Moderntensor: Detailed Guide to Formulas in the Decentralized AI Platform

Moderntensor Development Team

March 24, 2025

Abstract

This document provides a comprehensive overview of the key formulas in the Moderntensor system, a decentralized AI platform built on the Cardano blockchain. The formulas cover incentive distribution, performance evaluation, penalty mechanisms, validator weight calculation, resource allocation, and additional mechanisms such as DAO voting power. Each formula is presented with clear mathematical notation, detailed explanations of its meaning, purpose, and illustrative examples.

1 Introduction

Moderntensor is a decentralized AI platform that leverages Cardano blockchain technology to train, validate, and reward AI models in a transparent and fair manner. This document focuses on the core formulas driving the system's incentive mechanisms, performance evaluation, penalties, and governance.

2 Incentive Distribution

2.1 Incentive for Miners

The reward for miners is calculated based on trust score, weight, and performance scores from validators:

$$\text{Incentive}_{\text{miner}}(x) = \text{trust}_{\text{score}}(x) \times \frac{\sum_{j=0}^{m}(W_x \times P_{xj})}{\sum_{i=0}^{n}\sum_{j=0}^{m}(W_i \times P_{ij})}$$

Meaning: This formula ensures that miners with high performance and good reliability receive fair rewards, based on their weight and validator evaluations.

Parameters:

- W_x: Weight of miner x (based on stake or performance).
- P_{xj}: Performance score of miner x evaluated by validator j.
- $\text{trust}_{\text{score}}(x)$: Historical trust score of miner x.

Example: Suppose miner x has $W_x = 2$, $\text{trust}_{\text{score}}(x) = 0.9$, and is evaluated by 3 validators: $P_{x1} = 0.8$, $P_{x2} = 0.9$, $P_{x3} = 0.7$. The total system $\sum_{i=0}^{n}\sum_{j=0}^{m}(W_i \times P_{ij})$ is 50. Calculate:

$$\text{Incentive}_{\text{miner}}(x) = 0.9 \times \frac{2 \times (0.8 + 0.9 + 0.7)}{50} = 0.9 \times \frac{2 \times 2.4}{50} = 0.0864$$

2.2 Incentive for Validators

Validators are rewarded based on their weight and performance score:

$$\text{Incentive}_{\text{validator}}(v) = \text{trust}_{\text{score}}(v) \times \frac{W_v \times E_v}{\sum_{u \in V}(W_u \times E_u)}$$

Meaning: This formula encourages validators to maintain high performance and reliability, with rewards proportional to their contribution.

Parameters:

- W_v: Weight of validator v.
- E_v: Performance score of validator v.
- $\text{trust}_{\text{score}}(v)$: Historical trust score of validator v.

Example: Validator v has $W_v = 3$, $E_v = 0.95$, $\text{trust}_{\text{score}}(v) = 0.85$, and the total system $\sum_{u \in V}(W_u \times E_u)$ is 60. Calculate:

$$\text{Incentive}_{\text{validator}}(v) = 0.85 \times \frac{3 \times 0.95}{60} = 0.85 \times \frac{2.85}{60} = 0.0404$$

3 Performance Evaluation

3.1 Task Completion Rate ($Q_{\textbf{task}}$)

The task completion rate is calculated with a weight prioritizing recent tasks:

$$Q_{\text{task}} = \frac{\sum_t(\text{task}_{\text{success},t} \times e^{-\delta(T-t)})}{\sum_t(\text{task}_t \times e^{-\delta(T-t)})}$$

Meaning: This formula measures performance based on the task completion rate, with an exponentially decreasing weight for older tasks via $e^{-\delta(T-t)}$.

Parameters:

- $\text{task}_{\text{success},t}$: Number of successfully completed tasks at time t.
- task_t: Total number of tasks at time t.
- T: Current time.
- δ: Decay constant (e.g., 0.5).

Example: A miner has data: $t = 1$: 8/10, $t = 2$: 9/10, $t = 3$: 10/10, $T = 3$, $\delta = 0.5$. Calculate:

- Numerator: $8 \times e^{-0.5\times2} + 9 \times e^{-0.5\times1} + 10 \times e^0 \approx 18.407$
- Denominator: $10 \times e^{-1} + 10 \times e^{-0.5} + 10 \times e^0 \approx 19.75$
- $Q_{\text{task}} = \frac{18.407}{19.75} \approx 0.932$ (93.2%).

