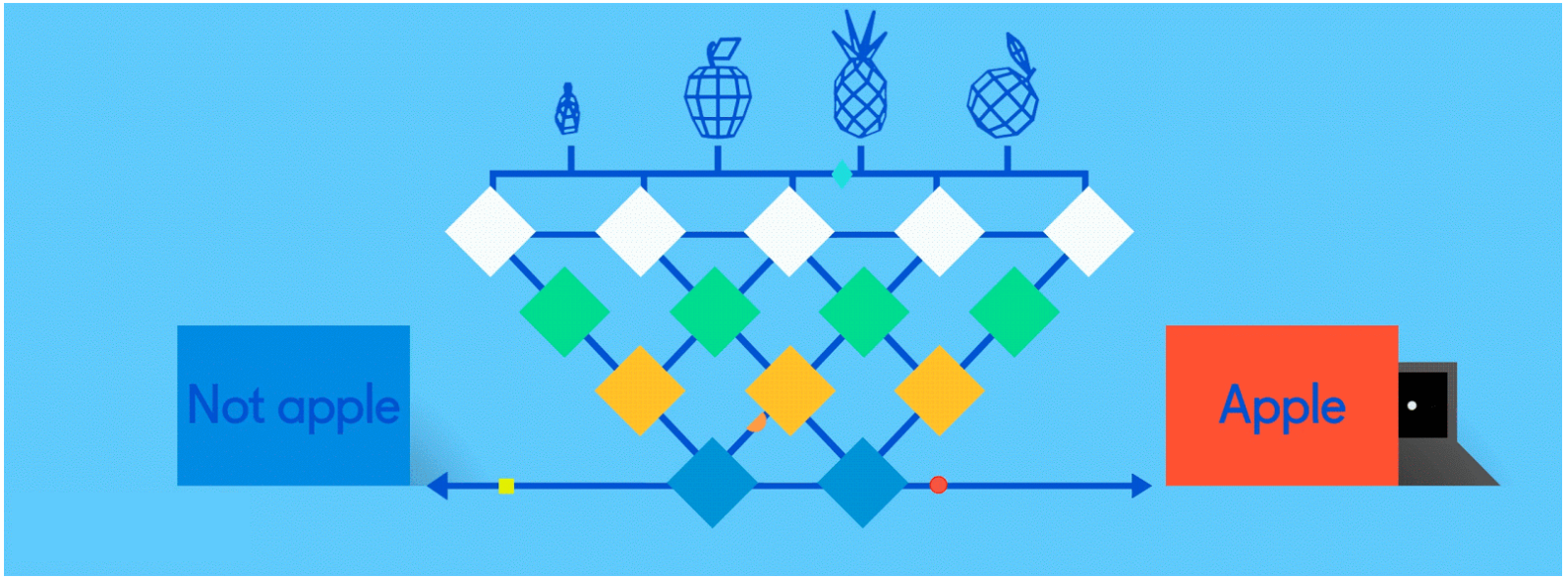
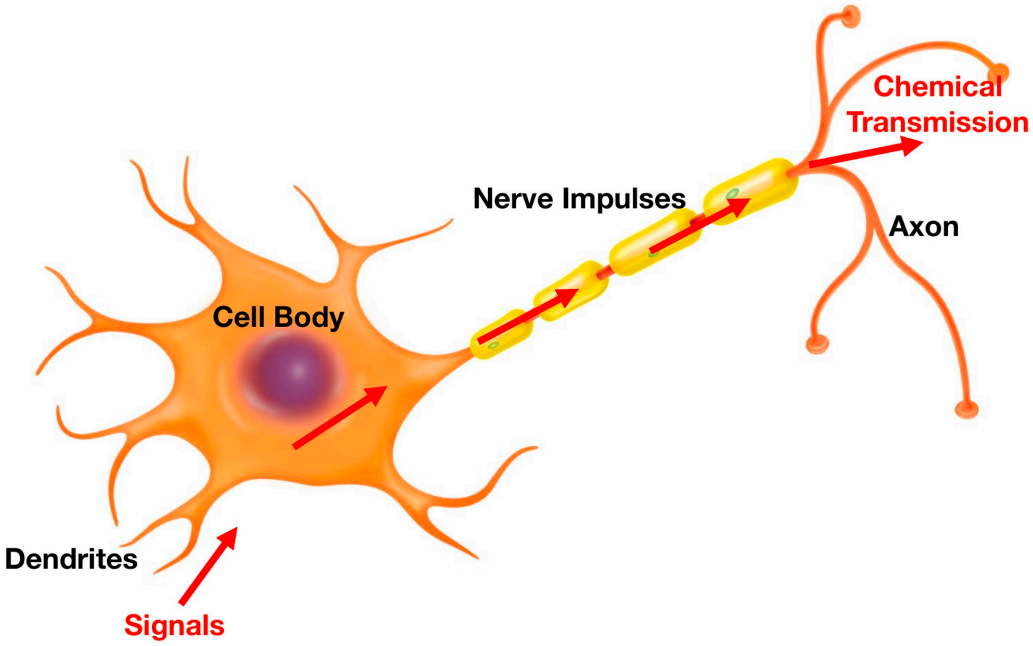
Neural Networks for Machine Learning

