Neural Networks and Deep Learning
Statistics

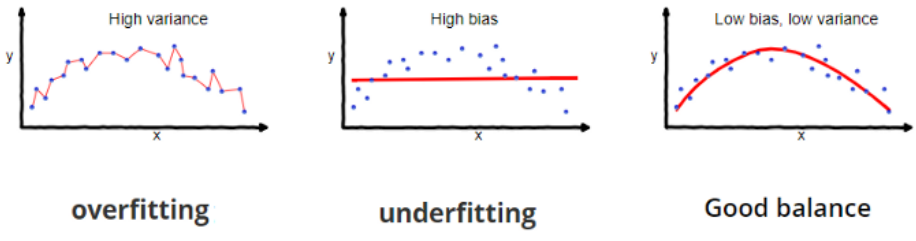
Machine Learning

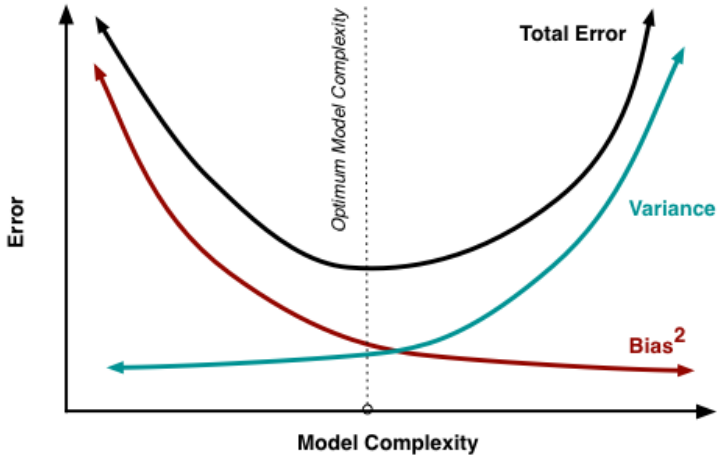
▼ What is the bias-variance tradeoff?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Models with high bias pay very little attention to the training data and oversimplifies the model. This leads to high error on training and test data.

Variance is the variability of model prediction for a given data point from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data leading to overfitting, this leads to poor performance on the data that it has not seen before. As a result, such models perform very well on training data but have high error rates on test data.

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time.





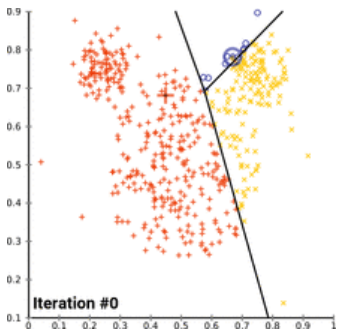
▼ How is KNN different from k-means?

K-Nearest Neighbors is a supervised classification algorithm, while k-means clustering is an unsupervised clustering algorithm. While the mechanisms may seem similar at first: using k points, they are totally different algorithms. In order for K-Nearest Neighbors to work, you need labeled data for the neighbors of the unlabeled point. K-means clustering requires only a set of unlabeled points and a threshold: the algorithm will take unlabeled points and gradually learn how to cluster them into groups by computing the mean of the distance between different points.

The critical difference here is that KNN needs labeled points and is thus supervised learning, while k-means doesn't — and is thus unsupervised learning.

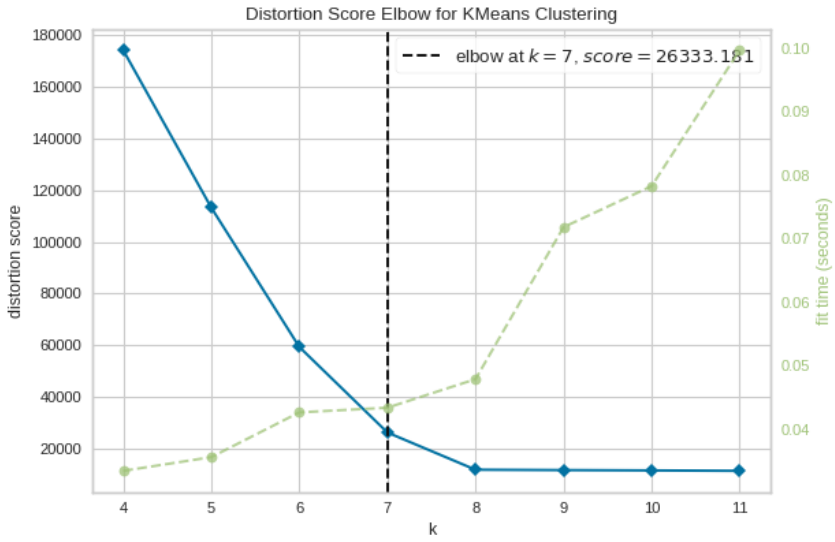
▼ How would you implement the k-means algorithm?

1. Specify number of clusters *K*.
2. Initialise centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
    a. Compute the sum of the squared distance between data points and all centroids.
    b. Assign each data point to the closest cluster (centroid).
    c. Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.



▼ How do you choose the k in k-means clustering?

By using the Elbow Method. In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the distortion, which is the sum of squared distances from each point to its assigned center as a function of the number of clusters. The elbow of the curve is then the number of clusters to use. The same method can be used to choose the number of parameters in other data-driven models, such as the number of principal components to describe a data set.



▼ What are the pros and cons of the k-means algorithm?

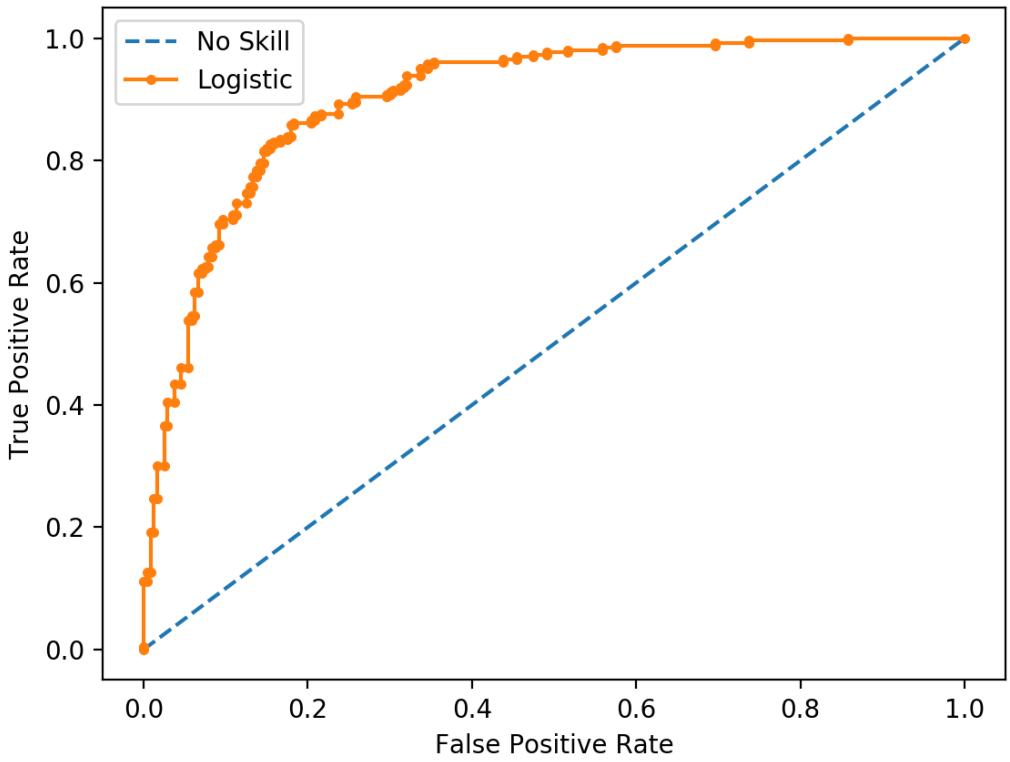
- Pros:
  - Simple to understand
  - Fast to cluster
  - Easy to implement
- Cons:
  - We need to pick number of clusters
  - Sensitive to initialization
  - Sensitive to outliers
  - Spherical solutions
  - Needs standardization

▼ What does an ROC curve show?

A ROC curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR). The true positive rate is the proportion of observations that were correctly predicted to be positive out of all positive observations (TP/(TP + FN)). Similarly, the false positive rate is the proportion of observations that are incorrectly predicted to be positive out of all negative observations (FP/(TN + FP)). For example, in medical testing, the true positive rate is the rate in which people are correctly identified to test positive for the disease in question.

The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

To compare different classifiers, it can be useful to summarise the performance of each classifier into a single measure. One common approach is to calculate the area under the ROC curve, which is abbreviated to AUC.



▼ What is the difference between a type 1 and type 2 error?

A type 1 error is when the null hypothesis is true but is rejected. This is a false positive error, basically asserting something as true when it was actually false. If we take a shepherd and wolf example where the null hypothesis is that there is no wolf present. A type 1 error or false positive would be 'crying wolf' i.e. saying that there is a wolf present when actually there was none.

A type 2 error occurs when the null hypothesis is false, but erroneously fails to be rejected. This is a false negative error when the test indicates that a condition failed, when it was actually successful. In the shepherd and wolf example this would be doing noting (not 'crying wolf') when there is actually a wolf present.

Error Types

Property	Null Hypothesis is true	Null hypothesis is false
Reject null hypothesis	**Type I Error** (False Positive)	True Positive
Fail to reject null hypothesis	True Negative	**Type 2 Error** (False Negative**)**

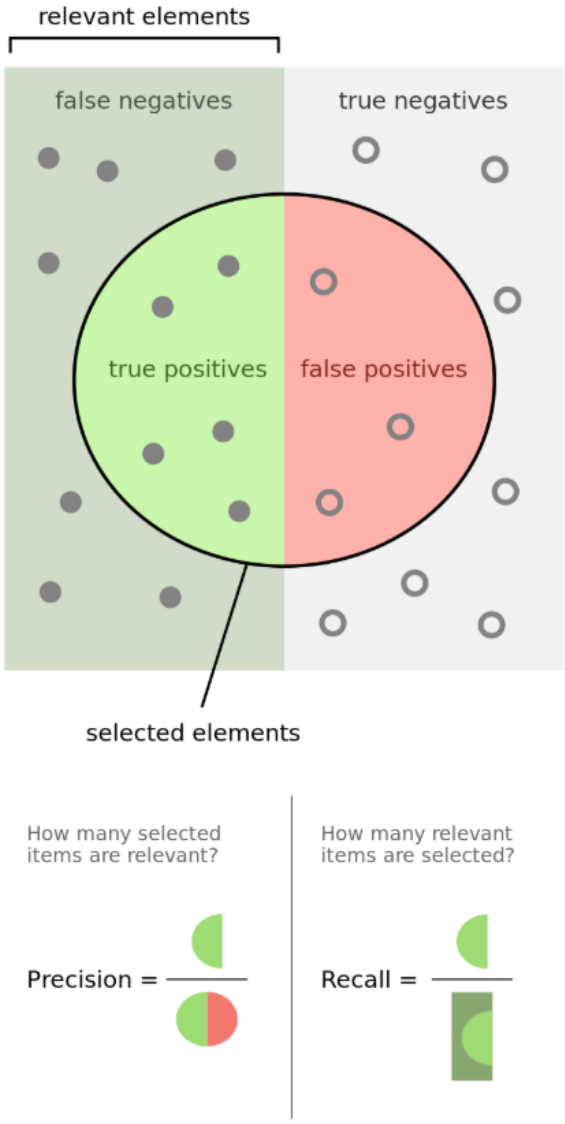
▼ Define precision and recall.

Precision is also known as the positive predictive value, and it is a measure of the amount of accurate positives your model claims compared to the total number of positives it claims. Recall is also known as the true positive rate: the amount of positives your model claims compared to the actual number of positives there are throughout the data.

It can be easier to think of recall and precision in the context of a case where you've predicted that there were 10 apples and 5 oranges in a case of 10 apples. You'd have perfect recall (there are actually 10 apples, and you predicted there would be 10) but 66.7% precision because out of the 15 events you predicted, only 10 (the apples) are correct.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$



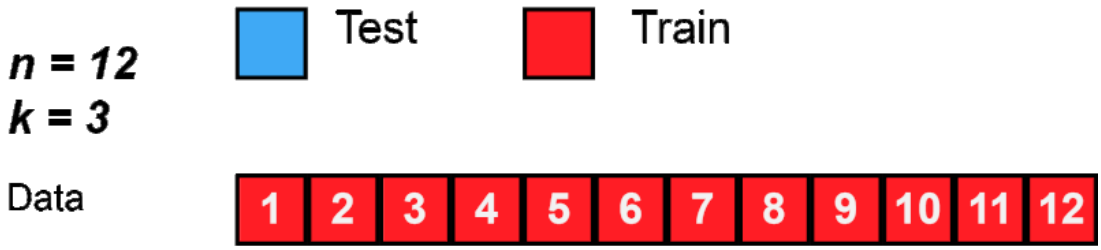
▼ What is k-fold cross validation?

Cross-validation is a statistical method used to estimate the skill of machine learning models.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    a. Take the group as a hold out or test data set
    b. Take the remaining groups as a training data set
    c. Fit a model on the training set and evaluate it on the test set
    d. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores



▼ Explain what a false positive and a false negative are. Provide examples when false positives are more important than false negatives, false negatives are more important than false positives.

A **false positive** is an incorrect identification of the presence of a condition when it's absent.

A **false negative** is an incorrect identification of the absence of a condition when it's actually present.

An example of when **false positives** are more important than false negatives is for spam detection in your email. If you miss a job offer or an email from your boss because it was mistakenly detected as spam there can be large consequences. However, if occasionally you get an email from a Nigerian prince offering money you can quite easily ignore it or mark it as spam.

An example of when **false negatives** are more important than false positives is when screening for cancer. It's much worse to say that someone doesn't have cancer when they do, instead of saying that someone does and later realising that they don't.

▼ When would you use random forests vs. SVM and why?

There are a couple of reasons why a random forest is a better choice of model than a support vector machine:

- Random forests allow you to determine the feature importance. SVM's can't do this.
- Random forests are much quicker and simpler to build than an SVM.
- For multi-class classification problems, SVMs require a one-vs-rest method, which is less scalable and more memory intensive.

▼ Why is dimension reduction important?

Dimensionality reduction is the process of reducing the number of features in a dataset. This is important mainly in the case when you want to reduce variance in your model (overfitting).

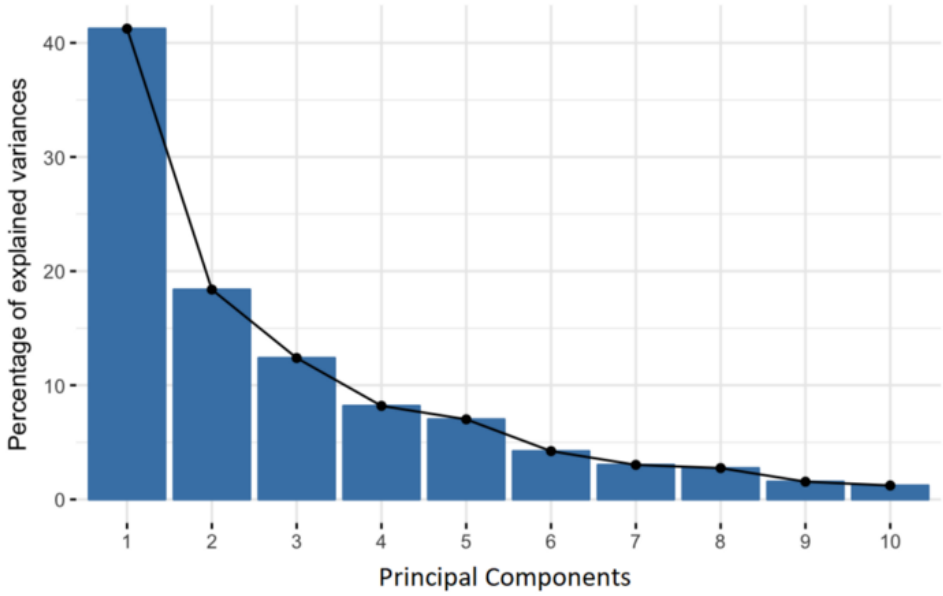
1. It reduces the time and storage space required
2. Removal of multi-collinearity (when two or more explanatory variables are related) improves the interpretation of the parameters of the machine learning model
3. It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D
4. It avoids the curse of dimensionality

▼ What is principal component analysis? Explain the sort of problems you would use PCA for.

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualise and make analysing data much easier and faster for machine learning algorithms without extraneous variables to process.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.



So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

PCA is commonly used for compression purposes, to reduce required memory and to speed up the algorithm, as well as for visualisation purposes, making it easier to summarise data.

▼ What are the assumptions required for linear regression? What if some of these assumptions are violated?

The assumptions are as follows:

1. The sample data used to fit the model is **representative of the population**
2. The relationship between X and the mean of Y is **linear**
3. The variance of the residual is the same for any value of X **(homoscedasticity)**
4. Observations are **independent** of each other
5. For any value of X, Y is **normally distributed**.

Extreme violations of these assumptions will make the results redundant. Small violations of these assumptions will result in a greater bias or variance of the estimate.

▼ What are some of the steps for data wrangling and data cleaning before applying machine learning algorithms?

There are many steps that can be taken when data wrangling and data cleaning. Some of the most common steps are listed below:

- **Data profiling:** Almost everyone starts off by getting an understanding of their dataset. More specifically, you can look at the shape of the dataset with and get a description of the numerical variables.
- **Data visualizations:** Sometimes, it's useful to visualize your data with histograms, boxplots, and scatterplots to better understand the relationships between variables and also to identify potential outliers
- **Standardization or normalization**: Depending on the dataset your working with and the machine learning method you decide to use, it may be useful to standardize or normalize your data so that different scales of different variables don't negatively impact the performance of your model.
- **Handling null values:** There are a number of ways to handle null values including deleting rows with null values altogether, replacing null values with the mean/median/mode, replacing null values with a new category (eg. unknown), predicting the values, or using machine learning models that can deal with null values.
- **Other things include:** removing irrelevant data, removing duplicates, and type conversion.

▼ What is multicollinearity and how do we deal with it?

Multicollinearity exists when an independent variable is highly correlated with another independent variable in a multiple regression equation. This can be problematic because it undermines the statistical significance of an independent variable.

You could use the Variance Inflation Factors (VIF) to determine if there is any multicollinearity between independent variables — a standard benchmark is that if the VIF is greater than 5 then multicollinearity exists. We can then remove one of the variables with multicollinearity.

▼ You are given a dataset on cancer detection. You have built a classification model and achieved an accuracy of 98%. Is this model ready to be used in production?

No, cancer detection results in imbalanced data. In an imbalanced dataset, accuracy should not be based as a measure of performance. It can be easy to achieve high accuracy on unbalanced data with a stupid model, such as just say everyone does not have cancer. If there are 96 images without cancer and 4 with cancer, this model would achieve 96% accuracy.

Hence, to evaluate model performance, we should use **Sensitivity (True Positive Rate), Specificity (True Negative Rate) and F measure** to determine the class wise performance of the classifier.

▼ How would you evaluate an algorithm on unbalanced data?

There are multiple ways that we can evaluate an algorithm on unbalanced data.

1. **Use the right evaluation metrics**. For unbalanced data sets, such as cancer detection where there are a lot of non-cancerous images and few cancerous images, good accuracy can be achieved by labelling all the data as non-cancerous. However by using other metrics, such as:
    - Precision: TP/TP+FP, How much were correctly classified as positive out of all positives?
    - Specificity: TN/FP+TN, Specificity of a classifier is the ratio between how much were correctly classified as negative to how much was actually negative.
    - Recall/Sensitivity: TP / FN+TP, Sensitivity of a classifier is the ratio between how much were correctly identified as positive to how much were actually positive.
    - F1 score: harmonic mean of precision and recall.
    - MCC: Matthews correlation coefficient between the observed and predicted binary classifications.
    - AUC: relation between true-positive rate and false positive rate.
2. **Resample the training set**. An alternative approach is to make a balanced data set out of an unbalanced data set by under-sampling and over-sampling
    a. **Under-sampling:** This balances the data set by reducing the size of the abundant class. This method is used when the quantity of data is sufficient. By keeping all the samples in the rare class and randomly selecting an equal number of samples in the abundant class a new balanced data set is created.
    b. **Over-sampling:** When the quantity of data is insuffient over-sampling is used to increase the amount of rare samples. The new rare samples are generated by repetition, bootstrapping or SMOTE (Synthtic Minority Over-Sampling Technique)
3. Use a cost function in your model that penalizes wrong classification of the rare cases more that wrong classes of the abundant cases, it is possible to design models that naturally generalize in favour of the rare class. For example, we can tweak an SVM model to penalize wrong classifications of the rare class by the same ratio that the class is underrepresented.

▼ You have the 95th percentile of web server response times generated every 2 seconds for the last year. You want to aggregate the percentiles for the last day, week and month. Is there a way that you can aggregate the percentiles?

