



ICTSS00120 - Artificial Intelligence Skill Set

Session 3: Essentials of ML and DL Technologies

Lecturer: Jordan Hill

Learning Objectives

- Introduction to the sci-kit learn package.
- Understand supervised vs. unsupervised learning.
- Introduction to deep learning (DL) and its place within AI.
- Learn about regression and classification techniques.
- Explore clustering and dimensionality reduction techniques.
- Understand reinforcement learning.
- Introduction to various deep learning architectures.

Introduction to Sci-Kit Learn

- **What is Sci-Kit Learn?**

- Open-source machine learning library for Python.
- Built on NumPy, SciPy, and Matplotlib.
- Provides simple and efficient tools for data mining and data analysis.
- Accessible and reusable in various contexts.

- **Why Sci-Kit Learn?**

- Rich set of machine learning algorithms.
- Excellent documentation and community support.

Classical Machine Learning

Task Driven
↓
Supervised Learning
(Pre Categorized Data)

Classification
(Divide the socks by Color)
Eg. Identity Fraud Detection

Regression
(Divide the Ties by Length)
Eg. Market Forecasting

Data Driven
↓
Unsupervised Learning
(Unlabelled Data)

Clustering
(Divide by Similarity)
Eg. Targeted Marketing

Association
(Identify Sequences)
Eg. Customer Recommendation

Dimensionality Reduction
(Wider Dependencies)
Eg. Big Data Visualization

Obj: Predictions & Predictive Models

Pattern/ Structure Recognition

Supervised vs. Unsupervised Learning

Supervised Learning

- **Definition:** Models are trained using labeled data.
- **Goal:** Predict outcomes for new data (e.g., classification, regression).
- **Examples**

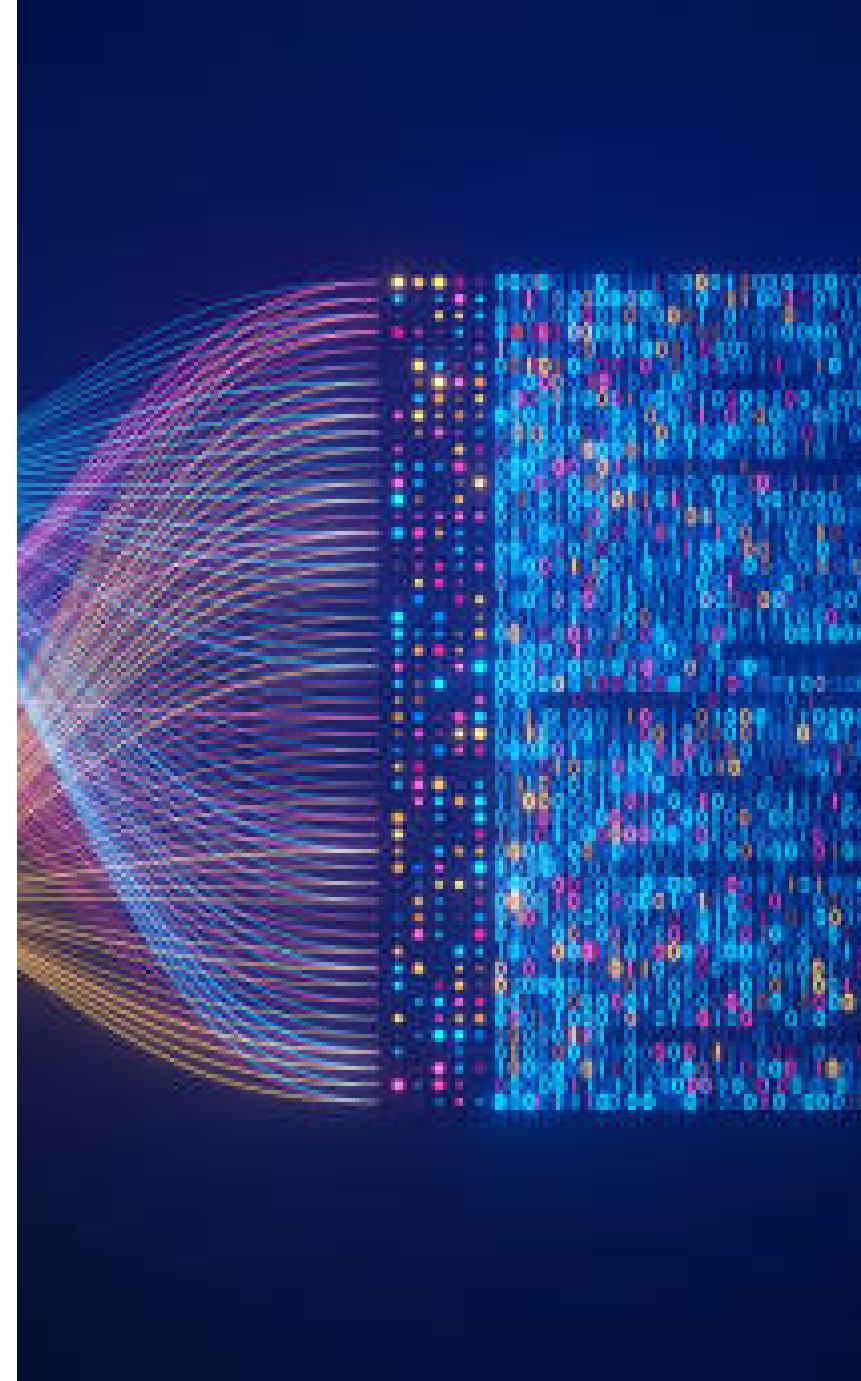
Unsupervised Learning

- **Definition:** Models are trained using unlabeled data.
- **Goal:** Discover hidden patterns or intrinsic structures.
- **Examples**

