Final Project and Neural Networks COSC 425

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Announcements

- Lab 4 due on October 27
- No class on Thursday, October 26
  - Use that time to meet with your group/discuss your final project
- Final Projects:
  - Finalization of dataset to be used by **November 3.**
    - 5-point penalty if the dataset is not selected by that date.

Questions about Lab 4?

Final Project

Final Project

- Requirements for each of the final projects include:
  - Final project report: 6 to 8-page 2 column IEEE conference format report
  - In-class presentation: 3 minute in-class presentation overview of the project
  - Video presentation: 8–10-minute video presentation of the project
- Final projects will include:
  - The selection of a dataset
  - The analysis and visualization of this dataset
  - The application of machine learning approaches to the dataset
  - A discussion of hyperparameters required for this dataset

Final Project Deadlines

- Team finalized by **October 19**.
- Finalization of dataset to be used by **November 3**.
  - 5-point penalty if the dataset is not selected by that date.
- Presentations will take place in class on **November 30 and December 5**.
- Final reports and video presentations will be submitted via Canvas and are due on **December 8**.

Final Project

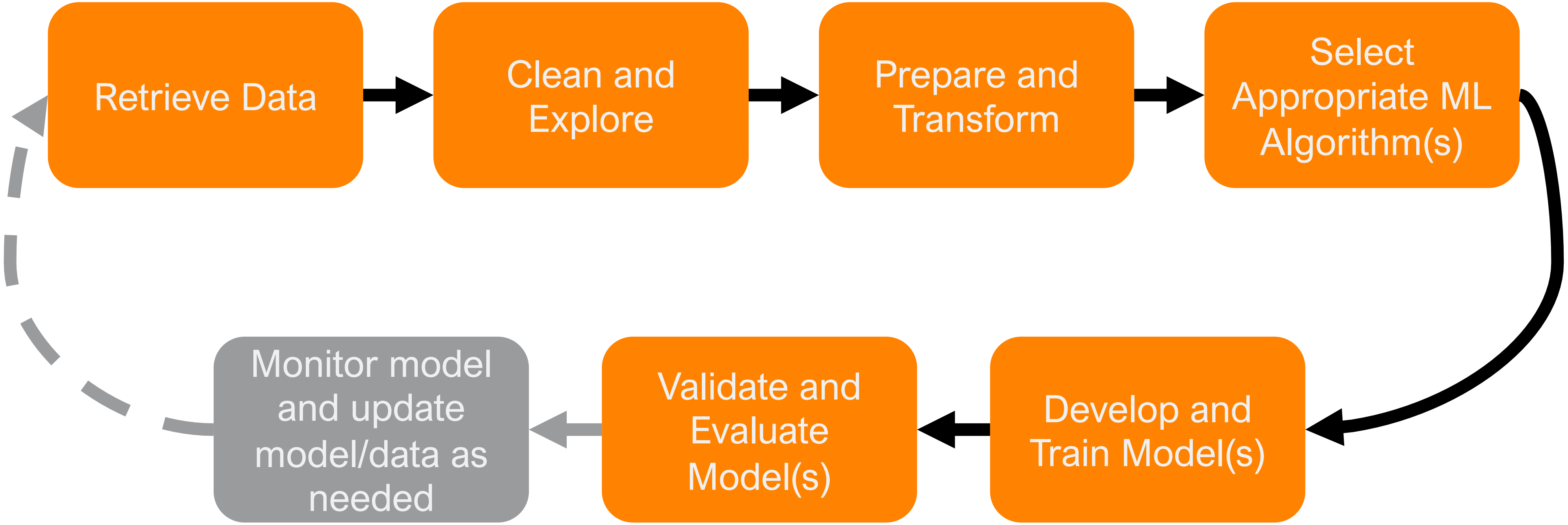


Figure adapted from: https://towardsdatascience.com/the-machine-learning-workflow-explained-557abf882079

Final Project

- You will be doing some data analysis on the dataset before you apply machine learning to it
  - Understanding and reporting the broader characteristics of the data
  - Cleaning the data
  - Visualizing the data (as best you can)
  - Preparing the data for machine learning approaches
    - Down-selecting
    - Scaling/transforming

Datasets

- Sources:
  - US government data: https://data.gov/
  - World bank data: https://data.worldbank.org/
  - World Health Organization data: https://www.who.int/data/gho/
  - Miscellaneous data:
    - Kaggle: https://www.kaggle.com/
    - UCI ML Repository: https://archive.ics.uci.edu/ml/index.php
    - Azure open datasets: https://docs.microsoft.com/en-us/azure/open-datasets/dataset-catalog
- Finalization of dataset(s) to be used by **November 3**.
  - 5-point penalty if the dataset is not selected by that date.

Machine Learning Approaches

- You will use machine learning approaches that we have covered in class.
  - **NOTE:** Your ML approach ***cannot*** be deep learning. We're covering that way too late and there is an entire class devoted to deep learning in this department.
- You can choose to use one or more machine learning approaches
- You will do an examination of the impact of hyperparameters on the machine learning approach(es)

Project Report

Project Report

- Your final project report should be structured as an academic paper for a conference or journal submission is structured.
  - Your final project report should use the IEEE 2-column conference format with the sigconf format.
  - There are both LaTeX and Word versions of this format
  - I recommend using Overleaf for LaTeX
- The final project report should be at least 6 pages and no more than 8 pages long.
- The report length includes any references, plots, and figures that you include.

Project Report Components

- Abstract
- Introduction and Motivation
- Dataset
- ML Approaches and Methodology
- Results
- Discussion, Conclusion, and Future Work
- Contributions of Team Members

Abstract

- 150-250 words to briefly introduce your dataset, the key question/problem you’re addressing with ML applied to the dataset, and a brief summary of your results.
- Someone should be able to read just the abstract of your report and have an idea of what you did and what the results were.

Introduction and Motivation

- You should address the following questions in the introduction and motivation section of your report
  - What is the dataset or datasets you chose for your project and why did you choose it/them?
  - What is the overarching goal of you're trying to achieve with ML on this dataset?
  - (Briefly) Which ML approaches are you using and why did you choose those?
- You should include a discussion here of what the goals are of the project and how you're determining whether you are successfully addressing those goals.
- Your introduction should include a summary of the rest of the paper as well.

Dataset

- In this section, you should describe in detail the dataset that you're using
- Use visualizations to illustrate characteristics of the dataset
- Describe what features you're using and why
- If you omitted any features that are usually part of this dataset, describe why you omitted them
- Describe any transformations or pre-processing you performed on the data (and why)
- You must include at least two plots describing aspects of the data in this section

ML Approaches and Methodology

- In this section, you should describe the ML approaches you chose to apply to this dataset and why you chose those ML approaches
- You should describe what hyperparameters you will be investigating of those approaches
- You describe which metrics you will be reporting and how those metrics are calculated
- You should describe in detail the experimental setup you have defined.
  - Are you comparing two different algorithms?
  - Are you comparing performance across hyperparameter values?
  - How are you defining whether the project is successful?

Results

- In this section, you will describe the results of your approach.
- You should depict the results visually through plots, and you should provide a discussion of each plot and the results you obtained.
- Did something unexpected happen in the experiments?
  - It may be worthwhile to probe into that further to try to explain why it happened.
- Where appropriate, you should visualize the machine learning approach (through decision boundaries or some other visualization technique)
- If you can interpret the machine learning approach results, you should include a discussion of what you learned about the dataset from this interpretation (which features are most important, etc.)

Discussion, Conclusion, Future Work

- You will provide a discussion of the results and any major conclusions you obtained
- You will also provide at least a paragraph of “future work” discussion.
  - If you had more time, what would you do next?
  - Did this project open up any new research questions?
  - Was there something else you would have liked to have done and didn’t get a chance to do?

Contributions of Team Members

- This is a short section that is NOT included in the page count
- Here, you should describe what each team member did in the project
- There should be clear contributions of each member of the team
- Note: If you're working alone, you can omit this section

In-Class Presentation

3-5 minutes – 4 slides

In-Class Presentation

- You will have four TOTAL slides in your in-class presentation
  - Title Slide: Includes the title of the project and the names of the project team members (you should spend very little time showing this slide)
  - Dataset slide: Include a description and visualizations of the dataset
  - Machine learning methods slide: Include a description of the machine learning approach(es) you are applying and which hyperparameters you're examining
  - Results: Include an overview of the key results you obtained
- Note: You should keep text at a minimum and include visuals instead
- You will have a strict time limit to deliver the presentation
- All team members will stand at the front, but it is fine for one person to take the lead on presenting

In-Class Presentation

- Most presentations will take place in class on December 5
- We will need at least 12 presentations to take place on **November 30**.
  - If you are willing to present on this day, please let me know via email
  - You will receive an automatic 5 points of extra credit on your final project for presenting early
- Slides for short presentations should be sent to Dr. Schuman by **November 29/December 4**

Recorded Presentation

8-10 minutes

Recorded Presentation

- You will create a video presentation of 8-10 minutes
- This should be a narration over a slide show
  - If you want to do it differently in some way, I'm open to that, but you need to get it approved first
- Every team member should present at least one part of the presentation
- You should submit this presentation as an mp4 or mov video file, and you will also submit the associated slides you presented as a PDF.
- I recommend recording this via Zoom.
  - Setup a Zoom session with the full group, have one person in charge of advancing slides and do a Zoom recording

Rubric

- Report (70 points):
  – Abstract: 5 points
  – Introduction and Motivation: 10 points
  – Dataset: 10 points
  – ML Approaches and Methodology: 10 points
  – Results: 25 points
  – Discussion, Conclusions, and Future Work: 10 points
- In-Class Presentation (10 points)
  - Content (5 points)
  - Presentation quality (5 points)
- Recorded Presentation (20 points)
  - Slide content (10 points)
  - Presentation quality (10 points)

Final Project Deadlines

- Team finalized by **October 19**.
- Finalization of dataset to be used by **November 3**.
  - 5-point penalty if the dataset is not selected by that date.
- Presentations will take place in class on **November 30 and December 5**.
- Final reports and video presentations will be submitted via Canvas and are due on **December 8**.

What are you submitting?

- A zip file that includes:
  - PDF of your report
  - PDF of the slides presented in class
  - PDF of slides from pre-recorded presentation
  - Video file of pre-recorded presentation
  - Dataset (if the dataset is not publicly available)
  - Code files generated as a result of this project
- Each member of the team should submit the same zip file.

