CSCE 580: Introduction to AI
CSCE 581: Trusted AI

Lecture 19: Machine Learning – NN, DL

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

31ST OCT 2023

Carolinian Creed: “I will practice personal and academic integrity.”

Credits: **Copyrights of all material reused acknowledged**

Organization of Lecture 19

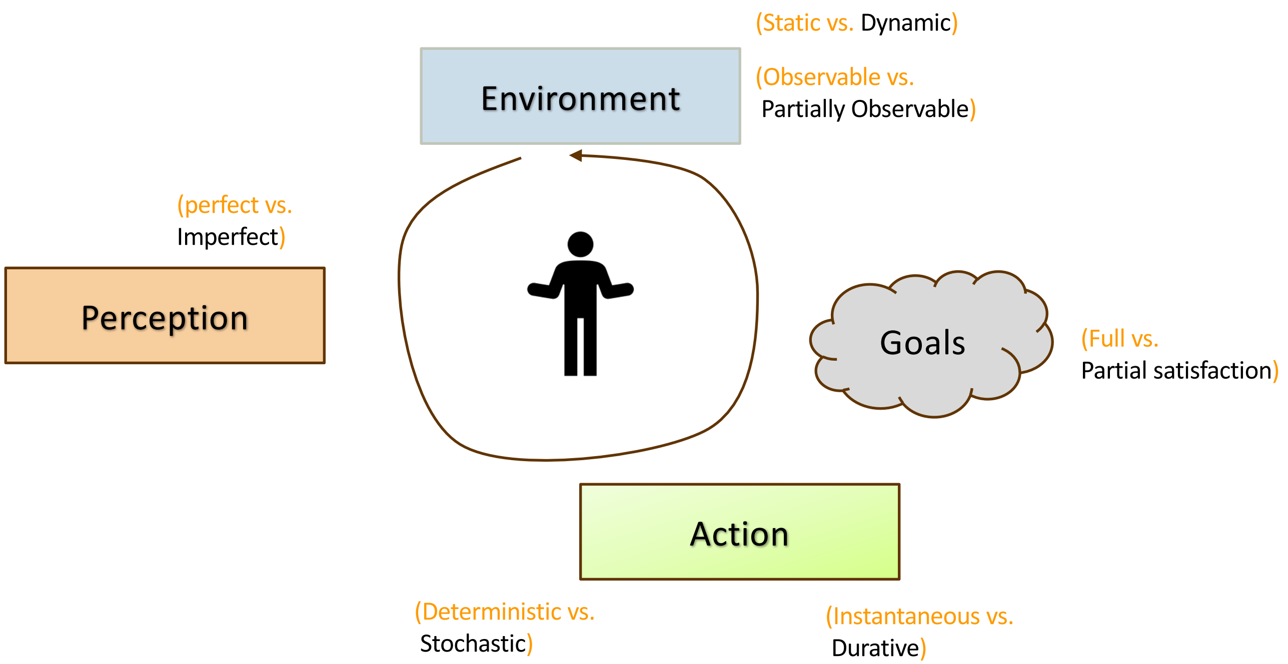
- Introduction Segment
  - Recap of Lecture 16
- Main Segment
  - Neural Networks
  - Deep Learning
  - Trust Issues
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 20
  - Ask me anything

