"From 2016 to 2020, the entire machine learning and data science industry has been dominated by two approaches: deep learning and gradient boosted trees. Specifically, gradient boosted trees is used for problems where structured data is available, whereas deep learning is used for perceptual problems such as image classification. . . . These are the two techniques you should be most familiar with in order to be successful in applied machine learning today"

F. Chollet [2021]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

Y. LeCun, Y. Bengio, G. Hinton [2015]

Neural Networks and Deep Learning

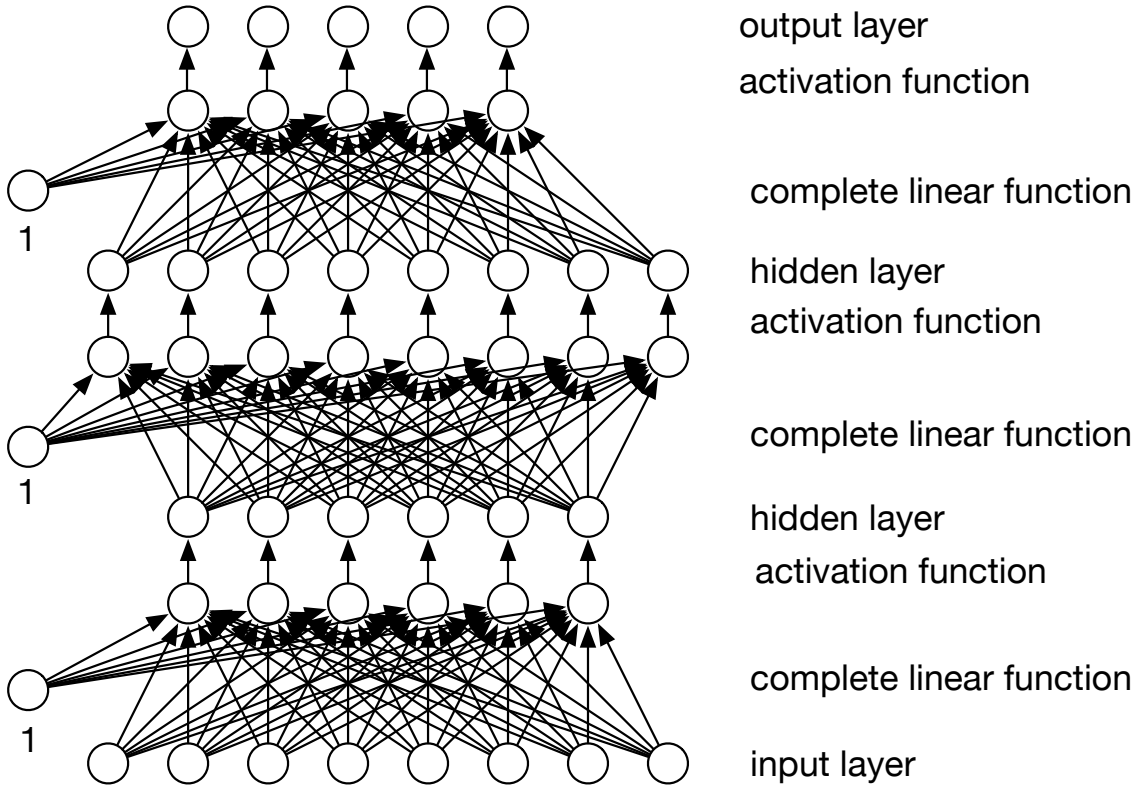
- Where do the features for linear or logistic regression come from? Either:
  — human engineered
  — learned
- Artificial neural networks have had considerable success in unstructured and perception tasks for which there is abundant training data such as for image interpretation, speech recognition, machine translation and, game playing.
- These representations are inspired by neurons and their connections in the brain.
- Artificial neurons, or units, have inputs, and an output. An output can be connected to the inputs of other units. These are *much* simpler than animal neurons.
- A unit is a parameterized non-linear function of its inputs.
- Learning occurs by adjusting parameters to fit data.
- Neural networks can approximate to any discrete or continuous function.

Why Neural Networks?

- As part of neuroscience, in order to understand real neural systems, researchers are simulating the neural systems of simple animals such as worms.
- It seems reasonable to try to build the functionality of the brain via the mechanism of the brain (suitably abstracted).
- The brain inspires new ways to think about computation.
- Neural networks provide a different measure of simplicity as a learning bias.

