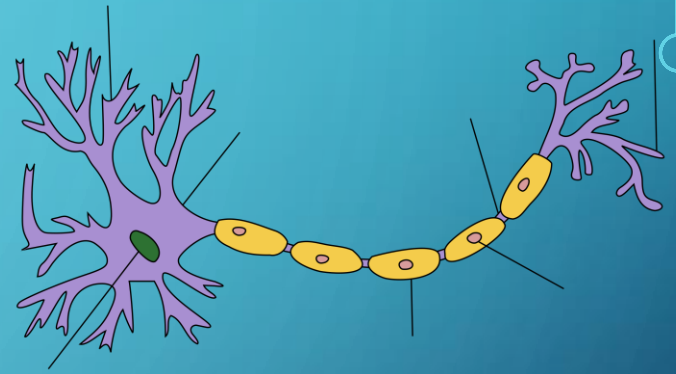
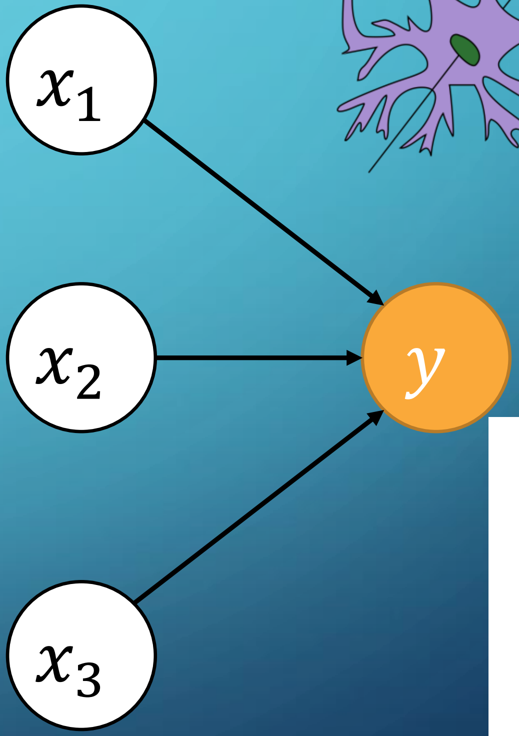
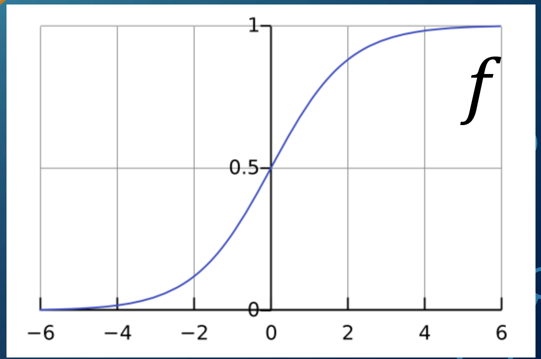
ARTIFICIAL NEURAL NETWORKS

Neurons take in multiple inputs and give an output **activation**

1. Multiply each input by a weight
2. Add the weighted inputs with a bias term
3. Apply a non-linear activation function to the sum