No, A simple way to demonstrate why any attempt at aggregating percentiles by averaging them (weighted or not) is useless, try it with a simple to reason about percentile: the 100%'ile (the max). E.g. If I had the following 100%'iles reported for each one minute interval, each with the same overall event count: [1, 0, 3, 1, 601, 4, 2, 8, 0, 3, 3, 1, 1, 0, 2]

The average of this sequence is 42. And it has as much relation to the overall 100%'ile as the phase of the moon does. No amount of fancy averaging (weighted or not) will produce a correct answer for "what is the 100%'ile of the overall 15 minute period?". There is only one correct answer: 601 was the 100%'ile seen during the 15 minutes period.

▼ What is Bayes' Theorem? How is it useful in a machine learning context?

Bayes' Theorem gives you the probability of an event given what is known as prior knowledge.

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\neg A)\Pr(\neg A)}$$

For example, if the risk of developing health problems is known to increase with age, Bayes' theorem allows the risk to an individual of a known age to be assessed more accurately (by conditioning it on his age) than simply assuming that the individual is typical of the population as a whole.

Even if 100% of patients with pancreatic cancer have a certain symptom, when someone has the same symptom, it does not mean that this person has a 100% chance of getting pancreatic cancer. Assume the incidence rate of pancreatic cancer is 1/100,000, while 1/10,000 healthy individuals have the same symptoms worldwide, the probability of having pancreatic cancer given the symptoms is only 9.1%, and the other 90.9% could be "false positives" (that is, falsely say you have cancer).

Based on incidence rate, the following table presents the corresponding numbers per 100,000 people.

```
Symptom\Cancer    Yes    No     Total
Yes               1      10      11
No                0      99989   99989
Total             1      99999   100000
```

Which can then be used to calculate the probability of having cancer when you have the symptoms:

$$\begin{aligned} P(\text{Cancer}|\text{Symptoms}) &= \frac{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer})}{P(\text{Symptoms})} \\ &= \frac{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer})}{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer}) + P(\text{Symptoms}|\text{Non-Cancer})P(\text{Non-Cancer})} \\ &= \frac{1 \times 0.00001}{1 \times 0.00001 + (10/99999) \times 0.99999} = \frac{1}{11} \approx 9.1\% \end{aligned}$$

▼ What is 'Naive' in a Naive Bayes?

Naive Bayes is 'naive' because it makes the assumption that features of a measurement are independent of each other. This is naive because it is (almost) never true.

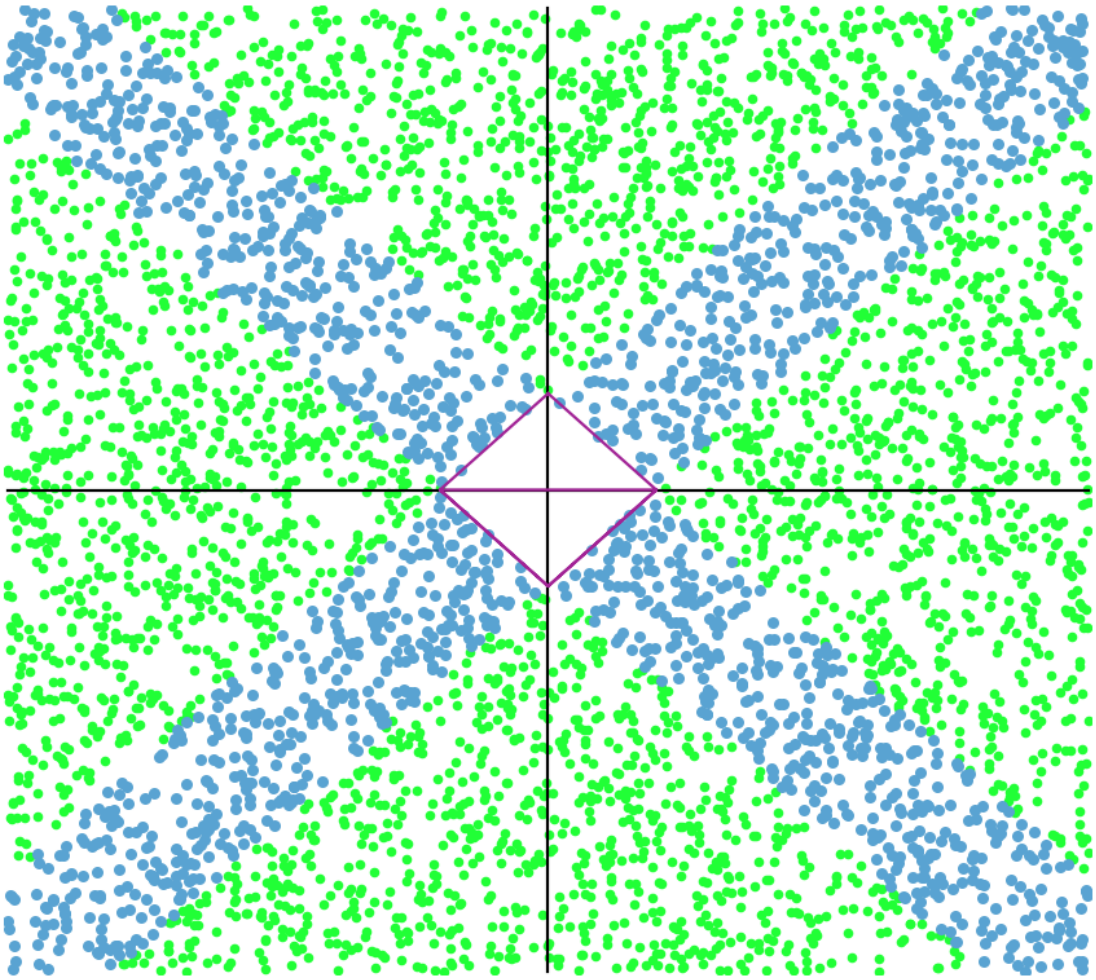
In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

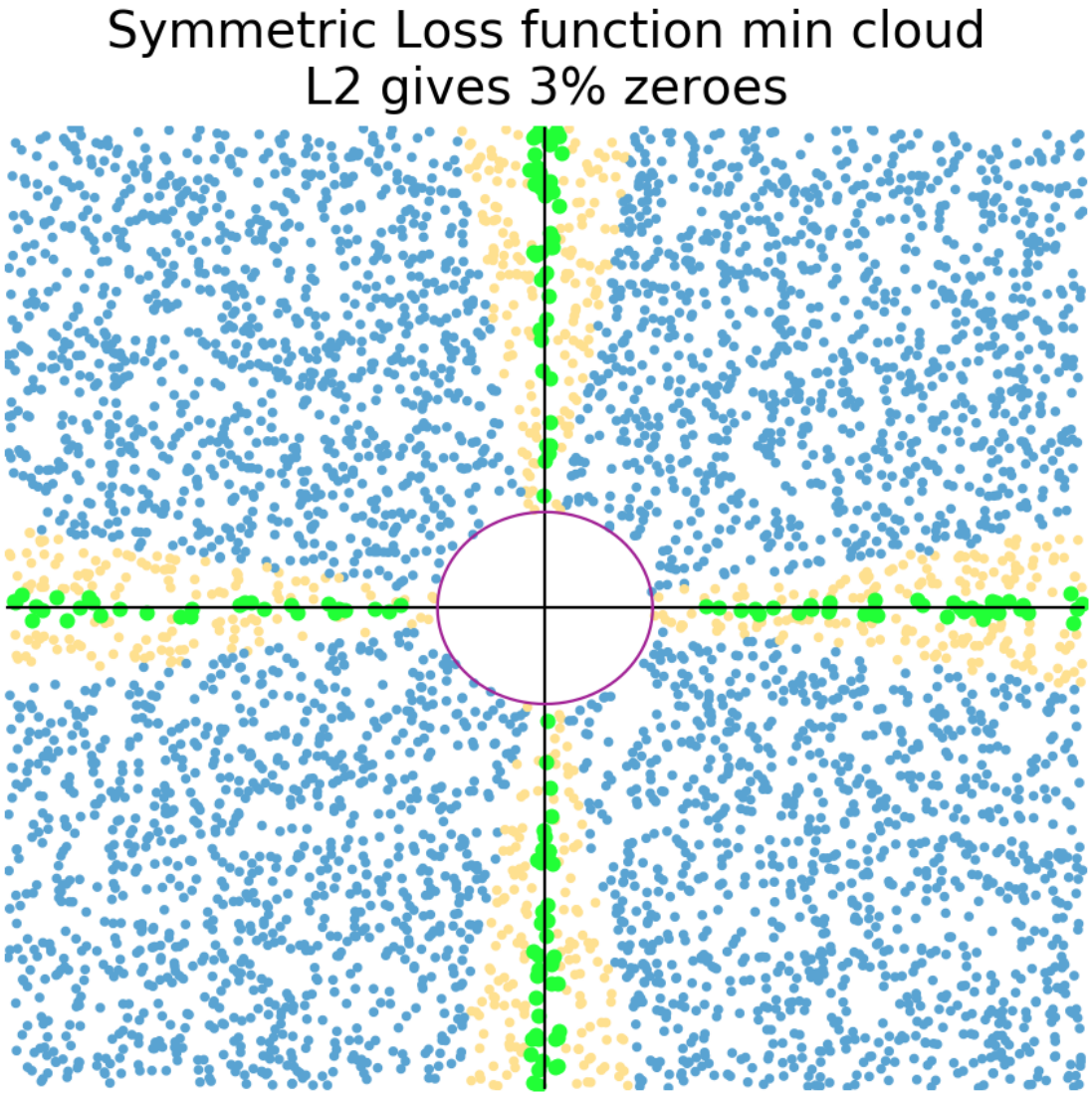
▼ Explain the difference between L1 and L2 regularization.

From a practical standpoint, L1 tends to shrink coefficients to zero whereas L2 tends to shrink coefficients evenly. L1 is therefore useful for feature selection, as we can drop any variables associated with coefficients that go to zero. L2, on the other hand, is useful when you have collinear/codependent features.

Let's do some two-variable simulations of random quadratic loss functions at random locations and see how many end up with a coefficient at zero. There is no guarantee that these random paraboloid loss functions in any way represent real data sets, but it's a way to at least compare L1 and L2 regularization. Let's start out with symmetric loss functions, which look like bowls of various sizes and locations, and compare how many zero coefficients appear for L1 and L2 regularization:

Symmetric Loss function min cloud
L1 gives 66% zeroes



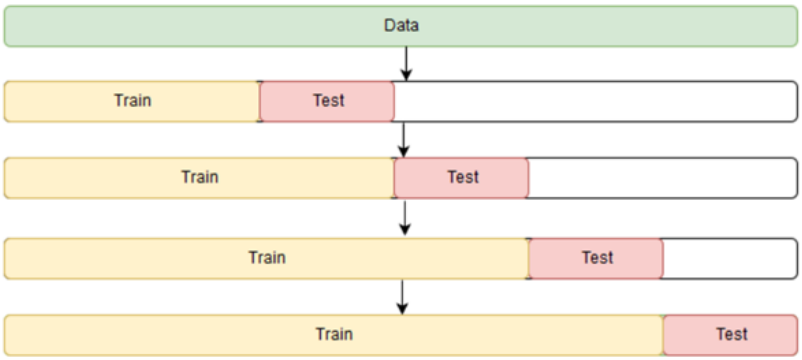


▼ What cross-validation technique would you use on a time series dataset?

Instead of using standard k-folds cross-validation, you have to pay attention to the fact that a time series is not randomly distributed data — it is inherently ordered by chronological order. If a pattern emerges in later time periods for example, your model may still pick up on it even if that effect doesn't hold in earlier years!

You'll want to do something like forward chaining where you'll be able to model on past data then look at forward-facing data.

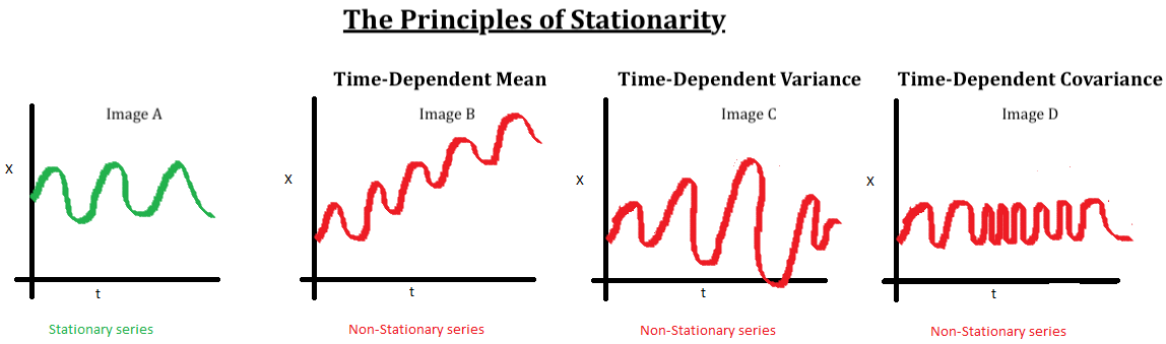
- fold 1 : training [1], test [2]
- fold 2 : training [1 2], test [3]
- fold 3 : training [1 2 3], test [4]
- fold 4 : training [1 2 3 4], test [5]
- fold 5 : training [1 2 3 4 5], test [6]



▼ How can a time-series data be declared as stationery? What statistical test would you use?

It is stationary when the variance, covariance and mean of the series are constant with time.

Here is a visual example:



A time series is stationary if it doesn't have a time-dependent mean, time-dependent variance and time dependent covariance. The first green graph shows a time series that meets all these conditions. In the second image we can see that there is a trend upwards, which indicates a time-dependent mean. In the third image we can see how the variance of the signal changes over time indicating a time-dependent variance. In the final image we can also see how the covariance changes over time indicating a time-dependent covariance.

We can use the Augmented Dickey-Fuller test to check if the time series has a unit root. If the p-value is > 0.05 we fail to reject the null hypothesis, the data has a unit root and is non-stationary. If the p-value is <= 0.05 we reject the null hypothesis, the data does not have a unit root and is stationary. A unit root is a random time series with drift.

▼ What are the main components of an ARIMA time series forecasting model?

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model.

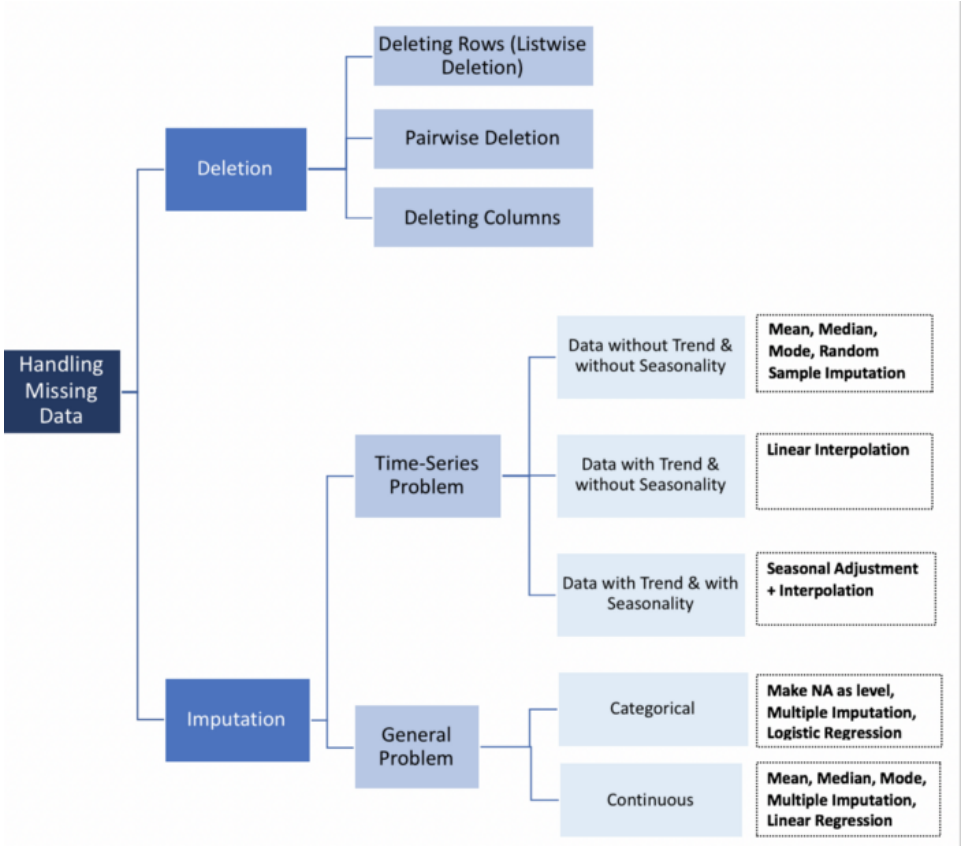
- An ARIMA(0, 0, 0) model is a white noise model.
- An ARIMA(0, 1, 1) model without constant is a basic exponential smoothing model.
- An ARIMA(0, 2, 2) model is equivalent to Holt's linear method with additive errors, or double exponential smoothing

▼ How do you ensure you're not overfitting with a model?

There are three main methods to avoid overfitting:

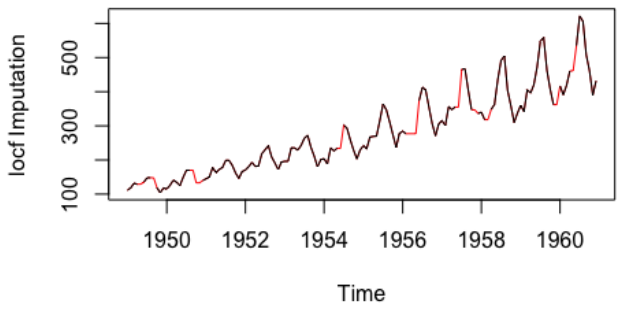
- Keep the model simpler: reduce variance by taking into account fewer variables and parameters, thereby removing some of the noise in the training data.
- Use cross-validation techniques such as k-folds cross-validation.
- Use regularization techniques such as LASSO that penalize certain model parameters if they're likely to cause overfitting.

▼ You are given a data set consisting of variables with a lot of missing values. How will you deal with this?



There are many ways to handle missing data that depend on the size and type of data set:

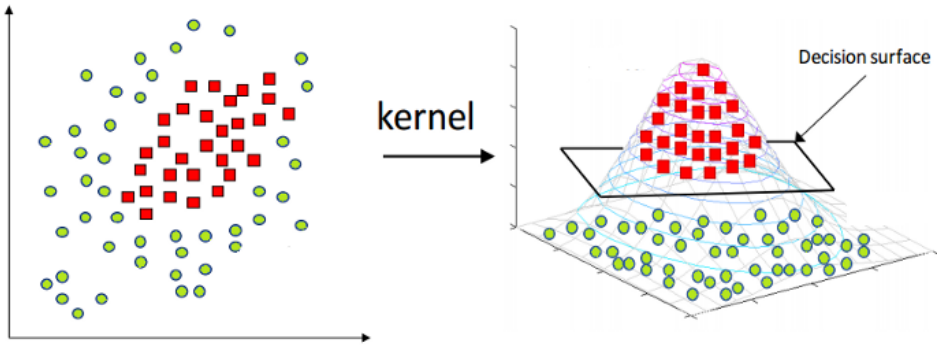
- If the data set is large, we can just simply delete the rows with missing data values. It is the quickest way, we use the rest of the data to predict the values.
  1. Deleting rows that are missing values
  2. Pairwise deletion analyses all cases in which the variables of interest are present and thus maximizes all data available by an analysis basis.
  3. Delete columns that are missing data
- For smaller data sets, we can impute missing values. If the data is time series we interpolate the missing data depending on whether the time series has trend and seasonality. For general continuous data we can use the mean, median, mode, multiple imputation and linear regression to fill in the missing values.



- For general categorical problems we can:
    1. Mode imputation is one method but it will definitely introduce bias
    2. Missing values can be treated as a separate category by itself. We can create another category for the missing values and use them as a different level. This is the simplest method.
    3. Prediction models: Here, we create a predictive model to estimate values that will substitute the missing data. In this case, we divide our data set into two sets: One set with no missing values for the variable (training) and another one with missing values (test). We can use methods like logistic regression and ANOVA for prediction
    4. Multiple Imputation: this is a general approach to the problem of missing data that is available in several commonly used statistical packages. It aims to allow for the uncertainty about the missing data by creating several different plausible imputed data sets and appropriately combining results obtained from each of them.

▼ What's the "kernel trick" and how is it useful?

The Kernel trick involves kernel functions that can enable higher-dimension spaces without explicitly calculating the coordinates of points within that dimension: instead, kernel functions compute the inner products between all pairs of data in a feature space. This allows them the very useful attribute of calculating the coordinates of higher dimensions, while being computationally cheaper than the explicit calculation of said coordinates. Many algorithms can be expressed in terms of inner products. Using the kernel trick enables us effectively run algorithms in a high-dimensional space with lower-dimensional data.



As you can see in the above picture, if we find a way to map the data from 2-dimensional space to 3-dimensional space, we will be able to find a decision surface that clearly divides between different classes. However, when there are more and more dimensions, computations within that space become more and more expensive. This is when the kernel trick comes in. **It allows us to operate in the original feature space without computing the coordinates of the data in a higher dimensional space.**

▼ When would you use gradient descent (GD) over stochastic gradient descent (SDG), and vice-versa?

