**Questions about Neural Networks:**

**Neurons:**

What is a neuron?

A collection of a set of inputs, a set of weights, and an activation function.

What can a single neuron learn?

On its own it’s essentially a linear classifier or linear regressor.

What are activation functions?

Functions that are used to determine whether a neuron should be activated by calculating the weighted sum of inputs and biases.

What are popular activation functions?

Sigmoid, ReLU, Softmax, Linear, Tanh, Step, Leaky ReLU

Are some activations better than others?

Yes, depending on the specific task and network architecture. Computational efficiency and ability to mitigate the vanishing gradient problem are general scales for activations (ReLU does this very well).

How does a neuron with a sigmoid activation relate to logistic regression?

A neuron with a sigmoid activation function is essentially equivalent to a single unit of logistic regression, so when a single neuron in a neural network uses a sigmoid activation function, it performs the same calculation as a logistic regression model, producing a probability-like output between 0 and 1 based on a linear combination of its inputs.

How does a neuron with the sign function as activation relate to a perceptron?

A neuron with a sign function as its activation is essentially the same as a basic perceptron, as the sign function acts like a step function, outputting either a positive value if the weighted sum of inputs is positive, and a negative value if the weighted sum is negative, effectively representing a binary decision just like a perceptron does.

**Neural Networks:**

What is a multi-layer neural network?

What can it learn?

Patterns in data, including complex non-linear relationships

What hyperparameters are there?

The hyperparameters are the activation function, number of hidden layers, neurons per layer, learning rate, regularization (adding a penalty term to the loss function), and batch size. NOT weights and biases, these are parameters that are learned during training and represent the model’s internal state.

**Loss Functions:**

What is a loss function?

A function measuring “how bad” a networks’ output is, usually relative to some gold-standard for what the output should be.

What loss function should I use for regression?

The standard loss function to use is Mean Squared Error (MSE), also known as L2 loss.

What loss function should I use for classification?

Most commonly cross-entropy loss is used for classification tasks.

**Training Neural Networks:**

What is backpropagation?

It is an algorithm to efficiently compute gradients of our loss with respect to the parameters in feed forward neural networks, it does this by making sure we’re building up our gradient of the loss in chain rule by going from the loss backwards and is caching intermediate results.

Why is backpropagation an efficient way to compute gradients?

Because it does not recompute things we already know – it leverages the chain rule of calc to propagate error information backward through the network, thus avoiding redundant calculations and reusing intermediate values.

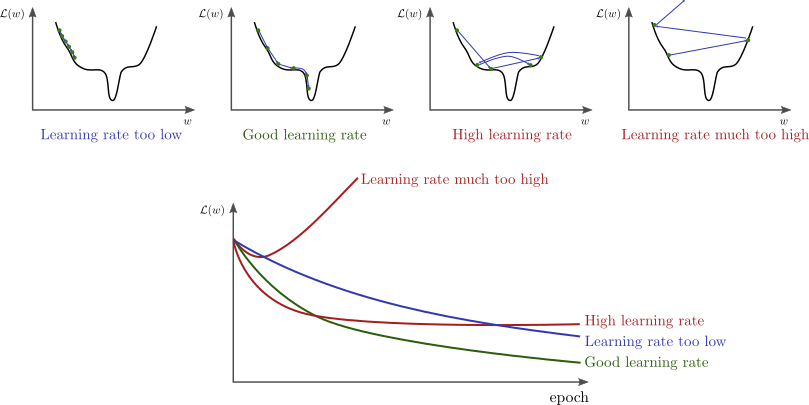
What is stochastic gradient descent?

An optimization algorithm used where instead of calculating the gradient based on the entire dataset, it updates model parameters using only a single data point or a small random subset of data at a time.

How does training behavior change with different settings of learning rate?

A higher learning rate: leads to faster updates in the model’s weights, resulting in quicker learning but potentially causing instability and overshooting the optimal solution

A lower learning rate: leads to slower learning, potentially getting stuck in suboptimal solutions or taking a very long time to converge to a good result



What is a computational graph?

A visual representation of the flow of data and operations within a neural network:

A diagram of a diagram of a layer

Description automatically generated

**Questions about Decision Trees**

**Decision Trees:**

What is a decision tree?

A tree structured model where:

* **internal nodes** perform tests against an individual input feature value,
* Each branch from an internal node represents a potential value or range of values of the tested attribute, and
* Each **leaf node** predicts an output (either class or continuous value).
  + Prediction is “blah” where “blah” can be either majority class of training data that arrived at leaf (classification), or average value of training data that arrived at that leaf (for regression).