Introduction to Deep Learning (DL)

- **What is Deep Learning?**
 - Subset of ML involving neural networks with multiple (hidden) layers.
 - Capable of learning from vast amounts of data.
- **Place within AI:**
 - Enables more complex and abstract representations.
 - Powers advancements in computer vision, speech recognition, NLP, etc.

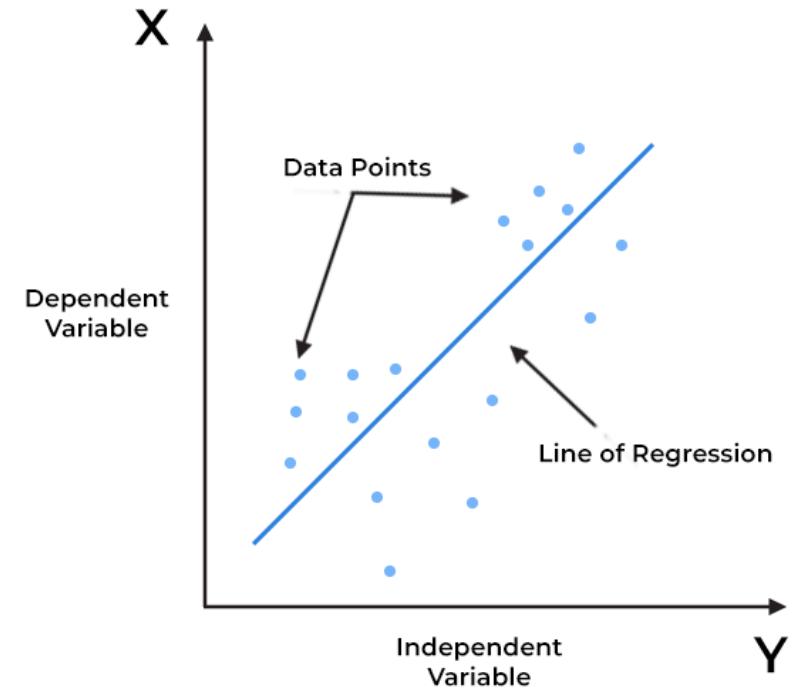
Key Tools: TensorFlow, PyTorch, Keras.



Regression & Classification - Supervised Learning

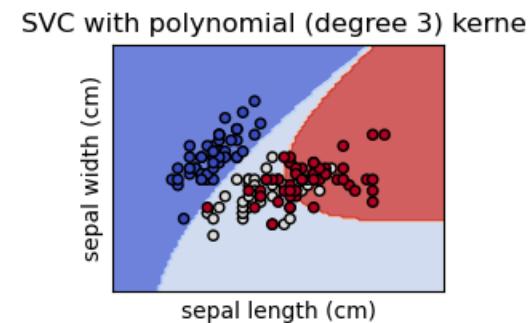
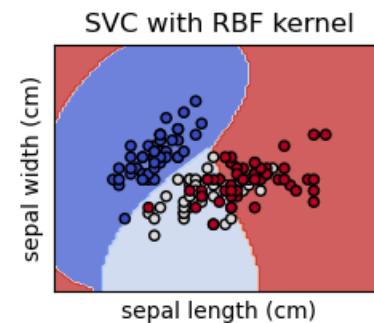
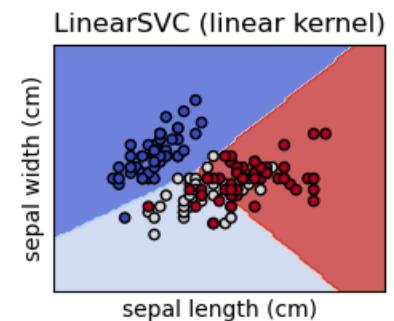
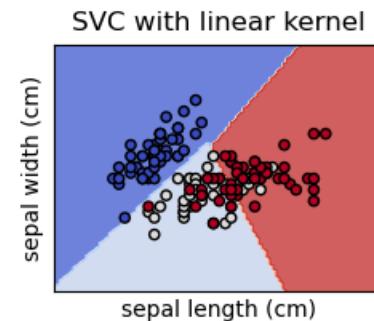
Linear Regression

- **Definition:** Linear approach to modeling the relationship between a dependent variable and one or more independent variables.
- **Application:** Predicting numerical values.



