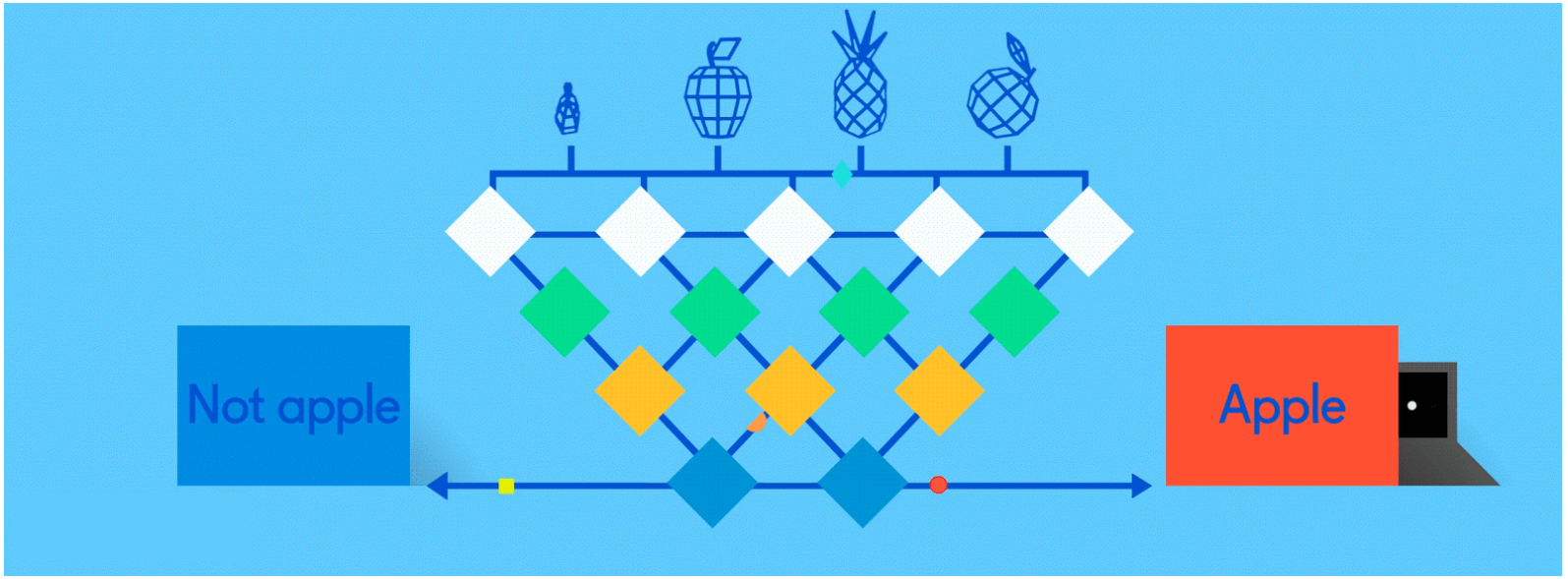
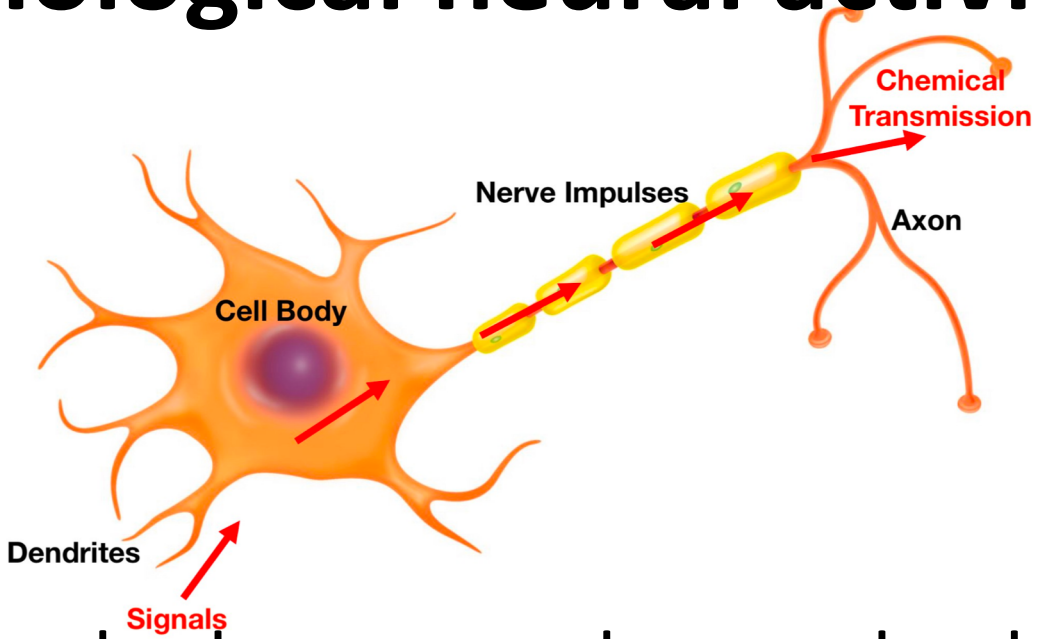
Neural Networks for Machine Learning