introduction



Biological neural activity



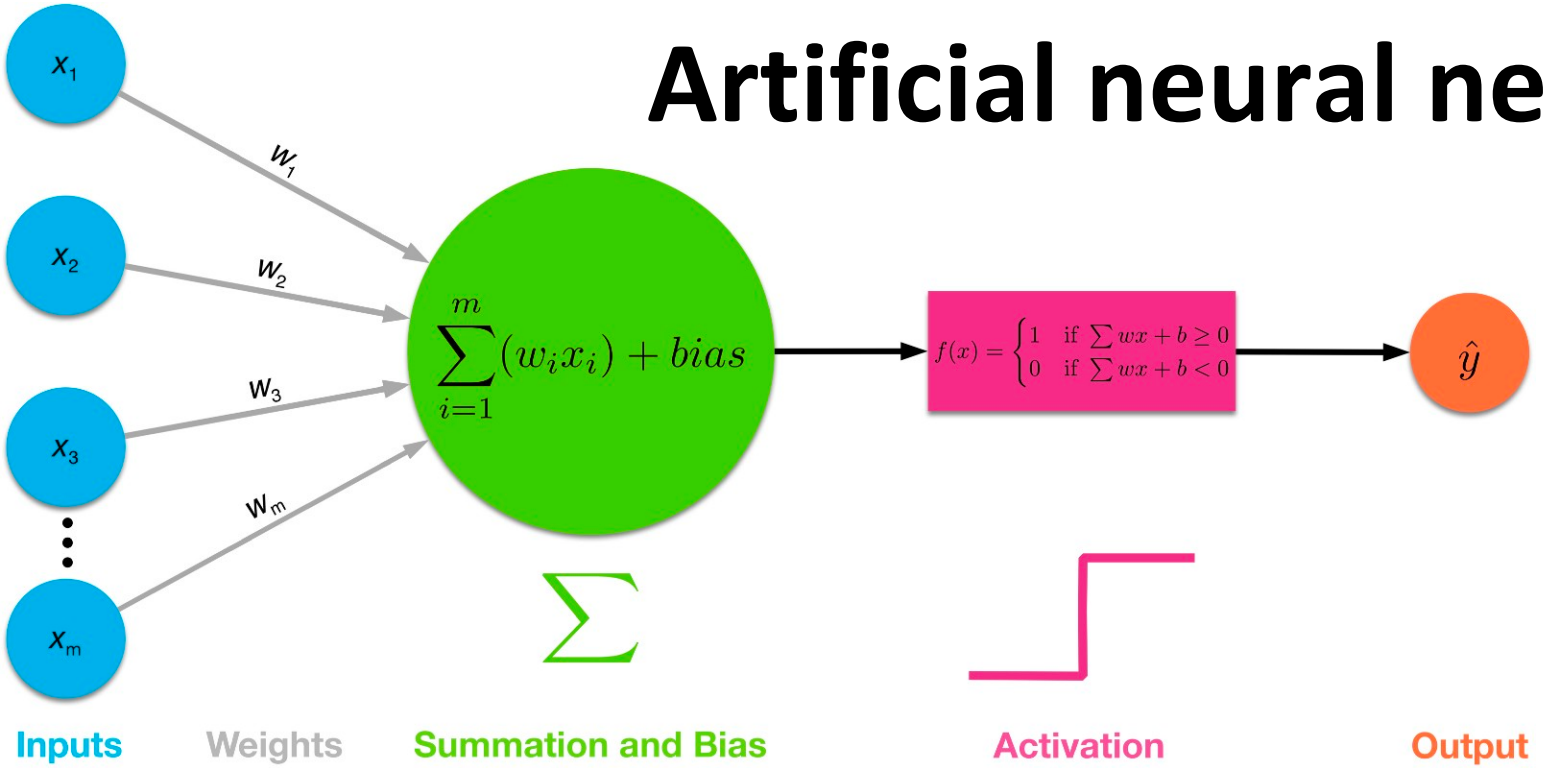
Neurons have body, axon and many dendrites

- In one of two states: firing and rest
- They fire if total incoming stimulus > threshold