Support Vector Machines (SVM)

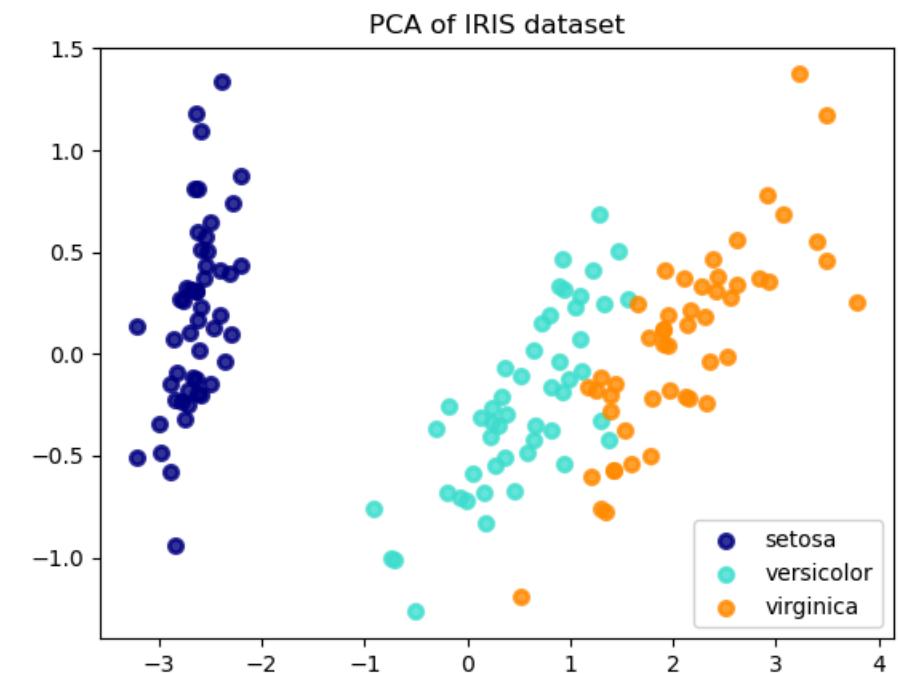
- **Definition:** Supervised algorithm that separates classes with a hyperplane.
- **Application:** Classifying data into categories.



Clustering & Dimensionality - Unsupervised Learning

Principal Component Analysis (PCA)

- **Definition:** Dimensionality reduction technique transforming data to a new coordinate system.
- **Application:** Reducing the number of features in datasets.



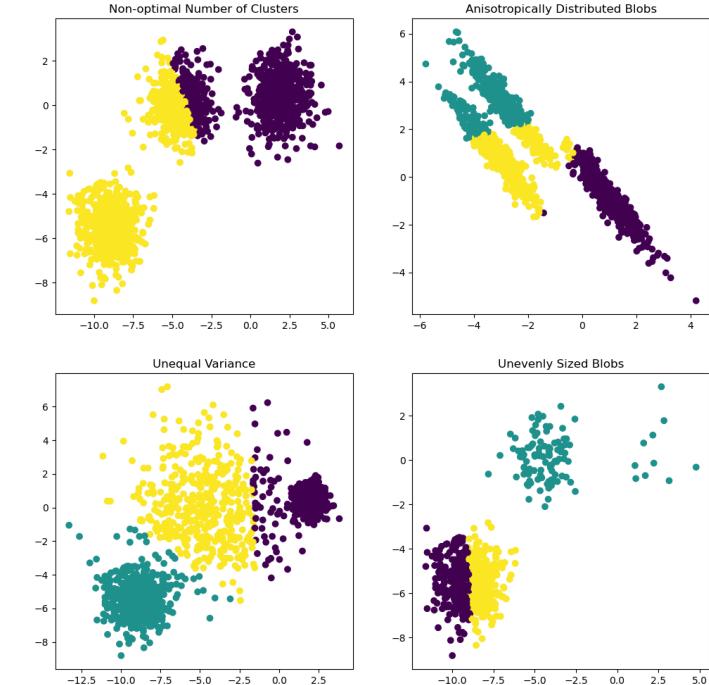
K-Means Clustering

- **Definition:** Partitions data into K clusters where each data point belongs to the cluster with the nearest mean.
- **Application:** Customer segmentation, market research.