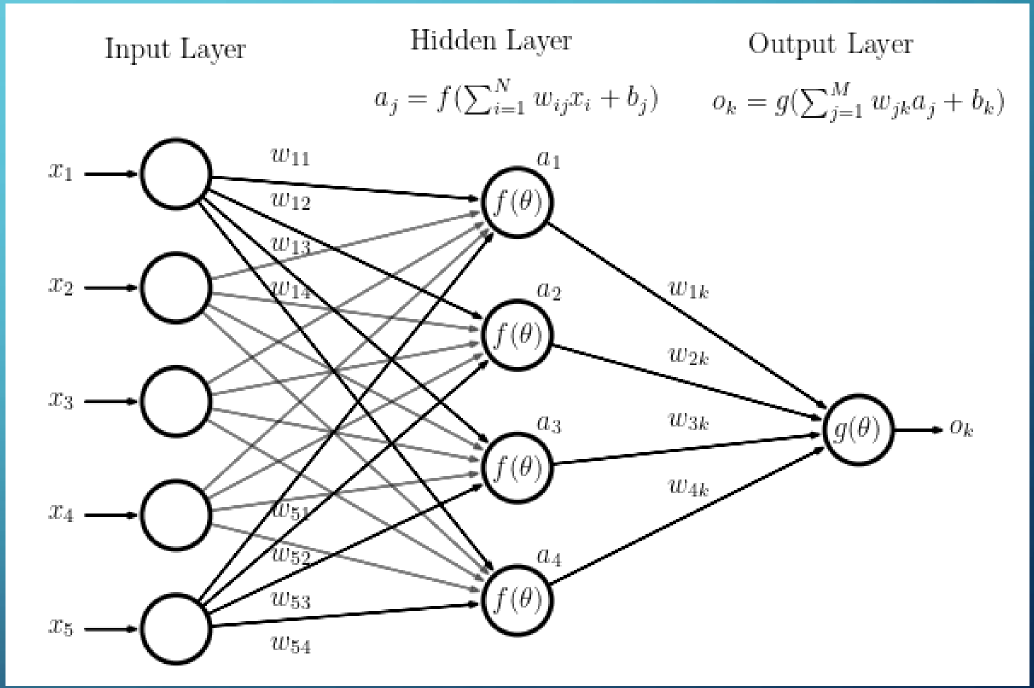
$$y = f(x_1 w_1 + x_2 w_2 + x_3 w_3 + b)$$

ARTIFICIAL NEURAL NETWORKS

Neurons in a **layer** take in the same inputs but can apply different weights and biases

If it is wide enough, a network with one hidden layer can approximate any function of the inputs

The ultimate in flexible models?

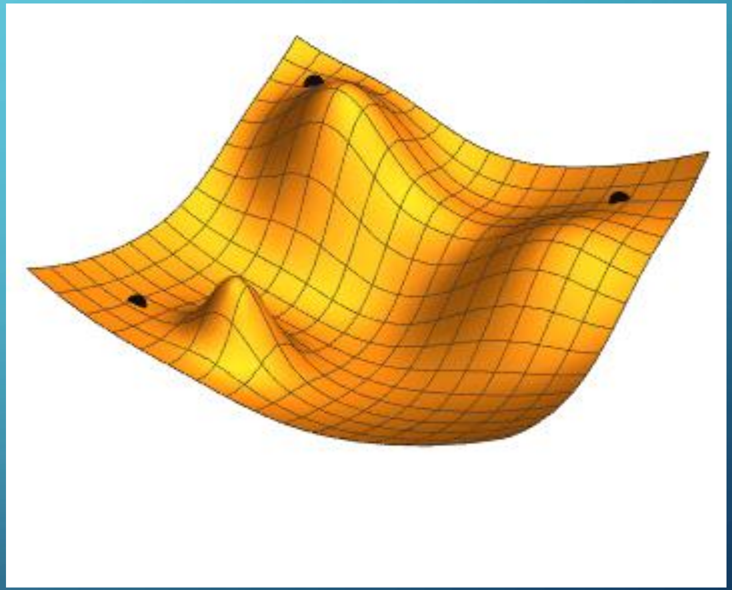


GRADIENT DESCENT

Need a way to optimize the weights and biases: **stochastic gradient descent (SGD)**

Gradient descent takes steps in the direction of the gradient of the loss function

