Gridspace

GRIDSPACE IAP LECTURE 3
ML/JAX

January 12, 2023

MACHINE LEARNING

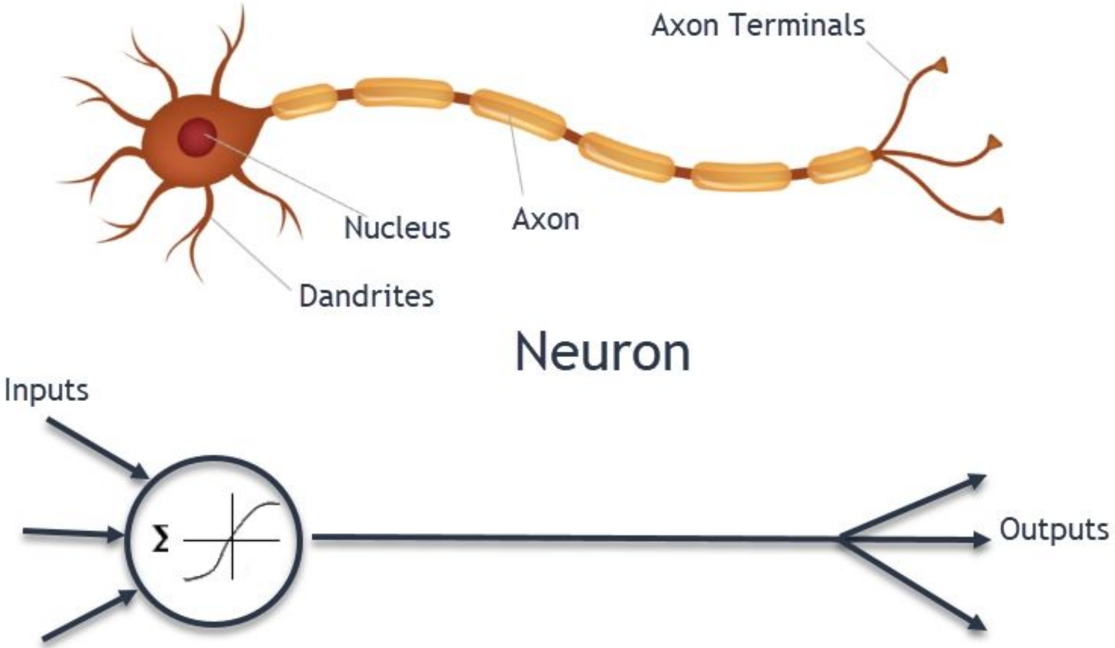
What is Machine Learning?

- In essence, it is the realization of human-like action through learning by machines.
- Transformative to many fields
- Examples:
  - Linear Regression (drawing a line of best fit connecting points, or dividing points of different categories)
  - Clustering
  - NLP (understanding and applying linguistic knowledge)
    - TTS / Speech Parsing
    - Understanding Linguistic Concepts
      - Emotional Valence
      - Dialog Acts