$$\sum_{i=0}^n \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

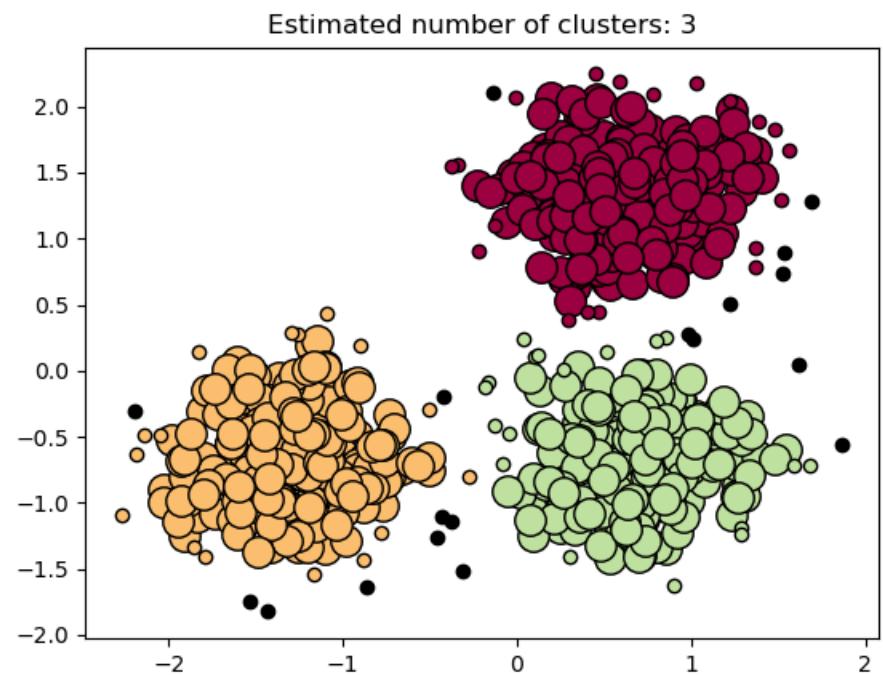
[Sci-kit learn Article](#)

Unexpected KMeans clusters



DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- **Definition:** Clustering algorithm that groups points closely packed together while marking outliers.
- **Application:** Identifying clusters of varying shapes and densities.



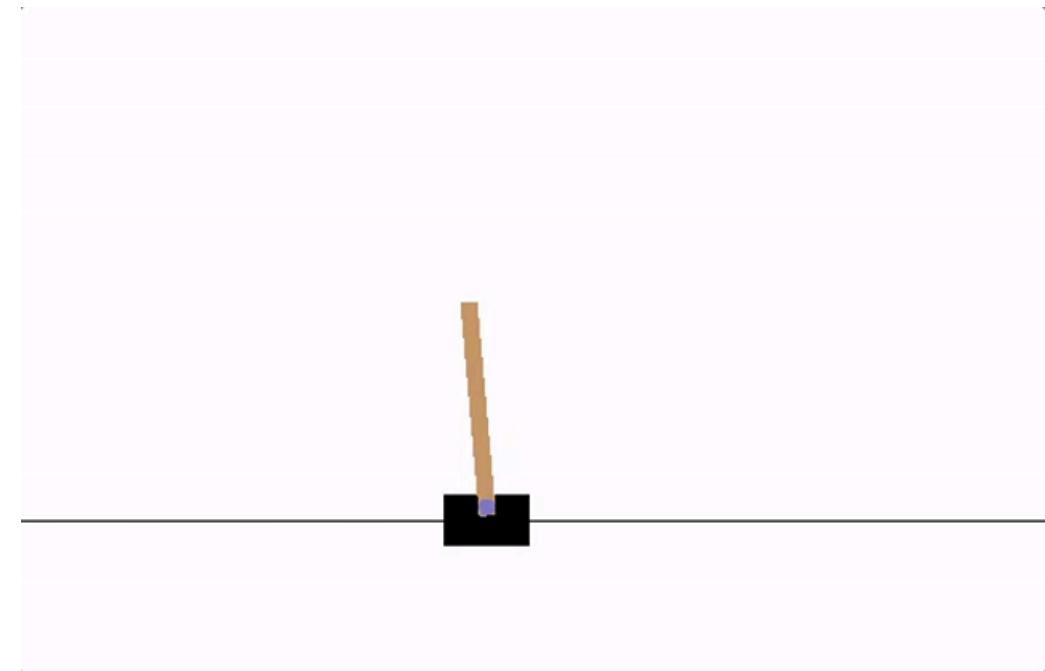
Other machine learning paradigms

Reinforcement Learning (RL)

- **Definition:** Learning paradigm where an agent learns by interacting with its environment to maximize cumulative reward.
- **Application:** Game playing, robotics, autonomous driving.

Not Supported by SciKit-Learn!!

Requires frameworks such as tensorflow, keras, or [pytorch](#) to manage more complex neural network architecture



Reinforcement learning is often thought of as a separate pillar of machine learning. Not quite supervised learning, not quite unsupervised.

It has a huge problem of alignment, however.

There is always some kind of proxy between the system/reward and the environment.

Rewards are often set arbitrarily and can be hard to do so without unforeseen consequences.

Semi-supervised learning

Overview

Semi-supervised learning is a machine learning technique that leverages both labeled and unlabeled data to build better models.

- Falls between supervised learning, which uses only labeled data, and unsupervised learning, which uses only unlabeled data.
- The driving idea behind semi-supervised learning is that the unlabeled data can provide useful information about the structure of the data distribution, thereby enhancing the learning process.

Approaches in Semi-Supervised Learning

1. Self-training:

- Initially train a model using the labeled data.
- Use this model to predict labels for the unlabeled data.
- Add the most confident predictions to the labeled dataset.
- Retrain the model using this augmented labeled dataset and repeat as needed.

2. Co-training:

- Train two models on different views (subsets of features) of the data.
- Each model uses the other's predictions on the unlabeled data to add confident examples to the labeled dataset.