Synapse: thin gap between axon of one neuron and dendrite of another

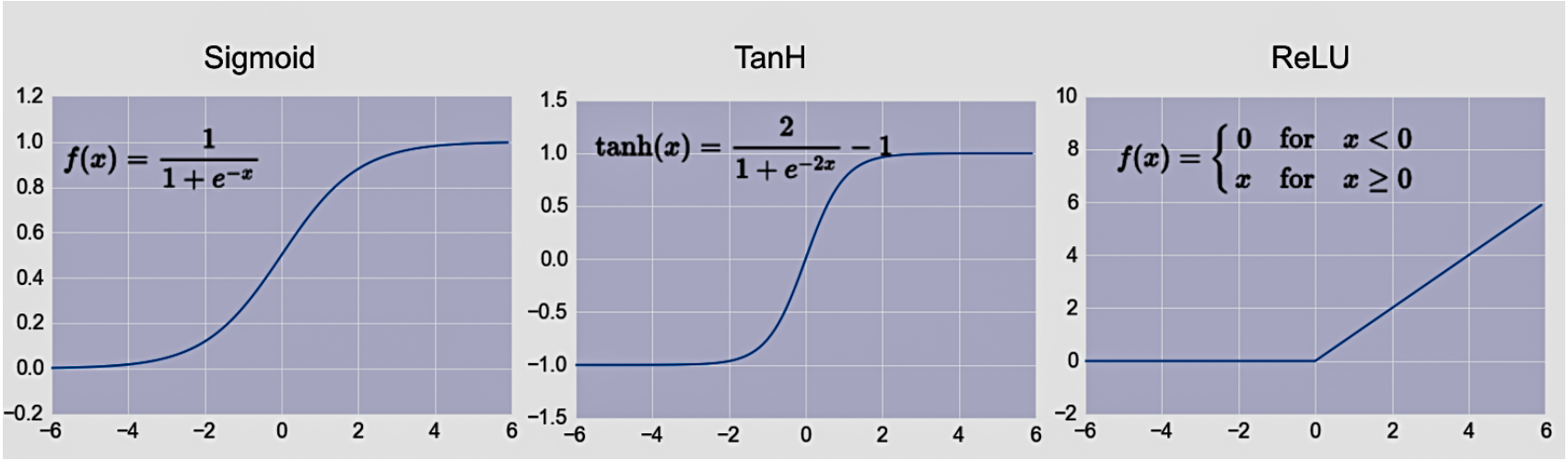
- Signal exchange

Artificial neural network



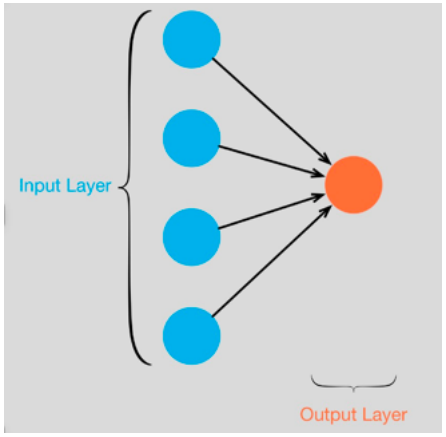
- Set of **nodes** with inputs and outputs
  Node performs computation via its **activation function**
- **Weighted connections** between nodes
- Connectivity gives network architecture
- NN computations depend on connections & weights

Common Activation Functions



Choice of activation function depends on problem and available computational power

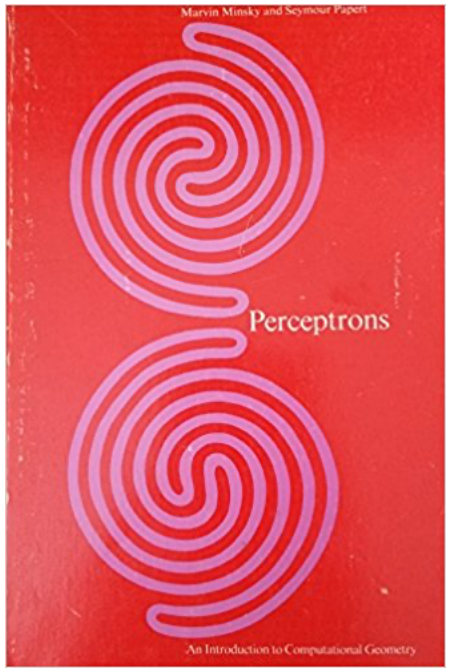
Single Layer Perceptron



NEW NAVY DEVICE LEARNS BY DOING; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

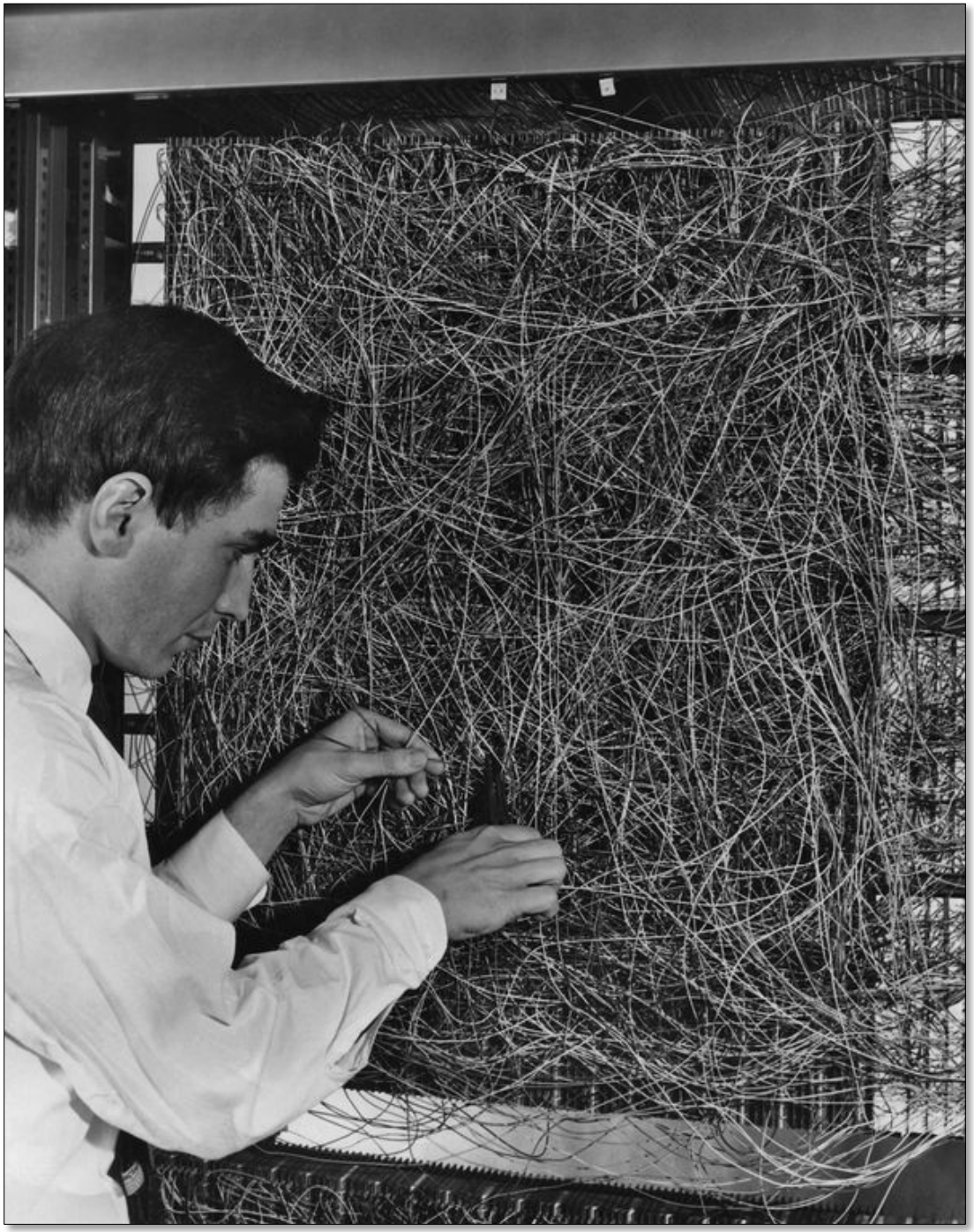
SPECIAL TO THE NEW YORK TIMES JULY 8, 1958