Late Penalty

- There will be 10 points off per day late
  - Submitting at 12:00 AM on December 9 is considered a day late!
- Your final project can be submitted no later than December 12
- If your project is missing any of the required components, you will also receive a penalty if you submit those late

Example Final Projects from 2022

- Compiler classification: GCC vs. Clang
- Spotify genre classification
- Stroke dataset classification
- Home price prediction
- Stock market prediction
- GTA Horse Betting

Recommended Steps for Getting Started

1. Identify a domain you want to work in
2. Identify what types of problems you might solve in that domain (classification, regression, clustering)
3. Formulate the problem
   – Identify what the inputs/features will be
   – Identify what the outputs will be

Final Project

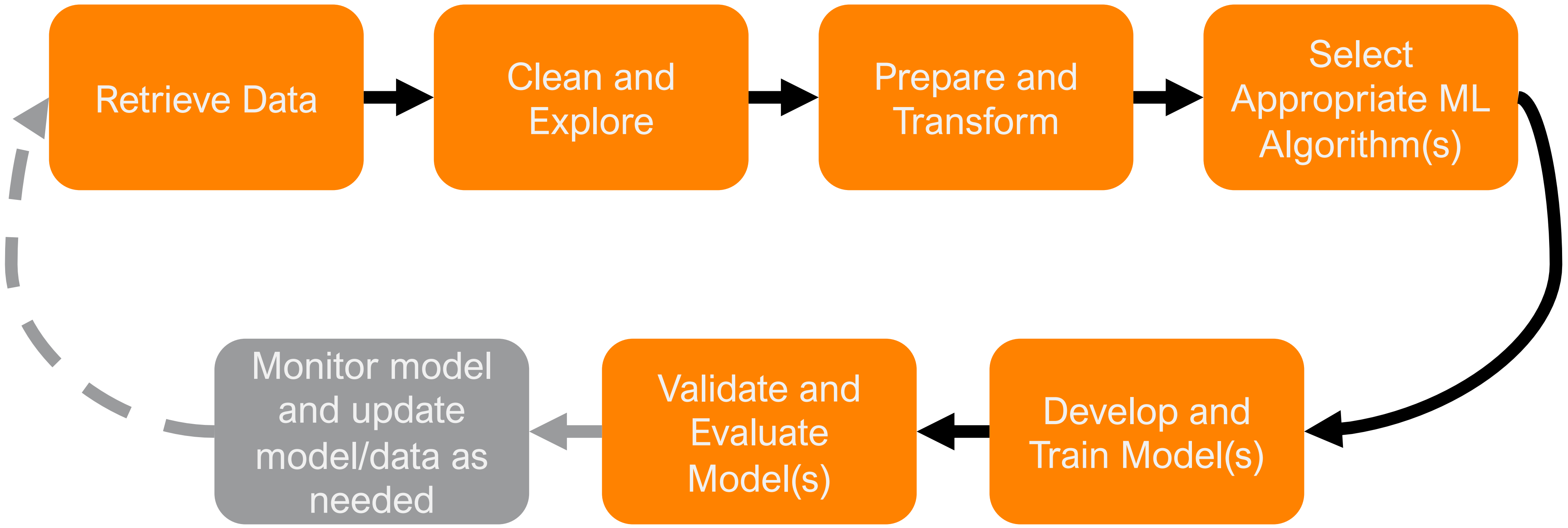


Figure adapted from: https://towardsdatascience.com/the-machine-learning-workflow-explained-557abf882079

Pop Quiz

cs425

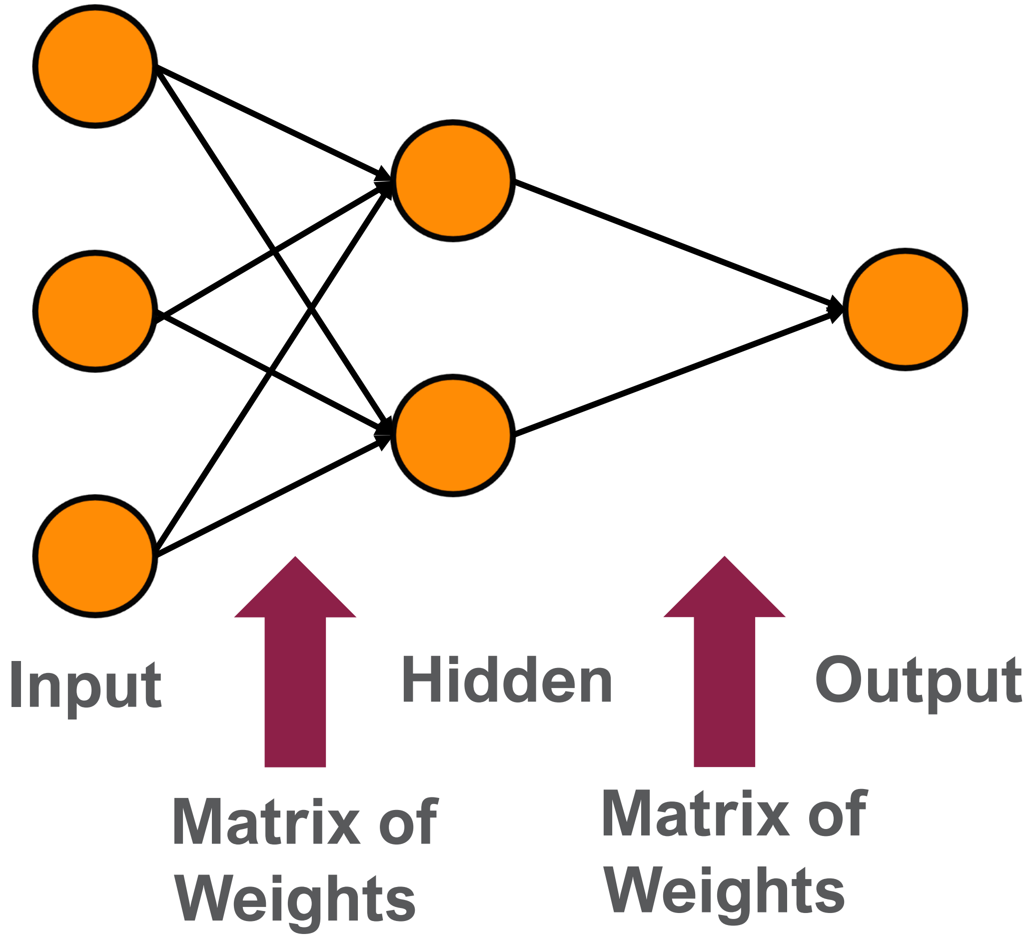
Question

- My team and I have a good idea of what dataset/problem we're planning to work on
  - A) Yes
  - B) Sort of
  - C) No

Multi-Layer Perceptrons

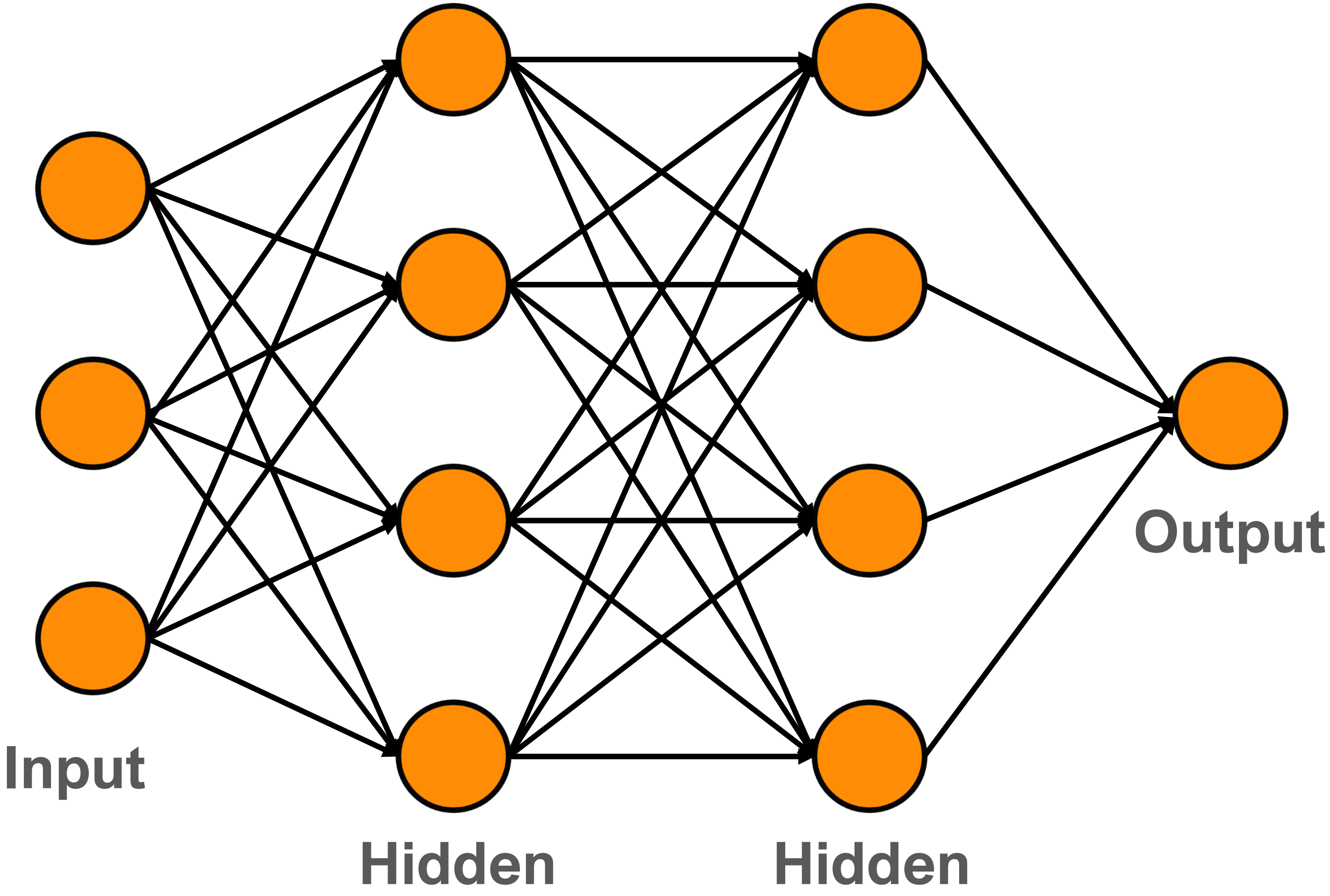
Review on Neural Networks

- We learned about multi-layer perceptrons and showed they can deal with data that is not linearly separable
- We learned that a network a single hidden layer is a universal function approximator
- We showed how to train them with back-propagation



