

## 1. TASKS

- Chinese Whispers, noiseless channel, 1 player
- Chinese Whispers, noiseless channels, 2 players
- Chinese Whispers, noiseless channels, n players
- Chinese Whispers, noisy input channel 10%, 1 player
- ...

## 2. INPUT STRINGS

Length of strings and number of characters must be specified.

- iid (uniformly distributed)
- iid (non-uniformly distributed)
- n-grams
- other distributions?
- English words/text?

## 3. INTERMEDIATE STRINGS

Length of strings and number of characters must be specified.

## 4. REWARD FUNCTIONS FOR THE LAST AGENT

Let  $n$  be the length of strings,  $I$  be the input string to the first agent,  $O$  be the output string of the last agent. Was Skalierung angeht bin ich mir nicht ganz sicher...

- "Hamming reward":  $r(I, O) = n - \text{Hamming}(I, O)$
- Levenshtein reward
- ...

## 5. REWARD FUNCTIONS FOR INTERMEDIATE AGENTS

Scale of this reward must be specified.

Let  $n$  be the length of strings,  $O$  be the output string of one intermediate agent.

- Zero:  $r(O) = 0$
- Shortness:  $r(O) = \#\text{NULLTOKENS}(O)$

## 6. NEURAL NETWORK BLOCKS

Intermediate dims etc. should be specified.

- 1-Convolution
- n-Convolution
- Fully Connected
- Self Attention

## 7. RL ALGORITHMS

Learning rates, batch size etc. should be specified.

- Policy Gradients (REINFORCE)
- Deep Deterministic Policy Gradient ?
- Actor Critic ?

## 8. LIST OF EXPERIMENTS

### 8.1. Experiment 1.

Chinese Whispers, noiseless channel, 1 player

Length 5, Number of characters 2, iid (uniformly distributed)

(No intermediate strings)

Hamming reward, mean-std-normalized as batch

(No intermediate agents)

Embedding / 1-Convolution; Total params: 10

REINFORCE, SGD(lr=0.01, momentum=0), batch size 64

Code: <https://github.com/Pausau/StillePost/blob/main/Experiment1.ipynb>

Results: Task fully solved once after  $\sim 6.26$  parameter updates.

### 8.2. Experiment 2.

Chinese Whispers, noiseless channel, 1 player

Length 5, Number of characters **5**, iid (uniformly distributed)

(No intermediate strings)

Hamming reward, mean-std-normalized as batch

(No intermediate agents)

Embedding / 1-Convolution; Total params: 55

REINFORCE, SGD(lr=0.01, momentum=0), batch size 64

Code: <https://github.com/Pausau/StillePost/blob/main/Experiment2.ipynb>

Results: Task fully solved once after  $\sim 22.62$  parameter updates.

### 8.3. Experiment 3.

Chinese Whispers, noiseless channel, **2** players

Length 5, Number of characters **2**, iid (uniformly distributed)

Intermediate: Length 5, Number of characters **2**

Hamming reward, mean-std-normalized as batch

Intermediate: No reward

Embedding / 1-Convolution; Params: 10 per player, total params: 20

REINFORCE, SGD(lr=0.01, momentum=0), batch size 64

Code: <https://github.com/Pausau/StillePost/blob/main/Experiment3.ipynb>

Results: Task fully solved once after  $\sim 18.52$  parameter updates.

### 8.4. Experiment 4.

Chinese Whispers, noiseless channel, 2 players

Length 5, Number of characters **3**, iid (uniformly distributed)

Intermediate: Length 5, Number of characters **3**

Hamming reward, mean-std-normalized as batch

Intermediate: No reward

Embedding / 1-Convolution; Params: 21 per player, total params: 42

REINFORCE, SGD(lr=0.01, momentum=0), batch size 64

Code: <https://github.com/Pausau/StillePost/blob/main/Experiment4.ipynb>

Results: 66% of the time, the task cannot be fully solved within 100 parameter updates. For the times where it was fully solved, the task was fully solved once after  $\sim 35.65$  parameter updates.

### 8.5. Experiment 5.

Chinese Whispers, noiseless channel, 2 players

Length 5, Number of characters **5**, iid (uniformly distributed)

Intermediate: Length 5, Number of characters **5**

Hamming reward, mean-std-normalized as batch

Intermediate: No reward

Embedding / 1-Convolution; Params: 55 per player, total params: 110

REINFORCE, SGD(lr=0.01, momentum=0), batch size 64

Code: <https://github.com/Pausau/StillePost/blob/main/Experiment5.ipynb>

Results: 100% of the time, the task cannot be fully solved within 150 parameter updates.