Biological neural activity



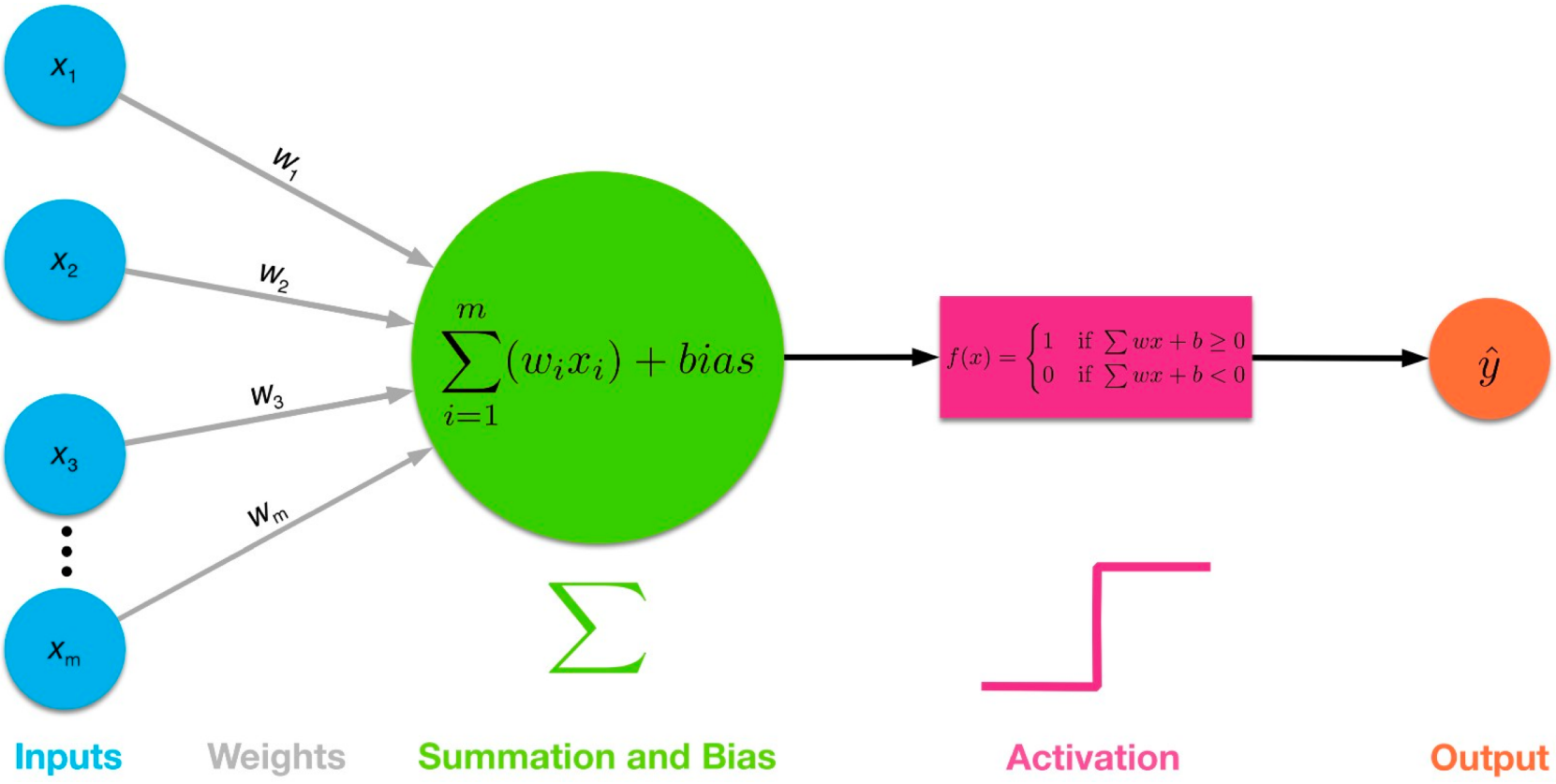
Neuron has body, axon and many dendrites

- In one of the two states: firing and rest
- Neuron fires if total incoming stimulus > threshold