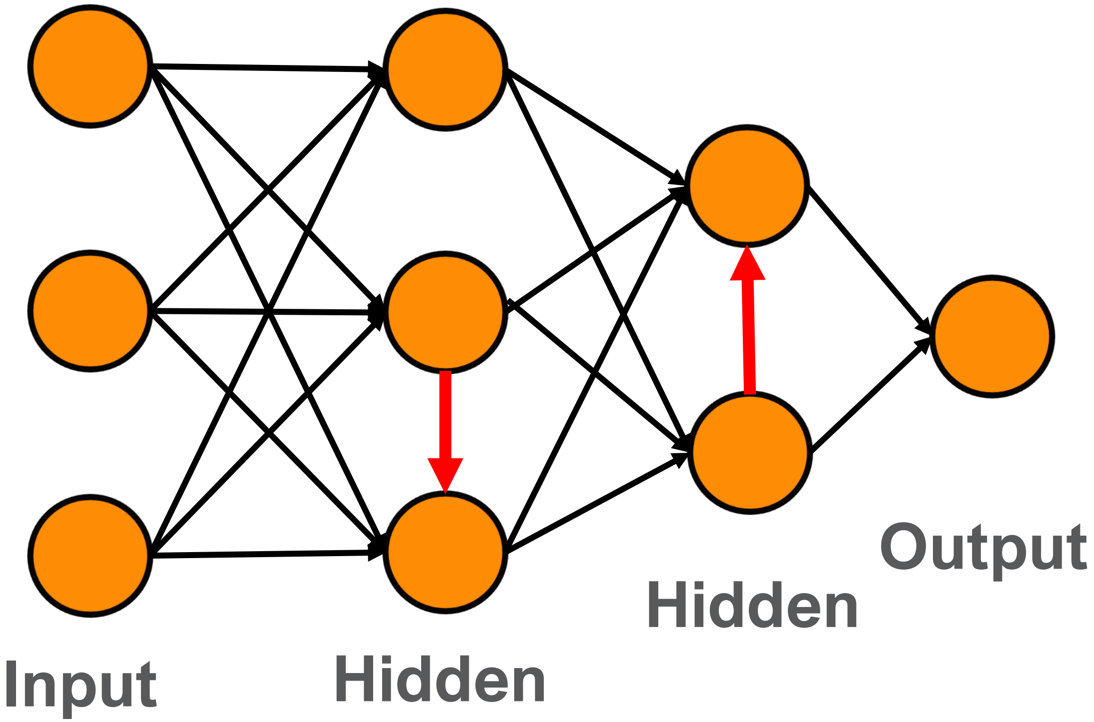
Beyond Two Layers

- The definition of neural networks and back-propagation can be extended beyond a single hidden layer to any arbitrary directed acyclic graph (DAG)
- The structure is usually in layers, where each layer is fully connected to the next



Beyond Two Layers

- However, all of the principles can be extended to any DAG
- In this example, we index our nodes in the graph as u, v.
- The activation before applying non-linearity at a node is a_u and after nonlinearity is h_u
- The graph has D-many inputs who activations are given by an input example
- An edge (u, v) is from a parent to a child.
- Each edge has a weight $w_{u,v}$
- $par(u)$ are all of the parents of u

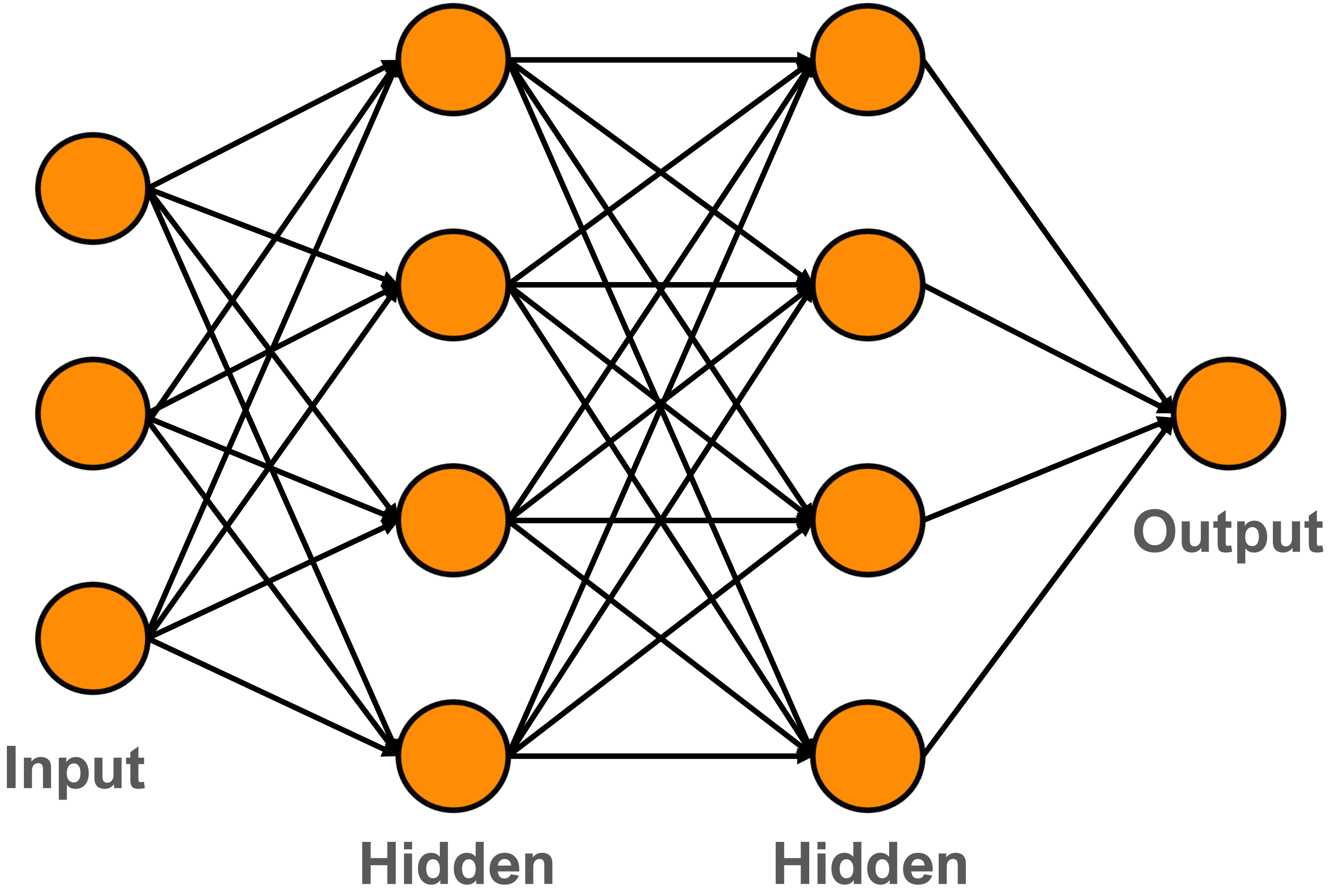


Beyond Two Layers: Forward Propagation

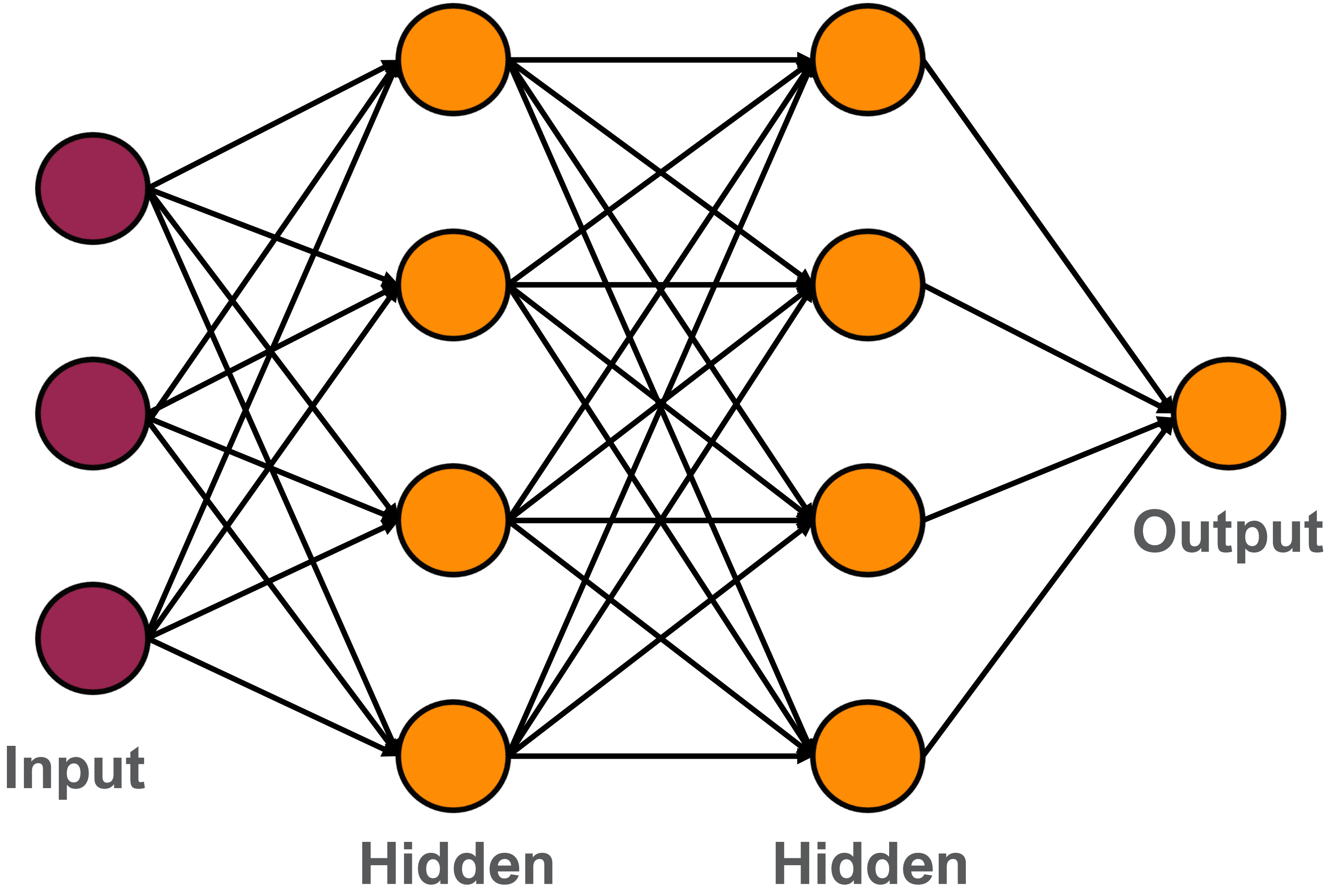
Algorithm 27 FORWARDPROPAGATION($\boldsymbol{x}$)

for all input nodes u **do**
    $h_u \leftarrow$ corresponding feature of $\boldsymbol{x}$
end for
for all nodes v in the network whose parent's are computed **do**
    $a_v \leftarrow \sum_{u \in par(v)} w_{(u,v)} h_u$
    $h_v \leftarrow \tanh(a_v)$
end for
return a_y