GD theoretically minimizes the error function better than SGD. However, SGD converges much faster once the dataset becomes large. That means GD is preferable for small datasets while SGD is preferable for larger ones. In practice, however, SGD is used for most applications because it minimizes the error function well enough while being much faster and more memory efficient for large datasets.

▼ Your organization has a website where visitors randomly receive one of two coupons. It is also possible that visitors to the website will not receive a coupon. You have been asked to determine if offering a coupon to website visitors has any impact on their purchase decisions. Which analysis method should you use? 1. One-way ANOVA 2. K-means clustering 3. Association rules 4. Student's t-test

The answer is 1. One-way ANOVA

In statistics, one-way analysis of variance (abbreviated one-way ANOVA) is a technique used to compare means of three or more samples (using the F distribution). This technique can be used only for numerical data. Typically, however, the one-way ANOVA is used to test for differences among at least three groups, since the two-group case can be covered by a t-test (Gosset, 1908).

▼ What is the differentiate between univariate, bivariate and multivariate analysis?

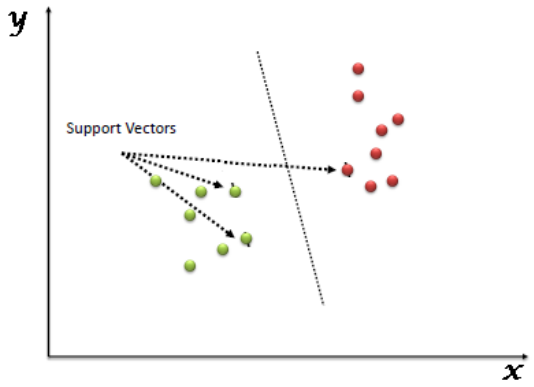
Univariate analyses are descriptive statistical analysis that deal with one variable at a time. For example, the pie charts of sales based on territory involve only one variable and the analysis can be referred to as univariate analysis.

Bivariate analysis attempts to understand the difference between two variables at a time as in a scatterplot. For example, analyzing the volume of sale and spending can be considered an example of bivariate analysis.

Multivariate analysis deals with the study of more than two variables to understand the effect of variables on the response.

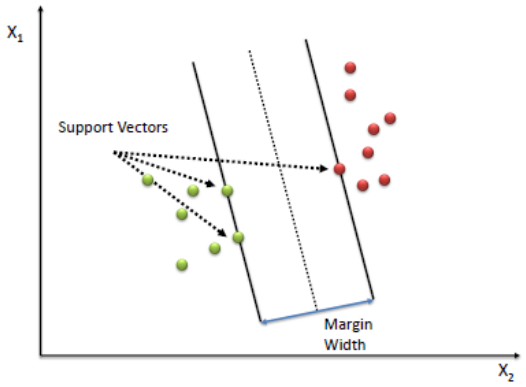
▼ Explain the SVM algorithm.

SVM stands for support vector machine, it is a supervised machine learning algorithm which can be used for both Regression and Classification, however, it is mostly used in classification problems. If you have n features in your training data set, SVM tries to plot it in n-dimensional space with the value of each feature being the value of a particular coordinate. The vectors (cases) that define the hyperplane are the support vectors.



Algorithm

1. Define an optimal hyperplane: maximize margin
2. Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications.
3. Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.



▼ How you formulate SVM for a regression problem statement?

For formulating SVM as a regression problem statement we have to reverse the objective: instead of trying to fit the largest possible street between two classes which we will do for classification problem statements while limiting margin violations, now for SVM Regression, it tries to fit as many instances as possible between the margin while limiting the margin violations.

▼ Are SVM's sensitive to feature scaling?

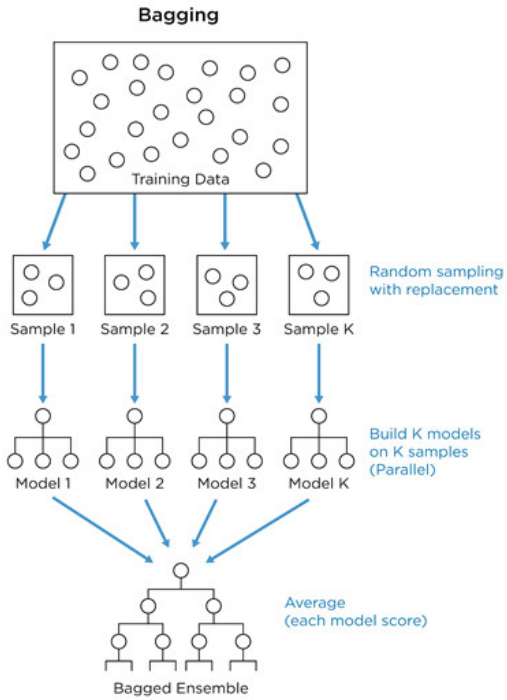
Yes, SVMs are sensitive to feature scaling as it takes input data to find the margins around hyperplanes and gets biased for the variance in high values.

▼ Describe in brief any type of Ensemble Learning.

Ensemble learning has many types but two more popular ensemble learning techniques are mentioned below.

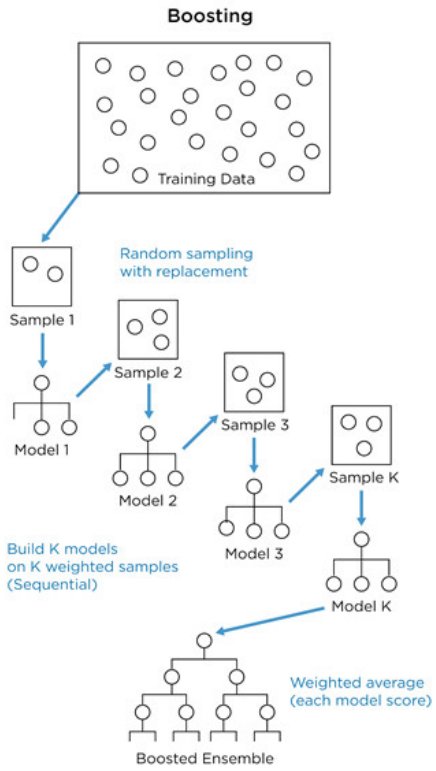
Bagging

Bagging tries to implement similar learners on small sample populations and then takes a mean of all the predictions. In generalised bagging, you can use different learners on different population. As you expect this helps us to reduce the variance error.



Boosting

Boosting is an iterative technique which adjusts the weight of an observation based on the last classification. If an observation was classified incorrectly, it tries to increase the weight of this observation and vice versa. Boosting in general decreases the bias error and builds strong predictive models. However, they may over fit on the training data.

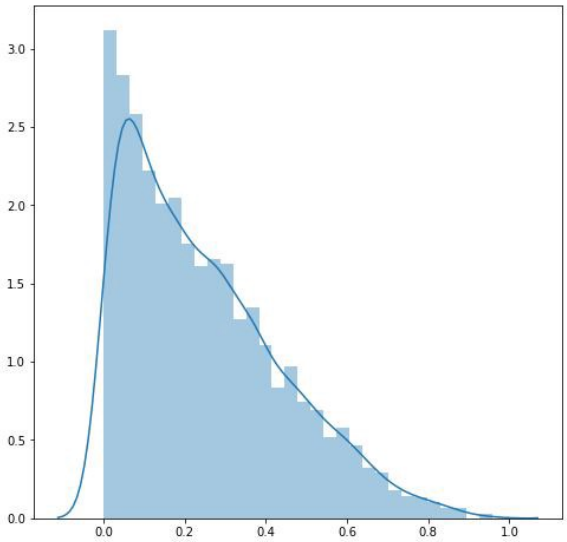


▼ What is a Box-Cox Transformation?

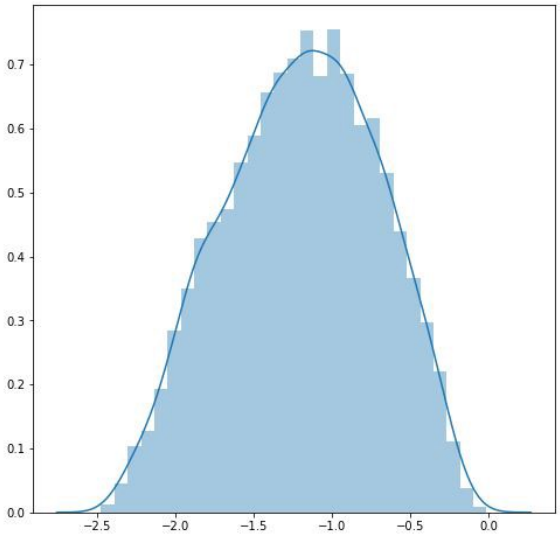
The Box-Cox transformation transforms our data so that it closely resembles a normal distribution. Normality is an important assumption for many statistical techniques, if your data isn't normal, applying a Box-Cox means that you are able to run a broader number of tests.

In many statistical techniques, we assume that the errors are normally distributed. This assumption allows us to construct confidence intervals and conduct hypothesis tests. By transforming your target variable, we can (hopefully) normalize our errors (if they are not already normal). Additionally, transforming our variables can improve the predictive power of our models because transformations can cut away white noise.

Suppose we had a Beta distribution, where alpha equals 1 and beta equals 3. If we plot this distribution, then it might look something like below:



We can use the Box-Cox transformation to transform the above into as close to a normal distribution as the Box-Cox transformation permits.

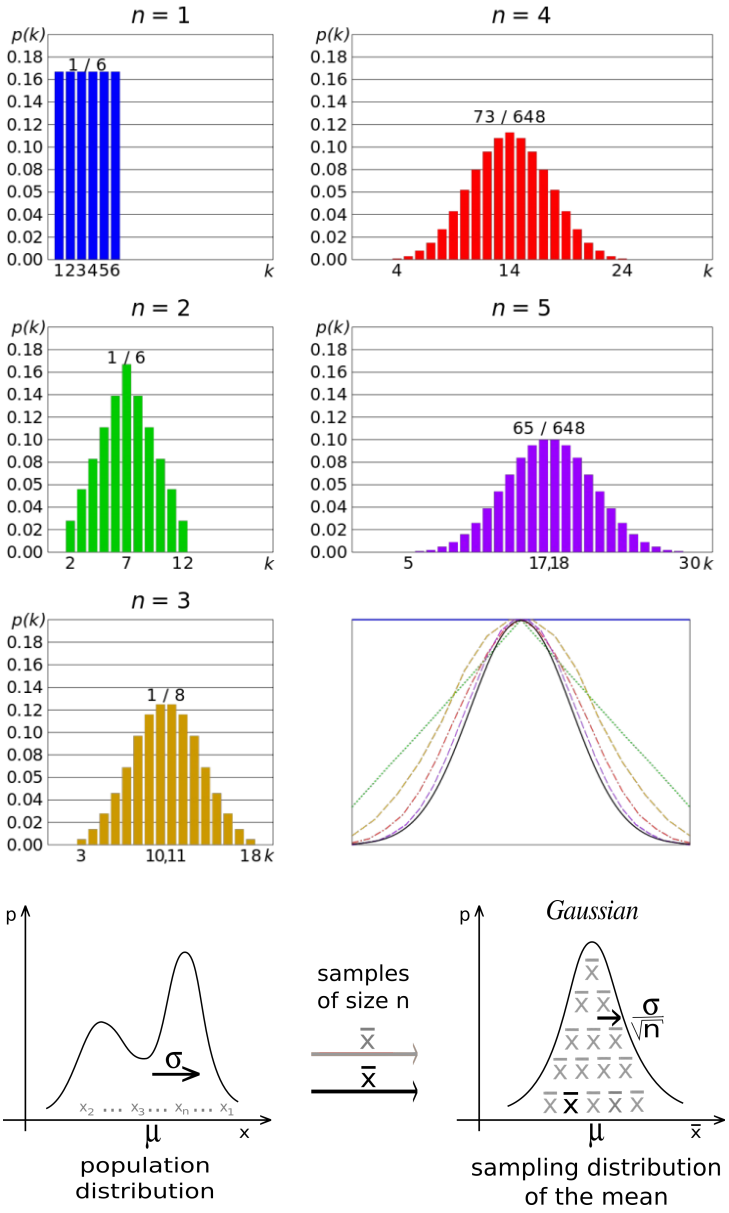


▼ What is the Central Limit Theorem?

The Central Limit Theorem states that the sampling distribution of the sample means approaches a normal distribution as the sample size gets larger — no matter what the shape of the population distribution. This fact holds especially true for sample sizes over 30.

Here’s what the Central Limit Theorem is saying, graphically. The picture below shows one of the simplest types of test: rolling a fair die. The more times you roll the die, the more likely the shape of the distribution of the means tends to look like a normal distribution graph.

The central limit theorem is important because it is used in hypothesis testing and also to calculate confidence intervals.



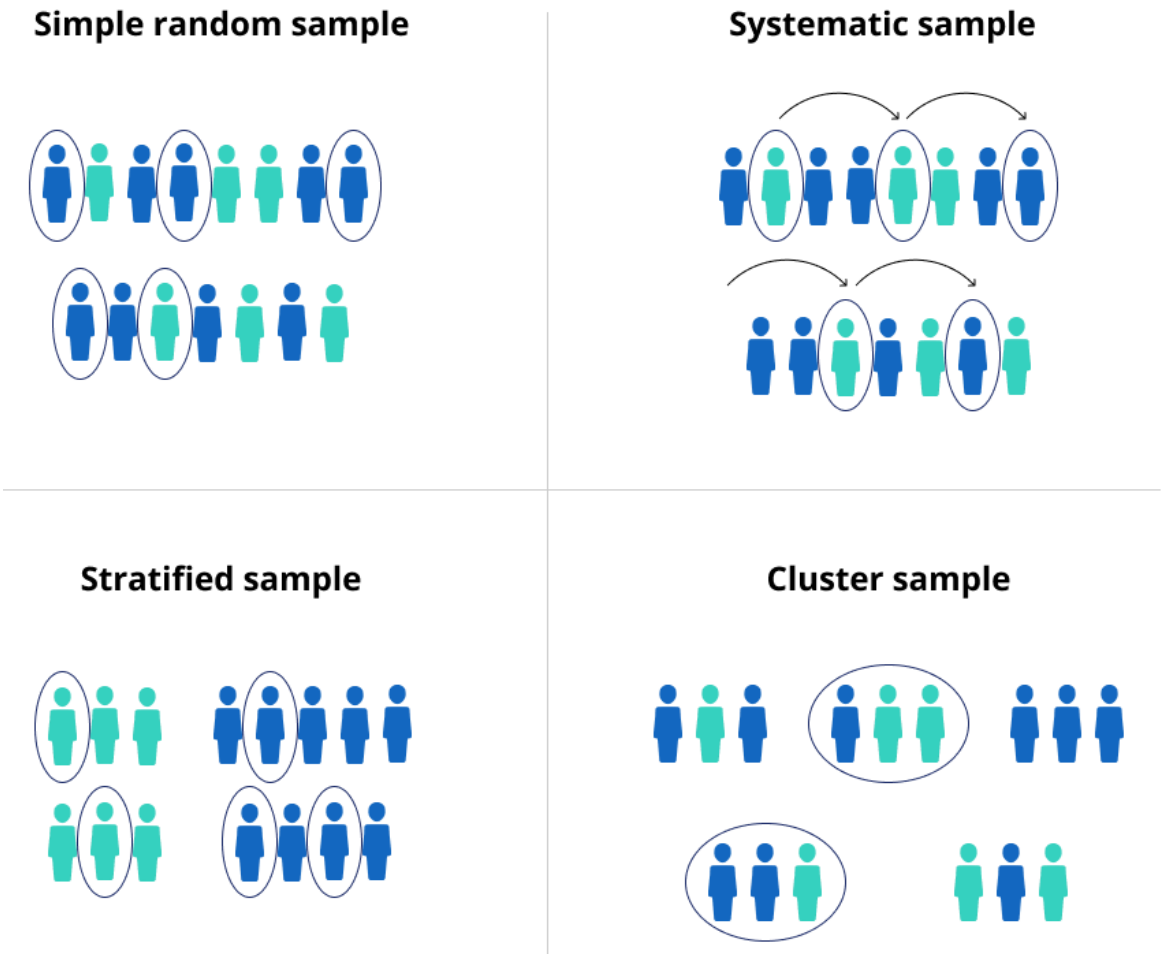
▼ What is sampling?

Data sampling is a statistical analysis technique used to select, manipulate and analyze a representative subset of data points to identify patterns and trends in the larger data set being examined. It enables data scientists, predictive modelers and other data analysts to work with a small, manageable amount of data about a statistical population to build and run analytical models more quickly, while still producing accurate findings. There are many different methods for drawing samples from data, the ideal sampling methods will depend on the data set and situation. Sampling can be based on probability, an approach that uses random numbers that correspond to points in the data set to ensure that there is no correlation between points chosen for the sample. There are also non-probability based methods, such as consecutive sampling where data is collected from every sample that meets the criteria until the predetermined sample size is met.

▼ Give 4 examples of probability-based sampling methods and how they work

- Simple random sampling: Randomly sample subjects from the whole population
- Systematic sampling: A sample is created by setting an interval at which to extract data from the larger population for example select every 10th row in a spreadsheet of 200 items to create a sample size of 20 rows to analyze.
- Stratified sampling: Subsets of the population are created based on a common factor and samples are randomly drawn from each subgroup using simple random sampling

- Cluster sampling: The larger data set is divided into clusters based on a defined factor, then a random sampling of clusters is analyzed. The sampling unit is the whole cluster, instead of sampling individuals from within each group, a researcher will study whole clusters.



▼ Give 4 examples of non probability-based sampling methods and how they work

Non-probability data sampling methods can also be used including:

- Convenience sampling: Data is collected from an easily accessible and available group
- Consecutive sampling: Data is collected from every subject that meets the criteria until the predetermined sample size is met
- Purposive or judgmental sampling: The researcher selects the data to sample based on predefined criteria
- Quota sampling: The researcher ensures equal representation within the sample for all subgroups in the data set or population (random sampling is not used).

Neural Networks and Deep Learning

▼ What are the advantages and disadvantages of neural networks?

Advantages:

- Have achieved state of the art performance on a range of problems including: Image processing, Language processing and translation, Speech recognition and time-series Forecasting
- Neural networks are quite robust to noise in the training data. The training examples may contain errors, which do not affect the final output.
- The input is stored in its own networks instead of a database, hence the loss of data does not affect its working.

Disadvantages:

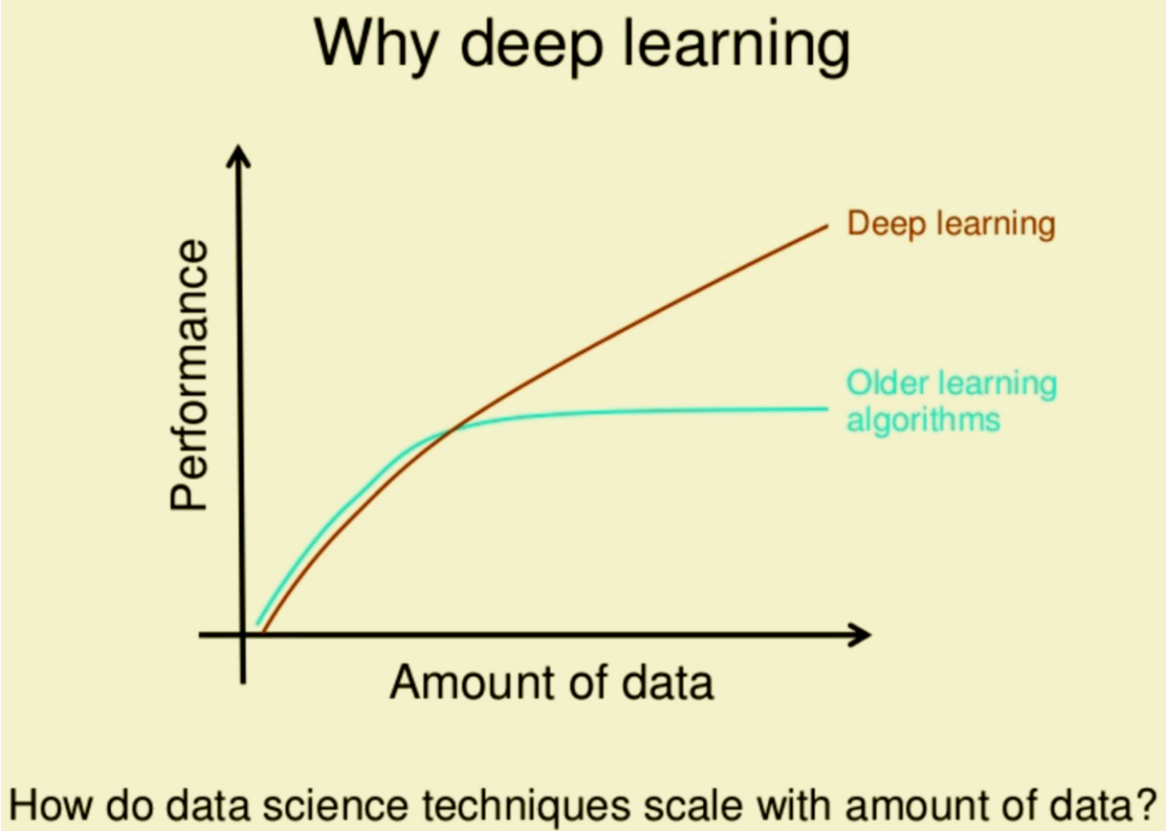
- Hard to interpret: Since with even one a one-layer feed-forward network the number of parameters is huge.
- Requires lots of data for efficient inference.
- Overfit easily if the amount of data is sufficient.
- No ground-truth for hyperparameter tweaking. (For example, still, it is not known whether adding more layers are improving or hurting the model, it is mainly done by brute-force).

