Chapter 1

Q1

- False
- False
- False
- False

Q2

- Natural language processing
- Computer vision
- Playing games
- Medicine
- Recommendation systems

Q3

Mark 1 Perceptron built by Frank Rosenblatt

Q4

- A set of processing units
- A state of activation
- An output function for each unit
- A pattern of connectivity among units
- A propagation rule for propagating patterns of activities through the network of connectivities
- An activation rule for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit
- A learning rule whereby patterns of connectivity are modified by experience

- An environment within which the system must operate

Q5

- In 1969, Minsky and Papert demonstrated in 'Perceptron' that a single layer of these devices was unable to learn some simple but critical mathematical functions (such as XOR). In the same book, they also showed that using multiple layers of the devices would allow these limitations to be addressed. Unfortunately, only the first of these insights was widely recognized. As a result, the global academic community nearly entirely gave up on neural networks for the next two decades.
- In the 1980's most models were built with a second layer of neurons. In theory, adding just one extra layer of neurons was enough to allow any mathematical function to be approximated with these neural networks, but in practice such networks were often too big and too slow to be useful. Although researchers showed 30 years ago that to get practical good performance you need to use even more layers of neurons, it is only in the last decade that this principle has been more widely appreciated and applied.

Q6

A special kind of processor in a computer that can handle thousands of single tasks at the same time, especially designed for displaying 3D environments on a computer for playing games. These same basic tasks are very similar to what neural networks do, such that GPUs can run neural networks hundreds of times faster than regular CPUs.

Q7

2

Q8

—

Q9

—

Q10

It is hard to define set rules for a traditional computer to recognize a cat or a dog which is not limited to a subset of images but scales to all given images of a cat or a dog. The different postures, different colors, etc can change these fixed rules completely, making it impossible to define fixed rules to identify a cat or a dog.

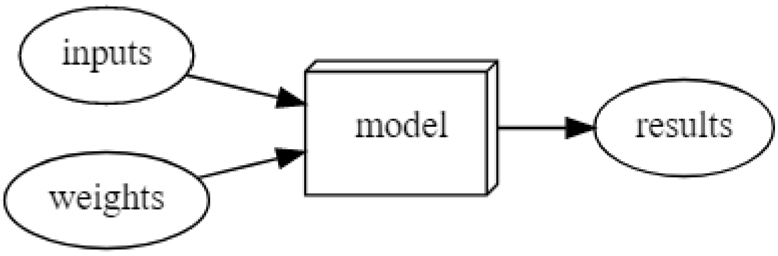
Q11

Weight assignment refers to the values of the parameters of the model. The weight assignment results in a certain performance, the meadurement and maximization of which (by altering the weights) results in the model learning how to solve a particular problem.

Q12

Parameters.

Q13



Q14

Simply, we have some input variables/data that are multiplied by some weights, giving us an output. We can understand which variables are more important and which are less important based on their weights. A similar logic might apply for a small neural network with 1-3 layers. However, deep neural networks have hundreds, if not thousands, of layers. It is hard to determine which factors are important in determining the final output. The neurons in the network interact with each other, with the outputs of some neurons feeding into other neurons. Altogether, due to the complex nature of deep learning models, it is very difficult to understand why a neural network makes a given prediction.

Q15

Universal Approximation Theorem

Q16

- Input data
- An architecture for the given problem
- An initial set of parameters
- Labels for the input data
- A loss function
- A way to update parameters (optimiser)

Q17

In a predictive policing model, we might end up with a positive feedback loop, leading to a highly biased model with little predictive power. For example, we may want a model that would predict crimes, but we use information on arrests as a proxy . However, this data itself is slightly biased due to the biases in existing policing processes. Training with this data leads to a biased model. Law enforcement might use the model to determine where to focus police activity, increasing arrests in those areas. These additional arrests would be used in training future iterations of models, leading to an even more biased model. This cycle continues as a positive feedback loop.

Q18

No. Increasing the memory size would lead to better performance at the cost of speed and memory usage.

Q19

Classification predicts a category while regression predicts a continuous quantity.

Q20

- The validation set is the portion of the dataset that is not used for training the model, but for evaluating the model during training. It is required to ensure the model is not overfit to the training data (that is the model performance is not due to the fact that the model has essentially memorized the training dataset instead of learning generalizaed patterns).

- Test set is a completely unseen part of the data used for final evaluation of the model. This is to make sure that the model generalizes to unseen data and is not tunes for the validation dataset specifically.

Q21

- It will randomly take 20% of the data and assign it as the validation dataset.

Q22

It should be representative of the new data we will see in the future. Sometimes random datasets don't work for that. For example for time series data set, you want a newer time-data for your validation dataset and an even newer time-data for your test dataset.

Q23

Overfitting refers to when the model fits too closely to a limited set of data but does not generalize well to unseen data. This is especially important when it comes to neural networks, because neural networks can potentially “memorize” the dataset that the model was trained on, and will perform abysmally on unseen data because it didn’t “memorize” the ground truth values for that data.

Q24

A metric is a function that serves as the measure of performance of the model's predictions on the validation set. Not the same as loss (internal measure of performance of the model used for the SGD step). Can sometimes be used in place of loss.

Q25

Pretrained models have been trained on other problems that may be quite similar to the current task. Pretrained models often contain domain knowledge, and hence can be further used for transfer learning to create accurate models for the current task with less data and time.

Q26

During transfer learning using a pretrained model, we often remove the last layer, since that is always specifically customized to the original training task , and replace it with one or more new layers with randomized weights, of an appropriate size for the dataset we are working with. This last part of the model is known as the head.

Q27

Earlier layers learn simple features like diagonal, horizontal, and vertical edges. Later layers learn more advanced features like car wheels, flower petals, and even outlines of animals.

Q28

Image models can be useful for other types of images like sketches, medical data, etc. Image models can also be used for other information represented as images such as spectrograms, timeseries plot, etc.

Q29

Functional form of the model, or the structure or network we are trying to fit.

Q30

At its core, segmentation is a pixelwise classification problem. We attempt to predict a label for every single pixel in the image. This provides a mask for which parts of the image correspond to the given label.

Q31

y_range is being used to limit the values predicted when our problem is focused on predicting a numeric value in a given range

Q32

Training models requires various other parameters that define how the model is trained. For example, we need to define how long we train for, or what learning rate (how fast the model parameters are allowed to change) is used. These sorts of parameters are hyperparameters.

Q33

1. Make sure a training, validation, and testing set is defined properly in order to evaluate the model in an appropriate manner.
2. Try out a simple baseline, which future models should hopefully beat.