SGD adds some stochasticity, avoiding local minima



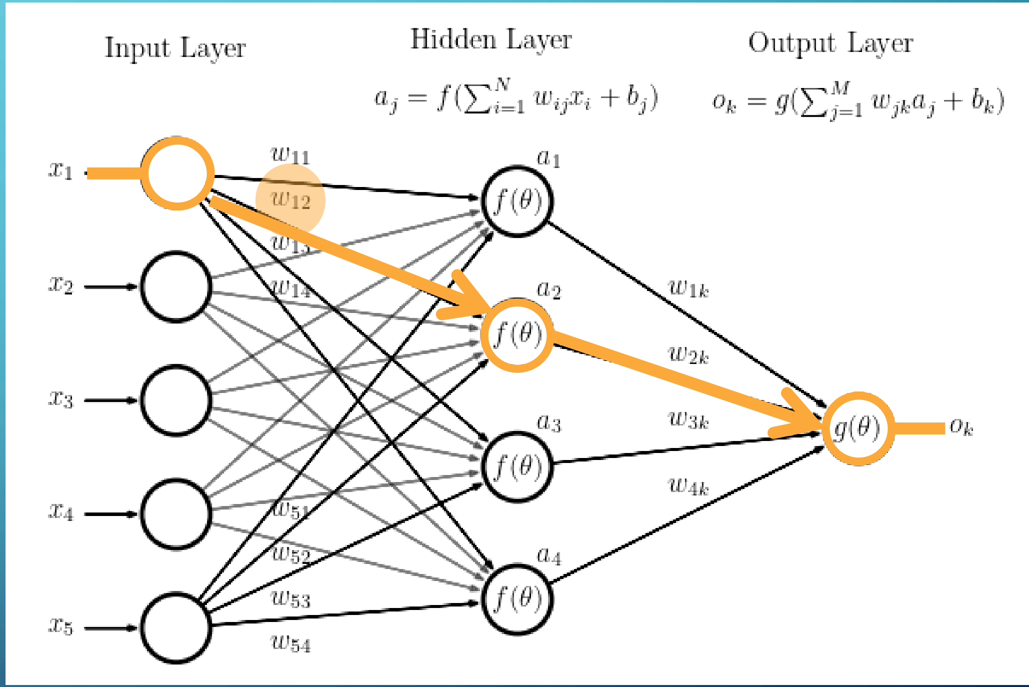
BACKPROPAGATION

Calculating gradients can be done by traversing network backwards: **backpropagation**

$$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial o_k} \cdot \frac{\partial o_k}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_{12}} = \frac{\partial L}{\partial o_k} \cdot g' \cdot w_{2k} \cdot f' \cdot x_1$$

