

Machine Learning: Data Science and ML Refresher

Shubhankar Agrawal

Abstract

This document serves as a quick refresher for Data Science and Machine Learning interviews. It covers mathematical and technical concepts across a range of algorithms. This requires the reader to have a foundational level knowledge with tertiary education in the field. This PDF contains material for revision over key concepts that are tested in interviews.

Contents

1	Machine Learning	1
1.1	Key Concepts	1
1.2	Data Preparation	1
	EDA • Data Cleaning • Feature Engineering • Dimensionality Reduction	
1.3	Regression	2
1.4	Classification	2
1.5	Ensembles	2
1.6	Metrics	2
1.7	Model Improvements	3
	Imbalanced Data • Regularization • Correlations	
2	Modern Methods	3
2.1	Deep Learning	3
	Activations • Optimizers • CNN • RNN • Attention	
2.2	Natural Language Processing	3
	Preprocessing • Embeddings • NER and POS Tagging • Topic Modelling	
2.3	Recommender Systems	4
	Collaborative Filtering • Association Rules • Ranking	
2.4	Reinforcement Learning	4
2.5	Advanced Topics	4
3	Real World Solutions	4
3.1	Retrieval Augmented Generation	4
3.2	Multi-Armed Bandit	4
3.3	Survival Analysis	4
4	Contact me	4

1. Machine Learning

1.1. Key Concepts

Table 1. Data Types

Data	Type	Description
Nominal	Categorical	No order
Ordinal	Categorical	Ordered
Interval	Numerical	Can be negative
Ratio	Numerical	Has a defined 0

Data Splits:

Train: Model learns from it.

Valid: Tune hyper-parameters, prevent over-fitting.

Test: Unseen data, evaluate performance.

Parameters: Weights the model learns

Hyper-parameters: Weights to adjust performance

Cross Validation: Expose all data (K Fold, LOOCV)

Table 2. Bias Variance Trade-Off

	Bias	Variance
What?	Error	Prediction Variability
Complexity	Too Simple	Too Complex
Fitting	Under	Over
Train Error	High	Low
Test Error	High	High
Formula	$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$	$\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$

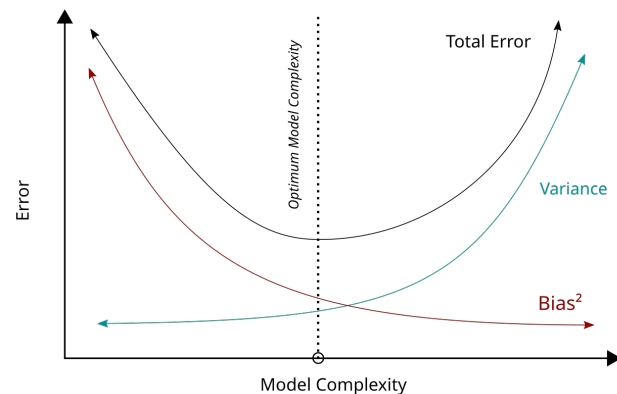


Figure 1. Bias Variance Trade-off [3]

1.2. Data Preparation

1.2.1. EDA

- Single / Multiple Variable Charts
- Correlations
- Outliers
- Investigate target

1.2.2. Data Cleaning

- Remove Outliers (Winsorization, Anomaly Detection)
- Fill Missing Values (Interpolate / Impute)

1.2.3. Feature Engineering

- Identify data types
- Generate lags, averages, aggregate features
- Encoding (Ordinal, One Hot, Target Encoding)
- Feature Pre-processing

1.2.4. Dimensionality Reduction

Feature space dimensionality can be reduced via feature selection methods:

- Correlation / ANOVA / Chi Square Tests
- Feature Importance (Random Forests)
- LASSO Regularization
- Forward / Backward Feature Selection
- Extraction with PCA / LDA / UMAP

Table 3. Pre-processing

Method	Formula
Standardization	$x_{\text{std}} = \frac{x - \mu}{\sigma}$
Normalization	$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$
Box Cox Transform	$y(\lambda) = \begin{cases} \frac{(x^\lambda - 1)}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$

Table 4. Dimensionality Reduction

	PCA	LDA	UMAP
What	Principal Components	Linear Discriminant	Uniform Manifold
How	Z-Score Eigenvectors	SSB/SSW Eigenvectors	Lower Dim Clusters
Supervised	No	Yes	Both