Feed-forward neural networks

- Feed-forward neural networks are directed acyclic graphs:



- Each hidden unit outputs a linear function of its inputs followed by a non-linear activation function.

Feed-forward neural networks

- A feed-forward neural network implements function

$$f(x) = f_n(f_{n-1}(\ldots f_2(f_1(x))))$$

- x is a vector of input values (the input layer)
- Each function f_i maps a vector into a vector.
- Each component of an output vector is called a unit.
- Function f_i is the ith layer.
- The last layer, f_n, is the output layer.
- The other layers are called hidden layers.
- The number of functions, n, is the depth of the network.
- "Deep" in deep learning refers to the depth of the network.

Feed-forward neural networks

Each layer f_i is

- a linear function with learnable parameters of each output given the input (similar to linear or logistic regression)
- followed by a non-linear activation function, ϕ.
- The linear function takes a vector *in* and an extra constant input with value "1", and returns a vector *out*:

$$out[j] = \phi(\sum_k in[k] * w[k,j])$$

  for a 2-dimensional array w of weights.
- The weight associated with the extra 1 input is the bias.
- There is a weight $w[i,j]$ for each input–output pair of the layer, plus a bias for each output.
- ϕ is an activation function
- The outputs of one layer are the inputs to the next.

Activation function: ReLU

- A common activation function is the rectified linear unit (ReLU):

$$phi(x) = max(0, x)$$

or

$$\phi(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

- The derivative of ϕ is

$$\frac{\partial \phi}{\partial x}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

Activation function for output

The activation function and what is being optimized depends on the type of the outputs:

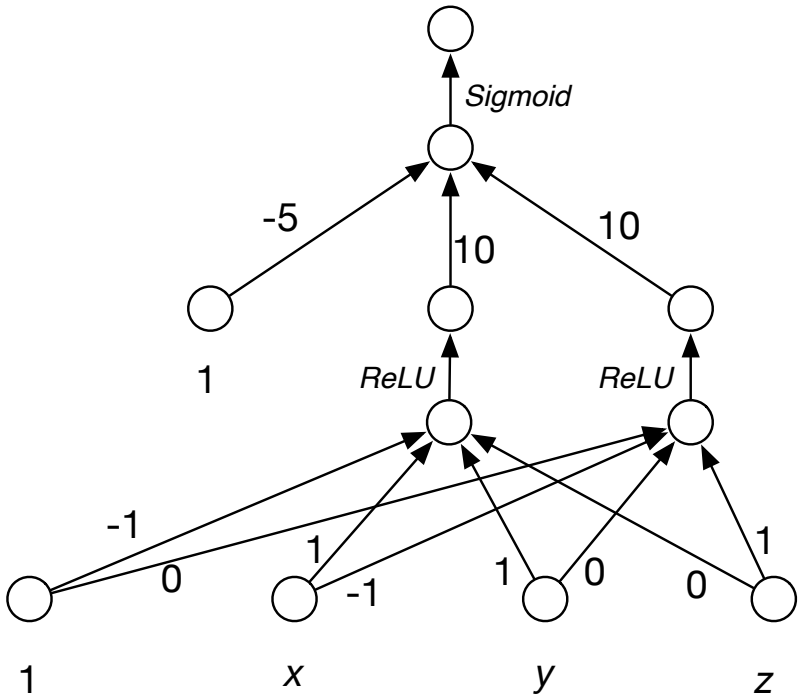
- If output is real, optimize squared loss, and use the identity function: $\phi(x) = x$
- If output is Boolean, use binary log loss, with a sigmoid (as in logistic regression):

$$\phi(x) = sigmoid(x) = \frac{1}{1 + \exp(-x)}$$

- If output y is categorical, but not binary, use categorical log loss with a softmax. The output layer has one unit for each value in the domain of y.

What can a neural network represent?

The function "if x then y else z" cannot be represented using logistic regression. It can be approximated with the neural network:



The function can be represented as $(x \wedge y) \vee (\neg x \wedge z)$

Neural network inputs

- The input of a neural network is a vector of real numbers.
- Boolean variables are represented using 1 for true and either 0 or -1 for false.
- Categorical variables can be represented using indicator variables – a binary variable for each value – forming a one-hot encoding

Neural network properties

- The depth of a neural network is the number of layers.
- The width of a layer is the number of elements in the vector output of the layer.
- The width of a neural network is the maximum width over all layers.
- The size of the output and input are usually specified as part of the problem definition.