What can decision trees learn?

Can represent arbitrarily complex functions – universal approximator. Basically can identify patterns in data and make predictions about future outcomes by creating a series of rules based on key features.

How are they used to make predictions for classification and regression?

* Prediction is “blah” where “blah” can be either majority class of training data that arrived at leaf (classification), or average value of training data that arrived at that leaf (for regression).

Why are decision boundaries for decision trees composed of axis-aligned segments?

At each split, a decision tree chooses a single feature and a threshold value to divide the data, this effectively makes a boundary that is perpendicular to the axis corresponding to that chosen feature.

What is the maximum depth a decision tree could reach?

With N training samples, the max depth would be N-1 (imagine the least effective split would be peeling off one training example per node).

When would a leaf of a fully-expanded decision tree have more than one datapoint in  
the leaf? AKA: What are the base cases for the learning algorithm?

All same features or all same label.

**Training Decision Trees:**

How are decision trees trained?

Find a decision tree that achieves minimum error on training data. Trivially achievable with large enough tree

Does the training algorithm you describe guaranteed to find the optimal tree?

Most popular methods build the tree greedily, this means we can’t guarantee finding the smallest (optimal).

What is Entropy?

Measures how random a probability distribution is, with the most random being like a completely uniform distribution. A measure of impurity in the data, where higher entropy indicates more mixed classes and lower entropy indicates purer data.

What is Conditional Entropy?

Conditional Entropy of a Random Variable Y Given X: Entropy of Y when only considering records where X = x\_j, weighted by probability of X = x\_j.

A math equation with red letters

Description automatically generated with medium confidence

What is Information Gain?

Information gain is calculated by subtracting the weighted average entropy of the child nodes after a split from the entropy of the parent node before the split, with the feature that provides the highest information gain being chosen as the best split.

Why do we rank splits by Information Gain?

Information gain is a metric used to determine the best feature to split a node on, essentially it is measuring how much a particular feature improves the purity of the data by separating it into distinct classes, thus helping to build the decision tree structure.

How can decision trees handle continuous variables?

Only considering splits in between your data points.

**Overfitting:**

How can you avoid overfitting in decision trees?

* **Option 1:** Add more hyperparameters to control tree size:
  + Limit depth / limit number of nodes / only split if at least K datapoints present
* **Option 2:** Early stopping based on validation performance
  + Monitor the validation accuracy and stop splitting a branch when performance saturates
* **Option 3:** Post Pruning

What is post-pruning?

* Grow full tree on the training set, then consider the impact of removing each node on validation performance.
* Greedily prune the node that most improves validation set performance

Can you perform post-pruning?

probs

**Questions about Bias and Variance:**

What is bias?

Error due to assumptions in the model not matching the problem (aka modelling error). More formally, average error between model and the true function over all possible datasets.

What is variance?

Error due to sensitivity to changes in the dataset (aka estimation + optimization). More formally, variance of error between model and the true function over all possible datasets .

Why do “weak” models tend to have high bias but low variance?

They are very simple in design, meaning they struggle to capture complex patterns in the data (high bias), but as a result of their simplicity, they are less susceptible to fluctuations caused by noise in the training data (low variance).

Why do “strong” models tend to have low bias but high variance?

These models with low bias learn the training data well (able to capture complex patterns in the data), but they become too sensitive to noise in the data, leading to overfitting and poor generalization (high variance).

How do bias and variance relate to sources of error we discussed earlier in the course (optimization / estimation / modelling / Bayes)?

Bayes error? Not at all.

Modeling error? Bias

Optimization and estimation? Variance

A comparison of the same type of data

Description automatically generated with medium confidence

**Questions about Ensemble Methods:**

**Bagging:**

When is it useful to apply bagging?

