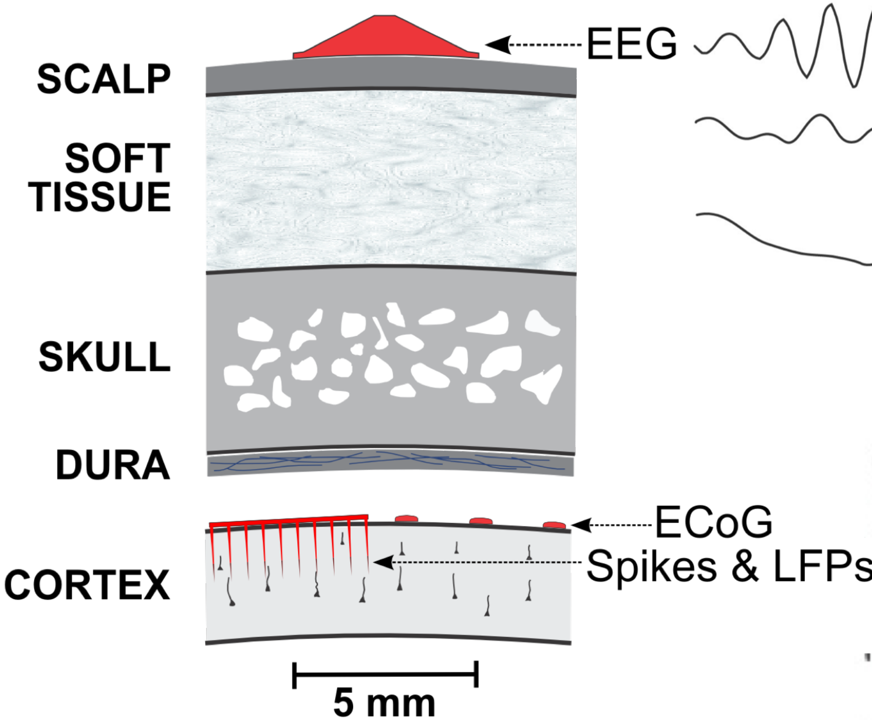
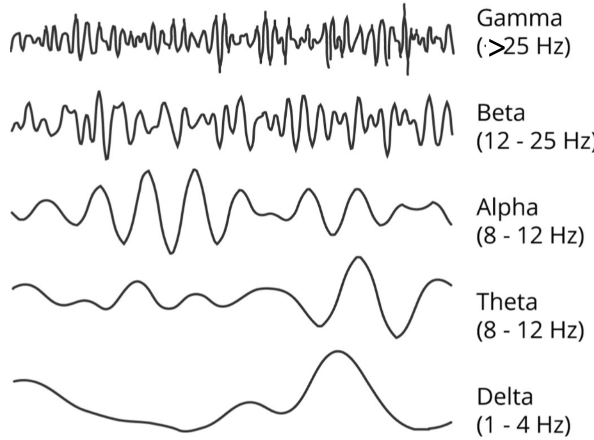
Workshop on Applied Deep Learning in Intracranial Neurophysiology

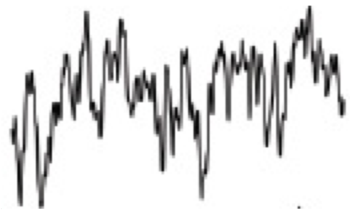
Part 2 – My first neural net

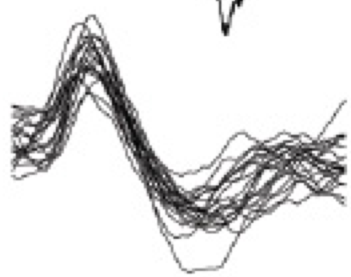
June 20, 2019

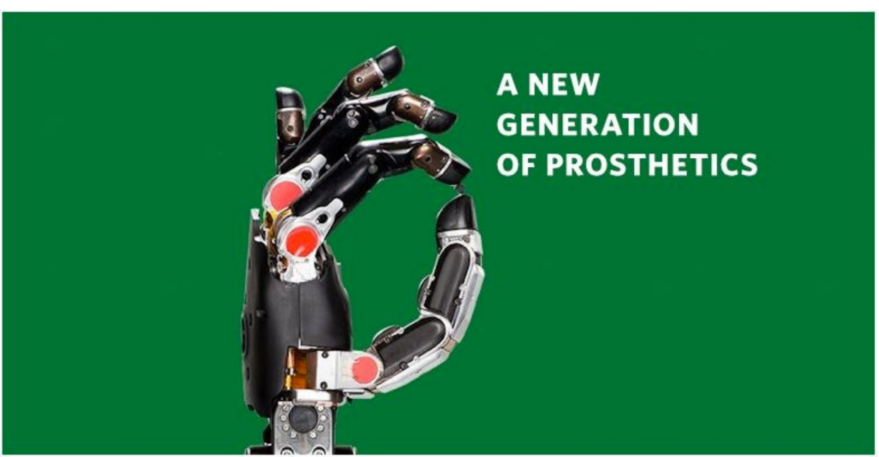
Presented by Chadwick Boulay, MSc, PhD
Sachs Lab







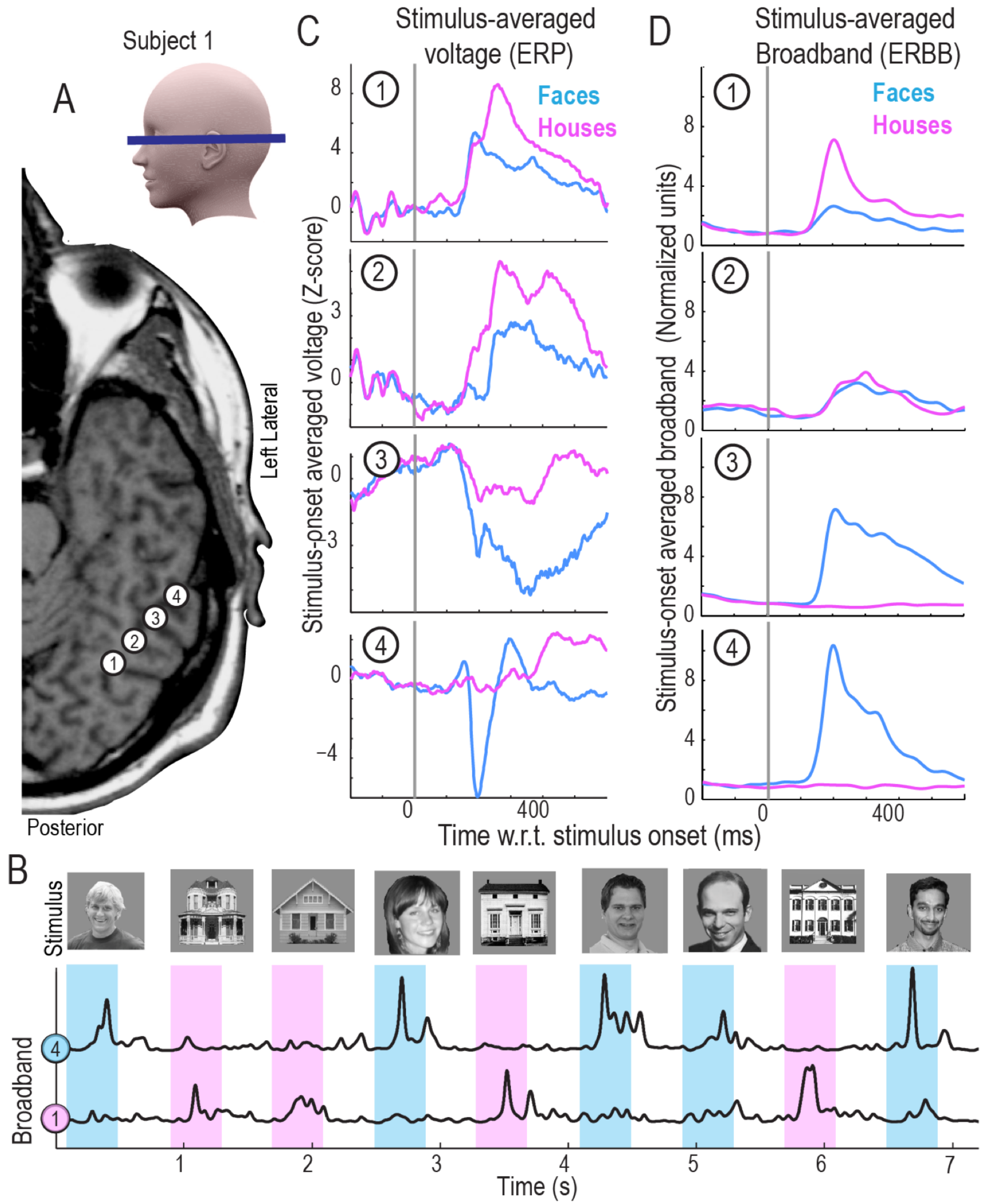




- EEG
  - 1 – 300 µV
  - 10^6 neurons
  - Synchronized synaptic activity
  - < 50 Hz
- ECoG
  - 1 – 5000 µV
  - $10^2 – 10^4$ neurons
  - < 250 Hz
- LFP
  - 1 – 1000 µV
  - 10^2 synapses
  - < 250 Hz
- Spikes
  - 200 – 500 µV
  - 1-4 neurons
  - 0.1 – 7 kHz