Introduction Section

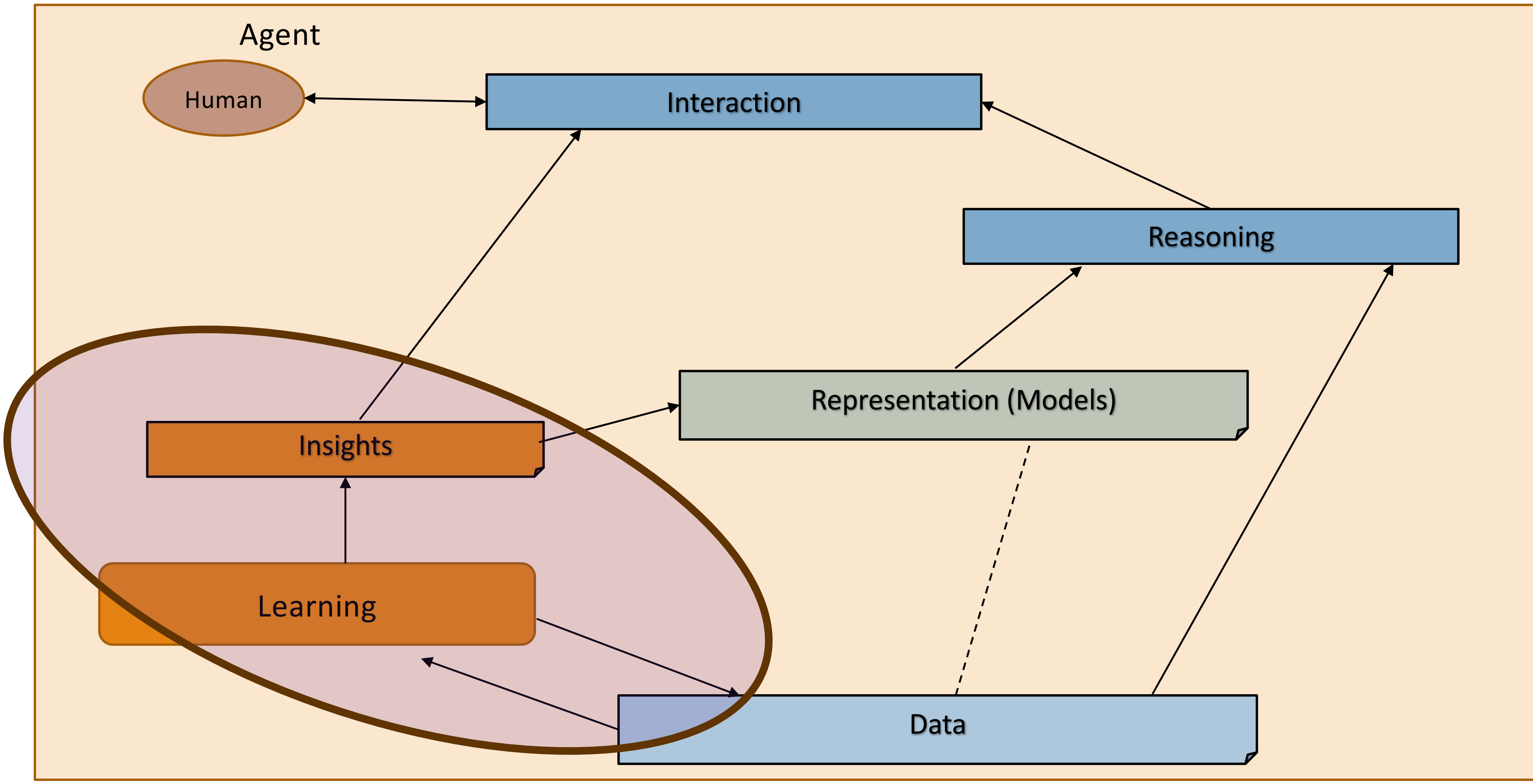
Recap of Lecture 18

- Topic discussed
  - Trust/ Explanations, LIME - Recap
  - Unsupervised ML Algorithms

Intelligent Agent Model



Relationship Between Main AI Topics



Where We Are in the Course

CSCE 580/ 581 – In This Course

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week 8: Machine Learning - Classification
- Week 9: Machine Learning - Classification – Trust Issues and Mitigation Methods
- Topic 10: Learning neural network, deep learning, Adversarial attacks
- Week 11: Large Language Models – Representation, Issues
- Topic 12: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 13: Planning, Reinforcement Learning – Sequential decision making
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Main Section