3. Graph-based Methods :

- Treat the dataset as a graph, where nodes represent data points and edges represent similarities between them.
- Use techniques like label propagation to spread the label information from labeled to unlabeled points based on the graph structure.

4. Generative Models :

- Learn the distribution of the data and generate new data points. Use methods such as [Variational Autoencoders](#) (VAEs) or [Generative Adversarial Networks](#) (GANs) to enhance the learning process.

Practical Implementation Example Using Python and Scikit-learn

Let's look at a simple example of semi-supervised learning using the [LabelSpreading](#) algorithm from Scikit-learn

First import our dependencies:

```
import numpy as np
from sklearn import datasets
from sklearn.semi_supervised import LabelSpreading
import matplotlib.pyplot as plt
```

Next, we'll generate a synthetic dataset to work with

```
# Create a synthetic dataset  
X, y = datasets.make_moons(n_samples=300, noise=0.1)
```

**Let's unlabel most of the data to
simulate a large amount of
unlabeled data with some small set
of labels**

```
y[50:] = -1 # Unlabel majority of the data
```

Lets train our model

```
# Create and fit the model  
label_spread = LabelSpreading(kernel='knn', alpha=0.8)  
label_spread.fit(X, y)  
  
# Predict labels for the entire dataset  
y_pred = label_spread.transduction_
```

[User guide in Sci-kit](#)

Results??

```
# Plot the results
plt.figure(figsize=(8, 5))
cm = plt.cm.RdBu
plt.scatter(X[:, 0], X[:, 1], c=y_pred, cmap=cm, edgecolors='k')
plt.title("Semi-Supervised Learning with Label Spreading")
plt.show()
```

Yay! we re-labeled our data

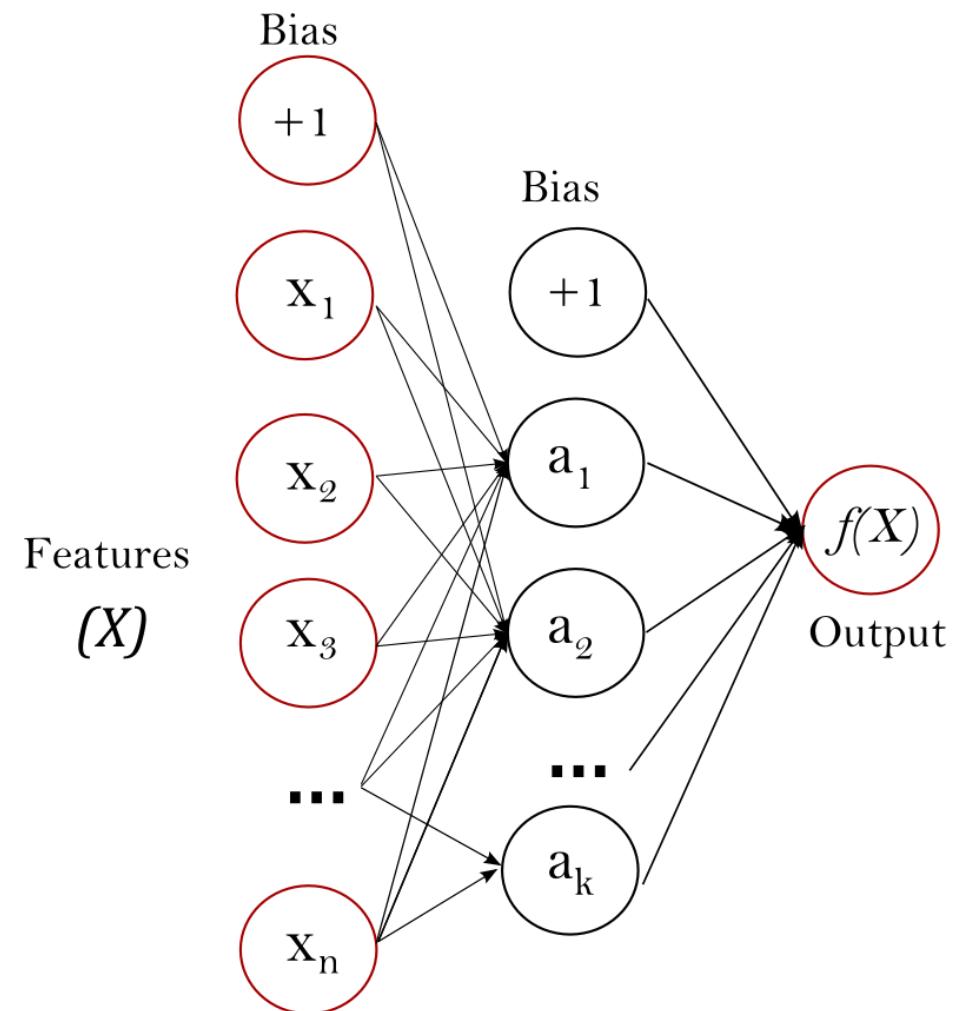
**Now we can train our model using
the whole dataset rather than the
10% that was labeled initially**

Deep Learning Architectures

Feed Forward Neural Networks (FNN/MLP)

- **Definition:** Simplest type of artificial neural network with information flowing only forward.
- **Application:** Basic pattern recognition.

