activation values) ⇒ 1 nums (activation value).

- The list of 4 nums is called input layer ($\vec{x}$)
- The middle layer, called the hidden layer, processes inputs to generate activations, which are used for final predictions. ($\vec{a}$)

Architecture of Neural Networks

- The architecture involves deciding the number of hidden layers and neurons per layer, impacting performance.
- The term "multilayer perceptron" refers to neural networks with multiple layers, highlighting their complexity and capability



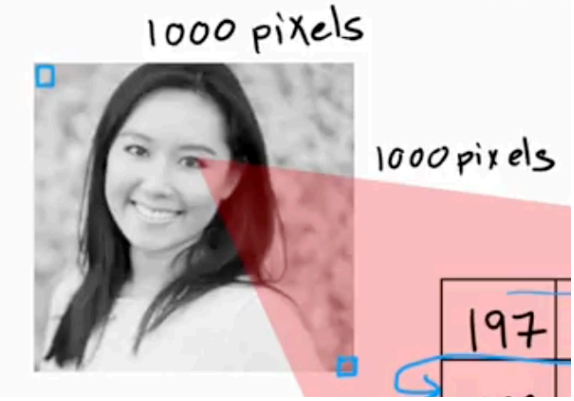
Multilayer perceptron

💡 Manually specifying how nodes in each layer should connect is **challenging** in practice. That's why neural networks start by linking every node to all nodes in the next layer. Their true power lies in learning the best connections automatically. Each layer builds on the patterns identified by earlier layers, allowing the network to convert basic inputs (like image pixels) into sophisticated representations (such as human faces). As data moves through the layers, the activations capture progressively more abstract and complex features.

Example: Recognizing Images

Face Recognition

- A neural network can be trained to take an image as input and output the identity of the person in the image.
- The image is represented as a matrix of pixel intensity values, which can be unrolled into a vector of one million values for a 1,000 by 1,000 pixel image.



(**hint:** python `numpy.reshape()`)

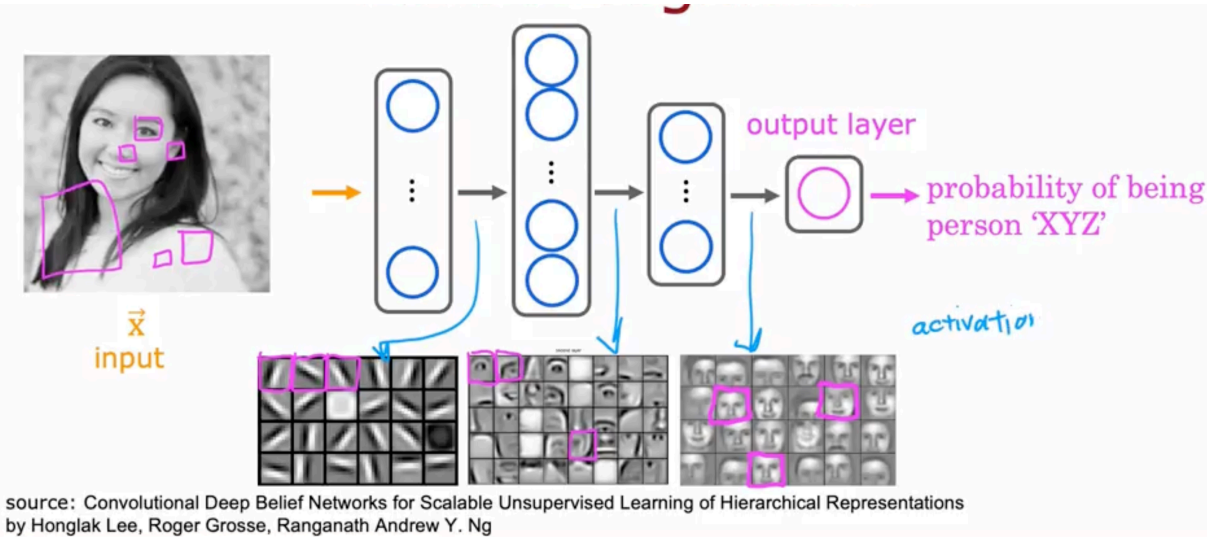
Layered Structure of Neural Networks

- The first hidden layer extracts basic features like edges, while subsequent layers combine these features to identify parts of faces, such as eyes and noses.
- The final output layer estimates the probability of the image corresponding to a specific person based on the features detected.

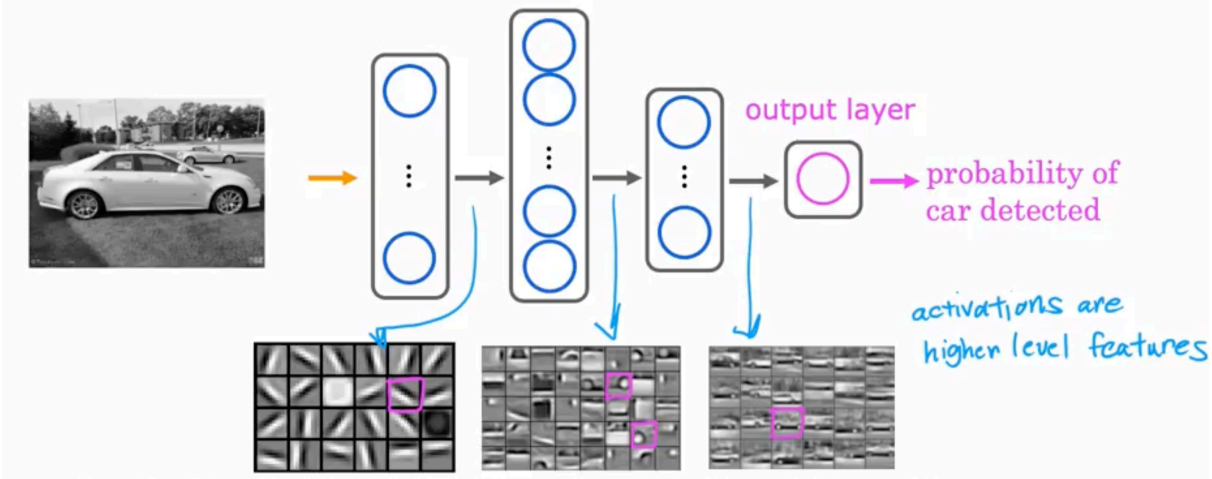
1. **Input Processing**:

- The neural network takes an image as input, represented as a matrix of pixel intensity values.

2. **Feature Extraction**:
   - The first hidden layers identify basic features, such as edges and lines.
   - Subsequent layers combine these features to detect more complex patterns, like facial parts (eyes, nose, mouth).
3. **Pattern Recognition**:
   - The network aggregates the detected features to recognize the overall structure of a face.
   - It learns to differentiate between different faces by analyzing the unique combinations of features.
4. **Output Generation**:
   - The final output layer estimates the probability of the input image corresponding to a specific identity based on the learned features.



Car Classification Example:



source: Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations by Honglak Lee, Roger Grosse, Ranganath Andrew Y. Ng

💡

1. The neural network learns to identify features autonomously from the training data, without explicit instructions on what to look for.
2. Different datasets (e.g., faces vs. cars) lead the network to learn different features, demonstrating its adaptability to various tasks.