ECoG Data Library

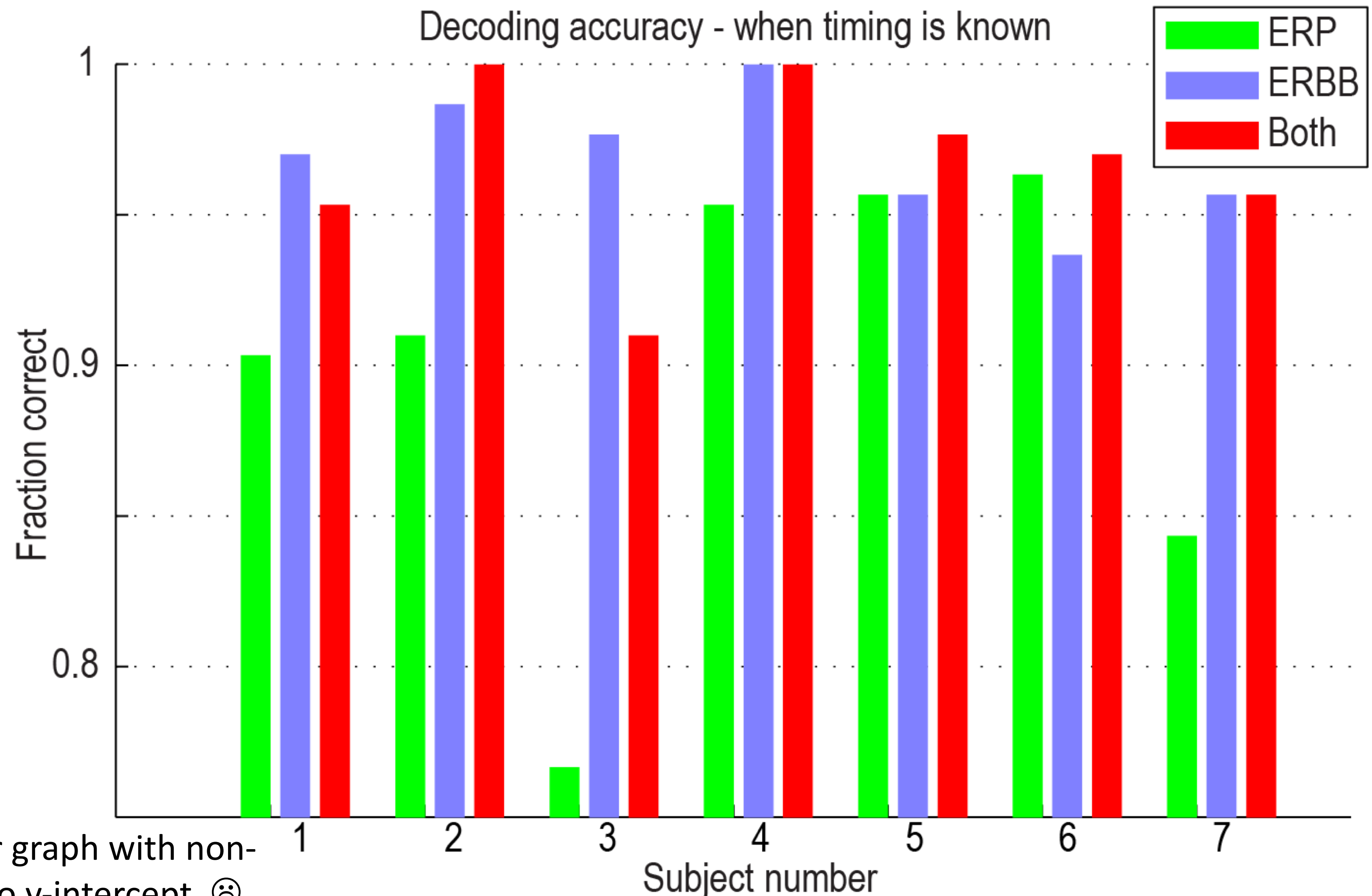
- Kai J Miller has made available a library of ECoG data he collected. Link.
- Today we will use one of the datasets from that library, found in the "faces_basic" folder.
  - Miller, Kai J., Gerwin Schalk, Dora Hermes, Jeffrey G. Ojemann, and Rajesh PN Rao. "Spontaneous decoding of the timing and content of human object perception from cortical surface recordings reveals complementary information in the event-related potential and broadband spectral change." PLoS computational biology 12, no. 1 (2016): e1004660.
  - **Ethics statement**: All patients participated in a purely voluntary manner, after providing informed written consent, under experimental protocols approved by the Institutional Review Board of the University of Washington (#12193). All patient data was anonymized according to IRB protocol, in accordance with HIPAA mandate. These data originally appeared in the manuscript “Spontaneous Decoding of the Timing and Content of Human Object Perception from Cortical Surface Recordings Reveals Complementary Information in the Event-Related Potential and Broadband Spectral Change” published in PLoS Computational Biology in 2016



Broadband: Power in 50-300 Hz

Event-Related Potential: Stimulus-locked time-domain signal

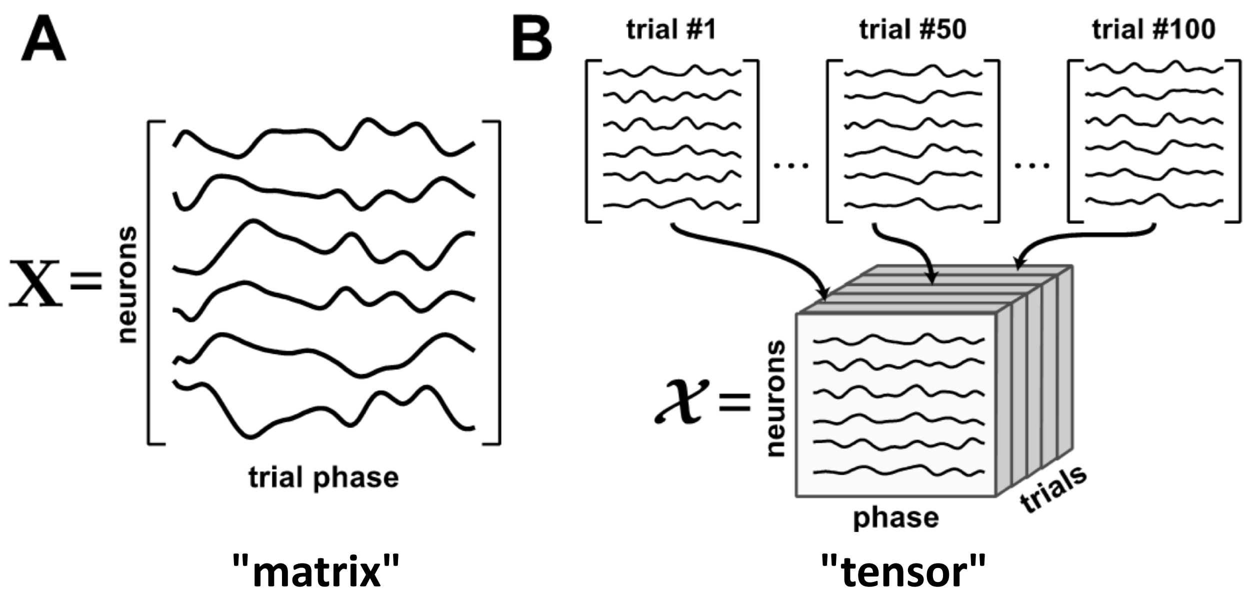
3 Conditions:
* Inter-Stimulus Interval
* face
* house