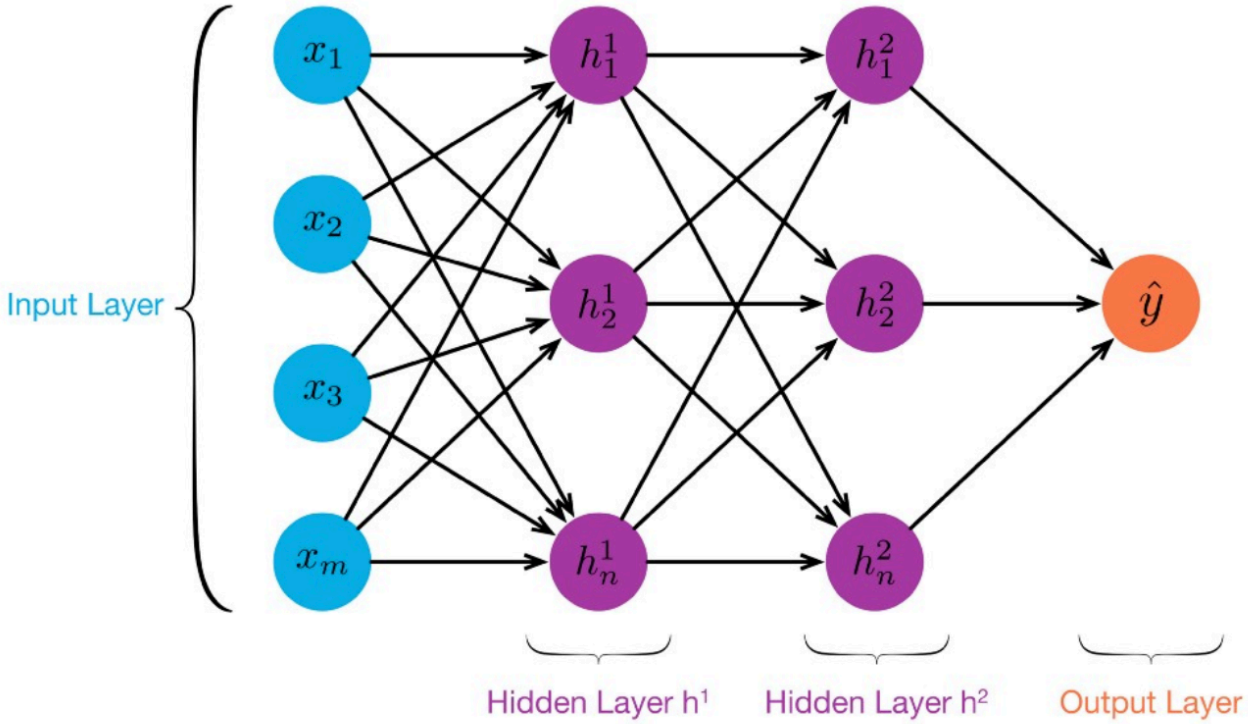
WASHINGTON, July 7 (UPI) -- The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.



- Full 1958 NYT article above here
- Rosenblatt: it can **learn** to compute functions by learning weights on inputs from examples
- Not all functions ☹, cf. Perceptrons

A man adjusting the random wiring network between the light sensors and association unit of scientist Frank Rosenblatt's Perceptron, or MARK 1 computer, at the Cornell Aeronautical Laboratory, Buffalo, New York, circa 1960. The machine is designed to use a type of artificial neural network, known as a perceptron. (Photo by Frederic Lewis/Archive Photos/Getty Images)



