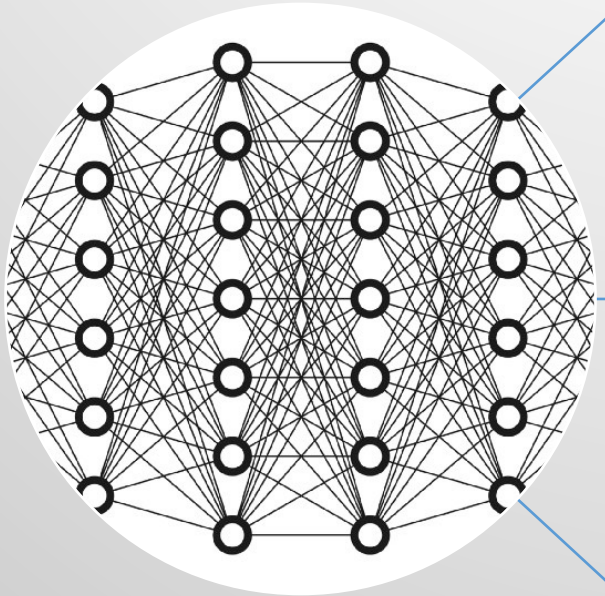
Natural Language Processing

Dr. Karen Mazidi

Part Six:
Deep Learning



Topics

- Deep Learning
- Keras/TensorFlow

Quizzes

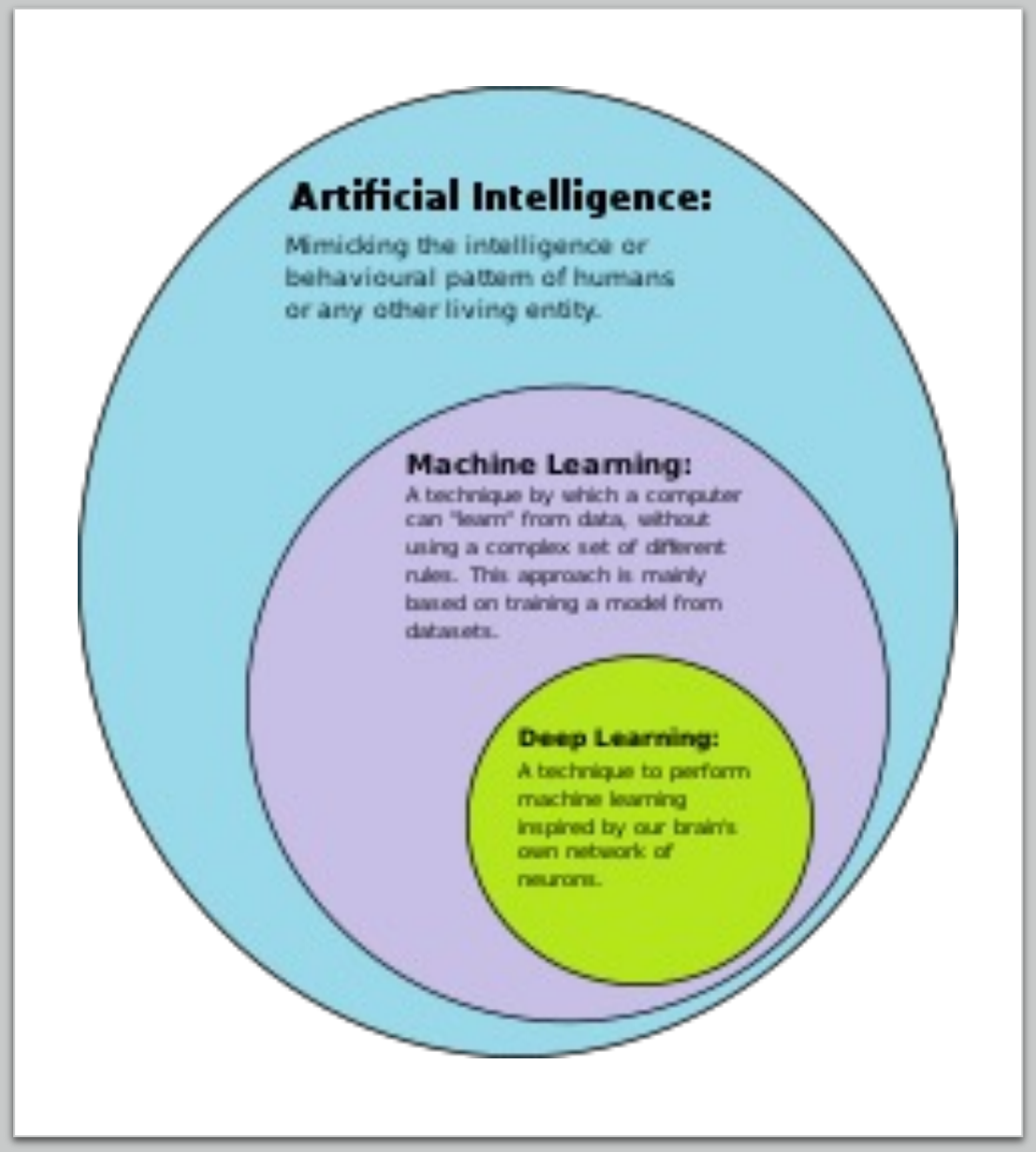
- Quiz on DL

Homework

- Homework: Text classification using DL

Deep learning

- A neural network with many hidden layers



Keras

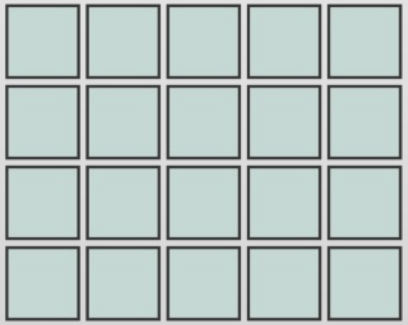
- Open-source API developed at Google
- March 2015 release
- 2017: Keras 2 release
  - integrated with TensorFlow 2

TensorFlow K Keras

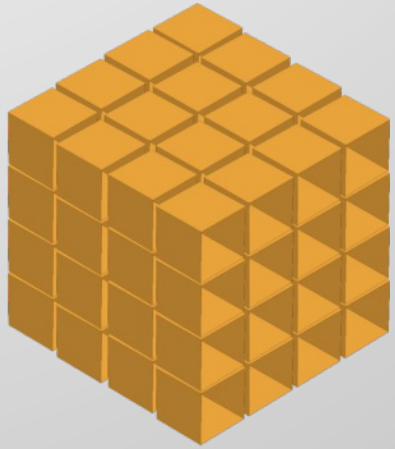
Data in Keras/TensorFlow

- Compatible with NumPy arrays, pandas data frames
- Tensors:
  - rank – number of axes
  - shape – tuple ( )
  - type – data type