1.3. Regression

$$y = X\beta + \varepsilon \quad (1)$$

Ordinary Least Squares

$$\begin{aligned} \hat{\beta} &= \arg \min_{\beta} (y - X\beta)'(y - X\beta) \\ \hat{\beta} &= (X'X)^{-1}X'y \end{aligned} \quad (2)$$

Gradient Descent

Table 5. Terminology

Variable	Symbol	Description
Loss/Cost Function	J	Penalizes predictions
Learning Rate	α	Learning step size

$$\begin{aligned} \text{MSE}(J(\beta)) &= \frac{1}{2n} \sum_{i=1}^n (y_i - X_i\beta)^2 \\ \beta^{(t+1)} &= \beta^{(t)} - \alpha \nabla J(\beta^{(t)}) \\ \nabla J(\beta) &= -\frac{1}{n} X'(y - X\beta) \\ \beta^{(t+1)} &= \beta^{(t)} + \frac{\alpha}{n} X'(y - X\beta^{(t)}) \end{aligned} \quad (3)$$

Other Models: Ridge (L2), Lasso (L1), ElasticNet (Both) Regularizations

1.4. Classification

Some common classification methods:

Logistic Regression: Probabilistic model to separate classes.

$$\text{Log Loss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (4)$$

Support Vector Machine: Geometric model to split hyperplanes

$$\text{Hinge Loss} = \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i \hat{y}_i) \quad (5)$$

Decision Tree: Tree based structure to perform splits.

To use these models as regressors: aggregate predictions to get average value.

Table 6. Multi Class Classification

	Binary	Multiclass
Loss	Binary Log Loss	Cross Entropy
Activation	Sigmoid	Softmax

$$\text{Sigmoid} \sigma(x) = \frac{1}{1 + \exp(-x)}$$

$$\text{Cross-Entropy Loss} = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K y_{ik} \log(\hat{y}_{ik}) \quad (6)$$

$$\text{Softmax} \sigma(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^n \exp(z_j)}$$

1.5. Ensembles

Bagging, Boosting and Stacking

Table 7. Approaches

	Bagging	Boosting
Concept	Average models	Iterate models
Sample	Random	Weak predictors
Purpose	Reduce Variance	Reduce Bias
Base Model	High Variance	Low Variance
	Low Bias	High Bias
Examples	Random Forest	AdaBoost
		Gradient Boosting

LightGBM: Gradient Based One Sided Sampling, Exclusive Feature Bundling

1.6. Metrics

Regression Metrics

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

$$R^2_{\text{adj}} = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - p - 1} \right)$$

Classification Metrics

Table 8. Confusion Matrix

Real / Pred	True	False
True	True Positive (TP)	False Negative (FN)
False	False Positive (FP)	True Negative (TN)

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\
 \text{Precision} &= \frac{TP}{TP + FP} \\
 \text{Recall / Sensitivity (TPR)} &= \frac{TP}{TP + FN} \\
 \text{Specificity (TNR)} &= \frac{TN}{TN + FP} \\
 \text{F1 Score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \\
 F_\beta &= (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}
 \end{aligned} \tag{8}$$

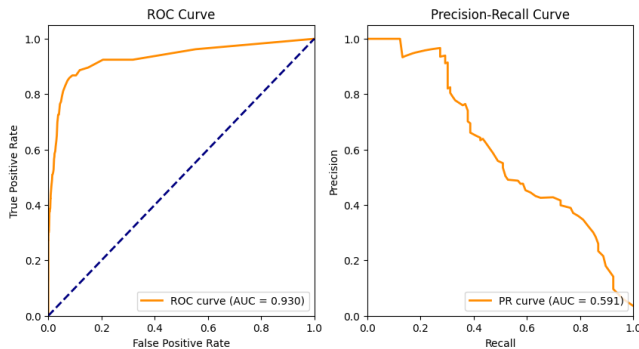


Figure 2. AUC ROC (and) AUC PR [2]

1.7. Model Improvements

1.7.1. Imbalanced Data

- Sample (Downsample / SMOTE - Up)
- Weighted costs and metrics
- Tree Based Methods (Robust)

1.7.2. Regularization

Control model complexity to prevent overfitting

- L1 - Reduce weights to 0
- L2 - Reduce weights to small values
- Reduce Tree Depth

1.7.3. Correlations

Remove correlated features by examining Variance Inflation Factor. Regress each variable on all others.

$$VIF(X_j) = \frac{1}{1 - R_j^2} \tag{9}$$

2. Modern Methods

2.1. Deep Learning

```

model.train(True)
optimizer.zero_grad()
outputs = model(inputs)
loss = loss_fn(outputs, labels)

loss.backward()
optimizer.step()

model.eval()
with torch.no_grad():
    outputs = model(inputs)

```

Code 1. PyTorch Model

2.1.1. Activations

Introduce Non-Linearity, summarized in Figure 3.

