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

\[
\sum_{i=1}^{m} (w_i x_i) + \text{bias}
\]

\[
f(x) = \begin{cases} 
1 & \text{if } \sum w_i x_i + b \geq 0 \\
0 & \text{if } \sum w_i x_i + b < 0 
\end{cases}
\]
Common Activation Functions

Choice of activation function depends on problem and available computational power.
Single Layer Perceptron

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

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

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](#)
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

Backpropagation Algorithm

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

![Diagram of RNN](gif from Adam Geitgey)
Tinker With a **Neural Network** Right Here in Your Browser.
Don’t Worry, You Can’t Break It. We Promise.

**DATA**
Which dataset do you want to use?

**FEATURES**
Which properties do you want to feed in?

**OUTPUT**
Test loss 0.435
Training loss 0.432

Colors shows data, neuron and weight values.

- Show test data
- Discretize output
Deep Learning Frameworks

- Popular open source deep learning frameworks 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
https://github.com/MinerKasch/applied_deep_learning

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Repo housekeeping</td>
<td>2 months ago</td>
</tr>
<tr>
<td>images</td>
<td>added dogvcat data</td>
<td>8 days ago</td>
</tr>
<tr>
<td>mnist</td>
<td>updated all</td>
<td>a year ago</td>
</tr>
<tr>
<td>tensorflow_tutorials</td>
<td>updated pig latin app</td>
<td>4 days ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td>Repo housekeeping</td>
<td>2 months ago</td>
</tr>
<tr>
<td>Day 2_ Applied Deep Learning ConvNets.pdf</td>
<td>added slides</td>
<td>6 days ago</td>
</tr>
<tr>
<td>Day 3_ Applied Deep Learning RNN.pdf</td>
<td>added slides</td>
<td>6 days ago</td>
</tr>
<tr>
<td>Day 4_ Applied Deep Learning GAN and Pr...</td>
<td>added slides</td>
<td>6 days ago</td>
</tr>
<tr>
<td>Day1_ Applied Deep Learning.pdf</td>
<td>added slides</td>
<td>6 days ago</td>
</tr>
<tr>
<td>Deep Learning.pdf</td>
<td>updated all</td>
<td>a year ago</td>
</tr>
<tr>
<td>Dogs vs Cats.ipynb</td>
<td>Updated code to most recent Keras</td>
<td>4 months ago</td>
</tr>
<tr>
<td>MNIST.ipynb</td>
<td>changed to py3</td>
<td>a year ago</td>
</tr>
<tr>
<td>MNIST_GAN.ipynb</td>
<td>added GAN notebook</td>
<td>4 months ago</td>
</tr>
<tr>
<td>MNIST - Solution.ipynb</td>
<td>Updated code to most recent Keras</td>
<td>4 months ago</td>
</tr>
</tbody>
</table>
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.

```python
# 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

```python
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
```
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