The ReLU activation function makes derivatives really easy

$$\mathrm{ReLU}(x) = \max(0, x)$$



PHOTOMETRIC REDSHIFTS

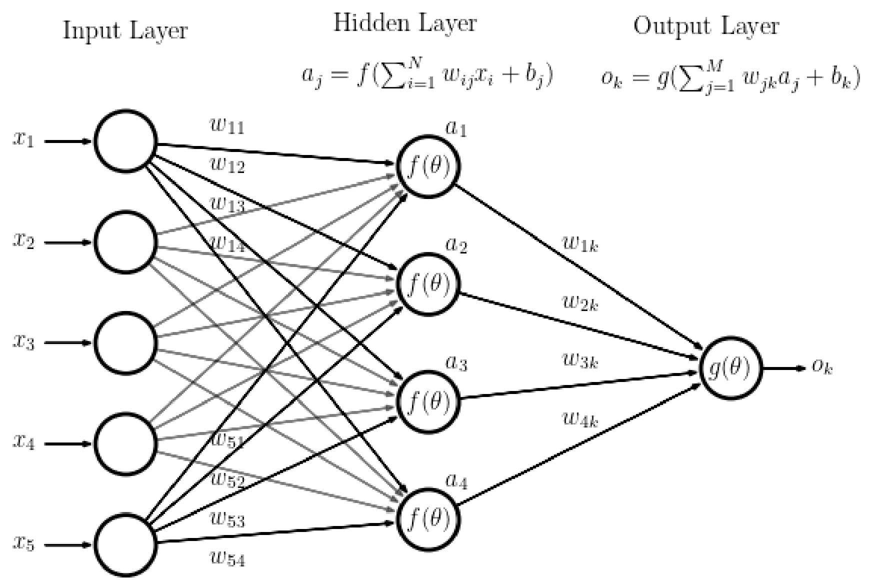
features: *ugriz* photometry

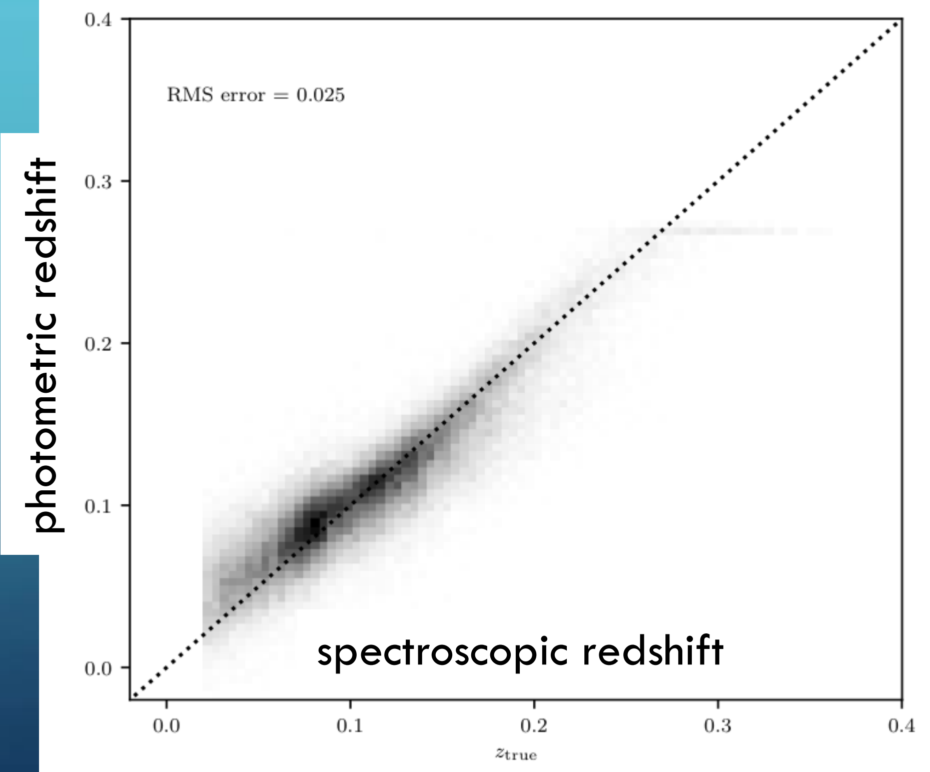
target: spectroscopic redshift

loss function: mean squared error

model: neural network