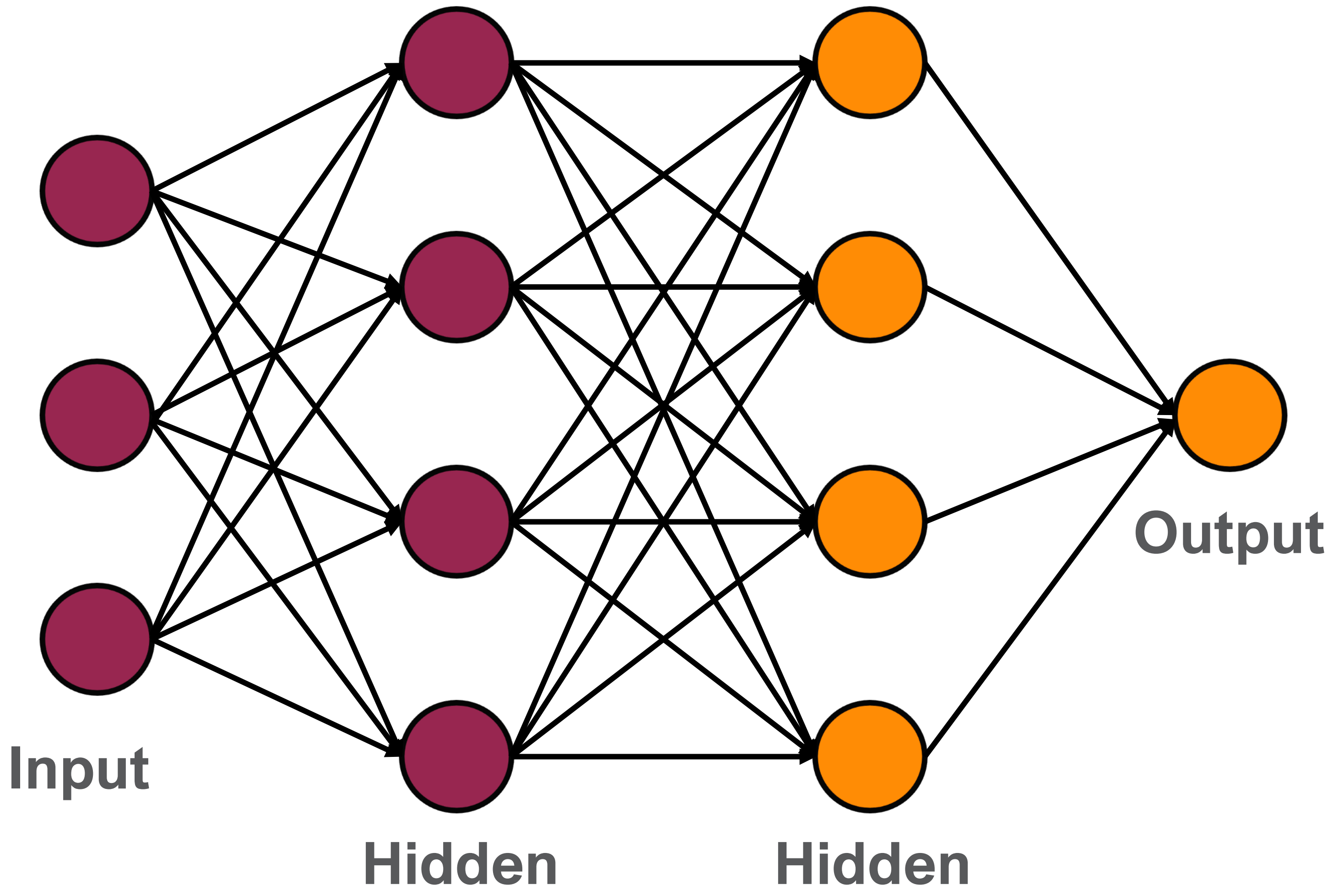
Beyond Two Layers: Forward Propagation



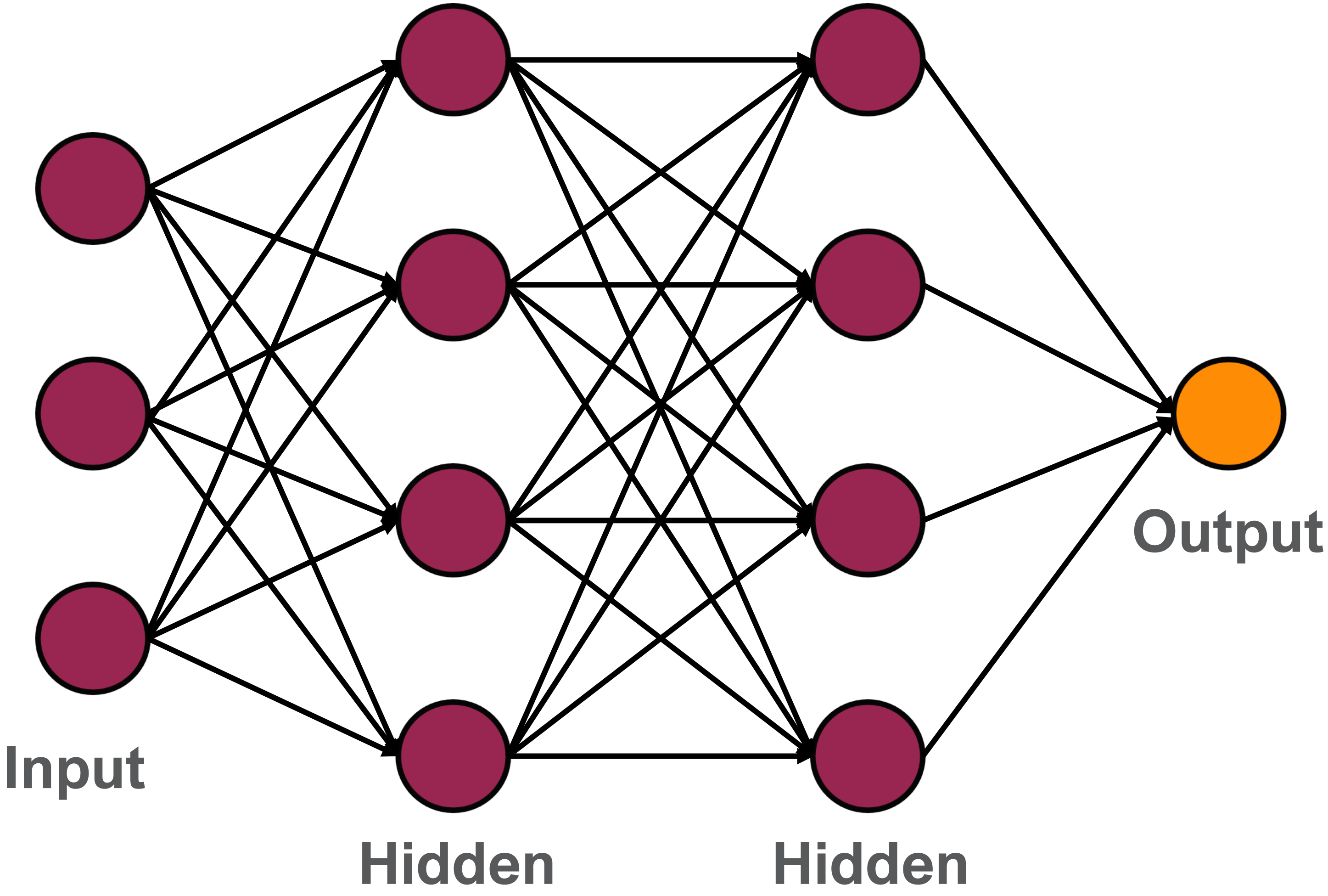
Beyond Two Layers: Forward Propagation



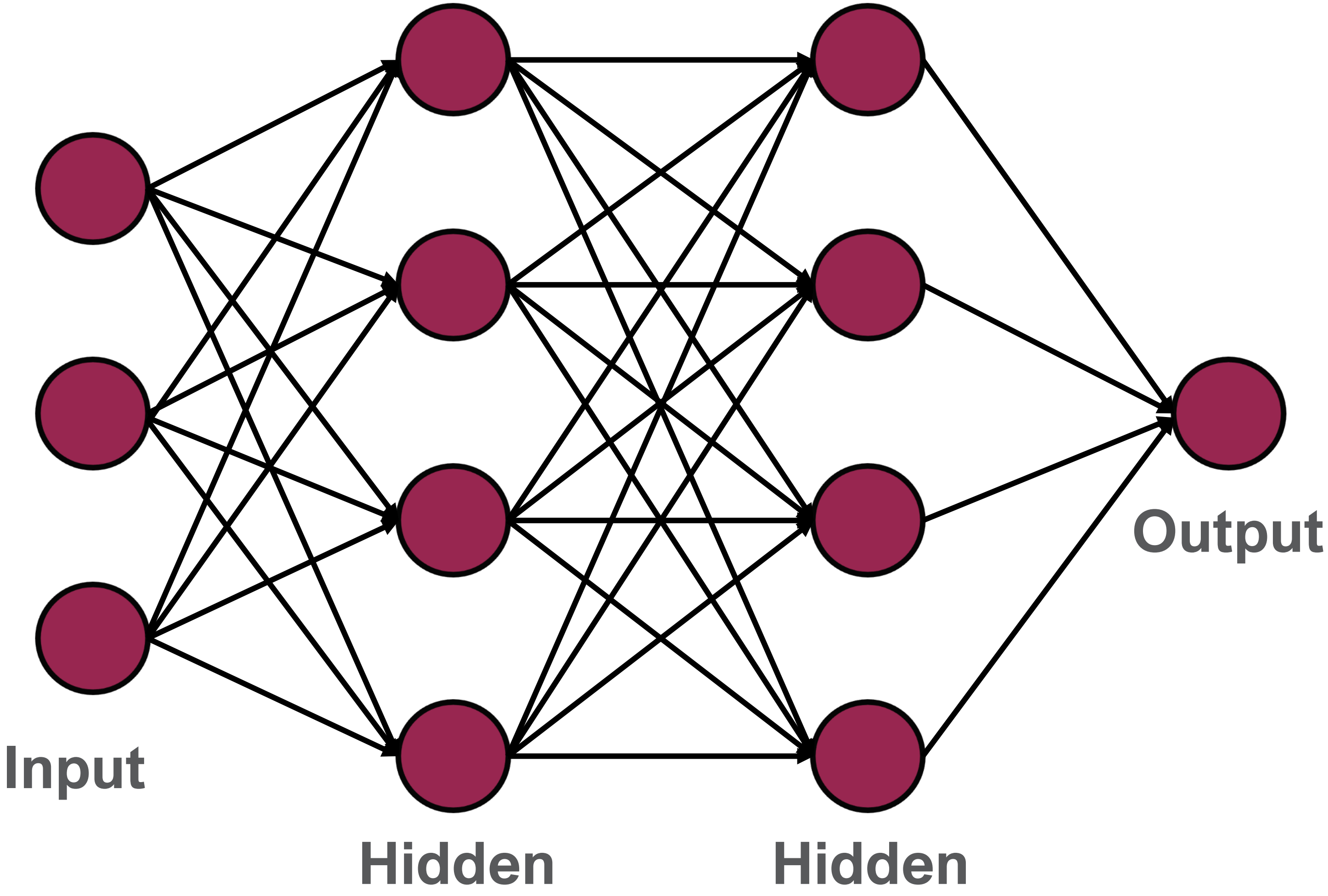
Beyond Two Layers: Forward Propagation



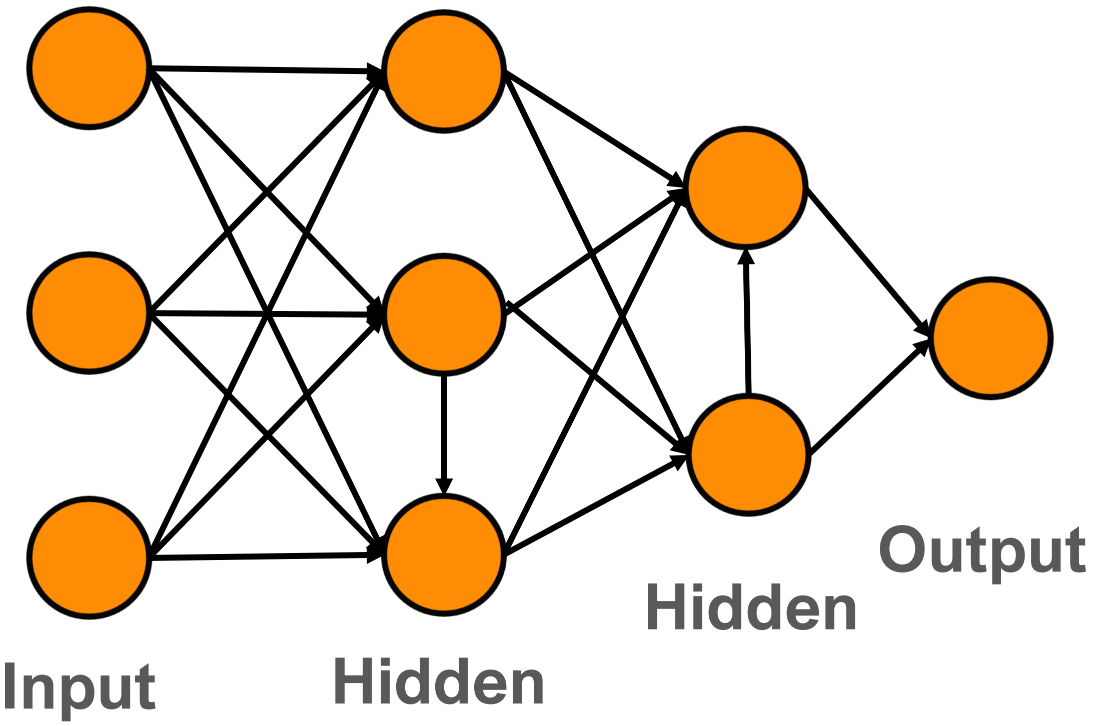
Beyond Two Layers: Forward Propagation



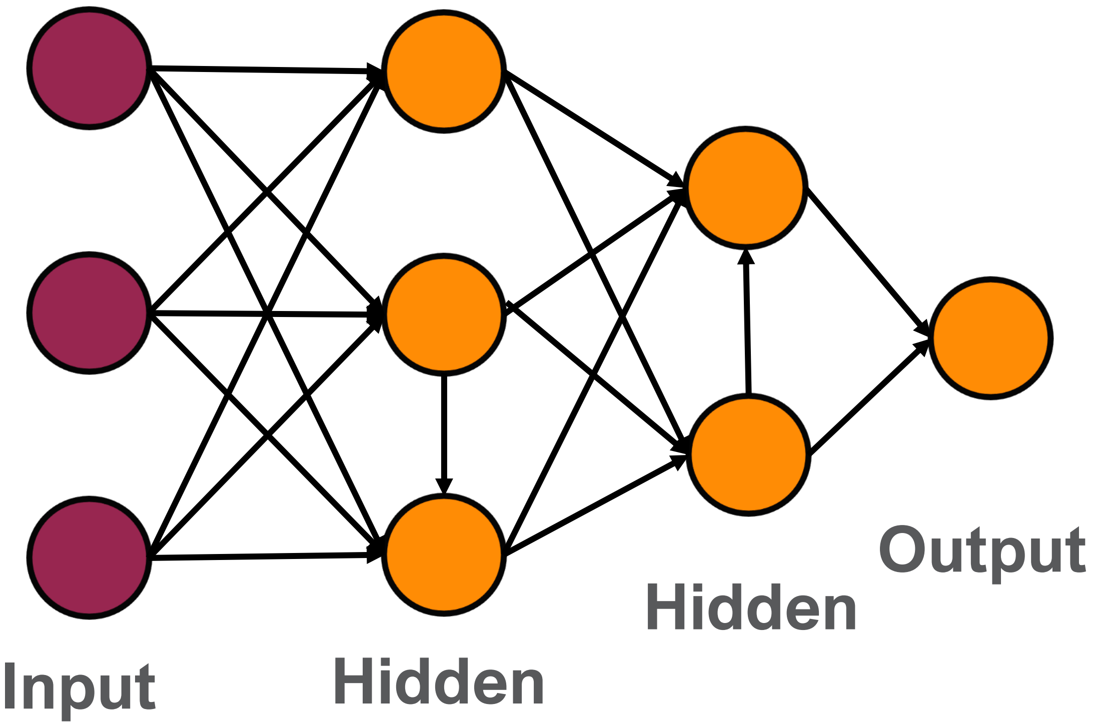
Beyond Two Layers: Forward Propagation



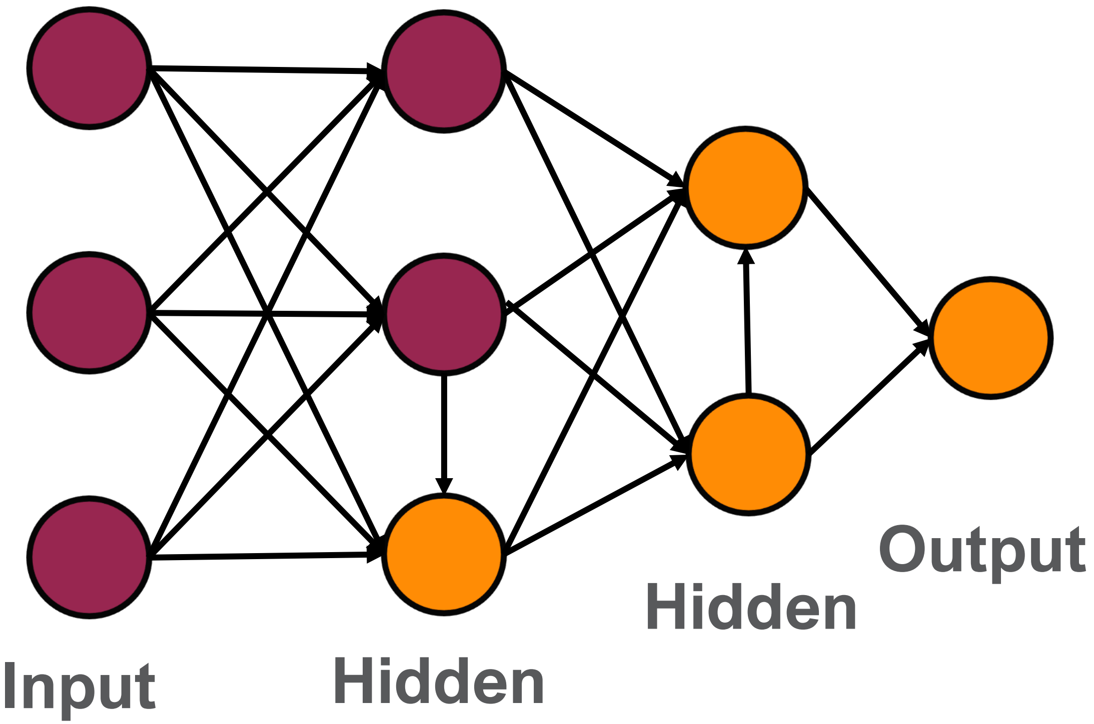
Beyond Two Layers: Forward Propagation



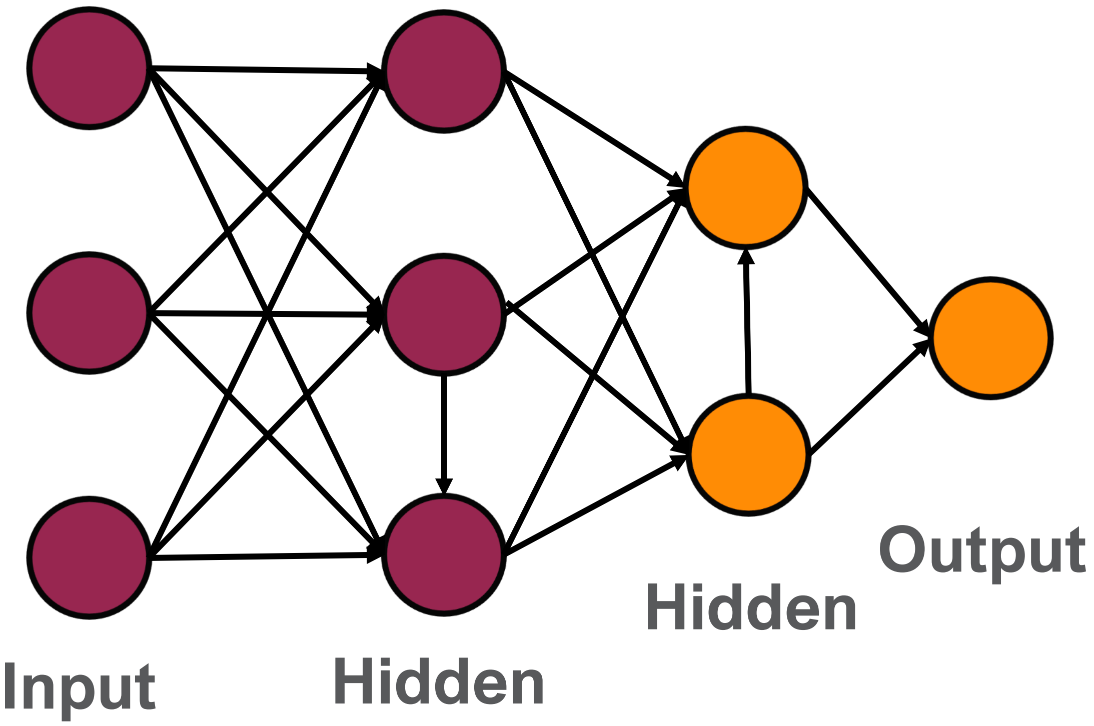
Beyond Two Layers: Forward Propagation



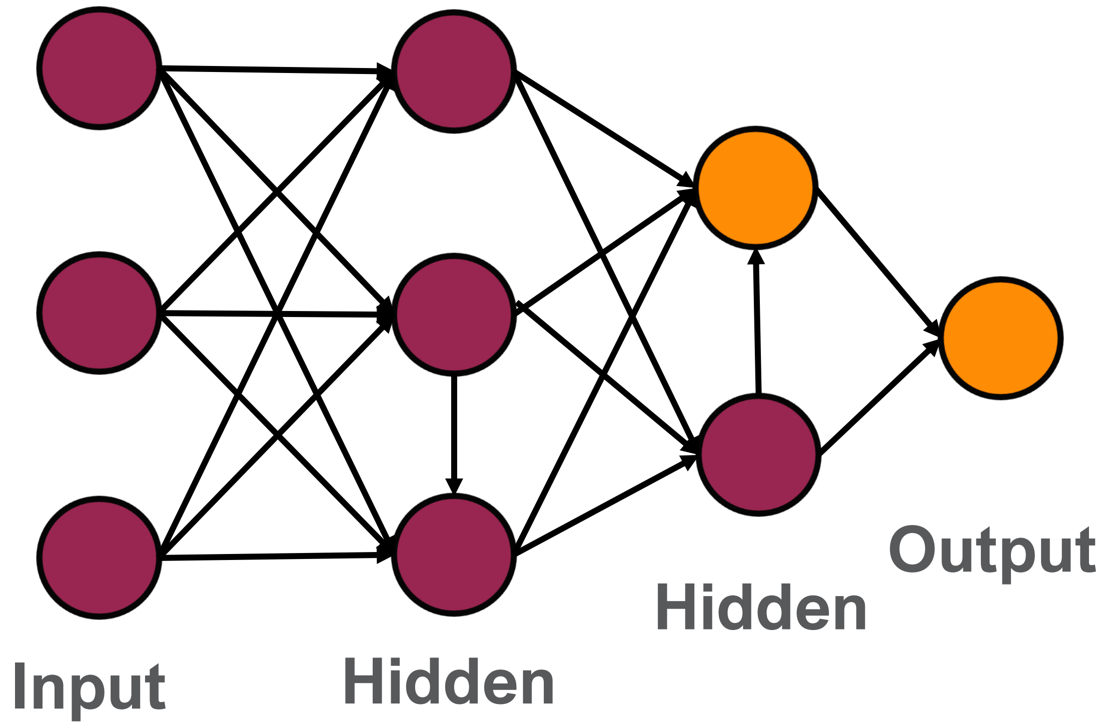
Beyond Two Layers: Forward Propagation



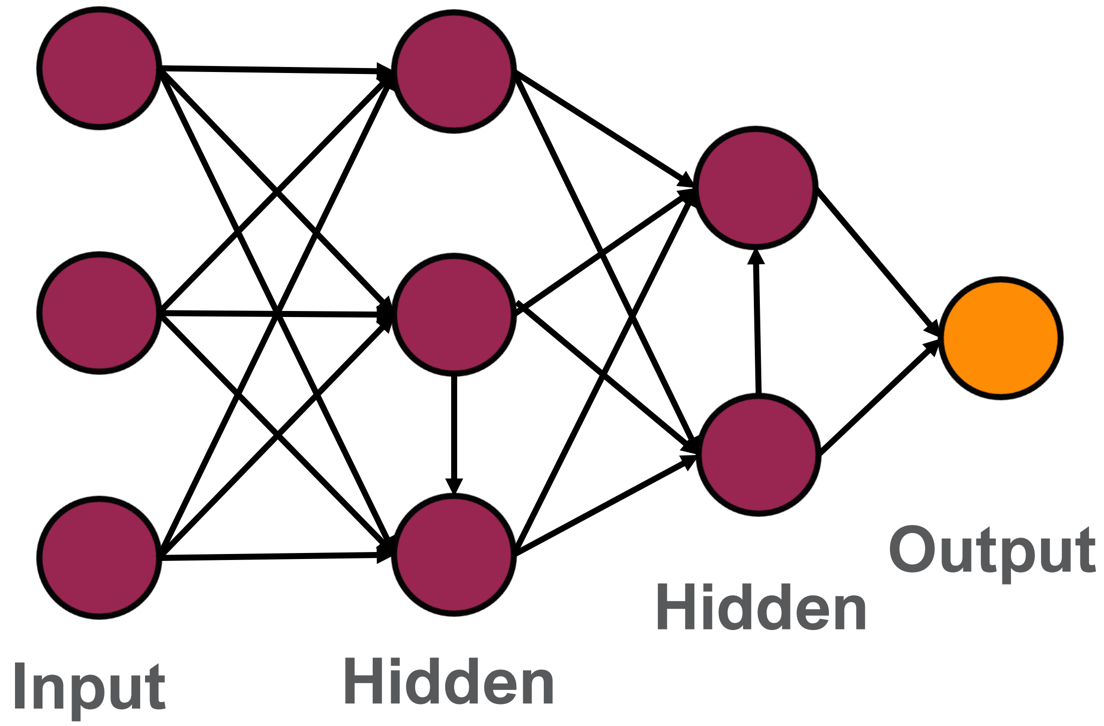
Beyond Two Layers: Forward Propagation



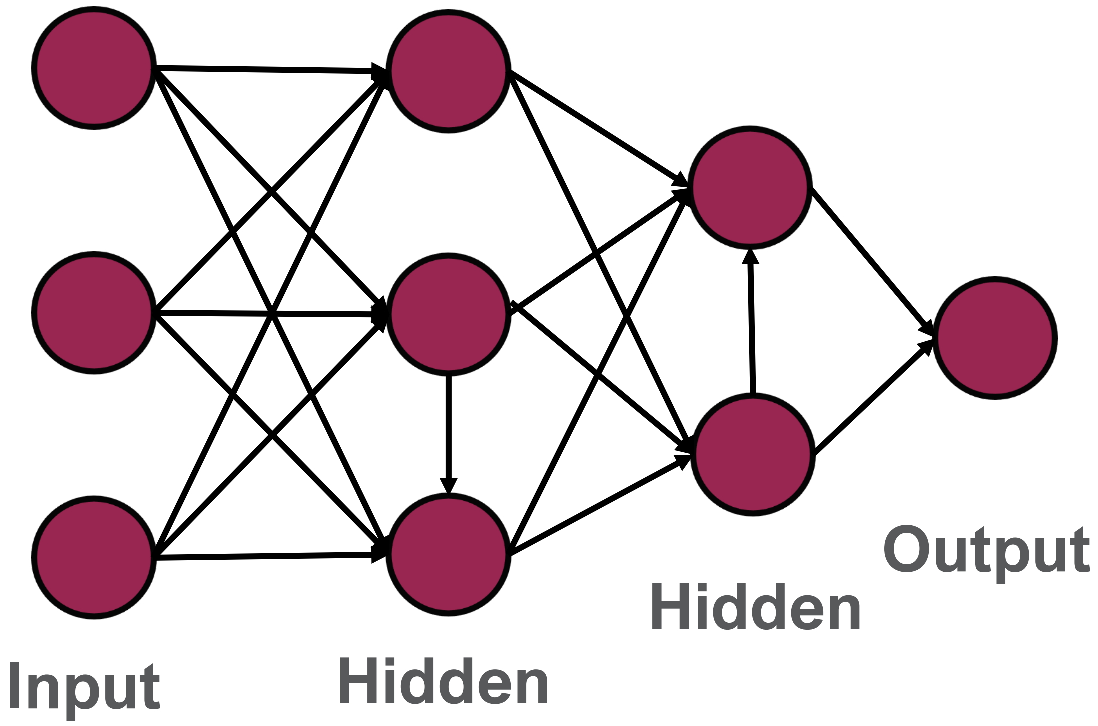
Beyond Two Layers: Forward Propagation



Beyond Two Layers: Forward Propagation



Beyond Two Layers: Forward Propagation



Beyond Two Layers: Back-Propagation

Algorithm 28 BackPropagation($\boldsymbol{x}$, y)

run ForwardPropagation($\boldsymbol{x}$) to compute activations

$e_y \leftarrow y - a_y$ // compute overall network error

for all nodes v in the network whose error e_v is computed **do**

  for all $u \in par(v)$ **do**

