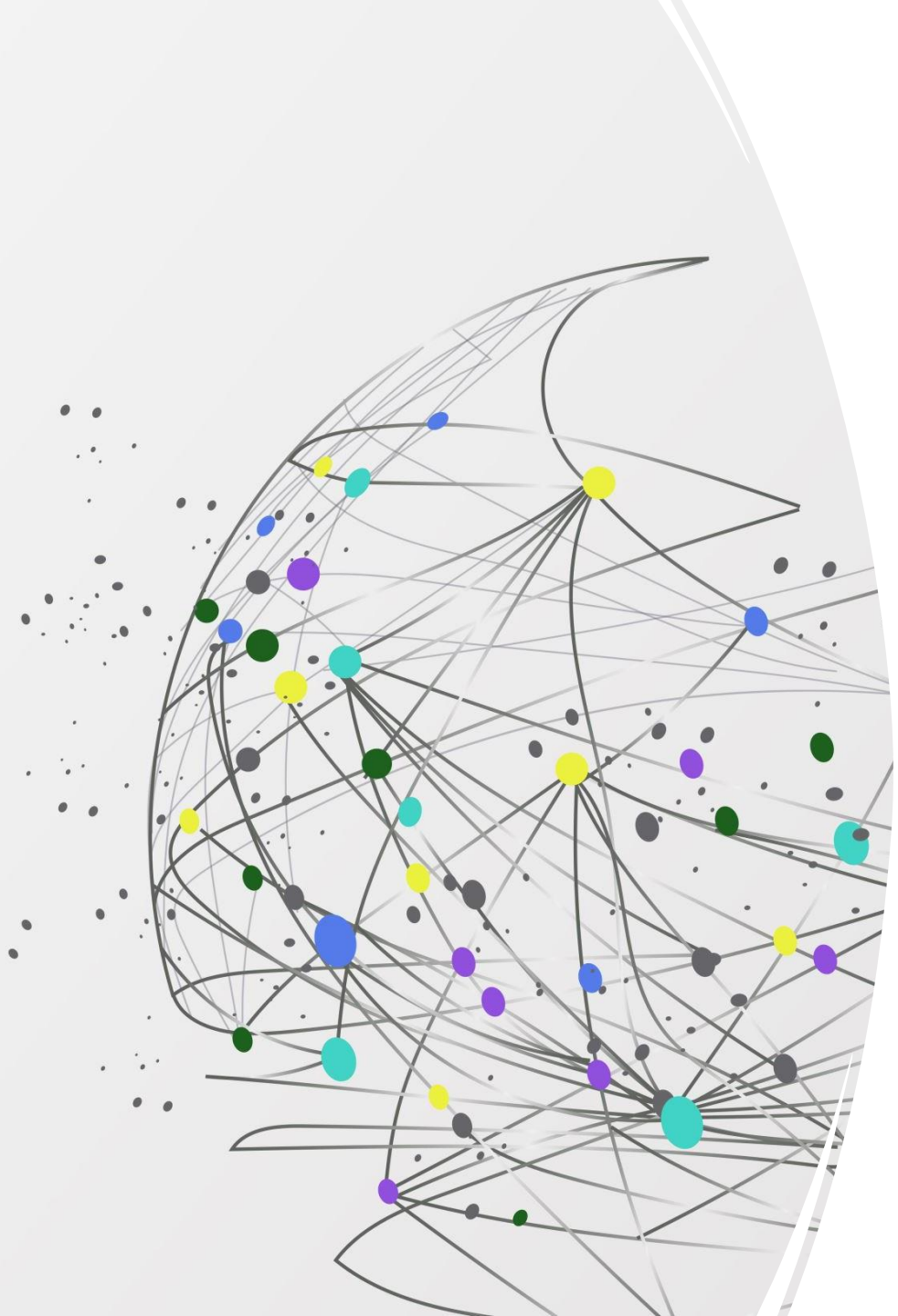


DSI DEEP LEARNING SHORT COURSE

Practical Introduction to DNNs with Keras

24 June 2024

Prof Mandla Gwetu [mgwetu@sun.ac.za]

Dr Gabriel Kallah-Dagadu [KallahDagaduG@ukzn.ac.za]

Dr Mohanad Mohammed [MohammedM1@ukzn.ac.za]