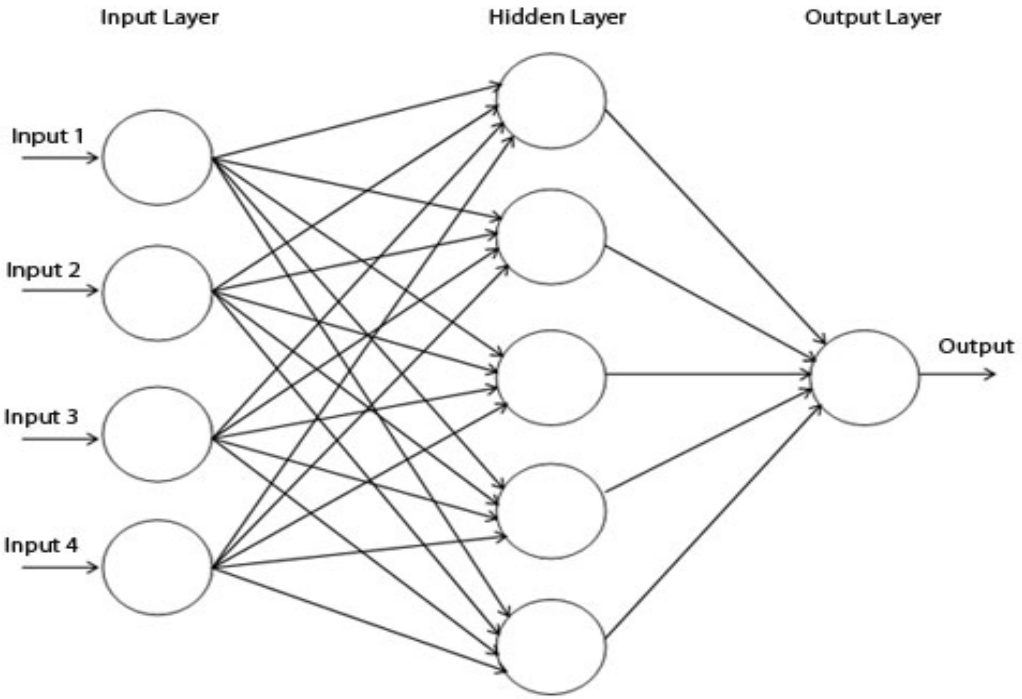


MLP: Multilayer Perceptron

- ≥ 1 "hidden layers" between inputs & output
- Can compute non-linear functions
- Training: adjust weights slightly to reduce error between output **y** and target value **t;** repeat
- Introduced in 1980s, still used today

Backpropagation

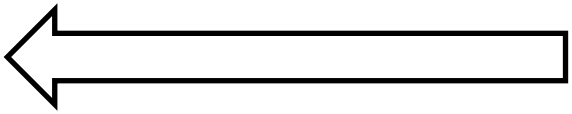
Forward direction



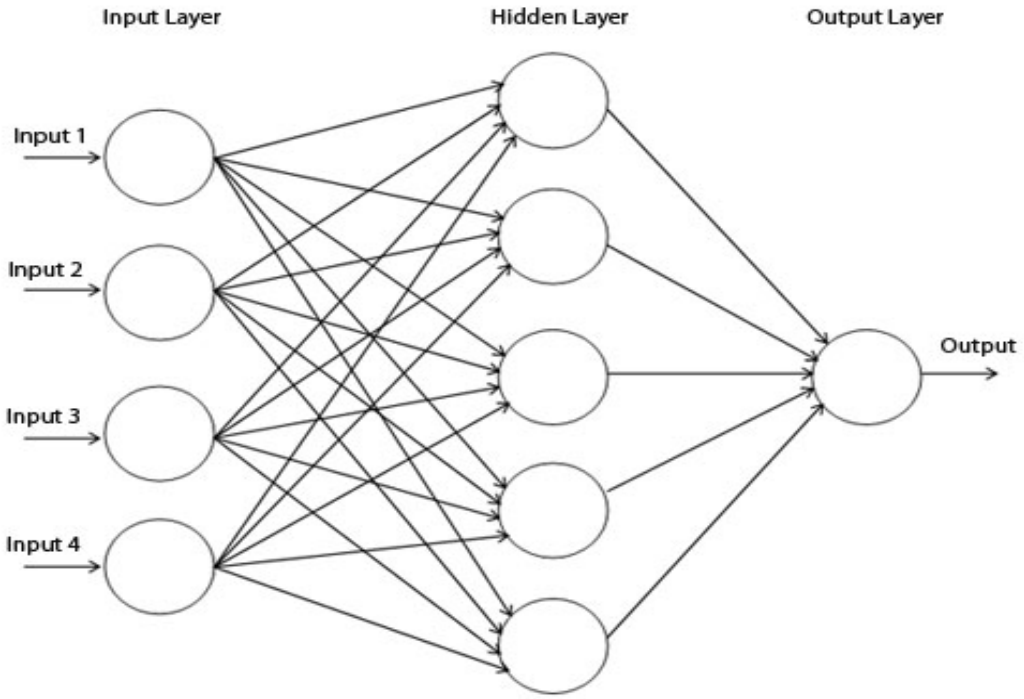
Calculate network and error

Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (8 October 1986). Learning representations by back-propagating errors. Nature. 323 (6088): 533–536.

Backpropagation



Backward direction



Backpropagate: from output to input, recursively compute $\frac{\partial E}{\partial w_{ij}} = \nabla_w E$ and adjust weights

Neural Network Architectures

Current focus on large networks with different “architectures” suited for different kinds of tasks

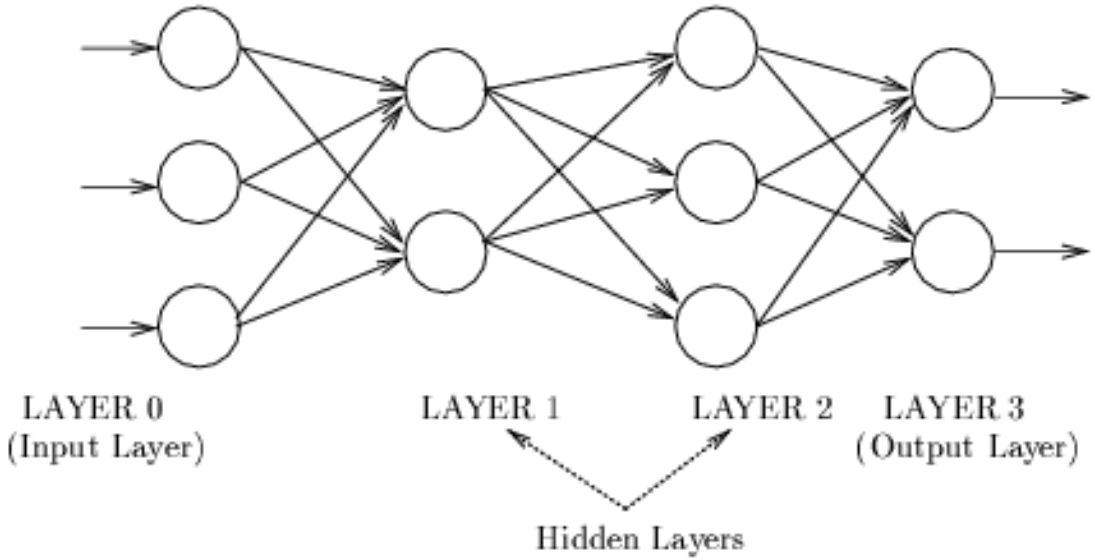
- Feedforward Neural Network
- CNN: Convolutional Neural Network
- RNN: Recurrent Neural Network
- LSTM: Long Short Term Memory
- GAN: Generative Adversarial Network