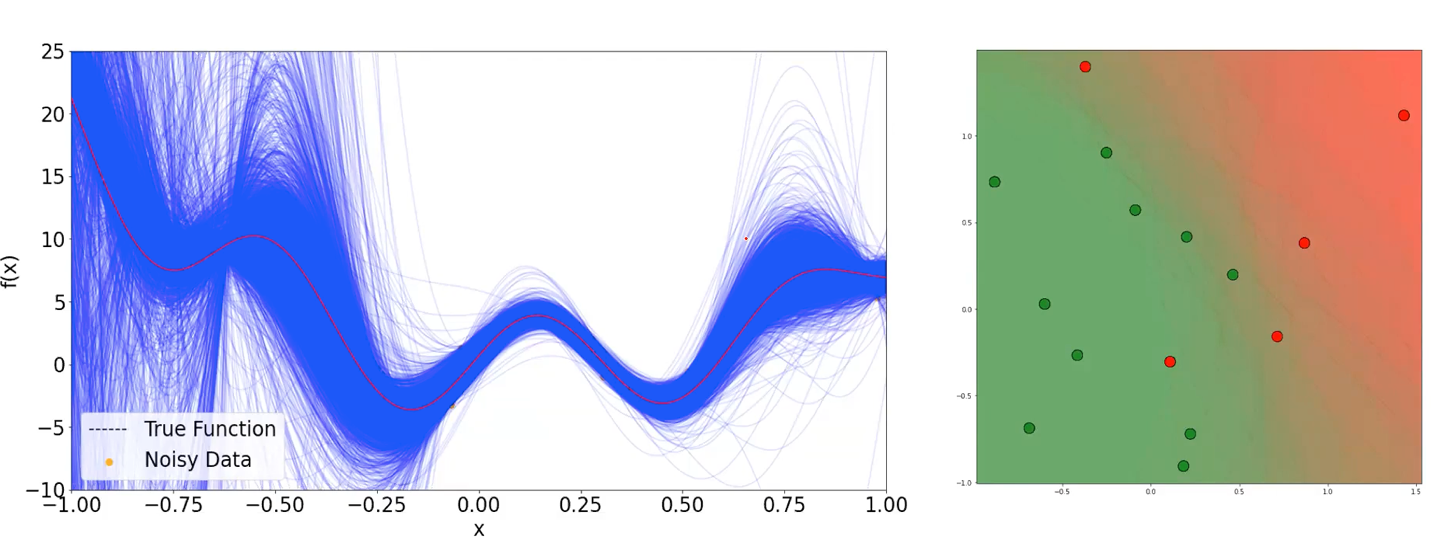
High variance models, like decision trees, when small changes in training data can lead to large changes in the predicted output.

Does bagging help reduce bias or variance?

Bagging is trying to reduce variance in strong learners. To where each member in the ensemble is not correlated with each other.

How does bagging work?

Average multiple models trained from resamples of the dataset:



How do ensembles make predictions?

Consider an ensemble of M models that takes the majority vote of its members as the final output. Final output of the “ensemble” of models is a weighted combination of each model’s output.

How does the correlation between model outputs effect the performance of a bagged ensemble?

A low correlation between individual model outputs is crucial for optimal performance in a bagged ensemble. It allows the ensemble to benefit from the diverse perspectives of each model, effectively reducing variance an improving the overall prediction accuracy.

What are Random Forests?

An example of bagging! Applying bagging to decision trees. We train M trees (an ensemble of decision trees) on different resamplings of the data and call it a forest. During tree learning, at each split consider only a random subset of attributes / thresholds when choosing what to split on (often √d features are considered at each split for d-dimensional inputs). We make predictions by majority vote over the trees.

Why do we want to introduce additional randomness in Random Forests?  
Introducing additional randomness through feature selection helps to create a diverse ensemble of decision trees, which leads to better generalization and reduces the risk of overfitting by preventing any single feature from dominating the prediction process, ultimately resulting in a more robust model with improved accuracy on unseen data

**Boosting:**

When is it useful to apply boosting?

Linear models and really short decision trees. As well as applications like image recognition and natural language processing.

Does boosting help reduce bias or variance?

Boosting is trying to reduce bias in weak learners.

How does boosting work in general?

Training sequence of models where each model tries to improve the errors of the ones that came before it.

How does L2 boosting work?

A group of math equations

Description automatically generated

**Questions about KMeans and GMM:**

**K-means:**

What is clustering?

In general, clustering can be viewed as an exploratory procedure for finding interesting subgroups in a dataset.

What is k-means?

Partition Algorithm (“Flat” Clustering) that groups data points into a specified number of clusters (k) based on their similarity. A partitional clustering algorithm that assigns each point to one cluster by iteratively updated centroids.

How is k-means implemented?

Initialize k centroids randomly, while not converged (if assignments don’t change, or if we’ve hit our max number of iterations), associate each point with its nearest centroid, update each centroid as the average of associated points, then return centroids and associations.

What does k-means optimize?

The process of finding the best set of centroids that minimizes the sum of squared distances between each data point and each centroid.

How is coordinate descent related to the k-means algorithm?

K-Means is coordinate descent to minimize sum-of-squared error between centroids and datapoints.

Is the k-means algorithm guaranteed to converge? If so, to local or global optima?

Yes, the k-means algorithm is guaranteed to converge on a LOCAL optima.

How do we pick hyperparameters for k-means?

