SAMPLE QUESTIONS (TERMS, DERIVED) by lecture slides

1. Intro

- What is a **regression** problem?
  - Outputs are continuous numerical values (i.e. property value)
- What is a **classification** problem? (pattern recognition)
  - Outputs are discrete, categorical values (i.e. dog or cat)
- What is a **sequential decision-making** problem?
  - Make a series of decisions over time in order to make the best decision possible, an example would be Markov Decision Process. (i.e. chess, motion planning).

2. Intro

- What is **supervised learning**?
  - Data is labeled (input features and target labels), expensive.
  - Classification and Regression.
- What is **unsupervised learning**?
  - Data is unlabeled, 'learns without teacher'
  - Cluster application, anomaly detection, density estimation.
- What is **semi-supervised learning**?
  - Uses both. (i.e. photos identification)
- What is **reinforcement learning**?
  - Rewarded/punished based on outcome
  - Finding optimal policy to maximize return
  - Limitations: needs extensive learning data, lack generalization
- What is the method of **Lagrange multipliers**?
  - Used to solve optimization problems w/ additional constraints.
- What is the **Bayes Theorem**?
  - Fundamental concept in probability theory (discrete (adds to 1) and continuous (sum of interval = 1))
  - Useful for estimating unknown parameters of distributions
  - Update probabilities based on new evidence or data.
  - P(A|B) = P(B|A) x P(A) / P(B)
- Parametric vs Non-parametric Models?
  - Models are a compact distribution of dataset
  - Parametric (w/ parameters): y = mx + c
  - Non-Parametric (w/o parameters): y = mean(x1+x2…)

3. Linear Regression (Least squares, Invertibility, Quadratic Cost)

- What is the **method of least squares**?
  - Finding the line of best fit (y = mx+c) that minimizes cost function J(m,c).
  - J(m,c) = sum(y - (mx + c))^2
  - Solve using vector form (pseudo-inverse) or gradient descent
- Invertible (unique solution, determinant is nonzero)?

    - It means that the matrix has linearly independent columns, and therefore, a unique solution exists.
- Why do we **minimize $w_1^2+w_2^2$**?
    - To find the optimal decision boundary and maximize the margin between the two classes.
    - Generalization: the weights that correctly classify training data and generalize to new data
- Can you prove an expression is optimal when the cost is quadratic?
    - Yes. By finding the derivative of the cost function wrt weights and equating to 0.
- What is quadratic cost? When is it appropriate?
    - Loss function to measure the squared difference between actual and predicted
    - Convex and differentiable: has a unique global minimum value to converge to; has a well-defined gradient at every point for gradient-based optimisation.

4. Data Preparation and Regularization

- What is feature selection?
    - Selecting relevant features to improve the accuracy and efficiency, such as the number of parameters.
    - Important for generalization and computational cost.
- What is model selection?
    - Choosing the appropriate learning algorithm given hardware constraints, problem type, data type, non-linearity and such.
- What is data normalization? What is data imputation?
    - Scaling all data to a standard interval (i.e. 0 to 1)
    - A way to deal with missing features by replacing with average.
- What is the **bias-variance dilemma**?
    - A models ability to accurately capture an underlying pattern of training data
    - Too many parameters -> overfitting -> high variance
    - Too little parameters -> underfitting -> high bias
- What is **regularization**?
    - Solution to overfitting by adding a term that constraints the parameter
    - Penalizing the model for redundant features
    - L1 LASSO - small parameter vectors (absolute values, sets constraints to 0)
        - Shrinks the coefficients of less important features to zero (by adding a penalty), good for feature selection.
    - L2 Ridge - sparse parameter vectors (takes the squares of the coefficients)
    - Choice is based on prior knowledge, such as positive or interval related constraints.

5. Artificial Neural Networks

- What is **MLP (Multi-layer perceptron)**? How does it solve nonlinear problems?
    - Used in ANN when there is difficulty to apply GD to learn the parameters
    - Activation functions are continuously differentiable for gradient descent and backpropagation, such as logarithmic and tanh.
    - Multiple layers of neurons

    - Nonlinear functions can be described linearly in high dimensional spaces (i.e. XOR).
        - Minimum architecture of one hidden layer with two neurons
- What is an **RBF Network**?
    - It is a single layer network that uses radial basis function for activation
    - The neurons in the hidden layer respond to inputs based on their distance from a center point
    - Useful for highly non-linear and complex problems, easier to train and less prone to overfitting

6. Artificial Neural Networks

- What is ANN?
    - It is a neural network that consists of interconnected nodes (neurons) and three or more layers (input, output, hidden), in which an input data is passed through each layer and mathematical operations are performed to have a final output.
    - Solves a wide range of tasks, such as classification.
- What is **batch gradient descent**? **Stochastic gradient descent**?
    - Batch updates the model parameters using the average of the gradients of the cost function calculated on the entire data set. (once per epoch, with all training data.)
    - Stochastic updates on a single training sample at a time, making it faster than BGD. Gradient is calculated on a single training sample it can result in a noisy estimate and slower convergence.
- What is **backpropagation**?
    - Method used to iteratively update the hyperparameters, weights and bias, by propagating backwards.
    - Essentially finding the gradient of the loss respect to weights/bias and multiplying it by a defined learning rate, to adjust the parameters

7. Convolutional Neural Networks

- What is **CNN**?
    - It is a type of ANN that is used for computer vision applications such as images.
    - The input is passed through a convolutional layer, which applies filters to extract features, a pooling layer, which reduces dimensionality for defined focus and reduces complexity/overfitting, it is then flattened to a 1D tensor to be passed through layers for classification or regression.
- CNN vs LNN?
    - Linear neural networks can lose special features such as diagonals.
- What are **kernels**?
    - A small matrix used for the convolution operation, set of weights applied to a region of an image to extract features.
- What is a **receptive field**?
    - Portion of the input image that a particular neuron would respond to.