optimization method: SGD





NEURAL NETWORK HYPERPARAMETERS

Number of layers and their widths

Activation functions

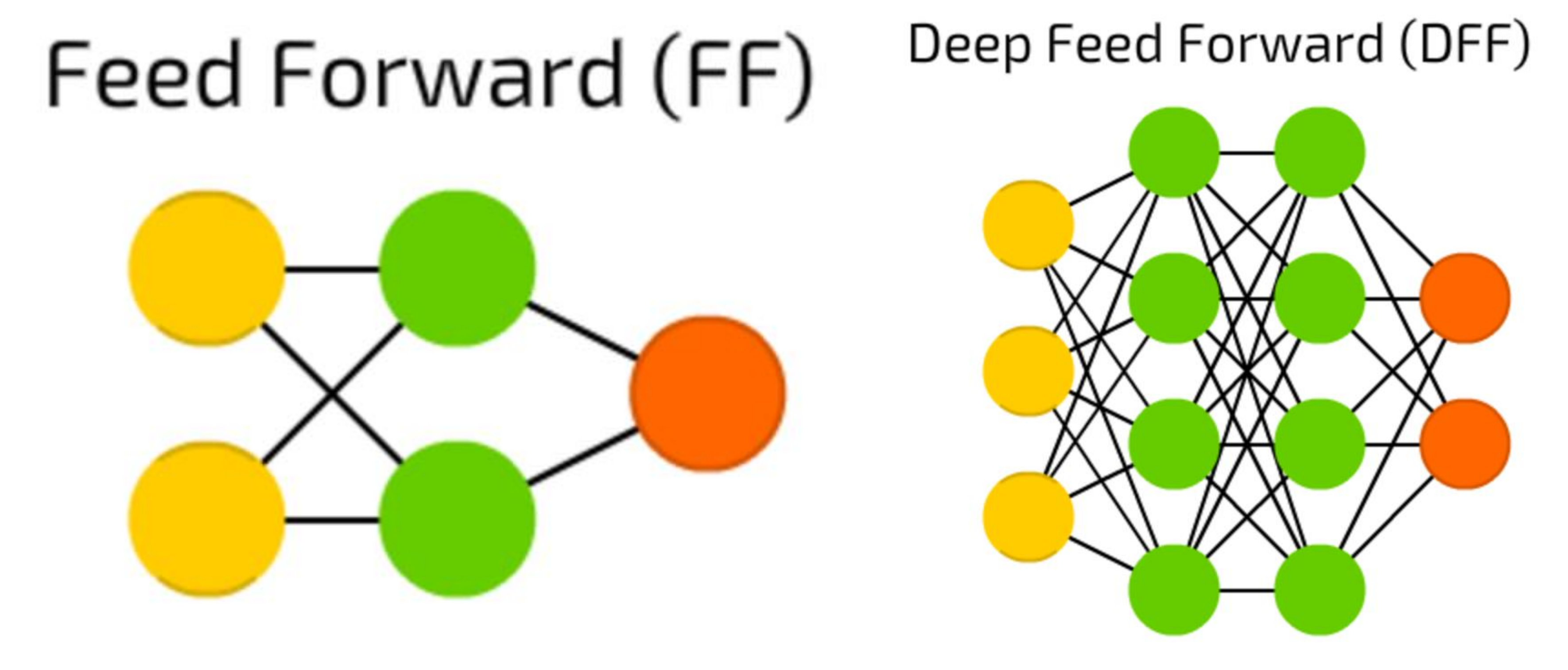
Optimizer and its hyperparameters

When to stop training

Random search – train many times, trying random values for the hyperparameters each time!

