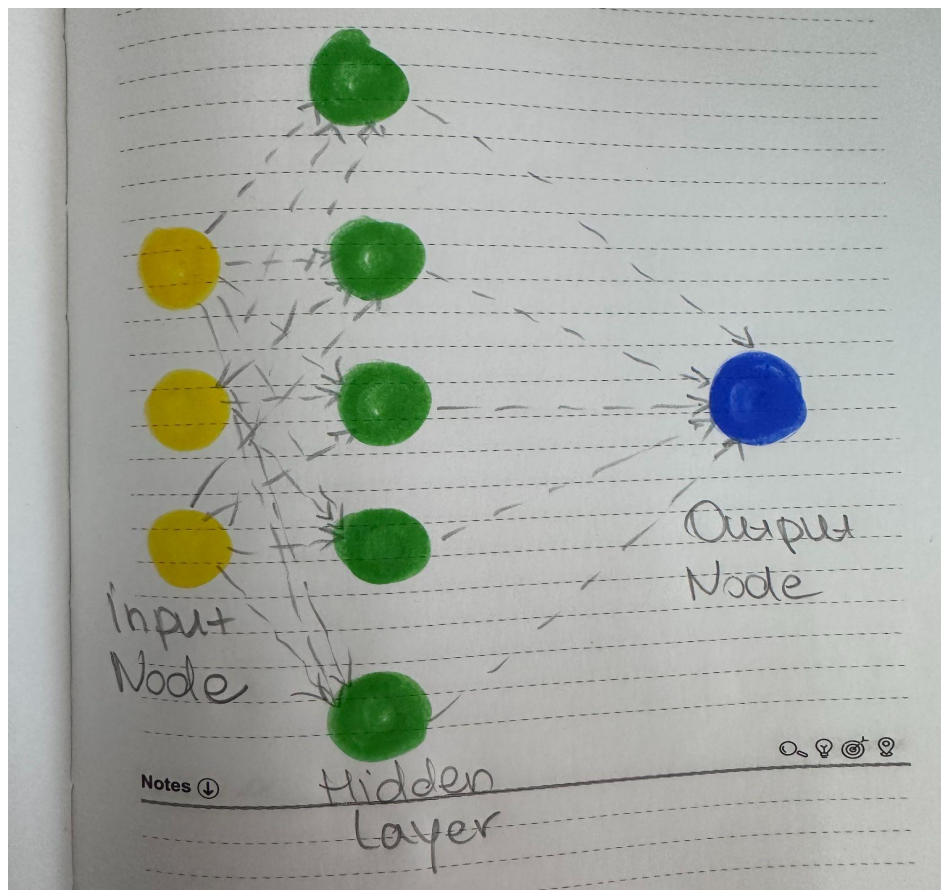


Conceptual Questions [25 pts]:

a. [4 pts] In your own words, what is a feed-forward neural network (FFNN)? Provide a graphical illustration for a feed-forward neural network with input, hidden and output layers. Supporting your answer with mathematical expressions is certainly encouraged.

Answer: A feed-forward neural network is a type of network that takes input data and processes it through multiple cross-connected layers and produces an output or a prediction.

Below is a graphical representation of FFNN with input, hidden and output layers:



Mathematically, it can be expressed as: $z = w * x + b$

- Each neuron in the input layer receives an input value denoted as " x ", which could be a set of numbers.
- This input value " x " is multiplied by a weight " w " and added to a bias " b "
- The result of this multiplication and addition is the weighted sum " z "

b. [4 pts] In your own words, what is a convolutional neural network (CNN)? What is the difference compared to a feed-forward neural network?

Answer: A CNN (convolutional neural network) is a type of neural network designed to process grid-like data. Ex: images. It uses special layers called convolutional layers to automatically learn patterns or features from the data.

The main differences between CNN and FFNN are that CNN have convolutional layers for capturing local patterns, using local connectivity and parameter sharing for efficient learning, and often incorporate pooling layers for downsampling. Overall, CNNs are specialized more for image-like data processing.

c. [4 pts] Explain the principle of minimizing a cost function C using gradient descent.

Answer: Gradient descent is a backbone of ML. It's a method that is used to minimize a cost function in machine learning by taking small steps in the direction of the steepest decrease in the cost function until the best model parameters are found.

It is like finding the quickest way downhill. It adjusts the model's parameters in small steps based on the slope of the cost function C . By constantly moving in the direction of steepest decrease, it helps the model find the best parameters that minimize the cost function C and improve its predictions.

d. [5 pts] In your opinion, why is it useful to use non-linear activation functions in a neural network? Conceptual or technical answers are both accepted.

Answer: Non-linear activation functions are useful in neural networks because they allow the model to learn complex patterns, capture diverse representations, and approximate any arbitrary function, making it more powerful, flexible, and capable of generalization.

e. **[5 pts]** In a multi-classification problem with K classes, a neural network has an output layer with size $K > 2$. Explain why having a size K instead of a size 1 at the output layer is useful for multi-classification. Second, explain why using 'Softmax' activation is better compared to 'Sigmoid' activation at the output layer with size K .

Hint for first question: Consider the case of $K = 3$ classes, when the prediction we want to express is 50% to get class 0 and 50% chance to get class 2.

Remark: In this context a linear activation function means the identity function.

Answer: Having a size K (greater than 1) at the output layer is better and allows the model to produce probabilities for K classes, making it easier to interpret and analyze.

'Softmax' activation is better than 'Sigmoid' activation at the output layer because it produces valid probability distributions and captures class relationships, leading to more accurate predictions, while 'Sigmoid' may suffer from issues like ambiguous predictions and lack of class dependencies.

f. **[3 pts]** Think of all the models and algorithms you've learned so far. Specifically, k-Nearest Neighbors (k-NN), Decision Trees and Linear Regression. What are some of the advantages and disadvantages of Neural Network models compared to those? What types of problems might Neural Networks be better or worse suited to than these algorithms (e.g. problems with more or less data, supervised vs unsupervised learning, etc.)?

Remark: This is a more open-ended question. We just want to see how you think- you don't need to compare/contrast neural networks to each of those three models in detail, no need to cover all bases. Feel free to just pick one of the models you have seen (e.g. Decision Trees, although you are welcome to do more). Keep your answers as short as you can.

Answer:

Neural Networks vs. XGBoost:

Advantages of Neural Networks:

- Learn complex patterns from large data
- Handle high-dimensional data and nonlinear relationships
- Can be used for supervised and unsupervised learning
- Adapt and generalize well to new data

Disadvantages of Neural Networks:

- Computationally expensive
- May require large amounts of data
- Lack interpretability
- Prone to overfitting

Neural Networks are better for:

→ Large datasets, complex patterns, adaptability

XGBoost is better for:

→ Smaller datasets, simple patterns, interpretability, faster training/inference.