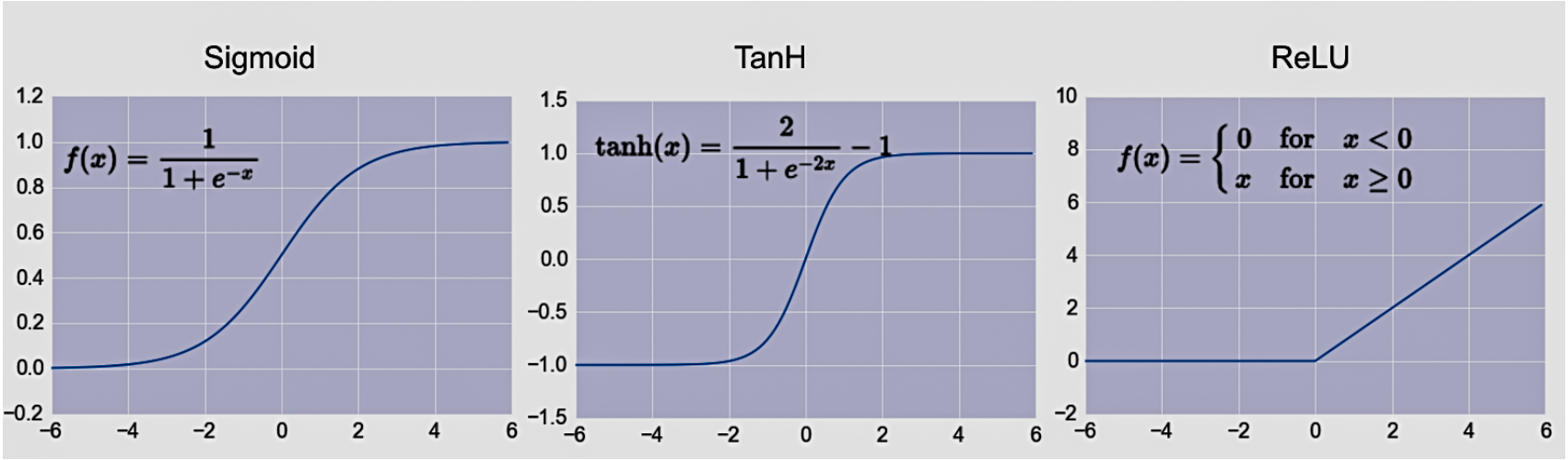
– Synapse: thin gap between axon of one neuron and dendrite of another

- Signal exchange

Artificial neural network

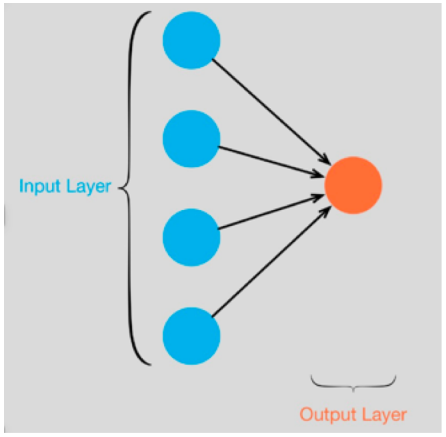


Common Activation Functions



Choice of activation function depends on problem and available computational power

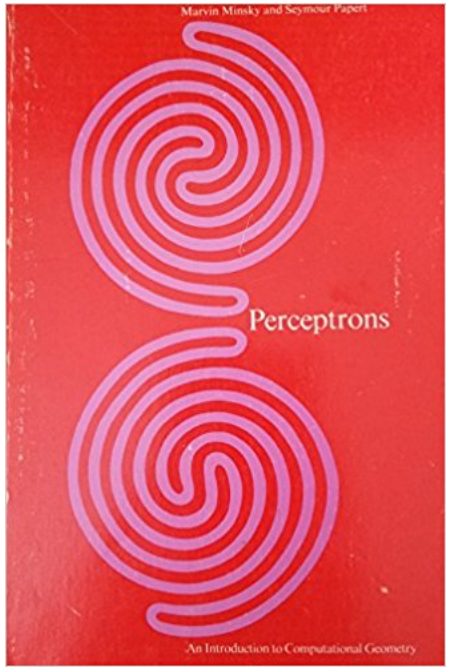
Single Layer Perceptron



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

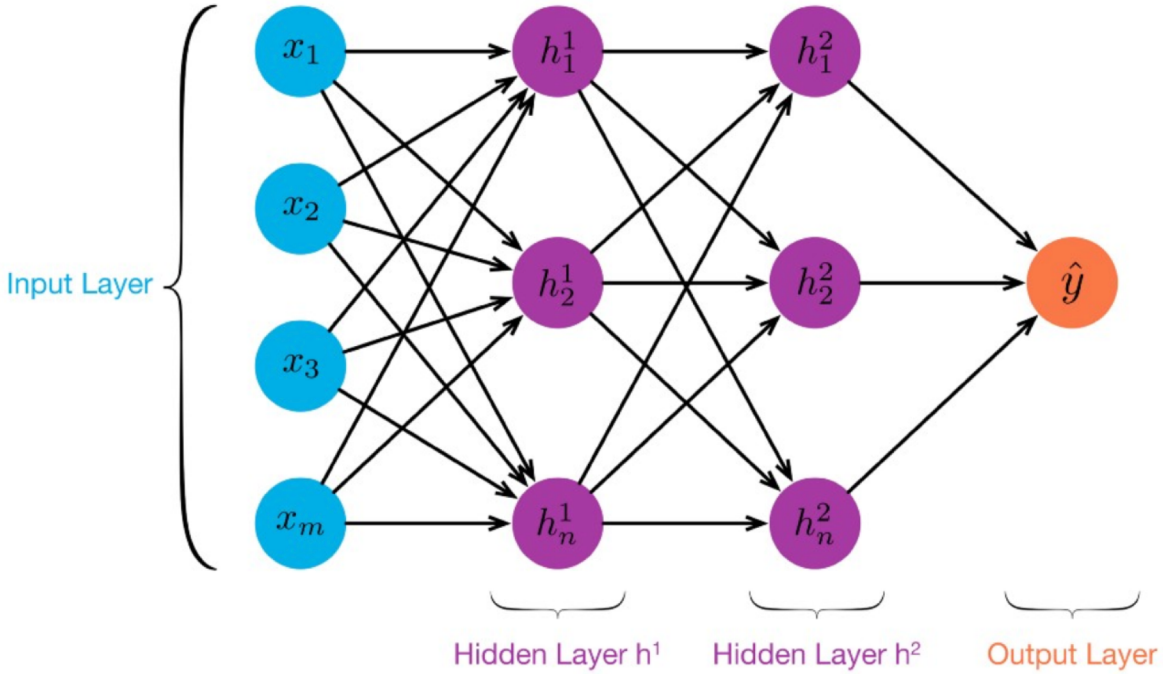
SPECIAL TO THE NEW YORK TIMES JULY 8, 1958