The number of clusters (and centroids), the initial centroid positions, maximum iterations/percentage of assignments changed, For k, we try to pick the elbow/knee of the SSE vs. k graph.

Is k-means sensitive to outliers?

Yes, it is highly sensitive to outliers. Why? This why:

A screenshot of a computer

Description automatically generated

Is k-means sensitive to initialization?

Yes, it is highly sensitive to initialization. We can try to minimize the effects of this through a few methods:

* Run multiple trials and choose the one with the lowest SSE (often used in practice)
* Try to initialize the centroids “well” – not clear how to do this but here’s some common heuristics:
  + Random Points: initialize the centroids by copying random datapoints – ensures your centroids are near data
  + Far-Away Points: try to make the cluster centers far from each other:
    - Samples a datapoint and set the first centroid to its value
    - Compute a weight for each datapoint proportional to its distance from the first centroid. Then sample according to this distribution.
    - Repeat with weights proportional to the nearest centroid.

What types of clusters does k-means work best for?

Similarly sized spherical clumps of data points/clusters (these hidden assumptions that create this pattern come from Gaussian).

**Gaussian Mixture Models:**

Expectation maximization (EM) Algorithm for GMM:

Two steps, similar to K-Means:

1. Computes soft assignments between points to gaussians
2. Uses soft assignments to update position and shape of those gaussians

What sort of model does GMM assume generated the data?

From k independent Gaussians.

What can GMM do that k-means can’t?

Learn variance shape.

It can model more complex configurations than k-Means but is slightly more costly and difficult to implement. It produces a full density model of the data which enables you to sample new synthetic data or evaluate the probability of some new point. And fractional assignments can be turned into hard clusterings by taking the argmax for each point.

Why is maximum marginal likelihood difficult to optimize?  
Because one way you can get it to go towards infinity is if one of the Gaussians “collapses” (put one of your gaussians on one datapoint and then make the variance really small so you basically assign infinite density to that one point, so the total sum log probability will be infinite)

What is the Expectation Maximization algorithm do at a broad level?

Essentially “filling in” missing data by iteratively calculating the expected values of those latent variables based on the observed data, then updating the model parameters to maximize the likelihood of observing the data given those estimated latent variables, repeating until convergence is reached.Common applications of EM include clustering with GGMs where each data point is assigned a probability of belonging to different clusters, and handling missing data in datasets.

What are some challenges in GMM optimization?

It uses the Expectation Maximization (EM) algorithm for optimization which may be slow to converge or end up in trivial optima and may need multiple restarts.

What assumptions must be made in GMM in order to recover the k-means algorithm?

Assume hard-assignment rather than fractional. Assume all Gaussians have the same isotropic covariance (everything has the same variance structure, aka don’t learn shape).

**Questions about Hierarchical Clustering**

**Hierarchical Agglomerative Clustering:**

How does HAC work?

We build our data set from one cluster per point all the way to one cluster for the entire data set by iteratively joining the closest clusters over and over again.

What is a “link function”?

How do the different link functions behave?

How do you measure distance between different clusters?

EXPECT TO SOLVE SINGLE-LINK OR COMPLETE-LINK ON TEST, EXAMPLE DONE IN CLASS.

Consider two clusters A and B, consider the following distance measures l(A, B) defined based on a point-wise distance function d(x,y):

1. Single-link: the distance between the nearest points
   1. l(A, B) = min d(x, y)
   2. A diagram of a red and green circle

      Description automatically generated
2. Complete link: the distance between the furthest points, taking the distance to be the maximum distance between clusters, but still combining the closest points first
   1. l(A, B) = max d(x, y)
   2. A diagram of a number of dots

      Description automatically generated
3. Centroid: the distance between the cluster means
4. Average-link: average distance between all cross-cluster pairs
   1. Most robust and most commonly used

What is a dendrogram?

A way of visualizing Hierarchical Clustering that shows which things merged early and which merged late and what distance they merged at

where the X-axis is all the points and the height of joint is the distance between the two merged clusters.

How to read dendrograms?

The X-axis is all the points and the height of joint is the distance between the two merged clusters. Merge distance monotonically increased as we merge more for single/complete/average linking (not for centroid). You can read it and get an idea of how many clusters your data naturally has.

How to create “flat” clusterings from HAC?  
Select a cut-off point on the dendrogram generated by HAC, where the desired number of clusters is formed by cutting the branches at that level. This essentially separates the clusters at that point, giving you a flat clustering with the specified number of clusters.

**Evaluating Clustering:**

What can we measure without knowing labels?