3.2 Miner Approval Rate ($D_{\textbf{miner}}$)

The approval rate of a miner by validators:

$$D_{\text{miner}}(i) = \frac{\sum(\text{approved}_i \text{ by validators})}{\text{total number of checks for miner } i}$$

Meaning: Measures how frequently a miner's work is approved by validators.

Example: Miner i is checked 10 times, approved 8 times: $D_{\text{miner}}(i) = \frac{8}{10} = 0.8$ (80%).

3.3 Validator Performance Score ($E_{\text{validator}}$)

The validator's performance score combines multiple factors:

$$E_{\text{validator}} = \theta_1 \times Q_{\text{task}} + \theta_2 \times \text{accuracy}_{\text{validator}} + \theta_3 \times e^{-k \times \frac{|\text{Evaluation}-\text{Average Score}|}{\sigma}}$$

Meaning: Evaluates validators based on task completion rate, accuracy, and consensus level.

Parameters:
- $\theta_1, \theta_2, \theta_3$: Weight coefficients ($\theta_1 + \theta_2 + \theta_3 = 1$).
- $\text{accuracy}_{\text{validator}}$: Evaluation accuracy.
- σ: Standard deviation of evaluation scores.

Example: $Q_{\text{task}} = 0.9$, accuracy $= 0.85$, deviation $\frac{|\text{Eval}-\text{Avg}|}{\sigma} = 0.2$, $k = 1$, $\theta_1 = 0.4$, $\theta_2 = 0.3$, $\theta_3 = 0.3$. Calculate:

$$E_{\text{validator}} = 0.4 \times 0.9 + 0.3 \times 0.85 + 0.3 \times e^{-0.2} \approx 0.36 + 0.255 + 0.245 = 0.86$$

3.4 Miner Performance (P_{miner})

The performance of a miner is calculated based on the task completion rate, with a weight prioritizing recent tasks:

$$P_{\text{miner}} = \frac{\sum_t (\text{task}_{\text{success},t} \times e^{-\delta(T-t)})}{\sum_t (\text{task}_t \times e^{-\delta(T-t)})}$$

Meaning: This formula measures the miner's performance by considering the task completion rate, with an exponentially decreasing weight for older tasks via $e^{-\delta(T-t)}$.

Parameters:
- $\text{task}_{\text{success},t}$: Number of successfully completed tasks at time t.
- task_t: Total number of tasks at time t.
- T: Current time.
- δ: Decay constant (e.g., 0.5).

Example: A miner has data: $t = 1$: 8/10, $t = 2$: 9/10, $t = 3$: 10/10, $T = 3$, $\delta = 0.5$. Calculate:
- Numerator: $8 \times e^{-0.5\times2} + 9 \times e^{-0.5\times1} + 10 \times e^0 \approx 18.407$
- Denominator: $10 \times e^{-1} + 10 \times e^{-0.5} + 10 \times e^0 \approx 19.75$
- $P_{\text{miner}} = \frac{18.407}{19.75} \approx 0.932$ (93.2%).

3.5 Adjusted Miner Performance ($P_{\text{miner adjusted}}$)

The adjusted performance of a miner is calculated based on the trust scores of validators:

$$P_{\text{miner adjusted}} = \frac{\sum_v (\text{trust}_{\text{score}_v} \times P_{\text{miner},v})}{\sum_v \text{trust}_{\text{score}_v}}$$

Meaning: This formula adjusts the miner's performance score by weighting the evaluations from validators according to their trust scores, reducing the impact of unreliable validators.

Parameters:
- $\text{trust}_{\text{score}_v}$: Trust score of validator v.

- $P_{\text{miner},v}$: Performance score of the miner as evaluated by validator v.

Example: Miner M1 is evaluated by V1 with $P_{\text{miner},1} = 0.9$ and $\text{trust}_{\text{score}_1} = 0.8$, and by V2 with $P_{\text{miner},2} = 0.7$ and $\text{trust}_{\text{score}_2} = 0.5$. Calculate:

$$P_{\text{miner adjusted}} = \frac{0.8 \times 0.9 + 0.5 \times 0.7}{0.8 + 0.5} = \frac{0.72 + 0.35}{1.3} \approx 0.823$$

3.6 Miner Weight (W_x)

The weight of a miner is calculated based on historical performance, with a weight prioritizing recent values:

$$W_x = \sum_t P_{\text{miner},t} \times e^{-\delta(T-t)}$$

Meaning: This formula aggregates the miner's historical performance, encouraging consistent high performance as recent performance has a greater impact.