2.1.2. Optimizers

Table 9. Optimizers

Name	Description
Momentum	Past Gradient
RMSProp	Running Average of Squared Gradients
AdaGrad	Sum of Past Squared Gradients
ADAM	Momentum + RMSProp

2.1.3. CNN

Convolutional & Pooling Output Size

$$\text{Output Size} = \frac{(W - K + 2P)}{S} + 1 \tag{10}$$

where:

- W is the input size (width or height),
- K is the filter size,
- P is the padding applied,
- S is the stride of the convolution.

2.1.4. RNN

Common RNN architecture is summarized in Figure 4.

Long Short Term Memory: Input, Output, Forget Gate

Gated Recurrent Unit: Update, Reset Gate

2.1.5. Attention

- Input + Position Embeddings
- Query Key Value Multihead Attention
- Cross Attention
- Token Masking

The architecture is demonstrated in Figure 5.

BERT: Masked Language Modelling, Next Word Prediction

2.2. Natural Language Processing

2.2.1. Preprocessing

- Tokenization
- Stemming (vs) Lemmatization
- Stop Word Removal
- Handling punctuation, non Alphanumeric
- N-grams

2.2.2. Embeddings

- Bag of Words
- Text Frequency Inverse Document Frequency
- Word2Vec (CBOW and Skip Gram)
- GloVe (Matrix Factorization)
- Sentence Level Embeddings (SBERT)

TF-IDF

$$\begin{aligned}
 \text{TF}(t, d) &= \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \\
 \text{IDF}(t) &= \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right)
 \end{aligned}$$

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \text{IDF}(t)$$

(11)

2.2.3. NER and POS Tagging

- Calculate hashed embeddings
- Pass through LSTM / CNN
- Main stack going through attention model

2.2.4. Topic Modelling

- Conventional - Latent Dirichlet Allocation
- State of the Art - BERTopic

LDA: latent model distribution of document topics.

Table 10. Text Preprocessing Methods

Method	Purpose	False
True	True Positive (TP)	False Negative (FN)
False	False Positive (FP)	True Negative (TN)

2.3. Recommender Systems

Content Based (vs) Collaborative Filtering

2.3.1. Collaborative Filtering

Matrix Factorization: Split the interaction matrix into a User matrix and an Item matrix. Can be solved with Alternating Least Squares / Stochastic Gradient Descent.

Neural network based approaches:

- Neural Collaborative Filtering
- Wide and Deep Model
- Two Tower Approach

2.3.2. Association Rules

$$\begin{aligned}
 \text{Support}(A) &= \frac{\text{Number of transactions containing } A}{\text{Total number of transactions}} \\
 \text{Confidence}(A \rightarrow B) &= \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \\
 \text{Uplift}(A \rightarrow B) &= \frac{\text{Support}(A \cup B)}{\text{Support}(A) \times \text{Support}(B)}
 \end{aligned} \quad (12)$$

2.3.3. Ranking

Learning To Rank: Pairwise model (LambdaRank)

Normalized Discounted Cumulative Gain: Continuous Target

Mean Average Precision: Binary Target

$$\begin{aligned}
 \text{NDCG} &= \frac{\text{DCG}_p}{\text{IDCG}_p} \\
 \text{DCG}_p &= \sum_{i=1}^p \frac{\text{rel}_i}{\log_2(i+1)} \\
 \text{IDCG} &= \text{Ideal Ranking}
 \end{aligned} \quad (13)$$

$$\begin{aligned}
 \text{MAP} &= \frac{1}{N} \sum_{q=1}^N \frac{1}{m_q} \sum_{k=1}^{m_q} P(k) \\
 \text{MRR} &= \frac{1}{N} \sum_{i=1}^N \frac{1}{\text{rank}_i}
 \end{aligned} \quad (14)$$

2.4. Reinforcement Learning

Receptive environment based algorithm with feedback. Key equations are as follows.