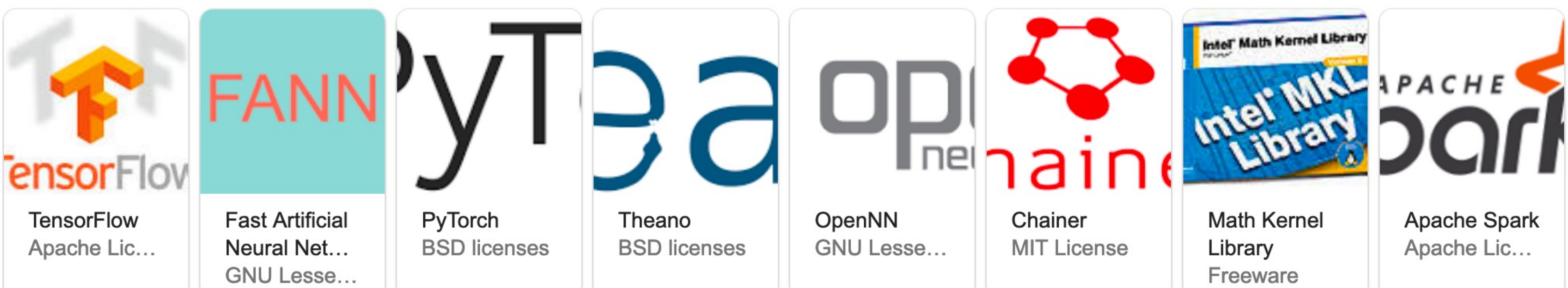
Credit: Retrieved from internet

Machine Learning – Insights from Data

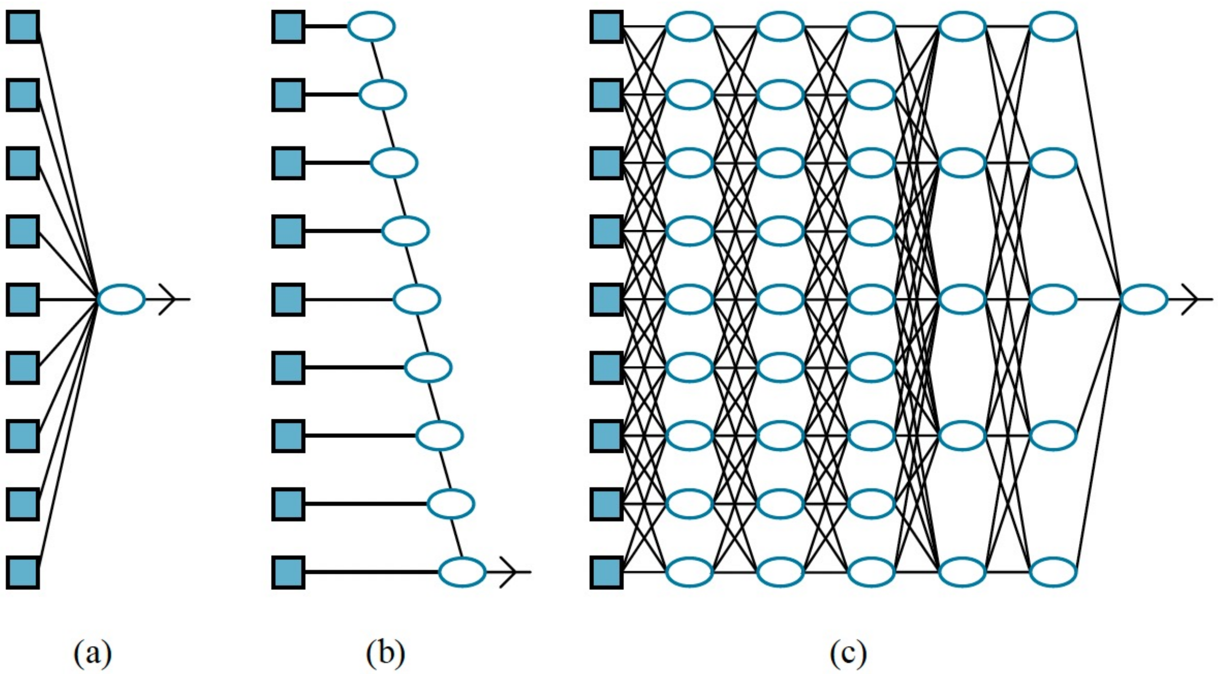
- Descriptive analysis
  - Describe a past phenomenon
  - **Methods**: classification (feedback from label), clustering, dimensionality reduction, anomaly detection, neural methods, reinforcement learning (feedback from hint/ reward)
- Predictive analysis
  - Predict about a new situation
  - **Methods**: time-series, neural networks
- Prescriptive analysis
  - What an agent should do
  - **Methods**: simulation, reinforcement learning, reasoning

- New areas
  - Counterfactual analysis
  - Causal Inferencing
  - Scenario planning
  - Representation learning

Neural Network Methods



Model Depth and Learning Ability

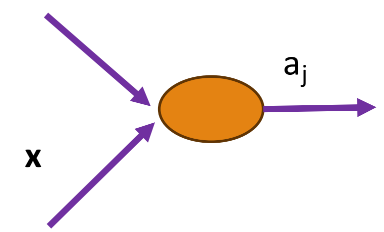


(a) A shallow model, such as linear regression, has short computation paths between inputs and output. (b) A decision list network has some long paths for some possible input values, but most paths are short. (c) A deep learning network has longer computation paths, allowing each variable to interact with all the others.