DEEP LEARNING

A deep neural network has more than one hidden layer

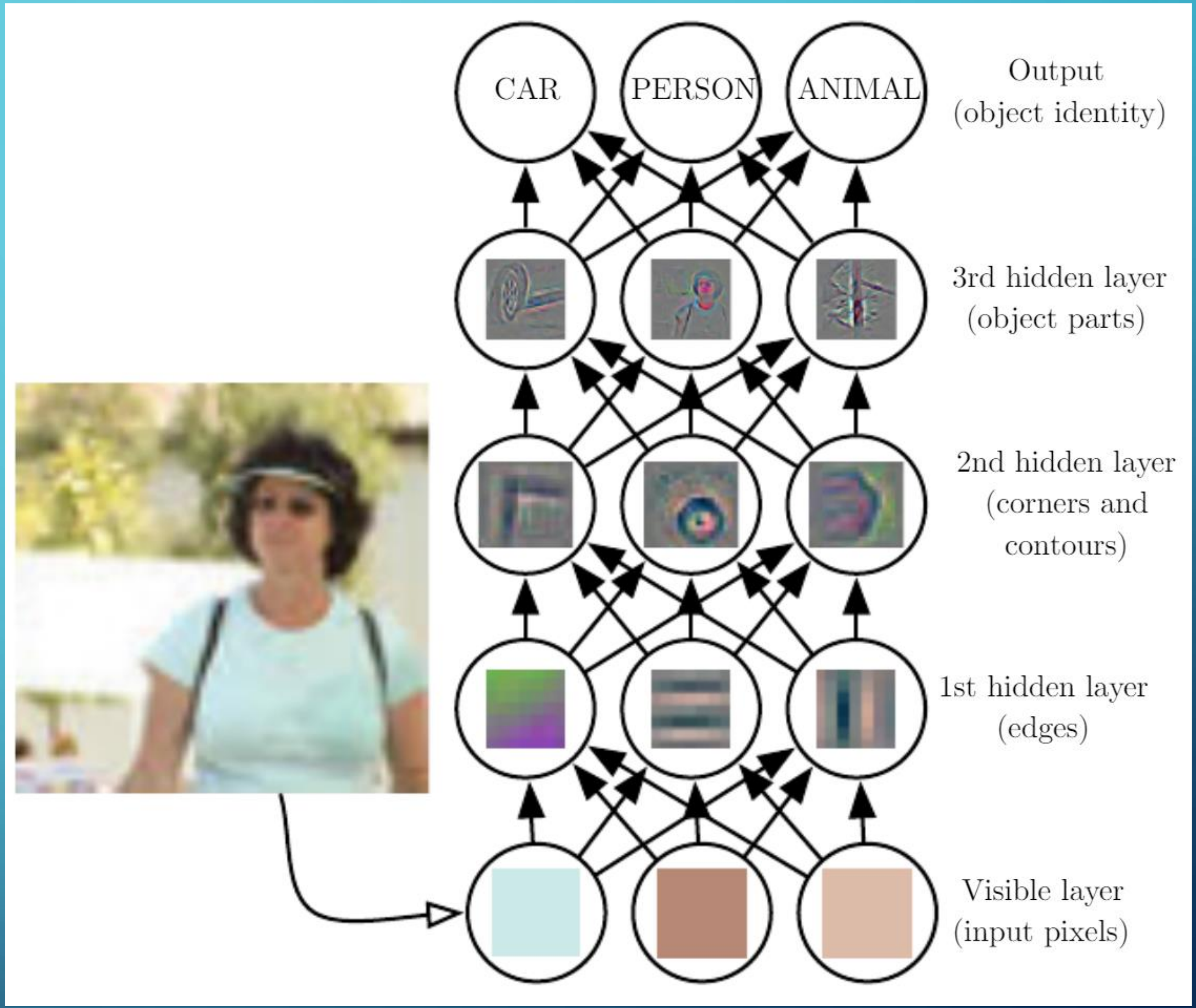


Neural Network Zoo, Fjodor van Veen

DEEP LEARNING

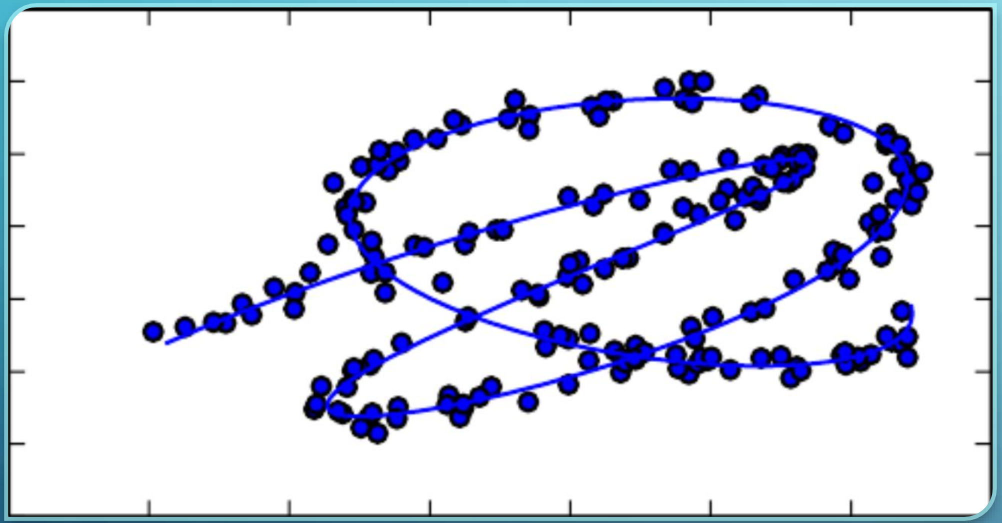
Each hidden layer extracts increasingly abstract information