Bar graph with non-zero y-intercept. ☹

Data Structure

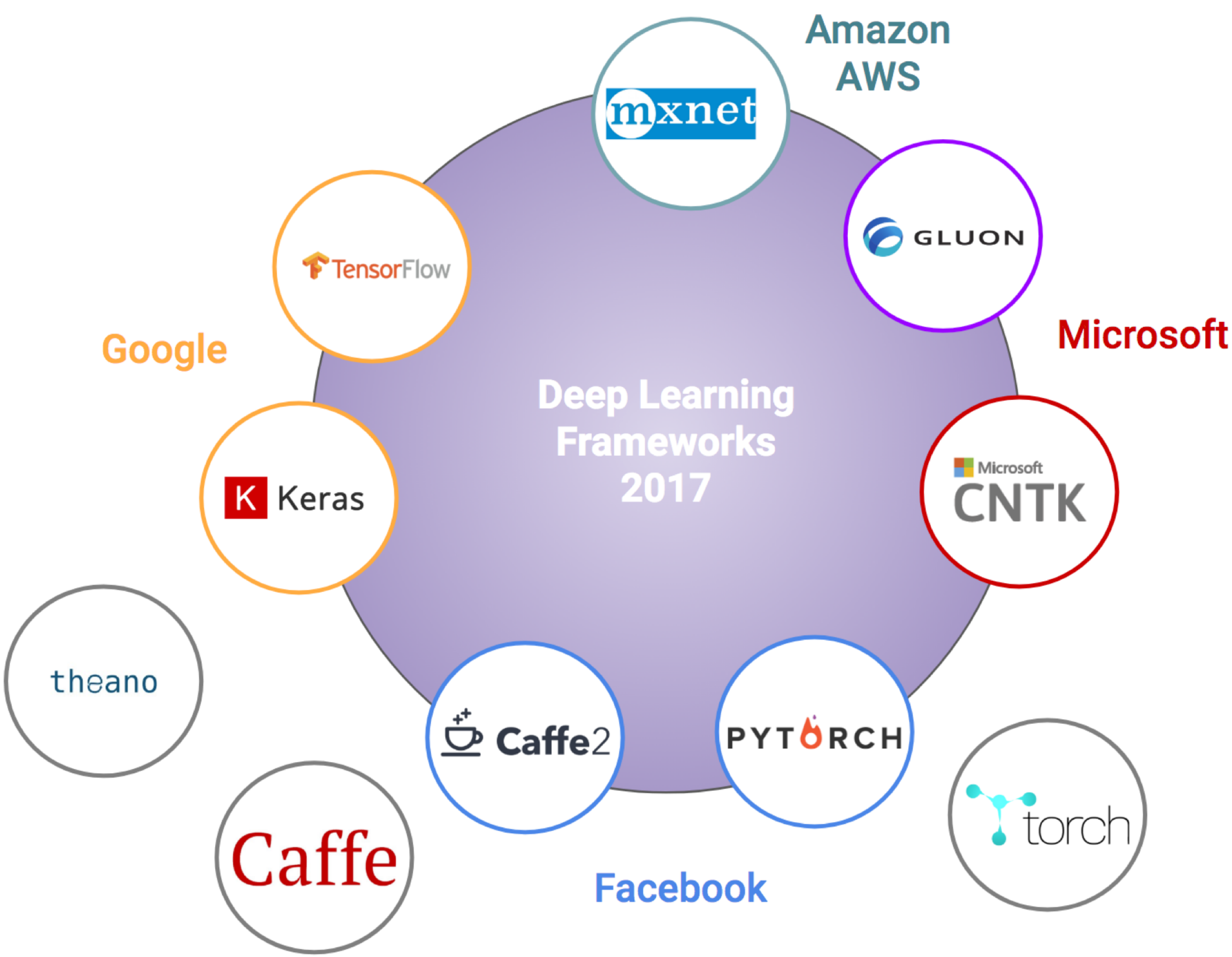
- Review 01_02_data_import.ipynb from Part 1
  - `The 'signals' chunk has data with shape (603, 17, 31).`
  - `(trials, samples, channels)`



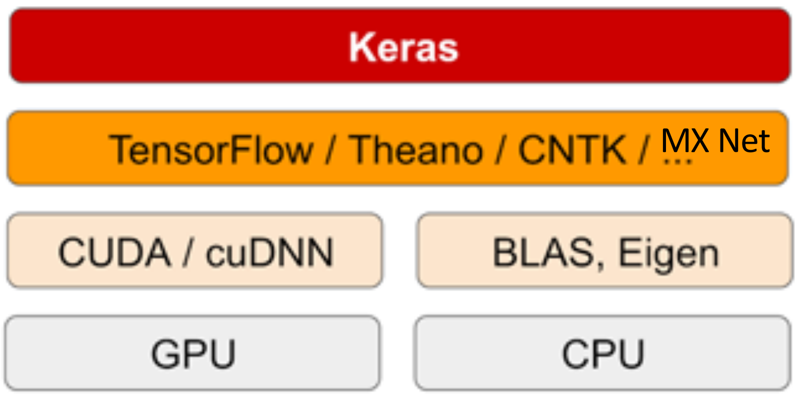
- Show preprocessing script

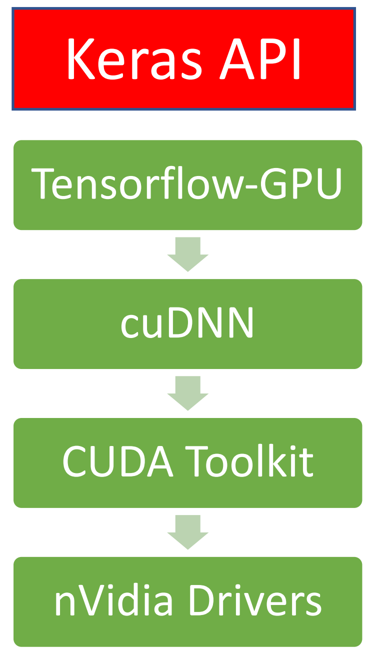
- 02_01_basic_lda.ipynb

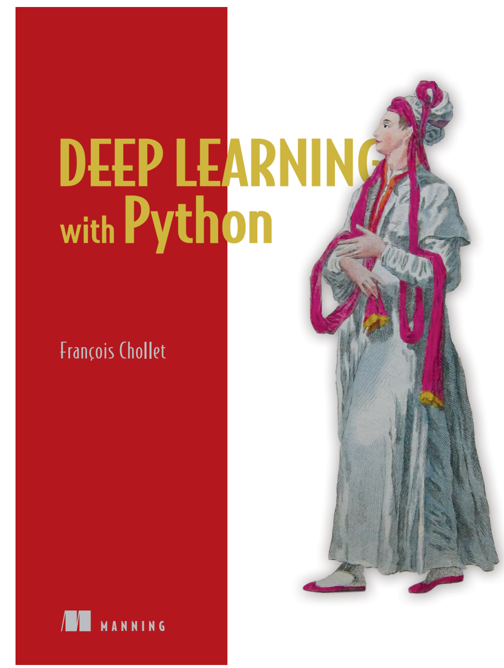
Creating and training a NN model with Tensorflow and Keras