    - Determined by the size of the kernel in the previous layer and stride of the convolution operation
- What is **parameter sharing**?
    - Technique in CNN to reduce the number of parameters in the network to improve its efficiency.
    - Same set of weights is used for all neurons in a particular layer, reducing the number of parameters in the network and preventing overfitting, as the network learns a small number of feature detectors that are generalizable to different inputs.
- How do we build complex features from simpler ones?
    - Hierarchical feature learning
    - Build a hierarchy of layers, with each layer learning progressively more abstract and complex features from the previous layers
    - Capture high level concepts and relationships in the data

8. Support Vector Machine

- What is **SVM**?
    - Supervised learning algorithm for classification and regression analysis
    - Find the best hyperplane to separate different classes of data points in a feature space and maximizes the margin
    - The data points closest to the hyperplane are called support vector, and they are calculated to determine the position and orientation of the hyperplane
- Limitations and solutions?
    - Constraints in every single point (computationally expensive)
    - Noisy data
        - **Hinge function** (returns 0 if hyperplane is found, if not function value is proportional to the decision boundary)
    - Dealing with non-linearities
        - Mapping the values in a higher dimension
        - Kernel function, calculates the difference between two data points in the higher dimension.
- Trade off between maximizing margin and considering mis-classified points
    - Optimise the hinge (soft margin)
    - Optimise the original formulation (hard margin)
- Update parameters using sub-gradients
- How is **Lagrangian** used?

9. **Unsupervised Learning (K-means)**

- What is unsupervised learning?
    - Dealing with unlabelled data
    - Problems regarding clustering (finding similar patterns in clusters), anomaly detection (finding outliers), density estimation (estimating PDF)
- What is the **K-means algorithm**?

    - Iterative process where an initial centroid is defined (k<M), values are associated to it, centroid is readjusted to the average, and repeated until the centroid does not change.
- How to optimize the cost and k numbers?
    - Elbow method
    - Cross validation - calculate cost J with test data and repeat with multiple subsets and k-values to find optimal choice
- Disadvantages?
    - Affected by outliers, finding initial points, limited to Euclidean distance metric.

10. Markov Chain

- What is the **Markov chain**?
    - Dynamic system that takes no memory and evolves probabilistically
    - Can be displayed with a state transition graph, that connects different states with probabilities
- What is **stationary distribution**?
    - Once the probability distribution is repeated many times (as in the probability of which state you'd be in after each stage), it converges to an equilibrium where the probability distribution will remain unchanged, and satisfies the condition piP = pi,
    - P is the transition matrix of the Markov chain and pi is the row vector of probabilities.
    - This can be found either by doing a P^infinitely high number or finding the eigenvector and eigenvalue of the matrix.
- What is the **mean first passage time**?
    - Expected time for a process to reach a particulate state, starting from an initial given state.

11/12. Markov Chain

- What is the **markov property**?
    - Memorylessness, we only depend on current memory and the future to graph probability of the next state.
- What is the **Perron-Frobenius theorem**?
    - Refers to non-negative matrices, if it is irreducible (all entries are positive and there is a path between every element) there is a unique largest eigenvalue that is real and positive. (provides info about the rate it converges to stationary distribution)
    - This theorem provides information about the long-term behavior of the chain.
    - For a finite-state markov chain, there exists a unique stationary distribution (probability distribution) that it will converge to regardless of the initial state.
- What are **Perron-eigenvectors**?
    - Eigenvector of the transition matrix with the largest eigenvalue, when it is at stationary distribution.
    - Most congested part of the network.

- What is the **Kemeny constant**?
    - Indicator of network efficiency. This is determined by all the eigenvalues of the graph.
    - Can be used to find and identify critical states, and find communities.
- Second Eigenvector?
    - Reveal hidden subcommunities

13/14. Reinforcement Learning

- What is **MDP (Markov Decision Process)**?
    - Method to model decision processes in which outcomes are partially random and influenced by the decision of an agent. It is commonly used in reinforcement learning
- What is its role in reinforcement learning?
    - The transition between states are probabilistic and based on rewards.
- What is the **Bellman equation**?
    - The Bellman function is a fundamental concept in reinforcement learning and dynamic programming.
    - The Bellman operator is used to compute the V and Q functions, to then evaluate how good a certain policy is.
    - It represents a relationship between the value of a state-action pair and the values of neighboring state-action pairs.
    - It is modelled with a value function to maximize the reward by taking a given action in a given state with probability of transition to a certain state.
    - The Q-function is similar but takes into account the value of the next state-action pair.
- What are **V and Q values**? How are they used?
    - The value function V and quality Q represent the total reward when following a certain policy in a given state.
    - Value function includes the reward obtained at a certain state, as well as a discount factor which is a constant that determines the importance of the future rewards, to denote the expected value.
    - Q represents the expected total reward by taking an action in a given state, and then following a certain policy.
    - They are used to determine the quality of different policies and play a central role in reinforcement learning, as they can optimize the behaviour of agents in a given environment.