Parameters:

- $P_{\text{miner},t}$: Miner's performance at time t.
- T: Current time.
- δ: Decay constant (e.g., 0.5).

Example: A miner has performance: $t = 1$: 0.8, $t = 2$: 0.9, $t = 3$: 1.0, $T = 3$, $\delta = 0.5$. Calculate:

$$W_x = 0.8 \times e^{-0.5\times2} + 0.9 \times e^{-0.5\times1} + 1.0 \times e^0 \approx 0.294 + 0.546 + 1.0 = 1.84$$

4 Trust Score and Fairness Mechanisms

4.1 Trust Score Update for Miners

The trust score of a miner is updated with a decay factor for inactivity:

$$\text{Trust Score}_{\text{new}} = \text{Trust Score}_{\text{old}} \times e^{-\delta\times\text{time since last evaluation}} + \alpha \times \text{Score}_{\text{new}}$$

Meaning: This formula reduces the trust score over time if the miner is not selected, encouraging continuous participation.

Parameters:

- δ: Decay constant (e.g., 0.1).
- α: Learning rate (e.g., 0.1).
- time since last evaluation: Number of cycles since the last evaluation.

Example: Miner M5 has $\text{Trust Score}_{\text{old}} = 0.5$, not selected for 2 cycles, $\delta = 0.1$, time since last evaluation $= 2$, and $\text{Score}_{\text{new}} = 0$ (since not selected). Calculate:

$$\text{Trust Score}_{\text{new}} = 0.5 \times e^{-0.1\times2} + 0 = 0.5 \times 0.8187 \approx 0.409$$

4.2 Miner Selection Probability

The probability of selecting a miner is increased for those who have not been selected recently:

$$\text{Selection Probability} = \text{trust score} \times (1 + \beta \times \text{time since last selection})$$

Meaning: This formula increases the chances of selection for miners who have not been selected for a while, ensuring fairness.

Parameters: - β: Bonus factor (e.g., 0.2). - time since last selection: Number of cycles since the last selection.

Example: Miner M5 has trust score = 0.409, not selected for 2 cycles, $\beta = 0.2$. Calculate:

$$\text{Selection Probability} = 0.409 \times (1 + 0.2 \times 2) = 0.409 \times 1.4 = 0.5726$$

5 Penalty Mechanism

5.1 Performance Adjustment

Validators can recover performance over time:

$$P_{\text{adjustor\_new}} = E_{\text{validator\_new}} + \gamma(E_{\text{validator\_base}} - E_{\text{validator\_new}})$$

Meaning: Allows validators to improve their performance score but limits it to a baseline level.

Parameters: - $E_{\text{validator\_base}}$: Baseline score. - γ: Recovery rate (e.g., 0.1).

Example: $E_{\text{new}} = 0.7$, $E_{\text{base}} = 0.9$, $\gamma = 0.1$:

$$P_{\text{adjustor\_new}} = 0.7 + 0.1 \times (0.9 - 0.7) = 0.72$$

5.2 Stake Slashing

Stake is reduced upon detection of severe fraud:

$$\text{Slash}_{\text{amount}} = \min(0.2 \times \text{stake}, \text{fraud severity} \times \text{stake})$$

Meaning: Deters fraudulent behavior by reducing stake.

Example: stake = 1000, fraud severity = 0.15:

$$\text{Slash}_{\text{amount}} = \min(0.2 \times 1000, 0.15 \times 1000) = \min(200, 150) = 150$$

5.3 Fraud Detection Threshold

A fraud flag is triggered if evaluations deviate excessively:

$$\text{Fraud Flag} = \begin{cases} 1 & \text{if } |\text{Evaluation} - \text{Average Score}| > 0.5 \text{ for 3 cycles} \\ 0 & \text{otherwise} \end{cases}$$

Meaning: Detects validators with consistently abnormal behavior.