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 showed how it can **learn** to compute functions by learning weights on inputs from examples
- Not all functions ☹, cf. Perceptrons

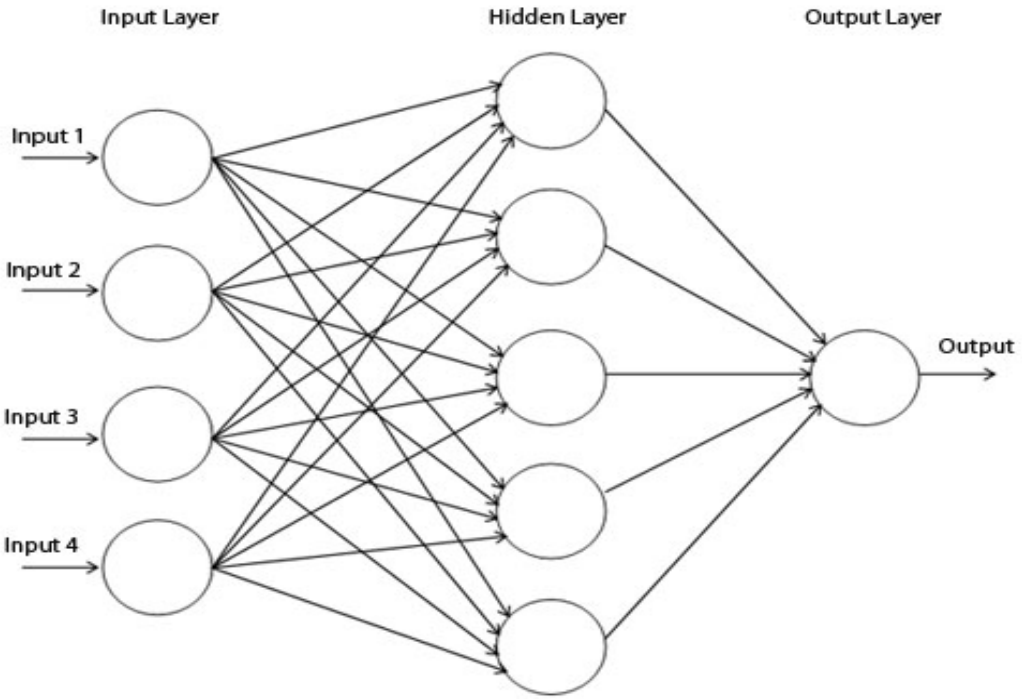
Multilayer perceptrons



- Can compute non linear functions
- Perceptron training rule: adjust weights slightly to reduce error between perceptron output **y** and target value **t;** repeat