    $g_{u,v} \leftarrow -e_v h_u$ // compute gradient of this edge

    $e_u \leftarrow e_u + e_v w_{u,v}(1 - \tanh^2(a_u))$ // compute the "error" of the parent node

  end for

end for

return all gradients g_e

Wide or Deep?

Breadth vs. Depth

Breadth vs. Depth

- Now, we know how to train both single hidden layer networks and arbitrary DAG networks
- We also know that single hidden layer networks are universal function approximators
- If single hidden networks are so great, why do we care about deeper (more layers) networks?

Network Complexity

- The reason is that we want to keep the networks as simple as possible!

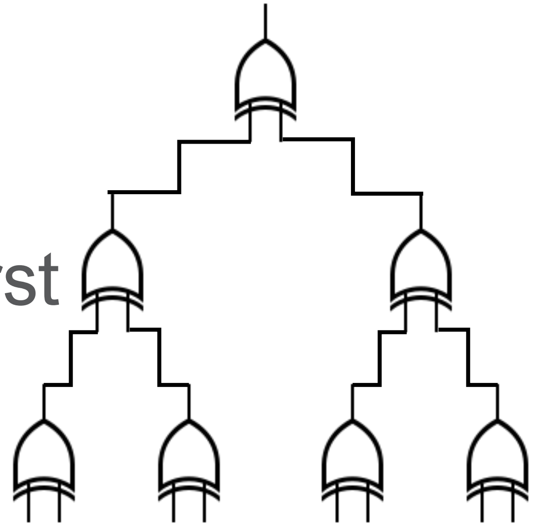
- We can show that there are functions that will require a huge number of hidden neurons if you force the network to be shallow (a single hidden layer) and only need a few neurons if you let it be deeper.

Parity Function

- The parity function is a generalization of the XOR problem:

$$parity(x) = \sum_d x_d \, mod 2 = \begin{cases} 1 & if\ the\ number\ of\ 1s\ in\ x\ is\ odd \\ 0 & if\ the\ number\ of\ 1s\ in\ x\ is\ even \end{cases}$$

- We can easily define a circuit with depth $O(\log_2 D)$ with $O(D)$-many gates for computing the parity function
- Each gate is an XOR gate
  - Organized in a complete binary tree
- You can do XOR with the network we went over in the first neural networks lecture



Parity Function

- What does this mean?
- If we can go deep with our network, we can create a circuit that computes parity with a number of hidden units that is linear in its dimensionality
- Can you do this with a shallow circuit?
- NO!

Parity Function Complexity

- Theorem: Any circuit of depth $K < \log_2 D$ that computes the parity function of D input bits must contain $O(e^D)$ gates.
- This is a famous result because it shows constant-depth circuits are less powerful than deep circuits
- It is generally believed that the same result holds for neural networks
- At the very least…it gives a strong indication that depth might be an important consideration in neural networks

Impact of Depth on Parameters

- Heuristic: You need one or two examples for every parameter
- With this heuristic, a deep model could require exponentially fewer examples to train than a shallow model
- If deep is so much better, why isn't everyone using deep networks?
- (Actually, deep networks are now the norm, i.e., deep learning)

Is deep always better?