Adapted from:
Russell & Norvig, AI: A Modern Approach

Node (Unit) of a NN

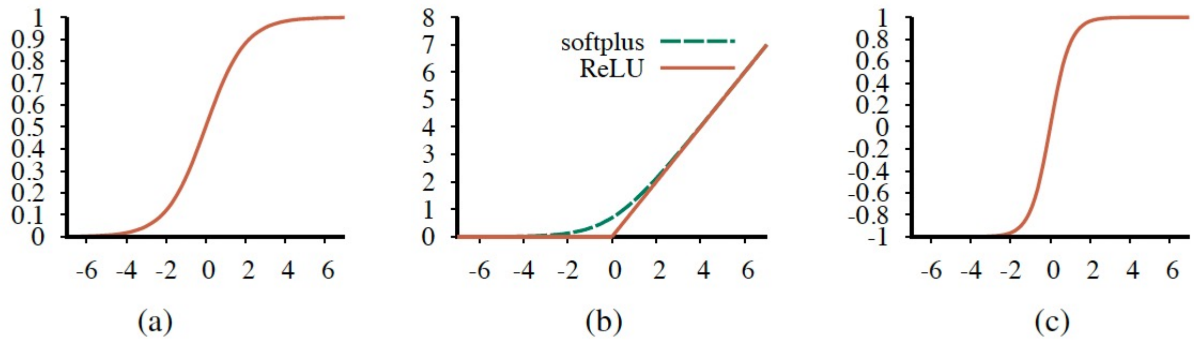
- Notations and meanings
  - a_j: output of a unit j
  - $w_{i,j}$: weight of link from unit i to unit j
  - $a_j = g_j\ (\Sigma\, w_{i,j}\, a_i\ )$, where g_j is a nonlinear activation function
- $a_j = g_j\ (\mathbf{w}^T \mathbf{x})$, where **w** is vector of weights leading into unit j and **x** is the inputs to unit j



Popular Activation Functions

- Logistics or sigmoid function: σ(x) = $1/(1 + e^{-x})$
- ReLU (rectified linear unit): max (0, x)
- Softplus function: $\log(1 + e^{x})$
  - Smooth version of ReLU
- $\tanh(x) = (e^{2x} - 1) / (e^{2x} + 1)$
  - Scaled and shifter version of sigmoid; tanh (x) = $2\sigma(2x) - 1$

a) the logistic or sigmoid function
b) the ReLU function and the softplus function
c) the tanh function.



Adapted from:
Russell & Norvig, AI: A Modern Approach

Note: All activation functions are non-linear

Loss functions

- Mean squared error

$$MSE = \frac{1}{n} \sum_{j=1}^{n} \left[f(X_{j\cdot}) - y_j \right]^2$$

- Categorical Cross Entropy

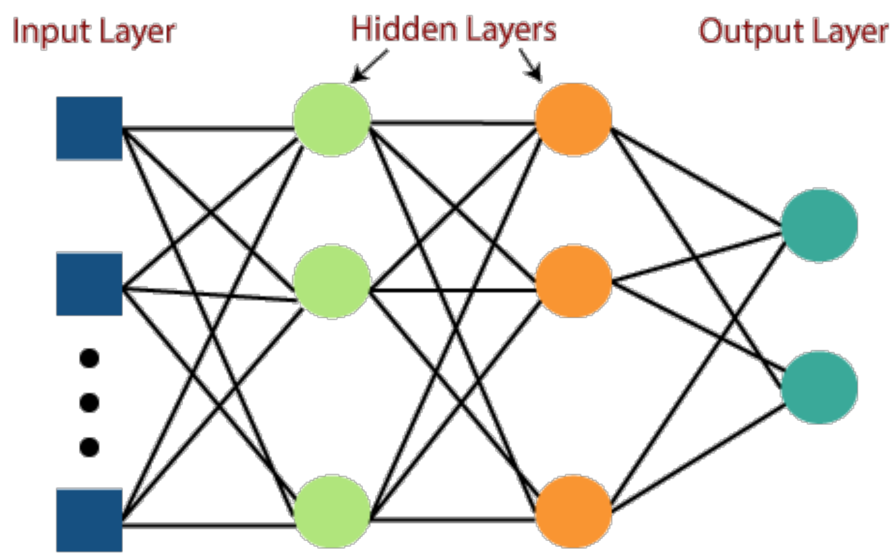
$$\text{Cost} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} [y_{ij} \log(\hat{y}_{ij})]$$