Backpropagation Algorithm

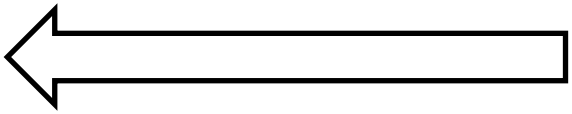
Forward direction



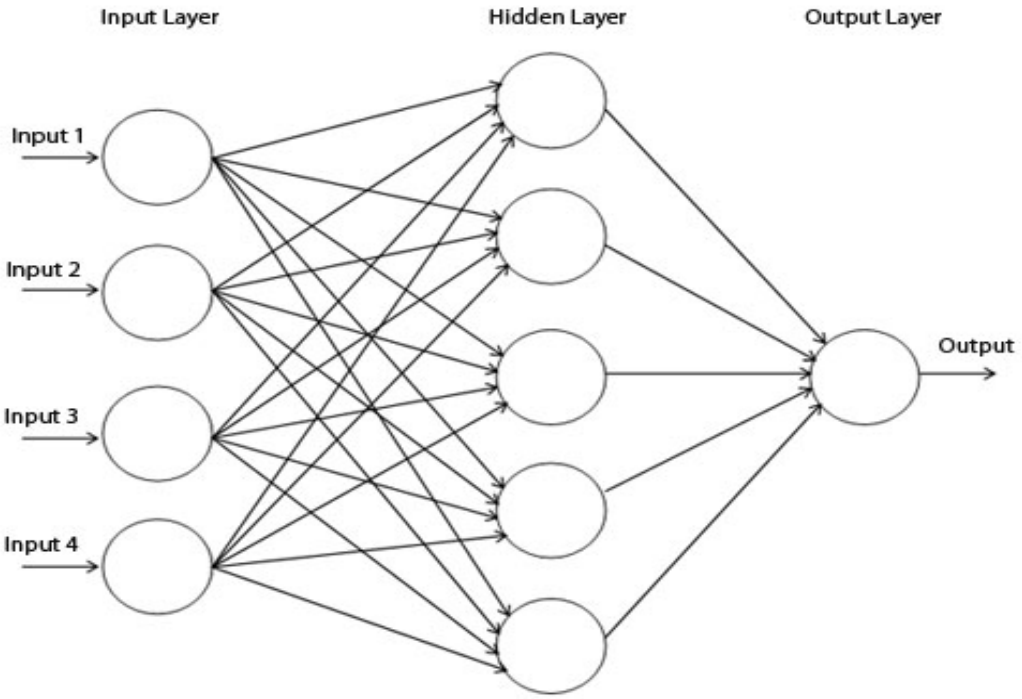
Calculate network and error

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

Backpropagation Algorithm



Backward direction



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

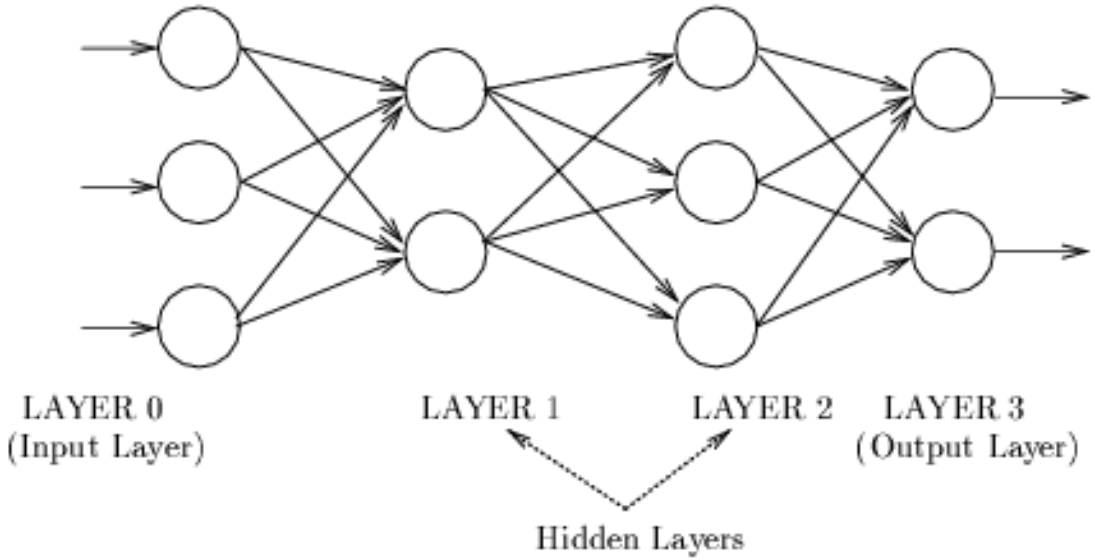
Neural Network Architectures

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 net

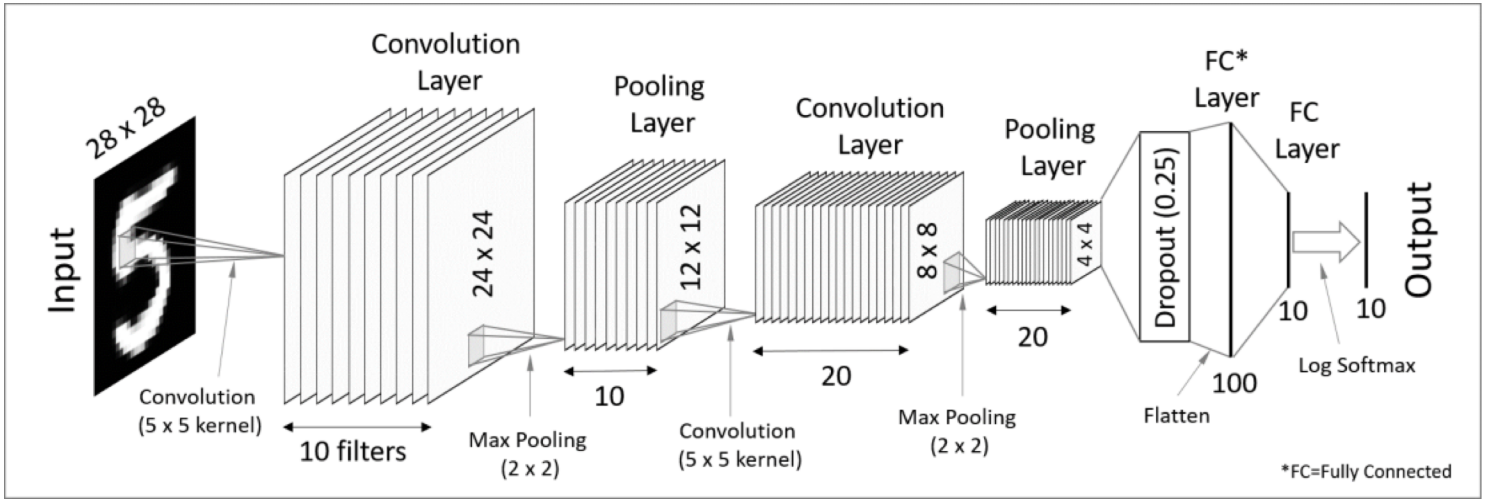
- Connections allowed from a node in layer *i* only to nodes in layer *i*+1
- Simple, widely used architecture.



