|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding by Artificial Intelligencewww.mpai.community |

|  |  |
| --- | --- |
| **N568** | 23/02/18 |
| **Source** | Requirements (NNW) |
| **Title** | Neural Network Watermarking initial use cases & functional requirements |
| **Target** | MPAI Community |

# Panorama of potential use cases

*Table 1: Use-cases list*

|  |
| --- |
| **USE CASES** |
| **Tracking Neural Network**  |
| UC.1 | *Ensuring the traceability of Neural Networks despite modifications*. The watermark is inserted before the distribution of the neural network and the owner shall be able to demonstrate ownership of the network by recovering the mark even if the network has been modified |
| UC.2 | *Create a digital notary for Neural Network*. By looking at the inference, it should be able to identify the owner of a Neural Network. |
| UC.3 | *Create a digital notary for Neural Network*. By looking at the neural network, it should be able to identify its owner. |
| **Detecting Neural Network alterations** |
| UC.4 | *Ensuring the integrity (modification or not) of Neural Network*. By looking at the network, it is possible to detect even a slight modification. |
| UC.5 | *Detecting the integrity (identifying the modification) of Neural Network*. By looking at the network, it is possible to detect the area that has been altered to significantly modify the performance of the network. |
| **Certifying Neural Network usage** |
| UC.6 | *Ensuring that the content was produced by a well identified Neural Network (creator side).* By looking at the inference, it is possible to identify that a Neural Network has been created by an identified owner. |
| UC.7 | *Ensuring that the content was produced by a well identified Neural Network (user side).* By looking at the inference, it is possible for a user to identify the owner of the neural network that produce it. |

## Tracking Neural Network

### Use case 1: *Ensuring the traceability of Neural Networks despite modifications*



The watermark is inserted before the distribution of the neural network and the owner shall be able to demonstrate ownership of the network by recovering the mark even if the network has been modified.

### Use case 2: Create a digital notary for Neural Network (black box)



By looking at the inference, it should be able to identify the owner of a Neural Network.

### Use case 3: Create a digital notary for Neural Network (white box)



By looking at the neural network, it should be able to identify its owner.

## Detecting Neural Network alterations

### Use case 4: Ensuring the integrity (modification or not) of Neural Network



By looking at the network, it is possible to detect even a slight modification.

### Use case 5: Detecting the integrity (identifying the modification) of Neural Network



By looking at the network, it is possible to detect the area that has been altered to significantly modify the performance of the network.

## Certifying Neural Network usage

### Use case 6: Ensuring that the content was produced by a well identified Neural Network (creator side)



By looking at the inference, it is possible to identify that a Neural Network has been created by an identified owner.

### Use case 7: Ensuring that the content was produced by a well identified Neural Network (user side)



By looking at the inference, it is possible for a user to identify the owner of the neural network that produce it.

# Functional requirements

**Watermarking techniques shall not affect the performances of the AI usage**

**The inserted watermark shall survive attacks such as:**

* Pruning: remove parameters of the network
* Quantization: store the parameters in less bits than original storage
* Operators’ approximations: multiplication is done with less precision (to save energy)
* Fine-tuning: the network is retrained which will affect parameters
* Transfer learning: the network is retrained on another task different than original one
* Random permutation of the layers’ parameters

**The watermark data payload is required to be a minimum of *D* bits and must contain specific information**

# Annex 1 – MPAI-NNW Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| White-Box  | Grant access to the network and make it possible for the watermark to be embedded inside the parameters. |
| Black-Box | Do not grant access to the network but to its output (result of an inference); in this case, the watermark information is inserted inside the output. |
| Imperceptibility | Prediction quality of the model on its original task should not be degraded significantly. |
| Robustness | Watermark should be robust against attacks. |
| Capacity | Allow for inclusion of a certain amounts of information, also called data payload. |
| Attacks | **Removal attacks**: the aim is to remove the watermark by extracting it or making it disappear. **Geometric attacks:** do not try to remove the mark, but rather to destroy their synchronization.**Cryptographic attacks**: detect and remove the mark without knowledge of the key, thanks to the knowledge of the embedded mark.**Protocol attacks**: the aim is to embed another watermark and create an ambiguous situation. |
| Fine-tuning | The network is trained on the same task.  |
| Transfer-learning | The network is trained on a similar task, for example on another dataset.  |
| Knowledge distillation | A surrogate model is trained by using the output of a well-trained network as label.  |
| Pruning | Is the process of removing parameters from a network: techniques can differ, but the aim is to keep performances while decreasing its computational cost? |
| Quantization | The parameters initially stored on 16 or 32bits are approximated by a set of discrete symbols or integer values. |
| Permutation | Let (S, S), D, and L respectively denote the size of the convolution filter, the depth of input to the convolutional layer, and the number of filters in the convolutional layer. **Neurons’ permutation:** We permute the filters, L dimension. **Channels’** **permutation:** We permute the channels, D dimension. We apply the same permutation to the filters (L) of the previous layer.  |
| Watermark overwriting | The attacker embeds another watermark to create an ambiguous situation where both parties can claim the ownership. |
| Watermark forging | The attacker tries to forge the watermark by creating its own trigger dataset. |
| Ownership metadata | The data carried by the watermark representing the owner and the usage conditions. |