Issues with Deep Networks

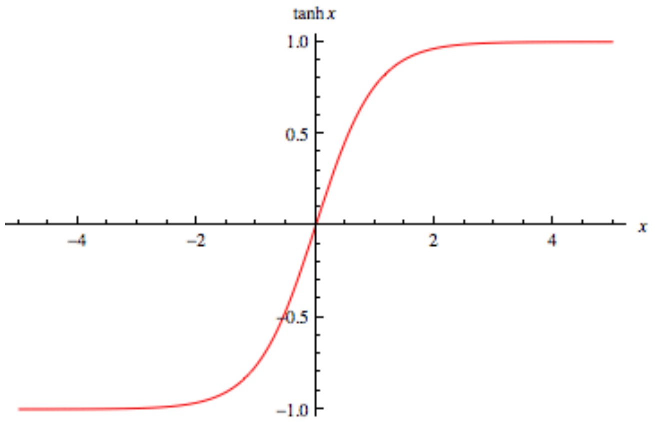
- Hyperparameter selection!
  - Deep networks have a lot more hyperparameters
- A single hidden layer network we only have to choose how many hidden units should appear in that layer
- With deep networks, we have to choose:
  - Number of layers
  - Number of units PER layer
- This can be daunting
- Hence, the field of neural architecture search

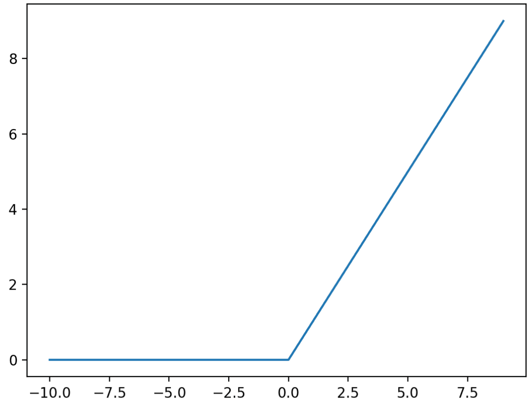
Issues with Deep Networks

- As back-propagation makes its way backward through the model, the sizes of the gradients shrink
- If you are at the beginning of a deep network, changing one weight is unlikely to have a huge impact on the output, because it has to go through so many other neurons/units before it gets to the output
- So, the derivatives at the beginning of deep networks are small!
  - Back-propagation never really moves weights at the beginning of deep networks far from their random initialization
- This is called the vanishing gradients problem!

Addressing these issues

- Obviously, we've overcome these issues with recent, massive, deep networks
- You can change the way initialization is done on a layer-by-layer basis, maybe in an unsupervised way
- You can use more sophisticated optimization beyond gradient descent
- Batch normalization can be used to help
- Other activation functions (like ReLU) that suffer less than activations like sigmoid
- Skip connections



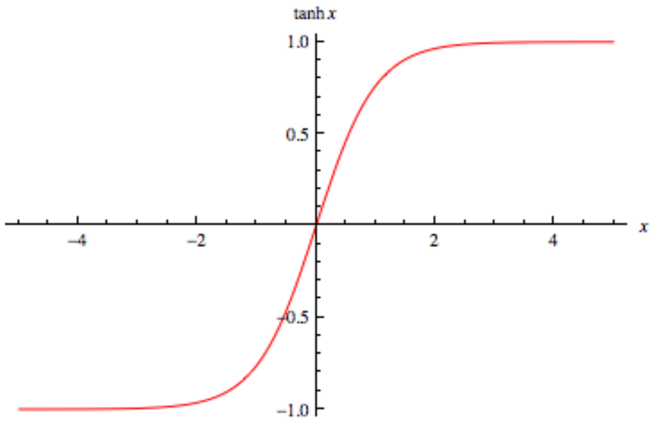


Basis Functions

Neural Networks

- We know neural networks can mimic linear functions
- We also know they can learn more complex functions
- Can they learn to mimic a KNN classifer?
- Can they do it efficiently (i.e., without too many hidden units)?

Swapping Activation Functions



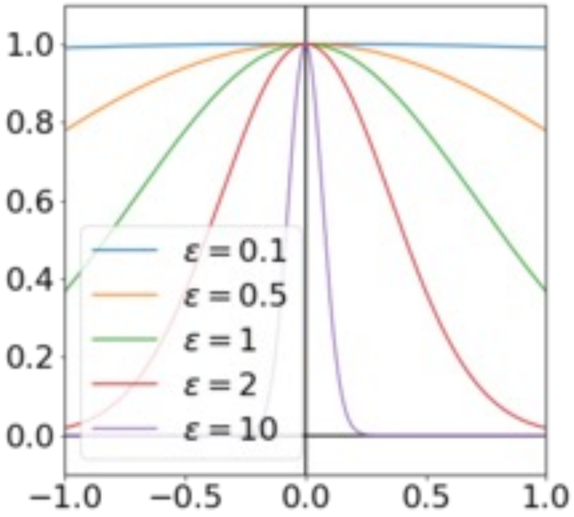
- In a sigmoid network, we use:

$$h_i = \tanh(w_i \cdot x)$$

- We can change the activation function to instead be the radial basis function (RBF):

$$h_i = \exp[-\gamma_i \|w_i - x\|^2]$$

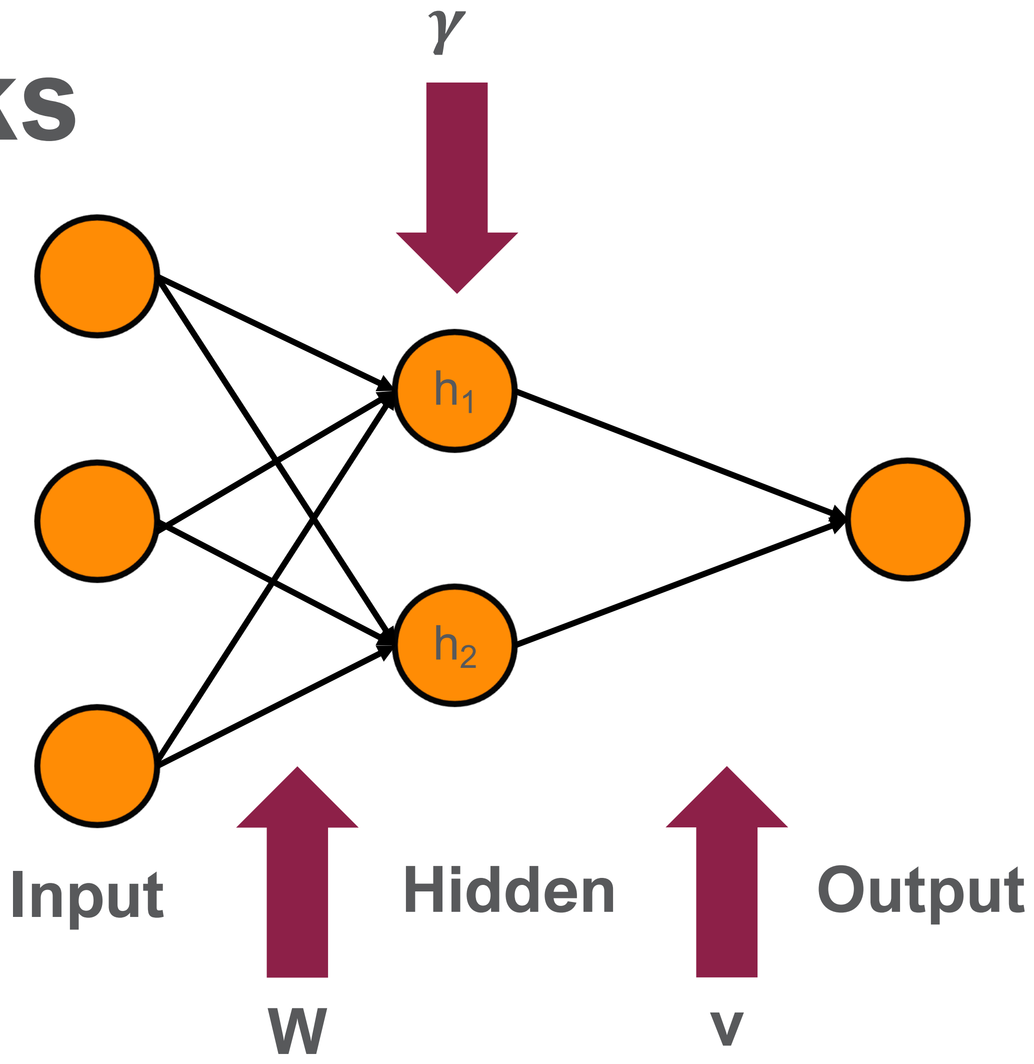
- Intuitively, these are little Gaussian bumps around the locations specified by the weight vectors, where γ_i specifies the width of the bump
- If γ_i is large, then only those that are really close to w_i have non-zero activations



Training RBF Networks

- To train a Radial Basis Function (RBF) network, we have to find the Gaussian widths γ_i, the centers of the Gaussian bumps w_i, and the connections between the Gaussian bumps and the output unit v.

- We calculate v the same way, but our derivatives change for the other variable.