RANK-1 TENSOR



RANK-2 TENSOR



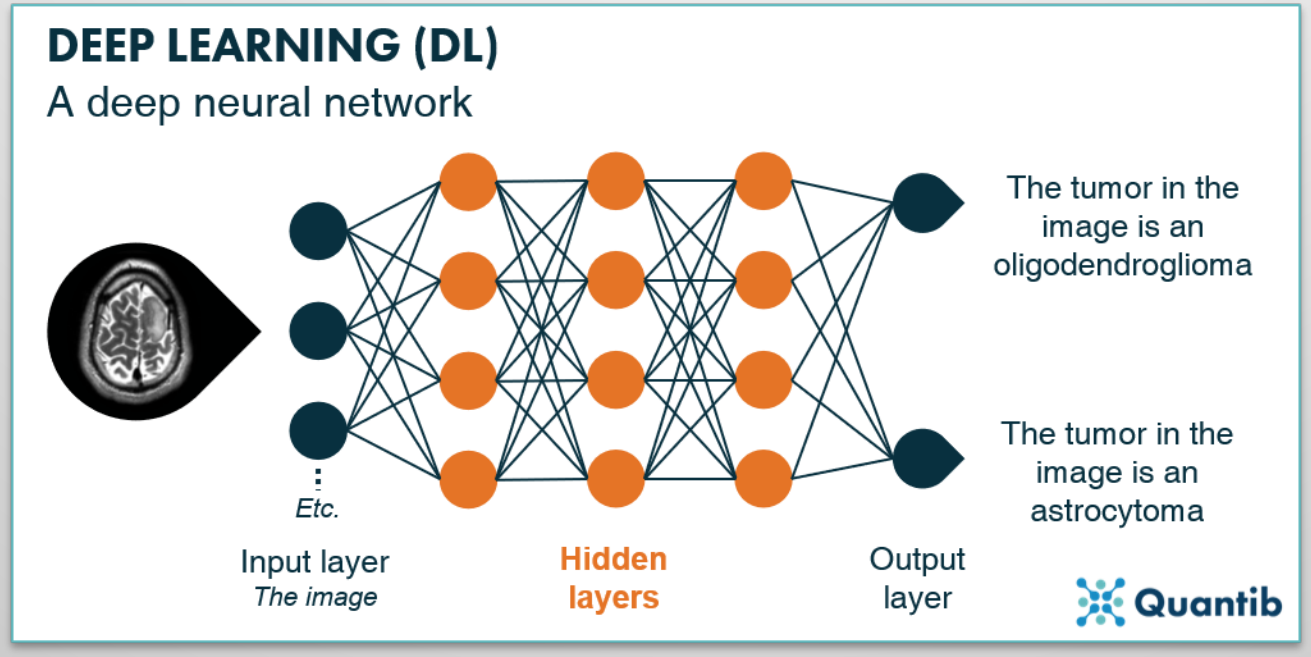
RANK-3 TENSOR

Data in Keras/TensorFlow

- scalar: 0D tensor
- vector: 1D tensor
- 2D tensor (samples, features)
- 3D tensor (samples, timesteps, features) – sequential data