▼ What in your opinion is the reason for the popularity of Deep Learning in recent times?

Now although Deep Learning has been around for many years, the major breakthroughs from these techniques came just in recent years. This is because of two main reasons:

- The increase in the amount of data generated through various sources
- The growth in hardware resources required to run these models

GPUs are multiple times faster and they help us build bigger and deeper deep learning models in comparatively less time than we required previously.



▼ Why do we use convolutions for images rather than just FC layers?

- FC layers don't exploit the local structure of images and are not equivariant under translation. Convolution layers on the other hand extract local features as the conv layer outputs are only dependent on adjacent pixels of the previous layer. Convolutional layers are are equivariant under translations.
- Another reason is the amount of parameters: FC layers have a huge number of parameters, convolutional layers have a lot less parameters, since they share kernel over the patches of the whole input image.

▼ What Is the Difference Between Epoch, Batch size, and Number of iterations in Deep Learning?

- One **epoch** = one forward pass and one backward pass of *all* the training examples
- **Batch size** = the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.

- Number of **iterations** = number of passes, each pass using [batch size] number of examples. To be clear, one pass = one forward pass + one backward pass (we do not count the forward pass and backward pass as two different passes).

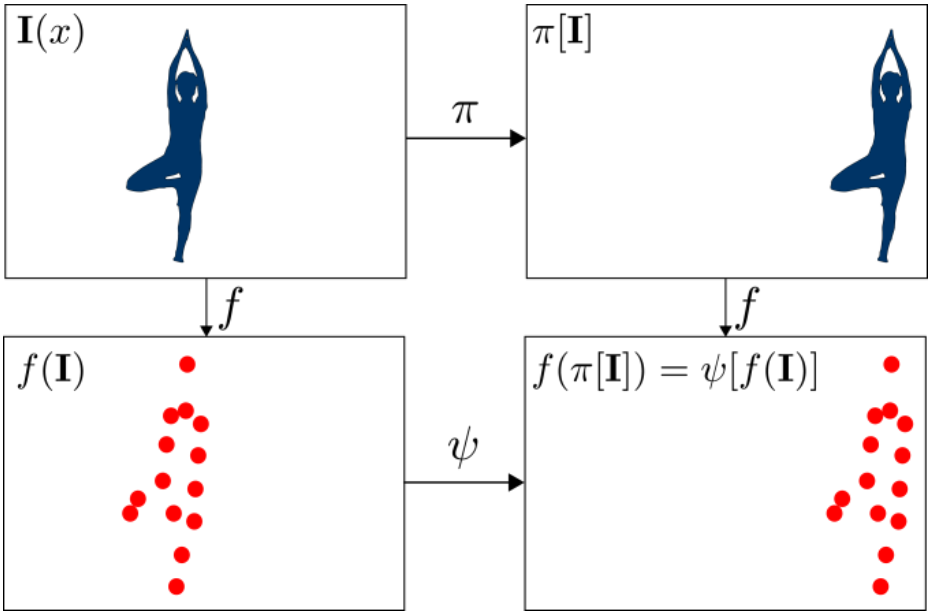
Example: if you have 1000 training examples, and your batch size is 500, then it will take 2 iterations to complete 1 epoch.

▼ What makes CNNs translation invariant?

Translational Invariance is a result of the pooling operation. In a traditional CNN architecture, there are three stages. In the first stage, the layer performs convolution operation on the input to give linear activations. In the second stage, the resultant activations are passed through a non-linear activation function such as sigmoid, tanh or relu. In the third stage, we perform the pooling operation to modify the output further.

In pooling operation, we replace the output of the convnet at a certain location with a summary statistic of the nearby outputs such a maximum in case of Max Pooling. As we replace the output with the max in case of max-pooling, hence even if we change the input slightly, it won't affect the values of most of the pooled outputs. Translational Invariance is a useful property where the exact location of the object is not required. For e.g if you are building a model to detect faces all you need to detect is whether eyes are present or not, it's exact position is not necessary. While in segmentation tasks, the exact position is required.

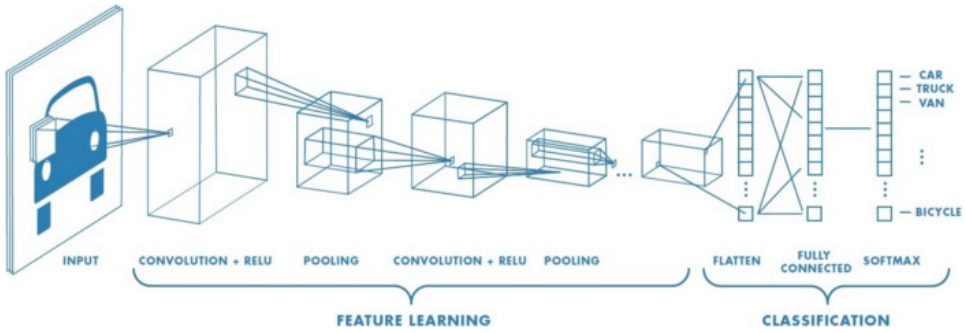
The use of pooling can be viewed as adding a strong prior that the function the layer learns must be invariant to translation. When the prior is correct, it can greatly improve the statistical efficiency of the network.



▼ What are the 4 main types of layers used to build a CNN?

There are four layers typically used to build a CNN:

1. **Convolutional Layer** – the layer that performs a convolutional operation, creating several smaller picture windows to go over the data.
2. **ReLU Layer** – it brings non-linearity to the network and converts all the negative pixels to zero. The output is a rectified feature map.
3. **Pooling Layer** – pooling is a down-sampling operation that reduces the dimensionality of the feature map.
4. **Fully Connected Layer** – this layer recognizes and classifies the objects in the image.

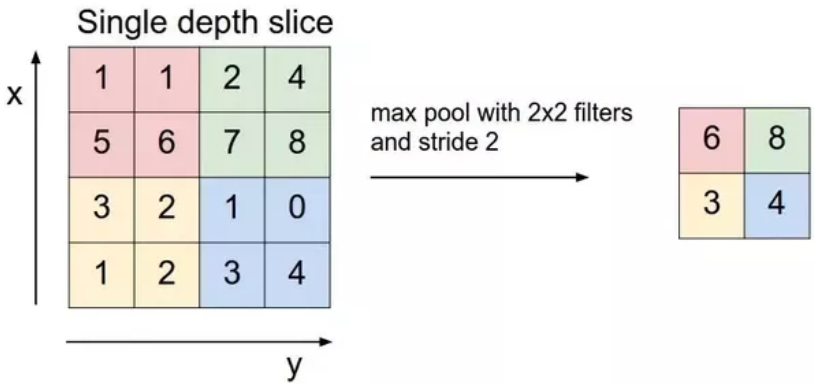


▼ What are 3 types of spatial pooling that can be used?

Pooling layers reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

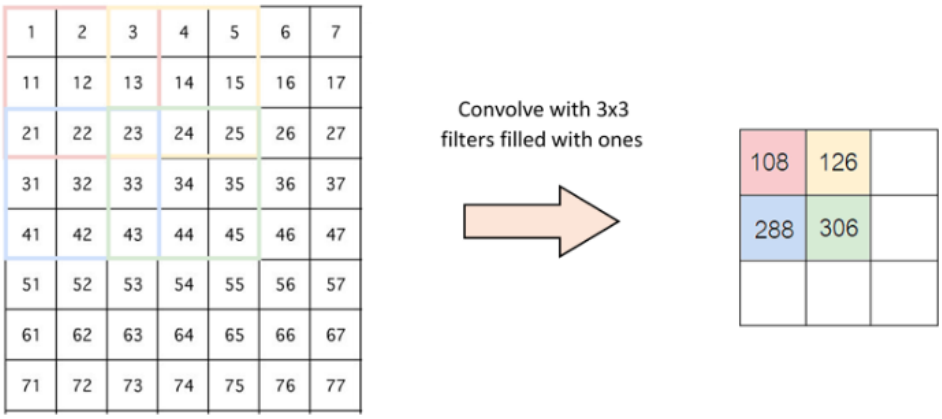
- **Max Pooling**
- **Average Pooling**
- **Sum Pooling**

Max pooling takes the largest element from the rectified feature map. Average pooling takes the average of all the elements in the feature map. Sum pooling takes the sum of all the elements in the feature map.



▼ What is the stride in convolutional layers?

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.



▼ What are vanishing and exploding gradients?

In a network of n hidden layers, n derivatives will be multiplied together. If the derivatives are large then the gradient will increase exponentially as we propagate down the model until they eventually explode, and this is what we call the problem of exploding gradient. Alternatively, if the derivatives are small then the gradient will decrease exponentially as we propagate through the model until it eventually vanishes, and this is the vanishing gradient problem.

▼ What are 4 possible solutions to vanishing and exploding gradients?

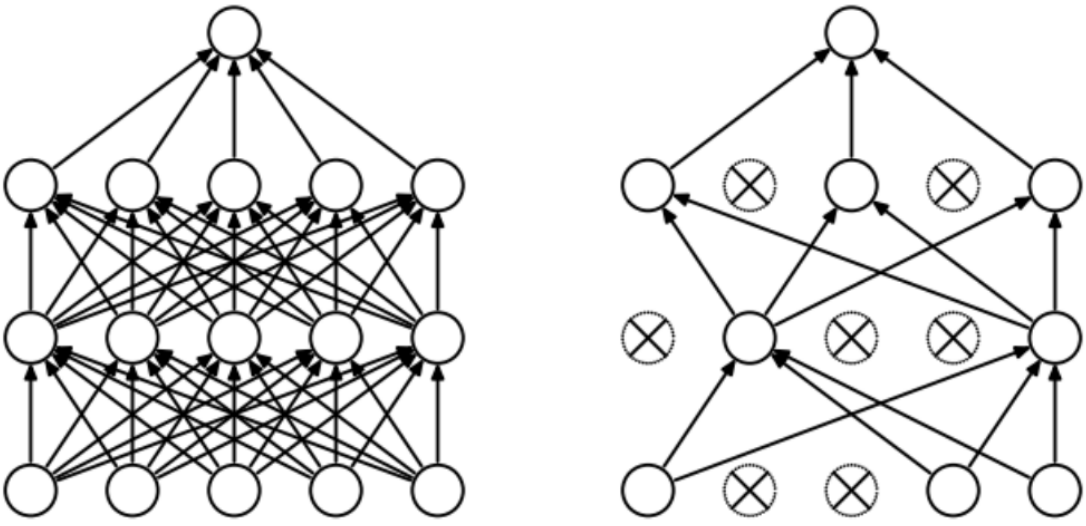
1. **Reducing the amount of Layers**
   This is the solution could be used in both, scenarios (exploding and vanishing gradient). However, by reducing the amount of layers in our network, we give up some of our models complexity, since having more layers makes the networks more capable of representing complex mappings.
2. **Gradient Clipping (Exploding Gradients)**
   Gradient clipping is a technique that tackles exploding gradients. The idea of gradient clipping is very simple: If the gradient gets too large, we rescale it to keep it small.
3. **Weight Regularization** Another approach, is to check the size of network weights and apply a penalty to the networks loss function for large weight values. This is called weight regularization and often an L1 (absolute weights) or an L2 (squared weights) penalty can be used.
4. **Use Long Short-Term Memory Networks** In recurrent neural networks, gradient exploding can occur given the inherent instability in the training of this type of network, e.g. via Backpropagation through time that essentially transforms the recurrent network into a deep multilayer Perceptron neural network. Exploding gradients can be reduced by using the Long Short-Term Memory (LSTM) memory units and perhaps related gated-type neuron structures. Adopting LSTM memory units is a new best practice for recurrent neural networks for sequence prediction.

▼ What is dropout for neural networks? What effect does dropout have?

Dropout is a regularization technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

As a neural network learns, neuron weights settle into their context within the network. Weights of neurons are tuned for specific features providing some specialization. Neighboring neurons become to rely on this specialization, which if taken too far can result in a fragile model too specialized to the training data. This reliant on context for a neuron during training is referred to complex co-adaptations.

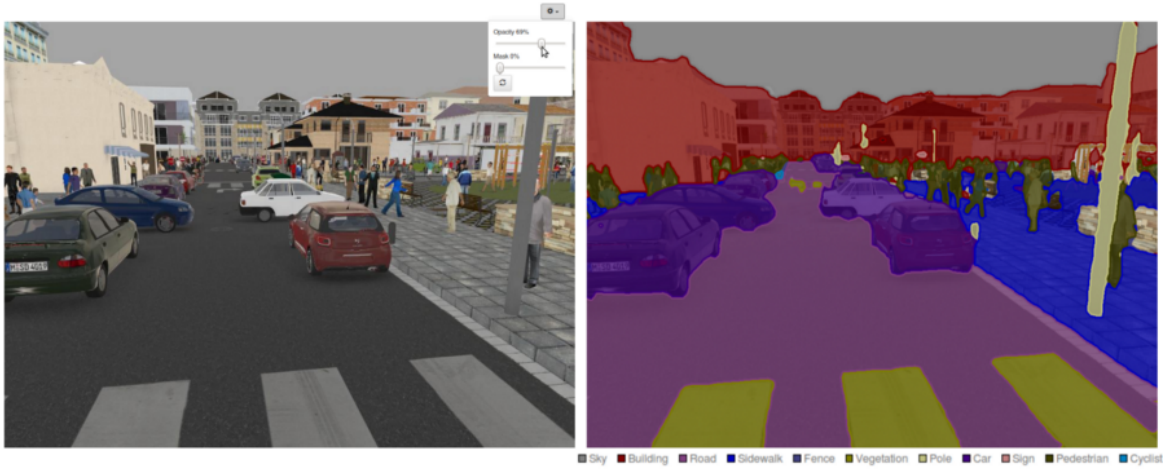
The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data.

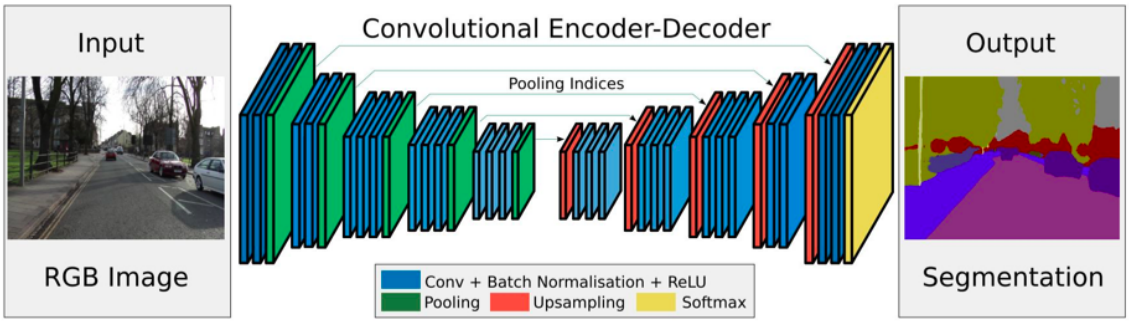


(a) Standard Neural Net (b) After applying dropout.

▼ Why do segmentation CNNs typically have an encoder-decoder style / structure?

The encoder CNN can basically be thought of as a feature extraction network, while the decoder uses that information to predict the image segments by “decoding” the features and upscaling to the original image size.





▼ What is batch normalization and why does it work?

Training Deep Neural Networks is complicated by the fact that the distribution of each layer’s inputs changes during training, as the parameters of the previous layers change. The idea is then to normalize the inputs of each layer in such a way that they have a mean output activation of zero and standard deviation of one. This is done for each individual mini-batch at each layer i.e., compute the mean and variance of that mini-batch alone, then normalize. This is analogous to how the inputs to networks are standardized.

How does this help? We know that normalizing the inputs to a network helps it learn. But a network is just a series of layers, where the output of one layer becomes the input to the next. That means we can think of any layer in a neural network as the first layer of a smaller subsequent network. Thought of as a series of neural networks feeding into each other, we normalize the output of one layer before applying the activation function, and then feed it into the following layer (sub-network).

▼ Why would you use many small convolutional kernels such as 3x3 rather than a few large ones?

There are 2 reasons:

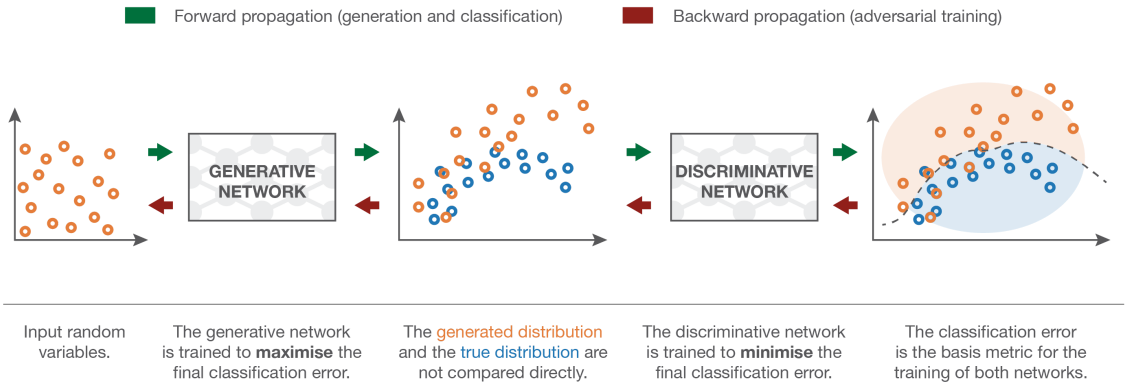
First, you can use several smaller kernels rather than few large ones to get the same receptive field and capture more spatial context, but with the smaller kernels you are using less parameters and computations.

Secondly, because with smaller kernels you will be using more filters, you’ll be able to use more activation functions and thus have a more discriminative mapping function being learned by your CNN.

▼ What is the idea behind GANs?

GANs, or generative adversarial networks, consist of two networks (D,G) where D is the “discriminator” network and G is the “generative” network. The goal is to create data — images, for instance, which are undistinguishable from real images. Suppose we want to create an adversarial example of a cat. The network G will generate images. The network

D will classify images according to whether they are a cat or not. The cost function of G will be constructed such that it tries to “fool” D — to classify its output always as cat.



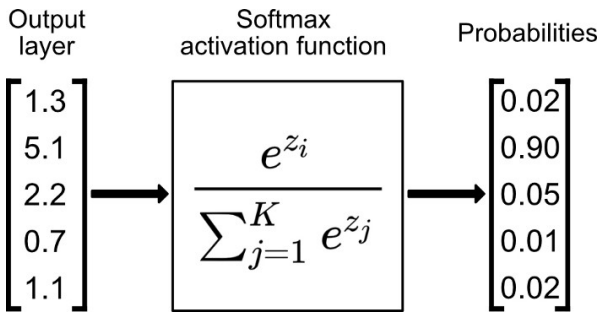
▼ Why we generally use Softmax non-linearity function as last operation in-network?

It is because it takes in a vector of real numbers (positive, negative, whatever, there are no constraints) and returns a probability distribution. The formula of the softmax function is:

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

It states that we need to apply a standard exponential function to each element of the output layer, and then normalize these values by dividing by the sum of all the exponentials. Doing so ensures the sum of all exponentiated values adds up to 1.

It should be clear that the output is a probability distribution: each element is non-negative and the sum over all components is 1.



▼ What is the following activation function? What are the advantages and disadvantages of this activation function?

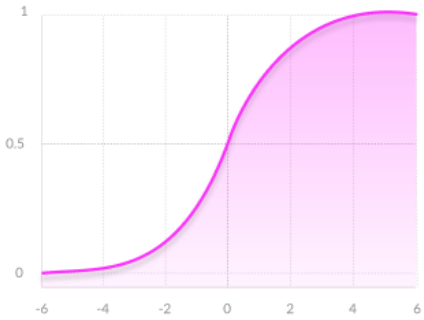
Sigmoid / Logistic

Advantages

- **Smooth gradient**, preventing “jumps” in output values.
- **Output values bound** between 0 and 1, normalizing the output of each neuron.
- **Clear predictions**—For X above 2 or below -2, tends to bring the Y value (the prediction) to the edge of the curve, very close to 1 or 0. This enables clear predictions.

Disadvantages

- **Vanishing gradient**—for very high or very low values of X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further, or being too slow to reach an accurate prediction.
- **Outputs not zero centered**.
- **Computationally expensive**



▼ What is the following activation function? What are the advantages and disadvantages of this activation function?

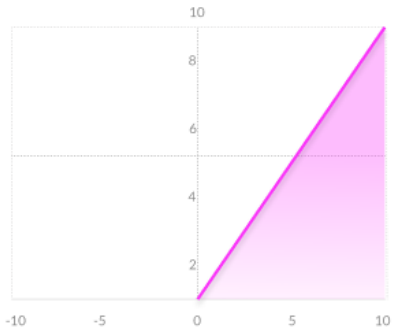
ReLU (Rectified Linear Unit)

Advantages

- **Computationally efficient**—allows the network to converge very quickly
- **Non-linear—**although it looks like a linear function, ReLU has a derivative function and allows for backpropagation

Disadvantages

- **The Dying ReLU problem**—when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.