downstream nodes tend to successively abstract features from preceding layers

HTTP://PLAYGROUND.TENSORFLOW.ORG/

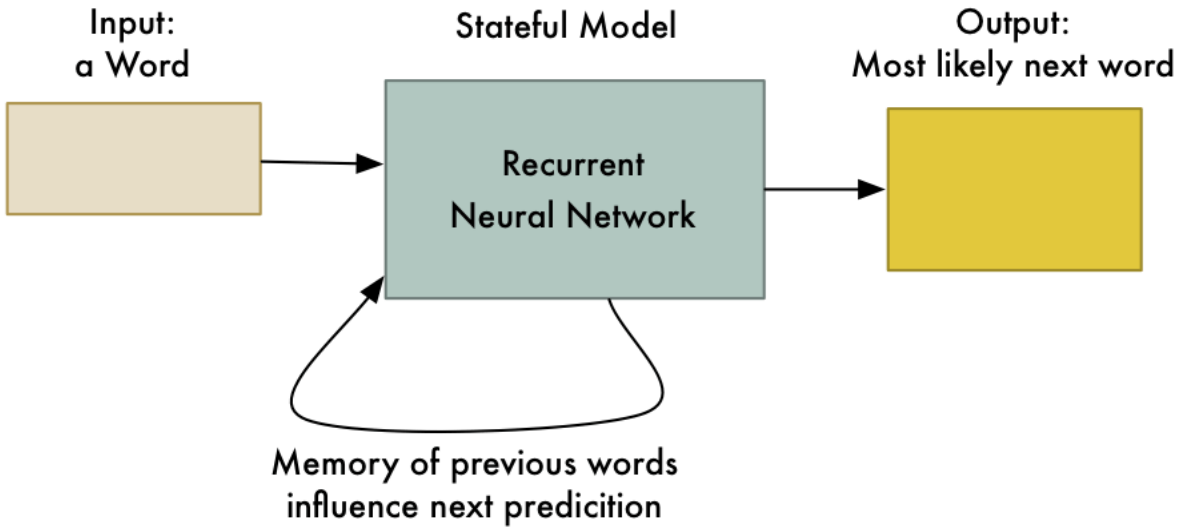
CNN: Convolutional Neural Network



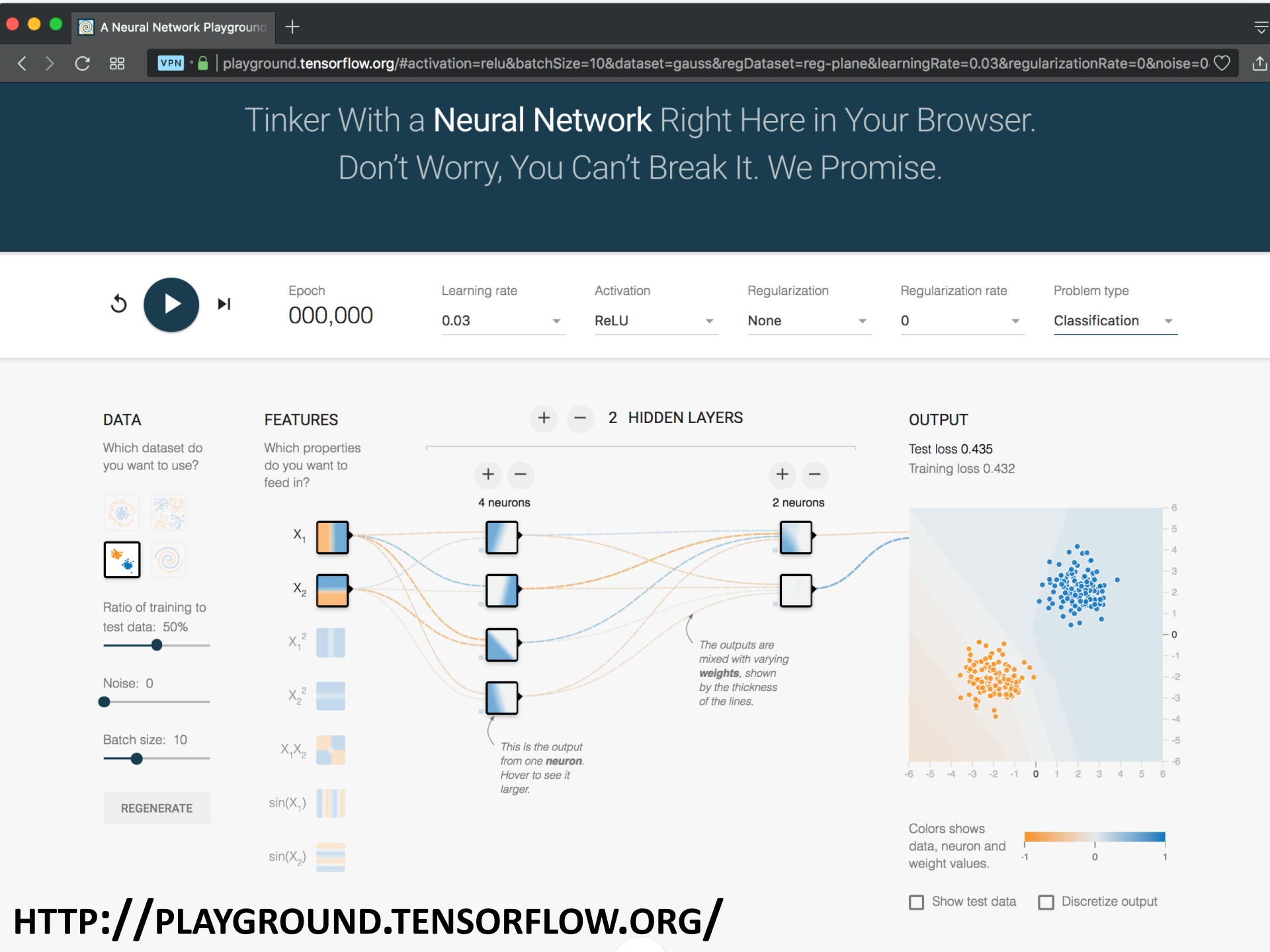
- Good for image processing: classification, object recognition, automobile lane tracking, etc.