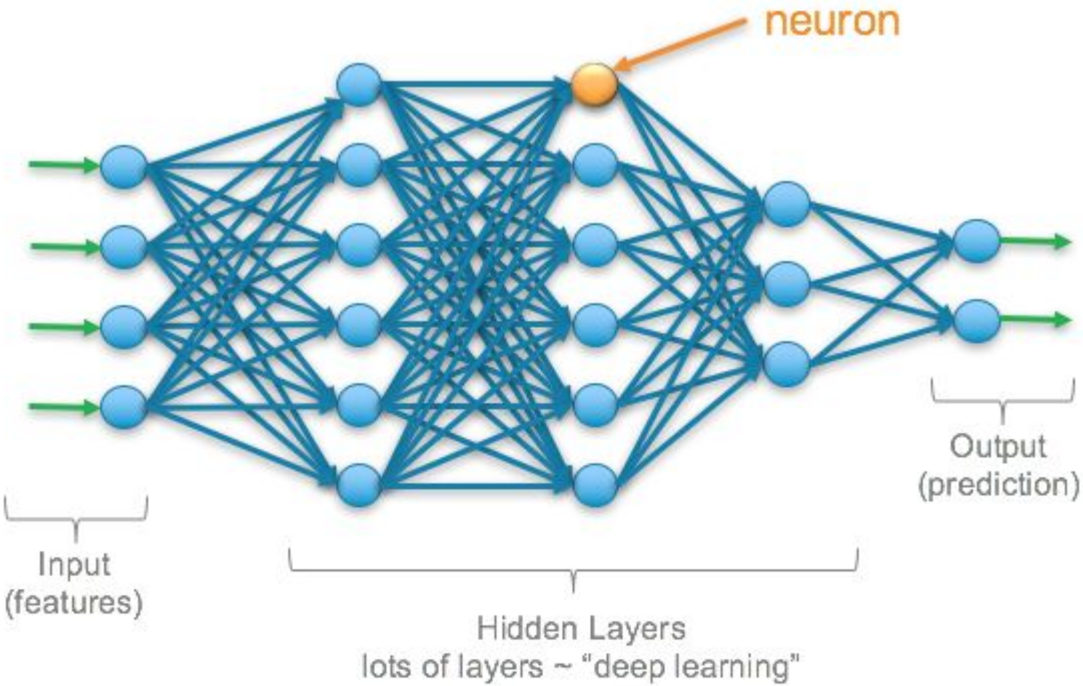
AI is the new electricity.

— CS229 (Machine Learning): Professor Andrew Ng

The Neuron & The Model of a Neuron

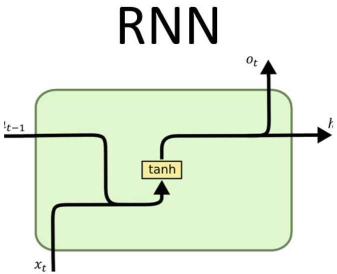


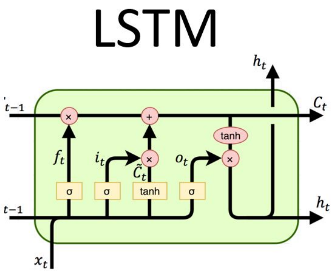
Neural Networks



Neural Networks Architectures

- FCNN (Fully Connected Neural Network) / FFNN (Feedforward Neural Network)
- RNN (Recurrent Neural Network) / FBNN (Feedback Neural Network) such as LSTM (Long Short-Term Memory)
  - Can encode temporal dependencies in ways FFNN can't
- CNN (Convolutional Neural Network)
  - encodes temporal locality and independence in a simpler manner than FFNN
- GAN (Generative Adversarial Network)
  - can be used unsupervised
- Transformers
  - encodes attention in a manner suited for lingustics





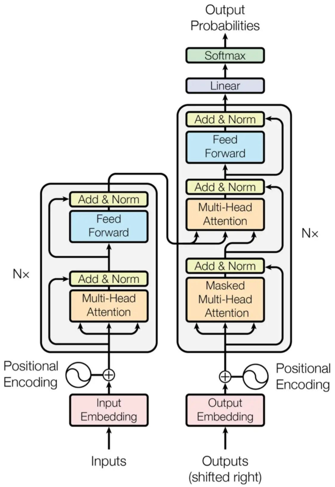


Figure 1: The Transformer - model architecture.

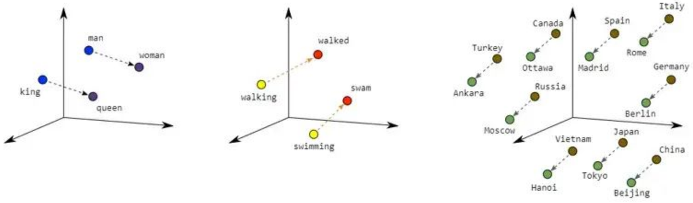
Neural Network Stages

- Training
    - Loss (Supervised vs Unsupervised Learning)
    - Optimizers
- Validation
    - Hyperparameters
- Production / Testing