▼ What is the following activation function? What are the advantages and disadvantages of this activation function?

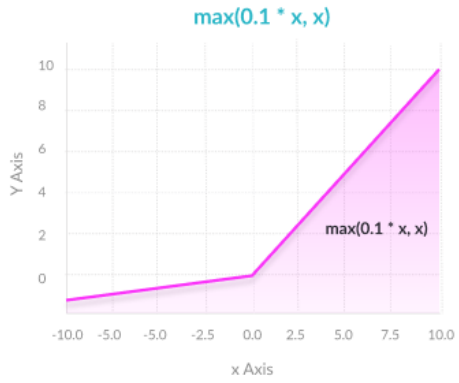
Leaky ReLU

Advantages

- **Prevents dying ReLU problem**—this variation of ReLU has a small positive slope in the negative area, so it does enable backpropagation, even for negative input values
- Otherwise like ReLU

Disadvantages

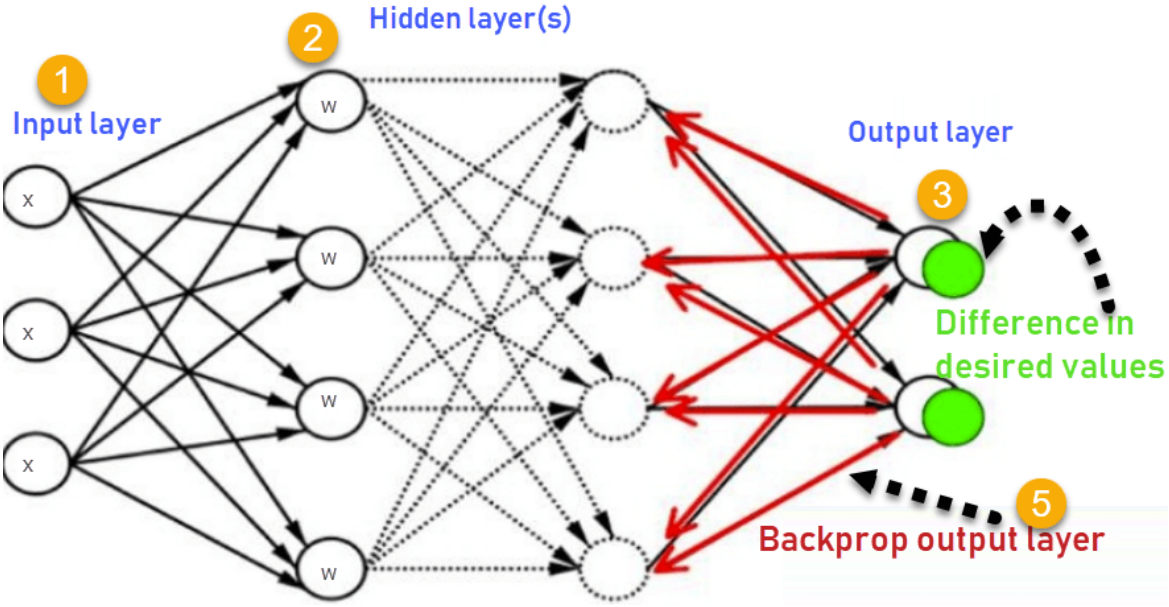
- **Results not consistent**—leaky ReLU does not provide consistent predictions for negative input values.



▼ What is backpropagation and how does it work?

After a neural network is defined with initial weights, and a forward pass is performed to generate the initial prediction, there is an error function which defines how far away the model is from the true prediction. There are many possible algorithms that can minimize the error function—for example, one could do a brute force search to find the weights that generate the smallest error. However, for large neural networks, a training algorithm is needed that is very computationally efficient. Backpropagation is that algorithm—it can discover the optimal weights relatively quickly, even for a network with millions of weights.

How Backpropagation Works



1. **Forward pass**—weights are initialized and inputs from the training set are fed into the network. The forward pass is carried out and the model generates its initial prediction.
2. **Error function**—the error function is computed by checking how far away the prediction is from the known true value.
3. **Backpropagation with gradient descent**—the backpropagation algorithm calculates how much the output values are affected by each of the weights in the model. To do this, it calculates partial derivatives, going back from the error function to a specific neuron and its weight. This provides complete traceability from total errors, back to a

specific weight which contributed to that error. The result of backpropagation is a set of weights that minimize the error function.

4. **Weight update**—weights can be updated after every sample in the training set, but this is usually not practical. Typically, a batch of samples is run in one big forward pass, and then backpropagation performed on the aggregate result. The *batch size* and number of batches used in training, called *iterations*, are important hyperparameters that are tuned to get the best results. Running the entire training set through the backpropagation process is called an *epoch*.

▼ What are the common hyperparameters related to neural network structure?

- Number of hidden layers
- Dropout
- Activation function
- Weights initialization

▼ What are the common hyperparameters related to training neural networks?

- Learning rate
- Epoch, iterations and batch size
- Optimizer algorithm
- Momentum

▼ What are 4 methods of hyperparameter tuning?

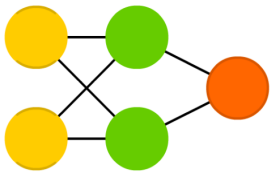
1. **Manual hyperparameter tuning**—an experienced operator can guess parameter values that will achieve very high accuracy. This requires trial and error.
2. **Grid search**—this involves systematically testing multiple values of each hyperparameter and retraining the model for each combination.
3. **Random search**— using random hyperparameter values is actually more effective than manual search or grid search.
4. **Bayesian optimization—** trains the model with different hyperparameter values over and over again, and tries to observe the shape of the function generated by different parameter values. It then extends this function to predict the best possible values. This method provides higher accuracy than random search.

Neural Network Key

▼ Using the above neural network key, state the name of the following network and give some basic information about it and examples of applications that it could be used for

Feed forward neural network (FFNN)

These a very simple networks, they feed information from the front to the back. The simplest somewhat practical network has two input cells and one output cell, which can be used to model logic gates. One usually trains FFNNs through back-propagation, giving the network paired datasets of “what goes in” and “what we want to have coming out”. The error being back-propagated is often some variation of the difference between the input and the output (like MSE or just the linear difference). Given that the network has enough hidden neurons, it can theoretically always model the relationship between the input and output. Practically their use is a lot more limited but they are popularly combined with other networks to form new networks.



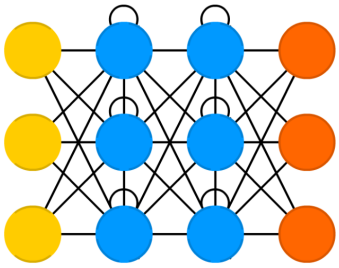
▼ Using the above neural network key, state the name of the following network and give some examples of applications it would be used for

Recurrent neural networks (RNN)

RNNs are FFNNs with a time twist: they are not stateless; they have connections between passes, connections through time. Neurons are fed information not just from the previous layer but also from themselves from the previous pass. This

means that the order in which you feed the input and train the network matters: feeding it “milk” and then “cookies” may yield different results compared to feeding it “cookies” and then “milk”. One big problem with RNNs is the vanishing (or exploding) gradient problem where, depending on the activation functions used, information rapidly gets lost over time, just like very deep FFNNs lose information in depth. Intuitively this wouldn’t be much of a problem because these are just weights and not neuron states, but the weights through time is actually where the information from the past is stored; if the weight reaches a value of 0 or 1 000 000, the previous state won’t be very informative.

RNNs can in principle be used in many fields as most forms of data that don’t actually have a timeline (i.e. unlike sound or video) can be represented as a sequence. A picture or a string of text can be fed one pixel or character at a time, so the time dependent weights are used for what came before in the sequence, not actually from what happened x seconds before. In general, recurrent networks are a good choice for advancing or completing information, such as autocompletion.

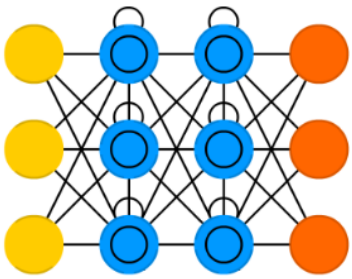


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Long short-term memory (LSTM)

Long / short term memory (LSTM) networks try to combat the vanishing / exploding gradient problem by introducing gates and an explicitly defined memory cell. These are inspired mostly by circuitry, not so much biology. Each neuron has a memory cell and three gates: input, output and forget. The function of these gates is to safeguard the information by stopping or allowing the flow of it. The input gate determines how much of the information from the previous layer gets stored in the cell. The output layer takes the job on the other end and determines how much of the next layer gets to know about the state of this cell. The forget gate seems like an odd inclusion at first but sometimes it’s good to forget: if it’s learning a book and a new chapter begins, it may be necessary for the network to forget some characters from the previous chapter.

LSTMs have been shown to be able to learn complex sequences, such as writing like Shakespeare or composing primitive music. Note that each of these gates has a weight to a cell in the previous neuron, so they typically require more resources to run.

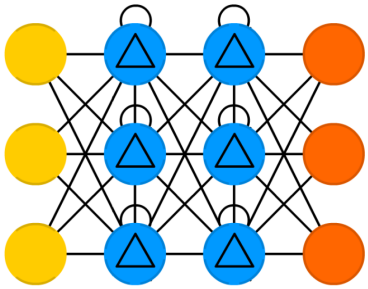


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Gated recurrent units (GRU)

Gated recurrent units (GRU) are a slight variation on LSTMs. They have one less gate and are wired slightly differently: instead of an input, output and a forget gate, they have an update gate. This update gate determines both how much information to keep from the last state and how much information to let in from the previous layer. The reset gate

functions much like the forget gate of an LSTM but it's located slightly differently. They always send out their full state, they don't have an output gate. In most cases, they function very similarly to LSTMs, with the biggest difference being that GRUs are slightly faster and easier to run (but also slightly less expressive). In practice these tend to cancel each other out, as you need a bigger network to regain some expressiveness which then in turn cancels out the performance benefits. In some cases where the extra expressiveness is not needed, GRUs can outperform LSTMs.

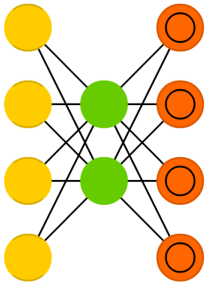


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Autoencoders (AE)

Autoencoders (AE) are somewhat similar to FFNNs as AEs are more like a different use of FFNNs than a fundamentally different architecture. The basic idea behind autoencoders is to encode information (as in compress, not encrypt) automatically, hence the name. The entire network always resembles an hourglass like shape, with smaller hidden layers than the input and output layers. AEs are also always symmetrical around the middle layer(s) (one or two depending on an even or odd amount of layers). The smallest layer(s) is/are almost always in the middle, the place where the information is most compressed (the chokepoint of the network). Everything up to the middle is called the encoding part, everything after the middle the decoding and the middle (surprise) the code. One can train them using backpropagation by feeding input and setting the error to be the difference between the input and what came out. AEs can be built symmetrically when it comes to weights as well, so the encoding weights are the same as the decoding weights.

Autoencoders can be used for a range of different applications including: Dimensionality Reduction, Image Compression, Image Denoising, Feature Extraction, Image generation, Recommendation system.

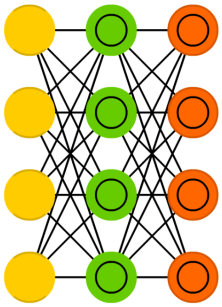


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Variational autoencoders (VAE)

Variational autoencoders (VAE) have the same architecture as AEs but are "taught" something else: an approximated probability distribution of the input samples. They rely on Bayesian mathematics regarding probabilistic inference and independence, as well as a reparameterization trick to achieve this different representation. The inference and independence parts make sense intuitively, but they rely on somewhat complex mathematics. The basics come down to this: take influence into account. If one thing happens in one place and something else happens somewhere else, they are not necessarily related. If they are not related, then the error propagation should consider that. This is a useful approach because neural networks are large graphs (in a way), so it helps if you can rule out influence from some nodes to other nodes as you dive into deeper layers.

Variational autoencoders are generative networks and can be used to generate new faces from a data set of faces and give those faces different emotional expressions and they can also be used for molecule and drug design, they have also been used to compose music by feeding in previous recorded songs.

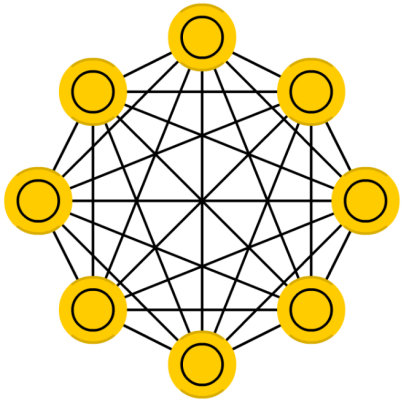


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Hopfield network (HN)

A Hopfield network (HN) is a network where every neuron is connected to every other neuron; it is a completely entangled plate of spaghetti as even all the nodes function as everything. Each node is input before training, then hidden during training and output afterwards. The networks are trained by setting the value of the neurons to the desired pattern after which the weights can be computed. The weights do not change after this. Once trained for one or more patterns, the network will always converge to one of the learned patterns because the network is only stable in those states. Each neuron has an activation threshold which scales to this temperature, which if surpassed by summing the input causes the neuron to take the form of one of two states (usually -1 or 1, sometimes 0 or 1). Updating the network can be done synchronously or more commonly one by one. If updated one by one, a fair random sequence is created to organize which cells update in what order (fair random being all options (n) occurring exactly once every n items). This is so you can tell when the network is stable (done converging), once every cell has been updated and none of them changed, the network is stable (annealed). These networks are often called associative memory because the converge to the most similar state as the input; if humans see half a table we can image the other half, this network will converge to a table if presented with half noise and half a table.

Hopfield networks are a form of associative memory (just like the human mind), and basically, it's initially trained to store a number of patterns, and then it's able to recognize any of the learned patterns by exposure to part or even corrupted information. Common applications are those where pattern recognition is useful, and Hopfield networks have been used for image detection and recognition, enhancement of X-Ray images, medical image restoration, etc. They are not used as much now and were big in the 80's.

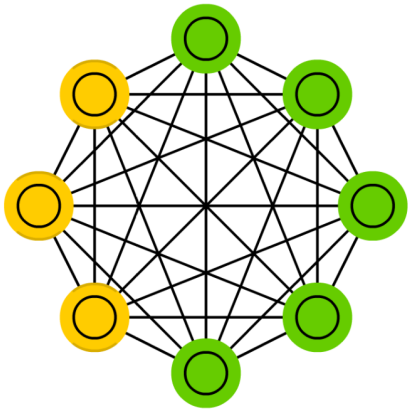


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Boltzmann machines (BM)

Boltzmann machines (BM) are a lot like HNs, but: some neurons are marked as input neurons and others remain “hidden”. The input neurons become output neurons at the end of a full network update. It starts with random weights and learns through back-propagation, or more recently through contrastive divergence (a Markov chain is used to determine the gradients between two informational gains). Compared to a HN, the neurons mostly have binary activation patterns. As hinted by being trained by MCs, BMs are stochastic networks. The training and running process of a BM is fairly similar to a HN: one sets the input neurons to certain clamped values after which the network is set free. While free the cells can get any value and we repetitively go back and forth between the input and hidden neurons. The activation is controlled by a global temperature value, which if lowered lowers the energy of the cells. This lower energy causes their activation patterns to stabilize. The network reaches an equilibrium given the right temperature.

Boltzmann machines with unconstrained connectivity have not proven useful for practical problems in machine learning or inference, but if the connectivity is properly constrained, the learning can be made efficient enough to be useful for practical problems, such as combinatorial optimization.

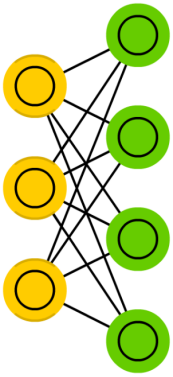


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Restricted Boltzmann machines (RBM)

Restricted Boltzmann machines (RBM) are remarkably similar to BMs (surprise) and therefore also similar to HNs. The biggest difference between BMs and RBMs is that RBMs are a better usable because they are more restricted. In restricted Boltzmann machines (RBM) there are only connections (dependencies) between hidden and visible units, and none between units of the same type (no hidden-hidden, nor visible-visible connections). RBMs can be trained like FFNNs with a twist: instead of passing data forward and then back-propagating, you forward pass the data and then backward pass the data (back to the first layer). After that you train with forward-and-back-propagation.

RBMs have found applications in dimensionality reduction, classification, collaborative filtering, feature learning, topic modelling and even many body quantum mechanics. They can be trained in either supervised or unsupervised ways, depending on the task.



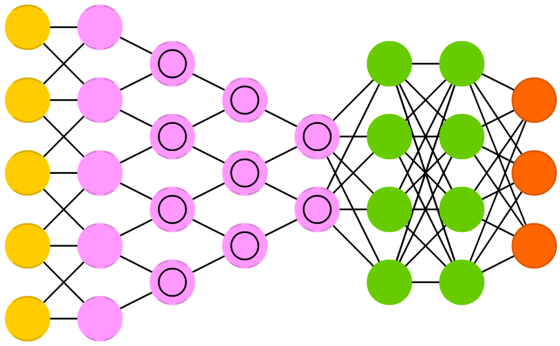
▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Convolutional neural networks (CNN or deep convolutional neural networks, DCNN)

Convolutional neural networks (CNN or deep convolutional neural networks, DCNN) are quite different from most other networks. They are primarily used for image processing but can also be used for other types of input such as as audio. A typical use case for CNNs is where you feed the network images and the network classifies the data, e.g. it outputs “cat” if you give it a cat picture and “dog” when you give it a dog picture. CNNs tend to start with an input “scanner” which is not intended to parse all the training data at once. For example, to input an image of 200 x 200 pixels, you wouldn’t want a layer with 40 000 nodes. Rather, you create a scanning input layer of say 20 x 20 which you feed the first 20 x 20 pixels of the image (usually starting in the upper left corner). Once you passed that input (and possibly use it for training) you feed it the next 20 x 20 pixels: you move the scanner one pixel to the right. Note that one wouldn’t move the input 20 pixels (or whatever scanner width) over, you’re not dissecting the image into blocks of 20 x 20, but rather you’re crawling over it. This input data is then fed through convolutional layers instead of normal layers, where not all nodes are connected to all nodes.

Each node only concerns itself with close neighboring cells (how close depends on the implementation, but usually not more than a few). These convolutional layers also tend to shrink as they become deeper, mostly by easily divisible factors of the input (so 20 would probably go to a layer of 10 followed by a layer of 5). Powers of two are very commonly used here, as they can be divided cleanly and completely by definition: 32, 16, 8, 4, 2, 1. Besides these convolutional layers, they also often feature pooling layers. Pooling is a way to filter out details: a commonly found pooling technique is max pooling, where we take say 2 x 2 pixels and pass on the pixel with the most amount of red. To apply CNNs for audio, you basically feed the input audio waves and inch over the length of the clip, segment by segment. Real world implementations of CNNs often glue an FFNN to the end to further process the data, which allows for highly non-linear abstractions. These networks are called DCNNs but the names and abbreviations between these two are often used interchangeably.

They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing and brain-computer interfaces.

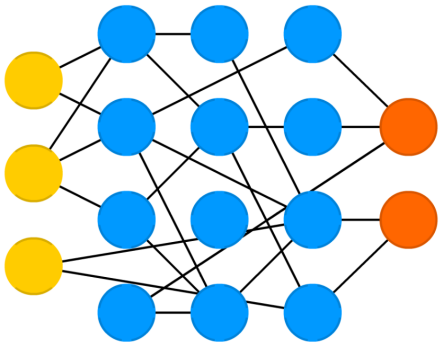


▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Echo state networks (ESN)

Echo state networks (ESN) are yet another different type of (recurrent) network. This one sets itself apart from others by having random connections between the neurons (i.e. not organized into neat sets of layers), and they are trained differently. Instead of feeding input and back-propagating the error, we feed the input, forward it and update the neurons for a while, and observe the output over time. The input and the output layers have a slightly unconventional role as the input layer is used to prime the network and the output layer acts as an observer of the activation patterns that unfold over time. During training, only the connections between the observer and the (soup of) hidden units are changed.

Echo state networks have been used for time series data mining, time series forecasting and sequence prediction.