- Classic demo: learn to recognize hand-written digits from MNIST data with 70K examples

RNN: Recurrent Neural Networks

- Good for learning over sequences of data, e.g., natural language understanding tasks
- LSTM (Long Short Term Memory) a popular architecture



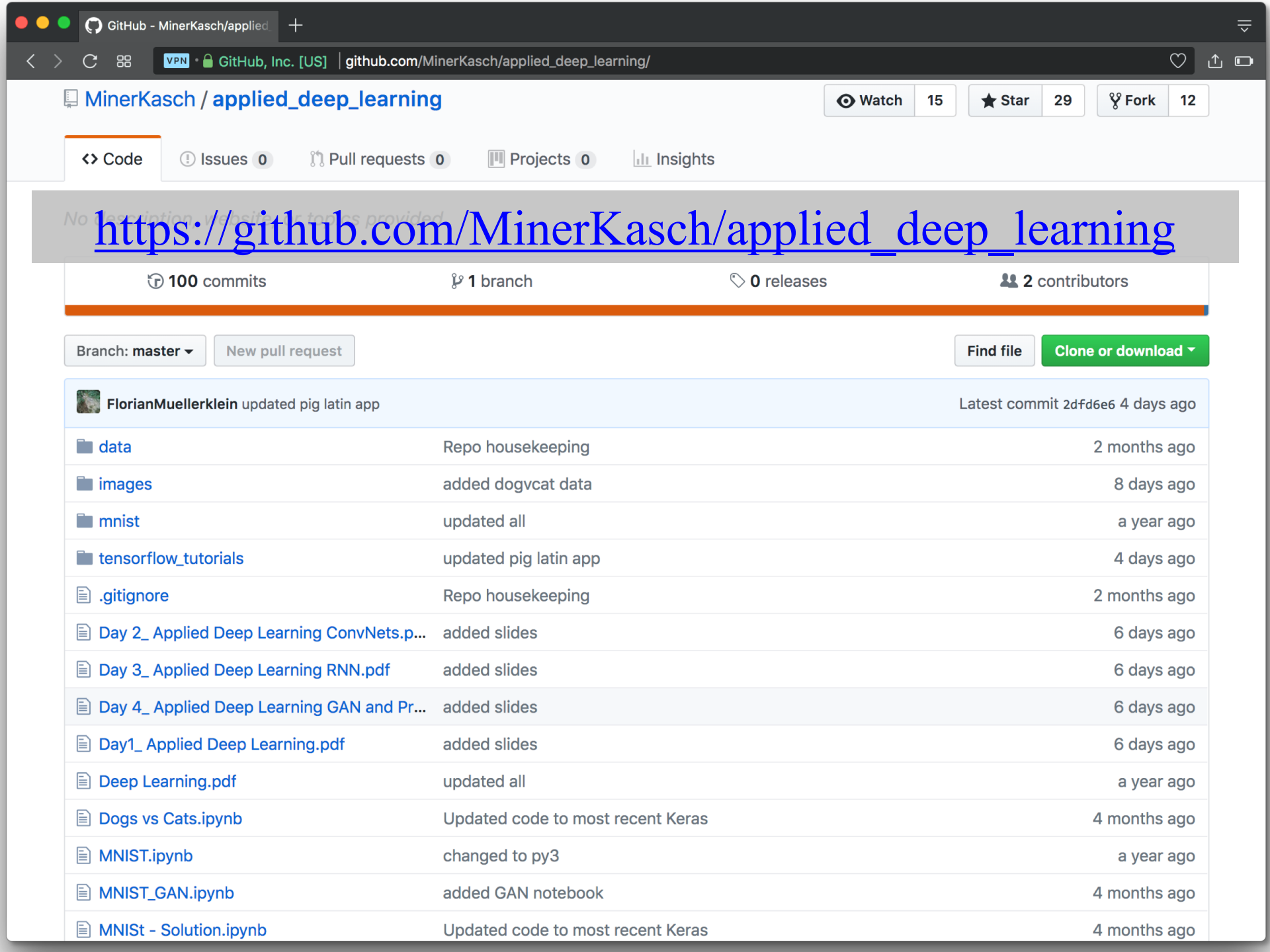
gif from [Adam Geitgey]