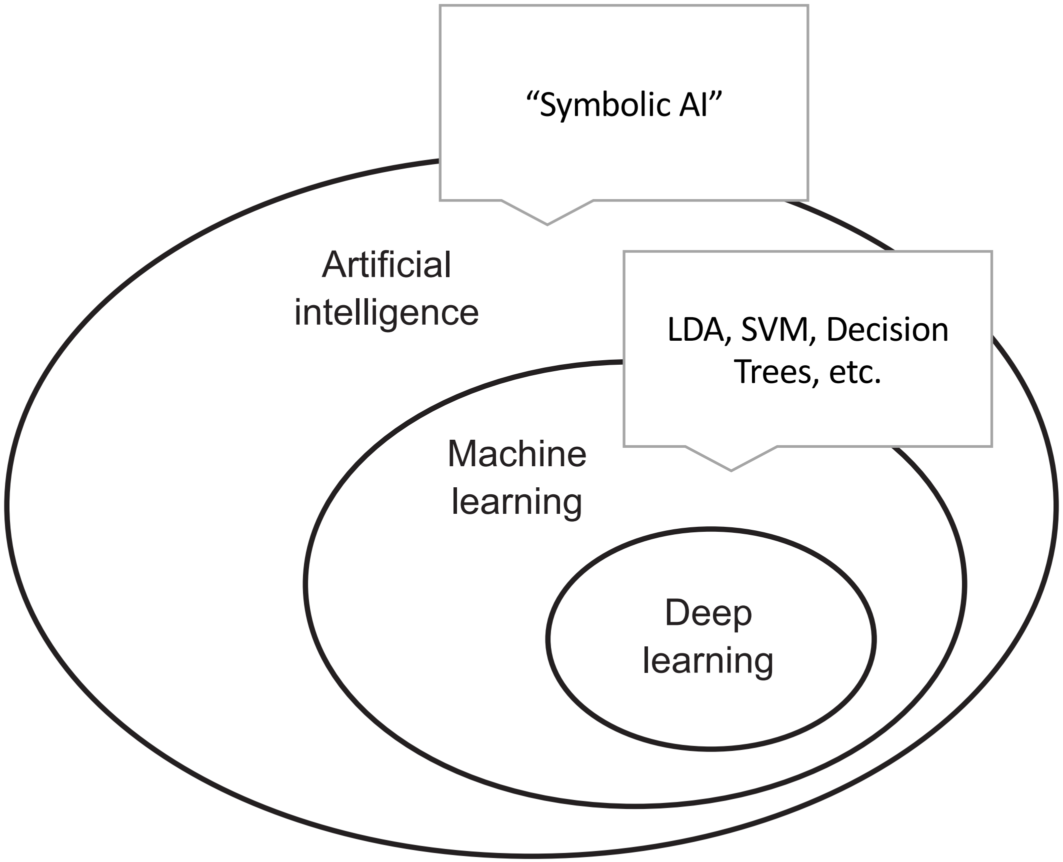


Creating and training a NN model with Tensorflow and Keras









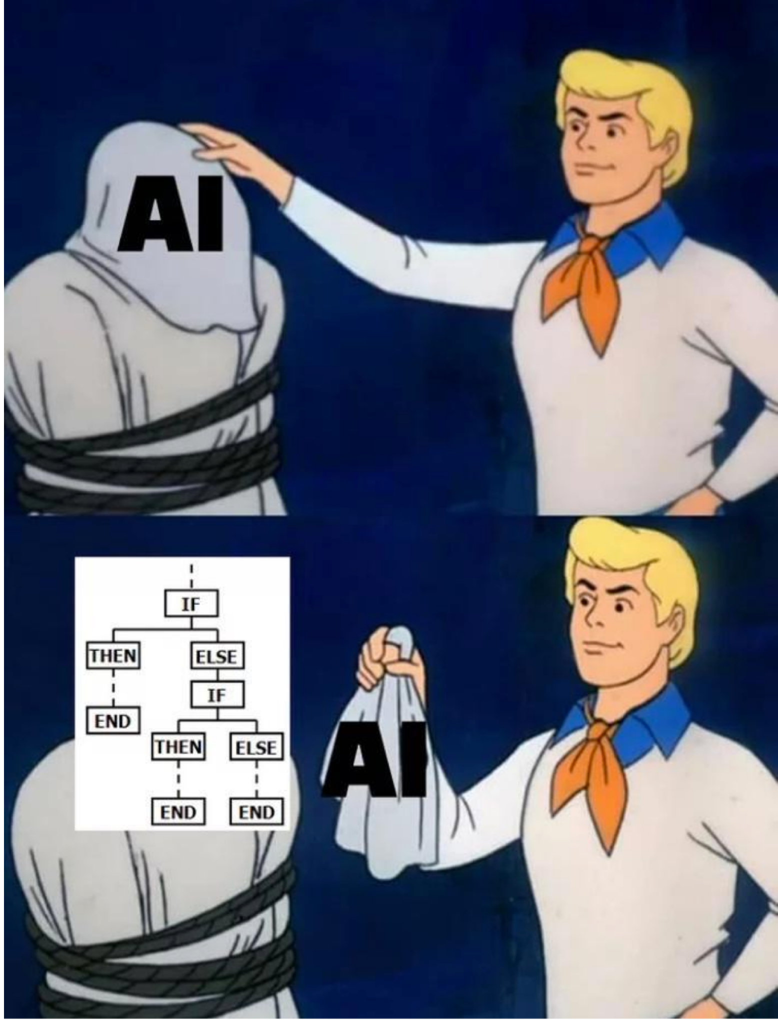


Figure 1.1 Artificial intelligence, machine learning, and deep learning

Deep Learning with Python by Chollet

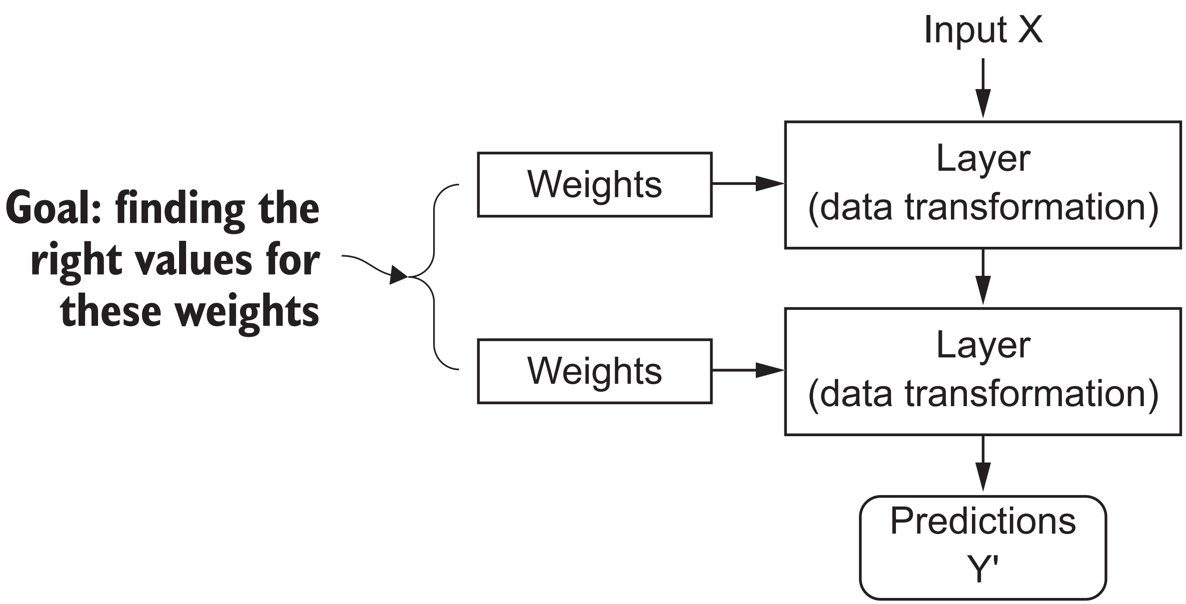


Figure 1.7 A neural network is parameterized by its weights.