Example: Evaluations deviate > 0.5 for 3 consecutive cycles: Fraud Flag = 1.

5.4 Trust Score Update

Trust score is updated based on fraud:

$$\text{trust}_{\text{score\_new}} = \text{trust}_{\text{score\_old}} \times (1 - \eta \times \text{Fraud Flag})$$

Meaning: Reduces the trust score of validators flagged for fraud.
Parameters: η: Penalty coefficient (e.g., 0.1).
Example: $\text{trust}_{\text{score\_old}} = 0.9$, Fraud Flag $= 1$, $\eta = 0.1$:

$$\text{trust}_{\text{score\_new}} = 0.9 \times (1 - 0.1 \times 1) = 0.81$$

6 Validator Weight Calculation

6.1 Validator Weight

Validator weight is based on stake, performance, and participation time:

$$W_{\text{validator}} = \lambda \times \frac{\text{stake}}{\sum \text{stake}} + (1 - \lambda) \times E_{\text{validator}} \times (1 + \log(\text{time participated}))$$

Meaning: Combines stake and performance, rewarding long-term participants.
Parameters:
- λ: Balancing coefficient (e.g., 0.5).
- $\log(\text{time participated})$: Reward for participation duration.
Example: stake $= 500$, $\sum$ stake $= 2000$, $E_{\text{validator}} = 0.9$, time $= 10$, $\lambda = 0.5$:

$$W_{\text{validator}} = 0.5 \times \frac{500}{2000} + 0.5 \times 0.9 \times (1 + \log(10)) \approx 0.125 + 0.585 = 0.71$$

7 Resource Allocation

7.1 Subnet Resource Distribution

Resources are allocated based on weight and performance:

$$R_{\text{subnet}} = \frac{\sum_{i \in \text{subnet}} (W_i \times P_i)}{\sum_{j \in \text{all subnets}} (W_j \times P_j)} \times R_{\text{total}}$$

Meaning: Ensures fair resource allocation across subnets.
Parameters:
- W_i, P_i: Weight and performance of the member i.
- R_{total}: Total resources.
Example: Subnet has $\sum_{i \in \text{subnet}} (W_i \times P_i) = 30$, system total $\sum_{j \in \text{all subnets}} (W_j \times P_j) = 100$, $R_{\text{total}} = 1000$:

$$R_{\text{subnet}} = \frac{30}{100} \times 1000 = 300$$

8 Additional Mechanisms

8.1 DAO Voting Power

The voting power of a participant in the DAO (Decentralized Autonomous Organization) is calculated based on their stake and the duration of their staking:

$$\text{Voting Power}(p) = \text{stake}_p \times \left(1 + \frac{\text{time staked}_p}{\text{total time}}\right)$$

Meaning: This formula determines a participant's influence in DAO governance, rewarding long-term commitment by increasing voting power based on staking duration.

Parameters:
- stake_p: Stake of participant p.
- time staked_p: Duration participant p has staked their tokens.
- total time: Total time period considered for staking duration.

Example: Participant p has $\text{stake}_p = 1000$, $\text{time staked}_p = 5$ years, total time $= 10$ years. Calculate:

$$\text{Voting Power}(p) = 1000 \times \left(1 + \frac{5}{10}\right) = 1000 \times 1.5 = 1500$$

9 System Operation and Practical Example

9.1 System Operation

The Moderntensor system operates through a series of well-defined steps to ensure fairness, transparency, and efficiency in its decentralized AI platform. Below is a detailed breakdown of the process:

1. **Task Assignment**: Validators select miners based on their trust scores, with a fairness mechanism to favor those not recently selected. The selection probability is:

   $$\text{Selection Probability} = \text{trust score} \times (1 + \beta \times \text{time since last selection})$$

   where β (e.g., 0.2) is a bonus factor to promote equitable participation.

2. **Task Execution**: Selected miners perform tasks, such as training AI models, and submit their results to validators.

