DataLab Preparation (Week 3, DataLab I, Monday)

**2a Describe in your own words the basic structure of a perceptron. What are the key components, and how do they function together to make a prediction?**

A perceptron is a type of artificial neuron used in neural networks. The key components are: binary inputs, weights, and a single binary output. It works by taking several inputs and producing an output. The importance of the inputs in making a prediction is determined by the individual weights attached to each input. The output (0 or 1) is then determined by wether the weighted sum is less than or greater than a set threshold value.

**2b Explain the role of weights and bias in a perceptron. How are they initialized, and what is their significance in the perceptron's output?**

The role of weights is to determine the importance of each input, i.e., how big the effect each input variable has on the output.

Bias is simply the threshold moved on the other side of the equation. It determines how easy it is to output a 1.

Weights and biases are initialized randomly at the beginning of training.

**2c Describe the architecture of a multilayer perceptron. How does it differ from a single-layer perceptron in terms of structure and processing capabilities?**

A multilayer perceptron (MLP) is composed of interconnected layers of neurons, including input, hidden, and output layers. Unlike single-layer perceptrons, MLPs have hidden layers that enable them to learn complex relationships in data. Each neuron in a layer is connected to every neuron in the the next layer and so on. Unlike single-layer perceptrons, MLPs capture intricate patterns in data, making them a better choice for various tasks requiring advanced learning capabilities.

**2d In a multilayer perceptron, what is the role of each layer? How does the data transform as it moves from the input layer to the output layer through hidden layers?**

In a multilayer perceptron, we have 3 types of layers: input layer, hidden layer(s), and output layer. The input layer receives the raw input data, such as features of a dataset. The hidden layer(s) is/ are responsible for learning and extracting features from the input data. The output layer receives the transformed data from the hidden layers and produces the final output of the MLP.

As data moves from the input layer to the output layer through hidden layers, it undergoes a series of transformations. Each hidden layer applies linear transformations (weighted sum) followed by nonlinear transformations (activation functions) to the input data, resulting in the extraction of increasingly complex features.

**2e How do multiple layers in an MLP improve its ability to model complex functions compared to a single-layer perceptron?**

In a single-layer perceptron, which consists of an input and output layer only, the model can only learn linear decision boundaries, limiting its ability to capture complex relationships in the data. However, the hidden layers in an MLP allow for the extraction of nonlinear features from the input data.

For example, when trying to classify images of handwritten digits, a single-layer perceptron would struggle to distinguish between different digits because the relationship between pixel values in the input image and the target class is nonlinear and complex. However, using an MLP with multiple hidden layers allows the model to learn hierarchical features such as edges, shapes, and textures from raw pixel values.

**2f When designing a neural network to handle image inputs, what defines the number of neurons in the input layer of the network?**

The number of neurons in the input layer of such a network is determined by the number of pixels of the images. For example, if we have 28x28 pixel images, in the input layer there will be 784 (28\*28) neurons.

**2g How many neurons would be needed in the input layer of a neural network designed to receive images of size 64 x 64 pixels as input?**

If the images are grey-scale, 64 \* 64 = 4096 input neurons,

If they are colored images, 64×64×3=12,288 input neurons.