Deep Learning with Python by Chollet

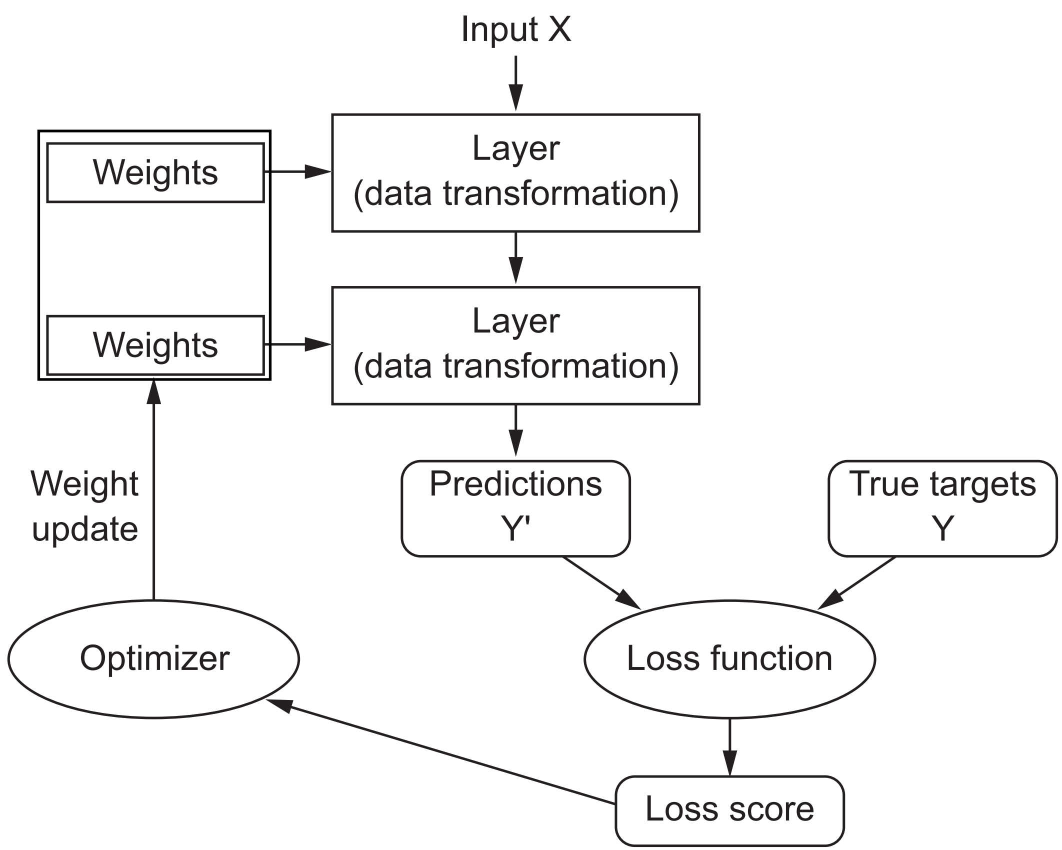
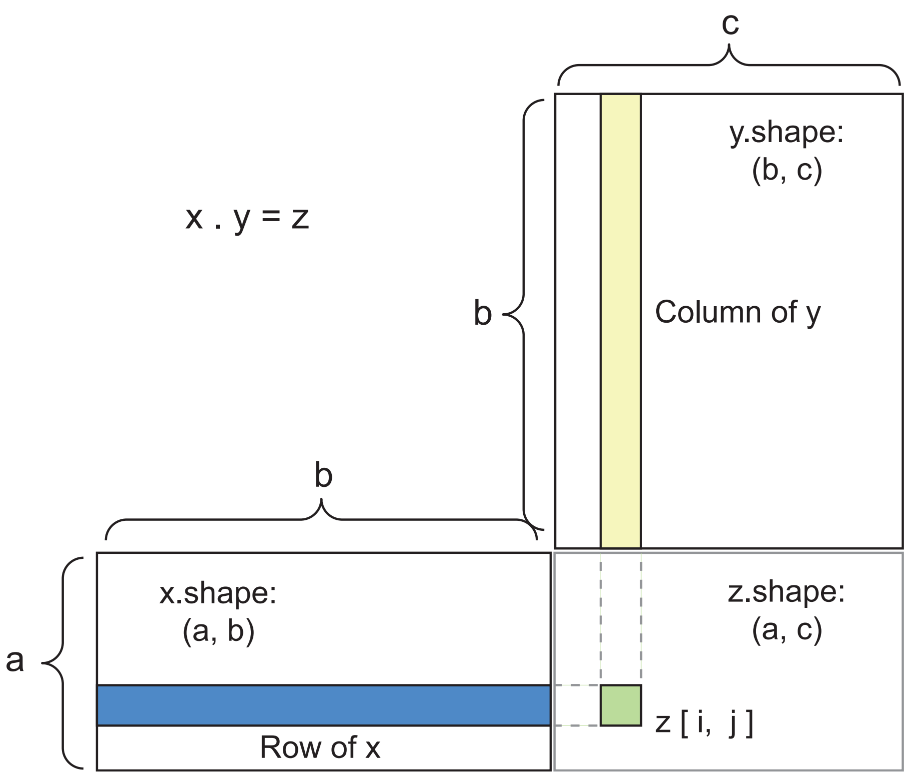


Figure 1.9 The loss score is used as a feedback signal to adjust the weights.

Deep Learning with Python by Chollet



Each transformation layer performs two operations:

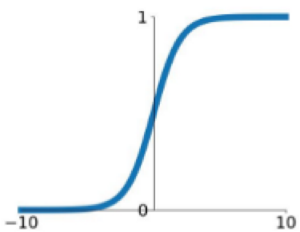
- A tensor dot product (data . Weights represented by x . y)
- An activation function
  (non-linear)

x = np.arange(-10, 11, 1), y = f(x), where f is …

Activation Functions

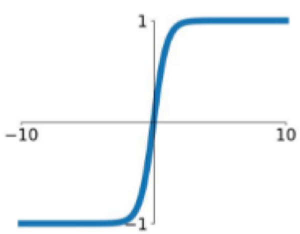
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



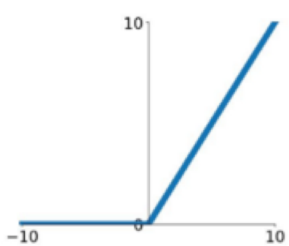
tanh

$$\tanh(x)$$



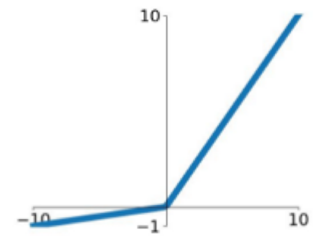
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

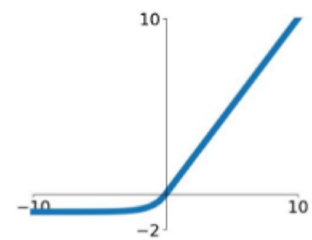


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Each transformation is a geometrical operation, changing the representation of the data to something more useful (less lossy).

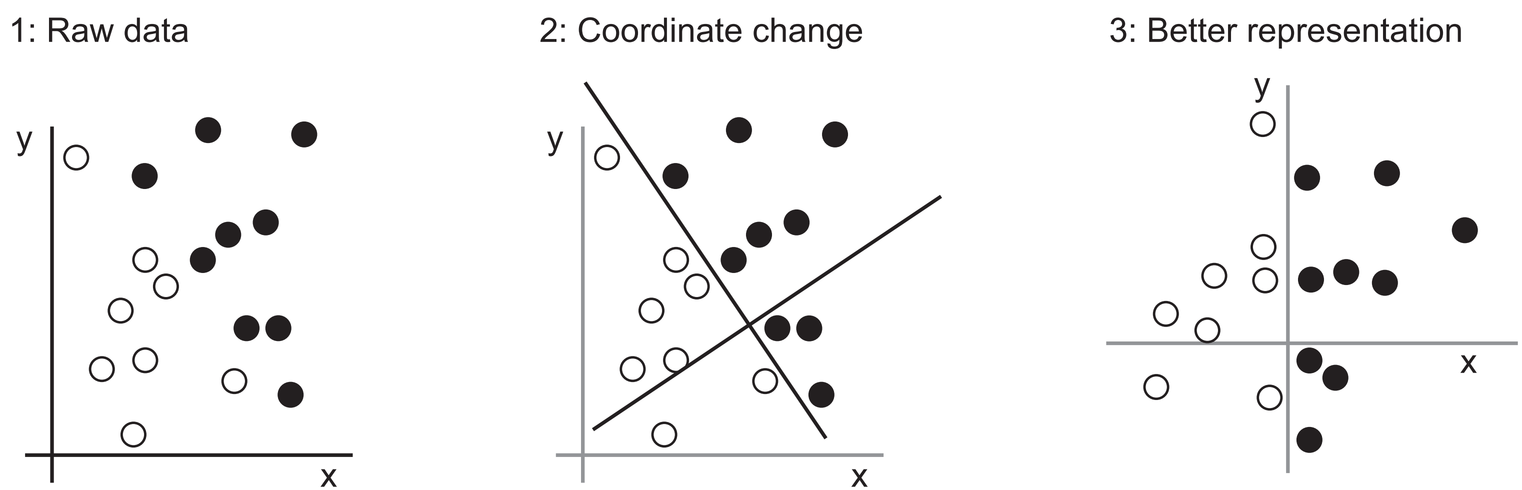


Figure 1.4 Coordinate change

Deep Learning with Python by Chollet

- Each one of many (non-linear) transformations is a step in a sequence.