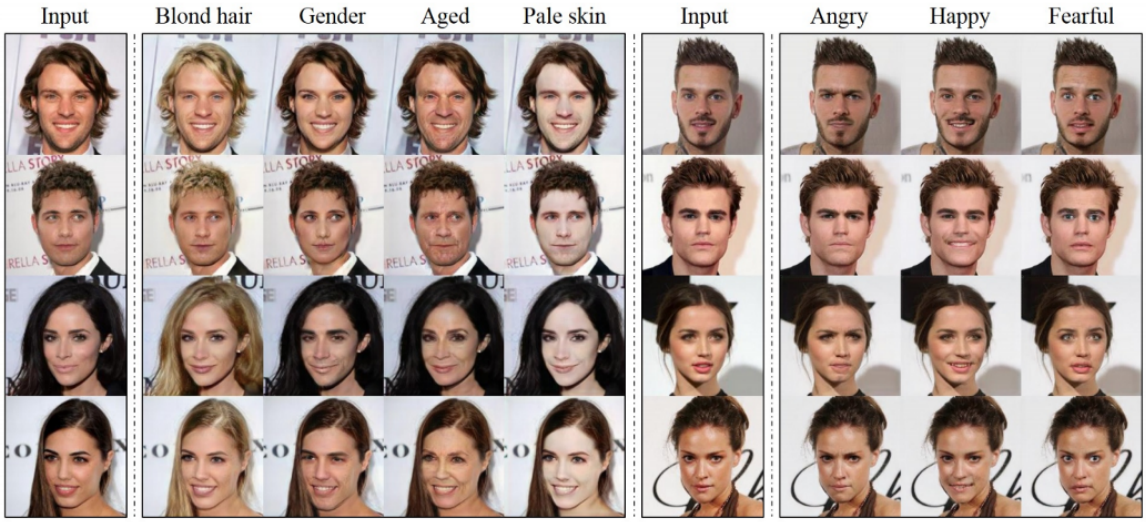


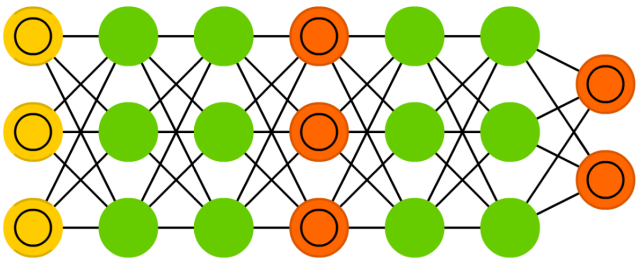
▼ Using the neural network key, state the name of the following network and give some examples of applications it would be used for

Generative adversarial networks (GAN)

Generative adversarial networks (GAN) are from a different breed of networks, they are twins: two networks working together. GANs consist of any two networks (although often a combination of FFs and CNNs), with one tasked to generate content and the other has to judge content. The discriminating network receives either training data or generated content from the generative network. How well the discriminating network was able to correctly predict the data source is then used as part of the error for the generating network. This creates a form of competition where the discriminator is getting better at distinguishing real data from generated data and the generator is learning to become less predictable to the discriminator. This works well in part because even quite complex noise-like patterns are eventually predictable but generated content similar in features to the input data is harder to learn to distinguish. GANs can be quite difficult to train, as you don't just have to train two networks (either of which can pose it's own problems) but their dynamics need to be balanced as well. If prediction or generation becomes to good compared to the other, a GAN won't converge as there is intrinsic divergence.

GANs have a number of application including Generate Examples for Image Datasets, Generate Photographs of Human Faces, Generate Cartoon Characters, Text-to-Image Translation, Semantic-Image-to-Photo Translation, Generate New Human Poses, Photos to Emojis, Video Prediction, 3D Object Generation.





Statistics

▼ The probability that an item is sold on Amazon by seller A is 0.6 and 0.8 for seller B. What is the probability that item would be found on Amazon website?

The probability of finding the item on Amazon can be explained as so:

We can reword the above as P(A) = 0.6 and P(B) = 0.8. Furthermore, let's assume that these are independent events, meaning that the probability of one event is not impacted by the other. We can then use the formula…

P(A or B) = P(A) + P(B) — P(A and B)

P(A or B) = 0.6 + 0.8 — (0.6*0.8)

P(A or B) = 0.92

▼ You randomly draw a coin from 100 coins — 1 unfair coin (head-head), 99 fair coins (head-tail) and roll it 10 times. If the result is 10 heads, what is the probability that the coin is unfair?

This can be answered using the Bayes Theorem. The extended equation for the Bayes Theorem is the following:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|\neg A)P(\neg A)}$$

Assume that the probability of picking the unfair coin is denoted as P(A) and the probability of flipping 10 heads in a row is denoted as P(B). Then P(B|A) is equal to 1, P(B | ¬A) is equal to 0.5^{10}, and P(¬A) is equal to 0.99.

If you fill in the equation, then P(A|B) = 0.9118 or 91.18%.

▼ Find the total number of ways 5 people can sit in 5 empty seats.

Factorial Formula: n! = n x (n -1) x (n — 2) x … x 2 x 1

= 5 x 4 x 3 x 2 x 1 = 120

▼ A code has 4 digits in a particular order and the digits range from 0 to 9. How many permutations are there if one digit can only be used once?

Permutations: $P(n,r) = n!/(n-r)!$

$$P(n,r) = 10!/(10–4)! = (10*9*8*7*6*5*4*3*2*1)/(6*5*4*3*2*1) = 5040$$

▼ To win the lottery, you must select the 5 correct numbers in any order from 1 to 52. What is the number of possible combinations?

Combinations Formula: $C(n,r) = (n!)/[(n-r)!r!]$

$C(n,r) = 52!/(52–5)!5! = 2,598,960$

▼ A box has 12 red cards and 12 black cards. Another box has 24 red cards and 24 black cards. You want to draw two cards at random from one of the two boxes, one card at a time. Which box has a higher probability of getting cards of the same color and why?

The box with 24 red cards and 24 black cards has a higher probability of getting two cards of the same color. Let's walk through each step.

Let's say the first card you draw from each deck is a red Ace.

This means that in the deck with 12 reds and 12 blacks, there's now 11 reds and 12 blacks. Therefore your odds of drawing another red are equal to 11/(11+12) or 11/23.

In the deck with 24 reds and 24 blacks, there would then be 23 reds and 24 blacks. Therefore your odds of drawing another red are equal to 23/(23+24) or 23/47.

Since 23/47 > 11/23, the second deck with more cards has a higher probability of getting the same two cards.

▼ You are at a Casino and have two dices to play with. You win $10 every time you roll a 5. If you play till you win and then stop, what is the expected payout?

1,1	2,1	3,1	4,1	5,1	6,1
1,2	2,2	3,2	4,2	5,2	6,2
1,3	2,3	3,3	4,3	5,3	6,3
1,4	2,4	3,4	4,4	5,4	6,4
1,5	2,5	3,5	4,5	5,5	6,5
1,6	2,6	3,6	4,6	5,6	6,6

1,1	2,1	3,1	4,1	5,1	6,1
1,2	2,2	3,2	4,2	5,2	6,2
1,3	2,3	3,3	4,3	5,3	6,3
1,4	2,4	3,4	4,4	5,4	6,4
1,5	2,5	3,5	4,5	5,5	6,5
1,6	2,6	3,6	4,6	5,6	6,6

- Let's assume that it costs $5 every time you want to play.
- There are 36 possible combinations with two dice.
- Of the 36 combinations, there are 4 combinations that result in rolling a five (*see blue*). This means that there is a 4/36 or 1/9 chance of rolling a 5.
- A 1/9 chance of winning means you'll lose eight times and win once (theoretically).
- Therefore, your expected payout is equal to $10.00 * 1 — $5.00 * 9= -$35.00.

▼ You are about to get on a plane to London, you want to know whether you have to bring an umbrella or not. You call three of your random friends and ask each one of them if it's raining. The probability that your friend is telling the truth is 2/3 and the probability that they are playing a prank on you by lying is 1/3. If all 3 of them tell that it is raining, then what is the probability that it is actually raining in London.

You can tell that this question is related to Bayesian theory because of the last statement which essentially follows the structure, "What is the probability A is true **given** B is true?" Therefore we need to know the probability of it raining in London on a given day. Let's assume it's 25%.

- P(A) = probability of it raining = 25%
- P(B) = probability of all 3 friends say that it's raining
- P(A|B) probability that it's raining given they're telling that it is raining
- P(B|A) probability that all 3 friends say that it's raining given it's raining = $(2/3)^3 = 8/27$

Step 1: Solve for P(B)

- P(A|B) = P(B|A) * P(A) / P(B), can be rewritten as
- P(B) = P(B|A) * P(A) + P(B|not A) * P(not A)
- P(B) = $(2/3)^3$ * 0.25 + $(1/3)^3$ * 0.75 = 0.25*8/27 + 0.75*1/27

Step 2: Solve for P(A|B)

- P(A|B) = 0.25 * (8/27) / (0.25*8/27 + 0.75*1/27)
- P(A|B) = 8 / (8 + 3) = 8/11

Therefore, if all three friends say that it's raining, then there's an 8/11 chance that it's actually raining.

▼ You are given 40 cards with four different colors- 10 Green cards, 10 Red Cards, 10 Blue cards, and 10 Yellow cards. The cards of each color are numbered from one to ten. Two cards are picked at random. Find out the probability that the cards picked are not of the same number and same color.

Since these events are not independent, we can use the rule:

- P(A and B) = P(A) * P(B|A) ,which is also equal to
- P(not A and not B) = P(not A) * P(not B | not A)

For example:

- P(not 4 and not yellow) = P(not 4) * P(not yellow | not 4)
- P(not 4 and not yellow) = (36/39) * (27/36)
- P(not 4 and not yellow) = 0.692

Therefore, the probability that the cards picked are not the same number and the same color is 69.2%.

▼ How do you assess the statistical significance of an insight?

You would perform hypothesis testing to determine statistical significance. First, you would state the null hypothesis and alternative hypothesis. Second, you would calculate the p-value, the probability of obtaining the observed results of a test assuming that the null hypothesis is true. Last, you would set the level of the significance (alpha) and if the p-value is less than the alpha, you would reject the null — in other words, the result is statistically significant.

▼ Explain selection bias (with regard to a dataset, not variable selection). Why is it important? How can data management procedures such as missing data handling make it worse?

Selection bias is the phenomenon of selecting individuals, groups or data for analysis in such a way that proper randomization is not achieved, ultimately resulting in a sample that is not representative of the population.

Understanding and identifying selection bias is important because it can significantly skew results and provide false insights about a particular population group.

Types of selection bias include:

- **sampling bias**: a biased sample caused by non-random sampling
- **time interval**: selecting a specific time frame that supports the desired conclusion. e.g. conducting a sales analysis near Christmas.
- **exposure**: includes clinical susceptibility bias, protopathic bias, indication bias.
- **data**: includes cherry-picking, suppressing evidence, and the fallacy of incomplete evidence.
- **attrition**: attrition bias is similar to survivorship bias, where only those that 'survived' a long process are included in an analysis, or failure bias, where those that 'failed' are only included
- **observer selection**: related to the Anthropic principle, which is a philosophical consideration that any data we collect about the universe is filtered by the fact that, in order for it to be observable, it must be compatible with the conscious and sapient life that observes it.

Handling missing data can make selection bias worse because different methods impact the data in different ways. For example, if you replace null values with the mean of the data, you adding bias in the sense that you're assuming that the data is not as spread out as it might actually be.

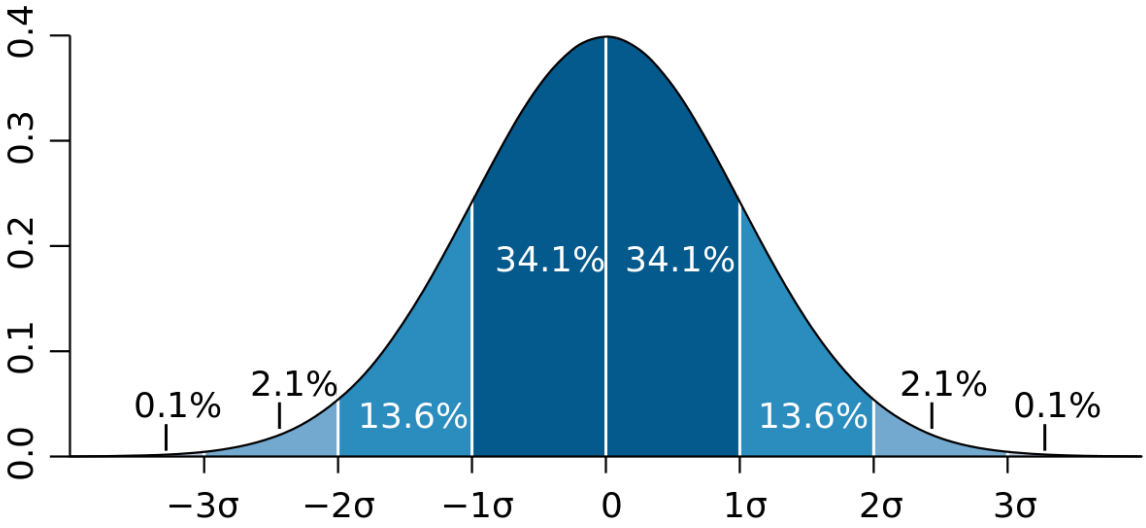
▼ What is an outlier? Explain how you might screen for outliers and what would you do if you found them in your dataset. Also, explain what an inlier is and how you might screen for them and what would you do if you found them in your dataset.

An **outlier** is a data point that differs significantly from other observations.

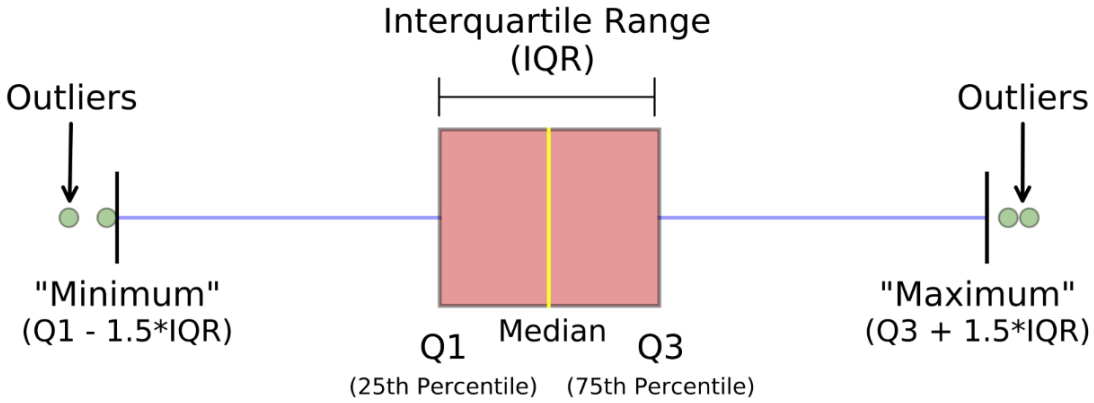
Depending on the cause of the outlier, they can be bad from a machine learning perspective because they can worsen the accuracy of a model. If the outlier is caused by a measurement error, it's important to remove them from the dataset. There are a couple of ways to identify outliers:

Z-score/standard deviations: if we know that 99.7% of data in a data set lie within three standard deviations, then we can calculate the size of one standard deviation, multiply it by 3, and identify the data points that are outside of this range. Likewise, we can calculate the z-score of a given point, and if it's equal to +/- 3, then it's an outlier. Note: that

there are a few contingencies that need to be considered when using this method; the data must be normally distributed, this is not applicable for small data sets and the presence of too many outliers can throw off z-score.



Interquartile Range (IQR): IQR, the concept used to build boxplots, can also be used to identify outliers. The IQR is equal to the difference between the 3rd quartile and the 1st quartile. You can then identify if a point is an outlier if it is less than Q1–1.5*IRQ or greater than Q3 + 1.5*IQR. This comes to approximately 2.698 standard deviations.



Other methods include DBScan clustering, Isolation Forests, and Robust Random Cut Forests.

An **inlier** is a data observation that lies within the rest of the dataset and is unusual or an error. Since it lies in the dataset, it is typically harder to identify than an outlier and requires external data to identify them. Should you identify any inliers, you can simply remove them from the dataset to address them.

▼ How do you handle missing data? What imputation techniques do you recommend?

There are several ways to handle missing data:

- Delete rows with missing data
- Mean/Median/Mode imputation
- Assigning a unique value
- Predicting the missing values
- Using an algorithm which supports missing values, like random forests

The best method is to delete rows with missing data as it ensures that no bias or variance is added or removed, and ultimately results in a robust and accurate model. However, this is only recommended if there's a lot of data to start with

and the percentage of missing values is low.

▼ Give an example where the median is a better measure than the mean

When there are a number of outliers that positively or negatively skew the data, such as wealth.

▼ Given two fair dices, what is the probability of getting scores that sum to 4? to 8?

There are 4 combinations of rolling a 4 (1+3, 3+1, 2+2):

- P(rolling a 4) = 3/36 = 1/12

There are combinations of rolling an 8 (2+6, 6+2, 3+5, 5+3, 4+4):

- P(rolling an 8) = 5/36

▼ What is the Law of Large Numbers?

The Law of Large Numbers is a theory that states that as the number of trials increases, the average of the result will become closer to the expected value.

Eg. flipping heads from fair coin 100,000 times should be closer to 0.5 than 100 times.

▼ How do you calculate the needed sample size?

You can use the margin of error (ME) formula to determine the desired sample size.

- t/z = t/z score used to calculate the confidence interval
- ME = the desired margin of error
- S = sample standard deviation
- n = sample size
- σ = population standard deviation

$$ME = t * \frac{S}{\sqrt{n}} \ or \ z * \frac{\sigma}{\sqrt{n}}$$

▼ What is A/B testing?

A/B testing is a form of hypothesis testing and two-sample hypothesis testing to compare two versions, the control and variant, of a single variable. It is commonly used to improve and optimize user experience and marketing.

▼ What is p-value?

When you perform a hypothesis test in statistics, a p-value can help you determine the strength of your results. p-value is a number between 0 and 1. Based on the value it will denote the strength of the results. The claim which is on trial is called Null Hypothesis.

Low p-value (≤ 0.05) indicates strength against the null hypothesis which means we can reject the null Hypothesis. High p-value (≥ 0.05) indicates strength for the null hypothesis which means we can accept the null Hypothesis p-value of 0.05 indicates the Hypothesis could go either way. To put it in another way,

High P values: your data are likely with a true null. Low P values: your data are unlikely with a true null.

▼ How do you prove that males are on average taller than females by knowing just gender height?

You can use hypothesis testing to prove that males are taller on average than females.

The null hypothesis would state that males and females are the same height on average, while the alternative hypothesis would state that the average height of males is greater than the average height of females.

Then you would collect a random sample of heights of males and females and use a t-test to determine if you reject the null or not.

▼ Infection rates at a hospital above a 1 infection per 100 person-days at risk are considered high. A hospital had 10 infections over the last 1787 person-days at risk. Give the p-value of the correct one-sided test of whether the hospital is below the standard.

Since we looking at the number of events (# of infections) occurring within a given timeframe, this is a Poisson distribution question.

$$P(k\ events\ in\ interval) = \frac{\lambda^k e^{-\lambda}}{k!}$$

The probability of observing k events in an interval

Null (H0): 1 infection per person-days

Alternative (H1): >1 infection per person-days

k (actual) = 10 infections

lambda (theoretical) = (1/100)*1787

p = 0.032372 or 3.2372%

Since p-value < alpha (assuming 5% level of significance), we reject the null and conclude that the hospital is below the standard.

▼ You roll a biased coin (p(head)=0.8) five times. What's the probability of getting three or more heads?

$$P(k\ out\ of\ n) = \frac{n!}{k!(n-k)!} * p^k(1-p)^{(n-k)}$$

- p = 0.8
- n = 5
- k = 3,4,5
- P(3 or more heads) = P(3 heads) + P(4 heads) + P(5 heads) = **0.94 or 94%**

▼ An HIV test has a sensitivity of 99.7% and a specificity of 98.5%. A subject from a population of prevalence 0.1% receives a positive test result. What is the precision of the test (i.e the probability he is HIV positive)?

$$Precision(PV) = \frac{Prevalence * Sensitivity}{(Prevalence * Sensitivity) + ((1 - Prevalence) * (1 - Specificity)}$$

- Precision = Positive Predictive Value = PV
- PV = (0.001*0.997)/[(0.001*0.997)+((1–0.001)*(1–0.985))]
- PV = 0.0624 or 6.24%

▼ You are running for office and your pollster polled hundred people. Sixty of them claimed they will vote for you. Can you relax?

- Assume that there's only you and one other opponent.
- Also, assume that we want a 95% confidence interval. This gives us a z-score of 1.96.

$$\hat{p} \pm z * \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Confidence interval formula

p-hat = 60/100 = 0.6

z* = 1.96

n = 100

This gives us a confidence interval of [50.4,69.6]. Therefore, given a confidence interval of 95%, if you are okay with the worst scenario of tying then you can relax. Otherwise, you cannot relax until you got 61 out of 100 to claim yes.

▼ Geiger counter records 100 radioactive decays in 5 minutes. Find an approximate 95% interval for the number of decays per hour

- Since this is a Poisson distribution question, mean = lambda = variance, which also means that standard deviation = square root of the mean
- a 95% confidence interval implies a z score of 1.96

- one standard deviation = 10

Therefore the confidence interval = 100 +/- 19.6 = [964.8, 1435.2]

▼ The homicide rate in Scotland fell last year to 99 from 115 the year before. Is this reported change really noteworthy?

- Since this is a Poisson distribution question, mean = lambda = variance, which also means that standard deviation = square root of the mean
- a 95% confidence interval implies a z score of 1.96
- one standard deviation = sqrt(115) = 10.724

Therefore the confidence interval = 115+/- 21.45 = [93.55, 136.45]. Since 99 is within this confidence interval, we can assume that this change is not very noteworthy.