[SciKit-Learn Article](#)



Recurrent Neural Networks (RNN) (pytorch etc..)

- **Definition:** Networks where connections form directed cycles, allowing temporal dynamic behavior.
- **Application:** Sequence prediction, time series analysis.

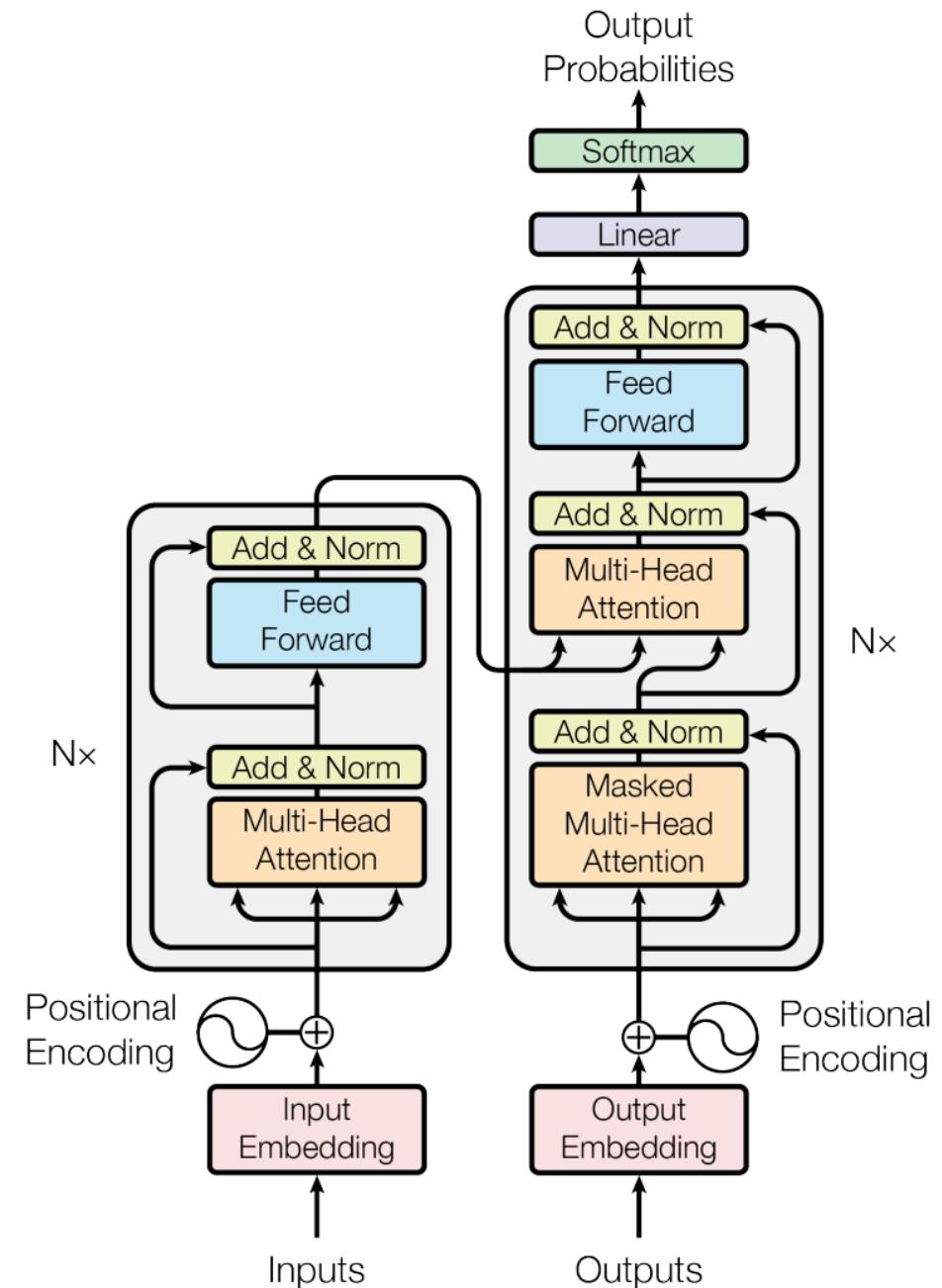
Great Article/Tutorial to read and try!