Word Embeddings vs Bag of Words

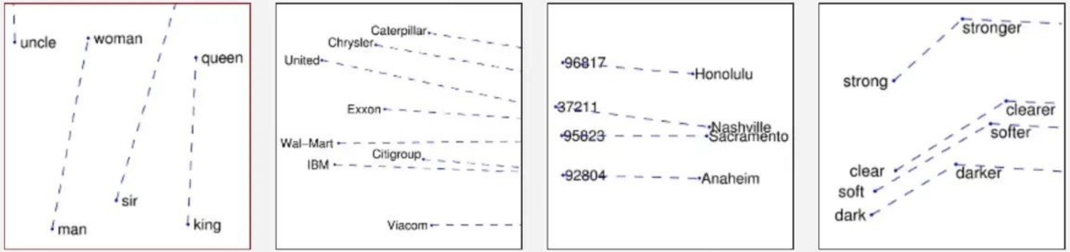
- Word2Vec
- GloVe: Global Vectors for Word Representation

Word2Vec



Male-Female Verb Tense Country-Capital

GloVe



man - woman company - ceo city - zip code comparative - superlative

PYTHON & JAX

Is Python too slow?

- Depends on what exactly you are doing
- For pipelining data or simply experimenting with algorithms, probably not.
- However, for hot loops, probably.

```python
from time import perf_counter_ns
import numpy as np
import jax.numpy as jnp
from jax import jit

def time_func(func, name):
    start = perf_counter_ns()
    res = func()
    end = perf_counter_ns()
    dur = end - start
    print(f"{name}() result = {res}")
    print(f"duration: {dur}ns ({dur/1_000:.0}µs) ({dur/1_000_000:.0}ms) ({dur/1_000_000_000:.0}sec)")

data = np.random.normal(size=(1_000, 1))

def basic_python():
    result = 0
    for i in range(0, 1_000):
        result += data.T[0, i] * data[i, 0]
    return result

def numpy():
    return data.T @ data

def jax_no_jit():
    return data.T @ data

@jit
def mm(data):
    return data.T @ data
def jax_jit():
    return mm(data)

time_func(basic_python,"basic_python”)
time_func(numpy, "numpy")
time_func(jax_no_jit,"jax_no_jit”)
time_func(jax_jit, "jax_jit")
time_func(jax_jit, "jax_jit_again”)
```

See
https://jax.readthedocs.io/en/latest/faq.html#is-jax-faster-than-numpy

Results:

basic_python() result = 1026.3824549102737
duration: 252750ns (3e+02µs) (0.3ms) (0.0003sec)

numpy() result = [[1026.38245491]]
duration: 187500ns (2e+02µs) (0.2ms) (0.0002sec)

jax_no_jit() result = [[1026.38245491]]
duration: 6666ns (7e+00µs) (0.007ms) (7e-06sec)