Bellman Equation

$$V^*(s) = \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^*(s')] \quad (15)$$

Q Learning

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[R(s, a, s') + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \quad (16)$$

where:

- $V^*(s)$ is the optimal value function for state s ,
- a is the action,
- $P(s'|s, a)$ is the transition probability,
- $R(s, a, s')$ is the reward function,
- $Q(s, a)$ is the action-value function for state s and action a ,
- γ is the discount factor.

2.5. Advanced Topics

Siamese Networks: Train with two inputs simultaneously passed through the same network with a final evaluation function to compare similarities or classify into categories. SBERT works like this to get sentence embeddings.

Zero / One / Few Shot Learnings: Build model trained on comparing similarities between predictions. This can work on unseen data by checking if the prediction is close to the target. Natural Language Inference models work like this.

3. Real World Solutions

3.1. Retrieval Augmented Generation

- Embeddings of documents in a fast retrieval database
- Retrieve most similar documents to query
- Prompt LLM with document context and query

3.2. Multi-Armed Bandit

A more sophisticated version of A/B Testing. Maximize reward by trading off between exploration and exploitation.

Explore between various actions to understand reward functions and slowly switch to using action with maximized reward. Can be implemented with an epsilon greedy approach.

3.3. Survival Analysis

Analyzing the expected duration until an event has to occur.

Cox Proportional Hazards: Estimates the time to an event using other variables.

$$h(t|X) = h_0(t) \exp(\beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p) \quad (17)$$

where:

- $h(t|X)$ is the hazard function at time t given covariates X_1, X_2, \dots, X_p ,
- $h_0(t)$ is the baseline hazard function,
- $\beta_1, \beta_2, \dots, \beta_p$ are the coefficients for covariates X_1, X_2, \dots, X_p .

4. Contact me

You can contact me through these methods:

- 🔗 [Personal Website - astronights.github.io](https://astronights.github.io)
- ✉ shubhankar.31@gmail.com
- 🌐 linkedin.com/in/shubhankar-agrawal

References

- [1] *Activation Functions*. [Online]. Available: <https://medium.com/the-modern-scientist/an-overview-of-activation-functions-in-deep-learning-97a85ac00460>.
- [2] *Area Under Curves*. [Online]. Available: <https://juandelacalle.medium.com/how-and-why-i-switched-from-the-roc-curve-to-the-precision-recall-curve-to-analyze-my-imbalanced-6171da91c6b8>.







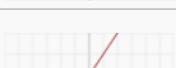


Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Figure 3. Activation Functions
[1]

- [3] *Bias Variance Tradeoff*. [Online]. Available: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff.
- [4] *Recurrent Neural Networks*. [Online]. Available: <https://www.linkedin.com/pulse/rnn-lstm-gru-why-do-we-need-them-suvankar-maity-joegc/>.
- [5] *Transformer Architecture*. [Online]. Available: <https://quantdare.com/transformers-is-attention-all-we-need-in-finance-part-i/>.

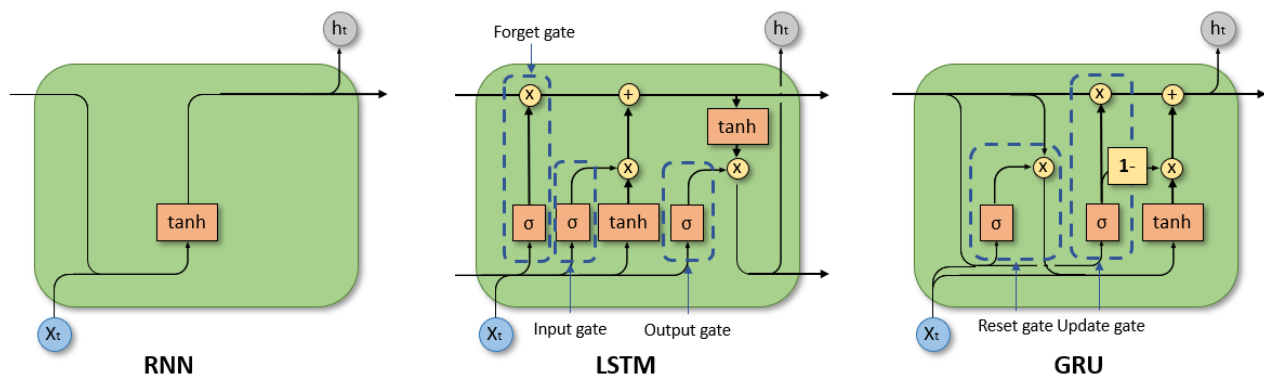


Figure 4. Recurrent Neural Networks
[4]