Feedforward Neural Network

- Connections allowed from a node in layer *i* only to nodes in layer *i*+1

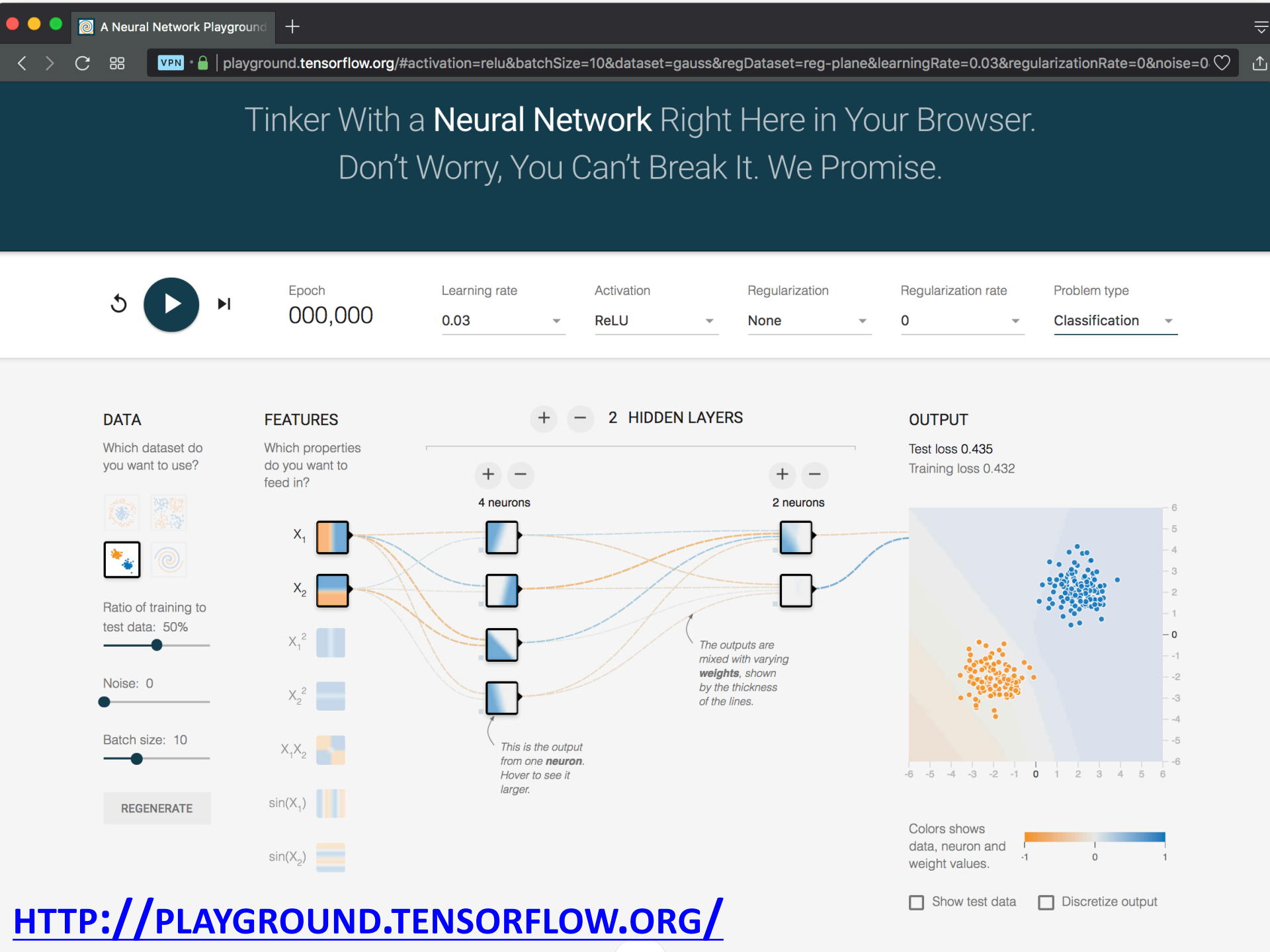
  i.e., no cycles or loops

- Simple, widely used architecture.