HTTP://PLAYGROUND.TENSORFLOW.ORG/

Deep Learning Frameworks

- Popular open source deep learning frame-works use Python at top-level; C++ in backend
  - TensorFlow (via Google)
  - PyTorch (via Facebook)
- Keras: popular API works with both and provides good support at architecture level
- Demo: MNIST CNN
- Demo: RNN sentiment


GitHub - MinerKasch/applied
VPN
GitHub, Inc. [US]
github.com/MinerKasch/applied_deep_learning/
MinerKasch / applied_deep_learning
Watch 15
Star 29
Fork 12
Code
Issues 0
Pull requests 0
Projects 0
Insights
https://github.com/MinerKasch/applied_deep_learning
100 commits
1 branch
0 releases
2 contributors
Branch: master
New pull request
Find file
Clone or download
FlorianMuellerklein updated pig latin app
Latest commit 2dfd6e6 4 days ago
data Repo housekeeping 2 months ago
images added dogvcat data 8 days ago
mnist updated all a year ago
tensorflow_tutorials updated pig latin app 4 days ago
.gitignore Repo housekeeping 2 months ago
Day 2_ Applied Deep Learning ConvNets.p... added slides 6 days ago
Day 3_ Applied Deep Learning RNN.pdf added slides 6 days ago
Day 4_ Applied Deep Learning GAN and Pr... added slides 6 days ago
Day1_ Applied Deep Learning.pdf added slides 6 days ago
Deep Learning.pdf updated all a year ago
Dogs vs Cats.ipynb Updated code to most recent Keras 4 months ago
MNIST.ipynb changed to py3 a year ago
MNIST_GAN.ipynb added GAN notebook 4 months ago
MNISt - Solution.ipynb Updated code to most recent Keras 4 months ago

Classifying digits with convolutional neural networks

This notebook contains the solution to the MNIST activity.

Load the data

Both Keras and TF-Learn contain the MNIST dataset that can be quickly loaded with some helper functions. This solution will use TF-Learn but the Keras solution will be commented out. The two libraries are very similar.

In [1]:

```
import numpy as np

import keras
from keras.datasets import mnist

# Load data from Keras
(X_train, y_train), (X_test, y_test) = mnist.load_data()


# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, 10)
y_test = keras.utils.to_categorical(y_test, 10)
```

Sentiment analysis with Recurrent Neural Networks

For this particular dataset a shallow method like tf-idf features into logistic regression will outperform the RNN. But, what this will illustrate is just how simple it is to implement an RNN for sentiment analysis with Keras and TF-Learn. The notebook was run with Keras and the equivalent TF-Learn code will be commented out.

Load the packages

In [5]:
```
import numpy as np

from keras.preprocessing import sequence
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Embedding
from keras.layers import GRU
from keras.datasets import imdb

#import tflearn
#from tflearn.data_utils import to_categorical, pad_sequences
#from tflearn.datasets import imdb
```

Conclusions

- This was a very quick introduction to neural networks and deep learning
- Find data and notebooks on github here
- Learn more by
  - taking UMBC's machine learning class
  - Self study online: try Miner/Kasch tutorial by Forian Muellerklein on applied deep learning
  - Working through examples
- and then trying your own project idea