3. **Evaluation**: Validators assess submissions and assign performance scores, adjusted by their trust scores for fairness:

   $$P_{\text{miner adjusted}} = \frac{\sum_v (\text{trust}_{\text{score}_v} \times P_{\text{miner},v})}{\sum_v \text{trust}_{\text{score}_v}}$$

4. **Consensus**: Validators reach a consensus on miners' performance using a weighted average:

   $$\text{Average Score} = \frac{\sum_{v=1}^{n} (\text{trust}_{\text{score}_v} \times \text{Evaluation}_v)}{\sum_{v=1}^{n} \text{trust}_{\text{score}_v}}$$

5. **Trust Score Update**: Miners' trust scores are updated, with a decay for inactivity:

$$\text{Trust Score}_{\text{new}} = \text{Trust Score}_{\text{old}} \times e^{-\delta \times \text{time since last evaluation}} + \alpha \times \text{Score}_{\text{new}}$$

   where δ (e.g., 0.1) is the decay constant and α (e.g., 0.1) is the learning rate.

6. **Reward Distribution**: Rewards are distributed based on the incentive formula:

$$\text{Incentive}_{\text{miner}}(x) = \text{trust}_{\text{score}}(x) \times \frac{\sum_{j=0}^{m}(W_x \times P_{xj})}{\sum_{i=0}^{n}\sum_{j=0}^{m}(W_i \times P_{ij})}$$

7. **Blockchain Update**: Results (scores, rewards) are recorded on the Cardano blockchain via smart contracts, with off-chain computations for efficiency and hashes stored on-chain.

8. **DAO Governance**: Participants vote on system parameters using their voting power:

$$\text{Voting Power}(p) = \text{stake}_p \times \left(1 + \frac{\text{time staked}_p}{\text{total time}}\right)$$

9.2 Practical Example

Consider a scenario with 5 miners (M1 to M5) and 3 validators (V1 to V3). Initial trust scores are: M1: 0.9, M2: 0.8, M3: 0.7, M4: 0.6, M5: 0.5; V1: 0.9, V2: 0.8, V3: 0.7.

1. **Task Assignment**: Validators select miners. V1 picks M1 and M2 (selection probability for M1: $0.9 \times (1 + 0.2 \times 0) = 0.9$), V2 picks M1 and M3, V3 picks M2 and M4. M5 is not selected (probability: $0.5 \times (1 + 0.2 \times 1) = 0.6$, but not chosen).

2. **Evaluation**: Validators score the miners:
   - V1: M1 = 0.85, M2 = 0.9
   - V2: M1 = 0.9, M3 = 0.75
   - V3: M2 = 0.8, M4 = 0.65

3. **Adjusted Performance**: For M1:

$$P_{\text{miner adjusted, M1}} = \frac{(0.9 \times 0.85) + (0.8 \times 0.9)}{0.9 + 0.8} = \frac{0.765 + 0.72}{1.7} \approx 0.873$$

   Similarly, M2: 0.856, M3: 0.75, M4: 0.65, M5: 0 (not selected).

4. **Consensus**: For M1:

$$\text{Average Score}_{M1} = \frac{(0.9 \times 0.85) + (0.8 \times 0.9)}{0.9 + 0.8} \approx 0.873$$

5. **Trust Score Update**: For M5 (not selected, 1 cycle):

$$\text{Trust Score}_{\text{new, M5}} = 0.5 \times e^{-0.1 \times 1} + 0 \approx 0.5 \times 0.9048 \approx 0.452$$

   For M1:

$$\text{Trust Score}_{\text{new, M1}} = 0.9 \times e^{-0.1 \times 0} + 0.1 \times 0.873 = 0.9 + 0.0873 = 0.9873$$

6. **Reward Distribution**: Assume total system $\sum_{i=0}^{n}\sum_{j=0}^{m}(W_i \times P_{ij}) = 50$, $W_{M1} = 2$. For M1:

$$\text{Incentive}_{\text{miner}}(M1) = 0.9 \times \frac{2 \times 0.873}{50} \approx 0.9 \times \frac{1.746}{50} \approx 0.0314$$

7. **Blockchain Update**: Scores (e.g., M1: 0.873) and rewards (e.g., M1: 0.0314) are hashed and stored on-chain.

8. **DAO Governance**: Participant p with $\text{stake}_p = 1000$, $\text{time staked}_p = 5$ years, total time $= 10$ years:

$$\text{Voting Power}(p) = 1000 \times \left(1 + \frac{5}{10}\right) = 1500$$

This example demonstrates how the Moderntensor system applies its formulas to ensure fair task assignment, evaluation, reward distribution, and governance.