▼ Consider influenza epidemics for two-parent heterosexual families. Suppose that the probability is 17% that at least one of the parents has contracted the disease. The probability that the father has contracted influenza is 12% while the probability that both the mother and father have contracted the disease is 6%. What is the probability that the mother has contracted influenza?

Using the General Addition Rule in probability:

P(mother or father) = P(mother) + P(father) — P(mother and father)

P(mother) = P(mother or father) + P(mother and father) — P(father)

P(mother) = 0.17 + 0.06–0.12

P(mother) = 0.11

▼ Suppose that diastolic blood pressures (DBPs) for men aged 35–44 are normally distributed with a mean of 80 (mm Hg) and a standard deviation of 10. About what is the probability that a random 35–44 year old has a DBP less than 70?

Since 70 is one standard deviation below the mean, take the area of the Gaussian distribution to the left of one standard deviation.

= 2.3 + 13.6 = 15.9%

▼ In a population of interest, a sample of 9 men yielded a sample average brain volume of 1,100cc and a standard deviation of 30cc. What is a 95% Student's T confidence interval for the mean brain volume in this new population?

$$\bar{X} \pm t\frac{s}{\sqrt{n}}$$

Given a confidence level of 95% and degrees of freedom equal to 8, the t-score = 2.306

Confidence interval = 1100 +/- 2.306*(30/3)

Confidence interval = [1076.94, 1123.06]

▼ A diet pill is given to 9 subjects over six weeks. The average difference in weight (follow up — baseline) is -2 pounds. What would the standard deviation of the difference in weight have to be for the upper endpoint of the 95% T confidence interval to touch 0?

Upper bound = mean + t-score*(standard deviation/sqrt(sample size))

0 = -2 + 2.306*(s/3)

2 = 2.306 * s / 3

s = 2.601903

Therefore the standard deviation would have to be at least approximately 2.60 for the upper bound of the 95% T confidence interval to touch 0.

▼ Given the following statistic, what is the probability that a woman has cancer if she has a positive mammogram result? 1% of women have breast cancer, 90% of women who have breast cancer test positive on mammograms and 8% of women will have false positives.

Step 1: Assign events to A or X. You want to know what a woman's probability of having cancer is, given a positive mammogram. For this problem, actually having cancer is A and a positive test result is X.

Step 2: List out the parts of the equation (this makes it easier to work the actual equation): P(A)=0.01 P(~A)=0.99 P(X|A)=0.9 P(X|~A)=0.08

Step 3: Insert the parts into the equation and solve. Note that as this is a medical test, we're using the form of the equation from example #2:

(0.9 * 0.01) / ((0.9 * 0.01) + (0.08 * 0.99) = 0.10.

The probability of a woman having cancer, given a positive test result, is 10%.

▼ A jar consists of 21 sweets, 12 are green and 9 are blue. William picked two sweets at random, find the probability that: 1.both sweets are blue. 2. one sweet is blue and one sweet is green. 3. A third sweet is randomly chosen. Find the probability that at least one of the sweet is blue?

1. P(both sweets are blue) = P(B, B) = $\frac{9}{21} \times \frac{8}{20} = \frac{6}{35}$
2. P(one sweet is blue and one sweet is green) = P(G, B) or P(B, G) = $\frac{9}{35} + \frac{9}{35} = \frac{18}{35}$
3. P(at least 1 sweet is blue) = 1 – P(all three sweets are green) = 1 - $\frac{22}{133} = \frac{111}{133}$

Practical Experience

▼ What languages you are most comfortable with for developing AI algorithms?

- Python?
- C++
- Golang

▼ What libraries have you used for Deep Learning?

- Tensorflow
- Pytorch
- Apache MXNet
- Theano
- Caffe
- CNTK
- Lasagne

▼ Have you worked with CUDA directly?

- Yes or no. Be able to describe the reason that you worked with it directly if you have.

▼ What is the best academic paper you read in the last year?

A Distributed Multi-Sensor Machine Learning Approach to Earthquake Early Warning, by Kévin Fauvel, Daniel Balouek-Thomert, Diego Melgar, Pedro Silva, Anthony Simonet, Gabriel Antoniu, Alexandru Costan, Véronique Masson, Manish Parashar, Ivan Rodero, and Alexandre Termier

▼ What is your favorite machine learning library?

PyTorch's ease of use combined with the default eager execution mode for easier debugging predestines it to be used for fast models and is the reason that I prefer it over other libraries.

▼ How long have you been working on machine learning problems?

X years. I have Y years as a student as my course was focused on machine learning problems. I also have experience working in industry as for the last Z years I have been working for XYZ company.

▼ How many academic papers have you had published?

X and be able to explain the research questions that you were hoping to answer in each paper if you have published papers.

▼ What academic researcher do you read a lot of papers from?

Geoffrey Hinton

▼ How much experience do you have writing code?

X years. I wrote a lot of code while in college, you can check out my Gitub. Since moving to industry I have worked on a number of different code projects, so I have really improved my skills in this area in recent years and am now able to deploy production ready models at a large scale.

▼ Describe your process for writing a software program from start to finish.

There are five main ingredients in the programming process:

1. Defining the **problem**.
2. **Planning** the solution.
3. Coding the program.
4. **Testing** the program.
5. Documenting the program.

Big Data Technologies

▼ Compare Hadoop and Spark

Aa Criteria	Hadoop	Spark
Dedicated storage	HDFS	None
Speed of processing	Average	Excellent
Libraries	Separate tools available	Spark Core, SQL, Streaming, MLlib, GraphX

▼ How is Hadoop different from other parallel computing systems?

Hadoop is a distributed file system, which lets you store and handle massive amount of data on a cloud of machines, handling data redundancy. Go through this HDFS content to know how the distributed file system works. The primary benefit is that since data is stored in several nodes, it is better to process it in distributed manner. Each node can process the data stored on it instead of spending time in moving it over the network.

On the contrary, in Relational database computing system, you can query data in real-time, but it is not efficient to store data in tables, records and columns when the data is huge.

▼ What modes can Hadoop be run in?

Hadoop can run in three modes:

- **Standalone Mode:** Default mode of Hadoop, it uses local file stystem for input and output operations. This mode is mainly used for debugging purpose, and it does not support the use of HDFS. Further, in this mode, there is no custom configuration required for mapred-site.xml, core-site.xml, hdfs-site.xml files. Much faster when compared to other modes.
- **Pseudo-Distributed Mode (Single Node Cluster):** In this case, you need configuration for all the three files mentioned above. In this case, all daemons are running on one node and thus, both Master and Slave node are the same.
- **Fully Distributed Mode (Multiple Cluster Node):** This is the production phase of Hadoop (what Hadoop is known for) where data is used and distributed across several nodes on a Hadoop cluster. Separate nodes are allotted as Master and Slave.

▼ Explain the major difference between HDFS block and InputSplit

- In simple terms, block is the physical representation of data while split is the logical representation of data present in the block. Split acts a s an intermediary between block and mapper. Suppose we have two blocks: **Block 1: ii nntteell Block 2: li ppaatt** Now, considering the map, it will read first block from ii till ll, but does not know how to process the second block at the same time. Here comes Split into play, which will form a logical group of Block1 and Block 2 as a single block.
- It then forms key-value pair using inputformat and records reader and sends map for further processing With inputsplit, if you have limited resources, you can increase the split size to limit the number of maps. For instance, if there are 10 blocks of 640MB (64MB each) and there are limited resources, you can assign ‘split size’ as 128MB. This will form a logical group of 128MB, with only 5 maps executing at a time.

- However, if the ‘split size’ property is set to false, whole file will form one inputsplit and is processed by single map, consuming more time when the file is bigger.

▼ Explain the difference between NameNode, Checkpoint NameNode and BackupNode.

- **NameNode** is the core of HDFS that manages the metadata – the information of what file maps to what block locations and what blocks are stored on what datanode. In simple terms, it’s the data about the data being stored. NameNode supports a directory tree-like structure consisting of all the files present in HDFS on a Hadoop cluster. It uses following files for namespace: fsimage file- It keeps track of the latest checkpoint of the namespace. edits file-It is a log of changes that have been made to the namespace since checkpoint.
- **Checkpoint NameNode** has the same directory structure as NameNode, and creates checkpoints for namespace at regular intervals by downloading the fsimage and edits file and margining them within the local directory. The new image after merging is then uploaded to NameNode. There is a similar node like Checkpoint, commonly known as Secondary Node, but it does not support the ‘upload to NameNode’ functionality.
- **Backup Node:** provides similar functionality as Checkpoint, enforcing synchronization with NameNode. It maintains an up-to-date in-memory copy of file system namespace and doesn’t require getting hold of changes after regular intervals. The backup node needs to save the current state in-memory to an image file to create a new checkpoint.

▼ What are the most common Input Formats in Hadoop?

There are three most common input formats in Hadoop:

- **Text Input Format:** Default input format in Hadoop.
- **Key Value Input Format:** used for plain text files where the files are broken into lines
- **Sequence File Input Format:** used for reading files in sequence

▼ What are the core methods of a Reducer?

1. **setup():** this method is used for configuring various parameters like input data size, distributed cache. public void setup (context)
2. **reduce():** heart of the reducer always called once per key with the associated reduced task public void reduce(Key, Value, context)
3. **cleanup():** this method is called to clean temporary files, only once at the end of the task public void cleanup (context)

SQL

▼ Write a SQL query to get the second highest salary from the `Employee` table. If there is no second highest salary the query should return `null`.

```
+----+--------+
| Id | Salary |
+----+--------+
| 1  | 100    |
| 2  | 200    |
| 3  | 300    |
+----+--------+
```

This query says to choose the MAX salary that isn’t equal to the MAX salary, which is equivalent to saying to choose the second-highest salary!

```
SELECT MAX(salary) AS SecondHighestSalary
FROM Employee
WHERE salary != (SELECT MAX(salary) FROM Employee)
```

▼ Write a SQL query to find all duplicate emails in a table named `Person`

```
+----+---------+
| Id | Email   |
+----+---------+
| 1  | a@b.com |
| 2  | c@d.com |
```

```
| 3  | a@b.com |
+----+---------+
```

```
SELECT Email
FROM (
    SELECT Email, count(Email) AS count
    FROM Person
    GROUP BY Email
) as email_count
WHERE count > 1
```

```
SELECT Email
FROM Person
GROUP BY Email
HAVING count(Email) > 1
```

▼ Given a `Weather` table, write a SQL query to find all dates' Ids with higher temperature compared to its previous (yesterday's) dates.

```
+---------+------------------+------------------+
| Id(INT) | RecordDate(DATE) | Temperature(INT) |
+---------+------------------+------------------+
|       1 |       2015-01-01 |               10 |
|       2 |       2015-01-02 |               25 |
|       3 |       2015-01-03 |               20 |
|       4 |       2015-01-04 |               30 |
+---------+------------------+------------------+
```

- **DATEDIFF** calculates the difference between two dates and is used to make sure we're comparing today's temperature to yesterday's temperature.

In plain English, the query is saying, Select the Ids where the temperature on a given day is greater than the temperature yesterday.

```
SELECT DISTINCT a.Id
FROM Weather a, Weather b
WHERE a.Temperature > b.Temperature
AND DATEDIFF(a.Recorddate, b.Recorddate) = 1
```

▼ The `Employee` table holds all employees. Every employee has an Id, a salary, and there is also a column for the department Id. The `Department` table holds all departments of the company. Write a SQL query to find employees who have the highest salary in each of the departments. More information about the tables is in the toggle.

```
+----+-------+--------+--------------+
| Id | Name  | Salary | DepartmentId |
+----+-------+--------+--------------+
| 1  | Joe   | 70000  | 1            |
| 2  | Jim   | 90000  | 1            |
| 3  | Henry | 80000  | 2            |
| 4  | Sam   | 60000  | 2            |
| 5  | Max   | 90000  | 1            |
+----+-------+--------+--------------+
```

The `Department` table holds all departments of the company.

```
+----+----------+
| Id | Name     |
+----+----------+
| 1  | IT       |
| 2  | Sales    |
+----+----------+
```

Write a SQL query to find employees who have the highest salary in each of the departments. For the above tables, your SQL query should return the following rows (order of rows does not matter).

```
+------------+----------+--------+
| Department | Employee | Salary |
+------------+----------+--------+
| IT         | Max      | 90000  |
| IT         | Jim      | 90000  |
| Sales      | Henry    | 80000  |
+------------+----------+--------+
```

SOLUTION: IN Clause

- The **IN** clause allows you to use multiple OR clauses in a WHERE statement. For example WHERE country = 'Canada' or country = 'USA' is the same as WHERE country IN ('Canada', 'USA').
- In this case, we want to filter the Department table to only show the highest Salary per Department (i.e. DepartmentId). Then we can join the two tables WHERE the DepartmentId and Salary is in the filtered Department table.

```
SELECT
    Department.name AS 'Department',
    Employee.name AS 'Employee',
    Salary
FROM Employee
INNER JOIN Department ON Employee.DepartmentId = Department.Id
WHERE (DepartmentId , Salary)
    IN
    (   SELECT
            DepartmentId, MAX(Salary)
        FROM
            Employee
        GROUP BY DepartmentId
 )
```

▼ Mary is a teacher in a middle school and she has a table `seat` storing students' names and their corresponding seat ids. The column **id** is a continuous increment. Mary wants to change seats for the adjacent students. *Can you write a SQL query to output the result for Mary?*

```
+---------+---------+
|    id   | student |
+---------+---------+
|    1    | Abbot   |
|    2    | Doris   |
|    3    | Emerson |
|    4    | Green   |
|    5    | Jeames  |
+---------+---------+
```

For the sample input, the output is:

```
+---------+---------+
|    id   | student |
+---------+---------+
|    1    | Doris   |
|    2    | Abbot   |
|    3    | Green   |
|    4    | Emerson |
|    5    | Jeames  |
+---------+---------+
```

Note:*If the number of students is odd, there is no need to change the last one's seat.*

SOLUTION: CASE WHEN

- Think of a CASE WHEN THEN statement like an IF statement in coding.
- The first WHEN statement checks to see if there's an odd number of rows, and if there is, ensure that the id number does not change.
- The second WHEN statement adds 1 to each id (eg. 1,3,5 becomes 2,4,6)

- Similarly, the third WHEN statement subtracts 1 to each id (2,4,6 becomes 1,3,5)

```
SELECT
    CASE
        WHEN((SELECT MAX(id) FROM seat)%2 = 1) AND id = (SELECT MAX(id) FROM seat) THEN id
        WHEN id%2 = 1 THEN id + 1
        ELSE id - 1
    END AS id, student
FROM seat
ORDER BY id
```

Deep Learning Libraries

▼ What are the differences between Pytorch and Tensorflow?

PyTorch	TensorFlow
Pytorch is closely related to the Lua-based Torch framework, which is used on Facebook.	**TensorFlow is developed by Google and actively used at Google.**
Pytorch is new compared to other competitive Technologies.	**TensorFlow is not new and is a to-go tool by many researchers and industry professionals.**
Pytorch includes everything imperatively and dynamically.	**TensorFlow has static and dynamic graphs as a combination.**
PyTorch includes Computation graph during runtime.	**TensorFlow does not have any runtime option.**
PyTorch includes deployment highlight for mobile and embedded frameworks.	**TensorFlow works better for embedded frameworks.**

▼ What are tensors in Pytorch?

A Tensor is a multi-dimensional matrix containing elements of a single data type. Tensors in PyTorch are same as NumPy array. Its manually compute the forward pass, loss, and backward pass. The most significant difference between the PyTorch Tensors and NumPy Array is that Pytorch can run either in CPU or GPU. To run operations on GPU, just cast the sensor in the file system.

▼ What are the 3 levels of abstraction in Pytorch?

- Tensor - Imperative n-dimensional Array which runs on GPU.
- Variable - Node in the computational graph. This stores data and gradient.
- Module - Neural network layer will store state otherwise learnable weights.

▼ What is nn Module in PyTorch?

The nn package define a set of modules, which are thought of as a neural network layer that produce output from the input and have some trainable weights. It is a type of tensor that considers a module parameter. Parameters are tensors subclasses. A fully connected ReLU networks where one hidden layer, trained to predict y from x to minimizing the square distance.

```
import torch
# define model
model= torch.nn.Sequential(
torch.nn.Linear(hidden_num_units, hidden_num_units),
torch.nn.ReLU( ),
torch.nn.Linear(hidden_num_units, output_num_units),
)
loss_fn= torch.nn.crossEntropyLoss( )
```

▼ What are tensors in PyTorch?

A Tensor is a multi-dimensional matrix containing elements of a single data type. Tensors in PyTorch are same as NumPy array. Its manually compute the forward pass, loss, and backward pass. The most significant difference between the PyTorch Tensors and NumPy Array is that Pytorch can run either in CPU or GPU. To run operations on GPU, just cast the sensor in the file system.

```
# import PyTorch
import torch
# define a tensor
torch.floatTensor([2])
2
Torch. float tensor of size 1

# import PyTorch
import torch
# define a tensor
torch.floatTensor([2])
2
Torch. float tensor of size 1
```

▼ What is Autograd module in PyTorch?

There is an automatic differentiation technique used in PyTorch. This technique is more powerful when we are building a neural network. There is a recorder which records what operations we have performed, and then it replays it backs to compute our gradient.

▼ What is the Optim Module in PyTorch?

Torch.optim is a module that implements various optimization algorithm used for building neural networks. Most of the commonly used syntax is already supported.
Below is the code of Adam optimizer

```
Optimizer = torch.optim.Adam(mode1, parameters( ), lr=learning rate
```

▼ What is the use of torch.from_numpy()?

The torch.from_numpy() is one of the important property of torch which places an important role in tensor programming. It is used to create a tensor from numpy.ndarray. The ndarray and return tensor share the same memory. If we do any changes in the returned tensor, then it will reflect the ndaaray also.

▼ How do we find the derivatives of the function in PyTorch? The derivatives of the function are calculated with the help of the Gradient. There are four simple steps through which we can calculate derivative easily.

These steps are as follows:
Initialization of the function for which we will calculate the derivatives. Set the value of the variable which is used in the function. Compute the derivative of the function by using the backward () method. Print the value of the derivative using grad.

▼ Give any one difference between torch.nn and torch.nn.functional?

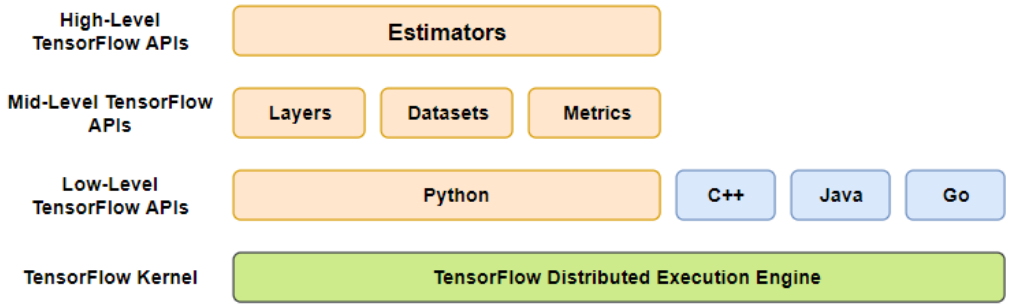
The torch.nn provide us many more classes and modules to implement and train the neural network. The torch.nn.functional contains some useful function like activation function and convolution operation, which we can use. However, these are not full layers, so if we want to define a layer of any kind, we have to use torch.nn.

▼ What is TensorBoard in Tensorflow?

TensorBoard is a suite of visualizing tools for inspecting and understanding TensorFlow runs and graphs. It is an easy solution to Tensorflow offered by the creators that let us visualize the graphs. It plots quantitative metrics about the graph with additional data like images to pass through it. TensorBoard currently supports five visualizations techniques such as **scalars**, **images**, **audio**, **histograms**, and **graphs**. It improves the accuracy and flow of graphs.

▼ Which client languages are supported in TensorFlow?

TensorFlow provides support for multiple client languages, one of the best among them is **Python**. There are some experimental interfaces which are available for C++, Java, and Go. A language bindings for many other languages such as **C#**, **Julia**, **Ruby**, and **Scala** are created and supported by the open-source community.



▼ Explain few options to load data into TensorFlow.

Loading the data into TensorFlow is the first step before training a machine learning algorithm. There are two ways to load the data:

- **Load data in memory:** It is the easiest method. All the data is loaded into memory as a single array. One can write a Python code which is unrelated to TensorFlow.
- **Tensorflow data pipeline:** TensorFlow has built-in APIs which help to load the data, perform the operations, and feed the machine learning algorithm easily. This method is mostly used when there is a large dataset.