k is classes,
y = actual value
Ŷ = prediction

- More loss functions:

https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/

NN – Multi Layer Perceptron



Content and Image Courtesy:
https://github.com/Thanasis1101/MLP-from-scratch

(Stochastic) Gradient Descent

Gradient Descent

w ← any point in the parameter space

While not converged do:

For each w_i in **w** do:

$w_i \leftarrow w_i - \alpha \ (\partial \ / \ \partial w_i) \ Loss(\mathbf{w})$

Calculate the gradient of the loss function with respect to the weights along the gradient direction to reduce the loss.

Stochastic Gradient Descent (SGD)

Randomly select a small number of training examples at each step

Sources:

- https://en.wikipedia.org/wiki/Stochastic_gradient_descent
- Russell & Norvig, AI: A Modern Approach, Chapter 19

Logistic Regression in a Slide

Function estimate (linear)
W: weight, b: bias

$$f(X_j) = X_j W + b$$

Update Weight

$$W^* = W - \eta \frac{dL}{dW}$$

Error Term (mean squared error)

$$MSE = \frac{1}{n} \sum_{j=1}^{n} \left[f(X_{j\cdot}) - y_j \right]^2$$

Common Code Pattern

```
y = tf.matmul(x, W) + b
loss = tf.reduce_mean(tf.square(y - y_label))
```

NN Concepts

- **Epoch**: The number of times the learning algorithm will iterate over the entire dataset
- **Batch**: how many samples are processed before updating the model's internal parameters.
  - **Batch Gradient Descent**: Batch Size = Size of Training Set
  - **Stochastic Gradient Descent**: Batch Size = 1
  - **Mini-Batch Gradient Descent**: 1 < Batch Size < Size of Training Set

Credit: https://rentry.org/llm-training

Universal Approximation Theorem

- A network with just two layers of computation units, first nonlinear, and the second linear, can approximate any continuous function to an arbitrary degree of accuracy.
- **Why**: a sufficiently large network can implement a lookup table for continuous functions
  - Nonlinear layer is the key

Sources:

- https://en.wikipedia.org/wiki/Universal_approximation_theorem
- Russell & Norvig, AI: A Modern Approach, Chapter 21

Datasets

- In keras, https://keras.io/api/datasets/
  - boston_housing
  - cifar10 module, cifar100, fashion_mnist, mnist
  - imdb module
  - reuters module

- In TF, https://www.tensorflow.org/datasets/catalog/overview#all_datasets

Keras Walkthrough

- Package: https://keras.io/about/
- Example model:
  - Sequential: https://keras.io/guides/sequential_model/
- Many examples: classification, image, text, audio
  - https://keras.io/examples/
- Future Keras: https://keras.io/keras_core/
  - Keras Core - run Keras workflows on top of TensorFlow, JAX, and PyTorch; preview of Keras 3.0

Code Examples With Keras and TF

1. Classification – diabetes

2. Try code
- Play with hyper-parameters
- Look at keras features used

Code location:
https://github.com/biplav-s/course-ai-tai-f23/tree/main/sample-code/Class19-To-21-DL

Code Examples With Keras and TF

1. Classification – diabetes

2. Prediction/ representation learning – autoencoder

Code location:
https://github.com/biplav-s/course-ai-tai-f23/tree/main/sample-code/Class19-To-21-DL

Code Examples With Keras and TF

1. Classification – diabetes
2. Prediction/ representation learning – autoencoder
3. **Classification – MNIST**

Code location:
https://github.com/biplav-s/course-ai-tai-f23/tree/main/sample-code/Class19-To-21-DL

Keras and TensorFlow

- By Example:
  - https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l11-nn-dl/Basic%20TensorFlow%20and%20Keras.ipynb
  - TensorFlow's NMIST tutorial
  - https://www.tensorflow.org/tutorials/quickstart/beginner