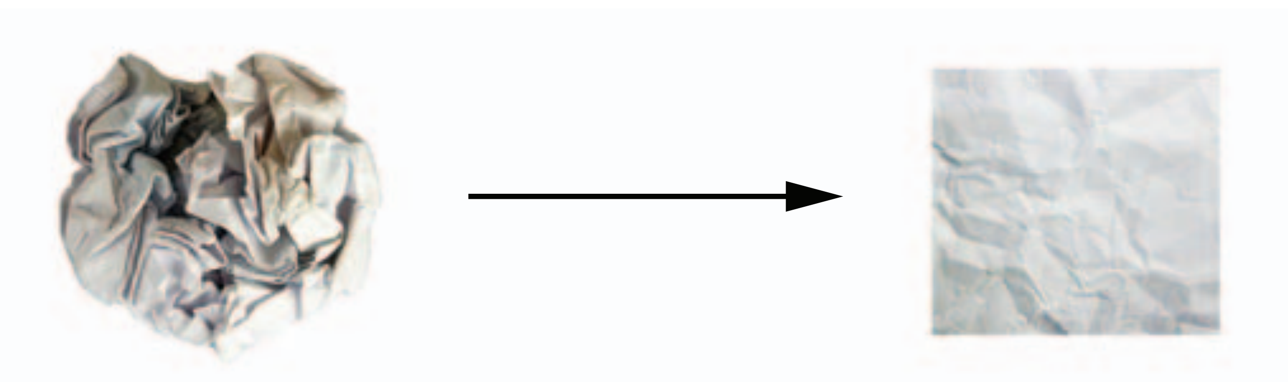
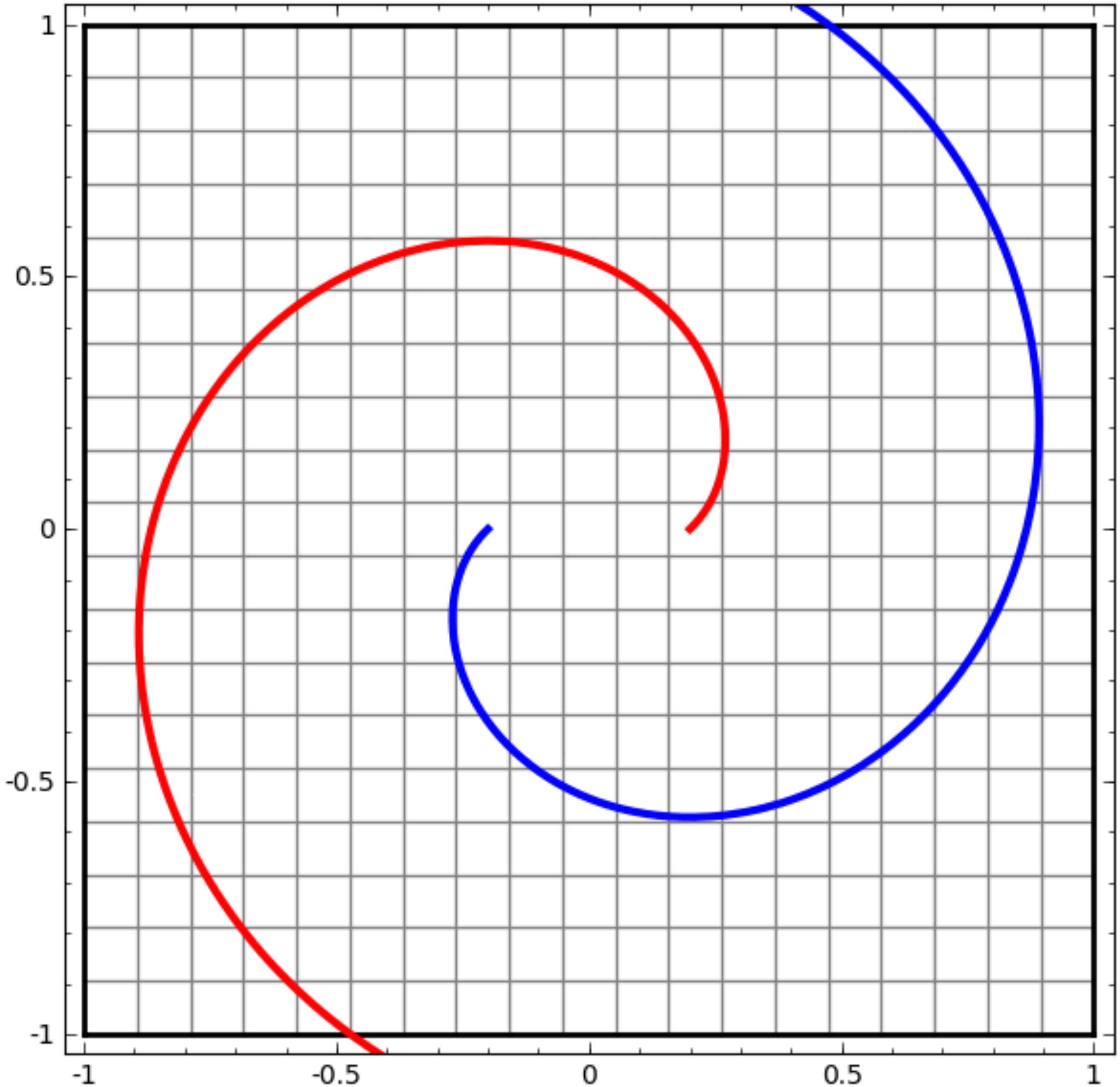