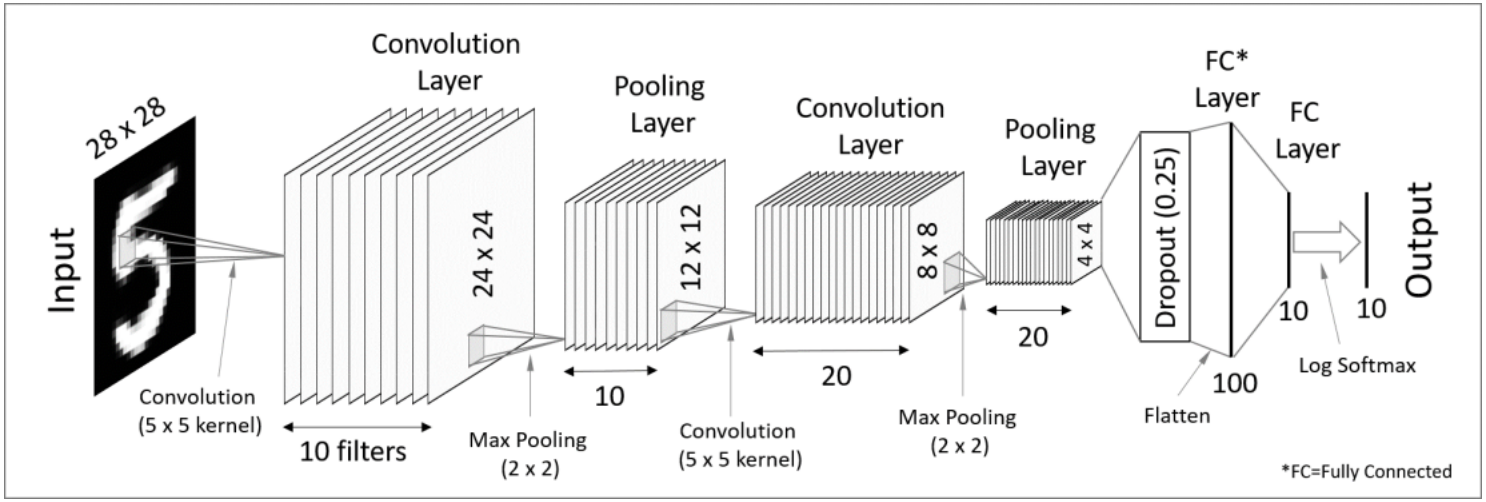


downstream nodes tend to successively abstract features from preceding layers

HTTP://PLAYGROUND.TENSORFLOW.ORG/


A Neural Network Playground
playground.tensorflow.org/#activation=relu&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0
Tinker With a Neural Network Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.
Epoch
000,000
Learning rate
0.03
Activation
ReLU
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
X1
X2
X1²
X2²
X1X2
sin(X1)
sin(X2)
2 HIDDEN LAYERS
4 neurons
2 neurons
The outputs are mixed with varying weights, shown by the thickness of the lines.
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.435
Training loss 0.432
Colors shows data, neuron and weight values.
Show test data
Discretize output
HTTP://PLAYGROUND.TENSORFLOW.ORG/

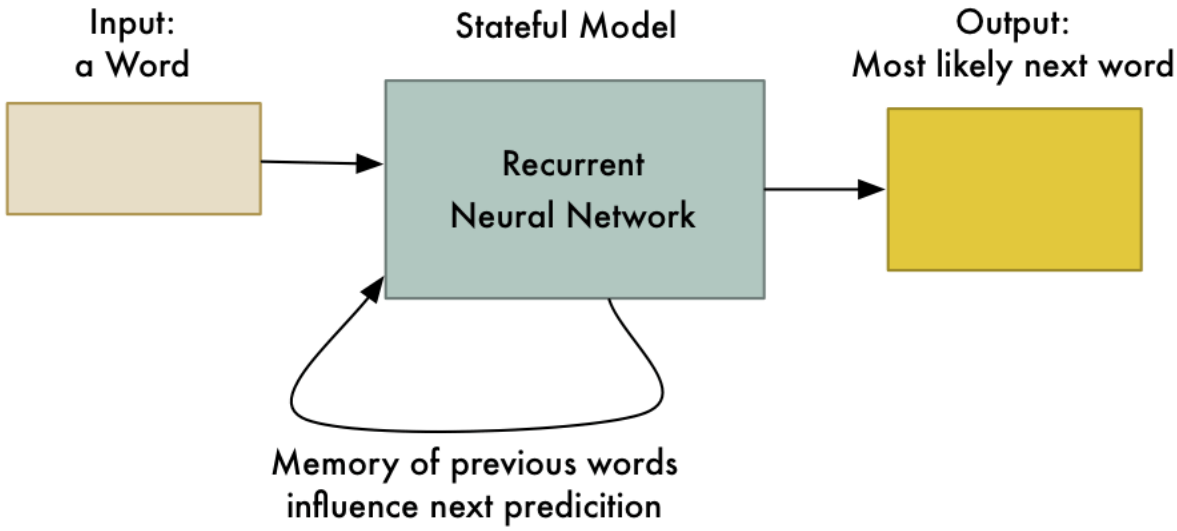
CNN: Convolutional Neural Network



- Good for image processing: classification, object recognition, automobile lane tracking, etc.
- Classic demo: learn to recognize hand-written digits from MNIST data with 70K examples

RNN: Recurrent Neural Networks

- Good for learning over sequences of data, e.g., a sentence orf words
- LSTM (Long Short Term Memory) a popular architecture