- More examples
  - Number Addition by sequence learning: https://keras.io/examples/nlp/addition_rnn/
  - AutoEncoder: https://machinelearningmastery.com/lstm-autoencoders/

NN/ MLP

- Code examples:
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l6-l7-l8-supervised-ml/Supervised-NaiveBayes-GradientBoost-NN-Classification.ipynb
- Scikit Library:
  - MLP: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Which NN/DL Tool to Use

- See:
  - https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article
  - In theory, keras supports all major ones
    - Pytorch used in academic research more
    - TF used in production systems

Resources and Books

- Understanding Deep Learning, https://udlbook.github.io/udlbook/
- Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, https://www.deeplearningbook.org/
- AI – A Modern Approach, Russell & Norvig, https://aima.cs.berkeley.edu/
- Websites of libraries – Keras.

Trust Issues with NN

- Robustness: can the model give the results in the presence of (input) perturbation? Noise?
- Computation/ footprint: why does the learning take so much compute resources?
- Data: is the data representative? How was the data obtained?
- Explainability: why does the model work?
- Fairness: Is the output fair to user groups?

Course Project

Project Discussion: What Problem Fascinates You ?

- Data
  - Water
  - Finance
  - …
- Analytics
  - Search, Optimization, Learning, Planning, …
- Application
  - Building chatbot
- Users
  - Diverse demographics
  - Diverse abilities
  - Multiple human languages

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

Project Discussion: Dates and Deliverables

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
- Nov 30, 2023
  - Project report due
- Dec 5 / 7, 2023
  - In-class presentation

Skeleton: A Basic Chatbot

- Run in an infinite loop until the user wants to quit
- Handle any user response
  - User can quit by typing “Quit” or “quit” or just “q”
  - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – “I do not know this information”.
- Handle known user query types // Depends on your project
  - “Tell me about N-queens”, “What is N ?”
  - “Solve for N=4?”
  - “Why is this a solution? ”
- Handle chitchat // Support at least 5, extensible from a file
  - “Hi” => **“Hello”**
  - …
- ***Store session details in a file***

Illustrative Project

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Discussion: Illustration

1. Create a private Github repository called “CSCE58x-Fall2023-<studentname>-Repo”. Share with Instructor (biplav-s) and TA (kausik-l)
2. Create Google folder called “CSCE58x-Fall2023-<studentname>-SharedInfo”. Share with Instructor (prof.biplav@gmail.com) and TA (lakkarajukausik90@gmail.com)
3. Create a Google doc in your Google repo called “Project Plan” and have the following by next class (Sep 5, 2023)

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Illustration: N-Queens

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
    - Method 1: Random solution
    - Method 2: Search – BFS
    - Method 3: Search - …
  - Human interaction: Develop a basic chatbot (no AI) as outlined
  - Deliverable
    - Code structure in Github
      - ./data
      - ./code
      - ./docs
      - ./test
    - Presentation: Make sprint presentation on Oct 12, 2023

Reference: Project Rubric - OLD

- **Project results** – 60%
  - Working system ? – 30%
  - Evaluation with results superior to baseline? – 20%
  - Considered related work? – 10%
- **Project efforts** – 40%
  - Project report – 20%
  - Project presentation (updates, final) – 20%

- **Bonus**
  - Challenge level of problem – 10%
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

Reference: Project Rubric - NEW

- **Project report** – 60%
  - Project description: problem, related work, approach, evaluation – 40%
  - Working system demo/ video – 10%
    - Well organized Github with code (./data, ./code, ./docs, ./test) – 10%
- **Project presentation** – 40%
  - Evaluation by peers, instructor and TA
- **Bonus**
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

Evaluation of Presentation

1. An online form will be available during presentation
2. During a presentation, three students will be assigned to review along with instructor and TA
3. They will enter following survey questions:
    1. Their name
    2. Presentation number
    3. How useful is the system – will you use it? [1-5 scale]
    4. How well have you understood the project from the presentation? [1-5 scale]
4. Top and bottom scores will be removed. Average of remaining three will be used for final presentation marks

Lecture 19: Summary

- We talked about
  - Neural Networks
  - Deep Learning
  - Trust Issues

Concluding Section

About Next Lecture – Lecture 20

Lecture 20: Deep Learning, Large Language Models

- Deep Learning
- Language Models (LM)
- Text processing and LLM