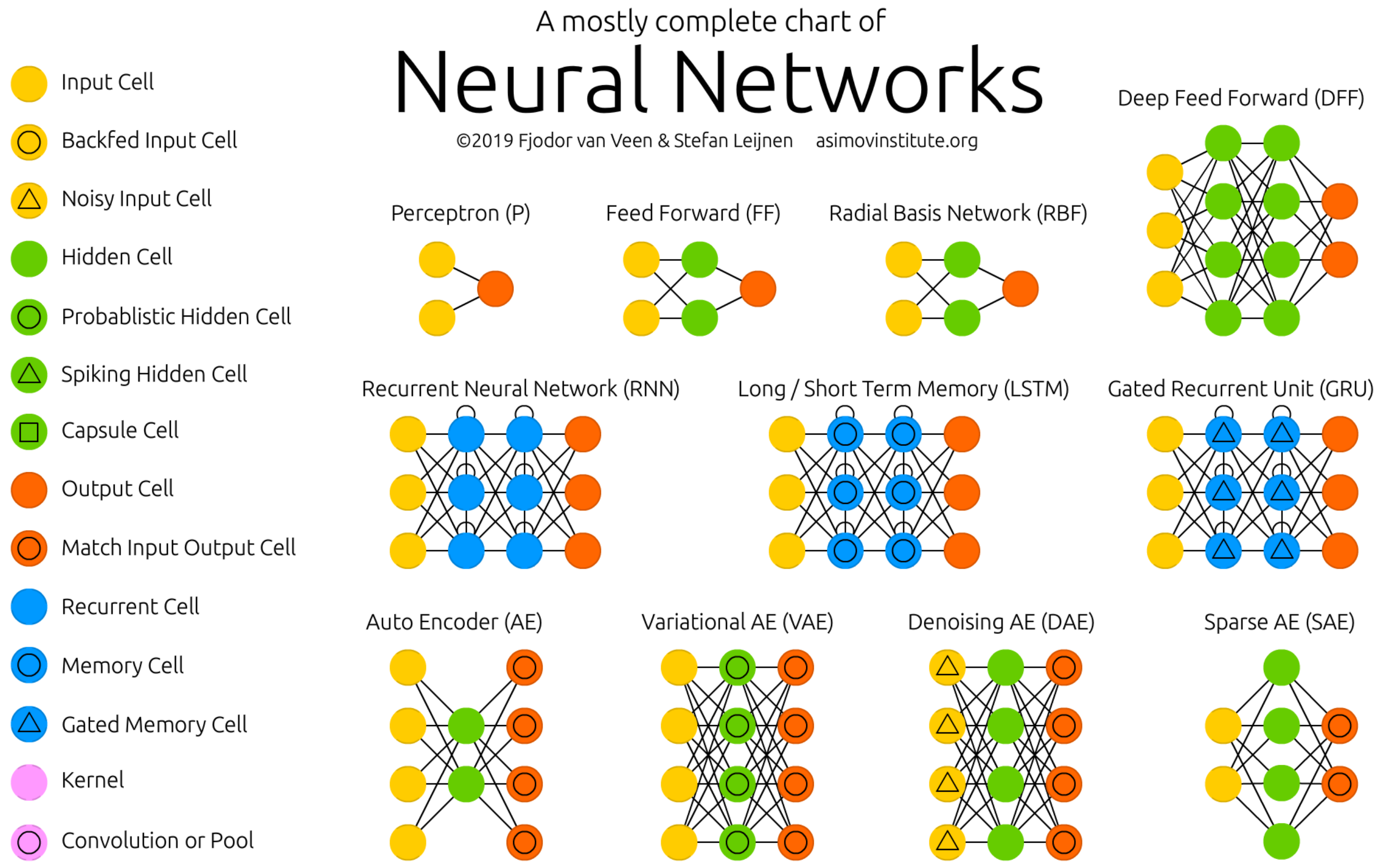


RBF Networks

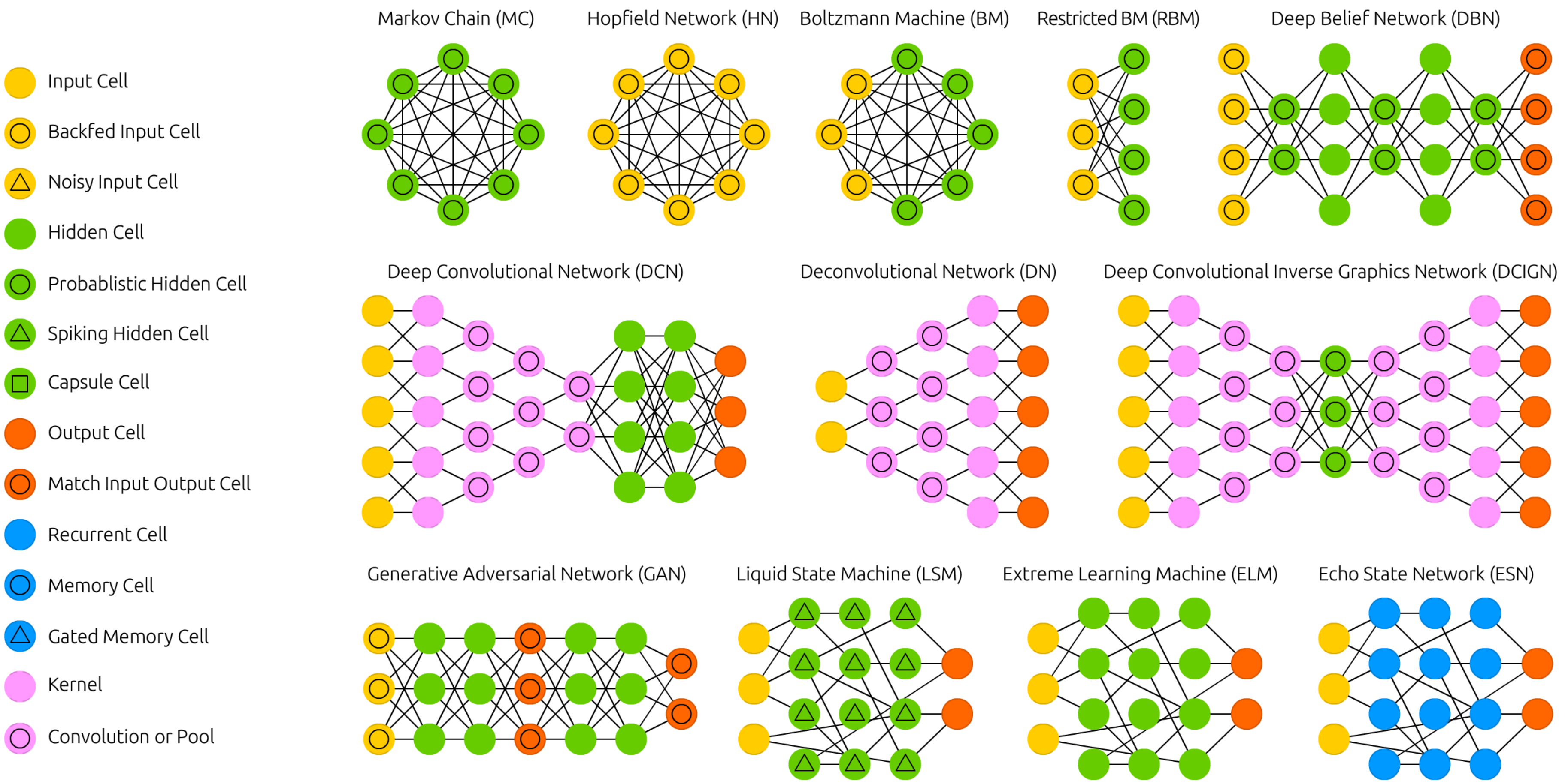
- Where should the Gaussian bumps be centered?
- You can apply back-propagation to find the weights OR you could specify ahead of time
- In particular, if you carefully choose γs and vs, you can obtain something that actually looks a lot like distance-weighted KNN
- You can go further! Use back-propagation to learn good Gaussian widths (γ_i) and voting factors (v) for the nearest neighbors approach!

We’ve barely scratched the surface

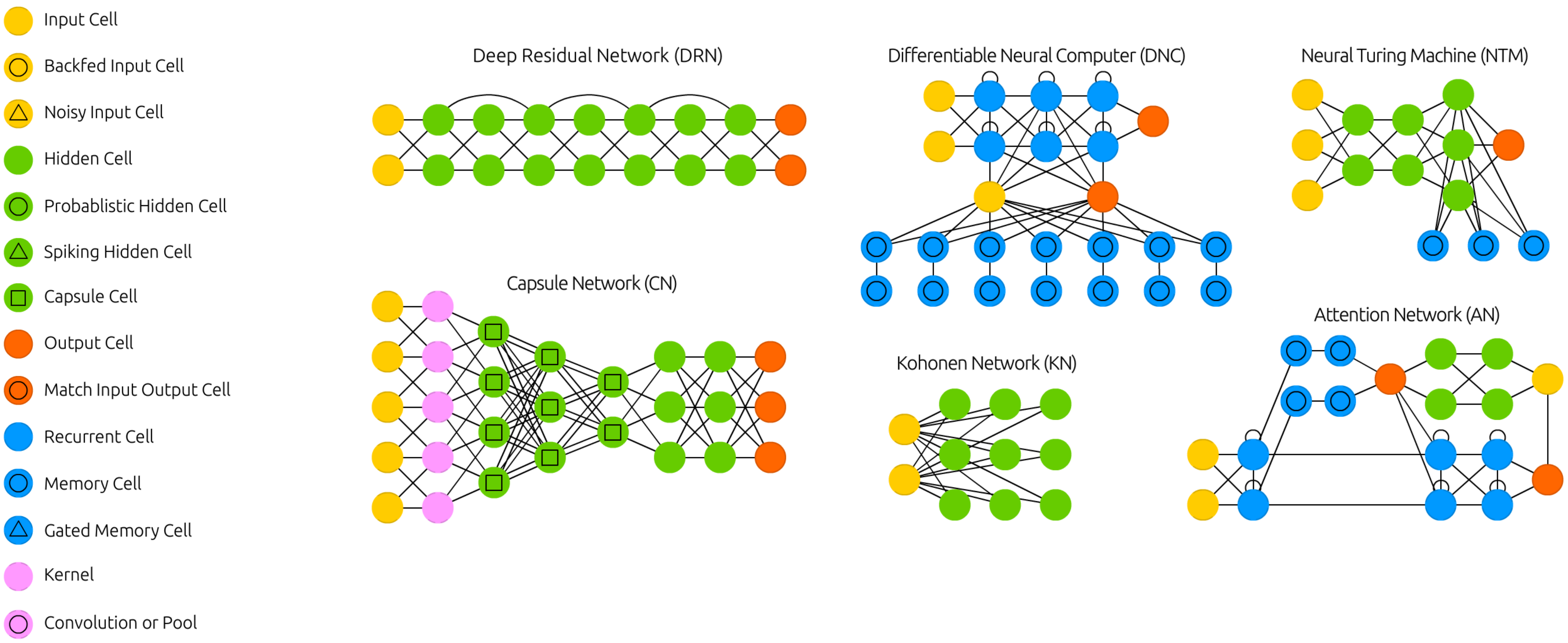
- Later in the semester we’ll briefly cover deep learning
- There are lots of types of artificial neural networks and we definitely do not have time to cover them all in this class!



More Neural Networks



More Neural Networks



If you want to learn more about different types of neural networks…

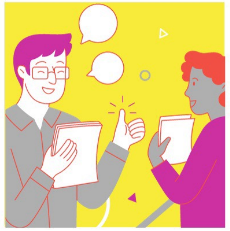
- COSC 420/COSC 527: Biologically-Inspired Computing
  - Offered in spring
  - Lots of topics, but it includes Hopfield networks, liquid state machines, spiking neural networks
- COSC 525: Deep Learning
  - Offered in spring
  - Convolutional neural networks, autoencoders, generative adversarial networks, recurrent neural networks

Announcements

- I will not have office hours tomorrow
- ***NO CLASS this Thursday, October 26***
- Lab 4 is due on Friday, October 27
- Final Projects:
  - Finalization of dataset to be used by **November 3.**
    - 5-point penalty if the dataset is not selected by that date.

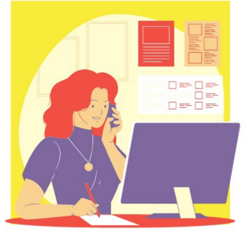
Announcements

EECS MINI INTERNSHIP & RESEARCH & CAREER FAIR



TUES
OCT 24TH
5:30-7:00 PM

MIN KAO
1ST & 2ND FLOOR
ATRIUM



Join us for EECS Mini Research/Internship/Career Fair!
Speak to companies and professors in a more relaxed environment.
Wearing jeans or business casual is recommended.
Don't miss a great networking experience! (Dinner provided)

HOSTED BY SYSTERS: WOMEN IN EECS