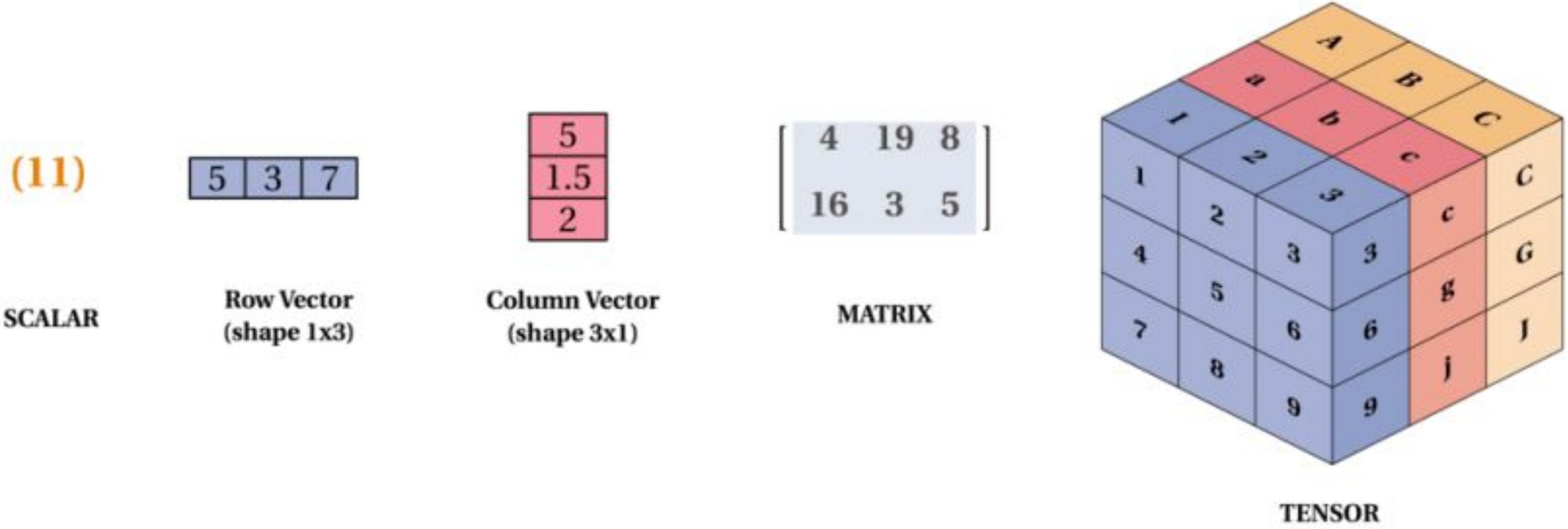
jax_jit() result = [[1026.3824]]
duration: 144373459ns (1e+05µs) (1e+02ms) (0.1sec)

jax_jit_again() result = [[1026.3824]]
duration: 12792ns (1e+01µs) (0.01ms) (1e-05sec)

Basics of Numerical Computation in Python: NumPy

- Provides 2 things:
  - Many ways to manipulate a mathematical concept
  - An efficient way to represent scalars, vectors, matrices, and higher dimensional groupings with a single Python type

The core of NumPy: Tensors (numpy.ndarray)



Python ML Libraries

- General ML
  - scikit-learn
- Deep Learning
  - Pytorch
  - Tensorflow
  - JAX

Why JAX?

https://jax.readthedocs.io/en/latest/index.html

- History
  - Autograd - Harvard Lab (make any python code differentiable)
  - XLA - Google (GPU processing of linear algebra code)
- JAX = Autograd + XLA (Differentiable Python on GPUs)
- Immutable NumPy
  - Means to an end
- Autograd (Function Differentiation)
  - The end
  - Handles "pure functions"
- JIT Compilation & GPU/TPU Processing
  - Extra benefits (differentiators)

Questions?

Let’s go through a simple application that uses JAX!

Exercises for Next Time

- If someone is whistling at a frequency of 16kHz, and you're recording them with a sampling rate of 20kHz, what frequencies could the 16kHz signal appear as?

The alias frequency is defined as $|f - fs*nearest\_int(f/fs)| = |16k - 20k*int(16k/20k)|$
$= |16k - 20k*1| = 4kHz$

- What sampling rate would you need to use to make sure the 16kHz frequencies are captured exactly?

By Nyquist's theorem, we would need to sample at at least twice the maximum frequency, so 2*16kHz = 32kHz

- Construct the following for the signal $y[n] = x[n] * h[n] = 3x[n] + 5x[n-1]$, where $x[n]$ is an input signal and $h[n]$ is the transfer function of our filter:
  - $h[n] = 3\delta[n] + 5\delta[n-1]$
  - $H(z) = 3 + 5z^{-1}$

An Exercise to the Reader for Next Time

- How do you handle randomness in JAX? Don’t think too hard about this, you can get the answer pretty easily online.
- Why would JAX not support JIT compiling side-effects (printing and globals) *nor* dynamically-sized argument-based values (e.g. passing in a length as an argument to use for a tensor)? (Hint: look at the purpose of the library as a whole.) They don’t directly state anything, but I would look in the sharp bits.
- What are Flax and Optax and where would we use them in the example application? This requires looking beyond the slides, but google searches might find the results.