- 4D tensor (samples, height, width, channels) – images
- 5D tensor (…, frames) - video

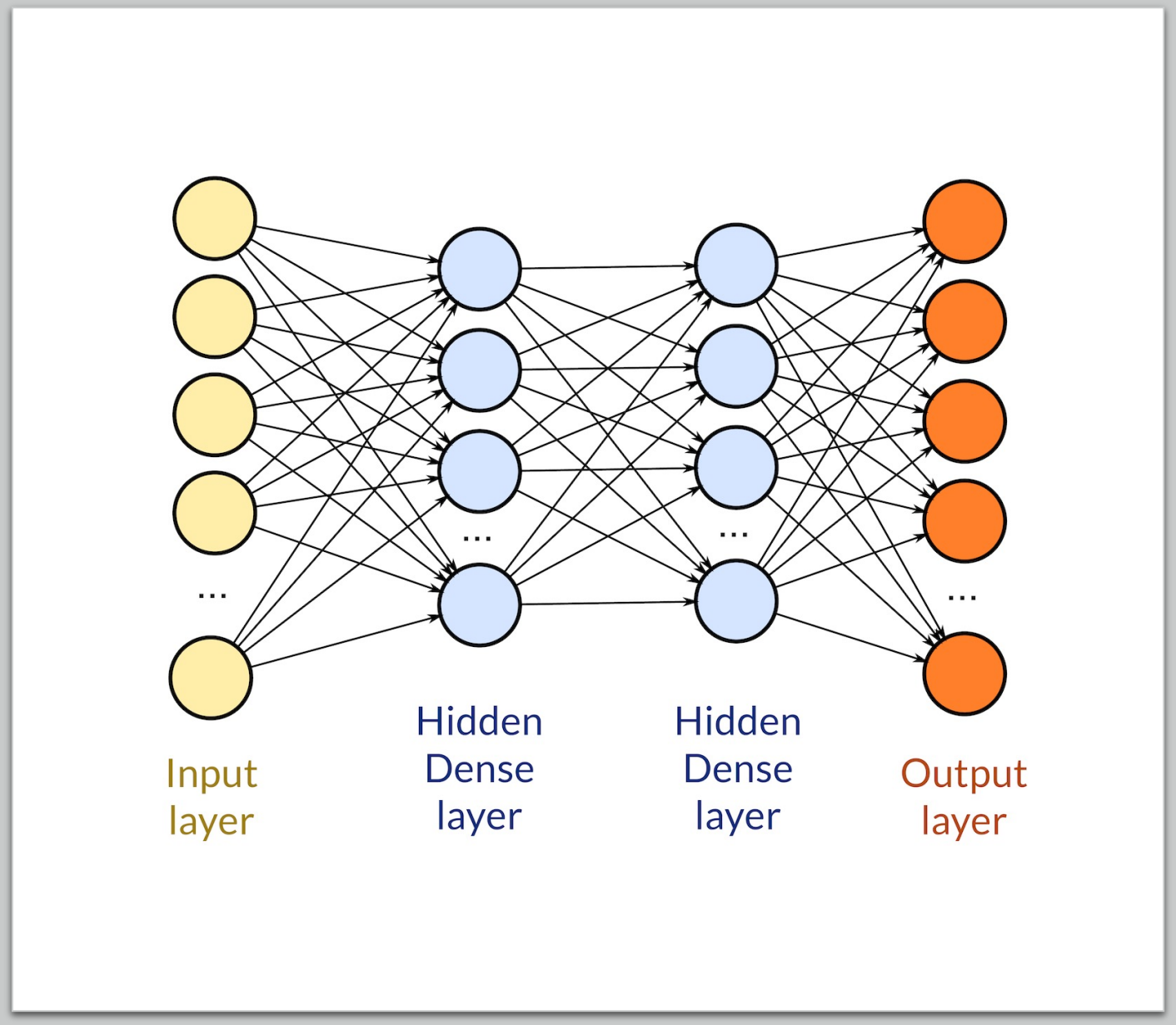
Models

- a model defines how layers are put together
- a layer represents a function that inputs tensors and outputs transformed tensors