[pytorch documentation RNN](#)

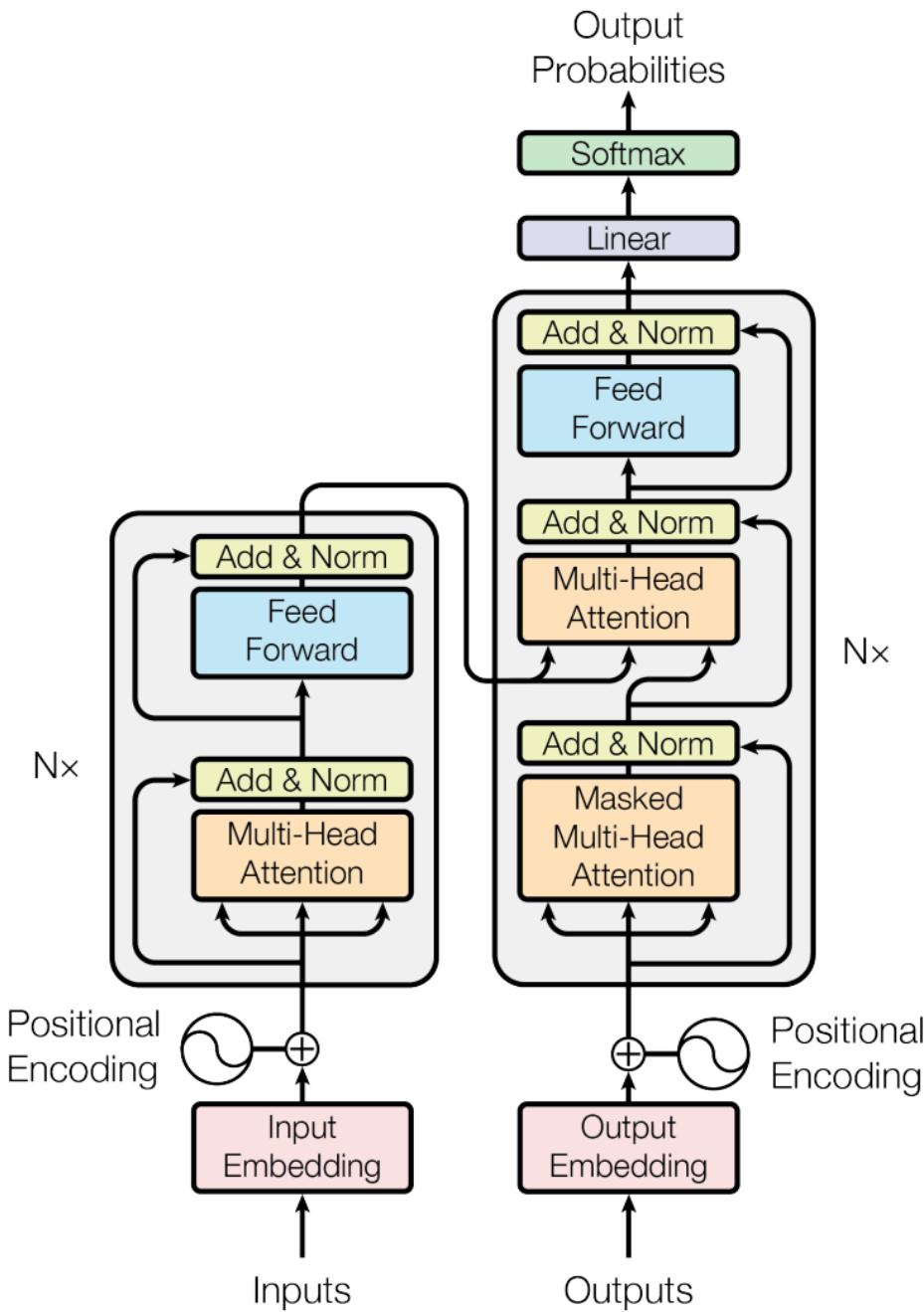
🤖 Transformers 😊

- **Definition:** Attention mechanism allowing the models to focus on different parts of the input sequence.
- **Application:** Natural Language Processing (NLP), such as BERT, GPT. (And much more...)

Transformer = Attention + FNN/MLP



BERT Encoder



GPT Decoder

Output
Probabilities

Softmax

Linear

Add & Norm

Feed
Forward

Add & Norm

Multi-Head
Attention

Add & Norm

Masked
Multi-Head
Attention

$N \times$

$N \times$

Positional
Encoding

Positional
Encoding

Input
Embedding

Output
Embedding

Inputs

Outputs

Limitations of Sci-Kit Learn

- **Limited Deep Learning Support:**
 - Primarily focused on traditional ML algorithms.
 - For deep learning tasks, TensorFlow, PyTorch, or Keras recommended.
- **Scalability:**
 - Not optimized for very high-dimensional data or extremely large datasets.
- **Real-Time Model Serving:**
 - Not designed for real-time application deployment.
- **Feature Set:**
 - May lack certain cutting-edge algorithms found in specialized deep learning frameworks.

Summary

- Introduction to the sci-kit learn package.
- Differences between supervised and unsupervised learning.
- Basics of deep learning and its architectures.
- Overview of key ML techniques (regression, classification, clustering).
- Understand reinforcement learning and its applications.
- Limitations of the sci-kit learn package.

Key Takeaways

- **Sci-Kit Learn:** Robust library for traditional ML tasks.
- **Learning Types:** Importance of supervised and unsupervised learning.
- **Deep Learning:** Know the basics of different DL architectures.
- **Algorithms:** Understanding various techniques available in sci-kit learn.
- **Limitations:** Knowing when to use other frameworks for deep learning tasks.