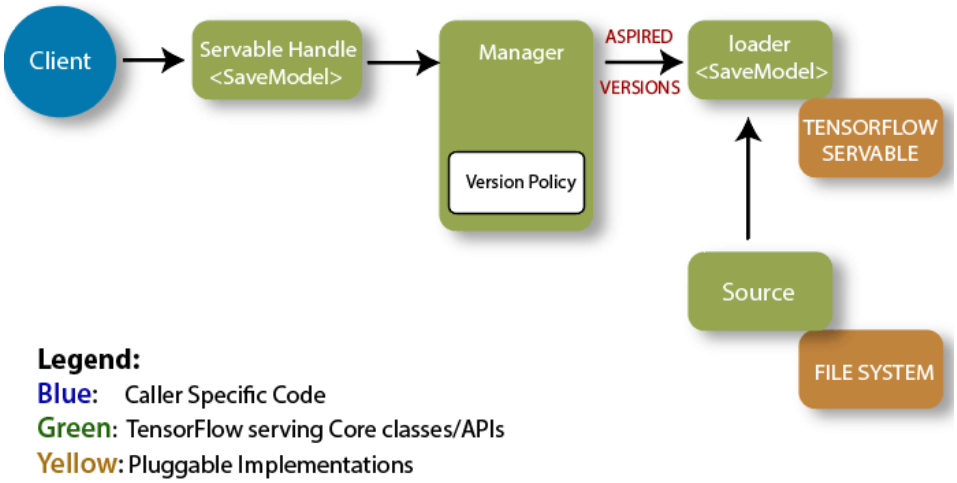
▼ Describe the common steps to most TensorFlow algorithms?

- Import data, generate data, or setting a data-pipeline through placeholders.
- Feed the data through the computational graph.
- Evaluate output on the loss function.
- Use backpropagation to modify the variables.
- Repeat until stopping condition.

▼ What are TensorFlow servables? Also, explain TensorFlow Serving.

The clients use some objects to perform the computations, and these objects are known as Servables. The size of the servable is flexible. A single servable might contain anything from a lookup table to a single model to a tuple of inference models. These servables are the central rudimentary units in TensorFlow Serving.

TensorFlow Serving is designed for production environments. It is a flexible, high-performance serving system used for machine learning models. TensorFlow Serving easily deploys new algorithms and experiments while keeping the same server architecture and APIs. TensorFlow Serving provides out-of-the-box integration with TensorFlow models. It can also be easily extended to serve other types of models and data whenever required.



▼ Differentiate between tf.variable and tf.placeholder.

The tf.variable and tf.placeholder both are almost similar to each other, but there are some differences as following:

tf.variable	tf.placeholder
It defines variable values which are modified with time.	• It defines specific input data that does not change with time.
It requires an initial value at the time of definition.	• It does not require an initial value at the time of definition.

▼ What are the different dashboards in TensorFlow?

There are different types of dashboards in TensorBoard which perform various tasks in the tensor board:

- Scalar Dashboard
- Histogram Dashboard
- Distributer Dashboard
- Image Dashboard
- Audio Dashboard
- Graph Explorer
- Projector
- Text Dashboard

▼ What are some statistical distribution functions provided by TensorFlow?

A wide variety of statistical distributions is available which is provided by TensorFlow and located inside:

> 💡 **tf.contrib.distributions**

It contains distributions like Beta, Bernoulli, Chi2, Dirichlet, Gamma, Uniform, etc. These are important building blocks when it comes to building machine learning algorithms, especially for probabilistic approaches like Bayesian models.

▼ What is the difference between Tensor.eval() and Session.run()?

In TensorFlow, we create graphs and provide values to that graph. The graph itself processes all the hardwork and generates the output based on the configuration that we have applied in the graph. Now, when we provide values to the graph, then first, we need to create a TensorFlow session.

> 💡 tf.Session()

Once the session is initialized, then we are supposed to use that session. It is necessary because all the variables and settings are now part of the session.

So, there are two possible ways that we can apply to pass external values to the graph so that the graph accepts them.

- The first one is to call the **.run()** while you are using the session and it is being executed.
- Another way to this is to use **.eval()**. The full syntax of .eval() is

> 💡 tf.get_default_session().run(values)

At the place of ***values.eval()***, we can put ***tf.get_default_session().run(values)*** and It will provide the same behavior. Here, eval is using the default session and then executing run().

▼ What do the TensorFlow managers do?

Tensorflow Managers handle the full lifecycle of Servables, including:

- Loading Servables
- Serving Servables
- Unloading Servables

Computer Science

▼ What is a recursive function?

A recursive function is one which calls itself, directly or calls a function that in turn calls it. Every recursive function follows the recursive properties − base criteria where functions stop calling itself and progressive approach where the functions try to meet the base criteria in each iteration. An important application of recursion in computer science is in defining dynamic data structures such as Lists and Trees.

▼ Give some examples greedy algorithms

- The below-given problems find their solution using greedy algorithm approach:
- Travelling Salesman Problem
- Prim’s Minimal Spanning Tree Algorithm
- Kruskal’s Minimal Spanning Tree Algorithm
- Dijkstra’s Minimal Spanning Tree Algorithm
- Graph – Map Coloring
- Graph – Vertex Cover
- Knapsack Problem
- Job Scheduling Problem

▼ What is binary search and how do you implement it?

A binary search works only on sorted lists or arrays. This search selects the middle which splits the entire list into two parts. First, the middle is compared. This search first compares the target value to the mid of the list. If it is not found, then it takes a decision on the weather. In computer science, binary search, also known as half-interval search, logarithmic search, or binary chop, is a search algorithm that finds the position of a target value within a sorted array.

```
# Returns index of x in arr if present, else -1
def binary_search(arr, low, high, x):

    # Check base case
    if high >= low:

        mid = (high + low) // 2

        # If element is present at the middle itself
        if arr[mid] == x:
            return mid

        # If element is smaller than mid, then it can only
        # be present in left subarray
        elif arr[mid] > x:
            return binary_search(arr, low, mid - 1, x)

        # Else the element can only be present in right subarray
        else:
            return binary_search(arr, mid + 1, high, x)

    else:
        # Element is not present in the array
        return -1

# Test array
arr = [ 2, 3, 4, 10, 40 ]
x = 10

# Function call
result = binary_search(arr, 0, len(arr)-1, x)

if result != -1:
    print("Element is present at index", str(result))
else:
    print("Element is not present in array")
```

▼ What is linear searching?

Linear search tries to find an item in a sequentially arranged data type. These sequentially arranged data items known as array or list, are accessible in incrementing memory location. Linear search compares expected data item with each of data items in list or array. The average case time complexity of linear search is O(n) and worst case complexity is O(n2). Data in target arrays/lists need not be sorted.

```
# Searching an element in a list/array in python
# can be simply done using \'in\' operator
# Example:
# if x in arr:
#   print arr.index(x)

# If you want to implement Linear Search in python

# Linearly search x in arr[]
# If x is present then return its location
# else return -1

def search(arr, x):

    for i in range(len(arr)):

        if arr[i] == x:
            return i

    return -1
```

▼ What is a stack and when do we use it?

In data-structure, a stack is an Abstract Data Type (ADT) used to store and retrieve values in Last In First Out method. The stack is the memory set aside as scratch space for a thread of execution.

▼ What operations can be performed on stacks?

The below operations can be performed on a stack:

- push() − adds an item to stack
- pop() − removes the top stack item
- peek() − gives a value of a top item without removing it
- isempty() − checks if a stack is empty
- isfull() − checks if a stack is full

▼ What is a linked-list and when do we use it?

A linked-list is a list of data-items connected with links i.e. pointers or references. Most modern high-level programming language does not provide the feature of directly accessing a memory location, therefore, linked-list is not supported in them or available in form of inbuilt functions. In computer science, a linked list is a linear collection of data elements, in which linear order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a group of nodes which together represent a sequence.

▼ What is the difference between Stack and Queue data structure?

One of the classical data structure interviews question. I guess every one know, No? Any way main difference is that Stack is LIFO(Last In First Out) data structure while Queue is a FIFO(First In First Out) data structure.

▼ What is difference between Singly Linked List and Doubly Linked List data structure?

This is another classical interview question on data structure, mostly asked on telephonic rounds. Main difference between singly linked list and doubly linked list is ability to traverse. In a single linked list, node only points towards next node, and there is no pointer to previous node, which means you can not traverse back on a singly linked list. On the other hand doubly linked list maintains two pointers, towards next and previous node, which allows you to navigate in both direction in any linked list.

▼ What is the average time complexity to search an unsorted array?

O(n)

https://www.bigocheatsheet.com/

Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

▼ What is the average time complexity of Quicksort?

O(n log(n))

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

Culture Fit

▼ General approach to answering cultural fit interview questions

STAR Interview Technique for Cultural Fit Interview Questions

Job interview questions about your behavior in the workplace are best answered by providing example situations according to the STAR method. This way, you can give interviewers exactly what they are looking for. Also, you can use this as an opportunity to provide a concise and to the point answer about how you behaved in previous work situations. Below the STAR acronym is broken down into each step.

Situation

When you give your answer to the interviewer, start by setting the stage. Provide context around the situation or challenge you were facing. Don't forget to provide relevant details.

Task

Secondly, talk about your specific responsibilities and what your role was. It's important that the interviewer gets an understanding of your task.

Action

After describing your task, talk about the actions you took to resolve the challenges you were facing. Provide the interviewer with a step by step description of what actions you took.

Result

Talk about the outcomes of your actions. Make sure to take credit for your behavior that led to the result. Here you answer questions such as What happened? What did you accomplish? Also, provide the interviewer with information about what you learned from the situation. **Make sure to focus on positive results and positive learning experiences.**

▼ Why do you want to work here?

There are several ways to answer this question. You can, for instance, include the reputation of the products of the company or their reputation as an employer. Think of answers such as:

'I believe in a collaborative approach to projects, and when I discovered this opportunity to join the product marketing team, I knew I had to apply. I have used your products for years, and I am very impressed with the innovations and consistent concern for helping your customers learn how to use them effectively. Teamwork environments always inspired me, and my background in sales and marketing has prepared me in the best way possible for this position. I would love to be a part of this innovative team and as an experienced marketer with an emphasis on tech products, I know I can add a lot of value to the team.'

Why is this a strong answer?

1. This answer shows that you have knowledge of the company and its products. In other words, you have done your research, and you're familiar with using their products as well.
2. The answer relates personal values and previous work experience to the job position that you're applying for.
3. The answer is convincing and logically structured. In a short and concise way, it demonstrates why you want to work there but also why your experience makes you the right person for the job.

▼ What appeals to you most about this position?

'I read an article on Bloomberg.com about how your new CEO Jack Johnson is working on implementing a new technology innovation plan. Your company always has been known to put a strong focus on innovation, and I would love to be part of an organization that's continuously striving to maintain its position as a leader in the market. This position would be a great fit for me, and a competitive team environment is a great place for me to apply my skills and to develop myself. The position matches my experience, and it would also allow me to take on greater responsibilities as well. Furthermore, your training and development programs sound very attractive to me to progress my career and knowledge-levels even further.'

Why is this a strong answer?

1. This answer shows not only that you have knowledge of the company; it also demonstrates that you're aware of the most recent developments. This will show that you're well-prepared and really interested in the position.
2. The answer indicates what appeals most to you about this position by mentioning specific details such as teamwork, training & development, and greater responsibilities. Also, complementing a company on its market position and achievements is a good idea but don't go too far with it to a point where it's too much.
3. The answer relates your experience to the job position that you're applying for.
4. In general, this answer demonstrates what appeals to you about the company and the position, and why you're the right person for the job.

▼ Do you rather work alone or in a team?

Now this question can be tricky and should be tailored to the position that you're applying for. If you're applying for a position that requires intensive teamwork, focus your answer towards collaborating with others. If the job requires a lot of individual work, focus your answer towards taking your responsibility on your individual tasks. Below an example is given of an answer that values both teamwork and individual work.

'My eight years of experience in different sales positions made me comfortable working within a team as well as working alone. A lot of sales meetings are one-on-one with clients who are good to discuss details discretely, but I definitely understand the value of teamwork too. Creative sessions within our sales teams really benefited my approach to sales strategy, setting targets, and general best practices. Also, having a team behind you can create greater confidence among the team members because there's always someone that can advise you in certain situations.'

Why is this a strong answer?

1. This answer demonstrates that you understand the value of both working alone and working on a team and that you gained this knowledge through experience.
2. The answer shows that you're flexible and that you can adapt to situations which is essential in the workplace.
3. The answer relates personal values and previous work experience to the job position that you're applying for.
4. In a short and concise way, the answer demonstrates why you want to work there but also why your experience makes you the right person for the job. The answer is convincing and logically structured.

▼ What's your approach to delegating tasks to employees, and how did you successfully do this in the past?

'What's your approach to delegating tasks to employees, and how did you successfully do this in the past?' When answering this question, you should take the position you're applying for into account. Make sure that you're management and leadership style align with those of the hiring organization. This way, you can demonstrate that you possess the right skills to successfully lead or manage a team.

'In my previous position, I was in charge of the sales department and was responsible for several smaller teams. At a certain time, we were invited to pitch to a new client for a long-term contract. As I was responsible for the success of this pitch, I understood that there was no room for error.

I composed a team of the most experienced employees and selected them based on their individual qualities and strengths to make sure to balance the team out. Together with the team, we made a planning and set goals and milestones to work on the pitch. After that, I delegated tasks based on the knowledge and experience levels of each team member. Also, I appointed a project manager to monitor the progress on a day to day basis and report to me on the milestone progress.

Because I distributed the responsibilities according to experience and knowledge levels, everyone on the team understood their responsibilities and the importance of the project. The team delivered everything on time without requiring intense oversight. We finished our pitch ahead of schedule and were able to provide the client with everything he asked for. The client told us that he was impressed by our efforts, and we landed the contract. This was a great achievement for the team as it was an effort that could not have been made without the people on it.'

Why is this a strong answer?

1. This answer shows multiple important elements, such as teamwork, leadership, creativity, and adaptability.
2. It's a logically structured answer according to the STAR method. Furthermore, it demonstrates your delegation skills.
3. The answer shows self-awareness but also great leadership by including your team and their efforts in the overall performance.
4. In a short and concise way, the answer shows how you delegate tasks to team members and based on what you made certain decisions.

▼ How do you handle conflicts in the workplace?

'My experience with conflict situations taught me that it's always good to try to see things from the other person's perspective and to approach the situation open-minded. By understanding the other person's perspective, you get a better feeling of how they really feel about things. This, in turn, gives you the opportunity to talk about how to reconcile different positions. This approach makes the situation less personal, which is a good way to start working from.

One time, in my previous position, a discussion started within the team about the budgets that needed to be allocated for the next quarter. The argument was about where to allocate the budgets in terms of teams and departments. Basically,

the team split up in two sides, and both sides thought they were right and really believed their priorities were correct. As it often goes during a discussion, the articulation and substantiation on why their priorities were that way, was not clear. Both teams made assumptions on the reasons behind each other's decisions. I tried to mediate the differences by asking specific questions to both sides to understand where they were coming from. Within 20 minutes, both teams were able to remove a great deal of the tension and started working on a constructive solution because they understood each other's logic behind their choices.'

Why is this a strong answer?

1. This answer directly addresses the question of how you handle conflicts by demonstrating your views and how you handled such a situation in the past.
2. The answer relates personal values and previous work experience to the job position that you're currently applying for.
3. The answer is convincing and logically structured. In a short and concise way, it demonstrates how you approach situations and how you solve conflicts.

▼ What gets you excited about coming to work?

▼ What was the last really great book you read?

▼ What surprises people about you?

▼ If you were going to start your own business, what would it be?

▼ What's the biggest problem in most offices today?

▼ What did you like most/least about your last company?

▼ Where/when/how do you do your best work?

▼ How would you describe your group of friends?

▼ Describe a time when you exceeded people's expectations.

▼ Why did you choose to apply here?

Questions for the Interviewer

▼ Can you tell me more about the day-to-day responsibilities of the role?

Asking this question enables you to learn as much about the role as possible. The interviewer's response will provide insight into what specific skills and experience are needed, and will also help you decide if the role is right for you.

The answer will give you an idea of what the employer's expectations are, so if you're offered the job there should be no surprises when you start.

▼ How could I impress you in the first three months?

This is a good question to ask at the end of a job interview because it shows potential employers that you're eager to make a positive contribution to the organisation.

Pay close attention to the recruiter's response as it will tell you how they want you to perform and will highlight particular areas of the job you should be focusing on during the first few weeks of employment.

▼ Are there opportunities for training and progression within the role/company?

Enquiring about development opportunities demonstrates to the interviewer that you're serious about your career and committed to a future with the organisation.

You don't want to be stuck in a dead-end job so if you're unsure of the typical career path for someone in this role, asking this question will help you to assess whether a long-term career with the company is a possibility, or if you'd need to move on to gain further responsibility.

▼ Where do you think the company is headed in the next five years?

The response you receive will give you an insight into the company's progression plans and its place in the market, while giving you a general idea about job security. You may also get a heads-up on any major upcoming projects.

Asking about future plans shows a real interest in the organisation and reiterates your commitment to the company.

▼ Can you describe the working culture of the organization?

Asking this question is a great way to assess the working environment of the company and it gives you the opportunity to discover whether you'll fit in.

From the recruiters response you'll learn if and how the organisation prioritises employee happiness, of any benefits on offer and what the work-life balance is like.

▼ What do you enjoy about your job?

Everybody loves to talk about themselves and this question enables you to build up a sense of camaraderie with your interviewer. This question requires a personal response, so you could learn a lot from their answer.

You'll get an insider's view of the company culture and working environment and you may even get to discover how your interviewer got their start in the business and how they progressed.

▼ Can you tell me more about the team I would be working in?

This will help you understand the way the company is structured, who you'll report to and the department the role sits within. These are the people you'll work most closely with, so it's worth trying to find out about the team dynamic and working methods.

Depending on the response, it may also give you the opportunity to mention any experience or success you've had working in similar teams - just to give the employer one final example of how well you'll fit in if you get the job.

Other useful questions to ask at interview include those about:

- performance appraisals
- opportunities or challenges facing the department/company
- company-specific projects or campaigns.

If the employer doesn't give an indication of what happens next then a good way to wrap up the interview is by asking about the next steps and when you can expect to hear from them.

▼ What does success look like in this position, and how do you measure it?

It’s crucial to have a deep understanding of how a company measures success. What are the key performance indicators (KPIs) for the role? How, and how often, are they measured?

▼ What challenges has this company faced in the last few years? What challenges do you anticipate in the coming years?

This is a great question if you’re interviewing with managers or senior leadership. It shows your interest in the performance of the company and can give you insight into the pain points they experience. If applicable, you can follow up their response by any experience you bring to the table that can help with these pain points/challenges.

▼ What changes or innovations in the industry are you most excited about?

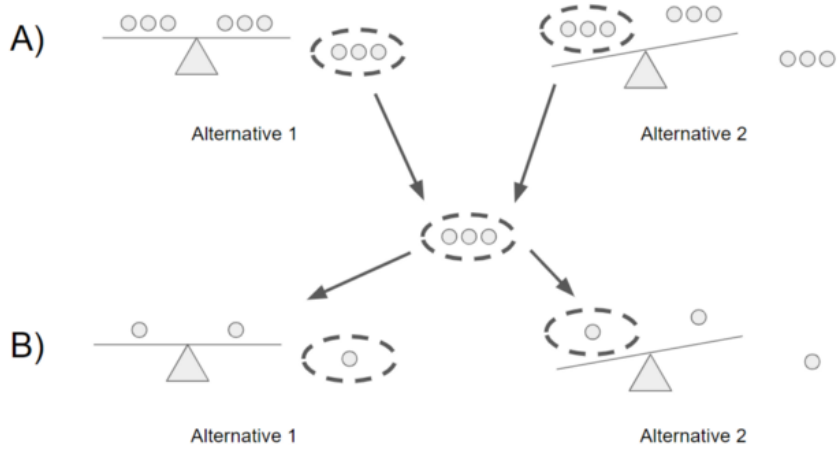
This question allows you to see how passionate the interviewer is about this company and industry. It also gives you the opportunity to follow up with what excited you the most about the industry during your research or through your past experience.

You can formulate next-level questions by asking about something that stems from what you read about the company in the news or on social media. For example:

- *“I noticed on your social media channels that you’ve opened several new offices lately. That kind of growth is exciting to me. It made me wonder what lines of the business are part of that expansion?”*
- *“I came across an interview with your CEO where she touched on several aspects of the company culture. What elements of the culture here do you like best?”*

Brainteasers

▼ If there are 8 marbles of equal weight and 1 marble that weighs a little bit more (for a total of 9 marbles), how many weighings are required to determine which marble is the heaviest?



Two weighings would be required (see part A and B above):

1. You would split the nine marbles into three groups of three and weigh two of the groups. If the scale balances (alternative 1), you know that the heavy marble is in the third group of marbles. Otherwise, you'll take the group that is weighed more heavily (alternative 2).
2. Then you would exercise the same step, but you'd have three groups of one marble instead of three groups of three.

▼ How can you generate a random number between 1 – 7 with only a die?

- Any die has six sides from 1-6. There is no way to get seven equal outcomes from a single rolling of a die. If we roll the die twice and consider the event of two rolls, we now have 36 different outcomes.
- To get our 7 equal outcomes we have to reduce this 36 to a number divisible by 7. We can thus consider only 35 outcomes and exclude the other one.
- A simple scenario can be to exclude the combination (6,6), i.e., to roll the die again if 6 appears twice.
- All the remaining combinations from (1,1) till (6,5) can be divided into 7 parts of 5 each. This way all the seven sets of outcomes are equally likely.