* User inspection – aka just look at it: Does a cluster seem to have a common theme?
  + CAUTION: HUMANS ARE GOOD AT IMAGINING PATTERNS
* Internal Criterion: measure properties of a clustering presumed to be “good”
  + High within-cluster similarity:
  + Low between-cluster similarityA white background with black dots

    Description automatically generated
  + This measure depends on the dataset and measure of distance used.

What can we measure if we do know labels?

* Then we can evaluate the clustering by how well it separated points from different labels:
* Rand Index – Given a clustering P and a ground truth label set G, measure the number of vector pairs that are:
  + a: in the same group in both P and G (same cluster, same labels)
  + b: in the same group in P but different in G (same cluster, different labels)
  + c: in different groups in P but same in G (different cluster, same labels)
  + d: in different groups in both P and G (different clusters, different labels)
* How is the Rand Index computed?
  + A black math equation

    Description automatically generated with medium confidence
* Adjusted Rand Index: correct rand-index by the average and rand-index of a random clustering of the data.

**Questions about Dimensionality Reduction:**

What is PCA?

Principal Component Analysis conceptually is where we find a line such that most of the variance of the data follows along that line, or equivalently, a line where the dataset projected onto that line maintains as much variance. Call this line your first principal component, then repeat this by finding the line orthogonal from this that captures the most variance. In 2D, there is only one line orthogonal to our first principal component so we must stop after two, but in higher dimensions there will be many and we will want to compute them.

What sort of subspaces can it find?

PCA can exclusively find linear subspaces.

How does PCA choose directions for the lower dimension representation?

The line that is orthogonal to the previous dimension’s principal component (line).

How can PCA be solved?

To perform PCA on your data matrix X (n x 100):

1. Compute the mean vector of your data (100 dim vector) and subtract it from X to center your data
   1. X = X – np.mean(X, axis=0)
2. Compute the covariance matrix by computing 1/n \* X^T \* X
   1. cov = X.T @ X / n
3. Call a package to compute the Eigenvalues/Eigenvectors of the covariance
   1. Eigvals, eigvecs = np.linalg.eig(cov)
4. Sort both in descending order by the Eigenvalues and return Eigenvectors corresponding to the top k. Usually as a (d x k) matrix W with each column being one Eigenvector.
   1. sorted\_indices = np.argsort(eigvals)[::-1]
   2. eigvals\_sorted = eigvals[sorted\_indices]
   3. eigvecs\_sorted = eigvecs[:, sorted\_indices]
   4. W = eigvecs\_sorted[:, :k]

To project your data to that lower dimension, simply compute the matrix product XW

How can we see what fraction of variance a dimension of PCA’s output captures?

The scalar value associated with each of the eigenvectors corresponds to the fraction of variance, so sort them and pick the one that explains the most variance.

How to project a point to the PCA lower dimension space?

Compute the matrix product XW

What sort of relationships can PCA not identify?

Non-linear relationships.

Why might applying PCA as a preprocessing step lead to issue?  
It is sensitive to the scale of your data, can be heavily impacted by outliers, and can sometimes result in less interpretable features due to the linear combination of original variables, potentially causing problems with understanding the underlying patterns in your data if not used carefully.

What is t-SNE?

Nonlinear way to do dimensionality reduction - it is directly optimizing some set of low dimensional vectors such that the relationships between points in the high dimension and the relationship with their corresponding low dimensional points are about the same in terms of neighbors – the high and low dimensional neighbors for each point are roughly the same.

What can t-SNE do that PCA can’t?

Capture complex, non-linear relationships between data points.

How does the perplexity parameter affect the output?

It determines how many neighbors a data point considers when being embedded in lower dimensions.A screenshot of a graph

Description automatically generated

What attributes of a t-SNE plot are interpretable?

The clustering of data points, where closely positioned points represent data samples with high similarity in the original high-dimensional space, while points far apart indicate greater dissimilarities; however, the exact distances between clusters on the plot are generally not meaningful and should not be overinterpreted as they can be and often are distorted. Cluster size doesn’t matter, distance between clusters might not be meaningful, shapes may be distorted, and topology is not always preserved.

Maximizing the variance with respect to the line we project going to it has critical points anywhere this is true: Covariance matrix of the data \* W = some scalar multiplier (lambda) \* W (an eigenvector decomposition), so we can find Ws and lambdas. Solving this equation will give us vector scalar pairs, and so we’re keeping the vectors corresponding to the highest scalars.

EXPECT:

Questions similar to that of SVM on midterm “here’s some data, what would happen? Draw me something, tell me about which of these would have a better explanation of variance?”