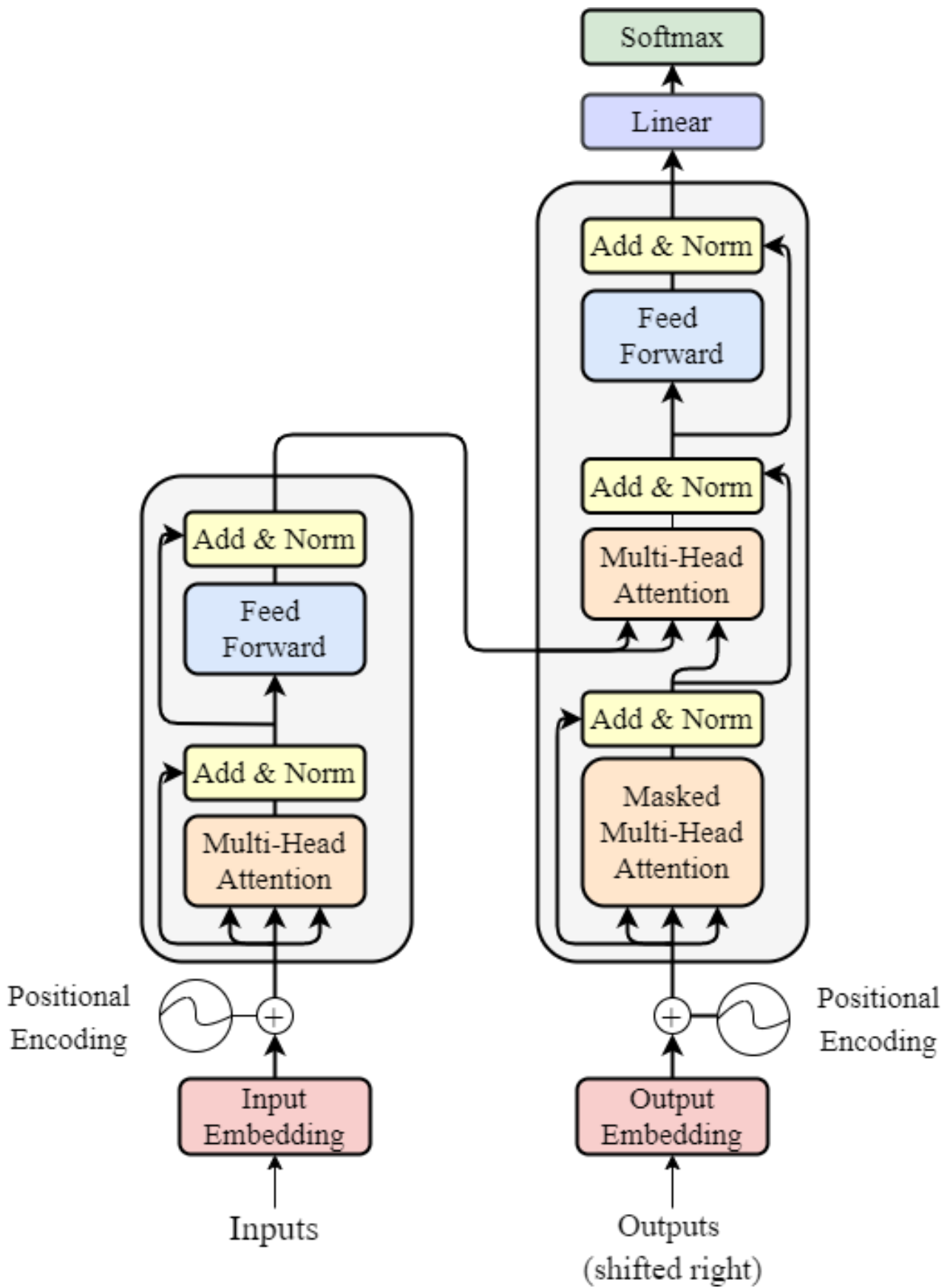


Figure 5. Attention Based Transformer
[5]