gif from Adam Geitgey

Deep Learning Frameworks

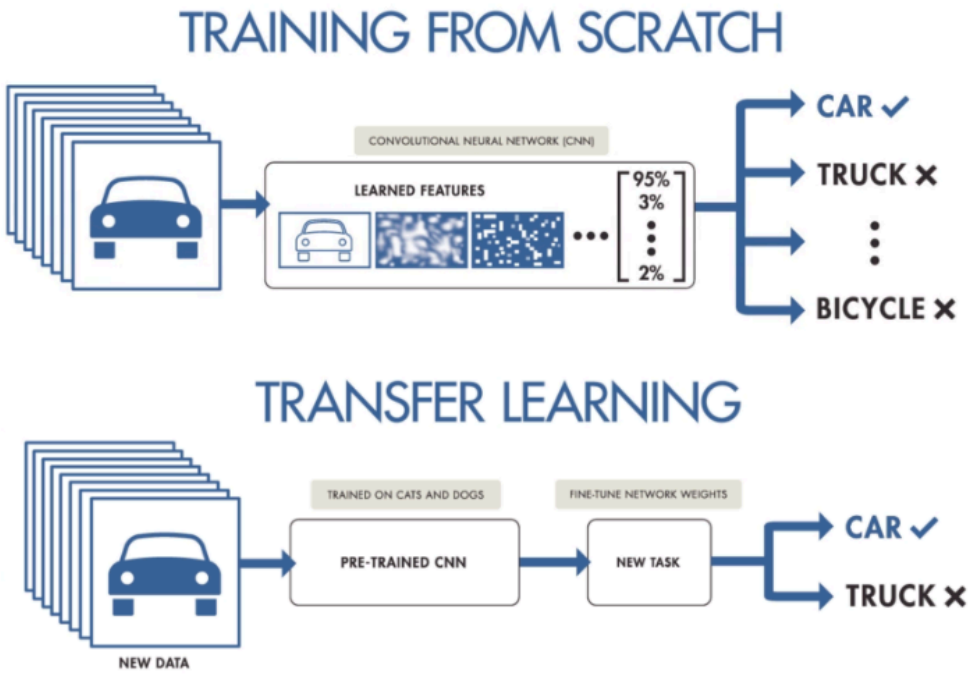
- Popular open source deep learning frame-works use Python at top-level; C++ in backend
  - TensorFlow (via Google)
  - PyTorch (via Facebook)
  - MxNet (Apache)
  - Caffe (Berkeley)
- Keras: popular API works with the first two and provides good support at architecture level

Good at Transfer Learning

- Neural networks effective for transfer learning

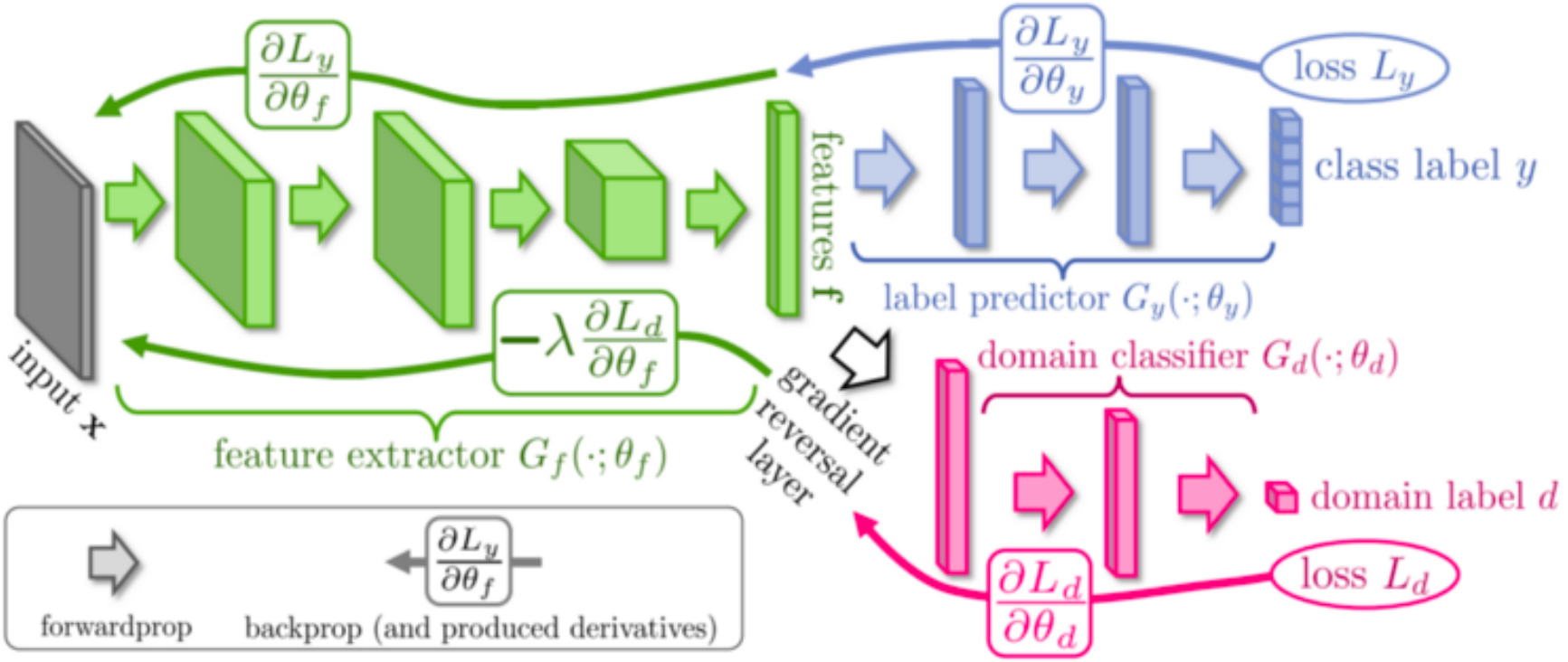
  Using parts of a model trained on a task as an initial model to train on a different task

- Particularly effective for image recognition



Good at Transfer Learning

- For images, the initial stages of a model learn high-level visual features (lines, edges) from pixels
- Final stages predict task-specific labels



source:http://ruder.io/transfer-learning/

Fine Tuning a NN Model

- Special kind of transfer learning
  - Start with a pre-trained model
  - Replace last output layer with a new one
  - Fix all but last layer by marking as trainable:false
- Retraining on new task and data very fast
  - Only the weights for the last layer are adjusted
- Example
  - Start: NN to classify animal pix with 100s of categories
  - Finetune on new task to classify pix of 15 common pets

Conclusions

- Quick introduction to neural networks and deep learning
- Learn more by
  - Take UMBC's CMSC 478 machine learning class
  - Try scikit-learn's neural network models
  - Explore Google's Machine Learning Crash Course
  - Try Miner/Kasch tutorial on applied deep learning
  - Work through examples
- and then try your own project idea