In-Class Activity/Workshop: Applying Sci-Kit Learn

Activity Overview

- **Objective:** Apply various machine learning techniques using the sci-kit learn package to a sample dataset.
- **Lab Sheet:** [Markdown Lab Sheet](#)

Discussion:

- Discuss the results with peers.
- Reflect on the performance of different techniques.
- Identify potential biases or issues in the dataset and model.

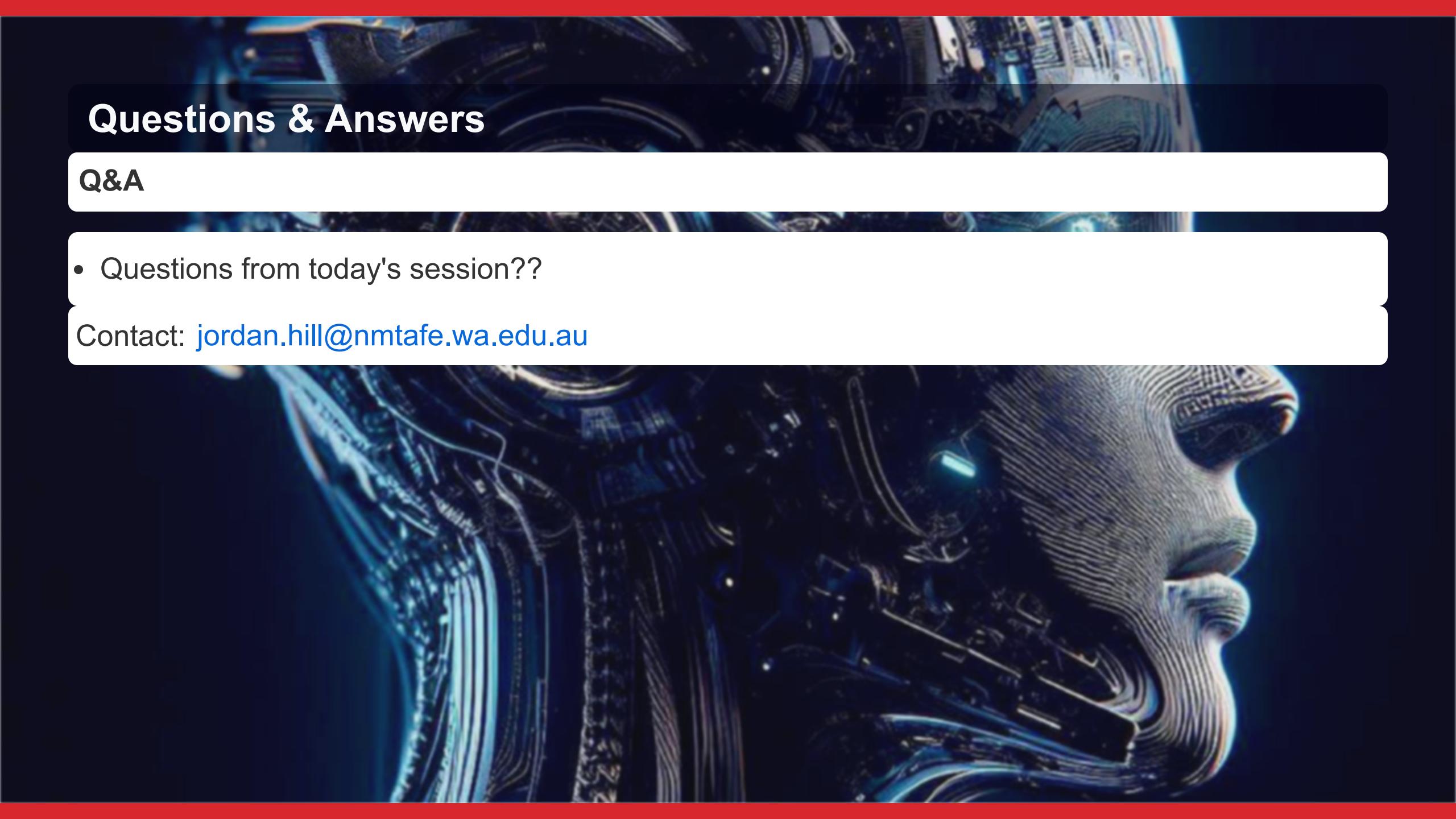


Questions & Answers

Q&A

- Questions from today's session??

Contact: jordan.hill@nmtafe.wa.edu.au



Next Week

Data Bias and Ethics in AI

Next Week's topics:

- Implementation Risks
- Ethics in AI, including Australia's AI Ethics Framework
- Alignment



Home Work!

Before class next week read:

[Australia's AI Ethics Principles](#)

"I Have No Mouth, and I Must Scream" by Harlan Ellison

"Robbie" by Isaac Asimov

We will be discussing these texts next week in-class

Choose one of the readings from the last 3 weeks:

Come to class with a question for the class about one of the assigned readings.

You will be asked to provide your question to the class tomorrow for discussion.

Think: How well does the AI Ethics Principles address issues raised in our readings?