Figure 2.9 Uncrumpling a complicated manifold of data

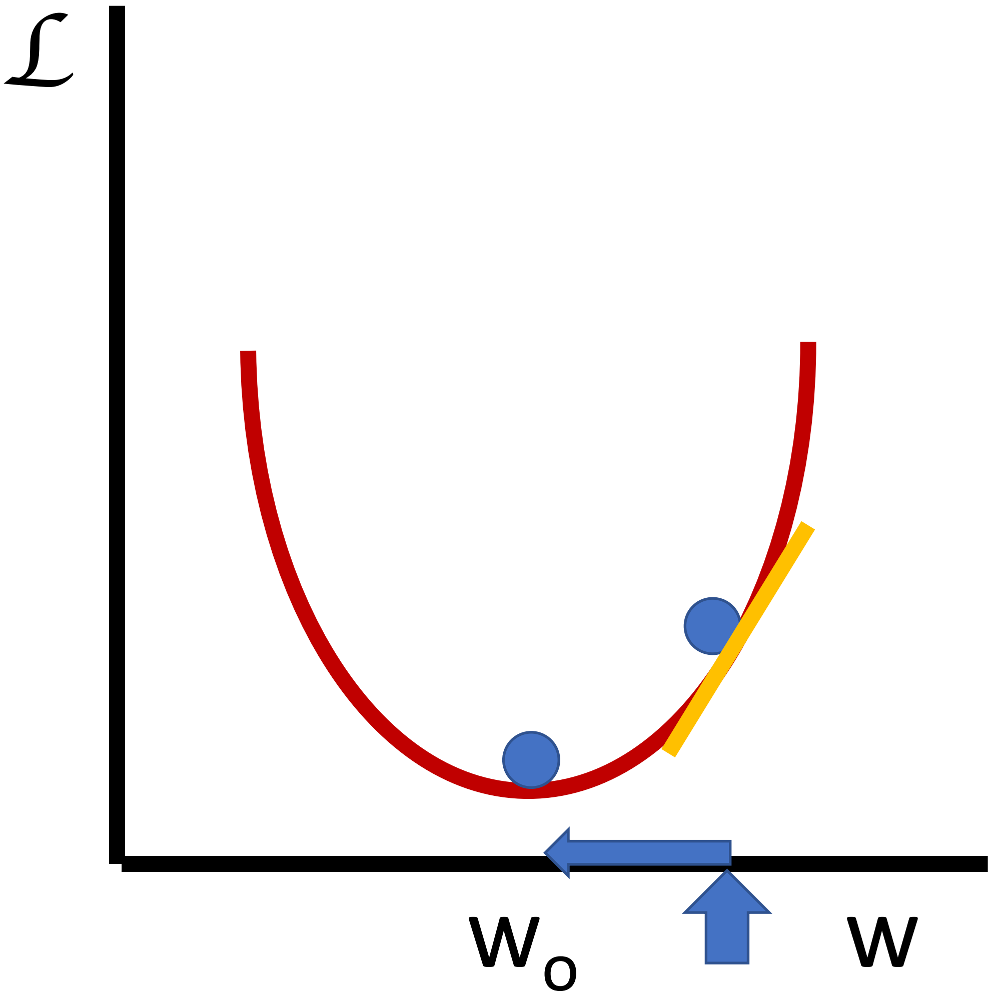
Deep Learning with Python by Chollet



Source

- 02_02_first_neural_net.ipynb

Loss & Gradients

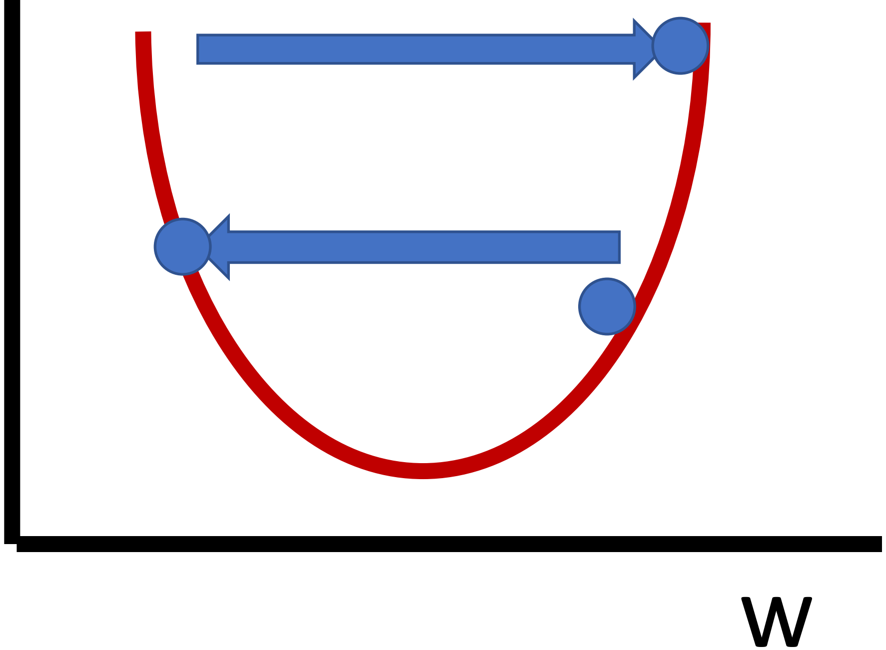


- Assume w_o is optimal value of w with minimum loss.
  - $\mathcal{L} = F(w - w_o)$
    $\mathcal{L} = (w - w_o)^2$
- Initialize w to some random value
- Calculate loss
- Calculate weight update
  - grad = $d\mathcal{L} / dw$
  - $w = w - \alpha * grad$

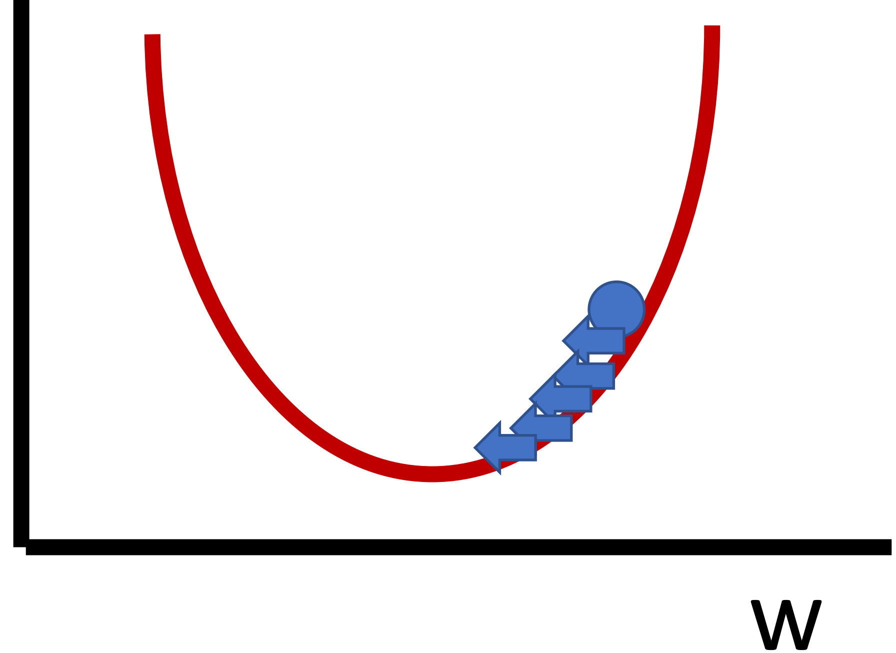
“Gradient Descent”

Learning Rate

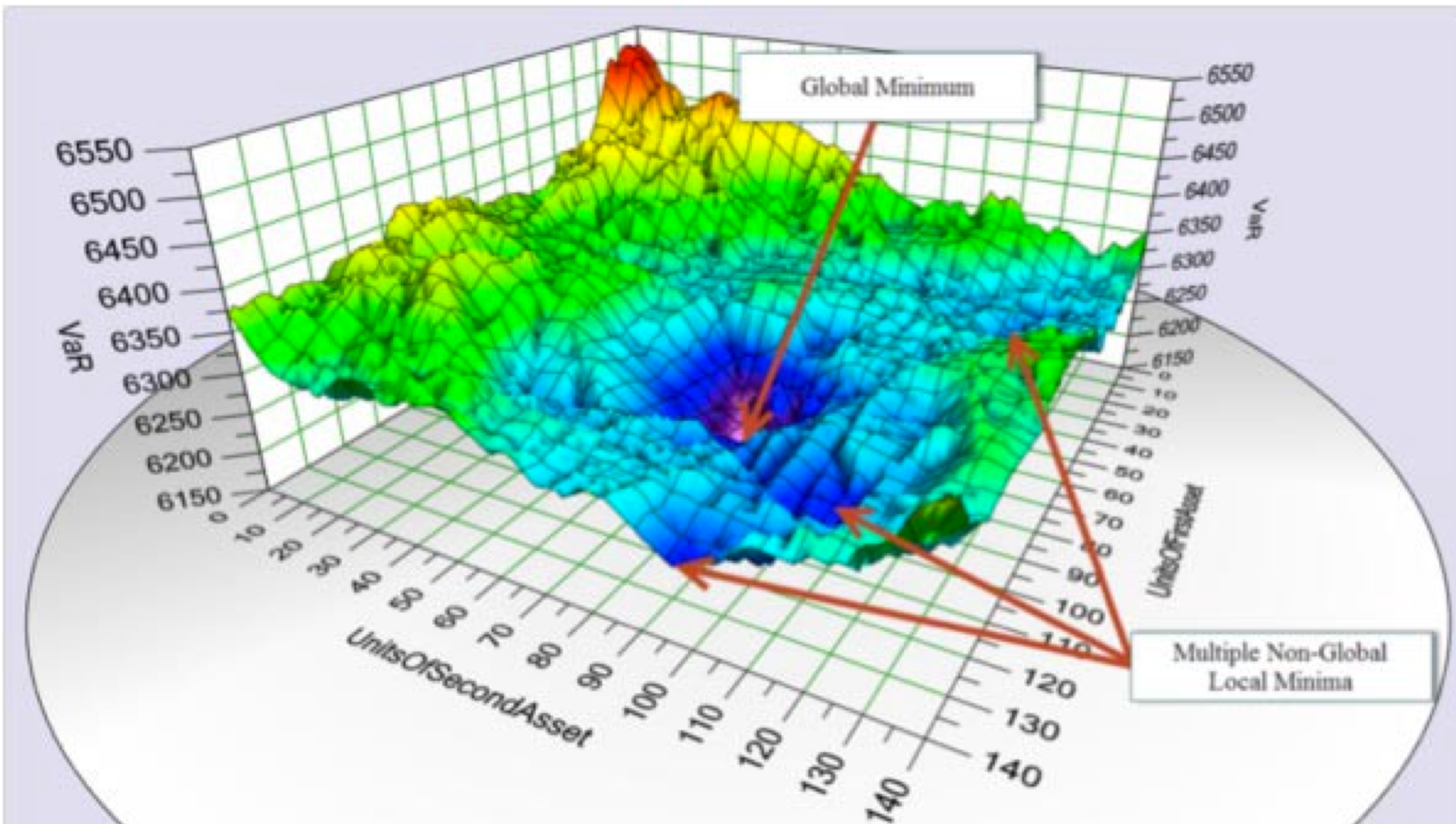
- Too High
  - Explodes!