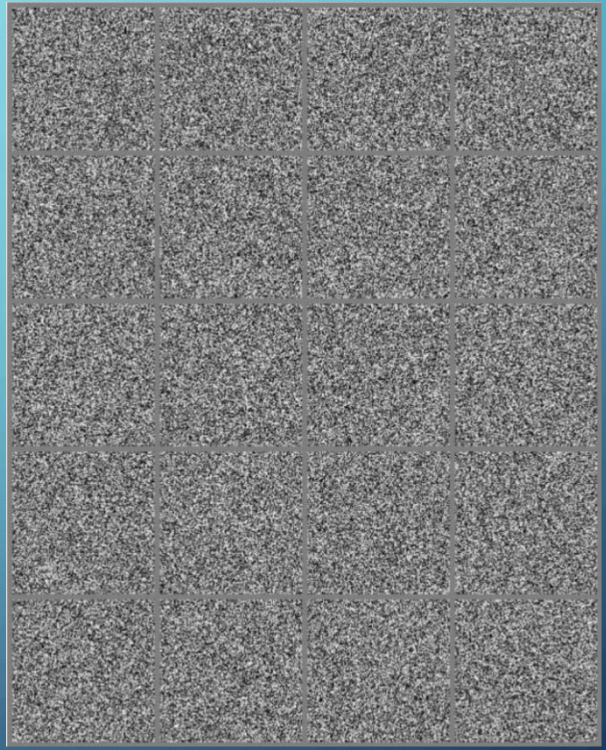
Don't need to hand-craft features – the network will make its own



Goodfellow, Bengio, and Courville (2016)
Zeiler and Fergus (2014)