PRESENTATION OUTLINE

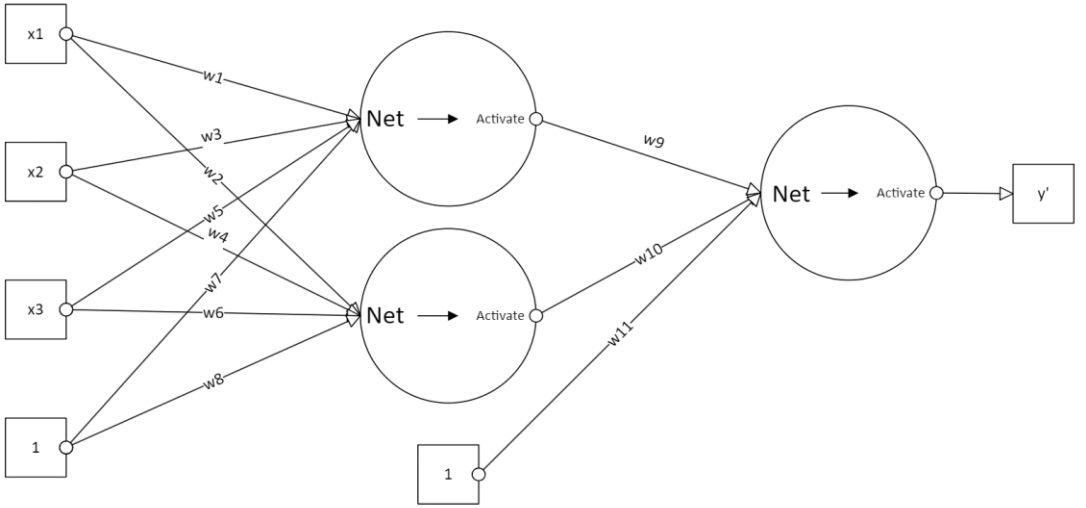
- Assumptions
- Theoretical concepts
- Keras
- Health toy dataset
- Keras DNN exercises

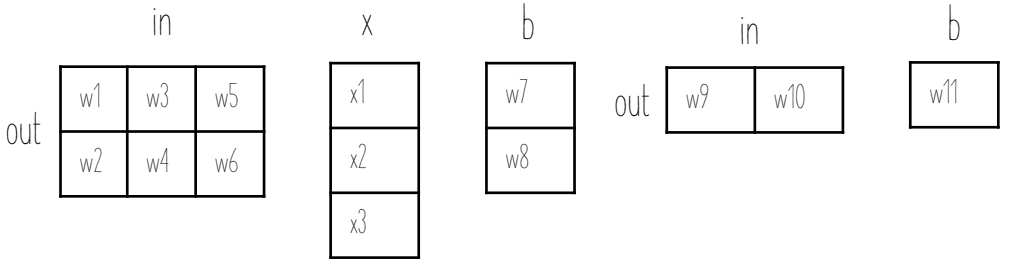
ASSUMPTIONS

- A Google account (gmail) for access to Google drive.
- Familiarity with Google colab.
- Familiarity with Python.
- Machine learning and deep learning foundations.

THEORETICAL CONCEPTS

- The chain-based feedforward computation of a DNN can be represented as: $h^{(l)} = g^{(l)}(W^{(l)\mathrm{T}} h^{(l-1)} + b^{(l)})$, where h, W, g and b represent the hidden layer, Weight matrix, activation function and bias, respectively and (l) shows that the components are at the l^{th} layer of the NN.
- Backpropagation guides the update of weights over multiple epochs using a loss function, ground truth and an optimizer.





KERAS

- The Keras python library allows for simplified DNN implementation.
- The packages has classes and functions that facilitate model construction, configuration, training, testing and deployment.
- There are two main APIs for model construction: sequential and functional.
- The sequential API is meant for traditional DNNs while the functional API allows for specialized tasks such as multiple inputs/outputs and layer sharing.
- The following functions characterize the typical DNN implementation workflow after construction: compile (configuration), fit (training), get_weights and set_weights (reading and hardcoding weights), summary (describe), evaluate (test) and predict (feedforward for regression or classification outputs).
- Other python libraries can be used to complement Keras functionality. For example, the scikit learn library has functions for splitting datasets in to training and testing sets (train_test_split) and it comes with inbuilt toy datasets.

SKLEARN BREAST CANCER DATASET

- Breast cancer dataset - consists of features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. It includes characteristics of the cell nuclei present in the image.
  - 569 instances to model tumour malignancy based on descriptive statistics of 10 tumour measurements.
  - Inputs: the mean, standard error and mean of the 3 largest values from 10 tumour measurements yielding 30 features.
  - Outputs: malignant or benign.

DNN IMPLEMENTATION EXERCISE

- Load the sklearn breast cancer dataset in a google colab notebook.
- Partition the dataset using a 75:25 training set to test set split.
- Design and build an appropriate DNN model for the dataset.
- Configure the model with an appropriate loss function and optimizer.
- Train the model using the training set.
- Adjust parameters (e.g. number of epochs) or architecture to try improve training performance, if necessary.
- Read the model's weights then draw the network.
- Test your model's performance.
- Search for this (or another) dataset online then save it in your google drive and update the notebook to load the dataset from google drive.