- Too Low
  - Takes too long
  - More likely to get stuck in local minimum.



Loss & Gradients



Learning Rate and Optimizers

- Learning Rate Strategies
  - Find “Goldilocks” rate
  - Learning rate schedule
    - E.g. exponential decay
  - Optimizer
  - Combination of the above

- Optimizers
  - Update learning rate based on characteristics of loss/gradient history

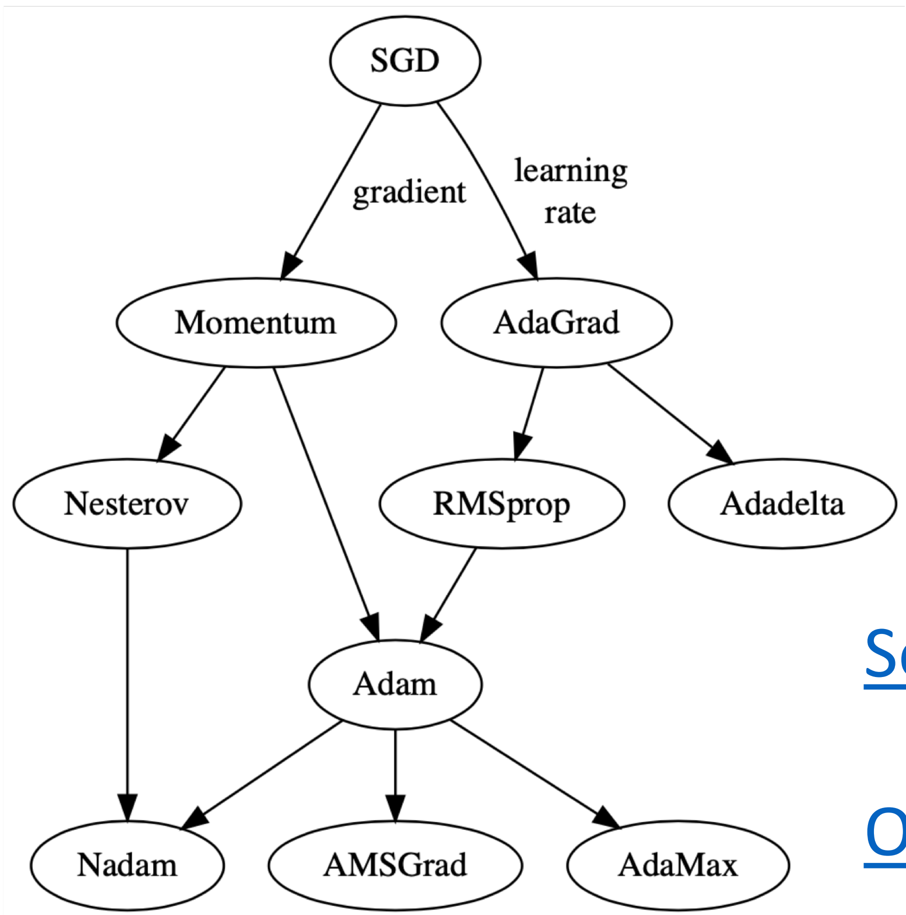
$$w_{\text{new}} = w - \alpha \frac{\partial L}{\partial w}$$

  - E.g. RMSprop

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t}$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[\frac{\partial L}{\partial w_t} \right]^2$$

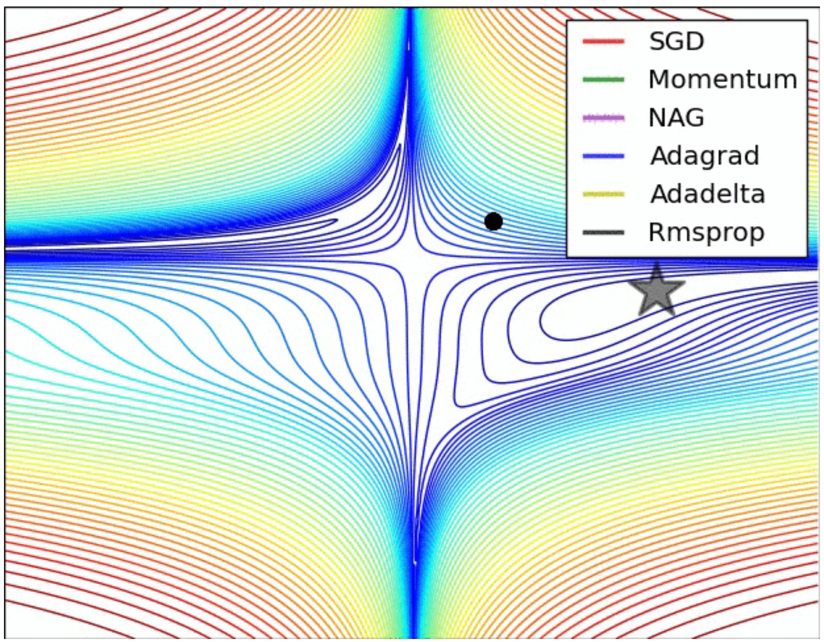
Optimizers



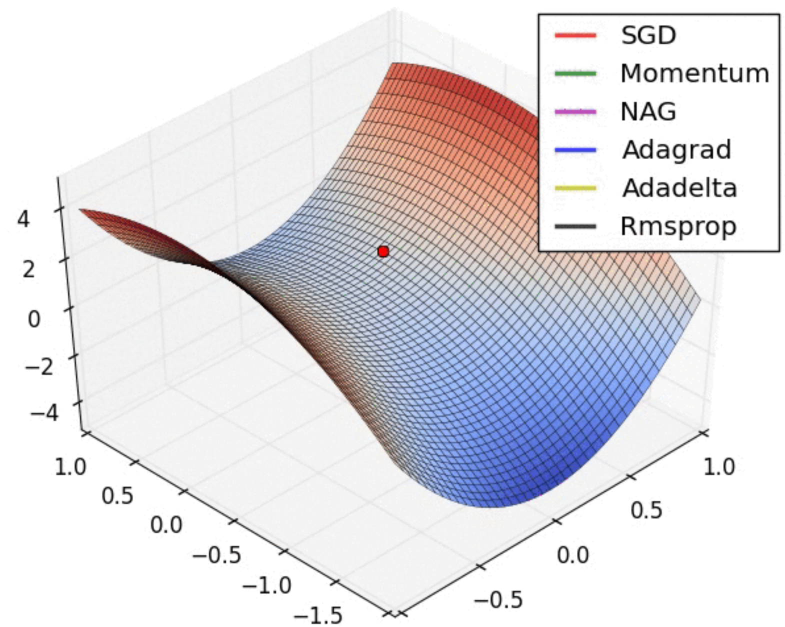
[Source]

[Other useful ref]

Optimizers



SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop



SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop
4
2
0
−2
−4
1.0
0.5
0.0
−0.5
−1.0
−1.5
1.0
0.5
0.0
−0.5

Back propagation

- When using a ‘deep’ model, how do we know which weights need modifying?
- See this
- Also this