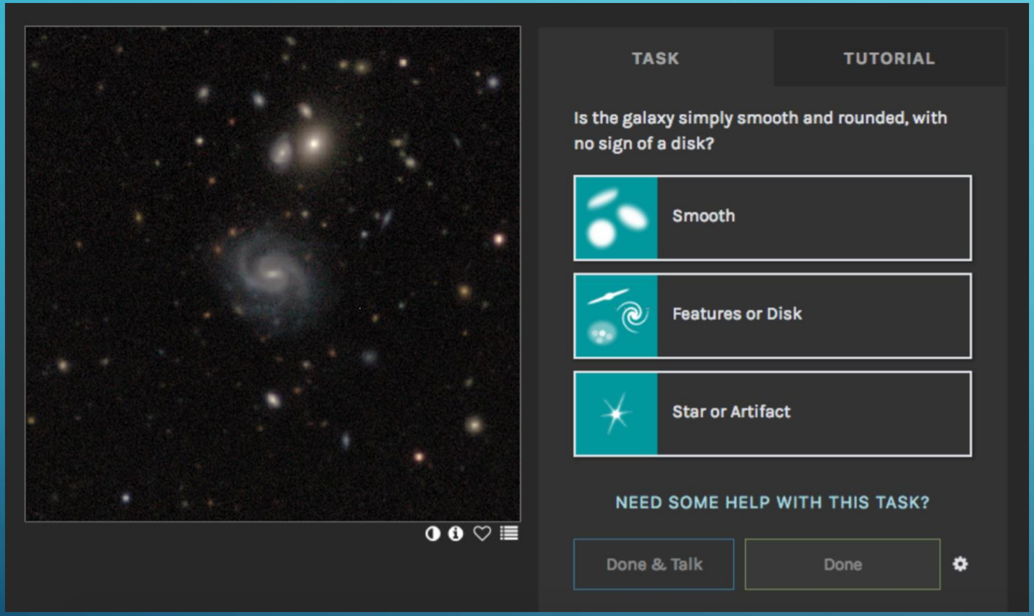
MANIFOLD HYPOTHESIS

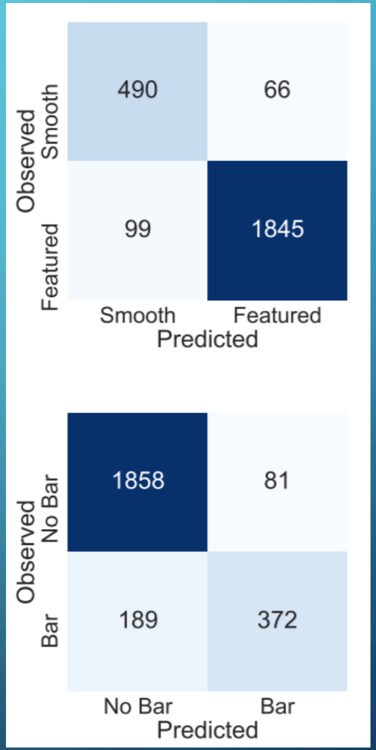




Goodfellow, Bengio, and Courville (2016)

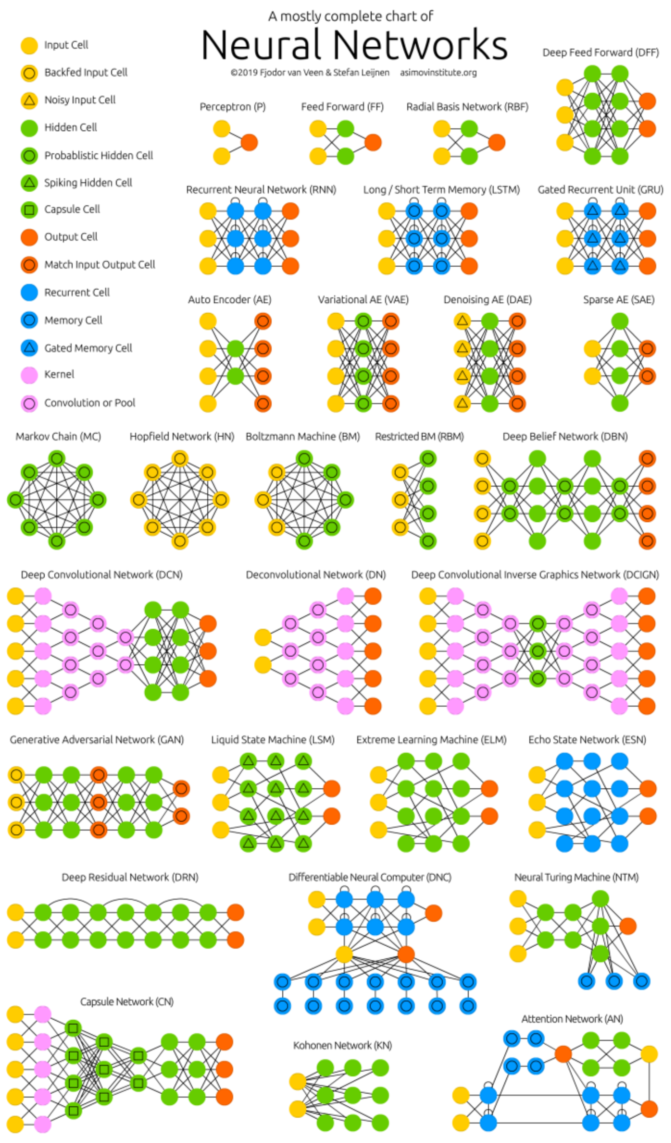
GALAXY ZOO





Walmsley et al. (2019)

THE NEURAL NETWORK ZOO



Neural Network Zoo, Fjodor van Veen

APPLICATIONS OF MACHINE LEARNING

1. Where a human does well, but rules are hard to codify

2. Datasets with complex correlations that are hard for humans to handle

In either case, lots of data is needed!