Sequential model

- a Sequential model is a regular feed-forward network
- most hidden layers will be Dense (densely connected layers)



Code Example
Part 6 Chapter 23

- Dense Sequential on the IMDB data

Deep learning advantages

- learn increasingly complex representations layer by layer over many iterations
- intermediate representations are learned jointly, so that as one feature changes, all others are adjusted

Suggestions

- make sure subsequent layers have fewer nodes to prevent bottleneck
- the smaller your data, the simpler the model should be
- Try to prevent overfitting:
  - regularization (L1, L2) to shrink weights
  - dropout – randomly set weights to 0

Activation functions

- linear is the default
- sigmoid: 0, 1 classification
- softmax: multi-class classification
- ReLu for intermediate layers

Loss functions

- Regression: MSE or MAE
- Binary classification: binary crossentropy
- Multi-class classification: categorical crossentropy

Code Example
Part 6 Chapter 23

- Functional – using different API
- Keras on the spam data

TensorFlow Functional API

- Keras:

```
model = models.Sequential()
model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

- Functional:

```
inputs = keras.Input(shape=(10000,))
dense = layers.Dense(16, activation='relu')

x = dense(inputs)
x = layers.Dense(16, activation='relu')(x)
outputs = layers.Dense(1)(x)

model = keras.Model(inputs=inputs, outputs=outputs,
                    name='functional_model')
```

TensorFlow Functional API

- Steps:

```
1. define the inputs
2. add a dense layer
3. add another dense layer
4. define the output layer
5. put all the layers together in a model
```

```
inputs = keras.Input(shape=(10000,))
dense = layers.Dense(16, activation='relu')

x = dense(inputs)
x = layers.Dense(16, activation='relu')(x)
outputs = layers.Dense(1)(x)

model = keras.Model(inputs=inputs, outputs=outputs,
                    name='functional_model')
```

Deep Learning

Why does it work so well?

Deep learning

- Basically, it’s just a parametric model trained with gradient descent
- with a sufficiently large model and a sufficient number of examples, a DL model can approximate many functions
- Feynman, talking about the universe:
  - It’s not complicated, it’s just a lot of it.

Deep learning

- every training example is a vector – a point in geometric space
- each layer in a deep learning model performs one simple geometric transformation on the data
- the chains of layers form one complex geometric transformation, broken into a series of simple ones
- this complex transformation attempts to map the input space to the target space, one point at a time
- since the loss function must be differentiable, this means that the transformation from inputs to outputs must be smooth and continuous

Deep learning

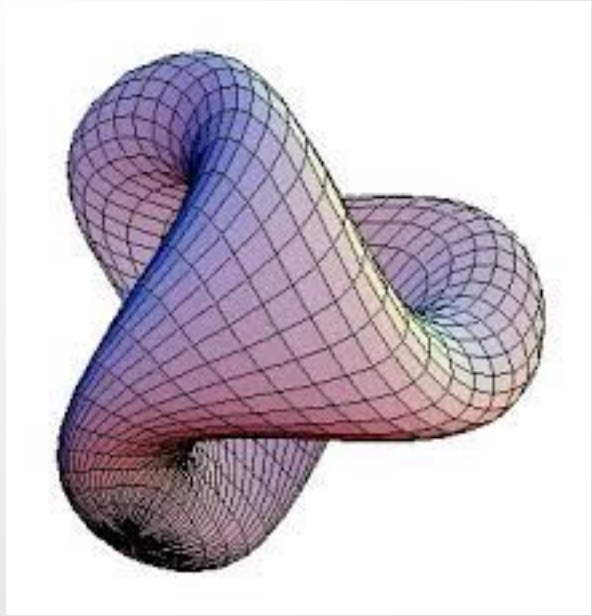
- key process: meaning is derived from the pair-wise relationship between vectors (words in a language, pixels in an image …)
- these relationships can be captured by a distance function
- vectors are efficient computationally for computers, but we have no idea how the brain performs its computations
- neural networks are not like the brain
- better terms: layered representations learning

Deep learning

- visualize DL learning as a person trying to uncrumple a paper ball
- the crumpled paper ball is the manifold of the input data that the model starts with
- each movement of uncrumpling is a geometric transformation
- DL models are mathematical machines for uncrumpling complicated manifolds of high-dimensional data

Manifold hypothesis of DL

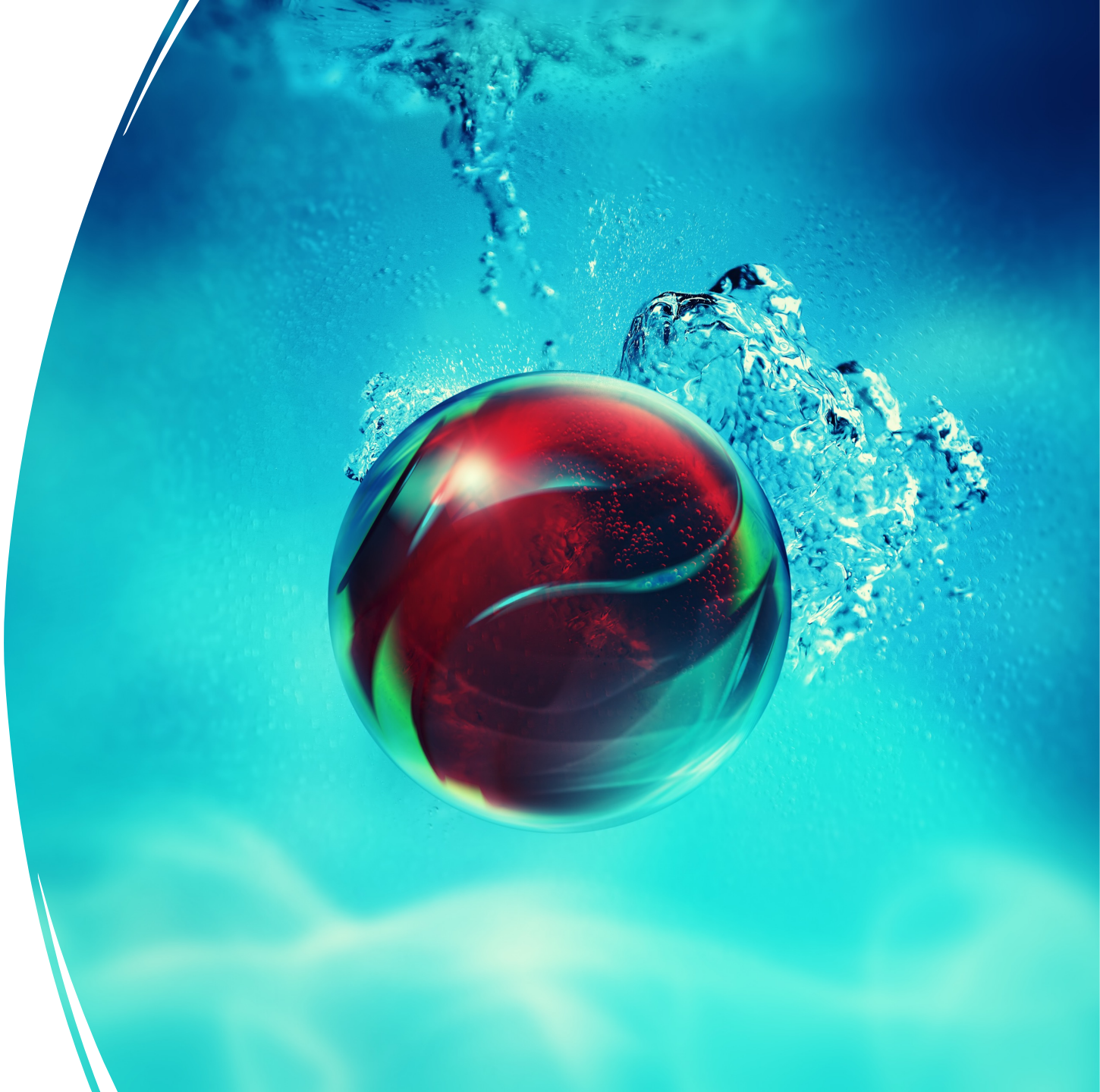
a manifold is a topological space that locally resembles a Euclidean space near each point



- many high-dimensional real-world data sets lie along low-dimensional manifolds inside the high-dimensional space
- this lower-dimensional space would need fewer variables to describe
- may also describe the effectiveness of dimensionality-reduction techniques

TF under the hood

- More TF here: https://github.com/kjmazidi/Intro-to-Deep-Learning



Essential points to note

- Keras makes it easy to build and train deep learning models
- The TensorFlow Functional API enables more complex models

Next topic

Deep Learning Variations